T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

IPv6 AĞLARINDA GERÇEK ZAMANLI UYGULAMALARA AİT PAKETLERİN HEDEF AĞ PARAMETRELERİNE GÖRE ÖNCELİKLENDİRİLMESİ

DOKTORA TEZİ

Sadettin DEMİR

Enstitü Anabilim Dalı	**:**	**BİLGİSAYAR VE BİLİŞİM MÜHENDİSLİĞİ**
Tez Danışmanı	**:**	**Prof. Dr. İbrahim ÖZÇELİK**

Haziran 2021

T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

IPv6 AĞLARINDA GERÇEK ZAMANLI UYGULAMALARA AİT PAKETLERİN HEDEF AĞ PARAMETRELERİNE GÖRE ÖNCELİKLENDİRİLMESİ

DOKTORA TEZİ

Sadettin DEMİR

Enstitü Anabilim Dalı : BİLGİSAYAR VE BİLİŞİM MÜHENDİSLİĞİ

Bu tez 09/06/2021 tarihinde aşağıdaki jüri tarafından oybirliği ile kabul edilmiştir.

BEYAN

Tez içindeki tüm verilerin akademik kurallar çerçevesinde tarafımdan elde edildiğini, görsel ve yazılı tüm bilgi ve sonuçların akademik ve etik kurallara uygun şekilde sunulduğunu, kullanılan verilerde herhangi bir tahrifat yapılmadığını, başkalarının eserlerinden yararlanılması durumunda bilimsel normlara uygun olarak atıfta bulunulduğunu, tezde yer alan verilerin bu üniversite veya başka bir üniversitede herhangi bir tez çalışmasında kullanılmadığını beyan ederim.

Sadettin DEMİR

09.06.2021

TEŞEKKÜR

Doktora eğitimim boyunca değerli bilgi ve deneyimlerinden yararlandığım, her konuda bilgi ve desteğini almaktan çekinmediğim, araştırmanın planlanmasından yazılmasına kadar tüm aşamalarında yardımlarını esirgemeyen, teşvik eden, aynı titizlikte beni yönlendiren değerli danışman hocam Prof. Dr. İbrahim ÖZÇELİK'e teşekkürlerimi sunarım.

Bu çalışmanın ortaya çıkmasında yardımlarını esirgemeyen ve kodlama konusunda destek olan arkadaşım Ali Ben ZARROUK'a teşekkür ederim.

Anlayış ve desteğinden ötürü çok değerli eşim Semra DEMİR'e teşekkür ederim.

İÇİNDEKİLER

SİMGELER VE KISALTMALAR LİSTESİ

ABR	: Kullanılabilir bit iletim hızı
AD	: Yönetimsel uzaklık
AF	: Garantili iletim
AS	: Otonom sistem
ATM	: Eşzamansız iletim modu
ATM	: Eşzamansız iletim modu
BC	: Kararlı patlama boyutu
BE	: Aşırı patlama boyutu
BECN	: Geri hata sıkışıklık bildirimi
BGP	: Sınır geçit protokolü
CBR	: Sabit bit iletim hızı
CDV	: Hücre gecikme değişimi
CDVT	: Hücre gecikme değişim toleransı
CER	: Hücre hata oranı
CIR	: Taahhüt edilen bilgi oranı
CLR	: Hücre kayıp oranı
CMR	: Hücre yanlış yerleştirme oranı
CSCP	: Sınıf seçici kod noktaları
CTD	: Hücre iletim gecikmesi
DE	: Atmak için uygun
DiffServ	: Farklılaştırılmış servisler
DSCP	: Farklılaştırılmış hizmetler kod noktası
DSCP	: Farklılaştırılmış hizmetler kod noktası
DSP	: Dijital sinyal işlemcisi
EB	: Exabayt

EF : Hızlandırılmış iletim
EGPs : Harici ağ geçidi protokolleri
EIR : Genişletilmiş bilgi oranı
FECN : İleri hata sıkışıklık bildirimi
FIFO : İlk giren ilk çıkar
FL : Akış etiketi
GFR : Garanti edilmiş çerçeve hızı
GZU : Gerçek zamanlı uygulamalar
HSDPA : Yüksek hızda veri paketi indirme bağlantısı
ICI : Arayüz kontrol bilgisi
ICMP : İnternet kontrol mesaj protokolü
IETF : İnternet mühendisliği görev grubu
IGPs : Dahili ağ geçidi protokolleri
IntServ : Bütünleştirilmiş servisler
IP : İnternet protokolü
IPTV : İnternet protokolü üzerinde video
ISDN : Tümleşik hizmetler dijital ağı
IS-IS : Orta seviyeden orta seviyeye sistem
ISO : Uluslararası standartlar teşkilâtı
LSP : Etiket anahtarlamalı yollar
LSR : Etiket anahtarlamalı yönlendiriciler
MBS : En yüksek patlama boyutu
MCR : En düşük hücre iletim hızı
MFS : Maksimim çerçeve boyutu
MPLS : Çok protokollü etiket anahtarlama
nrt-VBR : Gerçek zamanlı olmayan değişken bit iletim hızı
NVP : Ağ ses protokolü
OSI : Açık sistem arabağlantı
OSPF : En kısa yola öncelikli
PCM : Darbe kod modülasyonu
PCR : En yüksek hücre iletim hızı
PHB : Atlama-başına davranış

PQ : Öncelik tabanlı kuyruk
QoS : Servis kalitesi
RIP : Yönlendirme bilgisi protokolü
RTI : Gerçek zamanlı toleranssız
RTT : Gerçek zamanlı toleranslı
rt-VBR : Gerçek zamanlı değişken bit iletim hızı
SCR : Sürdürülebilir hücre iletim hızı
SECBR : Ağır hataya uğramış hücre bloğu oranı
SPF : Önce en kısa yol
TC : Trafik sınıfı
TCP : İletim kontrol protokolü
TOS : Servis tipi
UBR : Belirlenmemiş bit iletim hızı
UDP : Kullanıcı veribloğu protokolü
VLAN : Sanal yerel alan ağları
VLAN : Sanal yerel alan ağları
VLSM : Değişken uzunlukta alt ağ maskeleme
VoIP : İnternet protokolü üzerinde ses
WAN : Geniş alan ağı
WFQ : Ağırlıklı adil kuyruk
XNS : Xerox ağ sistemleri

ŞEKİLLER LİSTESİ

TABLOLAR LİSTESİ

ÖZET

Anahtar kelimeler: Uçtan-uca QoS, IPv6 akış etiketi, öncelik kuyruğu, gerçek zamanlı uygulamalar.

İnternet mimarisinin ilk oluşmaya başladığı yıllarda ilk ve önemli amaç, verilerin paket anahtarlamalı ağlar üzerinden iletilmesiydi. Günümüzde internet teknolojisinin gelişmesiyle VOIP, IPTV ve video konferans gibi gerçek zamanlı uygulamalar arttıkça bu mimari yetersiz gelmeye başlamış ve servislerin ihtiyaç duyduğu QoS desteği için çeşitli yöntemler geliştirilmiştir. Ancak bu geliştirilen servislerin çoğu sadece tanımlı oldukları ağ içerisinde bu desteği kısmen verebilmiş uçtan-uca bir destek sağlayamamıştır.

Ses ve video gibi gerçek zamanlı servisler zaman duyarlı oldukları için bu servislerin kalitesini etkileyen en önemli faktör gecikmedir. Gecikme değerlerinin iyileştirilmesi servis kalitesini arttıracaktır.

IPv6 ise başlığında bulundurmuş olduğu Akış Etiketi alanı ile yeni bir yaklaşım getirmiştir. Ancak, bu alanın kullanımıyla ilgili kesin kurallar tanımlanmamıştır. Bu alanın kullanımı ile ilgili sadece yaklaşımlar mevcuttur.

Bu çalışma, ses ile video servislerin ve kullanıcıların ihtiyaç duyduğu uçta-uca servis kalitesinin sağlanması amacı ile gerçek zamanlı uygulamalara ait hedef ağın yönlendirme protokol metrik değerini bir önceliklendirme parametresi olarak kullanarak bir önceliklendirme modeli önermektedir. Önceliklendirme değerinin uçtan-uca taşınması için IPv6 Akış Etiketi alanı kullanılmıştır. Bunun için Akış Etiketi alanının yeni bir kullanım önerisi yapılmıştır ve öncelik tabanlı bir tampon bellek ve kuyruk modeli önerilmiştir.

Tasarlanan önceliklendirme modeli dört farklı senaryo üzerinde uygulanmış ve gerçek zamanlı uygulamalara ait veri paketlerinin gecikme değerleri üzerinde 13ms ile 40ms arasında bir iyileştirme gözlenmiştir. Bu değerler de %9,92 ile %35,13 arasında bir iyileştirme oranına karşılık gelmektedir.

PRIORITIZATION OF PACKAGES OF REAL-TIME APPLICATIONS ON IPv6 NETWORKS ACCORDING TO WITH DESTINATION NETWORK PARAMETERS

SUMMARY

Keywords: End-to-end QoS, IPv6 flow label, priority queue, real time applications

The first and most important goal in the years when internet architecture started to occur was to transmit the data over packet-switched networks. As real-time applications such as VOIP, IPTV and video conferencing increase with the development of internet technology today, this architecture has become insufficient and various methods have been developed for QoS support those services need. However, most of these developed services were only able to provide this support only partially in the network in which they were defined. These technologies could not provide end-to-end support.

Since real-time services are time sensitive, the most important factor affecting the quality of these services is delay. Improving delay values will increase service quality.

IPv6, on the other hand, has brought a new approach with its Flow Label field. However, the exact rules for the use of this area are not defined. There are only approaches to the use of this area.

This study proposes a prioritization model by using the routing protocol metric value of the destination network of the real-time applications as a prioritization parameter to provide end-to-end service quality required by the services and users. The IPv6 Flow Label field is used to move the prioritization value from end to end. For this, a new usage suggestion of the Flow Label field has been made and a priority-based buffer and queue model has been proposed.

The designed prioritization model has been applied on four different scenarios and improvement of 13ms to 40ms has been observed on the delay values of data packets of real-time applications. These values also correspond to an improvement rate between 9.92% and 35.13%.

BÖLÜM 1. GİRİŞ

1.1. İnternet ve Servis Kalitesi

Geleneksel olarak internet, paket anahtarlamalı ağlarda "best effort" olarak adlandırılan ilk gelen ilk çıkar yapısı ile oluşturulmuştur. İnternetin doğduğu yıllarda veri trafiği, günümüzde olduğundan kat ve kat az olduğundan dolayı internet mimarisinin tasarımında öne çıkan yaklaşım ve öncelik, hangi koşulda olursa olsun verinin iletilmesi olmuştur. Bu yapı içerisinde, kaynaktan hedefe taşınan kullanıcı trafiği için ne protokol ne de ağı oluşturan ağ aygıtları tarafından kullanıcıya verinin iletimi konusunda herhangi bir garanti verilmemiştir. İnternet üzerinde bir tıkanıklık meydana geldiğinde veya bant genişliği yetersiz olduğunda paketler düşürülür. Bunun yanında internet yapısı, taşınan verinin ne zaman teslim edileceği konusunda da yani zaman parametresini de değerlendirerek herhangi bir garanti sunmamaktadır. Bundan dolayı geleneksel internet mimarisi içerisinde veri paketlerinin servis kalitesinin arttırılması amacı ile önceliklendirilmesi mevcut değildir. Bu ağların tasarımında Türkçede gecikme, gecikme süresi, bant genişliği ve seğirme olarak isimlendirilen delay, latency, bandwidth ve jitter gibi faktörlerin önceliği bulunmamaktadır.

İnternet ağları üzerinde ilk ses transferi 1973 yılında ARPANET için deneysel olarak gerçekleştiren Ağ Ses Protokolü (NVP) aracılığı ile gerçekleştirilmesine rağmen 1995 yılında ilk internet telefon yazılımı (VocalTec) piyasaya sürülünceye kadar bir gelişim göstermemiştir. Vocaltec yazılımı, ses sinyallerini sıkıştırıp sayısal sinyale çevirdikten sonra internet üzerinden dağıtılmasını sağladığından dolayı ilk internet protokolü (IP) üzerinde çalışan telefonu temsil etmektedir [1].

Günümüzde internet üzerinde video akışını sağlayan en önemli sıkıştırma tekniği olan Ayrık Kosinüs Dönüşümü (DCT) algoritması ilk olarak 1974 yılında duyurulmasına

rağmen 1990 yılından sonra internetin hızla gelişmesiyle başlayan sürece kadar yavaş bir gelişim göstermiştir [2]. 1991 yılında günümüzdeki video anlayışından uzak bir şekilde Cambridge Üniversitesi'nin bilgisayar laboratuvarında bir kahve makinasının birkaç dakikada bir çekilen görüntülerin ardışık şekilde internete verilmesi internet üzerindeki ilk video olarak tarihe geçmiştir [3]. 1993 yılında, Xerox PARC'ta verilen bir konserin çoklu yayın yöntemi ve saniyede 8 ile 12 çerçeve şeklinde 152x76 piksel çözünürlük ile gerçekleşen yayını tarihin ilk canlı video yayınıdır [4]. Wax, 1993 yılında internet üzerinde yayınlanan ilk filmdir ve bant genişliği sorunlarından dolayı saniyede 2 çerçeve şeklinde yayınlanmıştır [5]. 1995 yılında ise Microsoft TV üzerinden New York Yankees ve Seattle Mariners arasında oynanan beyzbol maçı canlı olarak internet üzerinden yayınlanmıştır [6]. 2005 yılında Youtube üzerinde yayınlanan ilk video da bu konudaki önemli adımlardan biri olmuştur.

1998 yılına kadar ABD'de oluşan tüm ses trafiğinin %1'i internet protokolü üzerinde ses (VoIP) şeklinde gerçekleşirken 2000 yılında bu oran %3'e çıkmıştır [1]. 2007'den sonra hızla büyüyen VoIP trafiği 2017 yılında toplam 1 milyar kullanıcıya ulaştığı tahmin edilmektedir [7].

Cisco System tarafında yayınlanan "Cisco Visual Networking Index: Forecast and Methodology, 2016–2021" isimli raporu ise video trafiği konusunda çok çarpıcı veriler sunmaktadır. 2015 yılında internet trafiğinin %70'ini oluşturan video trafiğinin 2021 yılında %82'ye ulaşacağı tahmin edilmektedir. Bu oluşan video trafiğinin %15'i ise canlı video yayınları olacağı öngörülmektedir. Bu rapor ayrıca, 2021 yılında aylık video trafiğinin 37 exabayta (EB) büyüklüğünde olacağını belirtmektedir.

20. yüzyılın son yıllarında başlayan ve 2000'li yılların başlangıcından günümüze kadar olan süreçte kullanıcıların geniş bant internet imkânı arttıkça ve internetin gelişmesiyle ses ve video gibi gerçek zamanlı uygulamalara (GZU) olan talep de artmıştır ve bu artış devam etmektedir. GZU'ların en önemli özelliği bu uygulamalara ait paketlerin belirlenen süre içerisinde hedefte olması istenmektedir. Böylece, bu uygulamalar internetin doğal yapısında bulunmayan zaman parametrelerinin öneminin ortaya

çıkmasına neden olmuştur. Bu sürecin sonucu olarak, servis kalitesi (QoS) terimi anlamlı bir kavram olmaya başlamıştır.

QoS, var olan ağın bir yetenek unsurunu belirtir ki bu unsur, taşınmakta olan kullanıcı veri trafiğinin belirli bir güvence ve tutarlılık dahilinde taşındığına dair kullanıcıya kesin bilgiler verir. QoS, trafik tiplerinin ayrımını sağlayan ve ağ boyunca ayrımı yapılan trafiği gerekli olan desteği verebilmek için farklı davranışlar sergileyen bir mekanizma olarak tanımlanabilir [8]. QoS mekanizmaları, kullanıcı veya uygulamanın ihtiyaçları doğrultusunda ağdan talep edilen kesin isteklerine garanti verir. Böylece garanti talebinde bulunulan iletimi ayrı bir sınıf veya akış içerisinde değerlendirir. Bunun yanında, bu paketlerin başarılı iletimi için gerekli olan ağ kaynaklarının kontrol ve yönetimi konusunda da ağ yöneticisine yardımcı olur. Bunun sonucu olarak QoS mekanizmaları, bant genişliği, gecikme, seğirme ve paket kaybı gibi parametreler dikkate alınarak rezerve edilen ağ kaynaklarına ihtiyaçlar doğrultusunda müdahale edebilme imkânı verir.

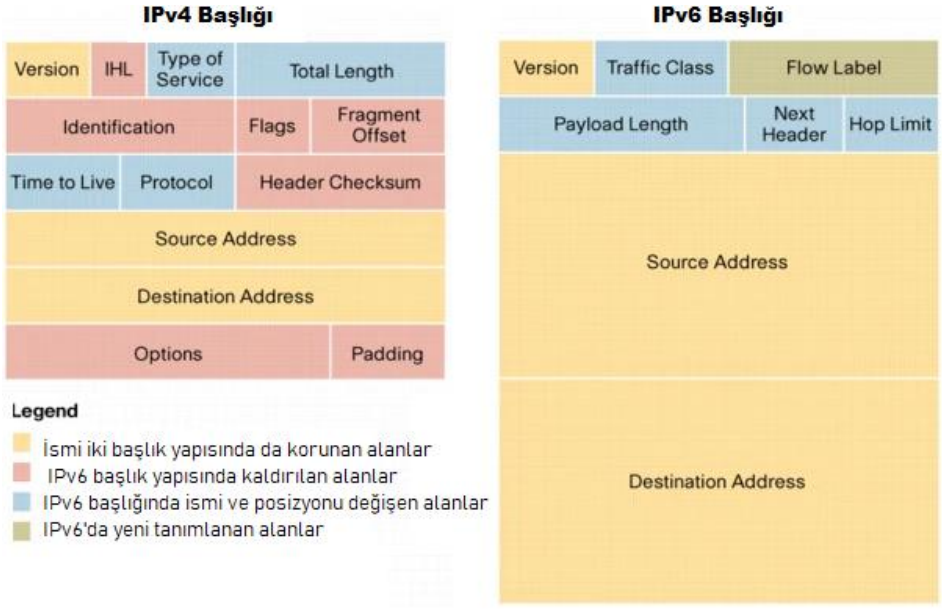
VoIP ve internet protokolü üzerinde video (IPTV), verilerinin iletimi gerçek zamanlı olması gerekmektedir. Bu sistemlerde, örnek alınarak oluşturulan ses ve video sinyalleri önce sayısal hale getirilir ve IP paket formuna dönüştürülerek paket anahtarlamalı ağlar üzerinden gönderilir. Bu ağlarda, yönlendirme protokolü sayesinde hedefe doğru en uygun yol bulunur. Ancak, kaynaktan yola çıkan RTA'ya ait paketler farklı yollar üzerinden hedefe gittiği için gecikmeli olarak ulaşabilir veya ağ problemlerinden dolayı kayıp olabilir. Kayıp GZU paketlerinin, alıcı tarafından sistem yapısından dolayı yeniden istenmesi mümkün değildir ve hedefe farklı zamanlarda ulaşan paketlerinin sıralanması, birleştirilmesi ve gerektiğinde tekrar analog sinyallere dönüştürülmesi gerekir [9]. Ancak buradaki problem bu işlemlerin belli bir zaman sınırı içinde yapılması gerekliliğidir. Bu işlemlerin yapılması için tanımlanan sınır değerlerin üzerinde bir zaman alması senkronizasyonun bozulmasına ve kalitenin düşmesine neden olacaktır. Ses ve video paketleri için bu değerler Tablo 1.1.'de verilmiştir [10].

Tablo 1.1. Ses ve video uygulamaları için QoS değerleri

	Ses	Etkileşimli Video	Akıcı Video
Kayıp	<%1	≤%1	≤%5
Tek yönlü gecikme	≤150ms	≤150ms	≤5s
Seğirme	≤30ms	≤30ms	--

Ancak IP paketleri halinde taşınan ses ve video bilgisi için geleneksel internet mimarisi içinde herhangi bir servis kalitesi tanımlı değildir. Bundan dolayı gerekli olan servis kalitesi ek yöntem, teknoloji ve protokollerle sağlanmaya çalışılmıştır.

Altı farklı servis sınıfı sunan Eşzamansız İletim Modu (ATM), ağ üzerinde tıkanıklık kontrolü sağlayan Frame Relay, Sanal Yerel Alan Ağları (VLAN) üzerinde önceliklendirme ve ayrı davranışa tabi tutmak amacı ile çerçeve etiketleme yeteneğine sahip 802.1Q teknolojileri Açık Sistem Arabağlantı (OSI) ağ modelinin ikinci katmanında yani veri-bağı katmanında tanımlıdır. Ağ katmanı ve veri-bağı katmanı teknolojilerinin karışımı bir çözüm sunmak için geliştirilen Çok Protokollü Etiket Anahtarlama (MPLS) teknolojisi de çerçeve etiketleme yeteneğine sahiptir. Ancak bu teknolojiler OSI modelinin ikinci katmanında tanımlı olduklarından dolayı sadece tanımlı oldukları ağ üzerinde çalışmaktadırlar ve kaynaktan hedefe yani uçtan-uça bir destek sunmaları mümkün değildir.



Şekil 1.1. IPv4 ve IPv6 başlık yapısı

Günümüzde kullanılan ve Şekil 1.1.'de gösterilen OSI modelinin üçüncü katmanında tanımlı IPv4 ve IPv6 protokol başlığı içerisinde bulunan Servis Tipi (TOS) ve Trafik Sınıfı (TC) alanlarını oluşturan Farklılaştırılmış Hizmetler Kod Noktası (DSCP) bitleri sayesinde GZU'ya ait veri paketleri işaretlenebilmektedir. Bu bit değerlerine göre Bütünleştirilmiş Servisler (IntServ) ve Farklılaştırılmış Servisler (DiffServ) teknolojileri işaretlenen GZU'ya ait veri paketlerini servis kalitesi desteği için ayrı davranışa tabi tutulabilmektedir.

GZU'ların örnekleri olan ses ve video uygulamasına ait veri paketleri de yukarıda anlatılan şekilde işaretlenip QoS desteği için ayrı davranış modelleri içerisinde değerlendirilmektedir. Ancak burada sorun şudur ki; bütün bu gerçek zamanlı uygulama paketleri hedef gözetmeksizin tek bir grup olarak işaretlenmekte ve aynı davranış modeli içinde değerlendirilmektedir. Bu projede amaçlanan hedef, bu soruna çözüm bulmaktır. Bu çalışma, GZU olan ses ve video paketlerinin hedef ağ metrik değeri ile işaretlenerek, bu öncelikli paketlerin tek bir davranış modeli içerisinde değil her birinin hedef gözetilerek ayrı davranışa tabi tutulacak şekilde önceliklendirilmesini amaçlamaktadır.

IPv4 başlık yapısı içerisinde bulunmayan Akış Etiketi (FL) alanı, Şekil 1.1.'de gösterildiği gibi IPv6 başlığı içerisinde tanımlanmıştır ve amacının bir düğüm tarafından bir akışa ait paketlerin etiketlenmesi olarak belirtilmiştir. Bir paket akışı için FL alanı kullanarak bir trafik tanımlaması yapılırsa, yönlendiriciler bu paket rtafiğini kolaylıkla tanıyabilirler ve özel olarak işleyebilirler. Böylece trafik tarafından ihtiyaç duyulan servis kalitesi, kullanılan IPv6 protokolü tarafından sağlanmış olur [11]. Bundan dolayı, uçtan-uca servis desteği için bu çalışmada IPv6 protokolü ve GZU paketlerinin hedef parametresine göre işaretlenmesi ve bunun da önceliklendirme değeri olarak kullanılması amacı ile FL alanı kullanılacaktır.

GZU paketlerinin kendi içerisinde sınıflandırmak için kullanılacak hedef parametresi, ilgili paketin hedefine ait yönlendirme protokol metrik değerleridir. Bu paketlerinin gidecekleri hedef ağlara ait metrik değerleri yönlendiriciler üzerinde, yönlendirme protokolleri tarafından tutulmaktadır. GZU çalıştıran bir bilgisayar, bu uygulamalara

ait paketleri işleme süreci sırasında TOS veya TC alanını kullanarak bu paketin bir GZU paketi olduğunu işaretlemektedir. Paket hedef ağa doğru ilerleme sırasında ilk uğradığı sınır yönlendirici tarafından hedef ağı belirlemek için zaten IP başlığını okumaktadır. Hedef ağ belirlendikten sonra bu ağa ait metrik değer yönlendirici tablosundan alınarak FL alanına yerleştirilecek ve hedefe doğru ilerlemekte olan paket uğramış olduğu her bir yönlendirici tarafından bu metrik değere göre önceliklendirip işleme tabi tutulacaktır.

1.2. Çalışmanın Amacı ve Önerilen Çözüm Yöntemi

GZU servisleri, verilerinin belirli zaman aralıklarında alıcıya ulaşmasını istemektedirler. Uygulamaların kaliteli ve sağlıklı çalışmasını engelleyen en büyük faktör gecikme faktörüdür. Ancak görülmektedir ki sistemler üzerinde gecikme hiçbir zaman sıfıra indirgenemez. Bunun yanında sabit gecikmeler üzerinde de çok ciddi yapısal değişiklikler yapılmadan düşürmek mümkün görünmemektedir.

Tüm bunların ışığında araştırmacıların en yoğunlaştıkları konu değişken gecikmelerin önüne geçmek veya mümkün olduğunca bu değerleri düşürmek olmuştur. Değişken gecikmelerin kaynağı ağ üzerindeki yayılım gecikmesi ve tamponlama/kuyruklama gecikmesi olduğundan dolayı araştırmaların büyük bir bölümü bu alanda yapılmıştır. Bundan dolayı farklı kuyruk yapılarının GZU üzerindeki performansları araştırılmış [12-17], hibrit kuyruklama teknikleri denenmiş [14], ancak yapılan çalışmalarda hedef ve hedefin durumu dikkate alınmamıştır. Bu çalışma literatürde örneği olmayan bir yaklaşımla GZU'lara ait gecikme değerini düşürmek için hedefe ait yönlendirme parametresini kullanacaktır ve bu parametreyi dikkate alan bir kuyruk tasarımı yapılacaktır.

Çalışmamız da GZU'lara ait paketlerin yönlendirme protokollerin kullandığı uzaklık (distance) parametresine göre FL yardımı ile önceliklendirilmesi ve bu önceliklendirilmiş paketlerin mümkün olan en kısa zamanda gönderilmesini sağlayacak bir kuyruk yönetimini kapsamaktadır. Bu önceliklendirme modeli

sayesinde gerçek zamanlı uygulamalara ait paketlerin gecikme değerlerini iyileştirmeyi amaçlamıştır.

Burada DiffServ mantığı üzerinden gidilerek, önceliklendirmede kullanılacak her bir metrik değeri için ayrı davranış modeli oluşturulmayacak, her bir paket önceden belirlenmiş olan davranış modeli sınıflarından birine atanacaktır. Böylece yönlendiriciler üzerlerinde her bir metrik değeri için ayrı durum bilgisi tutmak yerine sınırlı sayıda durum bilgisi tutarak işlem gecikmesi en az düzeyde tutan bir algoritma geliştirilecektir.

Bu algoritma öncelikle best effort trafik ile GZU trafiğini, paket içeriğindeki DSCP bitlerine bakarak ayırt edecektir. Eğer paket best effort trafiğe ait ise ilgili kuyruğa gönderecektir. Eğer paket GZU trafiğine ait ise bu paketin hedef ağını tespit edecek, hedef ağa ait yönlendirme metrik değerini yönlendirme tablosundan alacak ve bu değeri önceliklendirme değeri olarak kullanacak şekilde FL alanı içerisine yazacaktır. GZU paketi önceliklendirme değeri olarak kullanılan metrik değerine ilgili önceliklendirme kuyruğuna yerleştirilecek ve paketin öncelikli olarak gönderilmesi sağlanacaktır. Paket bir kere önceliklendirildikten sonra diğer yönlendiriciler bu değer üzerinden paketi direk olarak ilgili kuyruğa alıp gönderim işlemini yerine getireceklerdir.

Önerilen önceliklendirme modeline ait algoritmanın simülasyonu ve başarım testleri OPNET programı üzerinde gerçekleştirilmiştir. OPNET, iletişim ve haberleşme sistemlerinin modellenmesinde kullanılan bir benzetim (simülatör) programıdır. Modellenen sistemlerin, ayrık olay (discrete-event) yöntemi ile davranış ve başarım analizleri gerçekleştirilir. OPNET'in bu temel özelliklerinin bir sonucu olarak çok çeşitli sistem modellemelerinin gerçekleştirilmesi amacıyla bir platform olarak kullanılmaktadır. Bunun doğal sonucu olarak liretatürde bulunan birçok araştırmada kullanılmıştır [18].

Tez çalışması temelde iki adımdan oluşmaktadır. İlk adımın amacı, geliştirilen algoritmanın başarısını ölçmek amacı ile referans bir topoloji oluşturmaktır. Bunun

için daha önceden çalışılmış örnek topolojiler referans alınmıştır [19-20]. Bu referans topoloji üzerinden alınan başarım değerleri ikinci adımda uygulanacak önceliklendirme algoritmasının başarım değerleri ile karşılaştırmak için kullanılacaktır. Burada gerçek hayatta karşılaşabileceğimiz her türlü durum göz önüne alınarak hem düşük metrik değerine sahip hem de yüksek metrik değerine sahip hedef ağlara ait istatistiki veriler toplanacaktır.

İkinci adımda ise, referans topoloji üzerinde tasarlanan önceliklendirme algoritması uygulanacak ve alınan sonuçlar birinci adımdaki referans topolojiden alınan sonuçlar ile karşılaştırılarak geliştirilen önceliklendirme algoritmasının başarımı ölçülmeye çalışılacaktır.

1.3. Tez Organizasyonu

Bu tez çalışması 5 bölümden oluşmaktadır. Tezin 2. bölümünde tezin alt çalışma konuları olan servis kalitesi, IPv6 protokolü ve yönlendirme başlıklarında temel kavramlar açıklanacak ve bu konular üzerinde yapılan akademik çalışmalardan bahsedilecektir.

Tezin 3. bölümünde ise, önerilen yönlendirme protokolü metrik tabanlı önceliklendirme modelinde alınan sonuçların karşılaştırılması için kullanılan referans topolojinin oluşturulması ve önceliklendirme modelinin uygulanması anlatılacaktır.

Tezin 4. bölümünde referans topoloji ve önerilen önceliklendirme modelini içeren topoloji üzerinden alınan sonuçlar karşılaştırılmalı olarak verilecek ve önceliklendirme modelinin başarısı irdelenecektir.

Tezin son bölümünde ise GZU sistemler için önerdiğimiz yönlendirme protokolü metrik tabanlı önceliklendirme modelinin literatüre katkısı, uygulamadan çıkarılan sonuçlar ve gelecekte konu üzerinde yapılacak çalışmalar anlatılacaktır.

BÖLÜM 2. TEMEL BİLGİLER

Tezin bu bölümü, bu tez çalışmasına temel teşkil eden ve tez içinde kullanılan servis kalitesi, IPv6 ve yönlendirme olmak üzere 3 alt başlık halinde anlatılacaktır. Önce servis kalitesi ve önemi açıklanarak servis kalitesini oluşturan parametreler ve servis kalitesine etki eden faktörler ele alınacaktır. Sonrasında ise, servis kalitesini arttırmak için kadar uygulanan çözüm yöntemlerine değinilecektir. Yeni nesil internet protokolü olan IPv6 ve bu protokolün servis kalitesine getirmiş olduğu yenilikler ve FL konuları açıklanarak detaylandırılacak ve son bölümde yönlendirme ve yönlendirme protokollerine değinilecek ve yapılan akademik çalışmalardan bahsedilecektir.

2.1. Servis Kalitesi (QoS)

Günümüzde internet artık veri, ses ve multimedya trafiği taşıyan bir ortam haline gelmiştir. GZU uygulamaları, geleneksel yöntemlerden oldukça farklılaşmış uygulamalardır ve GZU olan veya olmayan trafik servislerine farklı seviyelerde ihtiyaç duyarlar [21]. Geleneksel internet mimarisi üzerinde, önceliklendirme ve zaman parametreleri kullanılmadığından GZU veri trafiğini taşımak için uygun değildir. Ancak günümüzde internet trafiğinin büyük bir bölümünü oluşturan ses ve video gibi GZU verileri, ağ üzerinde bir tıkanıklık olduğunda veya bir gecikme meydana geldiğinde kullanıcıya verinin teslimatı konusunda belirli güvencelerin verilmesini istemektedir.

Kesin bir QoS, trafik servislerinin farklı seviyelerde olması ile karşılanır ve bu da kullanıcının almış olduğu servis için istemiş olduğu parametrelerin karşılanmış olduğundan emin olmasını sağlar. İnternet üzerinde her geçen gün çeşitli ve farklı ihtiyaçları olan kullanıcı sayısı gittikçe artmaktadır. QoS ise kullanıcıların ihtiyaç

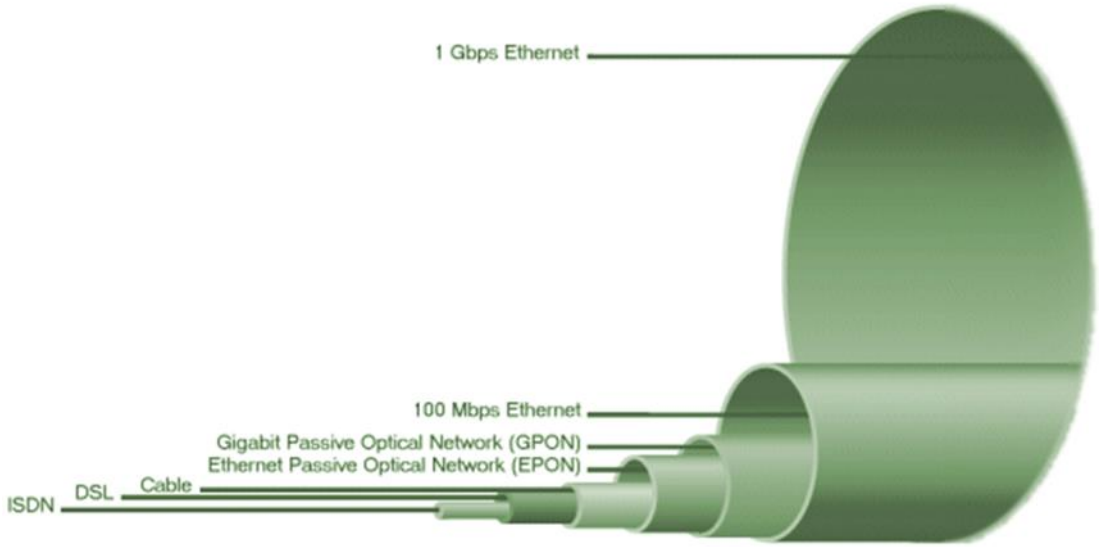
duydukları yük, gecikme ve seğirme gibi parametrelerin tutarlı ve ihtiyaçlar doğrultusunda talep edilen kadar olmasını sağlar [21]. QoS ağ üzerinde bir tıkanıklık meydana gelmesi durumunda hassas trafik için gecikmeyi önleyici akıllı mekanizmalar sunar [22].

2.1.1. Servis kalitesi parametreleri

Ağın QoS açısından değerlendirilmesi için bant genişliği, gecikme, tampon gereksinimleri, paket kayıpları ve seğirme gibi parametreler kullanılmaktadır [21].

2.1.1.1. Bant genişliği

Bir hattın bant genişliği, bu hat üzerinde transfer edilen veri oranıdır ve bu birim zaman içerisinde değerlendirildiğinde yaygın tabirle saniyedeki iletilen bit sayısıdır. Daha yüksek bir bant genişliği, daha yüksek transfer oranı ve verinin transferi için daha yüksek kapasite sunar. Bant genişliği, iletilecek çok fazla veri olduğu durumlarda, düşük bant genişliğine sahip hatların neden olduğu tıkanıklık, paket kaybı ve paket iletim gecikmesi gibi olguların meydana geldiği durumlarda QoS değerlendirmesini sağlayan önemli bir faktördür [23].



Şekil 2.1. İnternet erişim teknolojilerinin karşılaştırmalı bant genişliği

2.1.1.2. Gecikme süresi

Gecikme süresi, bir veri paketinin kaynaktan hedefe taşınması sırasında geçen toplam süre olarak tanımlanabilir. Gecikme süresi, yayılım gecikmesi, iletim gecikmesi, işlem gecikmesi ve diğer kaynaklı gecikmeler gibi tüm gecikmelerin bir kombinasyonudur. Genellikle kaynaktan hedefe ve hedeften geriye yani kaynağa hareket eden paket için alınan süre olarak ifade edilir ki bu değer gidiş-dönüş süresi (RTT) olarak bilinir [24]. Bu parametre, uçtan uca QoS sunarken göz önünde bulundurulması gereken önemli bir parametredir ve QoS tarafından yol boyunca üzerinden geçilen noktalar arasında en az olacak şekilde olması sağlanmalıdır.

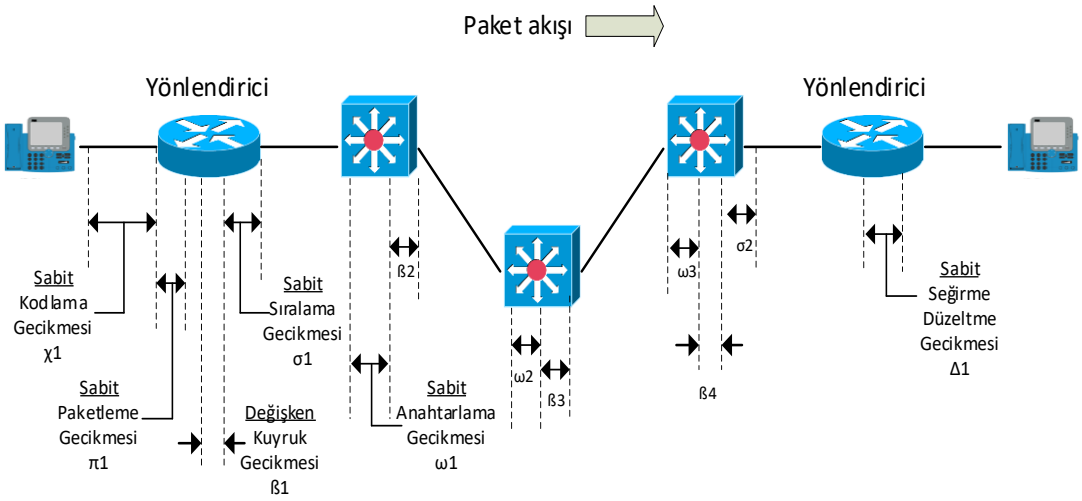
2.1.1.3. İşlem gecikmesi

Paket iletimindeki gecikmeler GZU trafik kalitesini etkiler. GZU, Gerçek Zamanlı Toleranslı (RTT) ve Gerçek Zamanlı Toleranssız (RTI) olmak üzere iki grup altında kategorize edilebilir. Video konferans ve multimedya trafiği RTI grubuna örnek olarak verilebilir. Bu trafiğe ait paketlerin ağ üzerindeki hareketlerinde en az gecikme gerektirir. Ses akışı gibi RTT grubuna ait bir veri iletimi oturumu üzerinde meydana gelen küçük bir gecikme, ağ üzerinde bulunan akıllı cihazlar tarafından sunulan tampon belleğe alma gibi özellikler sayesinde giderilebilir [24]. Yeterli bir QoS desteği için, gecikme minimal düzeyde olmalıdır.

Ağ üzerinde hareket etmekte olan ses paketlerini etkileyen iki tür gecikme mevcuttur.

- Sabit gecikme
- Değişken değişme

Sabit gecikme, ağ üzerindeki toplam gecikmeye direk olarak eklenen bir bileşendir. Değişken gecikmeler ise, çıkış sağlayan seri portlar üzerindeki çıkış tamponlarında bulunan kuyruk gecikmelerinden kaynaklanmaktadır. Bu tampon yapıları ağ üzerinde seğirme adı verilen değişken gecikmeler oluşturmaktadır [25].



Şekil 2.3. Ağ üzerinde gerçekleşen gecikme kaynakları

2.1.1.3.1. Kodlama gecikmesi

Kodlama gecikmesi, ses ve video sinyalinden alınan Darbe Kod Modülasyonu (PCM) örneklerinin bir bloğunun Dijital Sinyal İşlemcisi (DSP) tarafından sıkıştırılmak için harcanan zamandır. Bu olay aynı zamanda işlem gecikmesi ($\varkappa_n$) olarak da adlandırılmaktadır. Bu gecikme süresi kullanılan kodlayıcı ve işlemci hızı ile değişmektedir.

