

**AĐ ANAHTARLARINDA OPENFLOW PROTOKOLÜ VE POX DENETLEYİCİSİ
KULLANIMI**

SELÇUK YAZAR

**YÜKSEK LİSANS TEZİ
BİLGİSAYAR MÜHENDİSLİĐİ ANA BİLİM DALI**


2 0 13

**TRAKYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
DANIŞMAN: DOÇ. DR. ERDEM UÇAR**


T.Ü. Fen Bilimleri Enstitüsü onayı

Prof.Dr. Mustafa ÖZCAN
Fen Bilimleri Enstitüsü Müdürü

Bu tezin Yüksek Lisans tezi olarak gerekli şartları sağladığımı onaylarım.


Yrd.Doç.Dr. Tolga SAKALLI
Anabilim Dalı Başkanı

Bu tez tarafımca (tarafımızca) okunmuş, kapsamı ve niteliği açısından bir Yüksek Lisans tezi olarak kabul edilmiştir.


Doç. Dr. Erdem UÇAR
Tez Danışmanı

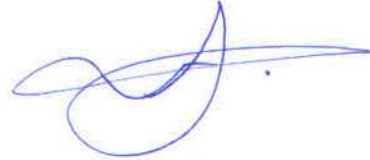
İkinci Tez Danışmanı (varsa)

Bu tez, tarafımızca okunmuş, kapsam ve niteliği açısından Bilgisayar Mühendisliği Anabilim Dalında bir Yüksek lisans tezi olarak oy birliği ile kabul edilmiştir.

Jüri Üyeleri (Ünvan, Ad, Soyad):

İmza

Doç. Dr Erdem UÇAR



Doç. Dr. Yılmaz KILIÇASLAN



Yrd.Doç.Dr. Oğuzhan ERDEM

Tarih: 05/04/2013

T.Ü.FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR YÜKSEK LİSANS PROGRAMI
DOĞRULUK BEYANI

İlgili tezin akademik ve etik kurallara uygun olarak yazıldığını ve kullanılan tüm literatür bilgilerinin kaynak gösterilerek ilgili tezde yer aldığını beyan ederim.

04/04/2013
Selçuk YAZAR



Yüksek Lisans Tezi

Trakya Üniversitesi

Fen Bilimleri Enstitüsü

Bilgisayar Mühendisliği Anabilim Dalı

ÖZET

Bu tez çalışmasında, İstanbul Teknik Üniversitesi Bilgi İşlem Daire Başkanlığı'nda bulunan OpenFlow etkin HP ağ anahtarları kullanılarak, OpenFlow protokolü üzerinde, D-ITG açık kaynak kodlu programıyla birlikte bir araştırma çalışması yapılmıştır. Bu çalışmayla, OpenFlow protokolünün bir controller vasıtasıyla ve controller olmadan performansları ölçülmüş, bahsi geçen POX denetleyicisi (controller) ile ilerde, OpenFlow protokolüne bir katkı yapılıp yapılamayacağı tartışılmak istenmiştir.

Çalışma esnasında 64, 128, 265, 512, 1024, 2048 byte'lık veri paketleri ile 8'er saatlik ağ trafiği oluşturulmuş ve bu çalışma sonucunda elde edilen veriler *GNU PLOT* programıyla görsel hale getirilerek, OpenFlow protokolü ile ilgili bir sonuç çıkarılmaya çalışılmıştır.

Anahtar Kelimeler: OpenFlow, Yazılım Tabanlı Ağ, OpenFlow denetleyicisi.

2013, 55 Sayfa

Master of Science Thesis

Trakya Üniversitesi

Institute of Natural Science

Department Computer Engineering

ABSTRACT

In this thesis, a research has been done on OpenFlow protocol with the open-source D-ITG software by using the Open Flow enabled HP network switch available in Istanbul Technical University, Directorate of Information Technologies.

In this study, the performance of OpenFlow protocol with a controller and without a controller has been measured and it has been intended to be discussed whether a contribution could be made to the OpenFlow protocol with the previously mentioned POX controller.

In the research, an eight-hour network traffic has been generated with 64, 128, 256, 512, 1024, 2048 bytes data packets and a conclusion concerning the OpenFlow protocol has been tried to be acquired through visualizing the obtained data by using *GNU PLOT* software.

Keywords: OpenFlow, Software Defined Network, OpenFlow Controller.

2013, 55 Pages

TEŞEKKÜRLER

Tez çalışmam için beni bu konuya yönlendiren, karşılaştığım zorlukları bilgi ve tecrübesi ile aşmamda yardımcı olan değerli danışman hocam Doç. Dr. Erdem UÇAR'a teşekkürlerimi sunarım. Ayrıca İstanbul Teknik Üniversitesi Bilgi İşlem Daire Başkanlığı'nda görevli, Sayın Gökhan AKIN ve Sayın Sınmaz KETENCİ'ye çalışmamda faydalanmak üzere sağladıkları teknik altyapı ve bilgileri için teşekkür ederim.

Tez çalışmam boyunca beni yalnız bırakmayan değerli aileme sonsuz sevgi ve saygılarımı sunarım.

İÇİNDEKİLER

ÖZET	I
ABSTRACT	II
TEŞEKKÜRLER	III
İÇİNDEKİLER	IV
KISALTMALAR	VI
ŞEKİLLER	VIII
TABLolar	IX
BÖLÜM 1	1
GİRİŞ	1
BÖLÜM 2	3
KAYNAK ARAŞTIRMASI	3
BÖLÜM 3	5
1 MATERYAL VE YÖNTEM	5
1.1 Ağ (Network) Kavramı	5
1.2 Ağ Protokolleri	6
1.3 OSI REFERANS MODELİ	6
1.3.1 Giriş	6
1.3.2 OSI Referans Modeli Katmanları	8
1.4 TCP/IP	10
1.4.1 TCP/IP Referans Modeli	10
1.4.2 OSI ve TCP arasındaki ilişki	10
1.4.3 Katmanlar (Layers)	11
1.5 ANAHTARLAMA, YÖNLENDİRME ve YÖNLENDİRİCİLER	12
1.5.1 Anahtarlama	12
1.5.2 Yönlendirme	13
1.6 OPENFLOW	14
1.6.1 OPENFLOW NEDİR?	14
1.7 OPENFLOW ANAHTARI	15
1.8 Flow ve Flow tablosu (Flow-table)	18
1.8.1 Flow	18
1.8.2 Flow tablosu	18

1.8.3	OpenFlow ağ anahtarı Eylemleri.....	22
1.9	POX Denetleyicisi.....	23
1.10	TEST ORTAMI	23
1.10.1	Ağ yapısı	23
1.10.2	Ağ Trafığı oluşturma yazılımı.....	25
1.10.3	HP E5406 zl Ağ Anahtarı	26
1.10.4	PYTHON	27
2	TARTIŞMA VE SONUÇ	28
2.1	Bit Oranı (Bitrate) sonuçları	28
2.2	Gecikme (Delay) sonuçları	29
2.3	Ağ gecikmesi dağılımı (Jitter) sonuçları	30
2.4	SONUÇ	32
2.4.1	64 byte'lık paketler için sonuçların değerlendirilmesi	32
2.4.2	128 byte'lık paketler için sonuçların değerlendirilmesi	34
2.4.3	512 byte'lık paketler için sonuçların değerlendirilmesi	35
2.4.4	1024 byte'lık paketler için sonuçların değerlendirilmesi	37
2.4.5	2048 byte'lık paketler için sonuçların değerlendirilmesi	38
2.4.6	Genel Değerlendirme	40
3	KAYNAKLAR	41
4	ÖZGEÇMİŞ	45