Tablo 2.1. En iyi ve en kötü durumlarda farklı algoritmalara ait kodlama gecikme süreleri

Kodlayıcı	Oran	Gerekli Örnek Bloğu	En İyi Durum Kodlayıcı Gecikmesi	En Kötü Durum Kodlayıcı Gecikmesi
ADPCM, G.726	32 Kbps	10 ms	2.5 ms	10 ms
CS-ACELP, G.729A	8.0 Kbps	10 ms	2.5 ms	10 ms
MP-MLQ, G.723.1	6.3 Kbps	30 ms	5 ms	20 ms
MP-ACELP, G.723.1	5.3 Kbps	30 ms	5 ms	20 ms

Ses bloklarının çerçeve yerleştirilmesi (decompression) sırasında geçen süre de yaklaşık olarak sıkıştırma süresinin %10'u kadardır. Bununla birlikte bu süre çerçeve başına düşen örnek sayısı ile doğru orantılıdır. Genellikle, G729 algoritması kullanıldığında tek bir çerçeve içerisine sıkıştırılmış iki veya üç blok yerleştirilirken, G723.1 kullanıldığında bir çerçeve içerisine bir sıkıştırılmış örnek yerleştirilir. Tablo

2.1. bazı algoritmaların en iyi ve en kötü durumlarda kodlama gecikme sürelerini ifade etmektedir.

2.1.1.3.2. Paketleme gecikmesi

Paketleme gecikmesi (π_n), kodlanmış ve sıkıştırılmış veri ile bir paket yükünü (payload) doldurmak için geçen süredir. Bu gecikme, tek bir çerçeve içerisinde bulunan blokların sayısı ile kodlayıcı tarafından gerekli olan örnek blok boyutunun bir fonksiyonudur. Aynı zamanda birikim gecikmesi (accumulation delay) olarak da adlandırılan bu gecikme, yayınlanmadan önce veri örneklerinin tampon bellek içerisinde birikme süresidir. Farklı kodlayıcılara ait paketleme gecikmesi değerleri Tablo 2.2.'de verilmiştir.

Tablo 2.2. Yük boyutuna göre paketleme gecikmesi değerleri

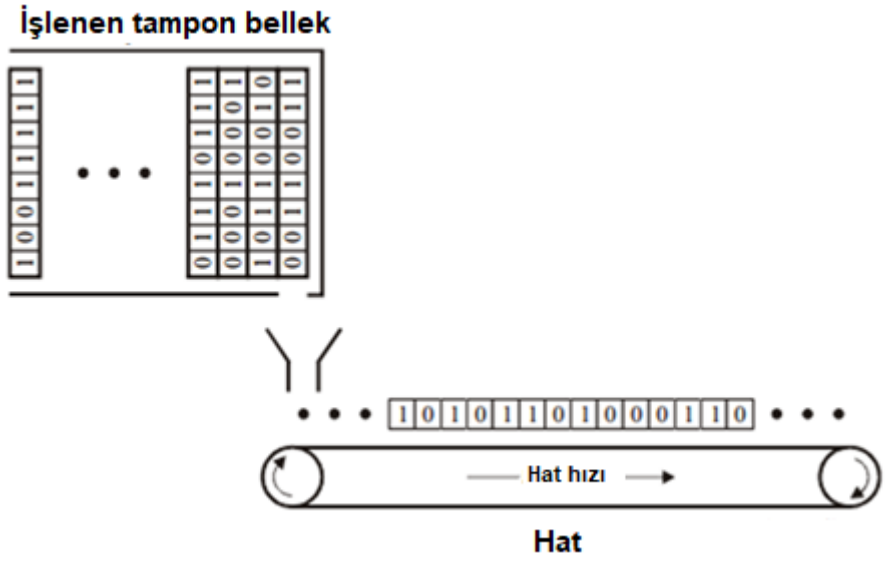
Kodlayıcı	Oran	Yük boyutu (Bayt)	Paketleme gecikmesi (ms)	Yük boyutu (Bayt)	Paketleme gecikmesi (ms)
PCM, G.711	64 Kbps	160	20	240	30
ADPCM, G.726	32 Kbps	80	20	120	30
CS-ACELP, G.729	8.0 Kbps	20	20	30	30
MP-MLQ, G.723.1	6.3 Kbps	24	24	60	48
MP-ACELP, G.723.1	5.3 Kbps	20	30	60	60

2.1.1.3.3. Sıralama gecikmesi

Sıralama gecikmesi (σ_n), ağ arabirimi üzerinde ses veya video çerçevesi için saat hızına bağlı olan sabit bit gecikmesidir. Direk olarak saat hızına bağlıdır. Tablo 2.3. farklı hat hızlarında farklı çerçeve boyutları için gerekli olan sıralama gecikmelerini göstermektedir. Bu tablonun hesaplanmasında toplam yük (payload) boyutu değil çerçeve boyutu dikkate alınmıştır. Şekil 2.3. sıralama gecikmesi gösterimini ifade etmektedir.

Tablo 2.3. Farklı çerçeve boyutlarına göre sıralama gecikmesi değerleri

Çerçeve Boyu (bayt)	Hat Hızı (Kbps)										
	19.2	56	64	128	256	384	512	768	1024	1544	2048
38	15.83	5.43	4.75	2.38	1.19	0.79	0.59	0.40	0.30	0.20	0.15
48	20.00	6.86	6.00	3.00	1.50	1.00	0.75	0.50	0.38	0.25	0.19
64	26.67	9.14	8.00	4.00	2.00	1.33	1.00	0.67	0.50	0.33	0.25
128	53.33	18.29	16.00	8.00	4.00	2.67	2.00	1.33	1.00	0.66	0.50
256	106.67	36.57	32.00	16.00	8.00	5.33	4.00	2.67	2.00	1.33	1.00
512	213.33	73.14	64.00	32.00	16.00	10.67	8.00	5.33	4.00	2.65	2.00
1024	426.67	149.29	128.00	64.00	32.00	21.33	16.00	10.67	8.00	5.31	4.00
1500	625.00	214.29	187.50	93.75	46.88	31.25	23.44	15.63	11.72	7.77	5.86
2048	853.33	292.57	256.00	128.00	64.00	42.67	32.00	21.33	16.00	10.61	8.00



Şekil 2.4. Sıralama gecikmesi gösterimi

2.1.1.3.4. Kuyruk gecikmesi

Sıkıştırılmış ses veya video yükü oluşturulduktan sonra bir başlık eklenerek çerçeve haline getirilir ve ağ bağlantısı üzerinden iletilmesi için kuyruğa alınır. GZU, gecikme hassasiyetinden dolayı yönlendirici veya ağ geçidi üzerinde önceliklendirme yapılmalıdır. Bir ses/video çerçevesi akışı devam etmekte olan bir veri çerçevesini veya kendinden önce bulunan diğer ses/video çerçevelerini beklemek zorundadır. Böylece kuyruğa gelen çerçeve, çıkış kuyruğundaki kendinden önceki çerçevelerin sıralama gecikmesi kadar bekler. Kuyruk gecikmesi (βn) değişken bir gecikmedir ve

hattın hızı ile kuyruğun durumuna bağlıdır. Kuyruk gecikmesi rastgele değişkenlerle ilişkilidir.

2.1.1.3.5. Ağ anahtarlama gecikmesi

Son noktaları bağlayan birbirine bağlayan açık ağlar ses ve video bağlantıları için en büyük gecikme kaynaklarıdır. Ağ anahtarlama gecikmesi (ω_n) ölçülmesi en zor olan gecikmedir.

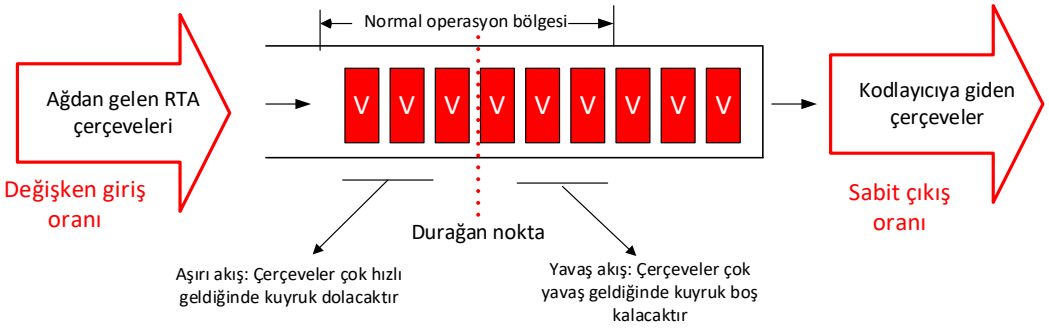
Genel olarak anahtarlama gecikmesini, ağ içerisindeki hatlarda sabit olan yayılım gecikmesi (propagation delay) ve ağ içerisindeki anahtarlama aygıtlarının girişlerindeki ve çıkışlarındaki saat hızına bağlı olarak değişken özellik gösteren kuyruk gecikmesi oluşturur. Özel bir ağ içerisinde kuyruk gecikmesini ölçmek mümkün olabilir veya geniş alan ağlarında her bir düğüm başına oluşabilecek değer tahmin edilebilir. Yayılım gecikmesinin hesaplanmasında G.114 önerisinde belirtilen değerler yaygın olarak kullanılmaktadır ve bu değerler 10 ms/mile ve 6 ms/km olarak belirlenmiştir. Bununla beraber ortamdaki çoğullama (multiplexing) aygıtları, mikrodalga hatlar ve diğer faktörler taşıyıcı hatlar üzerinde özel durumlar oluşturabilir.

2.1.1.3.6. Seğirme düzeltme gecikmesi

Ses ve video süreklilik arz eden bir bit-rate servistir ve tek bir iletime ait paketler arasındaki gecikme değişimi olarak tanımlanan, tüm değişken gecikmelerin neden olduğu seğirme, sinyal ağı terk etmeden önce ortadan kaldırılmalıdır. Bundan dolayı alıcı pozisyonundaki cihazlar seğirme düzeltici (Δn) tampon bellek ile yapılandırılmışlardır. Bu tampon almış olduğu veriyi işlenmek üzere son noktaya göndermeden önce belli bir süre üzerinde tutar. Böylece seğirme düzeltici tampon bellekler sayesinde değişken gecikmeler sabit gecikmeye çevrilir.

Değişken gecikmeden kaynaklanan ve bu gecikmenin neden olabileceği potansiyel sorunların önüne bu şekilde örneklerin çok kısa bir süre tamponda bekletilmesi oldukça uygun bir yöntemdir. Ancak burada dikkat edilmesi gereken husus örneklerin

bu tampon bellekte çok uzun tutulmaması gerekliliğidir. Çünkü bu durumda tampon bellek dolarsa paket kayıpları yaşanabilir bu da bir başka sorun olarak karşımıza çıkar. Ayrıca paketlerin çok uzun süre tampon bellekte tutulması kabul edilemez seviyede gecikme değerleri yaratabilir. Şekil 2.4. seğirme düzeltici tampon bellek yapısı ve çalışma şeklini göstermektedir.



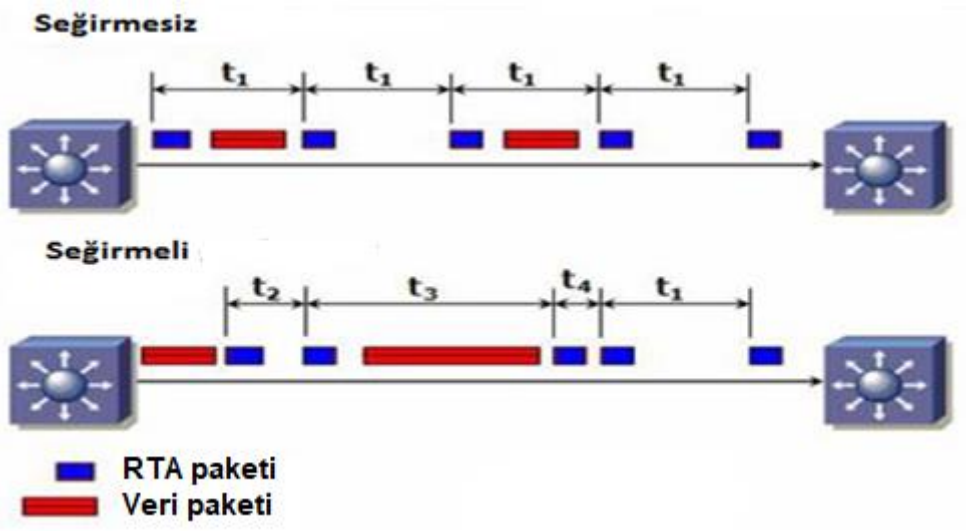
Şekil 2.5. Seğirme düzeltici tampon bellek operasyonu

2.1.1.4. Paket kayıpları

İletimde oluşan paket kayıpları, düşük bant genişliği ve bağlantı tıkanıklığı gibi etkenlere bağlıdır. Ağ üzerindeki akıllı cihazlar üzerinde çalıştırılan çeşitli algoritmalar sayesinde veri yeniden elde edilebilir. Ama aşırı paket kaybı durumunda veri yeniden elde edilemeyebilir [24]. Bunun yanında RTI grubuna ait uygulamalar paket kayıplarına karşı çok duyarlıdırlar.

2.1.1.5. Seğirme

Seğirme, tek bir iletime ait farklı paketler arasındaki gecikme değişimleri olarak tanımlanabilir. Seğirme zamanlama sorunları, bant genişliği sınırlamaları, ağ tıkanıklığı veya senkronizasyon sorunu nedeniyle ortaya çıkabilir. Uç sistemler tamponlama yeteneklerinden dolayı gecikmeleri düzenleyebilir ve senkronize edilip düzenlenmiş paketleri üst katmanlara aktarabilir. Bu parametre RTI trafik için oldukça önemlidir [24]. Bundan dolayı da QoS değerlendirilmesinde önemli bir parametre olarak değerlendirilir. Şekil 2.5. bir iletim ortamında veri paketleri arasındaki seğirme etkisini göstermektedir.



Şekil 2.6. İletim ortamlarında seğirme etkisi

2.1.2. Mevcut teknolojiler ve mimariler

Gerçek zamanlı uygulamaların ihtiyaç duydukları servis kalitesinin sağlanması için Frame Relay, Eşzamansız İletim Modu (ATM), 802.1Q, Çoklu Protokol Etiket Anahtarlama (MPLS) ve Internet Protokol (IP) gibi çeşitli protokoller ve teknolojiler ağ sistemlerine entegre edilmiştir [22]. Bunun yanında IntServ ve DiffServ gibi mimariler de geliştirilerek kullanılmıştır [21].

2.1.2.1. Frame relay

Frame Relay, Tümleşik Hizmetler Dijital Ağı (ISDN) üzerinde paket anahtarlamalı ağ servisleri sunmak için geliştirilmiş bir teknolojidir. Frame Relay, OSI referans modelinin fiziksel ve veri-bağı katmanında çalışan yüksek performanslı geniş alan ağı (WAN) protokolüdür [26].

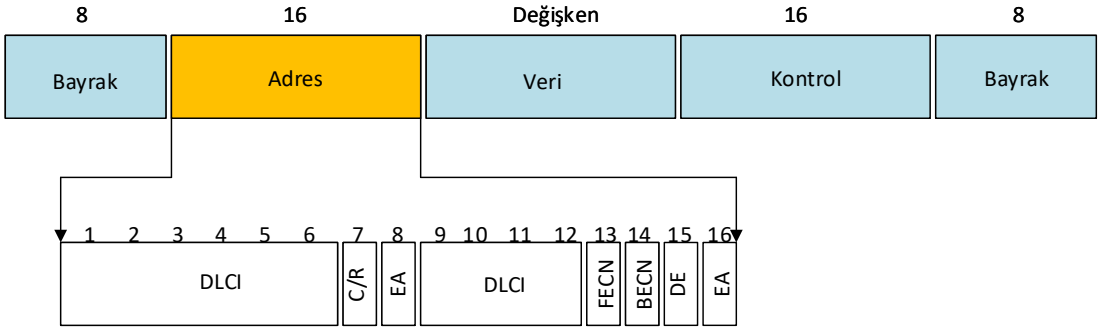
Frame Relay bir ağa sunulan yük yüksek olduğunda, bazı servislerde oluşan patlamalar (bursts) nedeniyle bir takım Frame Relay düğümlerinde aşırı yüklenme oluşabilir ve bu da ağ veriminin (throughput) düşmesine neden olur. Bundan dolayı Frame Relay ağlarda tıkanıklık kontrolü için bazı etkili mekanizmalar kullanmak gereklidir.

Frame Relay ağlarda tıkanıklık kontrolü aşağıdaki unsurları içerir:
Giriş denetimi: Bu yapı, Frame Relay içerisinde, kaynak tarafından ihtiyaçlar doğrultusunda talep edilen servisin kabul edilmesinden sonra, bu servisin verildiğinden kesin olarak emin olmak için kullanılan temel mekanizmadır. Ağ, ağ kapasitesini ve trafik tanımlayıcının isteğini ilişkilendirip karşılaştırarak yeni doğan bağlantı ihtiyacının kabul edilip edilmeyeceğini karar verir. Yapı içinde bulunan trafik tanımlayıcı, servis aboneliği zamanında veya çağrı ayarlama zamanında anahtarlama düğümlerinin haberleşme parametrelerini ayarlar ki bu olay bağlantının özelliklerini belirler. Trafik tanımlayıcı aşağıdaki üç unsurdan oluşmaktadır.

1. Taahhüt Edilen Bilgi Oranı (CIR): Frame Relay bağlantıları genellikle garanti edilen bant genişliği olan Taahhüt Edilen Bilgi Oranı (CIR) ve bit patlaması göz önüne alınarak tanımlanan Genişletilmiş Bilgi Oranı (EIR) olarak bilinen tanımlama ile verilirler. CIR, bir ağın hat üzerinden belirli durumlarda taşıyacağını belirttiği bilgi oranıdır. EIR, mevcut hat üzerinde kullanılabilir bant genişliği varsa eğer ağın koşullar sağlandığında CIR değeri üzerinde gönderebilecek fazla veri miktarıdır. CIR değerini aşarak gönderilen çerçeveler Atmak İçin Uygun (DE) olarak işaretlenir ki bunun anlamı eğer ağ üzerinde bir tıkanıklık meydana gelirse bu çerçeveler düşürülür. Kısaca bu değer, T aralığı süresince ağ üzerinden transfer edilecek bilgi için bit/s cinsinden garanti edilen ortalama hızdır [27,28].
2. Kararlı Patlama Boyutu (BC): T aralığı sürecince transfer edilebilecek verinin maksimum miktarıdır.
3. Aşırı Patlama Boyutu (BE): T süresi içinde ağın taşımayı deneyeceği ve taşımanın garanti edilmemiş olduğu verinin bit cinsinden maksimum miktarıdır.

Frame Relay ağın kenarında bulunan düğümün trafik akışını, ağ kaynaklarının gerçek kullanımının belirlenen miktardan fazla olmadığından emin olabilmek için izler. Bu, kullanıcı transfer oranının tanımlamada belirtildiği gibi olmasını ve aksi takdirde yani transfer oranı aşıldığında verinin atılarak trafiğin belirtilen orana düşürülmesini sağlar [28].

Tıkanıklık bildirimi, Frame Relay için tıkanıklıktan kaçınma politikası olarak önerilmiştir. Bu olay, istenen servis kalitesinin sağlanması amacıyla ağ işlemlerini arzu edilen denge noktasında tutmaya çalışır. Bunu gerçekleştirmek için özel tıkanıklık kontrol bitleri Frame Relay adres alanına dahil edilmiştir. Frame Relay başlığında bulunan İleri Hata Sıkışıklığı Bildirimi (FECN), Geri Hata Sıkışıklığı Bildirimi (BECN) ve Atmak İçin Uygun (DE) bitleri tıkanıklık kontrol mekanizmaları sağlamak için kullanılır [28].



Şekil 2.7. Frame Relay başlık yapısı

Eğer çerçeve iletim yönünde bir tıkanıklık oluşursa bunu göstermek için FECN biti 1 olarak ayarlanır ki böylece hedefe bir tıkanıklık oluştuğu bildirilir. Eğer çerçeve iletim yönünün tersi istikamette bir tıkanıklık oluşursa BECN biti 1 olarak ayarlanır ve bu şekilde tıkanıklık oluşumu göndericiye bildirilir. Böyle bir durumda DE biti ile bildirilen çerçeveler tıkanıklığın önlenebilmesi için düşürülürler.

2.1.2.2. ATM (Asenkron transfer mod)

Eşzamansız İletim Modu (ATM), farklı bant genişliği gereksinimleri ile ses, video ve veri servisleri için ortak bir format sağlamak amacıyla geliştirilmiş bir teknolojidir. Asenkron zaman bölmeli çoğullama tekniğini kullanır ve veriyi hücre ismi verilen sabit büyüklükteki veri yapıları içerisinde taşırlar. ATM, bu sabit 53 bayt uzunluğundaki bu hücreleri ses, video ve veriyi taşımak için kullanır [29].

ATM, QoS sağlamak için zengin özelliklere sahiptir. ATM mimarisi, bağlantı kurulumu sırasında talep edilebilecek altı farklı servis sınıfı sunar [30]. Bu sınıflar aşağıdaki gibidir;

1. Sabit Bit İletim Hızı (CBR): Bağlantı süresi boyunca sürekli kullanılabilir bant genişliğinin sabit olması istenilen bağlantılar tarafından kullanılır. Gerçek zamanlı uygulamalar için ideal çözüm bu servis ile sağlanır.
2. Gerçek Zamanlı Değişken Bit İletim Hızı (rt-VBR): Ses ve video uygulamaları gibi gecikme ve gecikme değişimini sıkı bir şekilde kısıtlayan uygulamalar tarafından kullanılır.
3. Gerçek Zamanlı Olmayan Değişken Bit İletim Hızı (nrt-VBR): Patlamalı trafik karakteristiğine sahip gerçek zamanlı olmayan uygulamalar tarafından kullanılan servistir.
4. Belirlenmemiş Bit İletim Hızı (UBR): Gecikme ve gecikme değişimine sıkı bir şekilde gereksinim duyulmayan gerçek zamanlı olmayan uygulamalar tarafından kullanılan servistir.
5. Kullanılabilir Bit İletim Hızı (ABR): Önceliği az olmasına rağmen garanti bant genişliği içeren servistir. Diğer servislerin kullanmadığı boş bant genişliğini kullanır.
6. Garanti Edilmiş Çerçeve Hızı (GFR): Minimum transfer oranı gerektirebilecek gerçek zamanlı olmayan uygulamalar tarafından kullanılan servistir.

ATM Forum, ATM bağlantısının trafik özelliklerini karakterize etmek için trafik tanımlayıcı olarak altı parametre belirlenmiştir [30]. Bu altı parametre ise aşağıda verilmiştir.

- En Yüksek Hücre İletim Hızı (PCR)
- Sürdürülebilir Hücre İletim Hızı (SCR)
- En Yüksek Patlama Boyutu (MBS)
- En Düşük Hücre İletim Hızı (MCR)
- Maksimim Çerçeve Boyutu (MFS)
- Hücre Gecikme Değişim Toleransı (CDVT)

ATM Forum, trafik parametrelerinin yanında altı tane QoS parametresi tanımlanmıştır [30]. Bunlar aşağıdaki gibidir.

- Hücre Gecikme Değişimi (CDV)
- Hücre Kayıp Oranı (CLR)
- Hücre Hata Oranı (CER)
- Hücre Yanlış Yerleştirme Oranı (CMR)
- Maksimum Hücre İletim Gecikmesi (maxCTD)
- Ağır Hataya Uğramış Hücre Bloğu Oranı (SECBR)

Tablo 2.4. hücre transfer performans parametrelerini ve onların ilgili olduğu QoS karakteristiğini göstermektedir [31].

Tablo 2.4. Hücre transfer performans parametreleri ve ilgili olduğu QoS karakteristikleri

ATM hücre transfer performans parametreleri	QoS değerlendirmesinde genel kriter
Hücre Hata Oranı (CER)	Doğruluk
Ağır Hataya Uğramış Hücre Bloğu Oranı (SECBR)	Doğruluk
Hücre Kayıp Oranı (CLR)	Güvenilirlik
Hücre Yanlış Yerleştirme Oranı (CMR)	Doğruluk
Hücre İletim Gecikmesi (CTD)	Hız
Hücre Gecikme Değişimi (CDV)	Hız

Tablo 2.5. ise ATM özniteliklerinin (trafik parametreleri, QoS parametreleri ve akış kontrolü) bir listesini sunmaktadır ve her bir servis kategorisinin nasıl desteklendiğini belirtmektedir [31].

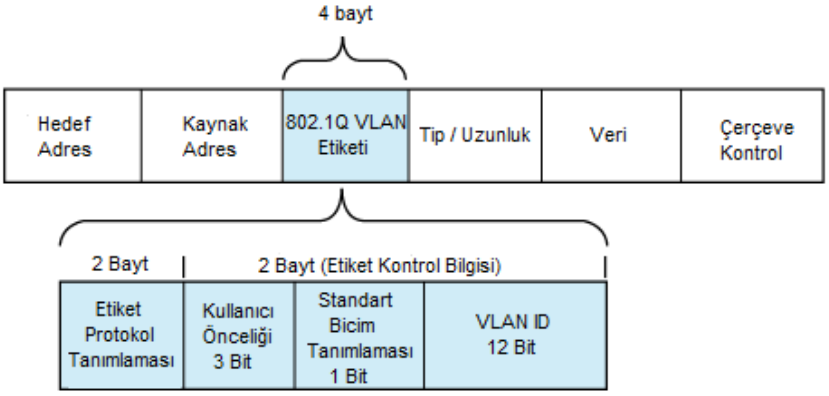
Tablo 2.5. ATM servis sınıfları özellikleri

Servis Sınıfı	CBR	nrt-VBR	rt-VBR	ABR	UBR
CLR	Evet	Evet	Evet	Evet	Hayır
CTD	Evet	Hayır	Evet	Hayır	Hayır
CDV	Evet	Evet	Evet	Hayır	Hayır
PCR	Evet	Evet	Evet	Hayır	Evet
SCR	Hayır	Evet	Evet	Hayır	Hayır
MBS	Hayır	Evet	Evet	Hayır	Hayır
Akış Kontrol	Hayır	Hayır	Hayır	Evet	Hayır

2.1.2.3. 802.1Q

802.1Q standardı, Sanal Yerel Alan Ağları (VLAN) çerçevelerinin etiketlenmesi amacıyla oluşturulan bir teknolojidir. Bu standart ile birden fazla bağımsız mantıksal ağ, tek bir fiziksel Ethernet hattını paylaşabilmektedir. Bu protokol farklı VLAN düğümlerinin ağ katmanı özelliği olan yani yönlendirme özelliği olan anahtarlama aygıtı veya yönlendirici üzerinden iletişim kurmasını sağlar [32].

Bu protokolün QoS ile ilgisi Ethernet çerçevesine eklemiş olduğu etiket içerisinde tanımlı olan öncelik (priority) alanından kaynaklanmaktadır. 802.1Q protokolü tarafından eklenen etiket içerisinde 3 bitlik kullanıcı önceliğini belirleyen bir alan tanımlanmıştır. Böylece, bu alan sayesinde 0 ile 7 arasında toplam sekiz farklı öncelik değeri tanımlanabilmektedir [32]. IEEE 802.1Q standardına ait etiket yapısı Şekil 2.7.'de, önceliklendirme tablosu ise Tablo 2.6.'da verilmiştir.



Şekil 2.8. 802.1q protokolü etiket yapısı

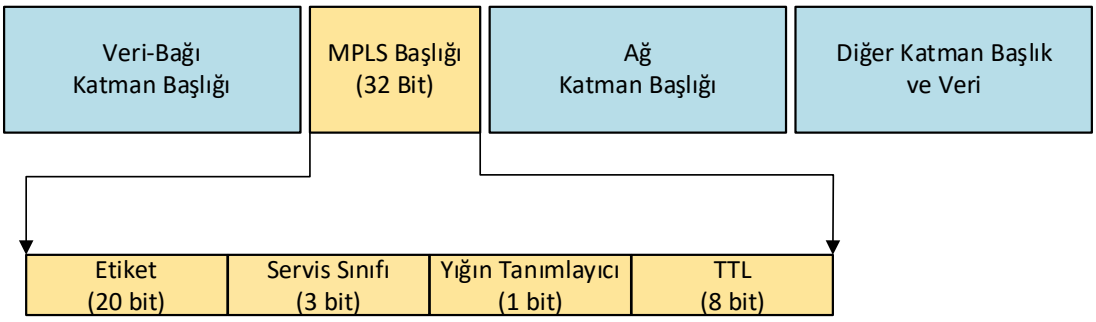
Tablo 2.6. IEEE 802.1q standardına ait önceliklendirme tablosu

Öncelik değeri		Trafik karakteristiği
İkilik değer	Onluk değer	
000	0 (en düşük)	Background
001	1	Best Effort
010	2	Excellent Effort
011	3	Kritik Uygulamalar
100	4	Video, <100 ms gecikme
101	5	Ses, <10 ms gecikme
110	6	Ağlar Arası Kontrol
111	7 (en yüksek)	Ağ Kontrol

2.1.2.4. Çoklu protokol etiket anahtarlama

Çoklu Protokol Etiket Anahtarlama (MPLS), ağ katmanı ve veri-bağı katmanı teknolojilerinin karışımı bir çözüm sunmak için geliştirilmiştir. Bu teknoloji en basit olarak, veri-bağı katmanındaki anahtarlama ve ağ katmanındaki yönlendirmenin birleştirilmesidir. Veri-bağı katmandaki anahtarlama işlemi donanımsal olarak hızlı gerçekleşirken ağ katmandaki yönlendirme ise yazılımsal olarak yapıldığı için daha yavaş gerçekleşir. MPLS ise bu iki teknolojiyi birleştirerek hızlı ve gelişmiş hizmet sunan ağlar oluşturmaktadır.

MPLS, bu ağa giren paketleri işaretlemek için etiketleme mantığını kullanır. Paketler, Etiket Anahtarlamalı Yönlendiriciler (LSR) adı verilen MPLS özelliği aktif edilmiş olan yönlendiriciler sayesinde ardışık olarak Etiket Anahtarlamalı Yollar (LSP) adı verilen yollar boyunca anahtarlanır ve yönlendirilirler [33]. Paket başlıkları MPLS ağa giriş noktasında sadece bir kez işlenir böylece etiket anahtarlama işlevi ağ içindeki ara yönlendiricilerin daha az başlık bilgisi işlemeye yarar ki bunun sonucu hızlı, basit ve ölçeklenebilir ağ olarak karşımıza çıkar [27,33].



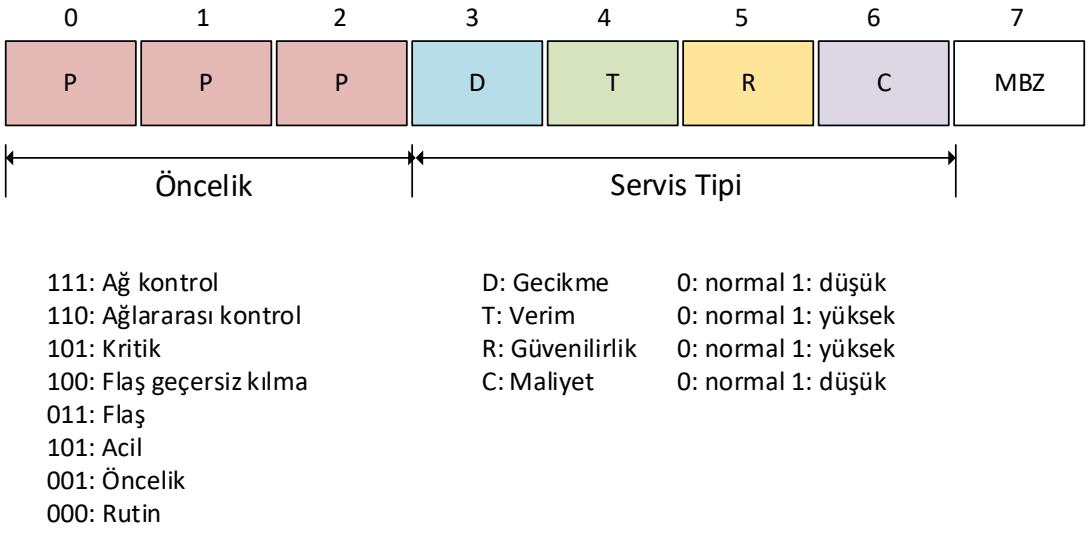
Şekil 2.9. MPLS başlık yapısı

Etiketleme işlemi, MPLS ağı içinde QoS sağlayan bir özelliktir. Paket MPLS ağa girerken IP başlığında bulunan ToS ve TC alanı gibi paket başlıkları içinde bulunan bilgilere dayalı olarak farklı etiketlerle işaretlenebilir [27]. Bu olay aynı zamanda ATM hücreleri için de geçerlidir. MPLS paketleri etiket tabanlı olarak çeşitli trafik ihtiyaçlarının karşılanması amacıyla ayrı ayrı oluşturulmuş farklı LSP'ler boyunca hareket edebilir.

2.1.2.5. İnternet protokol

IP protokolü ağ katmanında çalışmaktadır ve hiyerarşik yapısından dolayı günümüzde kullanılan uçtan uca adresleme protokollerinin en yaygın olanıdır. Aslında IP protokolü, bulunduğu protokol takımı sayesinde trafik akışını protokol, kaynak ve hedef portu, kaynak ve hedef adresi gibi verilere dayanarak sınıflayabilmektedir. Bunun yanında başlıkta bulunan ToS (IPv4) ve TC (IPv6) alanındaki IP öncelik bitlerini kullanarak da trafiği sınıflayabilmektedir [27].

IP başlığı içindeki ToS ve TC alanları IP servislerinin anahtar mekanizmasıdır ve kullanıcı veya uygulamanın istediği servisin kalitesini tanımlamak için kullanılır [34]. IP paketleri, genelleştirilmiş veya özet parametrelerle işaretlenebilir ki bu olay interneti oluşturan ağlar tarafından sağlanan servis seçiminin gösteriminde yönlendiricilere yardımcı olur. Servis tipleri gecikme, verim, güvenilirlik ve maliyet parametreleri tabanlıdır [27,35]. Başlık içerisindeki 8 bitlik bu alanların ilk 3 biti öncelik tanımlamasında kullanılır, sonraki 4 bit ise gecikme, verim, güvenilirlik ve maliyet parametrelerini belirtirken son bit bit ileriki kullanımlar için ayrılmıştır [34].



Şekil 2.10. IP başlığı içerisindeki önceliklendirme ve servis bitleri

Parasal maliyet olarak belirtilen parametre optimal yönlendirme çözümleri kullanılarak minimize edilebilir. Bu alanı kullanan yönlendirme protokolü desteği

OSPF ve IS-IS gibi protokollerle geliştirilmiştir ve alan içinde belirtilen değere göre yolları hesaplamak mümkündür [27,35-37].

Bu bitlerin kullanımı hiçbir zaman geniş ve düzgün bir kullanım sağlamamıştır [27,35]. Değerler sabittir ve QoS tanımlamalarının küçük ve sınırlı bir gösterimini sunmaktadır. Bu durum, daha büyük ve farklı trafikleri işleme söz konusu olduğunda büyük QoS tanımlamaları gerekmiş ve ölçeklenebilirlik sorunları ortaya çıkmıştır [36,38]. Kısaca, QoS parametrelerini belirtmek için kullanılan 4 bitin her defasında sadece bir tanesi set edilebilmekte, böylece dört bit kullanılarak elde edilebilecek on altı farklı durum yerine sadece beş farklı durum oluşmakta bu da büyük trafiklerin gerek duyduğu QoS gereksinimini sağlayamamaktadır. Bundan dolayı bu konudaki eksikliği gidermek adına IntServ ve DiffServ gibi mimariler geliştirilmiştir.

2.1.2.6. IntServ - Bütünleştirilmiş servisler

İnternet üzerinde gerçek zamanlı olan ve olmayan IP servislerine QoS desteği sağlamak için Bütünleştirilmiş Servisler (IntServ) geliştirilmiştir. IntServ mimarisi video konferans, IPTV ve dağıtık simülasyon uygulamaları gibi gerçek zamanlı olan veya olmayan uygulamaların artan ihtiyaçlarını karşılamak için geleneksel internet mimarisi üzerinde bir gelişim önermektedir [39]. IntServ mimarisi geleneksel internet mimarisi üzerinde bir değişim önermez, gerçek zamanlı uygulamaların ihtiyaçlarını karşılamak için gelişim önerir [27,28].

IntServ yapısı, uçtan-uca sistemlerde ara yönlendiricilerin tüm hat buyunca komşularıyla haberleşerek, gönderilecek verinin ihtiyaç duyduğu desteğin tüm hat boyunca sağlanması amacı ile gerçekleşen işlemler bütünüdür. IntServ sert-QoS tabir edilen yapıda işlemlerini gerçekleştirir. Sert-QoS ifadesi, uç noktalar arasında bant genişliği, gecikme ve paket kaybı olarak ifade edilen QoS gereksimlerinin belli bir değer altına düşmemesi gereken uygulamalar için kullanılan bir yapıdır. Bu nedenle IntServ mekanizması iki nokta arasında çalışan uygulamanın ihtiyaç duyduğu bant genişliğini tahsis eder. Bu eylem sonunda çalışan uygulama için arzu edilen seviyede servis verilmiş olur.

IntServ modelinde, GZU'ya ait ihtiyaçlar, uygulama verileri gönderilmeden önce kontrol edilir. Uygulama ihtiyaçları ve gereksimlerini ağa iletir. Bu ihtiyaçlar doğrultusunda talep edilen bant genişliği ve gecikme gibi ağ kaynaklarının ayrılmasını talep eder. Kenar yönlendirici, bu taleplerin karşılandığı bilgisini almadan verileri ortama göndermez. Onay alındığında ise taleplerin karşılandığından emin olarak veriler ortama aktarılır.

IntServ mimarisi üç farklı servis sınıfı tanımlar. Bunlar;

1- Garantili Oran: Belirtilen gereksinimleri karşılayacak şekilde verilen servistir. Bu serviste trafik;
   - Emin olunan veri hızı
   - Kuyruk gecikmesinde belirtilen üst sınır
   - Kayıpsız kuyruk yapısı

   ile karakterize edilmiştir[28,39].
2- Kontrollü Yük: Gönderilen verilerin en az kesinti ile iletileceği servistir. Bu serviste trafik;
   - Yüksüz durum altında best effort davranışı yaklaşımı
   - Kuyruk gecikmesinde üst sınır yoktur ama bu gecikme çoğunlukla maksimum iletim gecikmesini aşmaz
   - Best effort servislerden daha iyi bir değer veren neredeyse hiç kuyruk kaybı olmaması

   şeklinde karakterize edilmiştir[28,39].
3- Best Effort: Geleneksel internet servislerini ifade eder.

Özetlemek gerekirse, IntServ mimarisi veri aktarımı başlamadan önce akış tarafından talep edilen kaynaklar rezerve edilir. Kaynağın rezerve edilmesi ve bunun tahsisi, internet üzerinde gerçek zamanlı olan ve gerçek zamanlı olmayan uygulamaların talep ettikleri QoS değerlerinin garanti edilmesini sağlar. IntServ mimarisinde akış yolu üzerindeki tüm düğümlerde durum bilgisi, paket sınıflandırma süreci, işaretleme, politika geliştirme ve operasyonları şekillendirme süreçlerinin tüm bilgilerinin tutulmasını gerektirmektedir.

2.1.2.7. DiffServ - Farklılaştırılmış servisler

Farklılaştırılmış servisler (DiffServ), özet olarak ağ üzerindeki veri trafiğinin özelliklerine göre benzer sınıflara ayrılıp, aynı sınıf içerisinde yer alan veri trafiğinin yol boyunca tüm cihazlarda aynı servisi alması olarak tanımlanabilir. DiffServ mimarisinde IntServ mimarisinde olduğu gibi önceden bir kaynak rezervasyonu yapılmaz. Veri bir cihaza geldiğinde ilgili sınıf için tanımlanan uygulama ne ise ona göre işlenir.