KISALTMALAR

4G: 4th Generation

LAN : Local Area Network

WAN : Wide Area Network

MAN : Metropolitan Area Network

DSL : Digital Subscriber Line

UTP : Unshielded Twisted Pair

OSI : Open Systems Interconnection

ISO : International Organization for Standardization

PDU : Protokol Data Unit

JPEG : Joint Photographic Experts Group

MP3 : Moving Picture Experts Group Layer-3 Audio

MPEG : Moving Picture Experts Group

TCP : Transmission Control Protocol

UDP : User Datagram Protocol

MAC : Media Access Control

LLC : Logical Link Control

TCP/IP : Transmission Control Protocol/Internet Protocol

NIC : Network Interface Card

STP : Spanning Tree Protocol

NetFPGA : Network Field-programmable Gate Array

FPGA : Field-programmable Gate Array

CPU : Central Processing Unit

TCAM : Ternary Content Addressable Memory

UNIX : Uniplexed Information and Computing System

LINUX : Linus Torvald's UNIX

ICMP : Internet Control Message Protocol

ARP : Address Resolution Protocol

POX : Pythonic Network Operating System

NOX : Network Operating System.

ŞEKİLLER

Şekil 3-1 OSI Referans Modeli katmanları ve protokol veri birimleri	7
Şekil 3-2 OSI Referans Modeli katmanları ve TCP/IP katmanları	11
Şekil 3-3 OSI Referans Modeli katmanları ve cihazlar	14
Şekil 3-4 Ağ anahtarının genel yapısı	16
Şekil 3-5 OpenFlow ağ anahtarı genel yapısı	17
Şekil 3-6 Flow tablosu	18
Şekil 3-7 Flow başlık verisinin detayları	20
Şekil 3-8 Flow tablosundaki verilerle paketin karşılaştırılması. [36]	22
Şekil 3-9 (a) Dedicated OpenFlow ağ anahtarı, (b) OpenFlow protokolü etkin ağ anahtarı.....	22
Şekil 3-10 Çalışmada kullanılan ağ yapısı.....	24
Şekil 3-11 D-ITG uygulaması arayüzü	25
Şekil 3-12 OpenFlow destekli ağ anahtarı cihazı [40].....	26
Şekil 4-1 Bit oranı değerleri.....	29
Şekil 4-2 Ağ gecikmesi değerleri.....	30
Şekil 4-3 Ağ gecikmesi dağılımı değerlerinin hesaplanması [43].....	31
Şekil 4-4 Ağ gecikmesi dağılımı değerleri.....	32
Şekil 4-5 64 byte'lık Ağ gecikmesi dağılımı değerlerinin ölçümleri.....	34
Şekil 4-6 128 byte'lık paketler için Ağ gecikmesi dağılımı değerlerinin ölçümleri.....	35
Şekil 4-7 512 byte'lık paketler için Ağ gecikmesi dağılımı değerlerinin ölçümleri.....	37
Şekil 4-8 1024 byte'lık Ağ gecikmesi dağılımı değerlerinin ölçümleri.....	38
Şekil 4-9 2048 byte'lık paketler için Ağ gecikmesi dağılımı değerlerinin dağılımı.....	40

TABLULAR

Tablo 4-1 64 byte'lık paketler için Ağ gecikmesi dağılımı değerleri istatistiği.	33
Tablo 4-2 64 byte'lık paketler için Ağ gecikmesi dağılımı değerleri değerlendirmesi.	33
Tablo 4-3 128 byte'lık paketler için Ağ gecikmesi dağılımı değerleri istatistiği.	34
Tablo 4-4 128 byte'lık paketler için Ağ gecikmesi dağılımı değerleri değerlendirmesi.	35
Tablo 4-5 512 byte'lık paketler için Ağ gecikmesi dağılımı değerleri istatistikleri.	36
Tablo 4-6 512 byte'lık paketler için Ağ gecikmesi dağılımı değerleri değerlendirmesi.	36
Tablo 4-7 1024 byte'lık paketler için Ağ gecikmesi dağılımı değerleri istatistikleri.	38
Tablo 4-8 1024 byte'lık Ağ gecikmesi dağılımı değerleri değerlendirmesi.	38
Tablo 4-9 2048 byte'lık paketler için Ağ gecikmesi dağılımı değerleri istatistikleri.	39
Tablo 4-10 2048 byte'lık paketler için Ağ gecikmesi dağılımı değerleri değerlendirmesi.	39

BÖLÜM 1

GİRİŞ

Yazılım tanımlı ağ (software defined network), ağ yönetim teknolojilerinin gereksinimlerinin hızla artmasıyla daha çok uygulanır hale gelmiştir. OpenFlow protokolü, bu amaca yönelik olarak geliştirilmiş ve üniversiteler ve araştırma enstitüleri gibi kurumlarda, ağ yönetiminin daha çok araştırmaya imkân sağlanması beklenmektedir. OpenFlow protokolü ve POX denetleyicisi kullanarak, orta büyüklükte ağlarda, bu teknolojinin kullanılabilirliği hakkında niceliksel bir bilgi sağlamak, araştırmacı ve kurumların ağ sistemlerinde kullanabilecekleri bileşenler açısından önemlidir.

Bu çalışmada, OpenFlow protokolü ve POX denetleyicisinin performansları incelenmiştir. Farklı coğrafi lokasyonları kapsayan bir ağda yapılan analizlerle, OpenFlow protokolü ve POX denetleyicisi kullanılarak ulaşılan sonuçlar kullanılarak, bu sonuç grupları arasında çoklu karşılaştırmalar yapılarak varyans analizi çalışması yapılmıştır.

2 Eylül 1969'da UCLA üniversitesinde başlayan bilgisayar ağları deneyimi, bugün neredeyse hayalleri zorlayan bir yaygınlıkta ve inanılması güç bir hızda gelişerek, fark edilmese de günlük hayatın tüm alanını sarmış durumdadır [1].

Yaklaşık 50 yıl önce UCLA ve Stanford üniversiteleri arasında gerçekleştirilen ilk erişimden bu yana, bugün artık 4G teknolojisine geçmiş bir dünya ve veri iletişimi Mars'tan Dünya'ya müzik yayımına kadar ilerlemiştir [2].

Bilgisayar ağları artık sadece iki bilgisayar arasındaki veri transferinden çok, cep telefonları, uluslararası uzay istasyonu, otomobiller, akıllı ev sistemleri vb gibi akla gelebilecek insanla iletişim içinde olan her elektronik cihazı kapsar hale gelmiştir.

Birkaç 10 yıl sonra artık kuantum internet çağında girilecek olacağı varsayılırsa ağ iletişimde ve haberleşmede kablo veya radyo dalgalarından da kurtularak fiziksel engeller ortadan kaldırılacaktır. Şüphesiz bu da insanlık önünde yeni ufuklar açacaktır [3].

Günümüzde veri iletişimin hızla artması ve kuruluşların hızla artan bilgiye erişim ihtiyaçları, ağ teknolojilerinde öteden beri sürekli yeni yaklaşımların geliştirilmesi gereksinimini doğurmuştur. Özellikle son yıllarda IPv4 teknolojisinde yaşanan sıkıntılar, IPv6'nın geliştirilmesine olan ihtiyacı doğurmuş ve 2011 - 2012 yıllarında dünyada ve Türkiye'de bu ağ alt yapısına geçişler artarak hızlanmıştır [4].

Bugün servis sağlayıcılar, mobil operatörler, data merkezleri, büyük kuruluşlar ve özellikle araştırma geliştirme yapan üniversitelerde, ağ yazılımları ve ağ yönetim metotları geliştirmeye en çok ihtiyaç duyulan yerlerdir. Ağ iletişimde en büyük yük donanıma binmektedir. Ancak donanım üreticileri, kendi bilgilerini cihazların içine gömdüklerinden, bunlarla ilgili yeni düzenlemeler, geliştirmeler, yaklaşımlar geliştirmek her zaman donanım üreticilerinin elinde kalmıştır. Ayrıca buna imkân olsa bile, ileri düzeyde teknik bilgi ve deneyim gerektirmektedir.

Bu sıkıntılarının yanında günümüzde, sanallaştırma gibi artan trendler ağ yönetim sistemlerinin yönetim zorluklarını da beraberinde getirmektedir. Diğer yandan bulut bilişime olan ilgi ve gereksinimler ağ kontrolünün önemini daha arttırmaktadır.

Ayrıca ağın görsel olarak yönetiminin sağlanması, bu artan yüklerle birlikte daha önemli hale gelmektedir. Ağ yönetim sistemlerinin genel sıkıntısı, donanıma olan bağımlılık ve fiziksel olarak dağınık yapılarda, görsel olarak yönetimin olmamasıdır. Ayrıca gelişen ihtiyaçları ve değişen koşulları anında ağ içine katmanın güçlükleri de bu fiziksel kısıtlamaları arttırmaktadır.

OpenFlow protokolü ile birlikte ağ donanımının kontrol bölümünün dışarıda bir sunucuya aktarılması, ağ yöneticilerine büyük esneklik sağlayacaktır, bu esneklik büyük alanlardaki ağların yönetimlerini, günümüzün bir diğer sorunu olan enerji ihtiyaçları da düşünülerek, donanım ve alt yapıların daha verimli kullanılmasını sağlayacaktır.

BÖLÜM 2

KAYNAK ARAŞTIRMASI

Literatürde OpenFlow teknolojisi ile ilgili kaynakların azlığı nispeten yeni bir teknoloji olduğu için dikkat çekmektedir.

OpenFlow teknolojisinin temelleri ilk olarak 2007 yılında Japonya’ da yayımlanan SIGCOMM dergisinde “Ethane: Taking Control of the Enterprise” başlıklı makale ile duyurulmuştur [5].

Bundan sonra Kasım 2007’de Alpha sürümünden önceki OpenFlow protokolünün özellikleri yayınlanmıştır.

Daha sonra 2008 yılında Amerika’da ANCS konferansında, “Implementing an OpenFlow Switch on the NetFPGA Platform” makalesi ile OpenFlow destekli ağ anahtarı geliştirilmesi ve standartların belirlenmesine yönelik bir yayın yapılarak, bu protokolün, endüstri tarafındaki ilerlemesine katkısı olacak temeller atılmıştır [6].

2008 yılının Mart ayında, “OpenFlow: Enabling Innovation in Campus Networks” makalesi ile gerçek dünya uygulamaları hayata geçirilmiş, geliştiriciler için yazılımlar ve sanal ağ uygulamaları geliştirilmiştir.

2009 yılında, yazılım tabanlı bir OpenFlow ağ anahtarı kullanılarak OpenFlow protokol testleri yapılmış, ancak bu çalışmada, harici bir denetleyiciyle (controller) ve gerçek bir donanım tabanlı ağ anahtarı testlerinin yapılamadığı belirtilmiştir [7].

2010 yılında, Brezilya Amazon federal üniversitesinden Rodrigo Braga, “Lightweight DDoS Flooding Attack Detection Using NOX/OpenFlow”, isimli bildirisinde, NOX denetleyicisi ve OpenFlow protokolünü kullanarak, ağ güvenliği konusunda bir yayın yapılmış, burada NOX denetleyicisinin bu yönde bir performans ölçümünü yapmıştır [8]. OpenFlow protokolünde birbirinden ayrılan, flow yönlendirme

işlemi için de, Linux tabanlı bir bilgisayarda bulunan OpenFlow sanal ağ anahtarının performans ölçümleri, İtalya Politecnico di Torino Üniversitesi elektronik bölümünde bir makale ile duyurulmuştur [9]. Ayrıca aynı yıl Cornell ve Princeton Üniversitelerinde yapılan ortak bir çalışmayla, OpenFlow kontrolündeki ağlar için “*Frenetic*” isimli yüksek seviye bir programlama diline ait çalışmalar yayınlanmıştır [10]. Stanford Üniversitesi ve AT&T Laboratuvarları’nın ortak bir çalışmasında NOX denetleyicisi kullanılarak “*DIFANE:Doing It Fast ANd Easy*” kurumsal ağlarda daha kolay ölçeklenebilir ve yönetilebilir bir flow yönlendirme ve yönetme yapısı hakkındaki çalışma yayınlanmıştır [11]. Toronto Üniversitesi’ndeki araştırmacılar da, NOX denetleyicisini kullanarak, birden fazla denetleyici için dağıtımli bir yönetim sistemi üzerine çalışmalarını yayınlamışlardır [12].

2011 yılında OpenFlow ağ anahtarı özellikleri yayınlanmıştır. OpenFlow teknolojisinin hızla ağ teknolojilerini etkilemesiyle, bu alanın değişik bölümlerinde daha çok ilgi uyandırır olmuştur. NOX denetleyicisi kullanarak, sanal bir sanal yönlendirici servisi “*RouteFlow*” ile ilgili bir çalışma yayınlamıştır [13]. Koç Üniversitesi’nden araştırmacılar da OpenFlow destekli bir ağda görüntü taşınması işlemindeki kalite ile ilgili çalışmalarını yayınlamışlardır [14]. Almanya Wurzburg Üniversitesi’nde NOX denetleyicisi kontrolündeki bir ağdaki performans ve ölçeklenebilirlik sorunu incelenmiş ve sonuçlar Uluslararası 23. Teletraffic konferansında sunulmuştur [15].

2012 yılında, OpenFlow konusu daha çok ağ yönetimi ve büyük coğrafi alanlarda ağ yönetiminin getirdiği sıkıntılara aranan çözüm arayışları üzerine yapılan çalışmalar artış göstermektedir. Brezilyadaki bir çalışmada NOX denetleyicisi kontrolündeki bir ağda domainler arası yönlendirme ilgili bir çalışma yayınlanmıştır [16]. Kablosuz ağ yönetiminde OpenFlow denetleyicileri ve protokolü üzerinde Türkiye’de çalışmalar yayınlanmıştır [17].

Günümüzde, daha çok uygulama alanı üzerinden çalışmaları yürüten bu teknoloji, gelişime oldukça açıktır.

BÖLÜM 3

1 MATERYAL VE YÖNTEM

1.1 Ağ (Network) Kavramı

Bir veya birden fazla bilgisayarın; dosya ve veri alış verişi yapabilmesi veya haberleşmeleri için birbirine bağlanarak oluşturduğu yapılara network (ağ) denir. Ağ topolojisi 3 başlık altında incelenir; *LAN(Local Area Network)*, *WAN(Wide Area Network)* ve *MAN(Metropolitan Area Network)* olmak üzere yapılarına göre standartları vardır:

Local Area Network (LAN): Birden fazla bilgisayarın oluşturmuş olduğu en küçük bilgisayar ağlarına *LAN* adı verilir. Günümüzdeki en hızlı ağ yapısıdır ve yaygın olarak Ethernet Protokolü kullanılmaktadır. İçerisinde bir veya birden fazla Ethernet anahtarlama (switch) cihazı bulundurlar. *LAN* üzerindeki bilgisayarlar birbirlerine coğrafi olarak yakındır ve 10, 100, 1000 ve 10000 *Mbit* hız ile haberleşmektedirler.