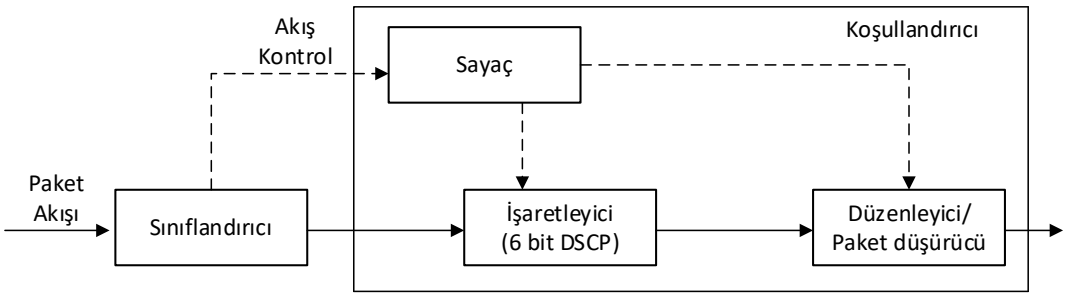
DiffServ mimarisinde veri akışı farklı trafik sınıflarına bölünür. QoS politikaları yardımı ile her bir sınıfa ayrı bir davranış tanımlanabilir. DiffServ mimarisinde QoS istekleri bağımsız bir şekilde her bir cihazda ayrı olacak şekilde tanımlanır ve buna yumuşak-QoS ismi verilir. Bu yapıda sert-QoS yapısında olduğu gibi iki uş nokta arasında aynı seviyede destek garanti edilmez. Bundan dolayı daha esnek bir uygulama ortamı sağlar [40].

DiffServ mimarisi, internet trafiğinin gereksinimlerini belirtmek için atlama-başına davranış (PHB) kullanır ki bu aynı zamanda IP paketlerini işaretlemede kullanılır. DiffServ modelde trafik ağa girdiğinde sınıflandırılır ve ağa giriş anındaki şarta bağlı olarak farklılaştırılmış hizmetler kod noktası (DSCP) tarafından tanımlanarak uygun bir davranış kümesine atanır [28,41]. Paketler, her bir DSCP ile ilişkili PHP değerine göre iletilir. Bundan dolayı kenar yönlendiriciler işaretleme, merkez yönlendiriciler sınıfa uygun iletim odaklı çalışırlar.

DiffServ mimarisinin karakteristik özellikleri aşağıda gösterildiği gibidir [28,41].

1- IP paketleri QoS gereksinimlerini belirtmek için ağ sınırında işaretlenir.
2- Gönderici ile alıcı arasında bir servis seviye anlaşması oluşturulur.
3- DiffServ mimarisi içinde çalışan yönlendiriciler benzer davranış özelliklerine ve aynı DSCP değerine sahip tüm paketlere aynı davranışı sergiler.

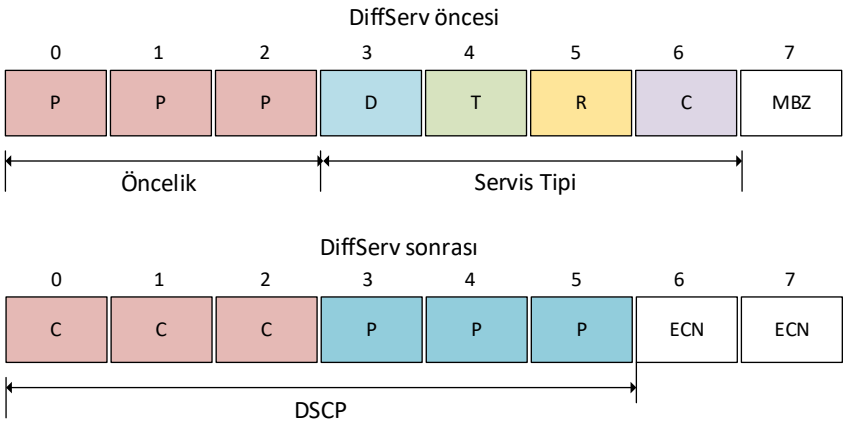
DiffServ mimarisini destekleyen her yönlendirici DSCP ile PHP tabanlı tek tek yapılandırılıp sınıflandırılmış paketleri iletir.



Şekil 2.11. Diffserv mimarisi çalışma yapısı

DiffServ modeli Şekil 2.10.'da gösterildiği gibi trafik şartlarını sağlayabilmek için Sınıflandırıcı, Sayaç, İşaretleyici, Düzenleyici ve Düşürücü olmak üzere beş mekanizma kullanır. Sınıflandırıcı, DSCP'de belirtilen değere bağlı olarak gelen paketleri ilgili sınıflara ayırır. Sayaç, belirtilen trafik profili ile trafik akışının aynı olup olmadığını teyit edip onaylar. İşaretleyici ise eğer gerekiyorsa paketin DSCP değerini tekrar işaretler. Düzenleyici, eğer veri akışı trafik profilinde belirtilen değerlere göre onaylanmadığı bir durumla karşılaşıldığında paketlerin tamamının trafik profilindeki değerlere göre veya gecikme değerlerine göre onaylanıp onaylanmadığını doğrular. Düşürücü ise eğer trafik akışı, trafik profilinde belirtilen değerlerle uyumlu değilse akışın düşürülecek paketleri için politika belirler.

Diffserv desteği, IPv4 de başlık içerisindeki ToS, IPv6 da ise TC alanına DS kodlarının yerleştirilmesiyle elde edilir [28,33]. DSCP alanını oluşturan 6 bitin ilk üçü Sınıf Seçici Kod Noktaları (CSCP) diye isimlendirilir ve ilgili paketin sınıfını tanımlamak için kullanılır. Böylece IP öncelik değerleri ile de uyum sağlanmıştır.



Şekil 2.12. TOS alanı değişimi

2.1.2.7.1. Hızlandırılmış iletim

Hızlandırılmış iletim (EF), olarak adlandırılan model IntServ yapısında olduğu gibi garanti edilmiş servisler yapısına benzer bir modeldir. DiffServ yapısında düşük paket kaybı, gecikme ve seğirme değerlerinin az olması ve bunun yanında bant genişliği ihtiyacı olan uygulamalarda kullanılan modeldir. Bir ağ üzerinde çok kritik olarak tanımlanan uygulamalara verilmesi gereken servistir. EF modelinde uygulama için ayrılmış kaynakların (bant genişliği, bellek miktarı ve işlemci yüzdesi vs) aşılması halinde fazlalık oluşturan paketler iptal edilir. EF modelinde, uygulamalar için önceden belirtilen miktarda kaynak ayrılır ve bu kaynaklar önceliklendirilmiş trafik için kullanılır [40].

2.1.2.7.2. Garantili iletim

Garantili iletim (AF), IntServ modelindeki kontrollü ve dengeli yük yapısına benzer bir yapıdır. Birbirinden ayrı 4 sınıf ve 3 düşürme önceliği içermektedir. Gösterilimi AFxy şeklindedir. Bu gösterimde x, AF sınıfını; y ise düşürme önceliğini ifade eder. Sınıfları AF1, AF2, AF3, AF4 olarak belirtilir. Bu sınıflara, mevcut ihtiyaç halinde bant genişliği, işlemci ve bellek tahsis edilebilir. Düşürme önceliği ise herhangi bir tıkanıklık meydana geldiğinde paketlerin çöpe atılma ihtimalini belirtir ve AFx1(düşük), AFx2(orta), AFx3(yüksek) şeklinde gösterilir.

Özetlemek gerekirse DiffServ mimarisi internet üzerinde kullanımı IntServ mimarisine göre daha kolay ve esnektir. Çünkü IntServ mimarisinde ağ üzerindeki yönlendiriciler her bir akış için durum bilgisi tutması gerekirken DiffServ mimarisinde benzer özelliklere sahip akışlar aynı grup içerisinde birleştirilerek yönlendiricilerin yükü azaltılmıştır. DiffServ mimarisinde her bir akış için ayrı durum bilgisi tutulmaz, tüm akış önceden belirlenmiş sınıflar içerisine bir PHB etiketi ile atanır ve buna göre davranış politikası belirlenir. Bu şekilde yönlendirici üzerinde tutulacak durum bilgisi sınırlandırılmış ve yönlendirici etkinliğinin artması sağlanmıştır.

2.1.3. Servis kalitesi hakkında önceki çalışmalar

Geleneksel internet mimarisinin desteklemediği GZU uygulamalarına ait ihtiyaçların karşılanması amacı ile servis kalitesi birçok alanda araştırma konusu olmuştur. Servis kalitesini etkileyen parametreler incelendiğinde en dikkat çekici olan parametrelerin gecikme ve paket kaybı olduğu görülmektedir. Bundan dolayıdır ki yapılan araştırmalar genellikle bu iki parametre üzerine yoğunlaşmıştır.

Ses ve video sıkıştırma kod çözücüleri, veri boyutunu küçülterek hem bant genişliği ihtiyacını azaltmaktadır hem de gecikme ve paket kayıpları parametrelerinin iyileştirilmesini sağlamaktadır. Dolayısı ile sıkıştırma kod çözücülerinin servis kalitesine etkileri araştırılması çalışmalara konu olmuştur [42,43]. Bunun yanında, ağ üzerinde herhangi bir tıkanıklık oluştuğunda daha düşük bit oranına sahip bir kod çözücüye anahtarlama ile çözüm üreten bir yaklaşım da sunulmuştur [44].

Kod çözücü gecikmesi sabit bir gecikme olduğundan dolayı yapısal değişiklikler dışında gecikmeye etkisi düşürülememektedir, bundan dolayı araştırmacılar daha çok değişken gecikme değerleri üzerine yoğunlaşmış ve farklı kuyruk tekniklerinin çeşitli geçek zamanlı ortam ve uygulamalarında karşılaştırmalı analizleri yapılmıştır [12-17]. Ses ve video trafiği altında ilk giren ilk çıkar (FIFO), DSCP tabanlı ağırlıklı adil kuyruk (WFQ) ve öncelik tabanlı kuyruk (PQ) teknikleri karşılaştırılmış ve gerçek zamanlı uygulamalarında gecikme ve seğirme gibi QoS parametreleri için en iyi sonucun PQ tekniği kullanıldığında alındığı görülmüştür [12,13]. Gerçek zamanlı uygulamalarda iyi sonuçlar veren WFQ ve PQ tabanlı hibrit bir kuyruk önerisi yapılan bir çalışmaya konu olmuş ve çalışmada bu kuyruğa ait performans verileri değerlendirilmiştir [14]. Mevcut kuyruk modellerinin performans değerlendirmelerin yanında alternatif öneriler de mevcuttur. QoS ve sistem performansını arttırmak için başlıca etken olan kuyruk gecikmesi ve bu gecikme değişimlerinin optimize etmek amacı ile PQ bir kuyruk modeli önerisi ve bu modelin performans analizleri yapılmıştır [15]. Bir başka çalışmada ise yüksek hızda veri paketi indirme bağlantısı (HSDPA) sistemler üzerinde GZU paketler için gecikme tabanlı üretilen bir parametre yardımı

ile PQ dinamik bir tampon bellek yönetimi şeması önerilmiş ve analizi yapılmıştır [16,17].

Ses ve video sinyallerinin paketlenme etkisi ve bant genişliği ilişkisi benzetimi yapılmış [45], ses ve video uygulamalarının performansları da ayrıntılı olarak incelenmiştir [46]. IPv4 ve IPv6 protokolleri üzerinde ses ve video gecikme değerlerinin araştırıldığı çalışmada ise [47] farklı IP sürümlerinin gecikme üzerine etkisi olmadığına, bulunan farkların ise önemsiz olduğu sonucuna varmıştır.

Bunun yanında MPLS ağlar üzerinde QoS çalışmaları da yoğun bir şekilde işlenmiştir [48-51]. MPLS ağlar üzerinde QoS desteği için geliştirilmiş olan IntServ ve DiffServ gibi mimarilerin performanslarının karşılaştırılması yapılmıştır [48]. Düşük trafik altında IntServ mimarisi daha iyi sonuç verirken trafik miktarı arttırıldığında DiffServ mimarisi daha iyi sonuçlar vermiştir.

GZU uygulamaları QoS için uçtan-uca desteğe ihtiyaç duymaktadırlar. Frame relay, ATM, MPLS gibi mevcut teknolojiler ise OSI modelinin veri-bağı katmanında tanımlı oldukları için sadece kendi omurgalarında bu desteği vermeye çalışmaktadırlar. Bu omurgayı terk eden paketler bu teknolojiler tarafından desteklenmeleri mümkün olmamaktadır. IntServ ve Diffserv mimarileri ise daha üst katmanlarda tanımlı olduklarından dolayı uçtan-uca destek verebilmekte ancak kısıtlı sınıflandırma olanakları ile arzu edilen desteği sunamamaktadırlar. Bu çalışmada, hem uçtan-uca destek vermek için IPv6 protokolünü kullanılacak hem de QoS kavramına yeni bir yaklaşım getiren IPv6 başlığı içerisinde yer alan FL alanı kullanılarak DiffServ yapısına uygun PQ kuyruk modeli ile literatürde bulunmayan yönlendirme protokol metriğine dayalı bir önceliklendirme önerisi sunulacaktır. Böylece GZU verisine uygulanan öncelik uçtan-uca sağlanacaktır. Bu amaçla sonraki bölüm yeni nesil internet protokolü olan IPv6 kavramının açıklanması olacaktır.

2.2. Yeni Nesil İnternet Protokolü ve Akış Etiketi

İnternet protokolünün en son sürümü Internet Protokol versiyon 6 (IPv6) olarak isimlendirilmiş ve RFC 2460 belgesi ile özellikleri tanımlanmıştır. Şu anda geçerli olan ve çoğunlukla kullanılan önceki sürüm olan IPv4'tür. IPv6, internetin büyümesiyle ortaya çıkan ve öncelikle adresleme yetersizliği ile başa çıkabilmek için IETF tarafından tasarlanan yeni sürüm protokoldür. Bu protokolün gelişim süreci tamamlanmış ve uygulanmaya başlanmıştır.

IP protokolünün yeni bir sürümünün geliştirilmesi için tetikleyici etkenin IPv4 sürümünün adres uzayının tükeneceği olarak algılansa da güvenlik ve QoS gereksinimleri gibi diğer birçok gereksinimleri karşılamak amacıyla ortaya çıkmış daha esnek ve daha gelişmiş bir protokoldür. IPv6 protokolünün geliştirilmesindeki temel amaçlar adres alanını arttırmak, güvenliliği geliştirmek, çoklu gönderim (multicasting) olayını basitleştirmek ve servis kalitesi özellikleri eklemek olarak sıralanabilir.

2.2.1. IPv6 temel özellikleri

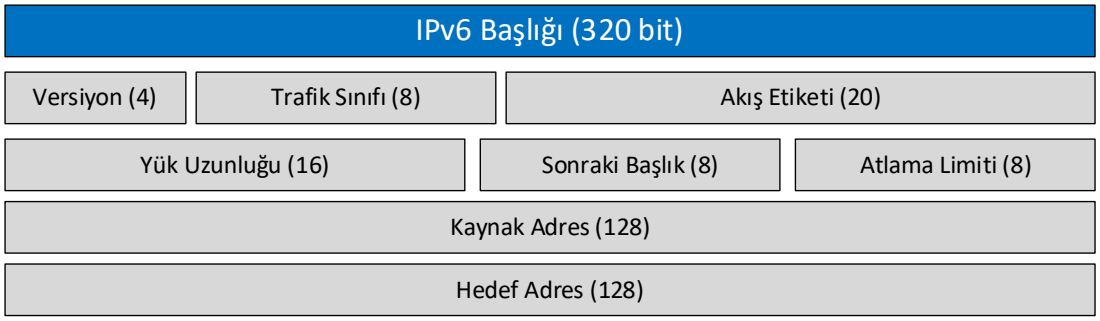
IPv4'ten IPv6'ya geçişte RFC 2460 belgesinde belirtilen önemli değişiklikler aşağıdaki gibidir.

2.2.1.1. Genişletilmiş adresleme yeteneği

IPv6, adresleme hiyerarşisi seviyelerini daha fazla desteklemek için, daha fazla sayıda adreslenebilir düğüm yaratmak ve adreslerin otomatik yapılandırılmasını basitleştirmek amacıyla IP adreslerinin boyutunu 32 bitten 128 bite çıkarmıştır. Bu şekilde yaklaşık olarak $3{,}4 \times 10^{38}$ adres elde edilebilmektedir. Çoklu gönderim yönlendirmenin ölçeklendirilmesi ile yeni tip adresler yaratılmış, bu yeni tip adresler "her noktaya gönderim adresi (anycast address)" olarak isimlendirilmiş ve bir grubun düğümlerinin herhangi birine paket gönderimi için kullanılmak üzere tanımlanmıştır.

2.2.1.2. Başlık yapısının basitleştirilmesi

IPv6 başlığının bant genişliği maliyetini sınırlamak ve paket taşıma sırasındaki işlem maliyetini azaltmak için IPv4 başlığındaki bazı alanlar ya terk edilmiştir ya da isteğe bağlı hale getirilmiştir. 128 bit olan IPv6 adres boyutu, 32 bitlik adres boyutuna sahip olan IPv4'ün 4 katıdır. Buna rağmen IPv6 başlık bilgisi 40 bayt ile 20 bayt olan IPv4 başlık bilgisinin sadece 2 katıdır [52].



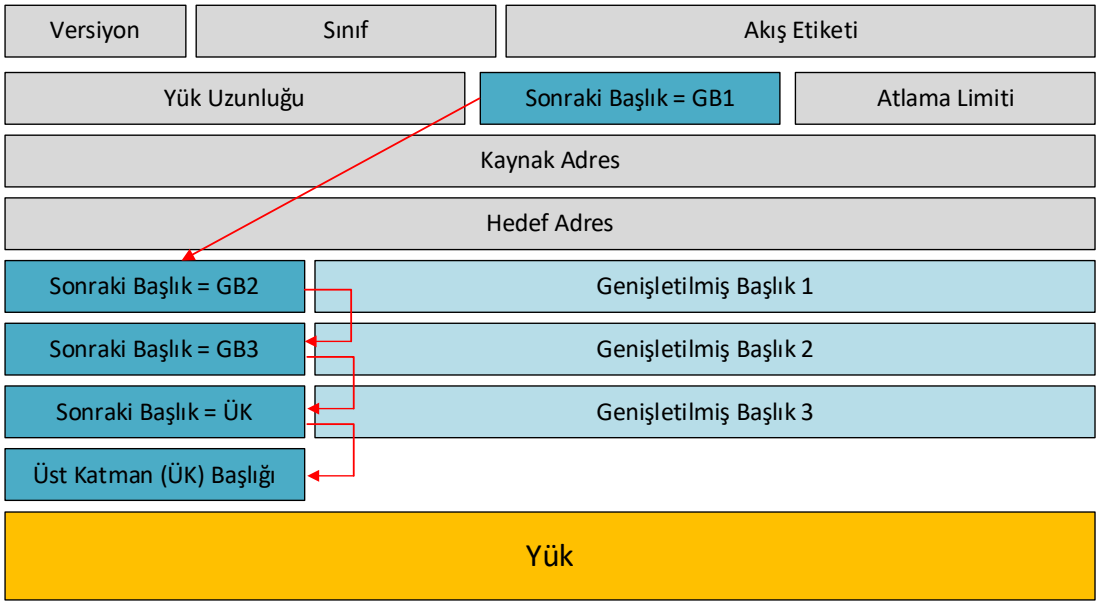
Şekil 2.13. IPv6 başlık yapısı

2.2.1.3. Uzantılar ve seçenekler için arttırılmış destek

IP başlığındaki alanların değiştirilmesi daha verimli yönlendirme, alan uzunluklarındaki sınırlamalarda kolaylık, gelecekte ihtiyaç olacak uygulamaların tanımlanmasında esneklik sağlar.

Bilgisayar ağlarında, gönderilen verilere ait adresleme yapısında kullanılan başlık bilgisi düzenli bir yapıda değilse, bu verinin işlenip yol bulma sürecinin tamamlanması yavaş olabilir. IPv6'da ise başlık bilgilerinin daha esnek hale getirilmesi sayesinde işlemcinin daha verimli çalışarak sürecin daha hızlı olması sağlanmaktadır. Başlıktaki yapısında bulunan esneklik, servisin kalitesine göre başlık bilgisinin değişebilirliği demektir; bunun sonucu olarak mesajlar kısa sürede işlenebilir ve yönlendirilebilir.[53]

Temel IPv6 başlığına ek olarak, paketlerin bölümlenmesi ve şifreleme gibi başlık bilgilerine ihtiyaç duyulması kullanılması için tanımlanmıştır ve gerektiğinde yenileri de eklenebilir.



Şekil 2.14. IPv6 genişletilmiş başlık yapısı

2.2.1.4. Kimlik doğrulama ve güvenlik özellikleri

IPv6 başlık uzantıları ile kimlik doğrulama, veri bütünlüğü ve isteğe bağlı olarak veri güvenliği ve gizliliği desteği sağlanabilir. IP v6 protokolünde güvenlik için IPsec (IP security protocol) ile uyum zorunludur. Aslında IPSec, IPv6 protokolünün bir parçasıdır [54].

IPv6 üzerinde IPSec işlemi iki genişletilmiş başlığın kullanılmasıyla sağlanır. Bunlar Kimlik Doğrulama Genişletilmiş Başlığı (authentication extension header) ve Şifreli Güvenlik Yükü (Encrypted Security Payload) genişletilmiş başlığıdır. Kimlik doğrulama genişletilmiş başlığı kaynak bütünlüğü ve kaynağın kimlik doğrulaması ile tekrar saldırılarına karşı koruma ve başlık alanlarının bütünlüğünün korunmasını sağlar. Şifreli güvenlik yükü genişletilmiş başlığı ise gizlilik, kaynağın kimlik doğrulaması, tekrar saldırılarına karşı koruma ve sınırlandırılmış trafik akışı gizliliği sağlar [54].

2.2.1.5. Akış etiketleme yeteneği

Yeni bir yetenek olarak karşımıza çıkmaktadır. Ön tanımlı olmayan ve özellik arz eden servis kalitesi veya gerçek zamanlı servisler gibi göndericinin gönderdiği paketler üzerinde özel işlem istediği durumlarda "akış" olarak adlandırılan belirli bir trafiğe ait paketlerin etiketlenmesini mümkün kılar. Böylece tüm paket trafiği içerisindeki etiketlenmiş ve belirli bir servis kalitesine ihtiyaç duyan paketler derinlemesine irdelenmeye gerek kalmadan bu etiketlerinden tanınarak gerekli olan servisi alabilirler.

2.2.2. IPv6 Protokolünde Servis Kalitesi

IPv6 başlık yapısı içerisinde trafiği etiketleyebileceğimiz iki alan bulunmaktadır. Bunlardan biri TC diğeri ise FL alanıdır. TC alanı IPv4 başlığı içerisindeki ToS alanıyla RFC 2474 belgesi ile önerilen kullanım yapısında eşdeğer olarak tanımlanmıştır [33].

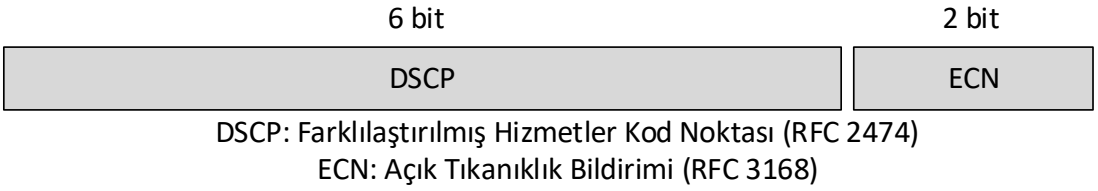
IPv6 başlığı içindeki TC alanı, servis kalitesi gereksinimlerine göre öncelik tabanlı veya farklı sınıflara ayırarak paketlerin sınıflandırılması şeklinde düğümlere destek sağlar. Aynı zamanda, yönlendiricilerin yönlendirme yapabilmesi için farklı sınıflara ait paketlerin tanımlamasını sağlar. RFC 2460 belgesi, TC alanında uygulanması gereken birkaç genel gereksinimleri belirtmiştir. Bu gereksinimler şu şekilde sıralanabilir [52].

- Bir düğüm içerisinde servis ara yüzünün IPv6 servisine destek sağlaması gerekir. Bunun anlamı, üst katman protokolü tarafından kaynağı belirtilen ve sınıflandırılmak istenen paketler üzerinde gerekli işlemin yapılabilmesi için TC'yi oluşturtan bitlerden bir değer sağlaması gerekir. Ön tanımlı değer TC'yi oluşturan sekiz bitinde 0 olmasıdır.
- TC'yi oluşturan bitlerin tamamının veya bir kısmının bir standart dâhilinde veya deneysel kullanım gibi özel bir kullanım için düğümlerin desteklenmesi gerekir, yani paketlerin kaynağı, iletilmesi veya kaydedilmesi gibi kullanım gereksinimleri için paket içindeki bu bit değerlerinin düğüm tarafından

değiştirilmesine izin verilmelidir. Eğer özel bir kullanım söz konusu değilse düğümler TC alanındaki bitlerin değerini değiştirmemeli bu alandaki değeri görmezden gelmelidir.

- Bir üst katman protokolü, alınan paket içerisindeki TC alanı bitlerinin değerini paketin kaynağı tarafından gönderilirken belirtilen değer ile aynı kabul etmemelidir.

RFC 2474 belgesi, TC alanı için bir kullanım yapısı önermektedir [33]. Bu yaklaşım, farklılaştırılmış servisleri (DS) desteklenmesi için TC alanının 8 bitlik DS alanı ile değiştirilmesini tanımlamaktadır. Buna göre DS alanı içerisindeki ilk 6 bit, seçilen atlama-başına davranış (PHB) tanımlamak için farklılaştırılmış hizmetler kod noktası (DSCP) olarak kullanılır. Bu şekilde 64 farklı davranış sınıfı oluşturulabilir ve paketler bu sınıflara dahil edilebilir. DS alanının son 2 biti kullanılmadan bırakılır. DS alanının yapısı Şekil 2.14.'te gösterildiği gibidir.



Şekil 2.15. IPv6 başlık yapısı içerisindeki DS alanı

IPv6 protokolü içerisinde bulunan TC alanının kullanımı IPv4 protokolünde bulunan ToS alanı ile aynı olduğundan dolayı detaylı açıklama Bölüm 2.1.2.5'de yapılmıştır. IPv6 protokolü, TC alanından başka başlık içerisinde QoS desteğinin sağlanabilmesi için Flow Label adı verilen 20 bitlik bir alan daha tanımlamıştır.

2.2.3. Akış etiketi alanı

IPv4 başlık yapısı içerisinde bulunmayan FL alanı, ilk olarak RFC 2460 belgesiyle tanımlanan IPv6 başlığı içerisinde tanımlanmıştır ve amacı bir düğüm tarafından bir akışa ait paketlerin etiketlenmesi olarak belirtilmiştir. İlk olarak RFC 3697 belgesiyle özellikleri tanımlanan bu alan için sonradan RFC 6437 belgesi yayınlansa da kullanım

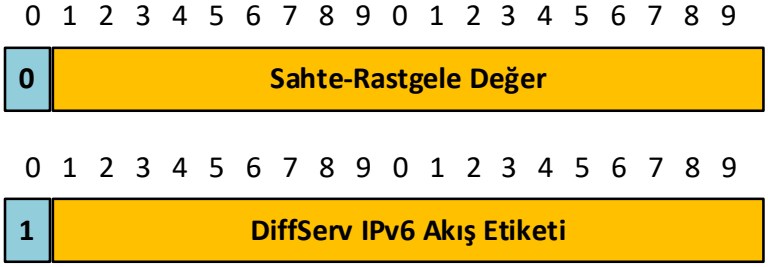
hakkında kesin yargılar bulunmamaktadır. Toplamda 20 bit olarak tanımlanan bu alan için bütün bitlerinin sıfır olması durumunda herhangi bir etiketleme yapılmadığı anlaşılsa da bu alanın nasıl kullanılacağı tam ve kesin olarak tanımlanmadığı için zaman içerisinde çeşitli kullanım yaklaşımları önerilmiş ancak bu yaklaşımların hiç birisi kabul edilip standart halini almamıştır.

FL alanı yardımı ile işaretlenen paketler sayesinde yönlendiriciler akışa ait trafiği kolayca tanır böylece akışın ihtiyacı olan özellikte işlenmesi sağlanır. Bu yapı sayesinde gerekli olan servis kalitesi, IPv6 protokolü tarafından sağlanmış olur [52]. Bundan dolayı servis kalitesi desteği için önerilmiş olan FL kullanım yaklaşımlarına göz atmakta fayda vardır.

2.2.3.1. Conta önerisi

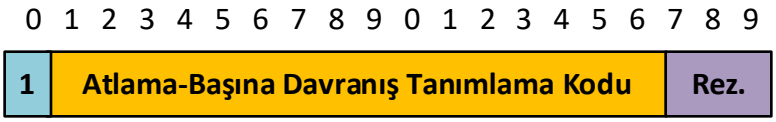
A. Conta ve B. Carpenter tarafından hazırlanan ve 2001 yılında "A proposal for the IPv6 Flow Label Specification" başlığıyla IETF tarafından taslak olarak yayınlanan öneridir [55].

Bu taslak IPv6 FL tanımlamasına bir yaklaşım önerisi sunmaktadır. Bu öneri, RFC2460 belgesinde belirtilmiş tanımlamanın değiştirilmesi şeklindedir. Buna öneriye göre eğer gerekiyorsa özgün FL değeri yol üzerindeki yönlendiriciler tarafından yeniden üretilebilir veya aynen korunur. Bu durum kaynak alıcıya özel bir bilgi iletmek istediğinde kesinlikle gereklidir. Bu öneriye göre eğer FL içerisinde komşu yönlendiriciler hakkında özel durum bilgileri taşınacaksa bu özellik ve yetenek mutlaka gereklidir. Bu öneride bir dezavantaj bulunmaktadır. Bu da FL'nin sabit yapısına rağmen kompleks bir öneridir. Taslağa göre önerilen format Şekil 2.15.'teki gibidir.



Şekil 2.16. Conta tarafından önerilen FL formatı

Bu öneri, RFC2460 belgesinde belirtilmiş olan FL değerinin belirlenmesinde rastgele sayı yapısına destek vermesidir. Bununla birlikte pakete verilecek olan DiffServ desteği bu öneri ile sağlanır. Bu taslakta IPv6 FL için önerilen DiffServ tanımı Şekil 2.16.'da verilmiştir.



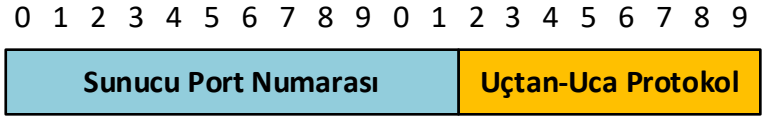
Şekil 2.17. Conta tarafından önerilen FL alanı DiffServ tanımlaması

Bu öneride, Atlama-Başına Davranış Tanımlama Kodu (PHB-ID) değeri için 16 bit ayrılmıştır. Bu sayı tabanlı bir öneridir ve RFC3140 belgesi içeriğinde tanımlanmıştır.

Sonuç olarak bu öneride Akış Etiketi yardımıyla IPv6 protokolü için DiffServ desteği sağlanabilir. Şekil 2.16.'da "Rez" olarak gösterilen bitler gelecekte kullanılmak üzere ayrılmış bitlerdir.

2.2.3.1.1. Sunucu port – kısa format önerisi

Conta sunmuş olduğu öneride [55] alternatif format yapıları da bunlunmaktadır ve bunlar tartışılmaya açılmıştır. Önerilen bu yaklaşımda, Akış Etiketi alanı içerisinde protokol tipi ve başlık bilgileri uçtan-uca taşınabilir. Bunun için önerilen format Şekil 2.17.'de verilmiştir.



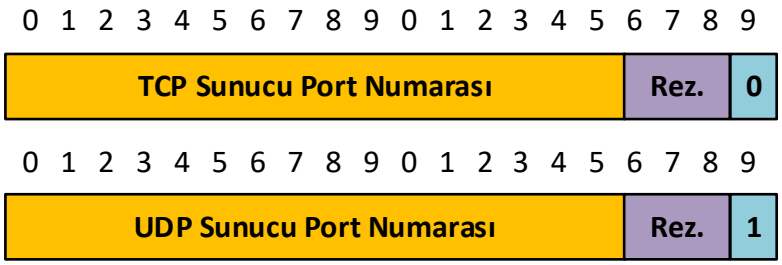
Şekil 2.18. Conta tarafından önerilen sunucu port – kısa format yapısı

Şekilde gösterilen "Sunucu Port Numarası" alanı ise istemci/sunucu yapılarında uygulama tipinin belirlenmesi için sunucunun yapmış olduğu port ataması numarasıdır. Bu yaklaşım ile uygulamalar için ihtiyaç duyulan QoS karakteristik yapıları tanımlanabilir. "Uçtan-Uca Protokol" alanı verileri uçtan-uca taşımak için kullanılan TCP, UDP veya benzeri protokoller olabilir ve uçtan-uca taşıma protokol

tanımlaması olarak kullanılmaktadır. Ancak şekilde görüldüğü gibi port numarası için 12 bit ayrılmıştır. Böylece sadece 4096 sayısı ile ifade edilebilecek portlar kapsanabilir ve tüm portları göstermek için yetersiz kalmaktadır.

2.2.3.1.2. Sunucu port – uzun format önerisi

Conta'nın yapmış olduğu bir diğer yaklaşım ise [55] uzun format yaklaşımıdır. Bu yaklaşımda istemcı/sunucu yapılarında sunucunun atamış olduğu TCP veya UDP port numarası için ilk 16 bit ayrılmıştır. Sıradaki 3 bit, ileride gelebilecek uygulamalara ayrılmıştır. En son bit ise protokol tanımlamasıdır ve 0 sayısı TCP'yi 1 ise UDP belirtecidir. Bu yaklaşım ile kısa format yapısında meydana gelen handikap giderilmiştir.

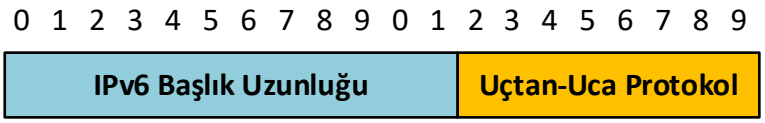


Şekil 2.19. Conta tarafından önerilen sunucu port format – uzun format yapısı

2.2.3.1.3. Başlık uzunluğu formatı önerisi

Conta tarafından önerilen [55] son yaklaşımdır. Bu öneride FL yapısındaki ilk 16 bit kullanılan IPv6 protokolüne ait başlık uzunluğunu göstermek için yapılandırılmıştır. Başlık uzunluğu ifadesi IPv6 başlığı ve uçtan-uca taşıma katmanı protokollerinden meydana gelen uzantı başlıklarının toplam uzunluğu olarak algılanmalıdır.

Bu yaklaşımda, IPv6 protokolü başlık uzunluğu şeklinde ifade edilen değer, uygulamanın kaynak ve hedef portlarını, adreslerini ve uçtan-uca taşımada kullanılan protokol için gerekli tanımlamaları içerdiğinden dolayı bu değerler yardımı ile DiffServ sınıflayıcının ihtiyaç duyduğu bilgiler kendisine sağlayabilir. Ancak Toplam Başlık Uzunluğu alanı yoktur ve bu değerin hesaplanması için ilave işlemler gereklidir. Bu da bir dezavantajtır.

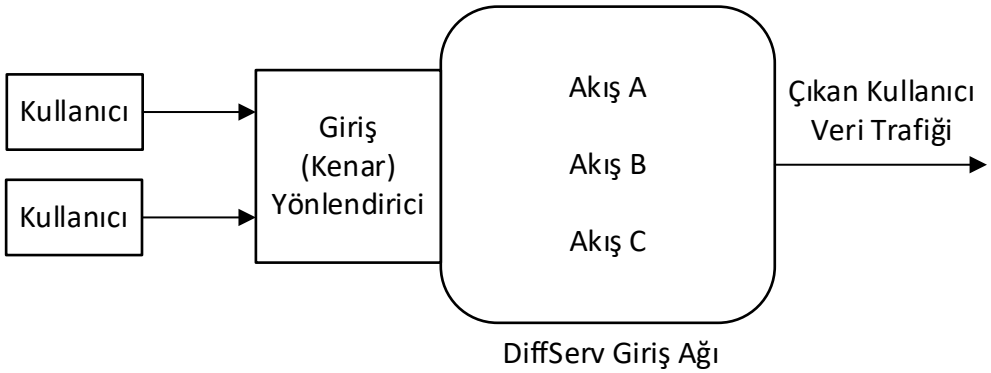


Şekil 2.20. Conta tarafından önerilen başlık uzunluğu formatı yapısı

2.2.3.2. Rajahalme önerisi

A. Conta ve J. Rajahalme tarafından hazırlanan ve 2001 yılında "A model for Diffserv use of the IPv6 Flow Label Specification" başlığıyla IETF tarafından taslak olarak yayınlanan öneridir [56].

Bu belge içerisinde, Diffserv mimarisi içerisinde IPv6 FL kullanımı için bir kavramsal model sunulmuştur. Ayrıca bu mimari için bir FL sınıflayıcı ve sınıflayıcı etiketi kullanım kuralları tanımlanmıştır.



Şekil 2.21. Diffserv mimari içerisinde FL sınıflandırıcının kullanımı

2.2.3.3. Banarjee önerisi

R.Banerjee, S.P. Malhotra, M. Mahaveer tarafından hazırlanan ve 2002 yılında "A Modified Specification for use of the IPv6 Flow Label for providing efficient Quality of Service using a hybrid approach." başlığıyla IETF tarafından taslak olarak yayınlanan öneridir [57].

Bu taslak daha önce yapılmış önermeler doğrultusunda FL alanında tanımlanmış

değerlerin kullanıbilirliğini değerlendirmektedir. Taslak ayrıca IPv6 protokolünün QoS desteklemesi için DiffServ ve IntServ yapılarını da kapsayan hibrit bir yaklaşımdır. Bu nedenle MultiServ isimle anılan deneysel bir QoS yapısı içermektedir. Bu öneride, FL alanını oluşturan bitlerin ilk 3 tanesi yaklaşım tipi olarak düzenlenmiştir. Kalan toplam 17 bit ise gerek kullanıcı gerekse uygulamalar için gerek duyulan QoS gereksimini tanımlar. Yaklaşım tipi için kullanılan ilk 3 bitin kullanımı Tablo 2.7.'de açıklanmıştır.

Tablo 2.7. Banarjee önerisine göre yaklaşım tipleri

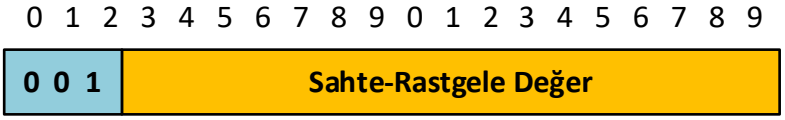
İlk 3 bitin değeri	Yaklaşım Tipi
000	Varsayılan değer, QoS ihtiyacı yok
001	Akış Etiketini tanımlamak için kullanılmak üzere rastgele bir sayı üretilmesi yapısı
010	QoS ve FL alan değerlerini yok sayan atlamadan atlamaya genişletilmiş başlık yapısı
011	Bu yaklaşımda belirtilen DiffServ PHB-ID
100	Akış Etiketi alanı içinde tanımlanan port ve protokol numarası değerleri kullanım yapısı
101	Bant genişliği, gecikme ve tampon ihtiyaç özellikleri
110	Gelecek uygulamalar için rezerve
111	Gelecek uygulamalar için rezerve

2.2.3.3.1. Varsayılan değer önerisi

Bu öneri [57], kullanılan uygulamarın veya kullanıcıların herhangi bir QoS isteği yoksa bu durumda kullanılacak olan öneridir. Bu durumda FL alanı için kullanılan değer 0 olacak şekilde ayarlanır ve bu öneride herhangi bir QoS isteği dikkate alınmaz.

2.2.3.3.2. Rastgele sayı önerisi

Bu öneride ise [57], sahte-rastgele bir sayı üretilir ve bu sayı kullanılır. Bu üretilmiş olan sayısal değer, akan trafiği etiketlemek için kullanılır ve FL alanı içerisindeki kalan 17 bitin toplam sayısal değeridir. Böylece üretilen rastgele sayının değeri 1 ile 1FFFF arasında bir sayıdır.

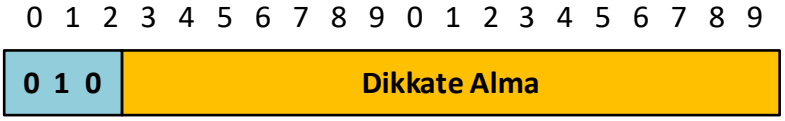


Şekil 2.22. Banarjee tarafından önerilen rastgele sayı yaklaşımı yapısı

Bu öneri IntServ yapısı içerisinde kullanılacaksa eğer üretilen rastgele sayı bir anlam ifade edecektir. Önceden öngörülemeyen üretilmiş olan bu sayının deterministik ağlarda herhangi bir anlamı olmayacaktır. Bundan dolayı IntServ yapısını içeren bilimsel çalışmalarda karşımıza çıkmaktadır [58].

2.2.3.3.3. Atlamadan atlamaya genişletilmiş başlık önerisi

Bu öneri [57], QoS isterlerinin belirlenmesi amacı ile atlamadan atlamaya genişletilmiş başlık yapısının nasıl kullanılması gerektiğini tanımlar.