Wide Area Network (WAN): Hız olarak *LAN* ağ yapısına göre yavaş olmasına rağmen, en az iki *LAN*'ın yönlendirici (router) ile birleşmesi sonucu kurulan ağlara *WAN* denir. Şehirlerarası ve uzak mesafeler için tasarlanmış bir ağ yapısıdır.

Metropolitan Area Network (MAN): Aynı şehir içinde bakır veya Fiber Optik altyapılar ile oluşturulan ağlara verilen isimdir. Günümüzde yaygın olarak MetroEthernet, *DSL* ve Kablonet teknolojileri bu amaçla kullanılmaktadır.

Bir network kurmak veya birden fazla bilgisayarı birbiri ile bağlamak kullanıcılarına şu faydaları sağlar:

- Dosya Paylaşımı
- Düşük Maliyet

- Çevre Cihazlarının Paylaşımı
- Düşük Maliyet İle Haberleşme
- Daha Kolay Yönetilebilirlik [18]

Ağların kurulumu gibi, bu ağların yönetimi de çok önemlidir. Özellikle bu konuda alanında uzman kişiler her zaman aranan insanlardır. Kişilerin haberleşmesi, dosya paylaşımı ve ortak uygulama ortamı beraberinde ağ güvenliği ile ilgili tehlikeleri de getirir. İnternetin yayılması ile günümüzde sanal suçlar olarak nitelendirilen bu tehlikeler daha da çoğalmıştır. Bu nedenle sistemlerin daha hızlı takip edilmesi ve ağ güvenliğinin sağlanması gerekmektedir.

1.2 Ağ Protokolleri

Ağ protokolleri verilerin nasıl paketleneceğini, kullanılacağını ve ağdan iletileceğini belirten anlaşmalardır. Satıcılar ve endüstriyel komiteler bu anlaşmaları geliştirirler ve firmalar bunlara uyan yazılımlar yazmaya çalışırlar. Ağlar heterojen oldukça, değişik sistemlerin kontrolü ve yönetimi daha zorlaşır. Denenmiş ve doğrulanmış standartların olmasına rağmen, karma ağları yönetmek hala karmaşık bir iştir. Ağlar verileri güvenilir şekilde kullanmak için protokollere ihtiyaç duyarlar. Ağ protokolleri kullanıcılara görünmese de, protokol mimarisi bir *LAN* veya *WAN* planlanırken ve kurarken seçilmesi gereken en önemli parçalardan biridir.

1.3 OSI REFERANS MODELİ

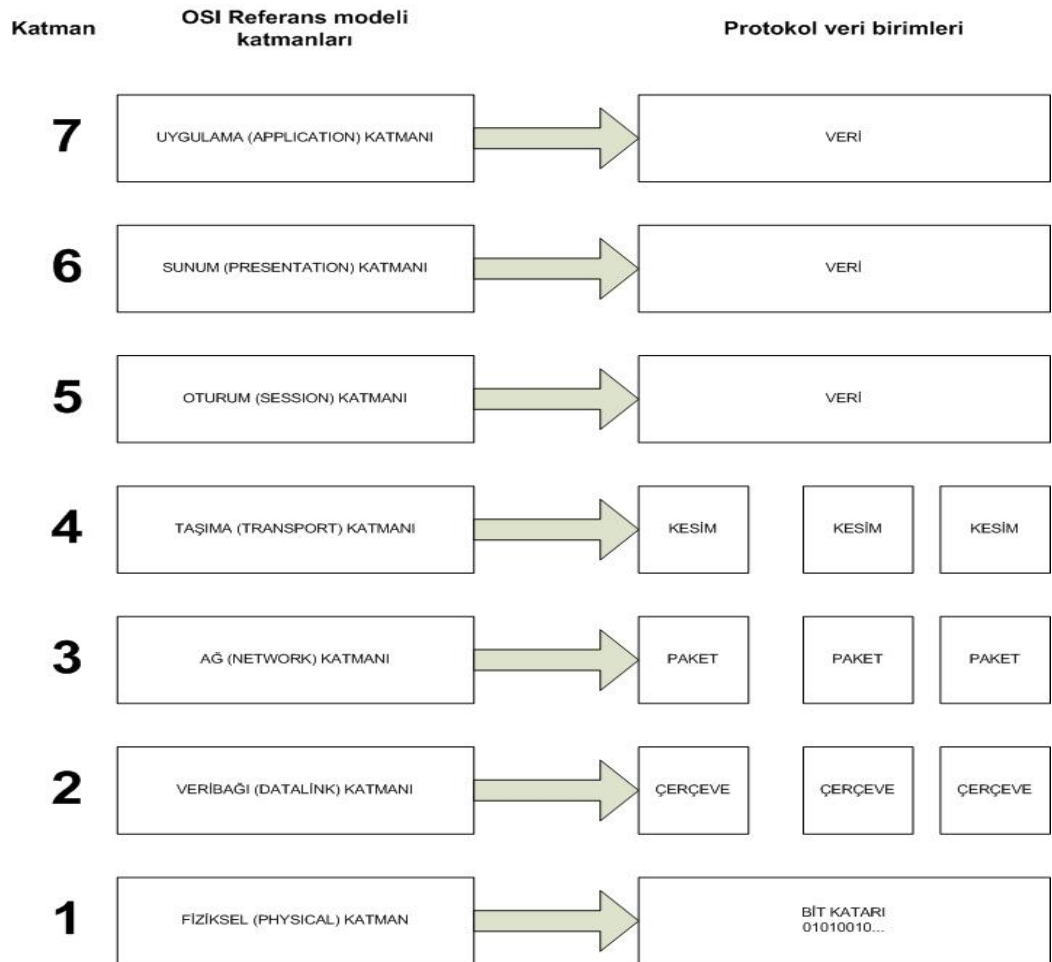
1.3.1 Giriş

OSI Referans Modeli, ağ protokollerinin tasarımı konusunda bir yol gösterici olarak *ISO (International Organization for Standardization)* tarafından geliştirilmiştir. *OSI* referans modeli ağ iletişim sürecini, her birinin kendisine özgü protokoller ve görevler içerdiği 7 farklı mantıksal katmanda inceler [19].

OSI referans modelinin orijinal adı *Open Systems Interconnection-Basic Reference Model*'dir. *ISO* kendi web sitesinde de belirttiği üzere *OSI* referans modeli, çeşitli sistemlerin birbirleri ile uyumlu çalışması üzerine geliştirilecek standartlar için

ortak zemin sağlanmasını ve mevcut standartların referans modeli üzerinden incelenmesini sağlar. *OSI* Referans Modeli standartlar geliştirilmesi veya iyileştirilmesi gereken alanları tanımlar. Uygulama aşamasındaki teknik şartlarını tanımlamaz. İlk olarak 1984 yılında kabul edilen *OSI* Referans Modeli, 1994 yılında yeniden düzenlenmiştir [20].

OSI Referans Modeli hiyerarşik bir yapıya sahip yedi katmandan oluşur. Her katman kendisinden bir üstündeki katmana ilgili servisleri sunmaktadır. Yedi katmanın her birinde, veri ilgili katmana özgü standartlar ve protokoller tarafından tanımlanmış özelliklerde birimlere dönüştürülürler. Her katmanda farklı bir protokol veri birimi (*Protocol Data Unit - PDU*) olarak kullanılır.



Şekil 1-1 OSI Referans Modeli katmanları ve protokol veri birimleri

1.3.2 OSI Referans Modeli Katmanları

OSI Referans Modeli katmanlarının işlevlerine aşağıda kısaca değinilmiştir.

Uygulama Katmanı: *OSI* Referans Modelinin en üst katmanıdır. Uygulama Programlarının ağa erişimi için gerekli işlevleri kapsar ve diğer katmanlardan farklı olarak hizmet verdiği başka bir üst katman bulunmakla birlikte uygulama süreçlerine doğrudan hizmet veren tek katmandır [21]. Kullanıcının etkileşimde bulunduğu uygulama programları direkt bu katmanla iletişim içindedir.

Sunum Katmanı: Veri haberleşmesi yapacak uygulamalar için verinin uygun sunum biçimini tanımlayan (veri transfer sentaksını) ve uygulama katmanına hizmet eden katmandır. Haberleşmenin gerçekleşebilmesi için kaynak ve hedef cihazda çalışan uygulamalar ortak veri transfer sentaksını seçerler. Bu katman uygulama katmanından gelen verinin içerdiği bilgilerin değişmeden bütünlüğünü korumasını sağlayarak oturma katmanına erişim sağlar. Verinin kaynaktan sıkıştırılıp (compression) hedefte sıkıştırılmış veri açılmasından (decompression) da sorumludur. Benzer şekilde şifreleme ve şifre çözme işlemleri bu katmanda yapılabilir [22]. Bu katman için *JPEG*, *MP3*, *MPEG* gibi görüntü, ses ve video formatları örnek gösterilebilir.

Oturum Katmanı: Uç düğümler arasında gerekli oturumun kurulması, yönetilmesi ve sonlandırılması işlerini kapsar. İletişimin mantıksal sürekliliğinin sağlanması için, iletişimin kopması durumunda bir eşleme noktasından başlayarak iletişimin kaldığı yerden devam etmesini sağlar.

Taşıma Katmanı: Bu katmanın ana görevi bir cihaz üzerinde aynı anda birden fazla uygulamanın ağ üzerinden haberleşmesini sağlamaktır. Adresleme yapısı olarak her uygulama için kaynak ve hedefte taşıma katmanında port numaraları kullanılır. Bilginin son alıcıda her tür hatadan arındırılmış olarak elde edilebilmesini sağlar. Ulaşım katmanının oluşturduğu bilgi bloklarına bölüm (segment) denir. Bunlar son alıcıya sırası bozulmuş olarak gelirse, düzgün olarak sıralanmalıdırlar. Bunun için bölümler numaralanır. *OSI* başvuru modelinin ilk üç katmanı ve 4 katmanın bir kısmı özellikleri ağ arabağlaşım cihazlarında kullanılan işlevleri yerine getirirler. İlk üç katman ise uygulama programları düzeyinde hizmetleri sunan işlevlere sahiptirler. En

yaygın kullanılan bağlantı odaklı taşıma katmanı protokolü *Transmission Control Protocol (TCP)*'dür. Bağlantısız taşıma katmanı protokollerinde ise taşınacak verilere en az ek yük getirecek biçimde verilerin hızlıca hedefe iletilmesi esas alınır. *User Datagram Protocol (UDP)* en yaygın kullanılan bağlantısız taşıma katmanı protokollerindendir. Bu katmana özgü *PDU*'lar kesim (segment) ismini alır.

Ağ Katmanı: Veri paketlerinin bir uçtan diğer uca ağdaki çeşitli düğümler (yönlendirici, geçityolu, vs.) üzerinden geçirilip yönlendirilerek alıcısına ulaşmasını sağlayan işlemlere sahiptir. Veri paketinin alıcısına giderken ağ koşullarına, önceliklere ve diğer parametrelere göre hangi yolun uygun olacağı bu katmanda değerlendirilir. Bu amaçla düğümlere ağ adresi denilen numaralar verilir. Ağ adresi taşıyan bilgi bloklarına paket adı verilmektedir. İnternet' in temel protokol kümesi olan TCP/IP'nin IP protokolü bu katmanda yürütülen bir protokoldür.

Veri Bağı Katmanı: Gönderilecek bilginin hatalara bağışık bir yapıda lojik işaretlere dönüştürülmesi, alıcıda hataların sezilmesi, düzeltilemiyorsa doğrusunun elde edilmesi için göndericinin uyarılması gibi işlevleri vardır. Gönderilen/alınan lojik işaret bloklarına çerçeve (frame) denir. Çerçevelerin içerdiği bit sayısının alt ve üst sınırları standartlarla belirlenmiştir ve genellikle değişken uzunluktadırlar (*HDLC, SDLC, vs.*). OpenFlow bu katmanda yer alır [23].

Fiziksel Katman: *OSI* Referans Modelinin en alt katmanı olan fiziksel katman verilerin farklı iletim ortamlarında nasıl iletileceğini tanımlar. Veri bağı katmanı çerçevelerini temsil eden ikili sayıların (binary digit-bit) fiziksel iletim ortamında sinyaller olarak iletilebilmesine uygun şekilde kodlanmasından sorumludur. Bu katmana özgü *PDU*'lar bit katarı ismini alır [24].

Bilgisayar ağlarınınve uygulamalarının tasarımı konusunda *OSI* Referans Modeli'nin katmanlı yapısının avantajları olarak aşağıdaki örnekler gösterilebilir.

i. Karmaşıklığı azaltır.

Ağ iletişimini sürecini küçük ve basit parçalara böler. Bu da, bileşen geliştirme ve sorun çözme ve tasarlama aşamalarında kolaylık sağlar.

ii. Arayüzleri standartlaştırır.

Birçok üreticinin standartlara uygun geliştirme sağlamasını sağlar ve rekabeti artırır.

iii. Modüler mühendislik kolaylaştırılır.

Çok farklı ağ cihazları ve teknolojinin birbirleriyle haberleşmesini kolaylaştırır.

iv. Birbiriyle uyumlu teknoloji üretimi sağlar

Bir katmanın diğer katmanın üzerinden teknolojik değişiklik yapmasını önler [25].

1.4 TCP/IP

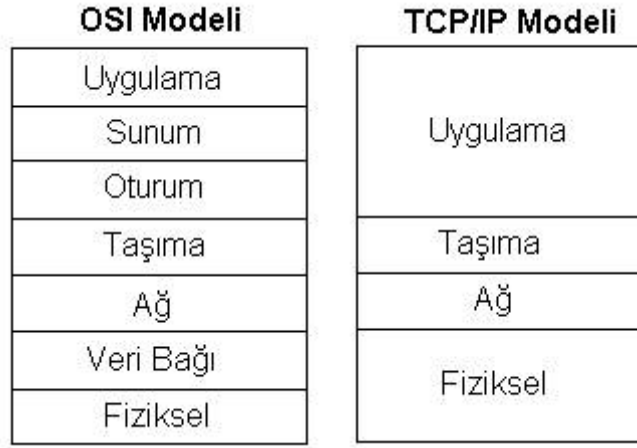
1.4.1 TCP/IP Referans Modeli

TCP/IP adından anlaşılacağı gibi, arkasında iki adet standardı barındıran internet protokolleri bütünüdür ve birbirine bağlı bilgisayar kümelerinin birbirleriyle haberleşmesi amacıyla kullanılabilir. OpenFlow teknolojisinin temellerinde yatan, Flow bilgilerinin ve flow tablolarının anlaşılabilmesi için, *TCP/IP* protokollerinin barındırdığı bileşenlerin anlaşılması gerekmektedir. OpenFlow temelde switch ve controller arasındaki iletişim protokolü olsa da, bu iletişim *TCP/IP* ile sağlanmaktadır.