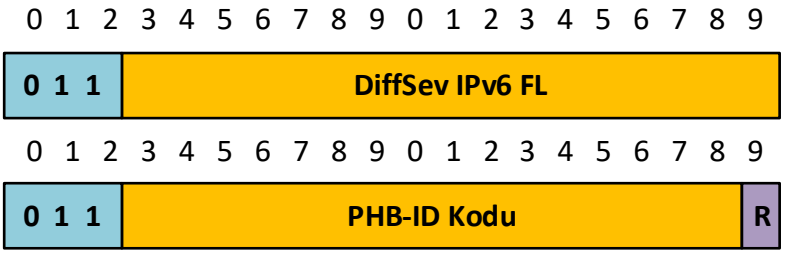


Şekil 2.23. Banarjee tarafından önerilen atlamadan atlamaya genişletilmiş başlık yaklaşımı yapısı

Bu öneride, QoS isterlerinin belirlenmesi için kullanmak amacı ile değerler bulunduran ve modifiye bir durumda atlamadan atlamaya genişletilmiş bir başlık yapısı bulunmaktadır. Bu öneride FL alanı yokmuş gibi davranılır. Bu öneri uygulamaların QoS ihtiyaçlarının tespitinde kullanılabilir ama bu ihtiyaçların karşılanmasında direk olarak kullanımı söz konusu değildir. Ancak kullanım için başka bir öneri bulunmuyorsa kullanımı uygun olabilir.

2.2.3.3.4. DiffServ PHB-ID önerisi

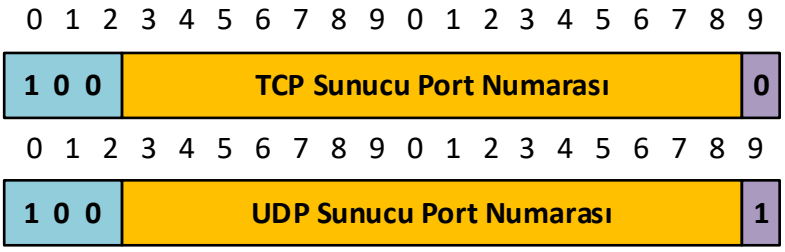
Bu öneri [57], FL alanı içerisinden sağlanan sayısal değerin DiffServ PHB-ID şeklinde kullanımının nasıl olacağını betimlemektedir. Bu öneri QoS gereksimlerinin belirlenmesi için FL içindeki değerin DiffServ sınıflayıcı aracılığı ile eşlenik olması için gerekli desteği sağlar. Akış Etiketi içindeki 16 bitin sayısal karşılığı PHB-ID olarak kullanılır. Son bit ilerisi için rezerv olarak ayrılmıştır.



Şekil 2.24. Banarjee tarafından önerilen DiffServ PHB-ID yaklaşımı yapısı

2.2.3.3.5. Port ve protokol önerisi

Bu öneri [57], FL alanından elde edilen değerin port ve protokol tanımlaması için kullanımını açıklar. Bu öneride 16 bit uygulamanın sunucu tarafından atanan port numarasını ifade ederken son bit, taşıma protokolü olarak TCP veya UDP'yi ifade etmesi için kullanılır. Bu öneri taşıma protokolü olarak sadece TCP ve UDP protokollerini destekler. Diğer protokolleri desteklemez.



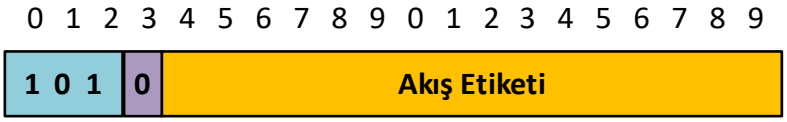
Şekil 2.25. Banarjee tarafından önerilen port ve protokol numarası yaklaşımı yapısı

2.2.3.3.6. Bant genişliği, gecikme ve tampon gereksimi önerisi

Bu öneri [57] bant genişliği, gecikme, seğirme, kayıp paket ve tampon bellek gereksinimi şeklinde tanımlanan QoS parametrelerinin kullanımını açıklar. Bu öneride seğirme ve paket kayıpları olabilecek en az değerde olması istendiğinden dolayı FL içerisinde tanımlanmasına gerek yoktur. FL alanı içerisinde tanımlanabilecek 3 parametrenin kullanımı ise aşağıdaki gibidir.

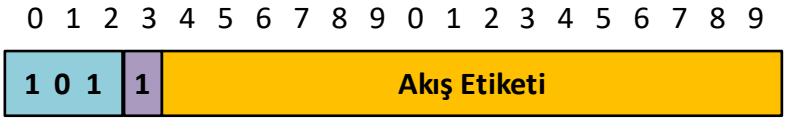
- Bant genişliği (kbps)
- Gecikme (nanosaniye)
- Tampon ihtiyaçları (bayt)

Bu öneri içerisinde FL alanı içinde bulunan 17 bitin en sonundaki tek bit Yumuşak-GZU ile Sert-GZU uygulamalarını ayırt etmek için kullanılır.



Şekil 2.26. Banarjee tarafından önerilen Yumuşak-GZU yaklaşımı yapısı

Yumuşak-GZU içeren uygulamalarda gerekli olan QoS ihtiyaçlarının hepsinin karşılanamıyor olsa bile uygulama yönetilebilir durumdadır ve dolayısı ile bu yaklaşım uygundur.



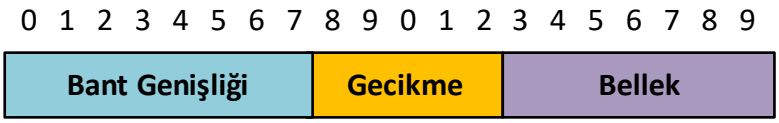
Şekil 2.27. Banarjee tarafından önerilen Sert-GZU yaklaşımı yapısı

Bu öneride ise FL alanı aracılığı ile bildirilen minimum veya maksimum değerlerin mutlaka karşılanması zorunludur. 20 bitlik alanın 16 biti bant genişliği, gecikme ve tampon gereksinimi gibi parametreleri belirtir. En uygulanabilir yaklaşımdır bundan dolayı bilimsel çalışmalarda baz alınarak kullanılmıştır [59].

2.2.3.4. Jagadeesan önerisi

H. Jagadeesan, T. Singh tarafından hazırlanan ve 2002 yılında "A Radical Approach in providing Quality-of-Service over the Internet using the 20-bit IPv6 Flow Label field." başlığıyla IETF tarafından taslak olarak yayınlanan öneridir [60].

Bu öneride de Banarjee önerisindeki gibi bant genişliği, gecikme ile tampon bellek gereksinimlerinin nasıl olması gerektiği açıklanmaktadır. Ancak bu öneride ifade edilen parametreleri kullanabilmek için FL alanı içindeki tüm bitlerin kullanımı zorunludur.

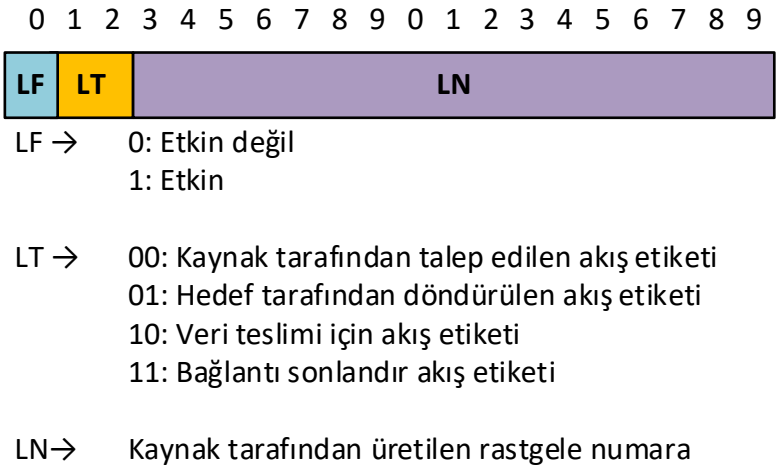


Şekil 2.28. Jagadeesan tarafından önerilen FL alanı değerleri

Bu öneride FL alanının ilk sırada bulunan 8 biti bant genişliği gereksinimini belirtmektedir. Bu 8 bitlik grubun ilk biti ise bant genişliğini ifade eden değerinin maksimum veya minimum olarak istendiğini gösterir. Kalan bitler ise yani kalan 7 bit bant genişliğinin sayısal değerini ifade eder. Sıradaki 5 bit ise gecikme değerini ve son 7 bit ise tampon bellek gereksinimini belirtir.

2.2.3.5. Lin önerisi

C. Lin, P. Tseng, ve W. Hwang tarafından 2006 yılında "End-to-End QoS Provisioning by Flow Label in IPv6" başlığıyla yayınlanan makale ile önerilen yaklaşımdır [61].



Şekil 2.29. Lin tarafından önerilen FL alanı

Uçtan uça QoS çözümünde kullanılmak üzere yeni bir FL çözüm önerisi sunmuşlardır. Bu yaklaşımda etiketin ilk biti Etiket Bayrağı olarak tanımlanır ve FL alanı kullanılacak ise bu ilk bit 1 olarak atanır. Sonraki 2 bit FL tipini tanımlamak amacıyla Etiket Tipi olarak kullanılır. Son 17 bit ise FL içerisinde akışı tanımlamak için

kullanılmıştır. Bu FL tanımı kullanılarak kaynaktan hedefe olan yol boyunca her bir akış benzersiz bir şekilde tanımlanarak uçtan uça QoS desteği verilir.

2.2.3.6. Chakravorty önerisi

S. Chakravorty tarafından 2008 yılında "Challenges of IPv6 Flow Label implementation" başlığıyla yayınlanan makale ile duyurulan ve ardından "IPv6 Label Switching Architecture (6LSA)" başlığıyla IETF tarafından taslak olarak yayınlanan ve 6LSA olarak bilinen öneridir [62,63].

Tablo 2.8. Chakravorty tarafından önerilen modele ait etiket değerleri

İlk 3 Bit	**Sonraki 4 Bit**	**Amaç**
Yok (000)	0000	
Alana Özgü (000)	0001 – 1111	
Sanal Özel Ağ (001)	0001	IPSec - Tünel modu
	0010	IPSec – İletim modu
	0011	Özel şifreleme
	0100	VRF
	0101	Uç ağ özellik
	0110 – 1111	Rezerv
QoS Arttırma (010)	0001	DiffServ
	0010	RSVP
	0011	RSVP-TE
	0100	SIP
	0101	H323
	0110	Büyük Dosya
	0111	Devre Benzetim
	1000	Sabit Bant Genişliği
	1001	Video Akışı
	1010	Çoklu Gönderim
	1011	Her Yere Gönderim
	1100	Kuyruk Bekliyor
	1101	Öncelik Duyarlı
	1110	Kurum Sinyal
	1111	Rezerv
Kapsülleme (011)	0001	IPv6 içinde IPv6
	0010	IPv6 içinde IPv4
	0011	Diğer IPv6
	0100	Kurum Özel
	0101 – 1111	Rezerv
Kurum Özel (111)	0000 – 1111	Rezerv

Ağ düğüm noktası özel davranışları ve uygulamaya özel akış karakteristikleri içeren FL tanımlaması sunar. Bu FL tanımlaması temelde Evrensel Etiket Değeri ve Yerel Etiket Değeri olmak üzere iki bölümden oluşur. Evrensel Etiket Değeri 7 bit içerir ve

uygulamaya özel akış karakteristiklerini ifade eder, Yerel Etiket Değeri de 13 bit içerir ve düğüm özel davranış gereksinimlerini ifade eder.

2.2.3.7. Akış etiketi kullanım yaklaşımlarının değerlendirilmesi

Yukarıda incelene tüm öneriler içinde Conta önerisi olarak incelenen yapı kompleks ve karmaşık bir yapıya sahiptir. Çünkü Akış Etiketi alanı için RFC 2460 belgesi ile ifade edilen durağan yapısına karşı olarak bu öneri ağ üzerinde hareket eden FL içeriğinin yönlendiricilerin kendi aralarında iletmek istedikleri bilgileri de taşıyabilmesi amacı ile değişken olabilmesi önerilmiştir ve dolayısı ile kullanım alanı olarak çok uygun bir yapı sunmamaktadır.

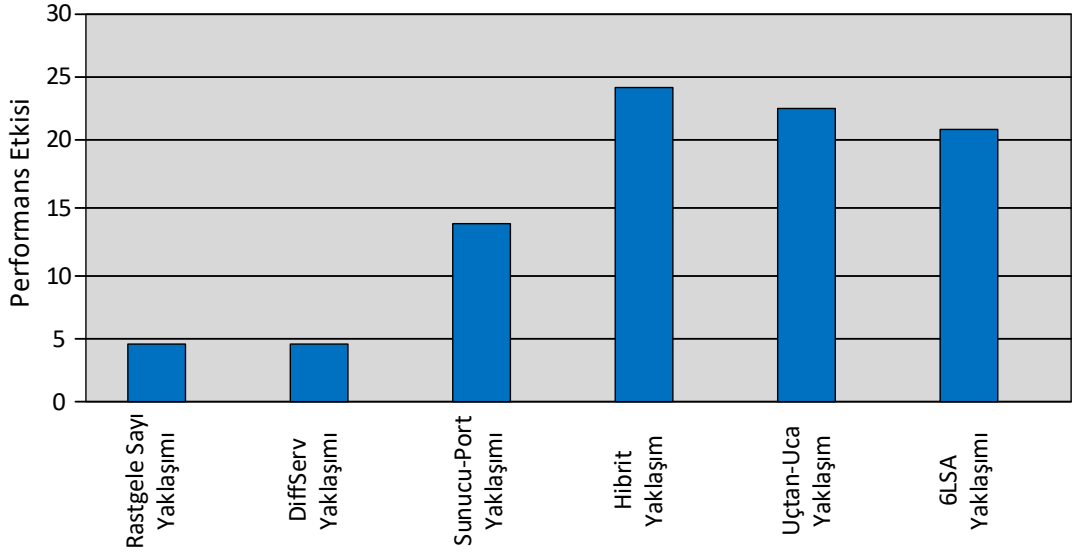
Jagadeesan önerisi ise uygulama gereksinimleri için herhangi bir sınıflama yamamaktadır ve sadece ihtiyaç duyuyan servis parametrelerini belirtmektedir. Bundan dlayı IntServ mimarisi için uygun bir yapı önermektedir. Literatürde herhangi bir kullanımına rastlanmamıştır.

Lin önerisinde ise model her bir akışı ayrı ayrı benzersiz olarak tanımladığı için IntServ mimarisine yakın olmaktadır. Bu modelin dezavantajı ise çekirdek ağlarda her bir akışı benzersiz olarak tanımlayan akış etiketinin çözümlenmesi ve doğrulanması uzun bir gecikme ortaya çıkarabilecektir.

Chakravorty tarafından önerilen yaklaşımda DiffServ mimarisine yakın bir mimari sunulmaktadır. Bu şekilde bir akış etiketi tanımlamasının avantajı servislerin gerektirdiği benzer kalite ihtiyacı olan birden fazla akışı yol boyunca birleştirebilir.

Ancak bu yaklaşımda sunulan 6LSA akış etiketi önerisi sadece IPv6 Etiket Anahtarlamalı mimarilerde kullanılabilir. IPv6 ağındaki gerçek bir uygulamadan önce olası tüm sınırlamaların keşfi tüm bu mevcut tanımlamanın performans karşılaştırması için gereklidir.

Yukarıda anlatılan önerileriçerisinde Banarjee tarafından önerilen model en esnek ve uygulanabilir öneri olmaktadır. İçerisinde hibrit bir yapı bulunmasından dolayı tüm mimarilere yani hem IntServ mimariye hem de DiffServ mimariye uygundur. Bu esnek yapısından dolayı literatürde FL alanı kullanımı ile ilgili yapılan çalışmalarda ve yayınlarda Banarjee önerilerinin uygulanması bulunmaktadır [58,59]. Akış Etiketi yaklaşım formatlarının karşılaştırması literatürde mevcuttur [64]. Diğerleri göz ardı edilerek Banarjee yaklaşımının farklı modelleri ile Chakravorty (6LSA) ve Lin (Uçtan-uca) modellerinin performans grafiği çıkarılmış [65] ve sonuç Şekil 2.29.'da gösterilmiştir.



Şekil 2.30. Çeşitli FL yaklaşımlarının performans grafiği

2.2.4. IPv6 ve akış etiketi hakkında önceki çalışmalar

IPv6 başlığı içerisinde yer alan FL alanı için kesin ve net kullanım tanımlaması olmadığından dolayı literatürde ağırlıkla bu alan için kullanım yaklaşımları ve bu yaklaşımların türevleri yer almaktadır. Temel yaklaşımlar IETF tarafından taslak olarak yayınlanmıştır [55-62]. Öneriler deneysel sonuçlarla desteklenmeye de çalışılmış [61], hatta öneriler karşılaştırmalı olarak değerlendirilmiştir [65].

Araştırmalarda FL yeteneklerinin bazı spesifik teknolojilere uygulandığı görülmüştür. Bunlardan biri MPLS ve DiffServ gibi faklı teknolojilerin FL ile kombine edilmesidir

[66]. Bir diğer araştırmada, IPv6 ağları üzerinde DiffServ domain içerisinde FL tabanlı paket zamanlaması üzerinde durmuştur [67]. Bunun yanında gerçek zamanlı trafiğin QoS ihtiyacını garanti altına almak için kaynak tarafından belirlenen QoS parametreleri doğrultusunda, omurga kaynaklarının kullanımını arttırmak için MPLS ve FL karışımından oluşan bir hibrit şema önerisi ile karşılaşılmıştır [68]. FL ile MPLS ve DiffServ gibi diğer teknolojilerin hibrit bir şekilde kullanılması başka araştırmalara da konu olmuştur [67-69].

Gerçek zamanlı trafiğin QoS ihtiyacı için FL kullanımı olduğu gibi [67-70] kablosuz ağların da QoS ihtiyacı için FL kullanımı araştırmalara konu olmuştur [71-73]. Bir başka araştırma ise, kablosuz tasarsız ağlar üzerinde FL kullanımı önerilmekte ve bu öneride Banarjee tarafından yapılan Bant genişliği, Gecikme ve Tampon İhtiyaçları Yaklaşımı'ndan türemiş bir FL format yapısı önerilmektedir [74]. İnternet üzerindeki IPv4 düğümleri üzerinde QoS desteği için IPv6 FL haritalama yine önceki öneriler doğrultusunda bir türev format önerisi yapılmış ve bu öneri simülasyon ortamında test edilmiştir [75]. Homojen olmayan ağlar üzerinde uçtan-uça QoS desteği için yine türev bir format yaklaşımı yapılmıştır [76].

Bu çalışmada, önceliklendirme için kullanılacak olan yönlendirme protokol metrik değeri IPv6 başlığı içerisinde bulunan FL alanına yazılacak ve uçtan-uca önceliklendirme desteğinin sağlanabilmesi için IPv6 protokolü kullanılacaktır. Bundan dolayı bu çalışma içerisinde Banarjee tarafından yapılan DiffServ PHB-ID yaklaşımı tabanlı bir FL kullanım önerisi yapılacaktır. Yapılan bu kullanım önerisi bu çalışmada referans alınan DiffServ modeline de uygundur.

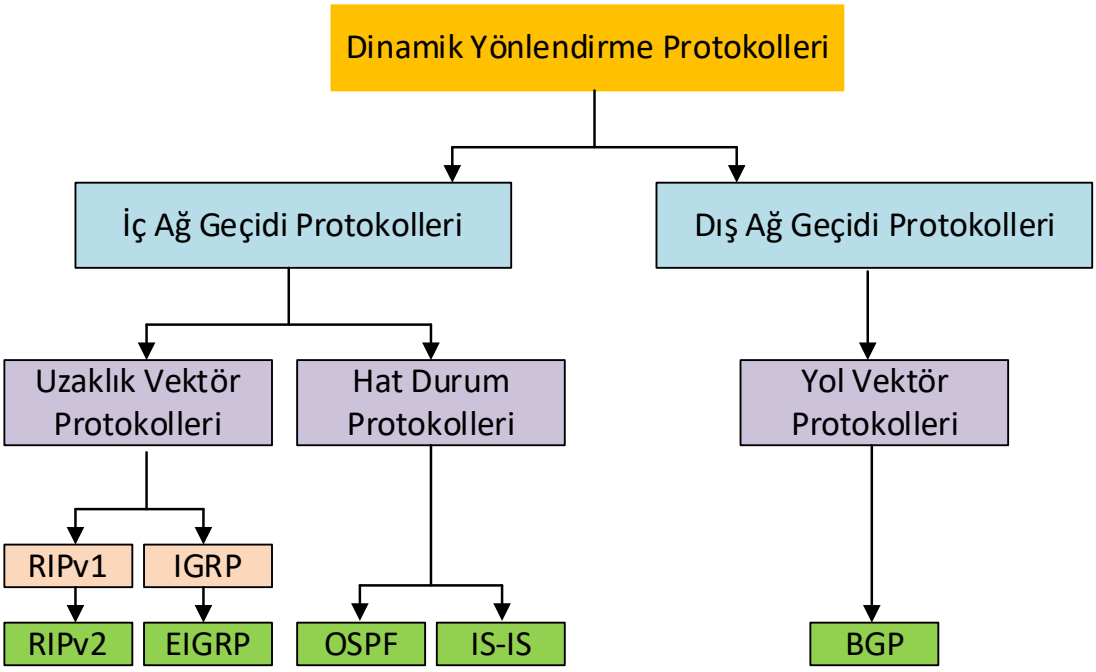
2.3. Yönlendirme

Yönlendirme, bir göndericiye ait verinin aynı veya farklı ağ üzerinde bulunan alıcıya nasıl ve hangi yol üzerinden gönderileceğini belirleme yöntemidir.

Ağ üzerinde yönlendirme işlemi, yönlendirici olarak isimlendiren cihazlar tarafından gerçekleştirilir. Yönlendiriciler veri paketlerinin yönlendirmesini paket içerisinde yer

alan IP başlık bilgisi içerisinde bulunan hedef IP adresine göre gerçekleştirir. Verinin gideceği yolun tanımlaması statik olarak yapılabileceği gibi yönlendiriciler üzerinde bulunan yönlendirme protokolleri aracılığı ile dinamik olarak da yapılabilir.

Dinamik yönlendirme protokolleri de otonom sistem (AS) içerisinde çalışanlar İç Ağ Geçidi Protokolleri (IGPs) ve otonom sistemler arasında çalışanlar da Dış Ağ Geçidi Protokolleri (EGPs) olarak temelde ikiye ayrılmaktadır. AS içerisinde çalışan protokollerde çalışma algoritmalarına göre yine kendi içerisinde ikiye ayrılmaktadır. Bu yapı Şekil 2.30.'da gösterilmiştir.



Şekil 2.31. Dinamik yönlendirme protokolleri

Burada önemli kavramlarda biri de otonom sistem kavramıdır. Otonom sistem, IP ağlarında bulunan havuz yapısı olarak ifade edilebilir. Amaç IP ağlarının coğrafi konumlarına göre gruplanması ve böylece yönetimin kolaylaştırılmasıdır.

2.3.1. Uzaklık vektör protokolleri

Bu protokol takımında, verinin ilerleyeceği yollar uzaklık ve doğrultu olarak ifade edilir. Uzaklık, geçilen aktif yönlendirici yani ağ katman aygıtı sayısına göre belirlenir

ve “hop count” olarak ifade edilir. Doğrultu da verinin çıkış arabirimini (interface) ifade etmektedir. Uzaklık Vektörü Protokolleri, en iyi yolu belirlemede Bellman-Ford algoritmasını kullanırlar. Sistem içinde bulunan bir yönlendirici, içinde bulunduğu tüm topolojinin yapısal haritasına sahip değildir. Bu algoritmayı kullanan protokollerde yönlendirici, yönlendirme tablosundaki (routing table) içeriğinin sadece bir bölümü değişse dahi, bütün yönlendirme tablosu içeriğini periyodik bir şekilde komşularına iletir. Bu çalışma mantığı, büyük ağlarda için önemli bir derecede trafiğe oluşturur. Bununla birlikte, paketler gönderilirken geçtikleri düğümler üzerinde içerikleri değiştirildiğinden, güncelleme ağ üzerinde yavaş olur. Uzaklık vektörü algoritması kullanan protokoller, en iyi yol kararı verirken basit algoritmalar kullanırlar ve bundan dolayı yönlendiricilerin işlemcisine fazla yük binmez; bununla beraber bazen en doğru yolu seçme konusunda başarısız olurlar. Bu protokoller; hiyerarşik bir bütünlük içermeyen basit ağlarda ve yakınsama yani topolojideki bütün yönlendiricilerin bütün ağları öğrenme süresinin önemli olmadığı durumlarda tercih edilir. RIPv1, RIPv2, RIPng, IGRP ve EIGRP protokolleri bu gruba dâhil olan protokollerdir.

2.3.1.1. RIP v1

İlk olarak Xerox Ağ Sistemleri (XNS) protokol kümesi içinde kullanılmış olan Yönlendirme Bilgisi Protokolü (RIP)’in ilk versiyonudur ve RFC 1058 belgesi ile tanımlanmıştır.

RIP, uzaklık-vektör tabanlı bir yönlendirme protokolüdür. Yönlendiriciler, kendilerine ait yönlendirme tablolarının tüm içeriğini 30 saniyede bir bütün ara yüzlerden komşu yönlendiricilere broadcast ile gönderir. En iyi yol seçiminde sadece hop count (paketlerin geçmiş olduğu yönlendirici sayısı) değeri dikkate alınır ve maksimum hop count değeri 15‘tir. Bunun anlamı, hop count değeri 16 olan ağlar erişilemez olarak ifade edilir. RIPv1 sadece sınıflı (classful) yönlendirmeyi kullanır. Yani bu versiyonda ağdaki tüm cihazlar aynı alt ağ maskesi (subnet mask) değerini kullanmak zorundadır [77].

2.3.1.2. RIP v2

RFC 1721, RFC 1722 ve RFC 2453 belgeleri ile tanımlanmış olan uzaklık-vektör tabanlı yönlendirme protokolüdür. Davranış özellikleri RIPv1 ile aynı olmasına rağmen prefix yönlendirme olarak da adlandırılan sınıfsız yönlendirme kullanır. Bu yöntem ile yönlendirme güncellemeleri sırasında alt ağ maske değeri de yönlendirme tablosu içerisinde gönderilir. Böylece çeşitli büyüklükteki ağlar oluşturulabilir ve bu ağlar RIPv2 ile birbirlerine bağlanabilir. Bu temel özelliğinin yanında Değişken Uzunlukta Alt Ağ Maskeleme (VLSM) desteği vardır ve Özetleme ve Kimlik Onaylama desteği de sunmaktadır [78].

2.3.1.3. RIPng

Gelecek nesil internet protokolünün yani IPv6'nın desteklenmesi için RIPv2'nin gelişmişi olarak RFC 2080 belgesi içinde tanımlanmıştır. Kendinden önceki versiyonlar ile arasındaki temel fark kimlik onaylama işlemi kullanmamasıdır. Çünkü desteklemiş olduğu IPv6 protokolünün içinde bu işlemi yapan IPSec desteği bulunmaktadır [78].

2.3.1.4. IGRP

İç Ağ Geçidi Yönlendirme Protokolü (IGRP), Cisco System tarafından geliştirilen uzaklık-vektör tabanlı bir yönlendirme protokolüdür ve bu firmanın lisansı altındadır. RIPv1 gibi sınıflı yönlendirmeyi kullanır. IGRP'de olabilecek en büyük hop count değeri 255'tir ve hop count değeri sadece 15 olan RIP protokolüne göre oldukça önemli bir gelişmedir. Bununla birlikte yönlendirme metriği olarak kullanılan tek özellik hop sayısı değildir. RIP protokolünden farklı olarak IGRP, hat gecikmesi, bant genişliği, güvenilirlik ve yük durumunu da metrik olarak kullanır. Bunların yanında geliştirilen bir başka parametre ise güncelleme tablolarının 90 saniyede bir gönderilmesidir [78].

2.3.2. Hat durum protokolleri

Bu protokoller ile yapılandırılmış yönlendiriciler, diğer yönlendiricilerden aldıkları bilgiler aracılığı ile tüm ağın yapısal topoloji haritasına sahip olabilirler. Kısaca herhangi iki nokta arasındaki mevcut tüm yolların bilgilerine sahiptirler. Bu sayede tüm alt ağları yapılarını tek bir ağaç yapısı içerisinde değerlendirip, Önce En Kısa Yol (SPF) algoritması sayesinde hedefe hangi yoldan gitmesi gerektiğine ait en doğru kararı verebilirler. Bununla birlikte topoloji haritası bir kez oluşturulunca, periyodik güncellemeler göndermeyip, sadece değişiklik meydana geldiğinde, küçük paket ile sadece değişikliğe dair güncelleme gönderilir ve bu çalışma mantığı aşırı trafik oluşmasını önler. Paketler, ağ üzerinde düğümleri geçerken üzerinde herhangi bir değişiklik yapılmadan aktarıldığı için uzaklık vektörü protokollerinde bulunan hızlı yakınsama sorunu bu protokollerde yoktur. Bunun yanında, karmaşık algoritmalar kullandıkları için, uzaklık vektör protokollerine göre daha fazla kaynağa ihtiyaç duyarlar. Hat durum protokolleri, hiyerarşik yapılı büyük ağlarda ve yakınsama süresinin kısalığının önemli olduğu durumlarda tercih edilir. OSPF ve IS-IS protokolleri bu grup altında yer almaktadır.

2.3.2.1. OSPF

En Kısa Yola Öncelikli (OSPF) protokolü, İnternet Mühendisliği Görev Grubu (IETF) tarafından geliştirilmiştir. OSPF, bir hat durum protokolüdür. Bu yönlendirme protokolleri, yapıları gereği topolojinin tamamını görebildikleri için ağda herhangi bir değişiklik meydana geldiğinde tetiklenmiş güncelleme bilgisi gönderirler. Bu sayede yönlendiriciler ağüzerindeki iki nokta arasında mevcut tüm yolların bilgisine sahip olduktan sonra Dijkstra algoritmasının bir türevi olan SPF algoritmalarını çalıştırarak hangi yol en iyi yoldur kararını verebilirler. Hat durumlarını güncellemek için her 30 dakikada bir periyodik olarak güncelleme paketleri gönderirler [79].

Sınıfsız bir yapıya sahiptir ve VLSM desteklediği için daha etkili adresleme yapar. Çoklu yol desteği vardır ve bu yollar üzerinde yük dengelemesi yapabilmektedir. OSPF protokolü en iyi yola ait metrik değerini hesaplarken “Cost” olarak adlandırılan

bant genişliği değeri ile ters orantılı bir değer kullanır. En iyi yol, en düşük cost yani metrik değerine sahip yoldur ve en iyi yol bilgisi yönlendirme tablosunda yer alır [79].

RFC 2740 ve RFC 5340 belgeleri ile duyurulan OSPFv3, yeni nesil IPv6 protokolünü desteklemek için geliştirilmiştir.

2.3.2.2. IS-IS

Orta Seviyeden Orta Seviyeye Sistem (IS-IS) protokolü Digital Equipment Corporation tarafından geliştirilmiştir. Uluslararası Standartlar Teşkilâtı (ISO) tarafından 1992 yılında ISO 10589 belgesi ile ağ aygıtları arasında haberleşme için standart haline getirilmiştir. Daha sonra RFC 1195 belgesi ile IP protokolünü destekleyerek Bütünleştirilmiş IS-IS adını almıştır. RFC 5308 belgesi ile IPv6 ağlarını destekler hale gelmiştir.

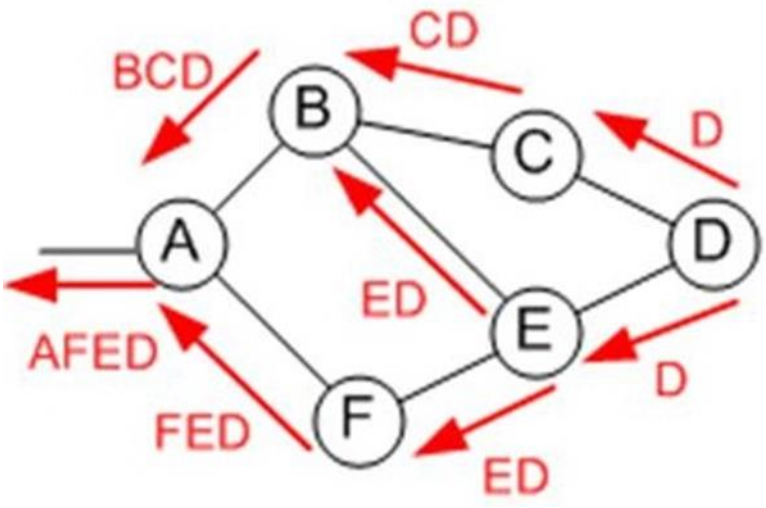
IS-IS hat durum protokol ailesindendir ve OSPF gibi en iyi yolun hesaplanmasında Dijkstra's algoritmasını kullanır. OSPF doğasında IP için tasarlanmıştır ve OSI modelinin 3. katmanında çalışmaktadır ancak IS-IS, 2. katman protokolüdür ve yönlendirme bilgi mesajlarının taşınmasında IP kullanmaz. IS-IS protokolü, OSPF gibi VLSM ve kimlik onaylama desteği bulunmaktadır. Çoklu yol üzerinde yük dengelemesi yapabilmektedir [80].

2.3.3. Yol Vektör Protokolleri

İnternet omurgası gibi birçok otonom sistemden kurulu büyük ağlar için yeni bir yaklaşım geliştirilmiştir ve bu yol vektör yönlendirme adını almıştır. Yol vektör yönlendirme uzaklık vektör yönlendirmeye benzer bir çalışma yürütür. Uzaklık vektör algoritmasına göre daha hızlı yakınsama sunmaktadır ve yol seçiminde esneklik sunmaktadır. Yönlendirme güncellemelerinde IP ağ adresleri ve otonom sistem numaralarını gönderirler. Güncellemeler bütün yönlendirme tablonun gönderilmesi şeklinde değil sadece değişen kısımlar güncellemeye eklenir. Güncelleme tablolarının

taşınmasında TCP protokolü kullanıldığından dolayı onaylama mekanizması bulundurmazlar.

Yol vektör yönlendirmede her bir otonom sistem içerisinde tüm otonom sistem adına hareket eden ve uygulamalarda birden fazla olabilen bir konuşmacı düğüm bulunur. Bu konuşmacı düğüm içinde bulunduğu otonom sistem için bir yönlendirme tablosu oluşturur ve komşu otonom sistemlerde bulunan diğer konuşmacı düğümlere gönderir. Bu konuşmacı kendi içinde veya diğer otonom sistemlere olan yol bilgilerini duyurur. Bu algoritmada komşu bir düğümden alınan duyuru üzerine her bir düğüm kendisini de ekleyerek diğer komşusuna aktadır. Böylece en sondaki düğüm duyuruyu aldığında hedefe doğru alternatif yollar ve her bir yolun kaç düğümden oluştuğunu bilgisine sahip olur.

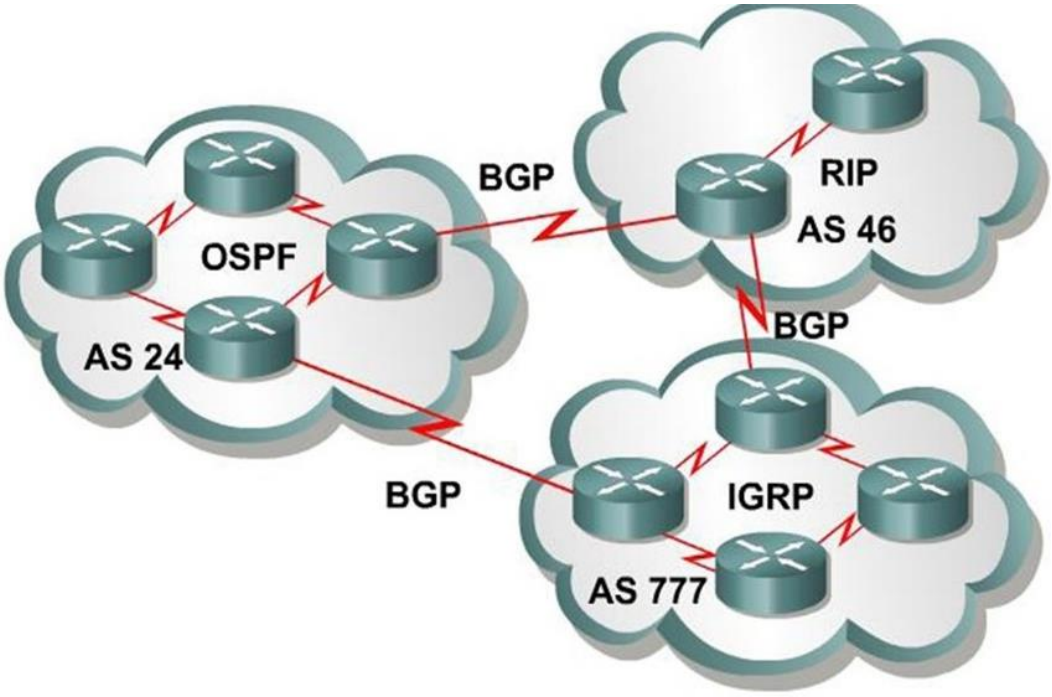


Şekil 2.32. Yol vektör algoritması çalışma mantığı

2.3.3.1. BGP

Sınır Geçit Protokolü (BGP), IETF tarafından RFC 1771 belgesi ile tanımlanan, otonom sistemler arası yönlendirme için tasarlanmış, yol vektör algoritması kullanan bir yönlendirme protokolüdür. BGP protokolü günümüzde internet altyapısının en temel protokollüdür [81].

BGP protokolü, sınıfsız çalışan bir yönlendirme protokolüdür ve VLSM desteği bulunmaktadır. Hem IPv4 hem de IPv6 desteği sağlar.



Şekil 2.33. BGP protokolü çalışma yapısı

BGP protokolü güncellemelerini yollarken, IP prefixleri ve otonom sistem bilgisini paylaşırlar. BGP, TCP üzerinden güncellemelerini gönderir ve onaylama işlemi BGP protokolünde mevcut değildir. Zaten TCP protokolü bu işlemi kullandığı için harici olarak tekrar bu operasyona ihtiyaç duyulmamaktadır. BGP protokolü güncellemelerini her defasında topluca yollamazlar sadece değişen bilgileri paylaşırlar. Ancak periyodik olarak komşularına çalışır olduklarına ilişkin mesaj gönderirler [81].

2.3.4. Protokoller arası rota ve metrik dağıtımı

Yönlendirme protokolleri, diğer yönlendirme protokollerinden öğrendikleri rotaları, statik olarak ayarlanmış rotaları ve direk bağlı oldukları rotaları duyururlar ki buna dağıtım adı verilir.

Her bir protokol rota hesaplamasında farklı bir algoritma kullanarak farklı metrik değerlerini göz önüne aldığından dolayı bir protokol tarafından hesaplanan metrik değeri direk olarak bir başka protokol tarafından kullanılamaz. Protokol yapıları içerisinde metrik dönüşüm işlemi tanımlı olmadığından dolayı farklı protokoller arası rota dağıtımında kullanılacak metrik değerleri manuel olarak yöneticiler tarafından tanımlanır.

Aynı hedef ağ için farklı protokollerden gelen rota bilgilerinden hangisinin kullanılacağı ise protokollerin sahip oldukları Yönetimsel Uzaklık (AD) değerine göre belirlenir. Düşük olan AD değerinin önceliği bulunmaktadır [82].

2.3.5. Yönlendirme hakkında önceki çalışmalar

Ağlar üzerinde QoS desteği için etkin bir yönlendirme yapılması oldukça önemlidir. Bundan dolayı yönlendirme protokollerinin etkin önceliklendirme yaklaşımlarında FL kullanımı literatürde bulunmaktadır [83,84].

Yönlendirme protokollerinin en iyi yol seçimi yaparken kullandıkları metrik değerlerine ek olarak kuyruk gecikmesini de rota seçim algoritmasına katan ve yönlendirmenin kuyruk gecikmelerini de dikkate alarak yapılmasını sağlayarak QoS etkisini arttırmaya çalışan çalışmanın [85] yanında yönlendirme protokol metrik değerlerinin ağırlıklarının değiştirilerek trafik mühendisliği yapılması servis kalitesine olumlu etki edeceğini belirten çalışmalar mevcuttur [86]. Bu doğrultuda ağ kaynaklarının kontrolü için politika tabanlı çok katmanlı QoS mimarisi, yönlendirme protokolleri için önerilmiştir [87]. Politika tabanlı QoS yönetimi başka çalışmalara da konu olmuştur [88,89].