TCP (Transmission Control Protocol) ve *IP (Internet protocol)* internet iletişiminin en temel iki bileşenidir [16]. Bu yüzden günümüzde internet iletişimi dendiğinde bu iki protokol akla gelmektedir.

1.4.2 OSI ve TCP arasındaki ilişki

TCP/IP'nin açılımı *Transmission Control Protocol/Internet Protocol*'dür. *TCP/IP* modeli *OSI* katmanlarından çok daha önce standartlaştığı için *OSI* içinde referans olmuş 4 katmanlı bir yapıdır [27].



Şekil 1-2 OSI Referans Modeli katmanları ve TCP/IP katmanları

1.4.3 Katmanlar (Layers)

TCP/IP uçtan uca bağlantıda iletilen verilerin ne şekilde formatlanacağı, adresleneceği, yönlendirileceği, iletileceği ve alınacağını belirler. *TCP/IP* kendi protokolleri olan 4 adet katmandan oluşmaktadır [28][29].

Uygulama (Application) katmanı; Bilgisayar ağları üzerinden gelen mesajların ya da gönderilecek mesajların gitmesi gereken yere ulaştığında son kullanıcıların bu mesajları herhangi bir sorunla karşılaşmaksızın görebilmesi için gerekli dönüştürme sürecinin gerçekleştiği katmandır. (HTTP, FTP, SNMP, vb.)

TCP Katmanı; Bu katmanda kullanıcıların büyük hacimli mesajları, datagram adı verilen küçük parçalara ayrılır ve bu parçaların gitmesi gereken alıcıya gönderilmesi işlemleri gerçekleştirilir. Her bir datagrama mesajın yerine ulaştığında tekrar eski haline getirilebilmesi için gerekli sıralama bilgileri, hatalar oluştuğunda bunları belirleyip düzelmeyi sağlayacak hata sezme bilgileri ve mesajın gitmesi gereken alıcının adres bilgileri gibi veriler içeren başlık ve son ekler eklenir.

Internet katmanı; (IP) Temel olarak bu katman bilgisayar ağları arasında mesajların yönlendirilmesinden sorumludur. Bir üstteki *TCP* katmanından gelen datagramlar bu katmanda paket adı verilen daha küçük kısımlara ayrılır ve her bir pakete gideceği bilgisayarın bulunduğu bilgisayar ağının adresi eklenir.

Link (Network Interface) katmanı; (genel olarak Ethernet olarak bilinir) OpenFlow protokolü bu katmanda yer almaktadır.

1.5 ANAHTARLAMA, YÖNLENDİRME ve YÖNLENDİRİCİLER

Farklı 3. katman ortak ağ adreslemesine sahip yerel alan ağları arasında iletişimin sağlanabilmesi için yerel alan ağının çıkış noktasında trafiğin uygun bir şekilde hedef ağa yönlendirilmesi gerekmektedir. Yönlendirme işlemi en az bir arayüzü ile yerel alan ağına, farklı bir arayüzü ile dış ağa erişim sağlayan yönlendirici(router) cihazları yapmaktadır. Yönlendirici cihazlarının temel görevi aldıkları her bir paketin hedefi için üçüncü katman adresine göre en iyi yolu belirlemek ve paketleri hedefi doğrultusunda iletmektir. Her yönlendirici en iyi yolun belirlenmesinde kendi üzerinde tuttuğu yönlendirme tablosunu kullanır.

1.5.1 Anahtarlama

Layer-2 anahtarlama (switching), donanım bazlı bir filtreleme yöntemidir ve bu yöntemde trafiği filtrelemek için ağ kartlarının *MAC* adresleri kullanılır. Layer-2 anahtarlama, filtreleme için Network katmanı bilgilerinin yerine çerçevelerdeki *MAC* adreslerini kullandığı için hızlı bir yöntemdir. Layer-2 anahtarlama kullanmanın en önemli sebebi, ağ çarpışma etki alanlarına(collision domain) bölmektir. Böylece ağ ortamı daha verimli kullanılmış olur. Ağ anahtarı kullanılarak ağ ortam segmentlere bölünebilir. Böylece ağdaki çarpışma etki alanı sayısı arttırıldığında çarpışma ve karışıklıklar azaltılmış olur. Fakat ağ anahtarı kullanılarak yapılan segmentasyon işleminden sonra bile mevcut ağ tek bir Yayın Etki Alanı (broadcast domain) olarak kalır. Yani yayınlanan mesajlar ağın tamamını etkiler. Eğer ağ birden fazla yayın etki alanına (broadcast domain) bölünmek istenirse segmentasyon işlemi için yönlendirici (router) kullanılmalıdır.

Layer-2 anahtarlamanın üç önemli fonksiyonu vardır. Bunlar sırasıyla;

Adres Öğrenme:

OSI Referans Modeli'nin üçüncü katmanındaki adresleme mantıksal adresleme olarak adlandırılmaktadır. Bir yerel alan ağdaki kullanıcılara/cihazlara (Şekil 3.3)

ağda kullanılan üçüncü katman protokolüne uygun, ortak ağ adreslemesine sahip adreslerin atanması ile üçüncü katmanda yerel alan ağlar tanımlanmış olur.

Layer-2 anahtarlar (switch) ve köprüler (bridge), her bir arayüzlerinden aldıkları çerçevelerin kaynak adreslerini işleyerek bu adresleri kendi *MAC* tablolarına kaydederler.

İletme/Filtreleme Kararı: Ağ anahtarı, arayüzlerinden aldığı her bir çerçevenin hedef adresine bakar ve bünyesinde bulundurduğu *MAC* veritabanına bakarak bu çerçevenin hangi arayüzünden çıkarılacağına karar verir [30].

Döngüden Kaçınma: Eğer ağdaki anahtarlar arasında birden fazla bağlantı varsa, bu anahtarlar arasında bir döngü ağ oluşabilir. Bunu engellemek için *STP* (Spanning Tree Protocol) protokolü kullanılır.

STP (Spanning Tree Protocol)

STP protokolü birden fazla link üzerinden birbirine bağlanmış ağ anahtarları arasında bir ağ döngüsü olmasını engeller. Bunun için, kullanılan yedek linkleri kapatır. Yani *STP* ağdaki tüm linkleri bularak bu linklerin yedek olanlarını kapatıp döngü oluşmasını engeller.

Bunu gerçekleştirmek için ağ üzerindeki anahtarlardan bir tanesi “root bridge” olarak seçilir. Bu anahtarın portları da “designated port” olarak adlandırılır. Bu portlar üzerinden trafik alış verişi olur. Ağdaki diğer anahtarlar ise “nonroot bridge” olarak adlandırılır. Root switch, ağ üzerinde daha düşük öncelikli *ID*'ye ve *MAC* adresine sahip olan ağ anahtarı olur. Root switch'in dışındaki anahtarlar kendileri ile root switch arasındaki en düşük tüketim değerine sahip yolu seçerler. Bu yolun haricindeki diğer yollar yedek olarak kalır ve birinci yol aktif olduğu müddetçe bu yollar kullanılmaz. *STP* protokolü, *BPDU* (Bridge Protocol Data Unit) tipinde çerçeveler kullanır [31].