Servislerin ihtiyaç duyduğu QoS desteği için veri trafiğinin yönlendirmesi de araştırma konusu olmuştur. Bunlardan biri de trafik mühendisliği ile, MPLS ve DiffServ ağlar üzerinde OSPF için en kısa olan yolun seçimi yerine tıkanıklığın önlenmesi ve yük dengelemesi için en uygun yolun seçimini amaçlayan çalışma [90] olmasına rağmen

QoS desteği ile yolların yeniden oluşturulması problemine henüz kesin bir çözüm bulunamamıştır [91-93].

Bu çalışmada önceliklendirme değeri olarak yönlendirme protokolü metrik değeri kullanılacaktır. Yönlendirme protokolü olarak, tek otonom sistemden meydana gelen topoloji üzerinde RIPng protokolü seçilmiştir. Birden çok otonom sistemden oluşan ikinci topolojide ise otonom sistem içinde RIPng protokolü, otonom sistemler arası BGP protokolü yönlendirme protokolü olarak kullanılacaktır.

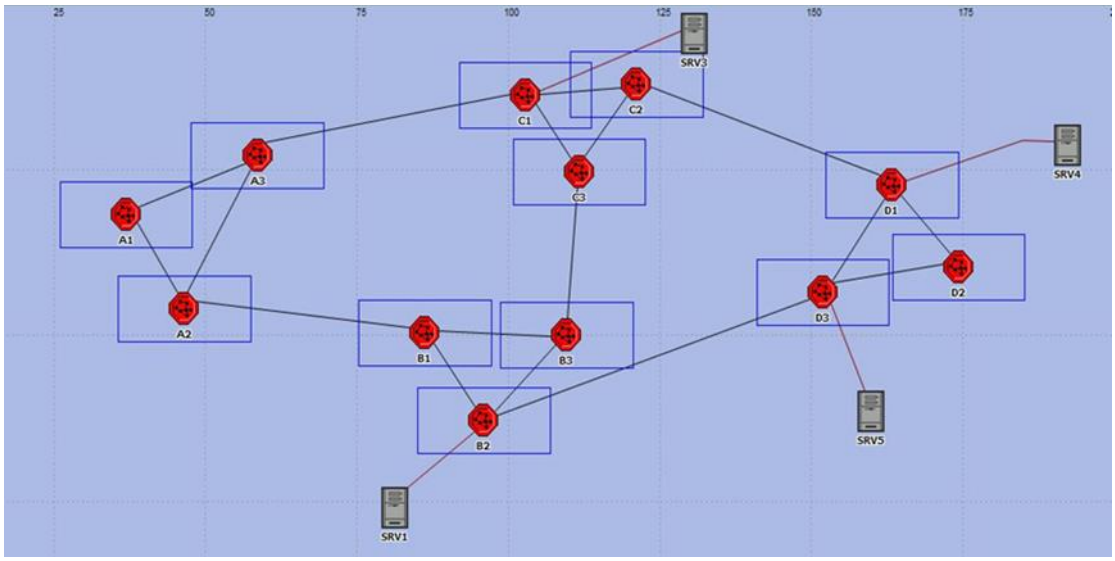
BÖLÜM 3. IPv6 TABANLI VE GERÇEK ZAMANLI (SES-VİDEO) UYGULAMALARI DESTEKLEYEN REFERANS TOPOLOJİNİN OLUŞTURULMASI

Proje temelde iki adımdan oluşmaktadır. İlk adımın amacı, geliştirilen algoritmanın başarısını ölçmek amacı ile referans bir topoloji oluşturmaktır. Bunun için daha önceden çalışılmış örnek topolojiler referans alınmıştır [19,20]. Bu referans topoloji üzerinden alınan başarım değerleri ikinci adımda uygulanacak önceliklendirme algoritmasının başarım değerleri ile karşılaştırmak için kullanılacaktır. Burada gerçek hayatta karşılaşabileceğimiz her türlü durum göz önüne alınarak hem düşük metrik değerine sahip hem de yüksek metrik değerine sahip hedef ağlara ait istatistiki veriler toplanacaktır. Bu çalışmada kullanılan test ortamı ve oluşturulan referans topoloji OPNET programı üzerinde planlanmıştır. Uçtan-uca destek verebilmek için topolojiyi oluşturan tüm cihazların IPv6 desteği bulunmaktadır ve IPv6 protokolü tanımlanacaktır.

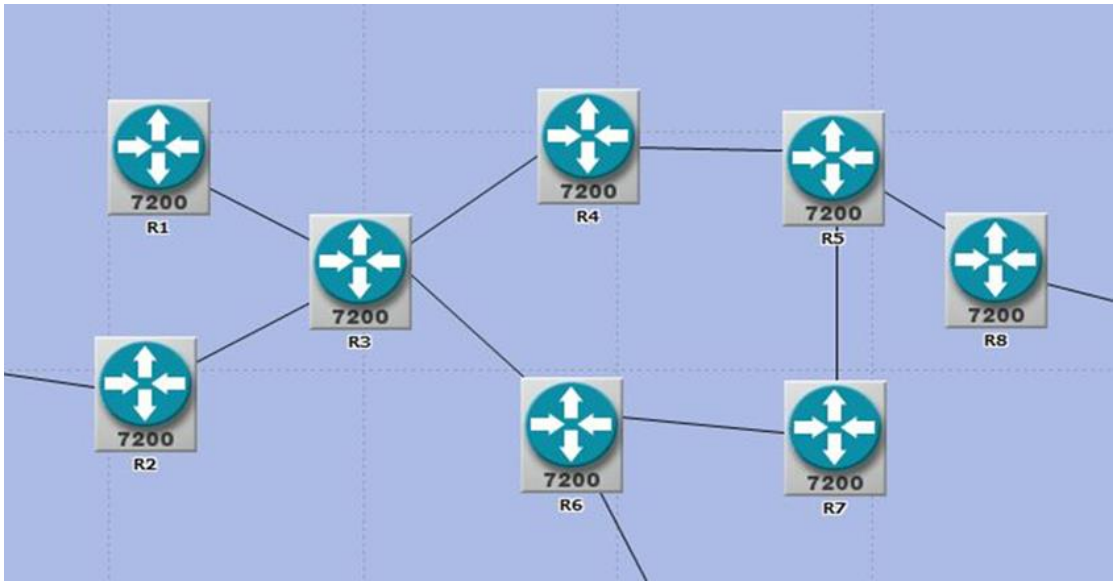
Oluşturulan referans topoloji ağı Şekil 3.1.'de görüldüğü gibi A, B, C ve D isimleri verilen 4 otonom sistemden ve 12 alt ağdan oluşmaktadır. Burada amaç gerçek internet dünyasına mümkün olduğunca yakın değerler elde etmeye çalışmaktır. Bundan dolayı tek bir otonom sistem veya tek bir alt ağ içinde çalışmak yerine otonom sistemler arası veri trafiğini de inceleyebilmek adına birden fazla otonom sistemden oluşan ağ yapısı tercih edilmiştir. Her bir otonom sistem üç alt ağ ve her bir alt ağ da Şekil 3.2.'de gösterildiği gibi sekiz yönlendiriciye sahiptir.

Bu topoloji içerisinde kullanılan yönlendiriciler Cisco 7200 serisi yönlendiricilerdir. Yönlendiriciler arasındaki hatlar, yoğun trafik altında aşırı yük oluşturmamak için 1,544 Mbps bant genişliğine sahip PPP DS1 hatları yerine 44,736 Mbps bant genişliği sağlayan PPP DS3 ile oluşturulmuştur. Ses ve video trafiğine ek olarak, gerçek

ortamlarda olduğu gibi diğer uygulamalara ait paket trafiği modellemek amacı ile sisteme FTP ve HTTP sunucular da topolojiye eklenmiştir. Böylece ortamda GZU trafiği ile diğer uygulamalara ait trafik de modellenmiştir.



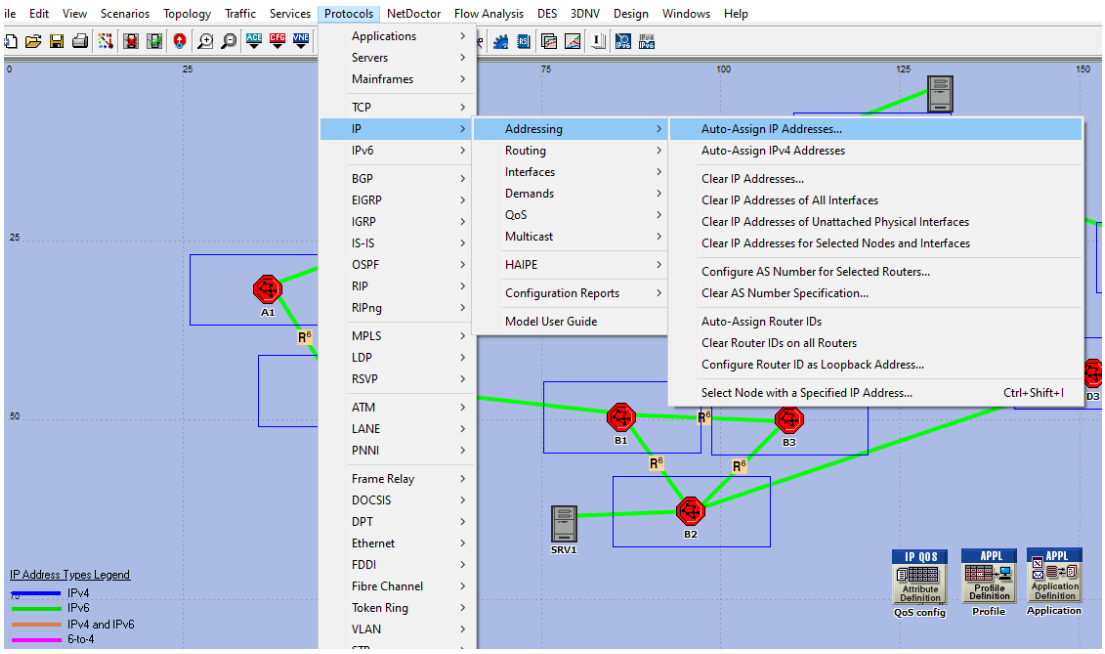
Şekil 3.1. Referans topoloji ağ yapısı



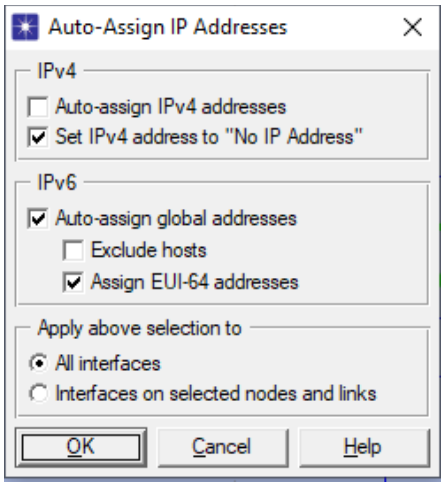
Şekil 3.2. Otonom sistemleri oluşturan her bir alt ağ modelinin iç yapısı

3.1. Ağ Aygıtlarına IPv6 Adresi Tanımlaması

Tasarlanan ağ yapısı üzerinde uçtan-uca QoS desteği verilebilmesi ve hedef ağ metrik değeri üzerinden elde edilecek önceliklendirme değerinin kaynaktan hedefe kadar değiştirilmeden taşınabilmesi için kullanılacak olan FL alanı desteğinden dolayı adresleme protokolü olarak IPv6 seçilmiştir. Bundan dolayı topolojiyi oluşturan tüm donanımsal aygıtların IPv6 desteği bulunmaktadır. Bu donanımların her bir ara yüzüne Şekil 3.3. ve Şekil 3.4.'te gösterildiği gibi IPv6 adresi verilmiştir.



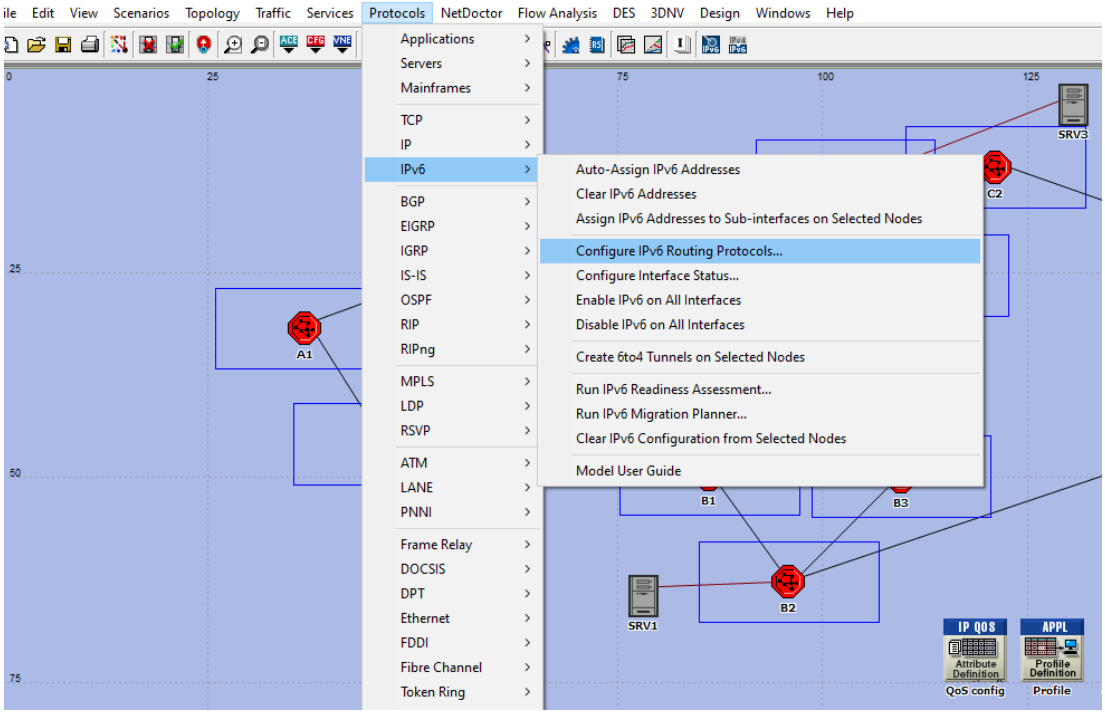
Şekil 3.3. Protokol seçim ve tanımlama penceresi



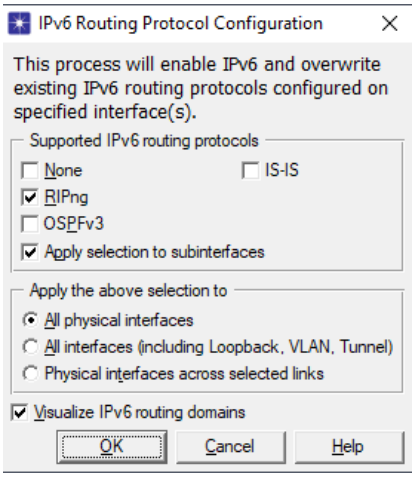
Şekil 3.4. IPv6 adreslerinin tüm arabirimlere otomatik atanması tanımlanması

3.2. Yönlendirme Protokolünün Tanımlanması

Donanımsal aygıtların tüm ara yüzlerine IPv6 adresi tanımladıktan sonra sırada yönlendirme protokolünün tanımlaması vardır. Burada dikkat edilmesi gereken nokta IPv6 desteği olan yönlendirme protokolü seçimidir. Henüz otonom sistem tanımlaması yapılmadığından dolayı tüm topoloji üzerinde yönlendirme protokolü olarak RIPng aktif edilmiştir. Bu adımlar Şekil 3.5. ve Şekil 3.6.'da gösterilmiştir.



Şekil 3.5. IPv6 desteği olan yönlendirme protokolünün ayarlanması



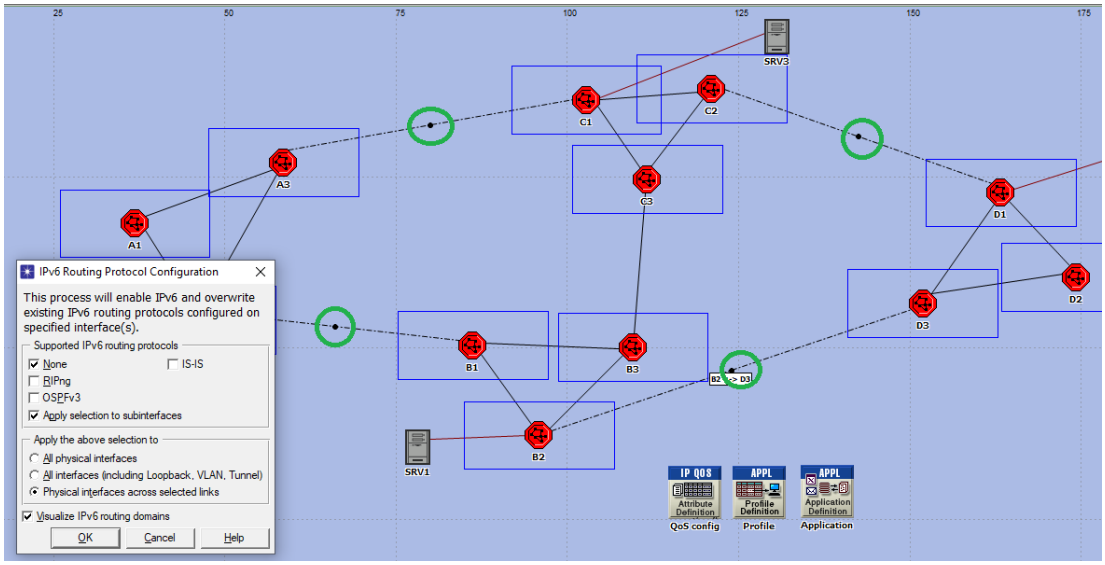
Şekil 3.6. Yönlendirme protokolü olarak RIPng protokolünün seçilmesi

3.3. Otonom Sistemlerin Tanımlanması

Şu anda çalışmakta olan sistem üzerinde herhangi bir otonom sistem tanımlaması yapılmamıştır. Bundan dolayı tüm network üzerinde RIPng protokolü çalışmaktadır. Sıradaki işlem otonom sistemlerin tanımlanması ve otonom sistemler arasında BGP protokolünün tanımlanması olacaktır. İlk adım otonom sistemlerin belirlenmesi olacaktır.

- A1, A2 ve A3 alt ağlarının birleşiminden AS1 oluşturulacaktır.
- B1, B2 ve B3 alt ağlarının birleşiminden AS2 oluşturulacaktır.
- C1, C2 ve C3 alt ağlarının birleşiminden AS3 oluşturulacaktır.
- D1, D2 ve D3 alt ağlarının birleşiminden AS4 oluşturulacaktır.

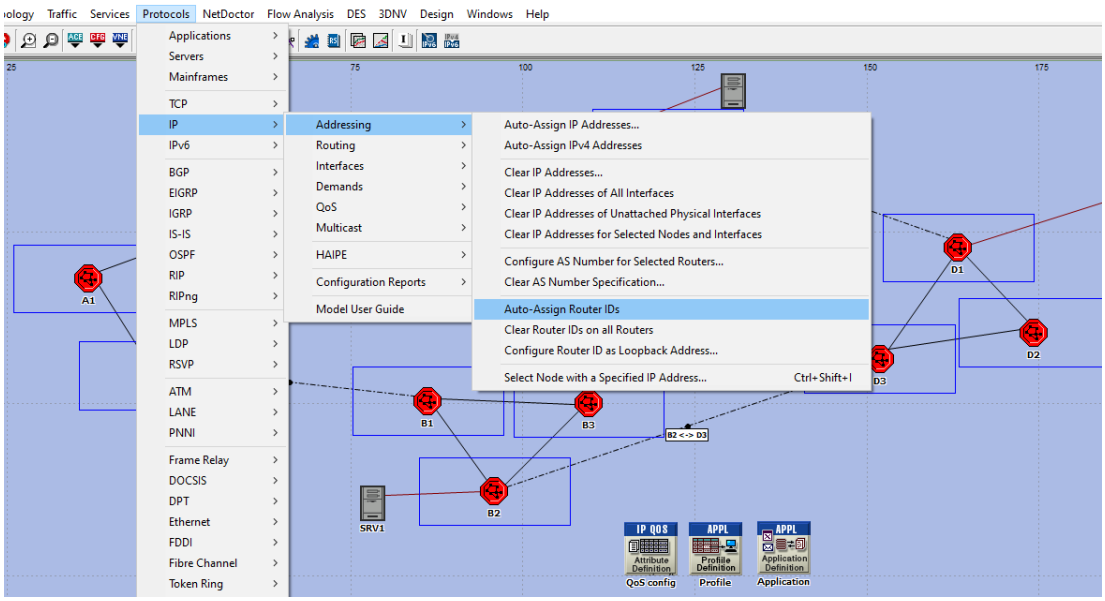
Otonom sistemlerin belirlenmesinden sonra, farklı otonom sistemleri birbirine bağlayan hatlar üzerinde daha önceden tanımlanmış olan RIPng protokolü, otonom sistemler arası çalışan BGP protokolünün uygulanabilmesi için kaldırılır. Bunun için A2 ile B1, A3 ile C1, C2 ile D1 ve D3 ile B2 arasına bulunan hatlar seçilir ve Şekil 3.7.'de gösterildiği gibi yönlendirme protokolü kaldırılır.



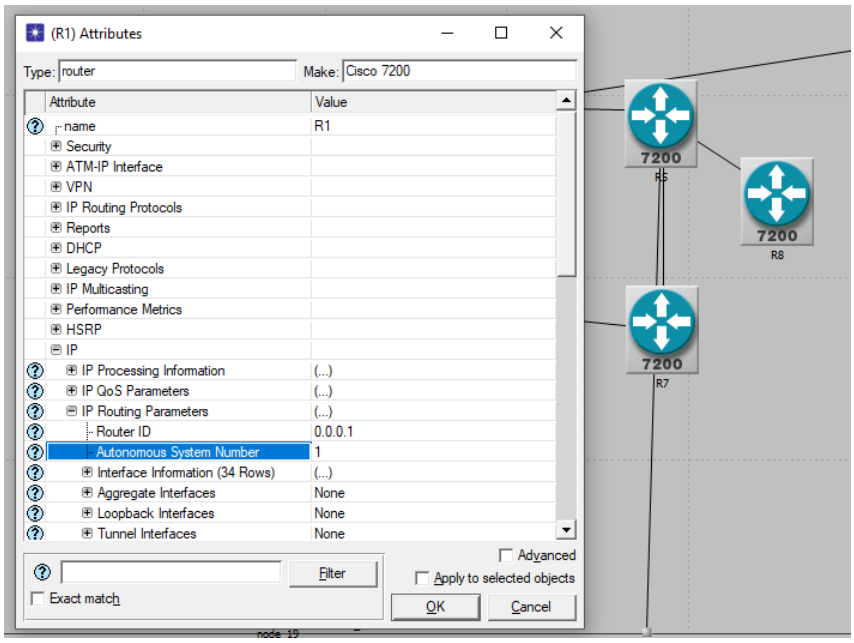
Şekil 3.7. Otonom sistemler arasındaki hatlar üzerinden RIPng protokolünün kaldırılması

Bu noktada dikkat edilmesi gereken husus, yapmış olduğumuz ayarlamaların sadece seçilmiş olan hatlar üzerinde uygulanması için “Physical interfaces across selected links” seçeneğinin işaretlenmiş olması gerekmektedir.

Sırada yönlendiricilere ID atanması işlemi vardır. OPNET programı üzerinde bu işlem otomatik olarak yapılabilmektedir. Bu işlemin yapılması Şekil 3.8.’de gösterilmiştir.



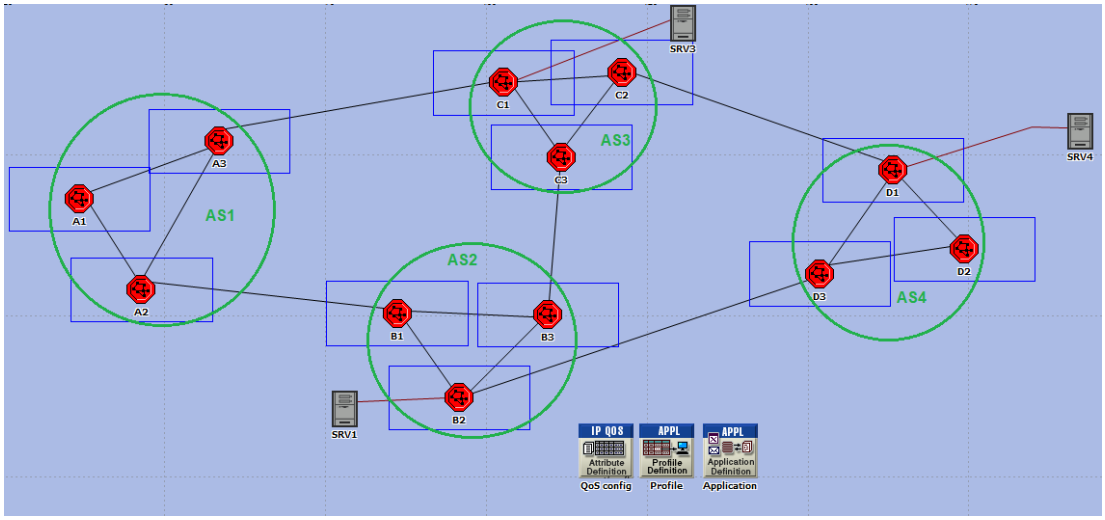
Şekil 3.8. Yönlendirici ID’lerinin otomatik olarak tanımlanması



Şekil 3.9. Yönlendiricilere otonom sistem atamasının yapılması

Yönlendirici ID'lerinin atanmasından sonra, her bir yönlendiricinin içinde bulunduğu otonom sistemin tanımlaması yapılmalıdır. Bunun için atama yapılacak yönlendirici seçilerek sağ tuş ile görüntülenen menü üzerinden "Edit Attributes" seçildiğinde açılan "properties" penceresi üzerinden gerçekleştirilir. Bu işlem Şekil 3.9.'da gösterilmiştir.

Bu işlemler sonucunda oluşan otonom sistemlerin eklenmiş olduğu topoloji yapımız Şekil 3.10.'de verilmiştir.

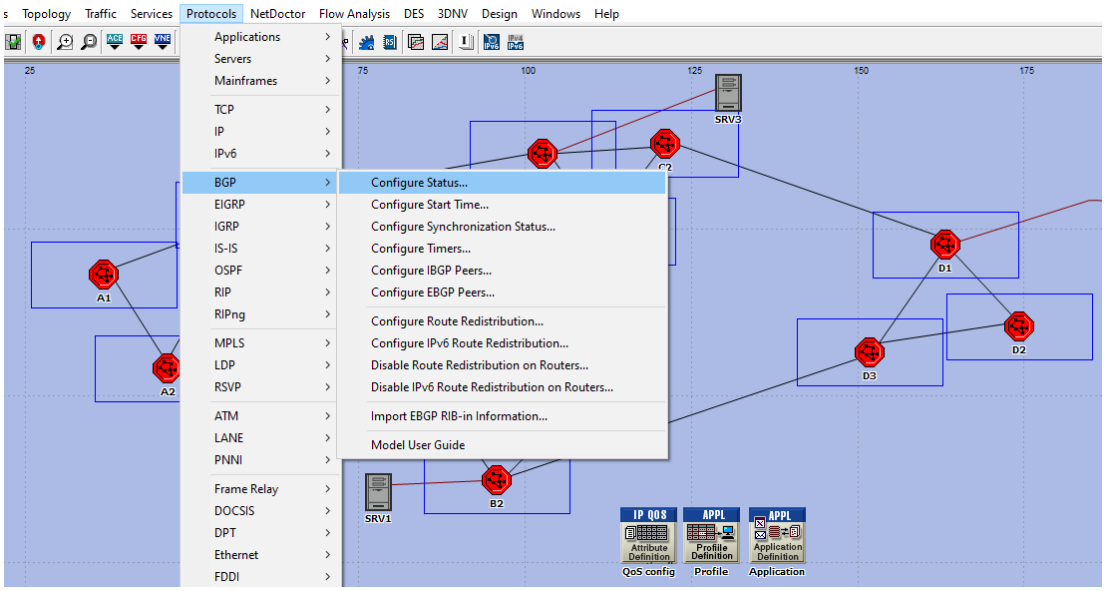


Şekil 3.10. Otonom sistem ayarlamalarından sonra oluşan topolojik yapı

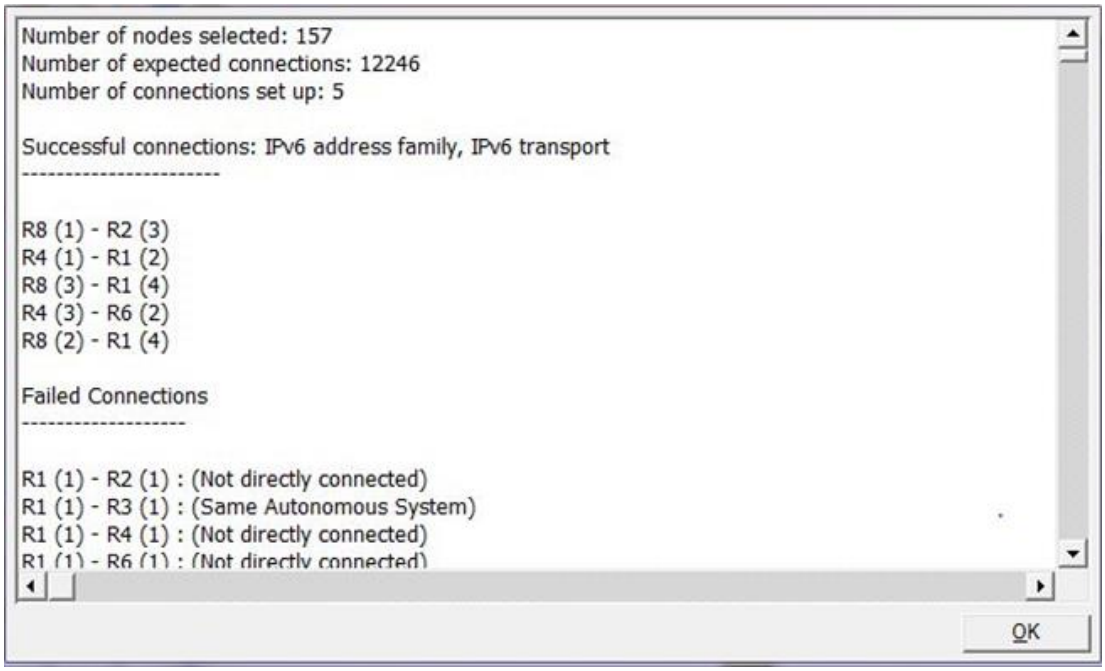
3.4. Otonom Sistemler Arasında BGP Protokolünün Tanımlanması

Otonom sistemler hazırlandıktan sonra, otonom sistemleri birbirine bağlayan hatlar üzerinde BGP protokolü tanımlanabilir durumdadır. Bu tanımlamayı yapmak için "Protocols->BGP->Configure Status" adımları takip edilir. Bu adım Şekil 3.11.'de gösterilmiştir.

BGP protokolünün aktif edilmesinden sonra, her yönlendirici için komşularının ayarlanması gerekmektedir. Bunun için "Protocols->BGP->Configure EBGP peers" yolu takip edilir. OPNET programı bu işlemden sonra atanmış olan AS numaralarına ve BGP protokolünün aktif edildiği yönlendiricilere dikkat ederek komşu ayarlamalarını gerçekleştirir. Şekil 3.12. topolojik ağ üzerinde OPNET tarafından oluşturulan BGP haritalama (mapping) sonucunu göstermektedir.



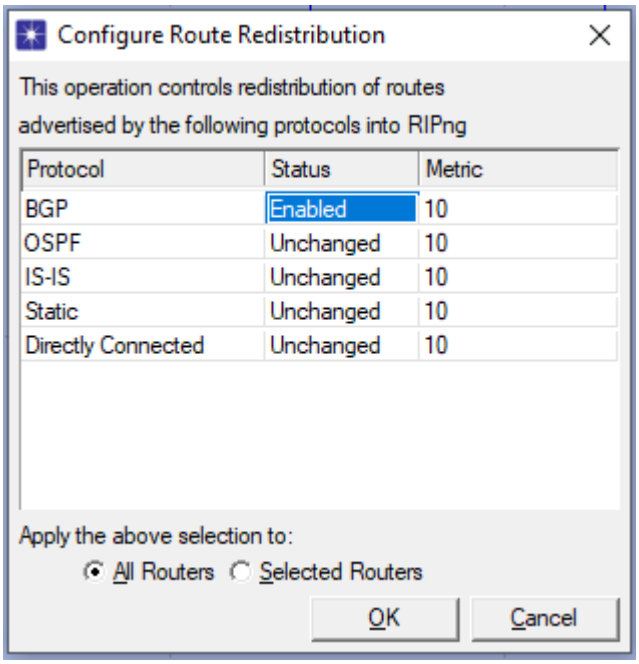
Şekil 3.11. Otonom sistemler arasında BGP protokolünün ayarlanması



Şekil 3.12. BGP haritalama (mapping) sonuçları

BGP protokolünün komşu ilişkileri başarılı bir şekilde kurulduktan sonra RIPng tarafından kaydedilen BGP route bilgilerinin dağıtımının (redistribute) yapılabilmesi için gerekli ayarlamalar yapılmalıdır. Bunun için "Protocols->RIPng->Configure Route Redistribution" yolu takip ederek Şekil 3.13.'te gösterilen ayarlamalara ulaşılır. Bu ayarlama üzerinde BGP rotalarının dağıtımı etkinleştirilir. Bu işlemin amacı, BGP

protokolüne ait yol bilgilerinin RIPng protokolü tarafından kullanılmasının sağlanmasıdır. Çalışmada, sınır yönlendirici üzerinde beraber çalışmakta olan BGP protokolünden RIPng protokolüne duyurulmuş olan rotalar için metrik değeri 10 olarak ayarlanmıştır. Otonom sistem sınırında bulunan yönlendirici tarafından otonom sistem içerisinde RIPng protokolü çalıştıran bir yönlendiriciye aktarılacak olan BGP tarafından duyurulan bir rota olursa eğer bu rota için metrik değeri 10 olacaktır.

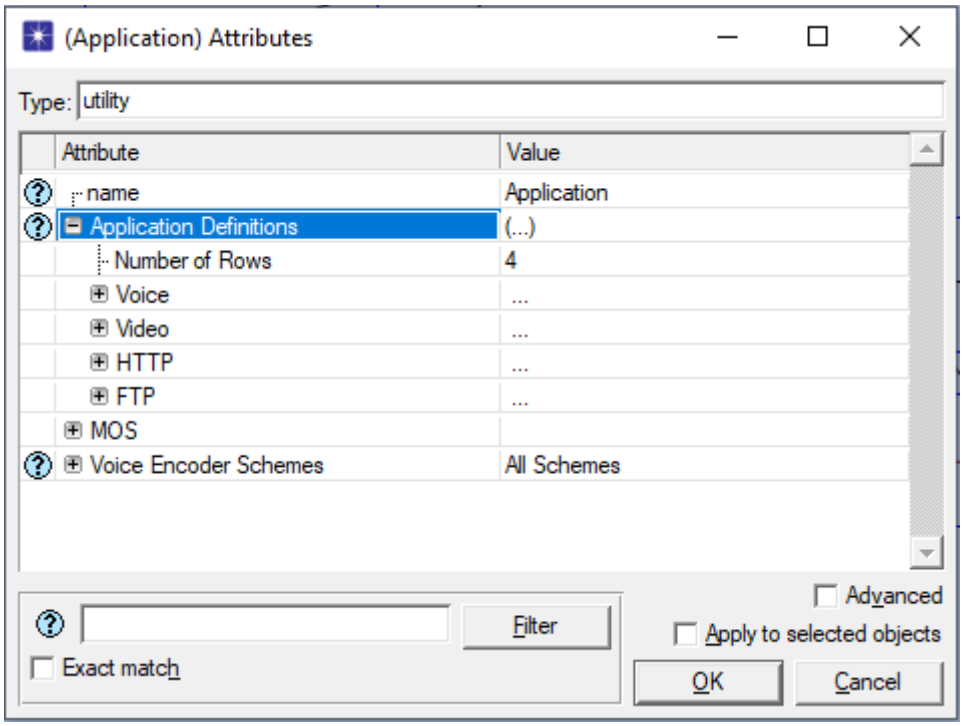


Şekil 3.13. BGP yol bilgilerinin dağıtımının ayarlanması

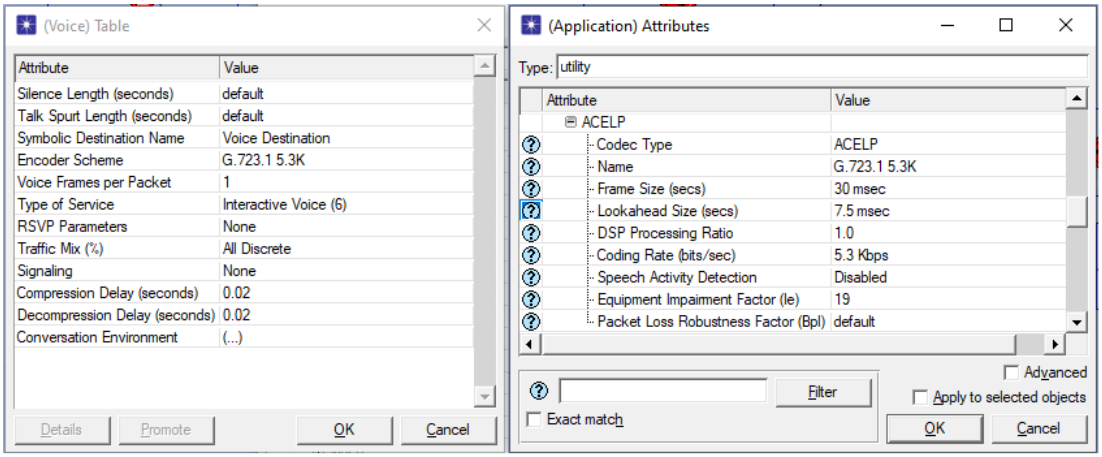
3.5. Çalışan Uygulamaların Ayarlanması

Ağ ayarları tamamlanmıştır. Sırada kullanıcılar üzerinde çalışacak uygulamaların ayarları yapılmalıdır. Bunun için Nesne Paleti (Object Palette) üzerinden bir tane Uygulama Düğümü (Aplication Node) ve bir tane Profil Düğümü (Profile Node) seçmelmelidir. Uygulama Düğümü ağ yapımız üzerinde çalışacak uygulamaları oluşturacağımız OPNET nesnesidir. Profil Düğümü ise çalışmakta olan uygulamaların davranışlarını düzenleyen OPNET nesnesi olmaktadır.

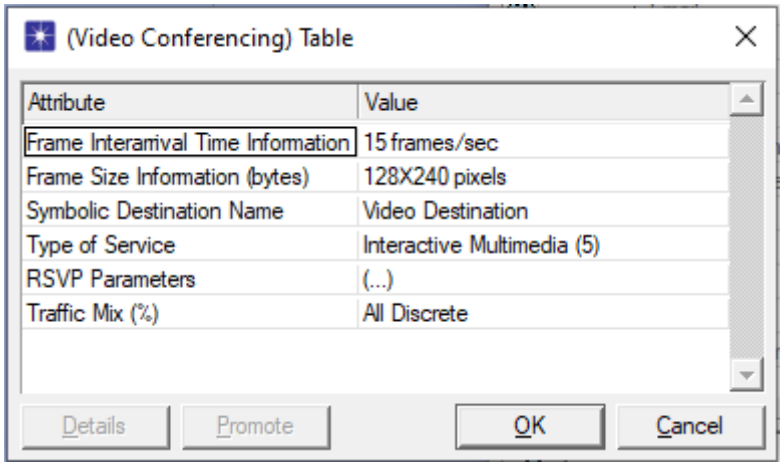
Sıradaki işlemde, uygulama düğümü üzerinde projemizde kullanılacak ses, video ve veri uygulamalarını yerleştirilmesi gerekmektedir. Bu işlem Şekil 3.14.'te görüldüğü gibi düzenlenmiştir. Şekil 3.15. ve Şekil 3.16. uygulamalara ait parametre seçimlerini göstermektedir.



Şekil 3.14. Uygulama düğümü üzerinde uygulamaların ayarlanması



Şekil 3.15. Ses uygulamasına ait parametrelerin seçimi



Şekil 3.16. Video uygulamasına ait parametrelerin seçimi

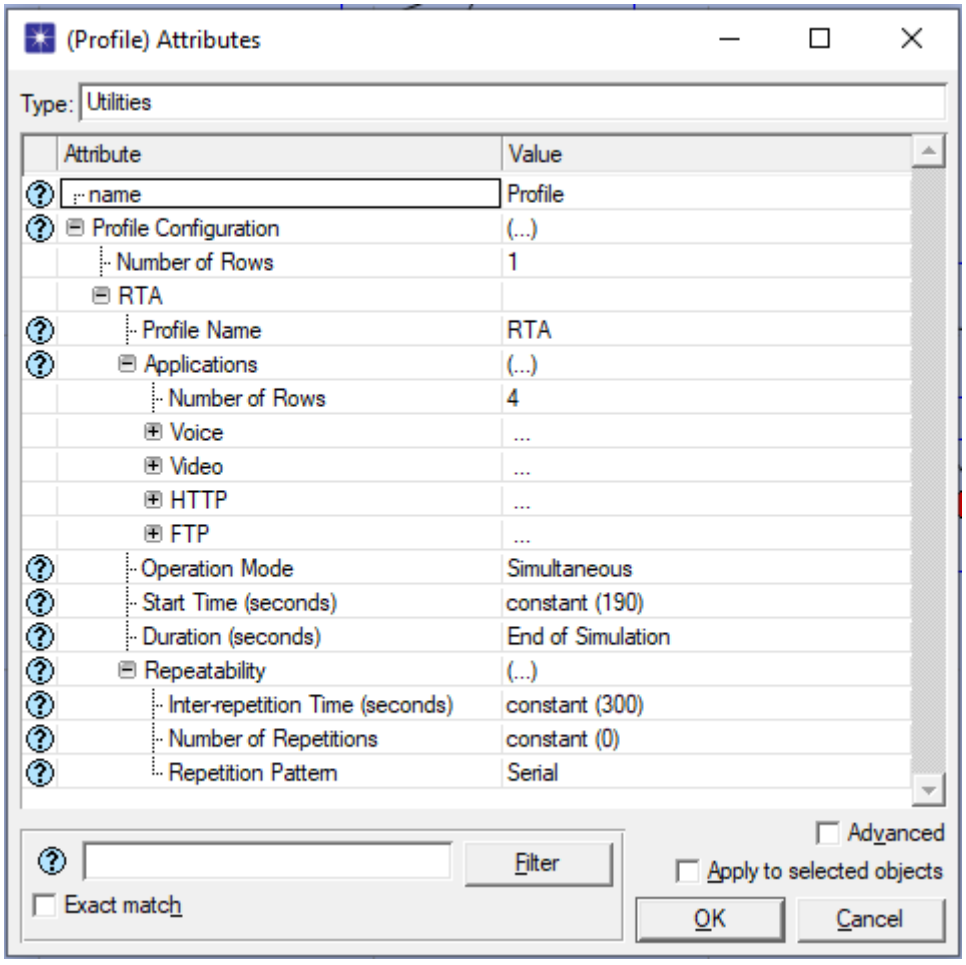
Uygulamalar için seçilmiş olan parametre değerleri literatürde kullanılmış değerlerdir ve bu çalışmada literatürde kullanılan değerlere bağlı kalınmıştır [94]. Tüm uygulamalar için geçerli olan değerler Tablo 3.1.'de verilmiştir.

Tablo 3.1. Uygulamalara ait temel parametreler

Tanımlama	Değer	Birim
Ortak tanımlar		
Çalışma modu	Eş zamanlı	
Başlangıç zamanı	Sabit (190)	Saniye
Süre	Benzetim sonu	
Dahili tekrar tamamlama zamanı	Sabit (300)	Saniye
Video uygulamasına ait değerler		
Video çerçeve oranı	15	Fps
Video çerçeve boyutu	128x240	Piksel
Multimedia tipi	Yüksek Çözünürlüklü Video	
Ses uygulamasına ait değerler		
Çözücü tipi	ACELP	
Kodlayıcı şeması	G723.1 5.3K	
Paket başına ses çerçevesi	1	
Ses çerçeve boyutu	0.030	Saniye
Çözücü oranı	5.3	Kbps
Sıkıştırma gecikmesi	0.02	Saniye
Açma gecikmesi	0.02	Saniye
FTP uygulamasına ait değerler		
FTP dosya boyutu	Sabit (1000)	Bayt
Dahili istek zamanı	Üstel (3600)	
HTTP uygulamasına ait değerler		
HTTP özellikleri	HTTP 1.1	
Nesne boyutu	Sabit (1000)	Bayt
Sayfa varış zamanı	Üstel (60)	Saniye

Uygulamaları tanımladıktan sonra, bir grup uygulama ve onların davranışlarını belirleyen bir profil oluşturulmalıdır. Profil Konfigürasyon Düğüm (Profile Configuration Node) özelliklerini düzenleyerek bir profil eklenir ve bu profile Şekil 3.17.'de gösterildiği gibi uygulamaların atamasını yapılır. Çalışma Modu (Operation Mode) üzerinde ise uygulamaların nasıl başlayacağı tanımlanır. Burada bulunan seçenekler:

- Seri-Sıralı (Serial-Ordered): Düzgün sırada birbiri ardına başlar (ilk satırdan son satıra doğru)
- Seri-Rastgele (Serial-Random): Rasgele bir sırada birbiri ardına başlar.
- Eşzamanlı (Simultaneous): Hepsi aynı anda başlar.

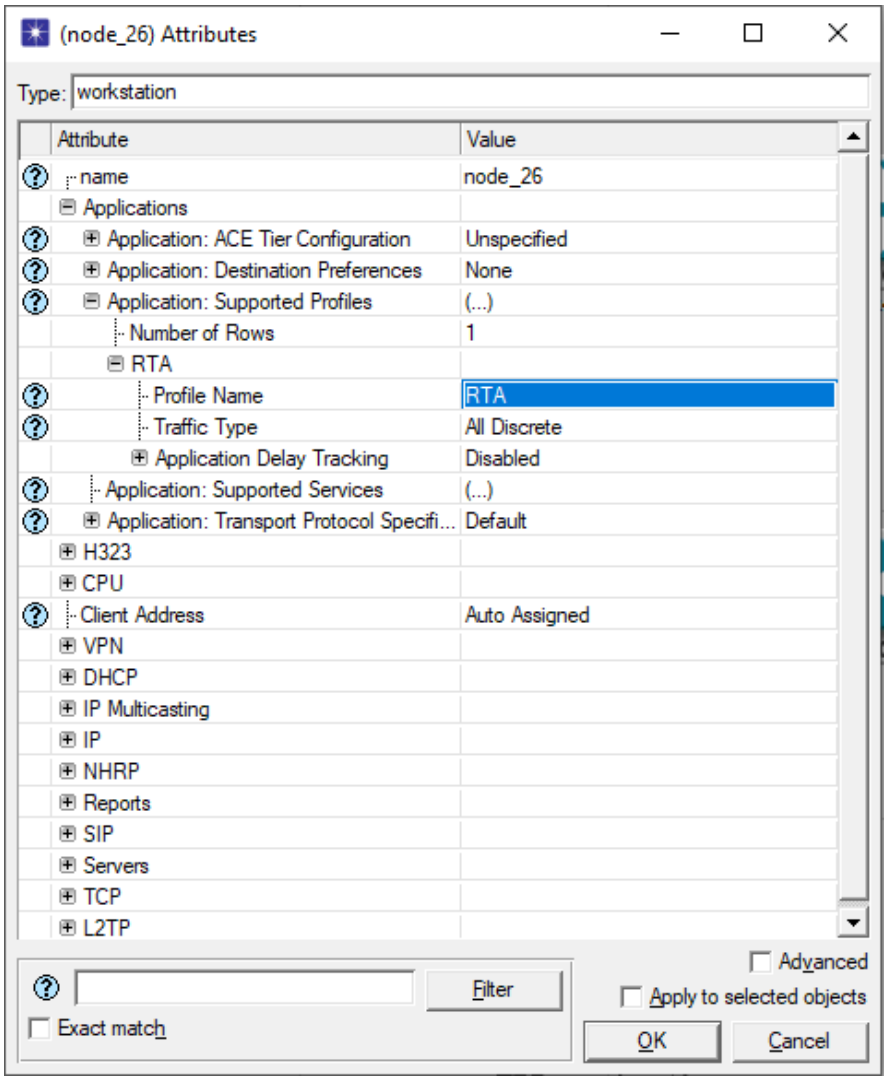


Şekil 3.17. Uygulamalara ait profil yapılandırması

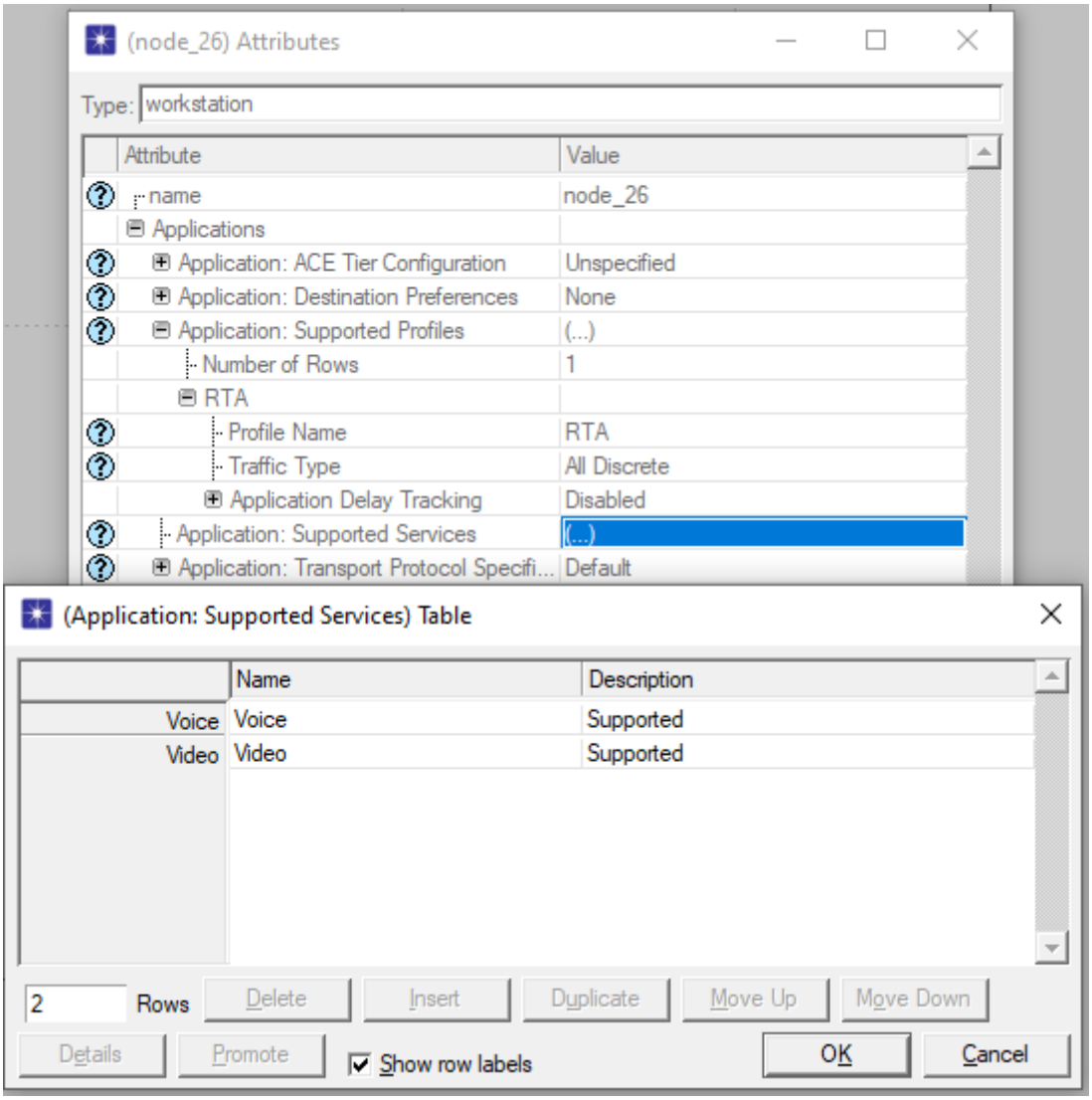
3.6. Kaynak ve Hedeflerin Belirlenmesi

Bir düğümü kaynak olarak ayarlayabilmek için özelliklerini düzenleyerek ona bir profil atanması gerekmektedir. “Application-> Application: Supported profiles” yolu takip edilerek daha önceden tanımlanan profil bu düğüme eklenir. Bu durum Şekil 3.18.’de gösterilmiştir. Bu uygulamada tüm düğümler kaynak olarak belirlenmiştir.

Bir düğümün hedef olarak konumlandırmak için yapılması gerekenler Şekil 3.19.’da gösterilmiştir. HTTP ve FTP trafiği için tüm sunucular hedef pozisyonunda olacaklardır. Bunun yanında ses ve video trafiğinde uçtan-uca etkileşimli haberleşme için tüm düğümler kaynak olduğu gibi aynı zamanda hedef olarak belirlenmişlerdir.



Şekil 3.18. Bir kullanıcıya profil eklenmesi

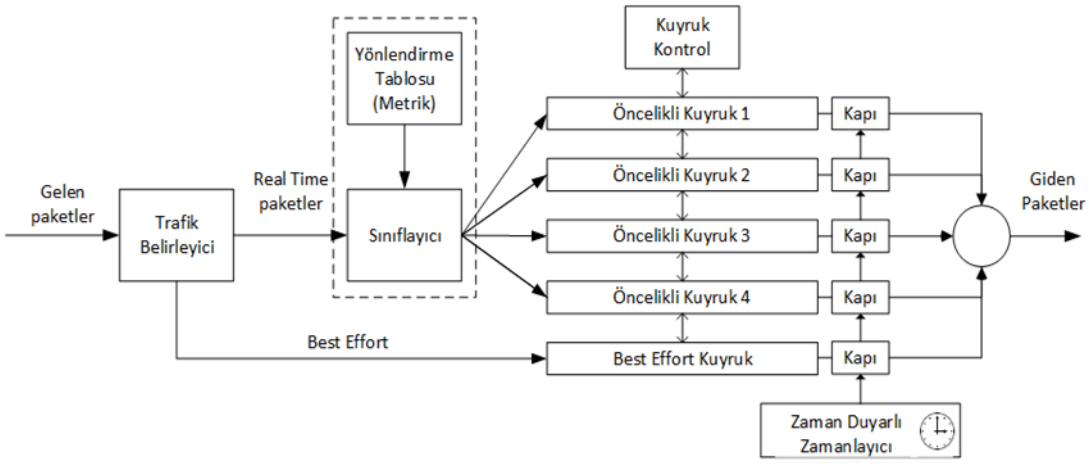


Şekil 3.19. Bir kullanıcının hedef olarak tanımlanması

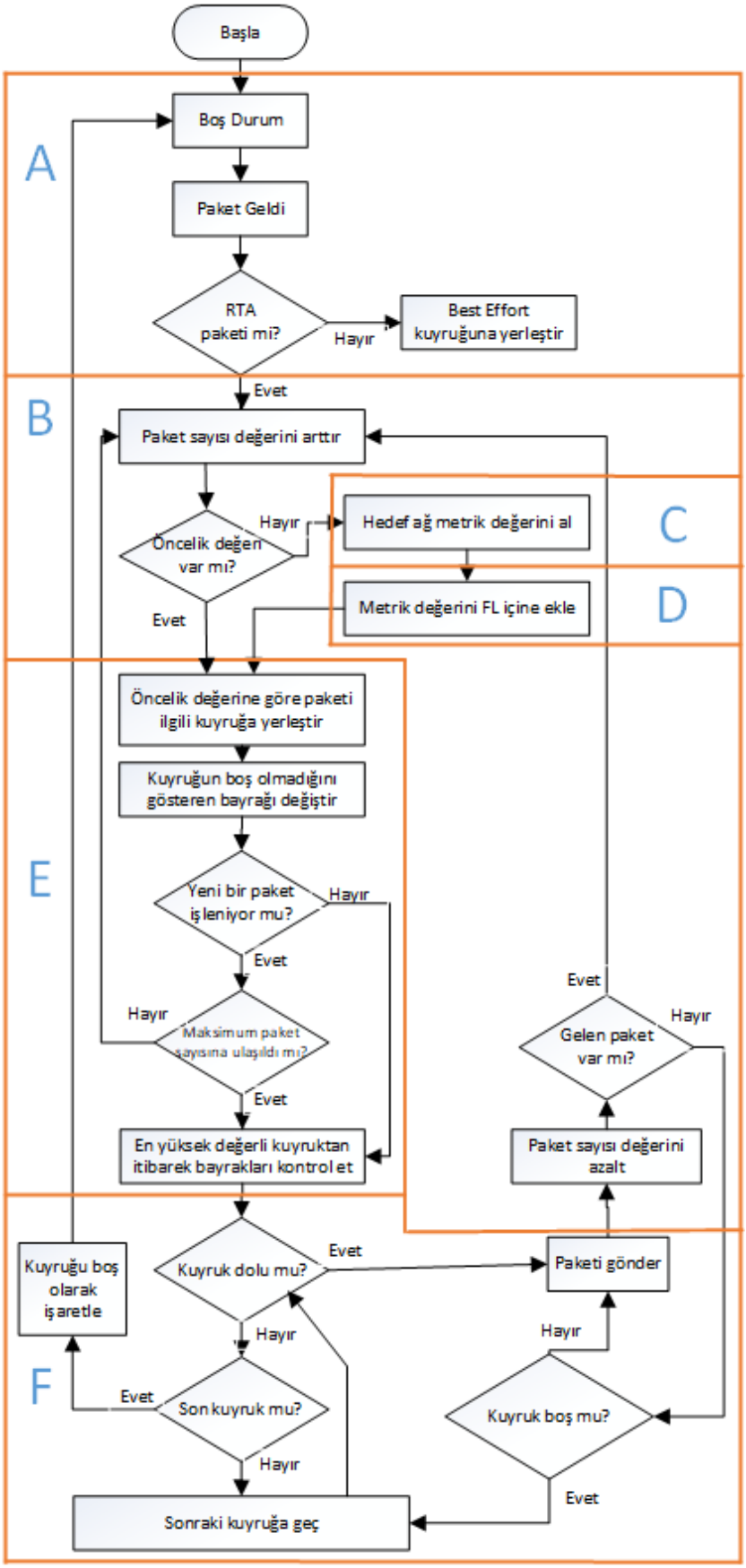
Bu tanımlamadan sonra artık ağ yapımız üzerinde tüm veri trafiği oluşmuş durumdadır. Artık benzetim çalıştırıp referans olarak kullanılacak istatistiki bilgiler toplanabilir durumdadır. Sonuçlar rapor sonunda karşılaştırmalı olarak verilecektir. Bu noktada ilk adım tanımlanmıştır. Burada oluşturulmuş olan referans topoloji üzerinden alınan istatistik sonuçları uygulanacak olan önceliklendirme algoritmasının başarımını ölçmek için kullanılacaktır.

BÖLÜM 4. HEDEF AĞ PARAMETRELERİNE (METRİK DEĞERİ) GÖRE ÖNCELİLENDİRME YAPAN ALGORİTMA MODELİNİN OLUŞTURULMASI

GZU uygulamalarının ihtiyaç duyduğu önceliklendirme yaklaşım ve çözümleri Bölüm 2.1.2'de verilmiştir. Bu çalışmada literatürde örneği bulunmayan bir yöntem ile GZU uygulamalarının önceliklendirmesi yapılacaktır. Önceliklendirme parametresi olarak GZU uygulama paketinin hedef ağ metrik değeri kullanılacak ve bu değer yönlendirme tablosundan alınarak uçtan-uca destek için IPv6 paket bağlığında bulunan FL alanı içine yazılarak taşınacaktır. Modelde literatürde en iyi sonuçları veren PQ kuyruk modeli kullanılmış olup önceliklendirilen paketler ilgili kuyruğa alınacak ve kuyruk önceliğine göre iletim yapılacaktır. Önerilen çözüm modelinde her bir akış için ayrı durum bilgisi tutulmayacak DiffServ modeli referans alınarak her bir akış önceden tanımlanmış olan ilgili kuyruğa yerleştirilecektir. Modele ait blok diyagram Şekil 4.1.'de algoritmaya ait akış şeması da Şekil 4.2.'de verilmiştir.



Şekil 4.1. Uygulanacak algoritmaya ait blok diyagram



Şekil 4.2. Uygulanacak algoritmaya ait akış şeması

Blok diyagramdan da anlaşılacağı üzere, sisteme gelen paketler ilk önce Trafik Belirleyici tarafından DSCP bitlerine bakılarak GZU veya best erfort paketi olup olmadıkları denetlenecek, best effort paketleri direk olarak ilgili kuyruğa gönderilirken GZU paketleri Sınıflayıcı'ya gönderilecektir. Sınıflayıcı, ilk önce kendisine gelen paketin FL alanını kontrol ederek kendine gelmeden önce bir başka sistem tarafından önceliklendirme değeri eklenip eklenmediğine bakacaktır. Eğer bir önceliklendirme değeri var ise direk olarak ilgili kuyruğa gönderecektir. Eğer önceliklendirme değeri yok ise ilgili paketin hedef ağını tespit ettikten sonra bu ağa ait metrik değeri yönlendirme tablosundan alarak FL içerisine yazıp, önceliklendirme değeri olarak kullanılan metrik değere göre ilgili önceliklendirme kuyruğuna gönderecektir. Kuyruk Kontrol, kuyrukların dolu, boş durumlarını ve önceliğe göre gönderimlerini denetlemektedir. Herhangi bir kuyruğun devamlı gönderimde kalıp diğer kuyrukları engelleme durumunun önüne geçmek amacı ile Round Robin tabanlı bir Zamanlayıcı ile kuyruk kapıları kontrol edilerek her bir kuyruğa adil bir davranış sergilenecektir. Böylece metrik değeri yüksek olan ağa gidecek GZU paketleri öncelikli olarak gönderilecek, düşük metrik değerli ağa gidecek GZU paketleri ise sonraya bırakılacaktır. Bu şekilde, GZU paketleri tek bir davranış modeli içinde değil hedef ağ metrik değerine göre kendi içerisinde bir önceliklendirmeye tabi olacaklardır.

4.1. Önceliklendirme Algoritmasının Tasarımı ve Uygulanması

Birinci adımda önceliklendirme modelinin başarımını ölçmek için kullanılacak referans topoloji tasarlanmıştır. İkinci adım da ise, bu referans topoloji üzerinde uygulanacak önceliklendirme algoritmasının tasarımı ve uygulaması yapılacaktır.

4.1.1. Gerçek zamanlı paketlerin ayırt edilmesi

Algoritma şeması üzerinde "A" ile gösterilen alandır. Eğer sisteme bir paket gelirse, ilk amaç önceliklendirme uygulanacak ses ve video paketlerin ayırt edilmesi olmalıdır. Bunun için IPv4 paketinde bulunan ToS alanına karşılık gelen IPv6 paketindeki TC alanı kullanılır.

QoS Değerleri

CoS = Servis Sınıfı
DSCP = Farklılaştırılmış Hizmetler Kod Noktası
ToS = Servis Tipi
AF = Garantili İletim
IPP = IP Öncelik
CS = Sınıf Seçici
DP = Düşürme Olasılığı
ECN = Hata Sıkışıklık Bildirimi

ToS / TC							
DSCP							ECN
AF (CS;DP)							
IPP=CS			DP				
			Gecikme	Verim	Güvenilir		

	8. bit	7. bit	6. bit	5. bit	4. bit	3. bit	2. bit	1. bit
ToS	128	64	32	16	8	4	2	1
DSCP	32	16	8	4	2	1		
CoS=IPP	4	2	1					

Uygulama	CoS=IPP	AF	DSCP	ToS	ToS HEX	DP	8. bit	7. bit	6. bit	5. bit	4. bit	3. bit	2. bit	1. bit
Best Effort	0	0	0	0	0		0	0	0	0	0	0	0	0
Önemsiz Veri	1	CS1	8	32	20		0	0	1	0	0	0	0	0
Toplu Veri	1	AF11	10	40	28	Düşük	0	0	1	0	1	0	0	0
	1	AF12	12	48	30	Orta	0	0	1	1	0	0	0	0
	1	AF13	14	56	38	Yüksek	0	0	1	1	1	0	0	0
Ağ Yönetim	2	CS2	16	64	40		0	1	0	0	0	0	0	0
İşlem Verisi	2	AF21	18	72	48	Düşük	0	1	0	0	1	0	0	0
	2	AF22	20	80	50	Orta	0	1	0	1	0	0	0	0
	2	AF23	22	88	58	Yüksek	0	1	0	1	1	0	0	0
Çağrı Sinyali	3	CS3	24	96	60		0	1	1	0	0	0	0	0
Kritik Görev	3	AF31	26	104	68	Düşük	0	1	1	0	1	0	0	0
Akıcı Video	3	AF32	28	112	70	Orta	0	1	1	1	0	0	0	0
	3	AF33	30	120	78	Yüksek	0	1	1	1	1	0	0	0
	4	CS4	32	128	80		1	0	0	0	0	0	0	0
Etkileşimli Video	4	AF41	34	136	88	Düşük	1	0	0	0	1	0	0	0
	4	AF42	36	144	90	Orta	1	0	0	1	0	0	0	0
	4	AF43	38	152	98	Yüksek	1	0	0	1	1	0	0	0
	5	CS5	40	160	A0		1	0	1	0	0	0	0	0
Ses	5	EF	46	184	B8		1	0	1	1	1	0	0	0
Yönlendirme	6	CS6	48	192	C0		1	1	0	0	0	0	0	0
Ağ Kontrol	7	CS7	56	224	E0		1	1	1	0	0	0	0	0

Şekil 4.3. ToS/TC alanındaki bitler ve anlamları

Ses paketlerinin ToS/TC alanı içerisindeki IP öncelik değeri 5, etkileşimli video paketlerinin ise 4 olarak tanımlıdır (Benzetimde kullanılan OPNET programı içerisinde numaralandırma 0'dan değil de 1'den başladığı için ses paketlerinin değeri 6, etkileşimli video paketlerinin 5 olarak tanımlıdır). Dolayısıyla IPv6 başlığı içerisinde TC alanında bulunan IP öncelik bitlerinin değeri 4 ve 5 ise bu paketler gerçek zaman uygulamalarına ait paketlerdir. Bu paketler blok diyagramda gösterildiği gibi Trafik Belirleyici tarafından seçilerek uygun önceliklendirmenin yapılabilmesi için Sınıflayıcı'ya gönderilecektir. Bu adımda best effort paketler ise algoritmaya girmeden direk olarak best effort kuyruğuna gönderilir.

4.1.2. Ses ve video paketlerinin denetim altına alınması

Algoritma şeması üzerinde "B" ile gösterilen alandır. Geliştirilen algoritma içerisinde, gelen her bir ses ve video paketini sayan bir değişken tanımlanmıştır. Bu değişken bir durum değişkenidir ancak bütün fonksiyonların erişim yapabilmeleri için global değişken olarak tanımlanmış olup sisteme giriş yapan ses ve video paketlerinde arttırılmış ve sistemden çıkış yapan her bir pakette azaltılarak, o an için sistem üzerinde ne kadar GZU paketinin olduğu denetim altına alınmıştır. Bu adımda, gelen GZU paket üzerinde eğer herhangi bir önceliklendirme değeri yok ise bu değerin eklenmesi için "C" adımına geçilir. Eğer paket daha önceden bir önceliklendime değeri aldıysa ve bu değere sahipse algoritma üzerinde "C" ve "D" adımları atlanarak direk olarak "E" adımına geçilir. Bu tanımlama "ip_output_iface.pr.c" dosyası üzerinde gerçekleştirilmiştir.

4.1.3. Metrik değerlerinin elde edilmesi

Algoritma şeması üzerinde "C" ile gösterilen alandır. En kritik problemlerden biri yönlendirme tablosuna erişim ve bu tablo içerisinde bulunan metrik değerinin elde edilmesidir. Bu nokta en önemli ve en uzun adımı içermektedir.

OPNET süreç tabanlı bir benzetim programıdır ve bu süreçler birbirleriyle mesajlaşma yoluyla haberleşmektedir. Bu iletişim, bu iki süreci birbirine bağlayan bağlantılar

üzerinden gönderilen mesajlar ile gerçekleşir. Bu mesajlar Arayüz Kontrol Bilgisi (ICI) mesajları olarak adlandırılır. ICI mesajlarının yapısı <anahtar, değer> şeklinde bir dizi tanımlaması gibidir.

Burada yapılması gereken, kapsülleme yapısını işleyen süreçten yönlendirme tablosunu oluşturan sürece doğru olan mesajlara erişim yapmaktır. Bu iki süreç "ip_dipatch" ve "ip_encap" isimleri ile anılan süreçlerdir. Bu erişimi sağlamak için bir ICI yapısı oluşturulur. Oluşturulan ICI mesaj yapısında aşağıdaki alanlar bulunmaktadır.

- Mesaj tipi: İstek için 0, cevap için 1 tanımlanmıştır.
- Adres: Metrik değerini istediğimiz adresin tanımlaması
- Metrik: Ön tanımlı değer 0 tanımlanmıştır.

ip_dipatch sürecinin hedef adrese ait metrik değerini sorabilmesi için oluşturulan ICI mesaj yapısı "ip_encap_v4.pr.c" dosyası içerisinde aşağıdaki fonksiyonda gösterildiği şekilde tanımlanmıştır.

```
ip_iciptr = op_ici_create ("metric_req_ind");
op_ici_attr_set (ip_iciptr, "metric_req" , 1);
op_ici_attr_set (ip_iciptr, "dest_addr" , dest_addr);
op_ici_install (ip_iciptr);
```

Artık sistem gerçek zamanlı uygulamalara ait bir paket aldığında, paket içerisindeki hedef adrese ait metrik değerini öğrenmek için "ip_dispatch" sürecine bir ICI mesajı gönderilir ve istenen metrik değerinin geri döndürülmesi için bir kesme yaratılır. Artık metrik değerinin istenmesiyle ilgili süreç tamamlanmıştır. Sonraki adım, "ip_dispatch" sürece gelen ICI mesajı ile bildirilen hedef adrese ait metrik değerinin elde edilmesidir. "ip_encap" süreçten bir ICI mesajı alındığında ip_proses, mesaj içerisindeki hedef adrese ait metrik değerini yönlendirme tablosundan getirecek ve istenilen metrik değerini içeren bir cevap ICI mesajı oluşturacaktır.

Bu işleme ait fonksiyon aşağıdaki gibidir.

```
ip_dispatch_assign_metric (void)
        {
        Ici*                        iciptr;
        InetT_Address*              addr_ptr;
        InetT_Address               dest_addr;
        int                         i_th_entry;
        int                         num_entries;
        IpT_Cmn_Rte_Table_Entry*    route_entry;
        char                        dest_addr_str [INETC_ADDR_STR_LEN];
        char                        dest_prefix_str [INETC_ADDR_STR_LEN];
        int*                        temp;
        char                        ch_arr[128];

FIN (ip_dispatch_assign_metric ());

printf("We got in the metric assignment func\n");
iciptr = op_intrpt_ici ();
printf("Took the ICI\n");
if ( iciptr == OPC_NIL )
        printf("iciptr NIL !\n");
else
        printf("iciptr not NIL!\n");


if (op_ici_attr_exists (iciptr, "dest_addr"))
        printf("dest_addr exists\n");

op_ici_format (iciptr, ch_arr);
        printf("ici name %s\n",ch_arr);

if (op_ici_attr_exists (iciptr, "metric"))
        printf("metric attribute exists\n");

if (OPC_COMPCODE_FAILURE == op_ici_attr_get (iciptr, "metric", &temp))
        printf("Did not passed op_ici_attr_get for metric\n");
else
        printf("passed op_ici_attr_get for metric %d\n",*temp);
```

```
if (OPC_COMPCODE_FAILURE == op_ici_attr_get_ptr (iciptr, "dest_addr",
&addr_ptr) )
        printf("Did not passed op_ici_attr_get for dest_addr\n");
else
        printf("passed op_ici_attr_get for dest_addr\n");
if ( &addr_ptr == OPC_NIL )
        printf (" NIL &addr_ptr\n");
if ( addr_ptr == OPC_NIL )
        printf (" NIL addr_ptr\n");

dest_addr = *addr_ptr;
dest_addr = inet_address_copy(*addr_ptr);

op_ici_destroy (iciptr);

num_entries = ip_cmn_rte_table_num_entries_get (module_data.ip_route_table,
InetC_Addr_Family_v6);

for (i_th_entry = 0; i_th_entry < num_entries; i_th_entry++)
        {

route_entry = ip_cmn_rte_table_access (module_data.ip_route_table,  i_th_entry,
InetC_Addr_Family_v6);
if (ip_cmn_rte_table_dest_prefix_addr_equal(route_entry->dest_prefix,dest_addr))
        {
        printf ("Nbr of next hops %d",op_prg_list_size (route_entry->next_hop_list));



        break ;
        }
        else
        {
inet_address_print (dest_addr_str, dest_addr);
ip_cmn_rte_table_dest_prefix_print (dest_prefix_str, route_entry->dest_prefix);
        printf ("Our destination address (%s) and the prefix dest address is %s\n",
        dest_addr_str,dest_prefix_str);

        }
        }
        FOUT;
}
```

4.1.4. Metrik değerinin öncelik değeri olarak FL alanına eklenmesi

Algoritma şeması üzerinde "D" ile gösterilen alandır. GZU paketinin gideceği hedef adrese ait metrik değeri elde edildiğine göre, bu değere göre oluşturulmuş önceliklendirme yapısı FL alanına uygulanabilir. FL alanı 20 bit uzunluğundadır ve önerilen yaklaşımın daha evrensel olabilmesi amacı ile geniş bir çerçevede değerlendirilmiştir. Başka çalışmalarda gerekli olabilecek, trafik tipi ve metrik değerini gönderen yönlendirme protokol tipinin belirlenebilmesi amacı ile gerekli tahsisler yapılmıştır. Bunun yanında ileride ortaya çıkabilecek durumlar için rezerve bit de ayrılmıştır. Önerilen FL kullanım yaklaşımı Şekil 4.4.'te verilmiştir.

Şekil 4.4. Önerilen FL alanı kullanım yaklaşımı

Önerilen Akış Etiketi kullanım yaklaşımı tasarlanırken Bnarjee tarafından önerilen DiffServ PHB-ID Yaklaşımı örnek alınmıştır. Bu yaklaşım hem performans olarak en yüksek değeri veren [65] yaklaşımdır, hem de önerilen önceliklendirme algoritması DiffServ yaklaşımına uygundur. Çünkü çalışmada kullanılan RIPng protokolünde en düşük metrik değeri 0 iken en büyük metrik değeri 15'tir. Böylece her bir akışa ait paketler bu değerlerden birini almak zorundadır. Böylece önerilen yaklaşımda RIPng protokolü için doğal olarak maksimum 16 servis sınıfı olacak anlamına gelmektedir. Dolayısıyla yönlendiriciler IntServ mantığında olduğu gibi her bir akış için ayrı bir durum bilgisi tutmak yerine sadece tanımlı olan ve maksimum 16 tane olabilecek durum bilgilerini tutacaktır. Aynı durum kullanılan BGP protokolü için de geçerlidir. BGP protokolü en iyi yol bulmak için metrik değeri olarak otonom sistem sayısını dikkate almaktadır. RIPng protokolünde kullanılan 16 servis sınıfı BGP için de yeterli olacaktır.

Buradaki bitlerin anlamları ise;

0: Önceliklendirme algoritması belirleyicisidir. 0 ise algoritma uygulanmamış, 1 ise uygulanmıştır.

1-3: Çalışmamızda gerçek zamanlı uygulama paketlerini işaretlemek için kullanılmıştır. Ancak toplamda sekiz farklı trafik tipi işaretlenebilecektir.

4-6: Yönlendirme protokol tanımlamasıdır. Toplamda sekiz farklı protokol tanımlanabilir.

7-18: Hedef networke ait metrik değeri tanımlaması.

Bu alanın RIPng ve BGP protokolünde kullanımı aşağıdaki gibidir.

000000000000: Metrik0

000000000001: Metrik1

000000000010: Metrik2

000000000011: Metrik3

.

.

.

000000001111: Metrik15

19: İleride doğabilecek ihtiyaçlar için rezerve edilmiştir.

Bu işlem için kullanılan kod aşağıdaki gibidir.

```
if (op_ici_attr_exists (iciptr, "metric"))
        {
        op_ici_attr_get_ptr (iciptr, "metric", &metric_returned );
        int flow_label_to_send_temp
        switch(metric_returned) {
        case 0 :
        flow_label_to_send_temp = 2 ^ 19 + 2 ^ 18 ;
                break ;
        case 1 :
        flow_label_to_send_temp = 2 ^ 19 + 2 ^ 18 + 2 ^ 1 ;
                break ;
```

```
case 2 :
flow_label_to_send_temp = 2 ^ 19 + 2 ^ 18 + 2 ^ 2 ;
        break ;
case 3 :
flow_label_to_send_temp = 2 ^ 19 + 2 ^ 18 + 2 ^ 2 + 2 ^ 1 + ;
        break ;
case 4 :
flow_label_to_send_temp = 2 ^ 19 + 2 ^ 18 + 2 ^ 3 ;
        break ;
case 5 :
flow_label_to_send_temp = 2 ^ 19 + 2 ^ 18 + 2 ^ 3 + 2 ^ 1 ;
        break ;
case 6 :
flow_label_to_send_temp = 2 ^ 19 + 2 ^ 18 + 2 ^ 3 + 2 ^ 2 ;
        break ;
case 7 :
flow_label_to_send_temp = 2 ^ 19 + 2 ^ 18 + 2 ^ 3 + 2 ^ 2 + 2 ^ 1 ;
        break ;
case 8 :
flow_label_to_send_temp = 2 ^ 19 + 2 ^ 18 + 2 ^ 4 ;
        break ;
case 9 :
flow_label_to_send_temp = 2 ^ 19 + 2 ^ 18 + 2 ^ 4 + 2 ^ 1 ;
        break ;
case 10 :
flow_label_to_send_temp = 2 ^ 19 + 2 ^ 18 + 2 ^ 4 + 2 ^ 2 ;
        break ;
case 11 :
flow_label_to_send_temp = 2 ^ 19 + 2 ^ 18 + 2 ^ 4 + 2 ^ 2 + 2 ^ 1 ;
        break ;
case 12 :
flow_label_to_send_temp = 2 ^ 19 + 2 ^ 18 + 2 ^ 4 + 2 ^ 3 ;
        break ;
case 13 :
flow_label_to_send_temp = 2 ^ 19 + 2 ^ 18 + 2 ^ 4 + 2 ^ 3 + 2 ^ 1 ;
        break ;
case 14 :
flow_label_to_send_temp = 2 ^ 19 + 2 ^ 18 + 2 ^ 4 + 2 ^ 3 + 2 ^ 2 ;
        break ;
case 15 :
flow_label_to_send_temp = 2 ^ 19 + 2 ^ 18 + 2 ^ 4 + 2 ^ 3 + 2 ^ 2 + 2^1 ;
        break ;
```

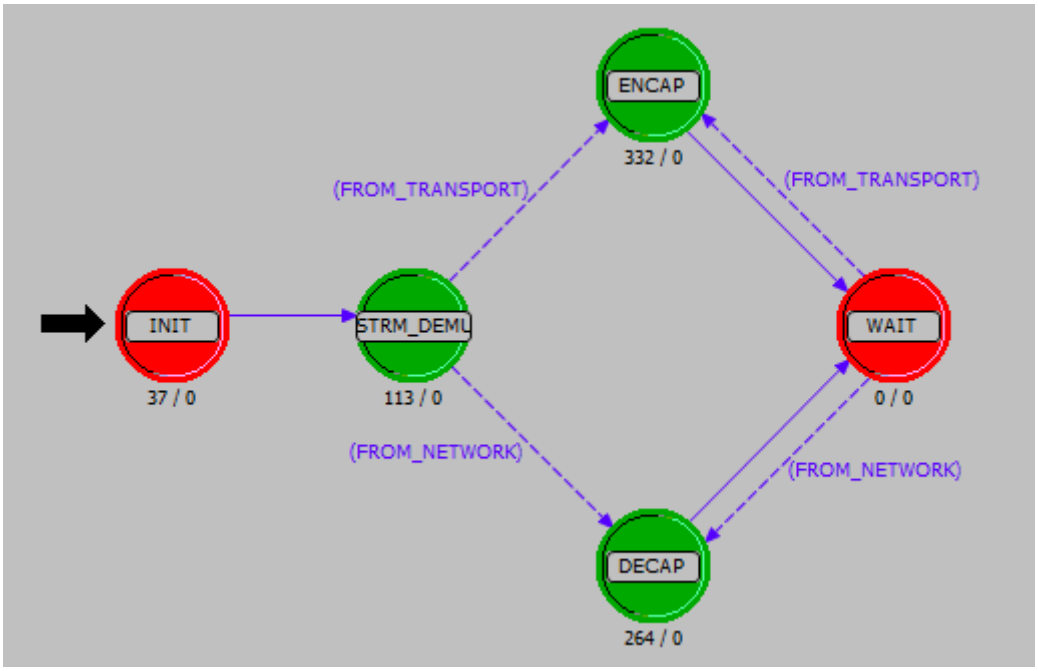
```
                    }
ip_dgram_fd_ptr->flow_label_field = flow_label_to_send_temp;

ip_dgram_fields_set (ip_pkptr, ip_dgram_fd_ptr);

op_pk_bulk_size_set (ip_pkptr, data_len * 8);

op_pk_send (ip_pkptr, outstrm_to_network);
exit (1);
}
```

Algoritmanın bu adımına kadar olan süreçte, "ip_encap proses", ip katmanında bir paket akışı kaydederse (Şekil 4.5.) bu paketler için bir önceliklendirme yapılması gerekiyor mu bunun kararı verilebilir. Eğer gerekiyorsa bu işlemi gerçekleştirir ve diğer paketler yani GZU paketi olmayan paketleri kaldığı yerden işlemeye devam eder.