1.5.2 Yönlendirme

IP Yönlendirmesi *OSI* referans modelinde Ağ Katmanı tarafından yapılır. Bu demektir ki, Ağ katmanı ara hostlar dahil, uçtan uça (kaynaktan hedefe) yönlendirme

dahil paket dağılımdan sorumlu iken, Veri Bağı Katmanı (DataLink Layer) aynı bağlantı üstünde, düğümler arası çerçeve dağıtımından sorumludur.

Ağ Katmanı, hizmet kalitesi ve hata kontrol fonksiyonları sağlayarak, kaynaktan hedefe, birden fazla ağ içinde verilerin işlevsel ve yordamsal aktarımını ifade eder.

Ağ katmanının sağladığı bazı fonksiyonlar Mantıksal adresleme, Yönlendirme, Datagram Kapsüllemesi, Parçalama Yeniden Birleştirme, Hata Yakalama ve Hata Teşhisidir [7].

Katman #	Name	Kapsülleme Birimi	Cihazlar ve Bileşenler
7	Uygulama	veri	PC
6	Sunum	veri	
5	Oturum	veri	
4	İletim	segment	
3	Ağ	paket	router(yönlendirici)
2	Veri Bağı	çerçeve	Köprü(bridge), ağ anahtarı, Ağ Kartı (NIC)
1	Fiziksel	bit katarı	yönlendirici, hub

Şekil 1-3 OSI Referans Modeli katmanları ve cihazlar

1.6 OPENFLOW

1.6.1 OPENFLOW NEDİR?

OpenFlow yenilikçi ağ protokollerini ve yeni yöntemlerin kullanımındaki ağlarda geliştirilmesini sağlayan bir açık standarttır [32]. Yazılım tabanlı ağ fikri, uzun süredir araştırmacıların kafasını kurcalayan bir problem olup buradaki temel problemler, günümüzdeki ağ bileşenlerinin, Üç bileşenli yapısının birbirinden ayrılması problemi olarak belirtilebilir. Bu üç bileşen aşağıdaki gibidir:

Yönetim Birimi (Management Plane)

İletim Birimi (Forwarding Plane)

Kontrol Birimi (Control)

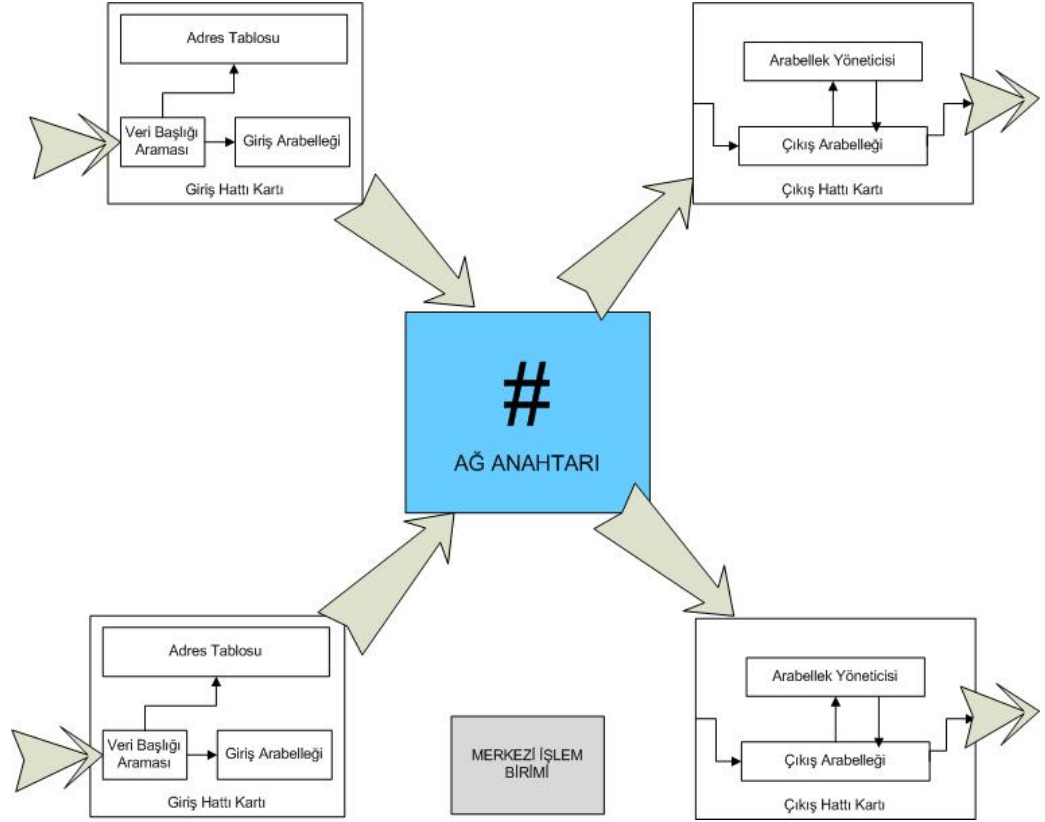
Bu katmanların birbirinden ayrılması için ilk kullanılan yöntemlerden biri NetFPGA isimli düşük bütçeli bir *PCI* karttır. Bu ürün programlanabilir bir FPGA bileşeni içeren bir üründür, ancak yine de bu cihazların gelişme açık yönleri oldukça sınırlıdır [33].

OpenFlow standardı tüm bu gelişim içinde, tüm eksikleri karşılayacak bir çözüm olarak doğmuştur. Yazılım tabanlı bir ağ olarak OpenFlow aynı zamanda sistem yöneticilerine ağdaki topoloji ve paket filtreleme değişikliklerinde yönetim imkânı sunan bir yetenek kazandırmaktadır [34].

OpenFlow içerisindeki en önemli bileşen denetleyici (controller) ve OpenFlow ağ anahtarlarıdır. Klasik olarak ağ anahtarı ve yönlendiricilerde paket iletimi (Data Path) ve yüksek seviye yönlendirme karar mekanizması (Control Path) aynı cihazda bulunur. OpenFlow ise, Data Path bileşenini yine fiziksel ortamda bırakarak Control Path bileşenini bir sunucuya taşımıştır [35].

1.7 OPENFLOW ANAHTARI

Bugün, kullanılan ağ anahtarlarının işlevi, yerel ağlardan oluşmuş kümeleri birbirine bağlamak olarak özetlenebilir. Genel yapısı Şekil (3.4)'teki gibidir.