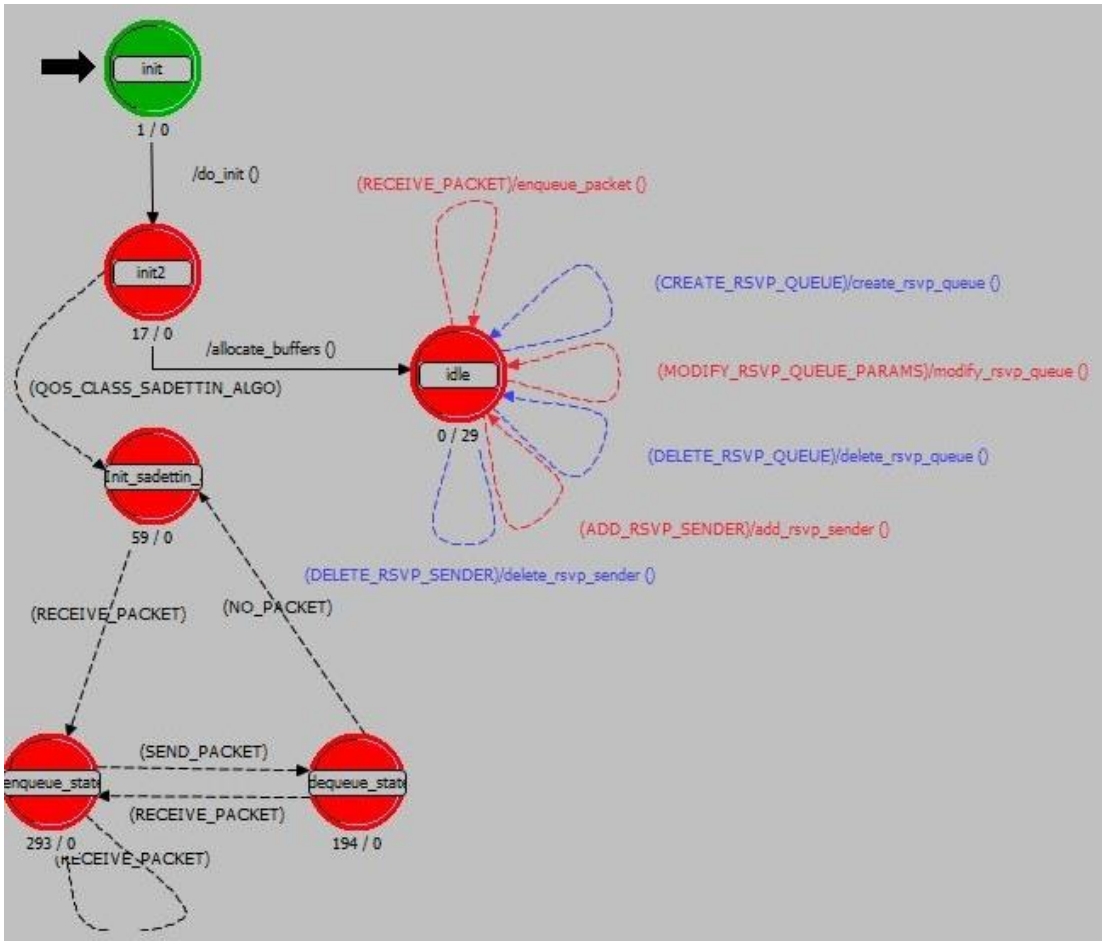
Şekil 4.5. Yönlendirici üzerinde paket hareketleri

Bu noktada gerçek zaman uygulamalarına ait paketlerin belirlenip metrik değerine göre işaretlenmesi bitmiştir. Artık uygun kuyruğa yerleştirme ve gönderme aşamasına geçilecektir.

4.1.5. Yönlendiriciler üzerinde önceliklendirme algoritmasının uygulanması

OPNET içerisinde, planlama tekniklerine ait süreçler, ip sürecinin bir yavru süreci olarak çalışırlar. Bu süreçlerden "ip_output_iface" süreci, yönlendirici üzerinde akan çeşitli veri paketlerinin ayrı kuyruklara yerleştirilmesinden ve planları dahilinde bu kuyruklarda bulunan paketlerin gönderilmesinden sorumludurlar.

Bu çalışmada önerilen önceliklendirme modelinin uygulanması için, OPNET üzerinde mevcut olan "ip_output_iface" süreç modeli üzerinde bir yavru süreç tanımlanmıştır. Yeni modele ait durum geçiş diyagramı init, enqueue, dequeue ve idle durumlarını içermektedir. Bu yapı Şekil 4.6.'da gösterilmiştir.



Şekil 4.6. Önceliklendirme algoritmasına ait durum geçiş diyagramı

4.1.5.1. Init state durumu

Init state durumunda bulunmak demek, gerçek zamanlı uygulamalara ait bir paketin gelmesi için beklemek anlamındadır. Bu durumda beklerken bir paket gelirse kuyruğa yerleştirme (Enqueue State) durumuna geçilir.

4.1.5.2. Enqueue state durumu

Algoritma şeması üzerinde "E" ile gösterilen alandır. Gerçek zamanlı uygulamaya ait bir paket gelirse bu duruma geçiş yapılır. Bu adım gelen paketin ilgili kuyruğa yerleştirme adımıdır.

Bu durum fonksiyonları aşağıdaki gibidir.

1. Bir paket alındığında, paket içerisine eklenen veya daha önceden eklenmiş ve mevcut olan önceliklendirme değerine göre paketi ilgili kuyruğa yerleştir.
2. Eğer bir kuyruğa bir paket yerleştirilirse ve bu kuyruk o ana kadar boş ise bu kuyruğa ait bayrağın durumunu değiştir. Bu bayrak bize kuyruğun içerisinde paket var mı yok mu sorusunun cevabını verecektir. Eğer kuyruk da paket var ise paket gönderim sürecinde bu kuyruk değerlendirilecektir.

```
queueFlags = allocate_memory ( 5 * sizeof (int) ) ;
```

3. Yeni bir paket geldi mi kontrol et. Eğer geldiyse kuyrukta bulunan paketler maksimum sayıya ulaştı mı? Maksimum sayıya ulaştı ise 4. adıma geç, eğer ulaşmadıysa 1. adıma dön.

```
checkPacketLabel : ip_pkptr_dec = op_pk_get (instrm_from_network);
if (ip_pkptr_dec == OPC_NIL) {
Op_intrpt_schedule_self (op_sim_time (), 102)
}
```

4. Eğer kuyruklar dolu ise veya işlenen bir paket yok ise kuyrukta bekleyen paketleri göndermek için en yüksek öncelikli kuyruktan itibaren kontrol et ve "dequeue state" durumuna geç

Bu çalışmada yönlendirme protokolü olarak RIPng ve BGP protokolü kullanılmıştır. RIPng protokolüne ait metrik değerleri 0-15 arasındadır, BGP protokolü için de aynı değerler kabul edilmiştir ve gerçek zamanlı paketler için 4 adet kuyruk tanımlanmıştır. Sistem üzerinde bir tane de best effort veri olmak üzere toplam 5 kuyruk vardır.

Çalışma öncelikle her bir metrik değeri için bir tane olmak üzere toplam 16 kuyruk tanımlanarak yapılmıştır. Ancak bu durumda algoritma ve işlem karmaşıklığı nedeni ile seğirme değeri sınır değerlerin üzerine çıkmıştır. Bu sorunu çözmek amacı ile kuyruk sayısı toplam 4 kuyruk olacak şekilde düzenleme yapılmış ve sistem bu şekilde çalıştırılmıştır.

Önceliklendirme algoritması üzerinde bu tanımlama kodları ip_output_iface içerisinde ip_dispatch sürecinin bir yavru süreci gibi tanımlanmıştır ve aşağıdaki gibidir.

```
int flow_label = ip_dgram_fd_ptr_dec -> flow_label_field;
int metric =flow_label –  (2 ^ 19 + 2 ^ 18 + 2 ^ 17 + 2 ^ 16 );
for (int j = 0 ; j < 16 ; j++ )
{
if (metric == 2 ^ j )
                    {
                    if  (( j > 0 ) && ( j < 4 ))
                    {
                    enqueue_packet ( ip_pkptr_dec , queueTab[3] ) ;
                    break ;
                    }
                    else if  (( j > 3 ) && ( j < 8 ))
                    {
                    enqueue_packet ( ip_pkptr_dec , queueTab[2] ) ;
                    break ;
                    }
                            else if  (( j > 7 ) && ( j < 12 ))
```

```
                    {
                    enqueue_packet ( ip_pkptr_dec , queueTab[1] ) ;
                    break ;
                    }
                            else if  (( j > 11 ) && ( j < 16 ))
                    {
                    enqueue_packet ( ip_pkptr_dec , queueTab[0] ) ;
                    break ;
                    }
            }
    }
```

4.1.5.3. Dequeue state durumu

Algoritma şeması üzerinde "F" ile gösterilen alandır. Eğer paketler gelmeye devam etse bile kuyruklar dolduysa veya gelen bir paket yok ise bu duruma geçilir ve kuyrukta bekleyen paketler gönderilir. Bu duruma ait fonksiyonlar da aşağıdaki gibidir.

1. En yüksek öncelik değerine ait kuyruktan başlamak üzere tüm kuyrukları kontrol et. Eğer kontrol ettiğin son kuyrukta boş ise tüm kuyrukların boş olduğuna dair bayrağı işaretle.

2. Eğer bir kuyruğun bayrağı kuyrukta paket var diye işaretlendiyse bu kuyruktaki paketleri gönder. Eğer bu kuyruğa ait gönderecek başka paket kalmadıysa kuyruğa ait bayrağı boş olarak işaretle.

3. Kuyruk üzerindeki paketleri gönderirken, kuyruk boşalmadan kuyruk için ayrılan süre dolduysa bir sonraki kuyruğa geç.

4. Yeni gelen paket yoksa veya tüm kuyruklar boş ise init state durumuna geç, aksi durumda 1. adıma dön.

```
for ( int i = 3 ; i = 0; i++ )
{
        if (queueFlag[i] == 1) {
                int notemply = 0 ;
                while (  notemply == 0)
        {
        pkptr = dequeue_packet (queueTab[i]);
        if ( pkptr == null) {
        notemply = 1;
        }
        else {


        op_pk_send (ip_pkptr, outstrm_to_network);
        NbrPackets--;
        }

        if ( (MAX_PACKET – NbrPacket ) == 10 )
        {

        Op_intrpt_schedule_self (op_sim_time) () ; 103);
        } } }
```

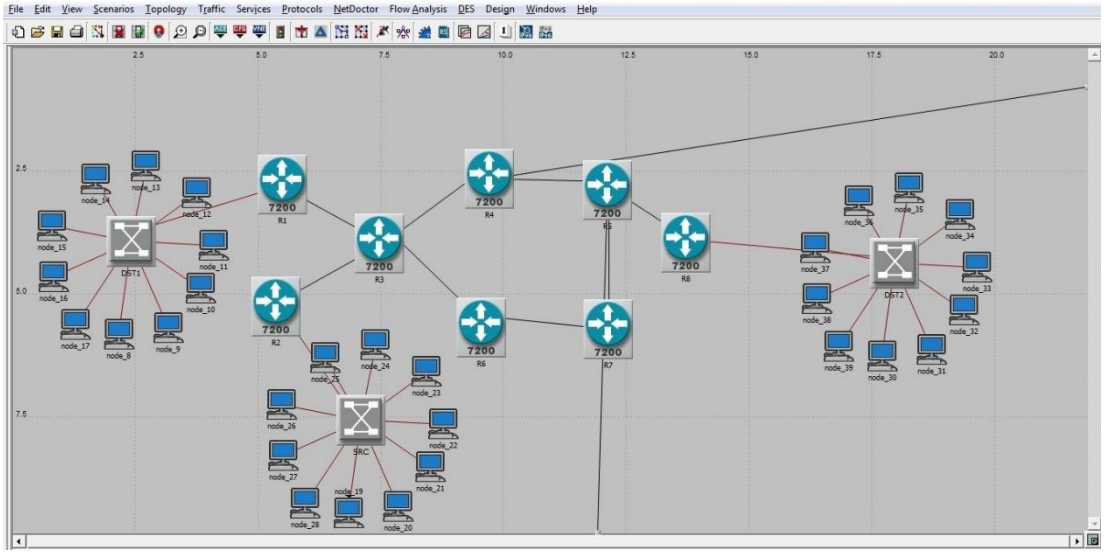
Bu noktada, tasarlanmış ve geliştirilmiş olan önceliklendirme algoritması referans model üzerinden elde edilen topoloji üzerindeki yönlendiricilere uygulanmış olmaktadır. Artık elimizde önceliklendirme algoritmasının başarısını değerlendirme amacı ile bir referans topoloji bir de önceliklendirme algoritmasının uygulandığı ikinci bir topoloji mevcuttur. Farklı senaryolar altında önceliklendirme modeli üzerinde sonuçlar alınacak ve referans topoloji üzerinden alınan değerler ile karşılaştırılarak önceliklendirme modelinin başarımı hesaplanacaktır.

4.2. Elde Edilen Sonuçlar

Tasarlanan önceliklendirme algoritması olası tüm ağ senaryolarında denenmiştir. Bu senaryolar ve bulunan sonuçlar aşağıda verilmiştir.

4.2.1. Birinci senaryo: Kaynak ve hedeflerin aynı otonom sistemde bulunması

Birinci senaryoda kaynak ile bir adet düşük bir adet de yüksek metrik değerli iki hedefin aynı otonom sistem içinde bulunması durumu değerlendirilmiştir. Senaryo ya ait topolojik yapı Şekil 4.7.'de gösterilmiştir.



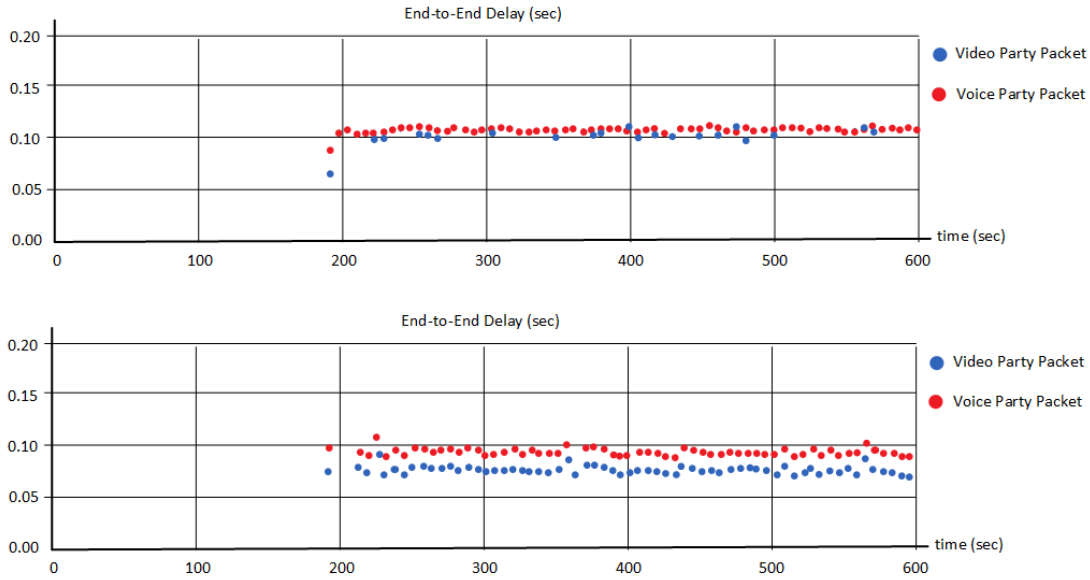
Şekil 4.7. Kaynak ve hedeflerin aynı otonom sistemde bulunduğu topolojik yapı

Kaynak, R2 yönlendiricisine bağlı LAN (SRC) ortamı iken hedefler R1 (DST1) ve R8 (DST2) yönlendiricisine bağlı LAN ortamları olarak seçilmiştir. Bu noktada dikkat edilen husus, iki ayrı hedefin metrik değerleri dikkate alınarak bu hedeflere ait paketlerin farklı kuyruklara yerleşmesi olmuştur. Yukarıda açıklandığı gibi sistemde önceliklendirme kuyruğu olarak 4 adet kuyruk yapısı vardır ve bazı paketler metrik değerleri farklı olsa bile aynı kuyruğa yerleşebilmektedir. Kaynağa ait R2 yönlendiricisi üzerinde bulunan yönlendirme tablosu ve hedef ağlara ait metrik değerleri Şekil 4.8.'de verilmiştir. Burada sadece ilgili ağlara ait bilgiler bulunmaktadır, diğer ağlara ait bilgiler dikkate alınmamıştır.

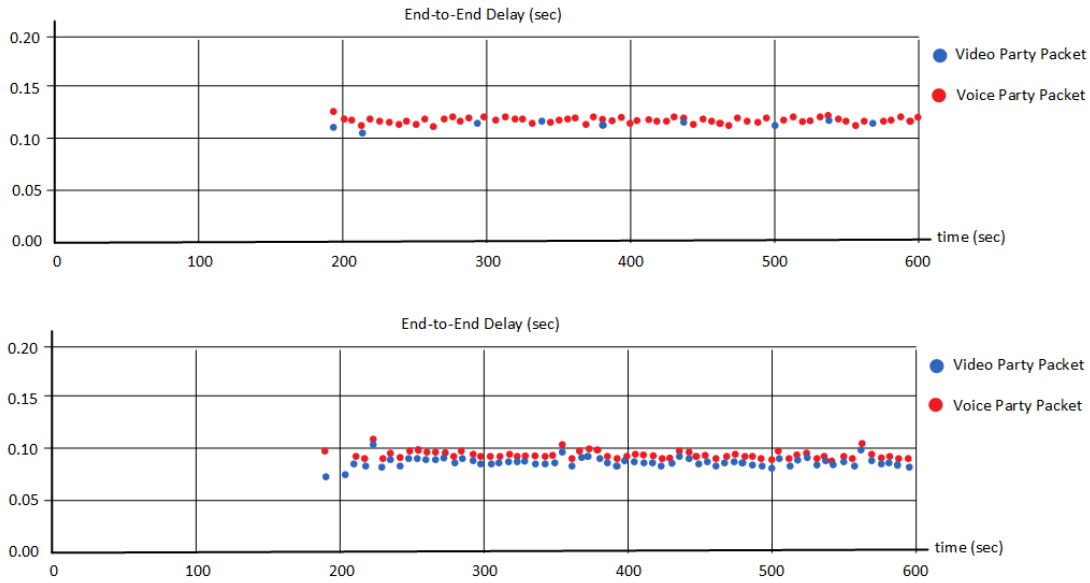
2005:0:0:72:0:0:0:0/64	RIPng	120	3	2005:0:0:1:0:0:0:1	Office Network.A1.R3	IF14
2005:0:0:73:0:0:0:0/64	Direct	0	0	2005:0:0:73:0:0:0:B	Office Network.A1.R2	IF2
2005:0:0:73:0:0:0:B/128	Local	0	0	2005:0:0:73:0:0:0:B	Office Network.A1.R2	IF2
2005:0:0:74:0:0:0:0/64	RIPng	120	5	2005:0:0:1:0:0:0:1	Office Network.A1.R3	IF14

Şekil 4.8. R2 (SRC) yönlendiricisine ait yönlendirme tablosu ve metrik değerleri

Bu seçimler kapsamında önce referans topolojiye ait sonuçlar sonra da geliştirdiğimiz önceliklendirme algoritmasının uygulandığı topolojiye ait sonuçlar karşılaştırmalı olarak verilecektir.



Şekil 4.9. SRC ağından DST1 ağına giden ses ve video paketleri için gecikme değerleri (a) önceliklendirme algoritması uygulanmadan önce (b) önceliklendirme algoritması uygulandıktan sonra



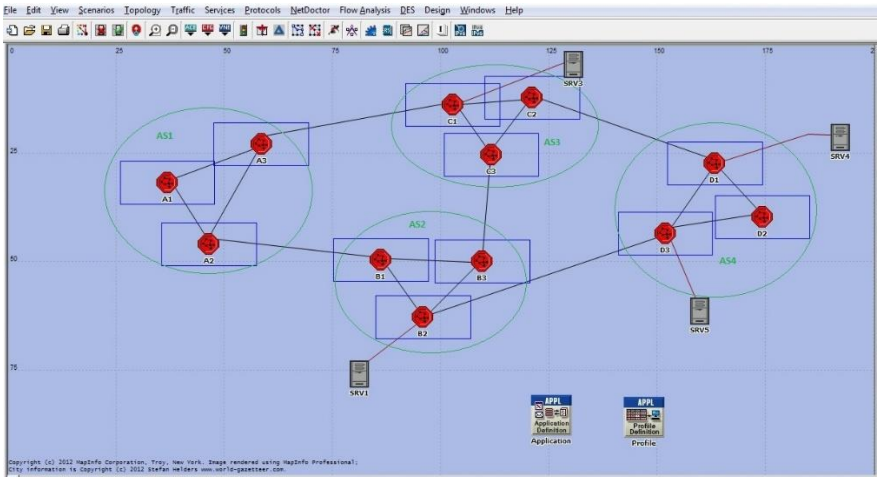
Şekil 4.10. SRC ağından DST2 ağına giden ses ve video paketleri için gecikme değerleri (a) önceliklendirme algoritması uygulanmadan önce (b) önceliklendirme algoritması uygulandıktan sonra

Kaynak olarak seçilen SRC ağından, hedef olarak seçilmiş olan düşük metrik değerine sahip DST1 ağına gitmekte olan ses ve video paketleri dikkate alınarak elde edilen sonuçlar Şekil 4.9.'da verilmiştir. Önceliklendirme algoritması uygulanmadan önce ses paketleri için gecikme değeri 113ms iken video paketleri için bu değer 111ms olarak gerçekleşmiştir. Tasarlanmış olan önceliklendirme algoritması uygulandıktan sonra ise gecikme değeri ses paketleri için 81ms, video paketleri için 72ms olarak ölçülmüştür. Ses paketleri için fark 32ms ve video paketleri için fark ise 39ms olmuştur. Bu durumda ses paketleri için iyileştirme oranı %28,31, video paketleri için ise %35,13 olarak gerçekleşmiştir.

Kaynak olan SRC ağından yüksek metrik değerine sahip DST2 ağına gitmekte olan ses ve video paketlerine ait veriler Şekil 4.10.'da gösterilmiştir. DST2 hedef ağı için önceliklendirme algoritması uygulanmadan önce ses paketleri için gecikme değeri 116ms, video paketleri için 119ms'dir. Önceliklendirme algoritması uygulandıktan sonra ses paketleri için gecikme değeri 94ms, video paketleri için 92ms olarak gerçekleşmiştir. Ses paketleri için fark 22ms, video paketleri için 27ms olmuştur. Bu durumda iyileştirme oranı ses paketleri için %19,96, video paketleri için %22,68'dir.

4.2.2. İkinci senaryo: Kaynak ve hedeflerin ayrı otonom sistemlerde bulunması

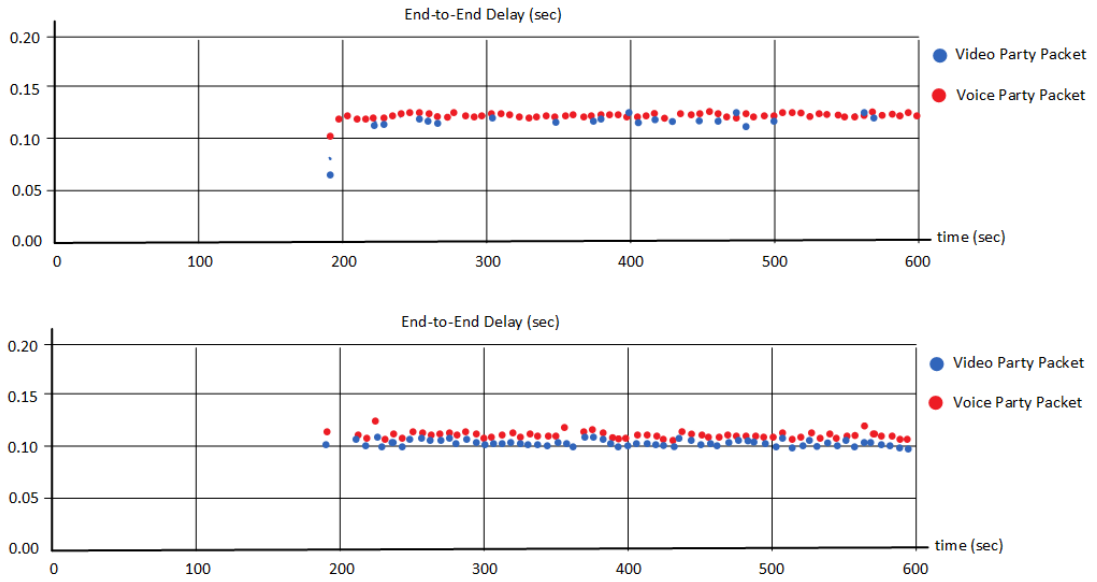
İkinci senaryoda ise kaynak ve hedefler farklı otonom sistemler içinde bulunmaktadır. Bu senaryoya ait topolojik yapı Şekil 4.11.'de verilmiştir.



Şekil 4.11. Kaynak ve hedeflerin ayrı otonom sistemde bulunduğu topolojik yapı

Bu senaryoda kaynak (SRC) AS1 otonom sistemi içindeki A1 alt ağı içinde bulunurken, hedefin biri (DST3) AS3 otonom sistemi içindeki C2 alt ağı içinde ve ikinci hedef ise (DST4) AS4 otonom sistemi içerisinde bulunan D2 alt ağı içerisinde bulunmaktadır.

Bu senaryoda otonom sistemleri arasında çalışan BGP protokolünün en iyi yola ait metrik değeri otonom sistem içinde çalışan RIPng protokolüne aktarılırken sabit 10 değeri kullanılmıştır. Böylece BGP protokolüne ait metrik değeri sabit bir dönüşüm ile RIPng protokolüne iletilmiş ve RIPng protokolü metrik değeri hesaplarken bu sabit değeri kullanmıştır.



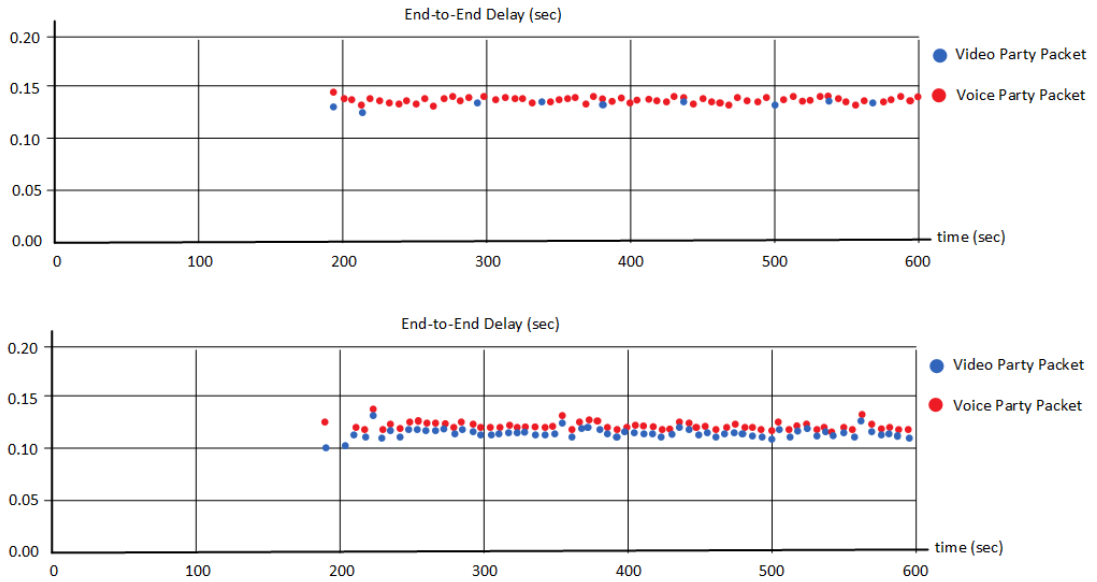
Şekil 4.12. SRC ağından DST3 ağına giden ses ve video paketleri için gecikme değerleri (a) önceliklendirme algoritması uygulanmadan önce (b) önceliklendirme algoritması uygulandıktan sonra

Kaynak olarak seçilen SRC ağından düşük metrik değerli DST3 ağına gitmekte olan ses ve video paketleri için önceliklendirme algoritması uygulanmadan önce ve uygulandıktan sonra elde edilen değerler Şekil 4.12.'de verilmiştir.

Önceliklendirme algoritması uygulanmadan önce referans topoloji üzerinde alınan sonuçlara göre ses paketleri için gecikme değeri 131ms olarak, video paketleri için ise gecikme değeri 134ms olarak ölçülmüştür. Önceliklendirme algoritması

uygulandıktan sonra ise ses paketleri için gecikme değeri 109ms, video paketleri için ise 105ms olarak gerçekleşmiştir. Bu sonuçlar doğrultusunda ses paketleri için oluşan gecikme değeri 22ms, video paketleri için 29ms iyileştirme sağlanmıştır. Bu durumda oluşan iyileştirme oranı ise ses paketleri için %16,79 olurken video paketleri için bu oran %21,64 olarak gerçekleşmiştir.

Kaynak olarak seçilen SRC ağından yüksek metrik değerli DST4 ağına gitmekte olan ses ve video paketleri için önceliklendirme algoritması uygulanmadan önce ve uygulandıktan sonra elde edilen değerler Şekil 4.13.'te verilmiştir.

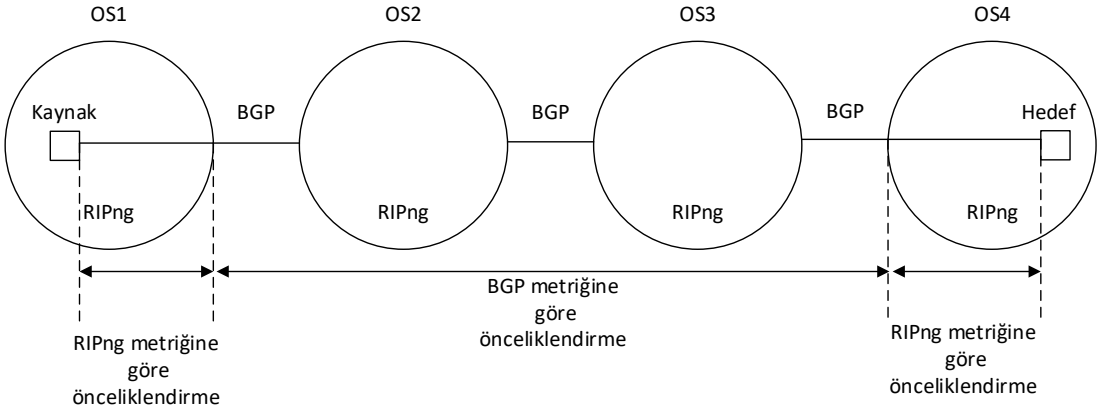


Şekil 4.13. SRC ağından DST4 ağına giden ses ve video paketleri için gecikme değerleri (a) önceliklendirme algoritması uygulanmadan önce (b) önceliklendirme algoritması uygulandıktan sonra

SRC kaynağı ile hedef olarak seçilen DST4 arasında gerçekleşen ses ve video trafiği üzerinde alınan sonuçlar ise; önceliklendirme algoritması uygulanmadan önce ses paketlerinin gecikmesi 139ms, video paketlerinin gecikme değeri 141ms olarak ölçülmüştür. Önceliklendirme algoritması uygulandıktan sonra ise ses için bu değer 111ms, video için 107ms olarak gerçekleşmiştir. Bu durumda ses paketleri için fark 28ms, video paketleri için ise fark 34ms olarak bulunmuştur. İyileştirme oranları ise ses için %20,14 olarak gerçekleşirken video için bu oran %24,11 olarak gerçekleşmiştir.

4.2.3. Üçüncü senaryo: Kaynak ve hedeflerin ayrı otonom sistemlerde bulunurken BGP protokolü üzerinde önceliklendirme yapılması

Üzerinde çalışılan ikinci senaryoda BGP protokolü tarafından hesaplanan metrik değeri RIPng protokolüne aktarılırken sabit bir dönüşüm yapılmıştır. Bu durum BGP tarafından hesaplanan metrik değerlerinin sağlıklı bir şekilde sonuca etki etmemesine neden olmuştur. Bu durumu ortadan kaldırmak için uygulanan üçüncü senaryoda BGP protokolü tarafından da önceliklendirme yapılması sağlanmıştır. Bu senaryoda uygulanan topoloji Şekil 4.11.'de kullanılan yapı ise Şekil 4.14.'te verilmiştir.

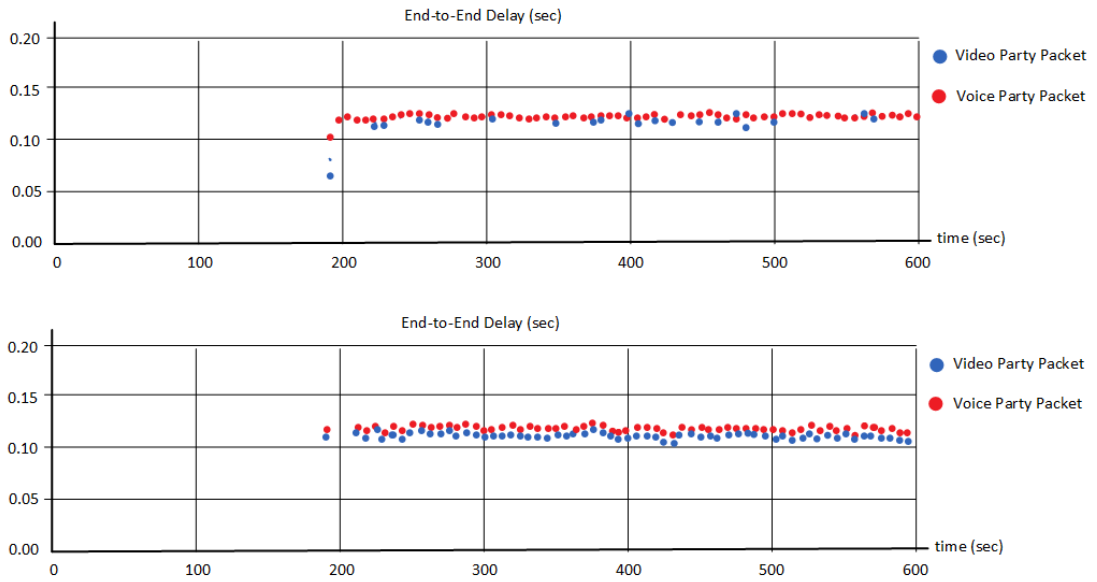


Şekil 4.14. BGP protokol metrik değerine göre yapılan önceliklendirmeye ait blok yapı

Bu yapıda otonom sistem içerisinde kaynak tarafından üretilen ses ve video paketi uğramış olduğu ilk yönlendirici üzerinde çalışmakta olan RIPng protokolü metrik değeri üzerinden önceliklendirilmiştir. Paket bu önceliklendirme değeri ile otonom sistem sınırında bulunan yönlendiriciye gelmiştir. Otonom sistem sınırında bulunan bu yönlendirici üzerinde RIPng ve BGP protokolü beraber çalışmaktadır. Bu yönlendiriciye gelen paket üzerindeki RIPng metrik değeri dikkate alınarak eklenen önceliklendirme değeri kaldırılarak BGP tarafından hesaplanan en iyi yol metrik değerine göre yeni bir önceliklendirme değeri eklenir. BGP protokolü en iyi yol değerini paketin geçeceği otonom sistemlerin numaralarından oluşan bir dizi şeklinde tutmaktadır. Bu dizi elemanları sayılarak bir değer elde edilmiş ve bu değer metrik değeri olarak önceliklendirme algoritması tarafından paket içerisine eklenmiştir. Bu

önceliklendirme değerine göre otonom sistemler arasında ilerleyen paket hedef otonom sistem sınırında tekrar bir önceliklendirme işlemine tabi tutulmaktadır. Hedef ağın bulunduğu otonom sistemin sınır yönlendiricisine gelen paket üzerindeki BGP metrik değerine göre eklenen önceliklendirme değeri kaldırılarak yine otonom sistem içinde çalışmakta olan RIPng protokol metrik değerine göre yeni bir önceliklendirme değeri eklenerek son otonom sistem içerisinde hedefe doğru iletilir.

Bu yöntemde otonom sistem içerisinde RIPng protokolüne göre yapılan önceliklendirme işlemi otonom sistemler arasında BGP protokolüne göre yapılmıştır. Bu senaryoda BGP protokol metriği sistem içerisinde gerçek değer olarak dikkate alınmasına rağmen aynı paket üzerinde toplamda üç defa önceliklendirme işlemi gerçekleştirilmiştir.

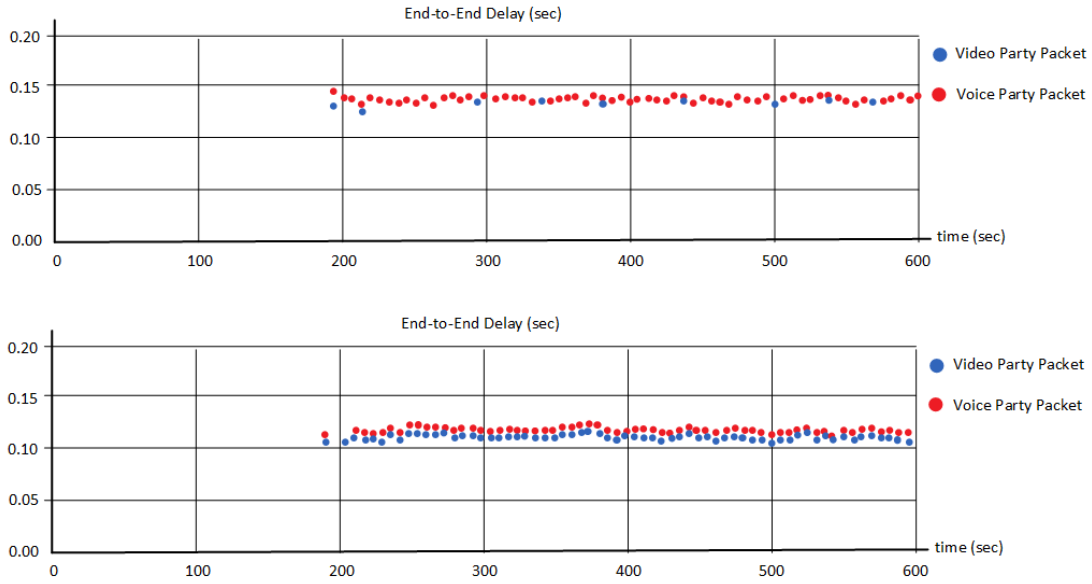


Şekil 4.15. SRC ağından DST5 ağına giden ses ve video paketleri için gecikme değerleri (a) önceliklendirme algoritması uygulanmadan önce (b) önceliklendirme algoritması uygulandıktan sonra

Kaynak olan SRC ağından hedef olan ve düşük metrik değerine sahip DST5 hedefine bu senaryo içerisinde bulunan sonuçlar Şekil 4.15.'te verilmiştir.

Bu senaryo dahilinde ses paketleri için referans topoloji üzerinde gecikme değeri 131ms olarak gerçekleşirken bu değer video paketleri için 134ms olarak gerçekleşmiştir. Önceliklendirme algoritması uygulandıktan sonra ise ses paketleri için

gecikme değeri 118ms olarak gerçekleşmiş ve video paketlerinin gecikme değeri 115ms olarak ölçülmüştür. Bu sonuçlar doğrultusunda ses paketleri için iyileştirme oranı %9,92 olarak bulunurken bu değer video paketleri için %14,17 olarak gerçekleşmiştir.



Şekil 4.16. SRC ağından DST6 ağına giden ses ve video paketleri için gecikme değerleri (a) önceliklendirme algoritması uygulanmadan önce (b) önceliklendirme algoritması uygulandıktan sonra

SRC kaynağından mevcut senaryo dahilinde yüksek metrik değerine sahip DST6 hedefine giden ses ve video paketlerine ait sonuçlar Şekil 4.16.'da gösterilmiştir.

SRC kaynağından DST6 hedefine gitmekte olan ses ve video paketleri için önceliklendirme algoritması uygulanmadan önce elde edilen gecikme değerleri sırası ile 139ms ve 141ms'dir. Bu değerler önceliklendirme algoritması kullanıldıktan sonra ses paketleri için 119ms ve video paketleri için 116ms olarak gerçekleşmiştir. Bu sonuçlardan sonra iyileştirme oranı ses paketleri için %14,38, video paketleri için %17,73 olarak hesaplanmıştır.

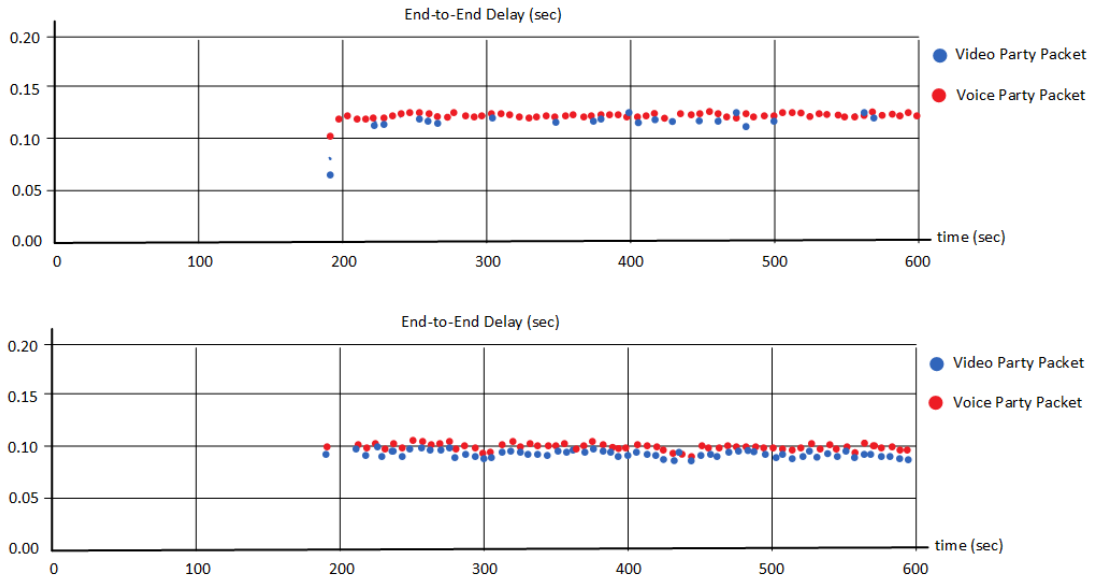
4.2.4. Dördüncü senaryo: Kaynak ve hedeflerin ayrı otonom sistemlerde bulunurken Hop Limit değerine göre önceliklendirme yapılması

Şimdiye kadar üzerinde çalışılan tüm senaryolarda yönlendirme protokolü metrik değeri üzerinden önceliklendirme yapılmıştır. Birinci senaryoda herhangi bir metrik dönüşümü gerekmediği için herhangi bir sorun yaşanmamıştır. İkinci senaryoda BGP ve RIPng protokolü metrik dönüşümünde sabit bir değer üzerinden dönüşüm yapıldığı için, metrik değer gerçek uzaklığı ifade etmemiştir. Üçüncü senaryoda ise otonom sistem içinde RIPng metrik değerine, otonom sistemler arası BGP metrik değerine göre önceliklendirme yapıldığı için gerçek uzaklık değeri olarak kabul ettiğimiz geçilen yönlendirici sayısını vermemiştir.

Dördüncü senaryoda farklı bir yaklaşım kullanılmıştır. İşletim sistemlerinde kullanılan ping komutu kaynaktan hedefe doğru bir kontrol mesajı gönderir (ICMPv6 Echo Repuest), bu mesajı alan hedef de kaynağa bir kontrol mesajı ile (ICMPv6 Echo Reply) ile cevap verir. Bu şekilde kaynak ve hedef birbirlerinin aktif olarak çalıştıklarından haberdar olurlar. Kontrol mesajı IPv6 paketi içine yerleştirilir ve gönderildiğinde bu kontrol mesajı bir IPv6 başlığına sahip olur. IPv6 bağlığında bulunan “hop limit” değeri ise 8 bit uzunluğundadır ve maksimum 255 değerini alabilir. Bu değer geçmiş olduğu her bir yönlendirici üzerinde 1 azaltılarak iletim yapılır ki eğer değer 0 olursa paket sistemden atılır. Hop limit değerinin 0 olması paketin döngüye girdiğini ve hedefe ulaşamadığını ifade etmektedir. Bu noktadan hareket ile kaynaktan hedefe gönderilen kontrol mesajına ait paket bağlığındaki hop limit değeri gerçekte yönlendirme protokolünden bağımsız olarak geçilen yönlendirici sayısını vermektedir.

İlk üç senaryoda, paketlerin önceliklendirilmesi yönlendiriciler üzerinde gerçekleştirilmiştir. Bu senaryoda ise, paketlerin önceliklendirilmesi kaynak üzerinde gerçekleştirilmektedir. Hop limit değerine göre paket kaynak tarafından üretim aşamasında önceliklendirilir. Kaynaktan çıkan paket yol boyunca uğradığı yönlendiriciler tarafından sadece öncelik değerine göre ilgili kuyruğa yerleştirme ve gönderim işlemini gerçekleştir.

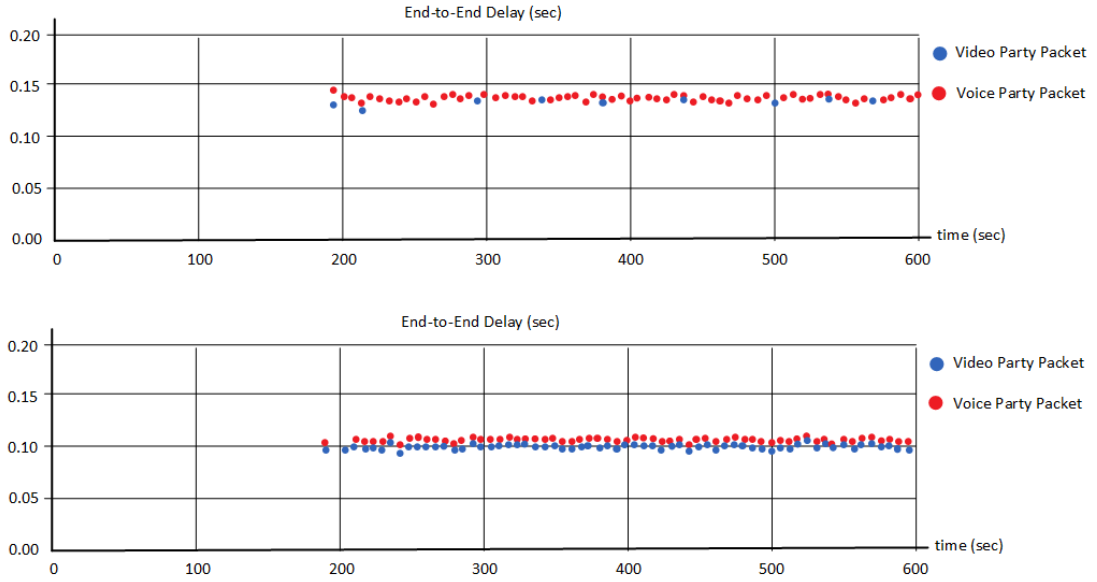
Bu senaryoda uygulanan topoloji Şekil 4.9.'da verilmiştir ve hop limit değeri 255 olarak ayarlanmıştır. Hedefin gönderdiği cevap paketinin hop limit değeri ile arasındaki fark geçilen yönlendirici sayısını verdiğinden dolayı bu fark önceliklendirme değeri olarak kullanılmıştır. Bu noktada düşük metrik değerine sahip ağ için hop limit değeri 14, yüksek metrik değerli ağa ait hop limit değeri 22 olarak bulunmuştur. Tasarlanan önceliklendirme yapısında 4 öncelik kuyruğu bulunduğu için; 0-9 arası hop limit değerine sahip paketler 4 nolu kuyruğa, 10-19 arası hop limit değerine sahip paketler 3 nolu kuyruğa, 20-29 arası hop limit değerine sahip paketler 2 nolu kuyruğa, 30-39 arası hop limit değerine sahip paketler 1 nolu kuyruğa yerleştirilecek şekilde ayarlama yapılmış ve bu iki ağa ait paketlerin aynı kuyruğa yerleşmesi engellenmiştir. Sonuçların alınmasında, sadece ses ve video paketlerin gecikme değerleri dikkate alınmıştır. Sistemin çalışmaya başladığı andan ilk GZU paketinin gönderilmesi arasında gerçekleşen ICMP Echo ve Reply mesajlarının süreleri gecikme değerlerine dahil edilmemiştir.



Şekil 4.17. SRC ağından DST7 ağına giden ses ve video paketleri için gecikme değerleri (a) önceliklendirme algoritması uygulanmadan önce (b) önceliklendirme algoritması uygulandıktan sonra

Kaynak olan SRC ağından hedef olan ve düşük metrik değerine sahip DST7 hedefine bu senaryo içerisinde bulunan sonuçlar Şekil 4.17.'de verilmiştir.

SRC kaynağından düşük metrik değerli olan DST7 hedefine gitmekte olan ses paketleri için önceliklendirme yapılmadan önce 131ms ve video paketleri için 134ms olan gecikme değerleri öncelikle yapıldıktan sonra ses paketleri için 103ms ve video paketleri için 96ms olarak gerçekleşmiştir. Bu sonuçlara göre ses paketlerinde iyileştirme oranı %21,37 olarak gerçekleşirken video paketleri için bu oran %28,50 olarak bulunmuştur.



Şekil 4.18. SRC ağından DST8 ağına giden ses ve video paketleri için gecikme değerleri (a) önceliklendirme algoritması uygulanmadan önce (b) önceliklendirme algoritması uygulandıktan sonra

Kaynak olan SRC ağından yüksek metrik değerine sahip hedef olan DST8 ağına bu senaryo dahilinde bulunan sonuçlar Şekil 4.18.'de verilmiştir. Önceliklendirme algoritması uygulanmadan önce ses paketleri için gecikme değeri 139ms, video paketleri için bu değer 141ms olarak bulunmuştur. Önceliklendirme algoritması uygulandıktan sonra elde edilen gecikme değeri ses paketleri için 105ms ve video paketleri için 101ms olarak gerçekleşmiştir. Bu durumda iyileştirme oranları ses paketleri için %24,46 iken video paketleri için %28,36 olarak bulunmuştur.

4.3. Sonuçların Değerlendirilmesi

Tasarlanan önceliklendirme algoritması dört farklı senaryo üzerinde hem düşük metrik değerine sahip hem de yüksek metrik değerine sahip toplam sekiz farklı hedef üzerinde denenmiştir.

Kaynak ve hedeflerin aynı otonom sistem içinde yer aldığı ve yönlendirme protokol metriğine göre önceliklendirmenin yapıldığı birinci senaryoda en yüksek iyileştirme %35,13 oranı ile düşük metrik değerine sahip ağa gitmekte olan video paketleri üzerinde elde edilmiştir.

Kaynak ve hedeflerin ayrı otonom sistem içinde yer aldığı, BGP protokol metrik değerinin sabit dönüşüm yapıldığı ve önceliklendirme için metrik değerlerinin dikkate alındığı ikinci senaryoda ise en yüksek iyileştirme değeri %24,11 ile yüksek metrik değerli ağa gitmekte olan video paketleri üzerinde elde edilmiştir.

Yine kaynak ve hedeflerin ayrı otonom sistemlerde yer aldığı ancak otonom sistem içerisinde RIPng protokol metriğine göre, otonom sistemler arası BGP metrik değerine göre önceliklendirmenin yapıldığı üçüncü senaryoda ise gecikme değerinde elde edilen en yüksek iyileştirme değeri %17,73 ile yüksek metrik değerli ağa gitmekte olan video paketlerinde elde edilmiştir.

Dördüncü senaryo ise ilk üç senaryodan farklı olarak yönlendirme protokol metrik değerine göre değil, hop limit değerine göre önceliklendirme yapılmıştır. İlk üç senaryoda önceliklendirme yönlendiriciler üzerinde yapılırken bu senaryoda kaynakta önceliklendirme yapılmıştır. Bu senaryo da ise gecikme değeri üzerinde elde edilen en yüksek iyileştirme oranı %28,36 ile yüksek metrik değerli hedefe giden video paketleri üzerinde olmuştur.

Bulunan tüm sonuçlar Tablo 4.1.'de toplu olarak ve karşılaştırmalı bir şekilde verilmiştir.

Tablo 4.1. Önceliklendirme algoritması kullanılmadan önce ve kullanıldıktan sonra kaynak ağdan hedef ağlara giden ses ve video paketleri için gecikme değerleri

			Önceliklendirme algoritmasından önce gecikme değerleri (ms)	Önceliklendirme algoritmasından sonra gecikme değerleri (ms)	Fark (ms)	İyileştirme oranı
Birinci Senaryo	DST1	Ses	113	81	32	28,31%
		Video	111	72	39	35,13%
	DST2	Ses	116	94	22	19,96%
		Video	119	92	27	22,68%
İkinci Senaryo	DST3	Ses	131	109	22	16,79%
		Video	134	105	29	21,64%
	DST4	Ses	139	111	28	20,14%
		Video	141	107	34	24,11%
Üçüncü Senaryo	DST5	Ses	131	118	13	9,92%
		Video	134	115	19	14,17%
	DST6	Ses	139	119	20	14,38%
		Video	141	116	25	17,73%
Dördüncü Senaryo	DST7	Ses	131	103	28	21,37%
		Video	134	96	38	28,50%
	DST8	Ses	139	105	34	24,46%
		Video	141	101	40	28,36%

Tabloda da görüldüğü gibi tüm senaryolarda video paketlerinin gecikme değerleri ses paketlerinin gecikme değerlerine göre daha iyi sonuçlar elde edilmiştir. Bunun nedeni önceliklendirme algoritması tasarımında toplam dört adet kuyruk kullanılması ve aynı metrik değerine göre önceliklendirilen hem video hem de ses paketlerinin aynı kuyruğa alınmasıdır. Bir ses paketi kendinden önce işlenmekte olan video paketini, bir video paketi de kendinden önce işlenmekte olan ses paketinin beklemektedir. Video paket boyutları (~900-1500 byte) ses paket boyutlarından (~75-250 byte) çok daha büyük olduğu için kuyrukta bekleme süreleri göz önüne alındığında bir ses paketi video paketinden daha uzun süre kuyrukta beklemektedir. Bundan dolayı elde edilen sonuçlarda da görülmektedir ki ses paketlerine ait gecikme değerleri video paketlerine ait gecikme değerlerinden daha büyüktür.

BÖLÜM 5. TARTIŞMA VE SONUÇ

Geleneksel internet mimarisi “best-effot” olarak isimlendirilen ilk gelen ilk servis alır mantığı ile oluşturulmuş bir mimaridir ve herhangi bir servisin önceliği bulunmamaktadır. Çünkü geleneksel internet mimarisinin oluşturulduğu yıllarda mevcut olan trafik herhangi bir önceliklendirmeye gerek duymamaktadır.

Ancak 20. yüzyılın son on yılı içerisinde başlayan ses ve video gibi gerçek zamanlı trafik, senkronizasyon ve kalite standartları nedeni ile belli bir zaman sınırı içerisinde hedefine varmak zorunda olması internet üzerinde paketlerin önceliklendirilmesi gerekliliğini ortaya koymuştur. Bunun sonucunda ise servis kalitesi kavramı ortaya çıkmış ve anlam kazanmaya başlamıştır.

Gerçek zamanlı servislerin ihtiyacı olan servis kalitesinin sağlanması amacı ile tarihsel süreç içerisinde birçok mimari çözüm amacı ile ortaya çıkmıştır. Ancak geneli OSI referans modelinin ikinci katmanında çalışan (ATM, Frame Relay vb.) bu teknolojiler yapıları itibarı ile gerekli olan QoS desteğini sadece mevcut bulundukları omurga üzerinde sağlayabilmişler uçtan-uca bir destek verememişlerdir.

Bu çalışma servislerin ihtiyaç duyduğu uçtan-uca servis desteği sağlamak için yapılmıştır. Bundan dolayı kaynaktan hedefe yapısı ve karakteristiği korunan IPv6 paket bağlığı ve dolayısı ile IPv6 protokolü kullanılmıştır. IPv6 başlığı içerisinde tanımlı ancak kullanımı kesin olarak kurala bağlanmamış FL alanı yardımı ile yönlendirme protokolünün hedef ağa ait metrik değeri üzerinden bir önceliklendirme yapısı tasarlanmış ve bu önceliklendirme değeri FL alanı içine yerleştirilerek yol üzerindeki bütün yönlendiricilerin bu önceliklendirme değerine göre gelen paketleri işlemesi sağlanmıştır. Böylece gerçek zaman uygulamaları için geleneksel internet

mimarisi içerisinde bulunmayan ve mevcut teknolojilerin yetersiz kaldığı uçtan-uca kesintisiz bir QoS desteği bu çalışma ile sağlanmış bulunmaktadır.

5.1. Sonuçlar

Yapılan çalışmada önerilen önceliklendirme modelinin başarımını ölçmek amacı ile bir referans model oluşturulmuş ve referans model üzerinden alınan sonuçlar önceliklendirme modeli üzerinden alınan sonuçlar ile karşılaştırılmıştır.

Bu karşılaştırma gerçek internet ortamında bulunabilecek şekilde hem otonom sistem içerisinde hem de otonom sistemler arasında olacak şekilde dört farklı senaryo kullanılarak yapılmıştır.

Elde edilen sonuçlar göstermektedir ki, önerilen önceliklendirme modeli gerçek zamanlı paketlerin ihtiyaç duydukları gecikme parametresi üzerinde başarılıdır. Şöyle ki; kaynak ve hedeflerin otonom sistem içinde bulundukları senaryoda ses paketlerinin gecikme değerinde 32ms, video paketlerine ait gecikme değeri için 39ms olmak üzere bir iyileştirme sağlanmıştır. Bu değerler ses paketleri üzerinde %28,31, video paketlerinde %35,13 bir iyileştirmeye karşılık gelmektedir.

Kaynak ve hedeflerin farklı otonom sistem içinde bulundukları senaryoda ise, ses paketleri gecikme değerleri için 28ms, video paketleri için 34ms iyileştirme olmuştur. Bu değerler de ses paketleri üzerinde %20,14, video paketlerinde %24,11 iyileştirme değerlerine karşılık gelmektedir.

İlk üç senaryo üzerinde önceliklendirme değeri olarak yönlendirme protokolünden sağlanan metrik değeri üzerinden bir önceliklendirme yapılırken dördüncü senaryo üzerinde IPv6 başlığı içerisinde bulunan hop limit alanı kullanılarak geçilen yönlendirici sayısı dikkate alınarak bir önceliklendirme yapılmıştır. Bu senaryoda ise ses paketleri gecikme değeri için 34ms, video paketleri gecikme değeri için ise 40ms bir iyileştirme gerçekleşmiştir. Bu değerler yüzdesel olarak ses paketleri için %24,46, video paketleri için %28,36 bir iyileştirme oranına karşılık gelmektedir.

Yapılan çalışama da elde edilen bu değerler dikkate alındığında, tasarlanan önceliklendirme modeli hem gerçek zaman uygulamalarının ihtiyacı olan uçtan-uca gecikme desteği sağlamış hem de QoS kavramının en önemli parametresi olan gecikme değerleri üzerinde önemli bir iyileştirme sağlamıştır.

5.2. Çalışmanın Bilime Katkısı

Bu çalışmada kullanılan önceliklendirme parametresi yani yönlendirme protokol metrik değeri tabanlı önceliklendirme, literatürde örneği olmayan bir çalışmadır. Dolayısı ile bu çalışma günümüz internet dünyasında en çok ilerleme kaydeden gerçek zamanlı servislerin ihtiyaç duyduğu QoS desteği için yeni bir bakış açısı ve yeni bir önceliklendirme parametresi ortaya koymuştur. Çalışma ayrıca bu ortaya konan yeni parametrenin QoS desteği için gecikme parametresi üzerinde olumlu sonuçlar verdiğini de yapılan testlerde ve elde edilen sonuçlarda göstermiştir. Bu çalışmada önerilen model, ikinci katman mimarisinden ve teknolojik altyapısından bağımsız olarak gerekli olan uçtan-uca desteği verebilmektedir. Bu durumda esnek bir uygulama alanı sunmaktadır. Dolayısı ile gerçek zamanlı uygulamaların servis desteği için yapılacak çalışmalara önemli bir destek sağlamış ve literatüre yeni bir kavram eklemiştir.

Bununla beraber, bu çalışma IPv6 bağlığı içinde bulunan ve kullanımı konusunda kesin bir yargıya sahip olmayan FL alanının kullanımı için yeni bir yaklaşım ortaya koymuştur. Literatürde bulunan kullanım yaklaşımlarına yeni bir kullanım yaklaşımı eklemiştir.

5.3. İleriki Çalışmalar

Bu çalışmada önceliklendirme parametresi olarak uzaklık-vektör tabanlı RIPng ve yol-vektör tabanlı BGP protokollerine ait metrik değerleri kullanılmıştır. Bundan dolayı ileriki çalışmalarda bu çalışmada kullanılmayan hat-durum protokolleri metrik değerleri kullanılarak yeni çalışmalar yapılabilir. Mevcut tüm yönlendirme protokollerini kapsayacak bir model geliştirilebilir.

Bunun yanında yine bu çalışmada ses ve video paketleri için öncelik tabanlı toplam dört kuyruk kullanılmıştır ve hem ses hem de video paketleri aynı kuyruk içerisinde işlenmiştir. İleriki çalışmalarda kuyruk sayısı arttırılarak ses ve video paketlerinin ayrı kuyruklarda işlenerek performans artışı sağlanabilecek çalışmalar yapılabilir. Ayrıca bu çalışmada kullanılan önceliklendirme kuyruğu haricinde farklı kuyruk tipleri üzerinde denemeler yapılabilir ve bu önceliklendirme modeli içerisinde farklı kuyruk tiplerinin performansları ölçülebilir.

KAYNAKLAR

[1] http://www.voipinsights.com/voip_history.html., Erişim Tarihi: 26.04.2020.

[2] Zhu, C., Yuenan, L., Xiamu, N., Streaming media architectures, techniques, and applications: Recent advances, IGI Global, New York, 26-27, 2010.

[3] https://gizmodo.com/the-worlds-first-webcam-was-created-to-check-a-coffee-p-5993583, Erişim Tarihi: 26.04.2020.

[4] https://www.youtube.com/watch?v=HRa2pE5-Ny0,Erişim Tarihi: 26.04.2020.

[5] Markoff, J., Cult film is a first on internet, The New York Times, 1993.

[6] http://www.fundinguniverse.com/company-histories/realnetworks-inc-history, Erişim Tarihi: 26.04.2020.

[7] Olawuyi, K., Sun, L., Investigation of the performance of a secure VoIP system, Advances in Communications, Computing, Electronics, Networks, Robotic and Security, 12, 95-104, 2015.

[8] Fergusan, P., Huston, G., Quality of Service: Delivering QoS on the Internet and in Corporate Networks, Wiley Press New York, NY, 1998.

[9] Ünverdi, N. Ö., Hâkî, E.H., Ip üzerinden ses iletiminde hizmet kalitesi, Bursa Elektrik Elektronik ve Bilgisayar Mühendisligi Sempozyumu, 2002.

[10] Szigeti T., Hattingh C., End-to-End QoS Network Design: Quality of Service in LANs, WANs, and VPNs. İçinde: Quality of Service Design Overview. 1st Press, Cisco Press, 367-465, 2004.

[11] Deering, S., Hinden, R., Internet protocol version 6 (IPv6) specification. IETF RFC 2460, Network Working Group, 1998.

[12] Velmurugan, T., Chandra, H., Balaji, S., Comparison of Queuing Disciplines for Differentiated Services Using OPNET, ARTCom'09, p: 744- 746, 2009.

[13] Bin Mohd Noor, M.N., Yahaya, C.K.H.C.K., Comparative of different queuing technique in high resolution video conference, FTP and VoIP, ICT Convergence (ICTC), p: 765- 769, 2012.

[14] Islam, M.Z., Islam, M.S., Haque, A.K.M.F., Ahmed, M., A comparative analysis of different real time applications over various queuing techniques, Informatics, Electronics & Vision (ICIEV), 2012 International Conference, p: 1118- 1123, 2012.

[15] Jung-Shyr, W., Peir-Yuan, W., The performance analysis of SIP-T signaling system in carrier class VoIP network, AINA 2003, p: 39- 44, 2003.

[16] Yerima, S.Y., Al-Begain, K., A Dynamic Buffer Management Scheme for End-to-End QoS Enhancement of Multi-flow Services in HSDPA, NGMAST '08, p: 370- 375, 2008.

[17] Yerima, S.Y., Al-Begain, K., End-to-End QoS Improvement of HSDPA End-User Multi-Flow Traffic Using RAN Buffer Management, NTMS '08, p: 1-5, 2008.

[18] https://ieeexplore.ieee.org/ Erişim Tarihi: 03.05.2020

[19] Filsfils, C., Evans, J., Engineering a multiservice IP backbone to support tight SLAs. Computer Networks, 40: 131–148, 2002.

[20] Iliadis, I., Scotton, P., Bauer D., Dynamic Transition matrix generation for topology aggregation. Computer Communication, 25: 1497-1512, 2002.

[21] Hagen, S., IPv6 Essentials, 2.Edition O'Reilly Press, 54-56, Sebastopol, CA, 2002.

[22] http://www.ind.alcatel.com/library/e-briefing/eBrief_QoS.pdf., Erişim Tarihi: 17.05.2020.

[23] http://www.technologyuk.net/telecommunications/telecom_principles/bandwidth.shtml., Erişim Tarihi: 17.05.2020.

[24] Prekash, B., Using The 20 Bit Flow Label Field in The IPv6, Graduate Thesis, Collage of Engineering, 2000.

[25] Cisco System, Understanding Delay in Packet Voice Networks. Document ID: 5125, 2006.

[26] Cisco System, Internetworking Technologies Handbook: FrameRelay, 2004. http://www.cisco.com/univercd/cc/td/doc/cisintwk/ito_doc/frame.pdf Erişim Tarihi: 27.05.2020

[27] Fergusan, P., Huston, G., Quality of Service: Delivering QoS on the Internet and in Corporate Networks, Wiley Press New York, NY, 1998.

[28] Stalling, W., Data & Computer Communications. Sixth Ed, Addison Wesley Longman Press, Delhi, India, 2001.

[29] Onvural, R.O., Asynchronous transfer mode networks: performance issues. Artech House Norwood, MA, 122, 1994.

[30] David, E., Darren L., ATM Theory and Applications. McGraw-Hill, Montreal, 1999.

[31] Neelakanta, P.S., A Textbook on ATM Telecommunications, Principles and implementation. CRC Press, 2000.

[32] Hucaby, D., Mcquerry, S., VLANs and Trunking. Cisco Press, 2002.

[33] Nichols, K., Blake, S., Baker, F., Black, D., Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers. IETF RFC 2474, Network Working Group, 1998.

[34] Postel, J., Internet Protocol Darpa Internet Program Protocol Specification. IETF RFC 791, Virginia, 1981.

[35] Stevens, R., TCP/IP Illustrated Volume 1: TheProtocols. Addison Wesley Longman Press, Delhi, India, 1999.

[36] Callon, R., Use of OSI IS-IS for Routing in TCP/IP and Dual Environments. IETF RFC 1195, Network Working Group, 1990.

[37] Moy, J., OSPF Version 2. IETF RFC 2328, Network Working Group, 1994.

[38] Partridge, C., A Proposed Flow Specification. IETF RFC 1363, Network Working Group, 1992.

[39] Braden, R., Clark, D., Shenker, S., Integrated Services in the Internet Architecture: an Overview, IETF RFC 1633, Network Working Group, 1994.

[40] H3C Technologies, QoS Technology White Paper. H3C Technologies Co., Hangzhou, 2008.

[41] Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z., An Architecture for Differentiated Services, IETF RFC 2475, Network Working Group, 1998.

[42] Tebbani, B., Haddadou, K., CODEC-based Adaptive QoS Control for VoWLAN with Differentiated Services. 1st IFIP Wireless Days (WD), 1-5, 2009.

[43] Cao, J., Gregory, M., Performance Evaluation of VoIP Services using Different CODECs over a UMTS Network. Telecommunication Networks and Applications Conference (ATNAC 2008), 67-71, 2008.

[44] Cristofaro, N.D., McGill, G., Sallahi A., QoS evaluation of a voice over IP network with video: A case study. 2009 Canadian Conference on Electrical and Computer Engineering, St. John's, NL, Canada, 2009.

[45] Ngamwongwattana, B., Effect of Packetization on VoIP Performance, 5th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), 373-376, 2008.

[46] Kos, A., Klepec, B., Tomazie, S., Techniques for Performance Improvement of VoIP Applications, 11th Mediterranean Electrotechnical Conference (MELECON), 250- 254, 2002.

[47] Yasinovskyy, R., Wijesınha, A., Karne, R., Khaksari, G., A Comparison of VoIP Performance on IPv6 and IPv4 Networks. International Conference on Computer Systems and Applications (AICCSA), 603-609, 2009.

[48] Fgee, E.B., Kenney, J.D., William, J., Robertson, W., Sivakumar, S., Comparison of QoS performance between IPv6 QoS management model and IntServ and DiffServ QoS models. Communication Networks and Services Research Conference, pp.287 – 292, 2005.

[49] Liu, N., Cao, J., Liu, M., Flow-Based Reservation Marking in MPLS Networks. Jiazhi Zeng Communications Conference (ICC'08), pp 414 – 418, 2008.

[50] Sun, X., Research on QoS of next generation network based on MPLS. IEEE International Conference on Information Science and Technology (ICIST'12), Wuhan, Hubei, China, 2012.

[51] Rahimi, M., Hashim, H., Rahman, R.A., Implementation of Quality of Service (QoS) in Multi-Protocol Label Switching (MPLS) networks. International Colloquium on Signal Processing & Its Applications (CSPA'2009), pp 98-103, 2009.

[52] Deering, S., Hinden, R., Internet Protocol, Version 6 (IPv6) Specification. IETF RFC 2460, Network Working Group, 1998.

[53] Kadayıf, İ., Kabal, O., Ipv4 Ağlarının Ipv6 Ağlarına Entegrasyonu. Çanakkale 18 Mart Üniversitesi Bilimsel Araştırma Projesi, Proje No: 2006 – 20, 2006.

[54] Kent, S., Atkinson, R., Security Architecture for the Internet Protocol. IETF RFC 2401, Network Working Group, 1998.

[55] Conta, A., Carpenter, B., A proposal for the IPv6 Flow Label Specification. IETF Internet Draft, IPng Working Groups, 2001.

[56] Conta, A., Rajahalme, J., A model for Diffserv use of the IPv6 Flow Label Specification. IETF Internet Draft, Diffserv Working Group, 2001.

[57] Banerjee, R., Malhotra, S., P., Mahaveer, M., A Modified Specification for use of the IPv6 Flow Label for providing efficient Quality of Service using a hybrid approach. IETF Internet Draft, IPv6 Working Group, 2002.

[58] Lee, I.H., Kim, S.J., A QoS Improvement Scheme for Real Time Traffic Using IPv6 Flow Labels. Computational Science and Its Applications (ICCSA2004), Lecture Notes in Computer Science, Volume 3043, pp 278-285, 2004.

[59] Tang, X., Tang, J., Huang, G., Siew, C., QoS Provisioning Using IPv6 Flow Label in the Internet. Fourth IEEE Pacific-Rim Conference on Multimedia ICICS-PCM 2003, p. 1253, 2003.

[60] Jagadeesan, H., Singh, T., A Radical Approach in providing Quality-of-Service over the Internet using the 20-bit IPv6 Flow Label field. IETF Internet Draft, IPv6 Working Group, 2002.

[61] Lin, C., Tseng, P., Hwang, W., End-to-End QoS Provisioning by Flow Label in IPv6. Proceedings of the Joint Conference on Information Sciences, 2006.

[62] Chakravorty, S., Challenges of IPv6 Flow Label Implementation. Proceedings of Military Communications Conference (IEEE MILCOM08), 2008.

[63] Chakravorty, S., Bush, J., Bound, J., IPv6 Label Switching Architecture (6LSA). IETF Internet Draft, 2009.

[64] Demir, S., Özçelik, İ., IPv6 başlığında bulunan akış etiketi alanının kullanım yaklaşımları. Ulusal IPv6 Konferansı, Ankara, 63-69, 2011.

[65] Ahmed, E., Aazam, M., Qayyum, A., Comparison of Various IPv6 Flow Label Formats for End-To-End QoS Provisioning, IEEE International Multitopic Conference (INMIC'09), pp 1-5, 2009.

[66] Fgee, E., B., Kenney, J., D., Phillips, W.J., Robertson, W., Implementing an IPv6 QoS management scheme using flow label & class of service fields. Electrical and Computer Engineering Canadian Conference, Vol: 2, 2, pp: 1049 – 1052, 2004.

[67] Lee, I.H., Kim, S.J., A QoS Improvement Scheme for Real-Time Traffic Using IPv6 Flow Labels. Lecture Notes in Computer Science, Springer-Verlag GmbH, Vol:3043, pp: 278 – 285, 2004.

[68] Fgee, E.B., Phillips, W.J., Robertson, W., Sivakumar, S.C., Implementing QoS capabilities in IPv6 networks and comparison with MPLS and RSVP. Electrical and Computer Engineering Canadian Conference, Vol: 2, pp: 851 – 854, 2003.

[69] Mohamad, I.J., Tat-Chee, W., Alzyoud, F.Y., Sumari, P., Optimizing the MPLS support for real time IPv6-Flows using MPLS-PHS approach. TENCON 2009, IEEE Region 10 Conference, pp 1 – 6, 2009.

[70] Huang, N., Chen, W., RSVP Extensions for Real-Time Services in Hierarchical Mobile IPv6. Mobile Networks and Applications 8, Kluwer Academic Publishers, 625-634, 2003.

[71] Padilla, J.J., Paradells, J, Rodriguez, A., Supporting QoS over IPv6 wireless networks with IntServ6. 17th Annual IEEE International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC'06), 2006.

[72] Cuong, D.H., Guha, D., Choi, J.K., Flow based Forwarding Scheme in Mobile Ipv6Networks to Support for Realtime Services. Network Operations and Management Symposium (NOMS'06), pp 1 – 4, 2006.

[73] Padilla, J.J., Paradells, J., Rodriguez, A., Supporting QoS over IPv6 wireless networks with IntServ6. Personal Indoor and Mobile Radio Communications, pp 1 – 6, 2006.

[74] Tai, W.Y., Tan, C.E., Lau, S.P., Towards utilizing Flow Label IPv6 in Implicit Source Routing for Dynamic Source Routing (DSR) in wireless ad-hoc network. Computers & Informatics (ISCI'12), pp 101-106, 2012.

[75] Tang, X., Tang, J., Huang, G., Kheong, C., QoS provisioning using IPv6 flow label in the Internet Siew Information. Communications and Signal Processing and Pacific Rim Conference on Multimedia. pp 1253 – 1257, 2003.

[76] Wang, Z., Sun, Q., Huang, X., Ma, Y., IPv6 end-to-end QoS provision for heterogeneous networks using flow label. Broadband Network and Multimedia Technology (IC-BNMT'10), pp 130-137, 2010.

[77] Cisco Networking Academy, Routing Protocols Companion Guide, Cisco Press, 2014.

[78] Hummel, S., Routing Protocol Selection Guide. Cisco Support Community, 2013.

[79] Cisco System, OSPF Design Guide. Technology White Paper, Document ID:7039, 2005.

[80] Cisco System, Intermediate System-to-Intermediate System Protocol. White Paper, Document ID:4356, 2004.

[81] Cisco System, BGP Best Path Selection Algorithm. White Paper, Document ID: 13753, 2014.

[82] Cısco System, Redistributing Routing Protocols. White Paper, Document ID:8606, 2012.

[83] Farooq, M.O., Aziz, S., QoS based Distributed Multipath Routing Algorithm for IPv6. Multitopic Conference (INMIC'08), pp 323 – 328, 2008.

[84] Fgee, E.B., Elalo, A., William, J., Elhounie, A., Using Routing Optimization in Next Generation Network to Achieve High QoS. Communication Networks and Services Research Conference (CNSR'10), pp 261 – 267, 2010.

[85] Ashwini, J.P., Sushma, M., Sanjay, H.A., Queuing delay aware path selection algorithm as extension to OSPF, 3rd International Conference on Electronics Computer Technology. Kanyakumari, India, 2011.

[86] Nanda, P., Simmonds, A., A Scalable Architecture Supporting QoS Guarantees Using Traffic Engineering and Policy Based Routing in the Internet. International Journal of Communications, Networks and System Sciences, pp 583-590, 2009.

[87] Cha, S., An EJB based Platform for Policy-Based QoS Management of DiffServ Enabled Next Generation Networks. ICN 2005, LNCS 3420, pp. 794–801, 2005.

[88] Kim, J.Y., Hahm, J.H., Kim, Y.S., Choi, J.K., Policy-based QoS Control Architecture Model using API for Streaming Services. International Conference on Systems and International Conference on Mobile Communications and Learning Technologies (ICNICONSMCL'06), pp. 102, 2006.

[89] Vallejo, A., Zaballos, A., Abella, J., Selga, J.M., Duz, C., Performance of a Policy-Based Management System in IPv6 Networks Using COPS-PR. Sixth International Conference on Networking (ICN'07), pp. 37, 2007.

[90] Lim, S.H., Yaacob, M.H., Phang, K.K., Ling, T.C., Traffic engineering enhancement to QoS-OSPF in DiffServ and MPLS networks. IEE Proceedings – Communications, 151(1): 101-106, 2004.

[91] Ma, Q., Steenkiste, P., Quality of service routing for traffic with performance guarantees. International Workshop on Quality of Service (IWQoS'97), New York, NY, 1997.

[92] Orda, A., Routing with end to end QoS guarantees in broad-band networks, IEEE/ACM Trans. Networking, vol. 7, no. 3, pp. 365–374, 1999.

[93] Sobrinho, J.L., Algebra and algorithms for QoS path computation and Hop-by-Hop routing on the internet. IEEE/ACMTrans. Networking, vol. 10, no. 4, pp. 541–550, 2002.

[94] Shehu, A., Hulaj, A., “The analysis of delays in the network for video and voice applications through OPNET software package” Recent Advances in Circuits, Systems, Telecommunications and Control, Paris, France, 2013.

ÖZGEÇMİŞ

Adı Soyadı : Sadettin DEMİR

ÖĞRENİM DURUMU

Derece	Eğitim Birimi	Mezuniyet Yılı
Doktora	Sakarya Üniversitesi / Fen Bilimleri Enstitüsü / Bilgisayar ve Bilişim Mühendisliği	2021
Yüksek Lisans	Süleyman Demirel Üniversitesi / Fen Bilimleri Enstitüsü / Elektronik ve Haberleşme Mühendisliği	2005
Lisans	Süleyman Demirel Üniversitesi / Mühendislik ve Mimarlık Fakültesi / Elektronik ve Haberleşme Mühendisliği	1999
Önlisans	Akdeniz Üniversitesi / Isparta Meslek Yüksekokulu / Elektronik Bölümü	1991
Lise	Akhisar Lisesi	1989

İŞ DENEYİMİ

Yıl	Yer	Görev
2001-Halen	Süleyman Demirel Üniversitesi	Öğretim Görevlisi

YABANCI DİL

İngilizce

ESERLER (makale, bildiri, proje vb.)

1. Tasarsız Ağlar için bir Güvenlik Simülatörü, 2. Ağ ve Bilgi Güvenliği Sempozyumu (ABG'08), 16-18 Mayıs 2008, Girne, KKTC

2. Uygun, Hızlı ve Verimli FTP Sunucusunun Bulunması, Akademik Bilişim 2009, Harran Üniversitesi, 11-2 Şubat 2009, Şanlıurfa

3. IPv6 Başlığında Bulunan Akış Etiketi Alanının Kullanım Yaklaşımları" IPv6 Konferansı, 12-13 Ocak 2011, Ankara

4. A priority-based queuing model approach using destination parameters for real-time applications on IPv6 networks, Turkish Journal of Electrical Engineering and Computer Sciences 28(2):727 – 742, April 2020

HOBİLER

Cumhuriyet ve Osmanlı dönemi para koleksiyonu