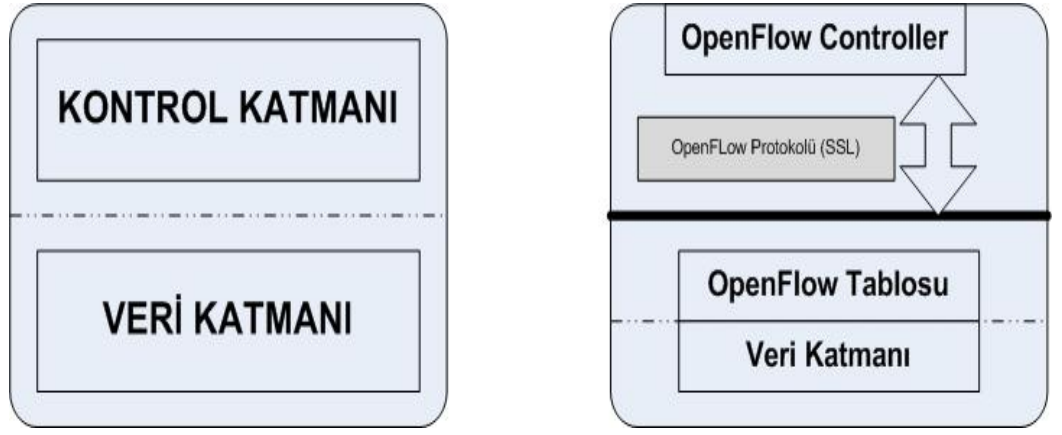
Şekil 1-4 Ağ anahtarının genel yapısı

Ağ anahtarlarındaki hat kartları, verinin ağ anahtarına girdiği ve ağ anahtarından çıktığı yerlerdir. Hat kartları, bilgisayar ağını bağlantı örgüsüne bağlar. Elektriksel-optik çevrim, seri-paralel dönüşümü, senkronizasyon, çerçeve işleme (frame processing) gibi fiziksel işlemler hat kartlarında bulunmaktadır. Bağlantı örgüsü, giriş ve çıkış hatlarını birbirine bağlar ve anahtarlama işini üstlenir. Anahtarlama sonrası, paketler çıkış hat kartlarının tampon belleğinde kuyruklanır. Çıkış hattına gönderilecek paket, çıkış hat kartındaki hizmet kalitesi çizelgeleyici tarafından seçilir. Hizmet kalitesi çizelgeleyicinin görevi, farklı trafik kaynaklarına hizmet ayrıcalıkları ve öncelikler sağlamaktır. *CPU*, adres tablosunun güncellenmesi, paket istatistiklerinin toplanması gibi genel yönetim ve bakım görevlerini üstlenmiştir [36].

Daha genel anlamda bakıldığında ağ anahtarı yapısı iki bölüme ayrılabilir, Veri katı ve Kontrol katı. Veri katı, ağ anahtarına gelen ve yönlendirilecek flow'ların tutulduğu kısımdır. Bu flow'ların işlenmesi kontrol katının görevidir. Günümüzde bir

şasi içinde bulunan bu yapı, üreticileri için dışarıdan müdahaleye izin vermeyen bir şekildedir.

Bu iki katman ile ilgili olarak daha esnek bir yapıya geçmek için OpenFlow switchler tasarlanmıştır. Böylece kullanılan ağ anahtarlarında büyük değişiklikler yapılmış ve veri katmanı ile kontrol katmanı birbirinden ayrılmıştır. Bu da veri katmanının bir güvenli bağlantı üzerinden dışarıdan kontrol edilebilmesini getirmiştir.



Şekil 1-5 OpenFlow ağ anahtarı genel yapısı

OpenFlow anahtarının ana fikri Kontrol katmanını ağ anahtarının dışına almaktır.

İki tip OpenFlow ağ anahtarı mevcuttur. Birincisi donanım bazlı olup ticari ağ anahtarlarıdır ve TCAM ile Flow tablosu ve OpenFlow protokolünü kullanırlar. Diğer tip ağ anahtarları ise yazılım tabanlı olup OpenFlow ağ anahtarı fonksiyonları bir adet UNIX/Linux işletim sistemi kullanarak yerine getirirler.

OpenFlow ağ anahtarı en az üç adet bileşenden oluşmaktadır (Şekil 3.5). Bunlar sırasıyla şunlardır:

Flow tablosu (FlowTable); anahtara bir flow'u nasıl işleyeceğini belirten flow tablosu bileşeni içeren veri bütünü.

Güvenli bağlantı; anahtarın kullandığı komutlara ait paketlerin controller arasında haberleşmede kullanılır.

OpenFlow protokolü; anahtarla ve controller'la, açık ve standart bir iletişim olanağı sunan yöntem.

OpenFlow ağ anahtarındaki flow table girdileri üzerinde kontrol imkânı sağlamaktadır.

1.8 Flow ve Flow tablosu (Flow-table)

1.8.1 Flow

Flow tanım olarak; aynı kaynak IP'si, hedef IP'si, kaynak portu, hedef portu, protokol demet öğelerini paylaşan ve ayarlanabilen bir D zaman aşımı süresiyle ayrılmış tek yönlü paketler serisidir [37].

1.8.2 Flow tablosu

Flow tabloları, ağ anahtarları, yönlendiriciler, ateş duvarları gibi, günümüzdeki ağ bileşenlerinin çoğunda flow bileşenlerinin saklandığı bir yapıdır. Ağ anahtarlarında ve yönlendirici cihazların içinde veri –satıhı (data-plane) içinde tutulmaktadır.

FLOW TABLOSU		
Başlık	Eylemler	Sayaçlar
Başlık	Eylemler	Sayaçlar
.	.	.
.	.	.
.	.	.
Başlık	Eylemler	Sayaçlar

Şekil 1-6 Flow tablosu

OpenFlow ağ anahtarlarında bulunan flow tabloları 3 adet temel bileşenden oluşmaktadır.

- **Flow'u tanımlayan paketin başlığı**
- **Paket in nasıl işleneceğini belirten bölüm (action)**
- **Flow paketlerinin, boyutu, izleme numarası ve zaman aşımı bilgilerinin bulunduğu istatistik veri bölümü.**

Flow tablosu içeriğinde flow'ları tanımlamak için iki adet tablo bulunmaktadır. İlk tablo lineer bir tablo olup flow'ları tanımlayan joker karakterler içermektedir. Bu tablodaki joker kayıt girdileri kayıt içindeki bir flow için sadece bazı karakterleri tanımlar (*MAC* adresi, *IP* adresi ve *Port* gibi). Bu ilk tablo da 100 joker flow kaydı içerir. OpenFlow ağ anahtarı bu tablo içindeki başlık alanlarıyla, gelen flow'un başlık alanlarını karşılaştırır.

İkinci tablo ise, tam bir karşılaştırma tablosudur ve *HASH* algoritması kullanarak bu kaydı girdilerini saklar ve içlerinde karşılaştırma ve arama yapabilir. Boyutu ise 131072 adet tam karşılaştırmalı flow girdisi kadardır. Tam karşılaştırmalı girdi ise bir flow'un tüm olası içeriğini tanımlar. Bunlar aşağıda sıralanmıştır:

- Giriş portu
- Kaynak / hedef MAC adresleri
- Ethernet protokolü
- Kaynak/hedef IP adresi
- Ağ protokolü
- Kaynak/hedef portu

Ayrıca OpenFlow ağ anahtarı *HASH* tablo verileriyle flow'ları karşılaştırarak işlem yapar.

Başlık	Eylemler	Sayaçlar								
Giriş Portu	Ethernet			VLAN		IP			TCP/UDP	
	Kaynak	Hedef	Tip	Vlan No	Öncelik	Kaynak	Hedef	Protokol	Kaynak	Hedef

Şekil 1-7 Flow başlık verisinin detayları

Flow tablosu için bir doğrusal girdi örneği aşağıdaki gibidir:

```
dpctl add-flow nl:0 dl_type=0x0800, nw_dst=10.0.01, idle_timeout=120,
actions=output:1
```

dpctl komutu, Linux sistemleri üzerinde OpenFlow yönetim sistemi üzerinden kullanılan (DataPath Control) bir komuttur ve denetleyicinin bağlı bulunduğu ağ anahtarlarına flow kaydı eklemek için kullanılabilir.

Flow tablosundaki bir tam-eşleşme örneğine ise,

```
dpctl add-flow nl:0 in_port=0, dl_vlan=0xffff, dl_src=00:08:74:2F:30:D7,
l_dst=B4:B5:2F:CE:F1:AE, dl_type=0x0800, nw_src=160.75.17.7,
nw_dst=160.75.17.11, nw_proto=21, tp_src=22, tp_dst=80, idle_timeout=3600,
actions=output:3
```

şeklindedir.

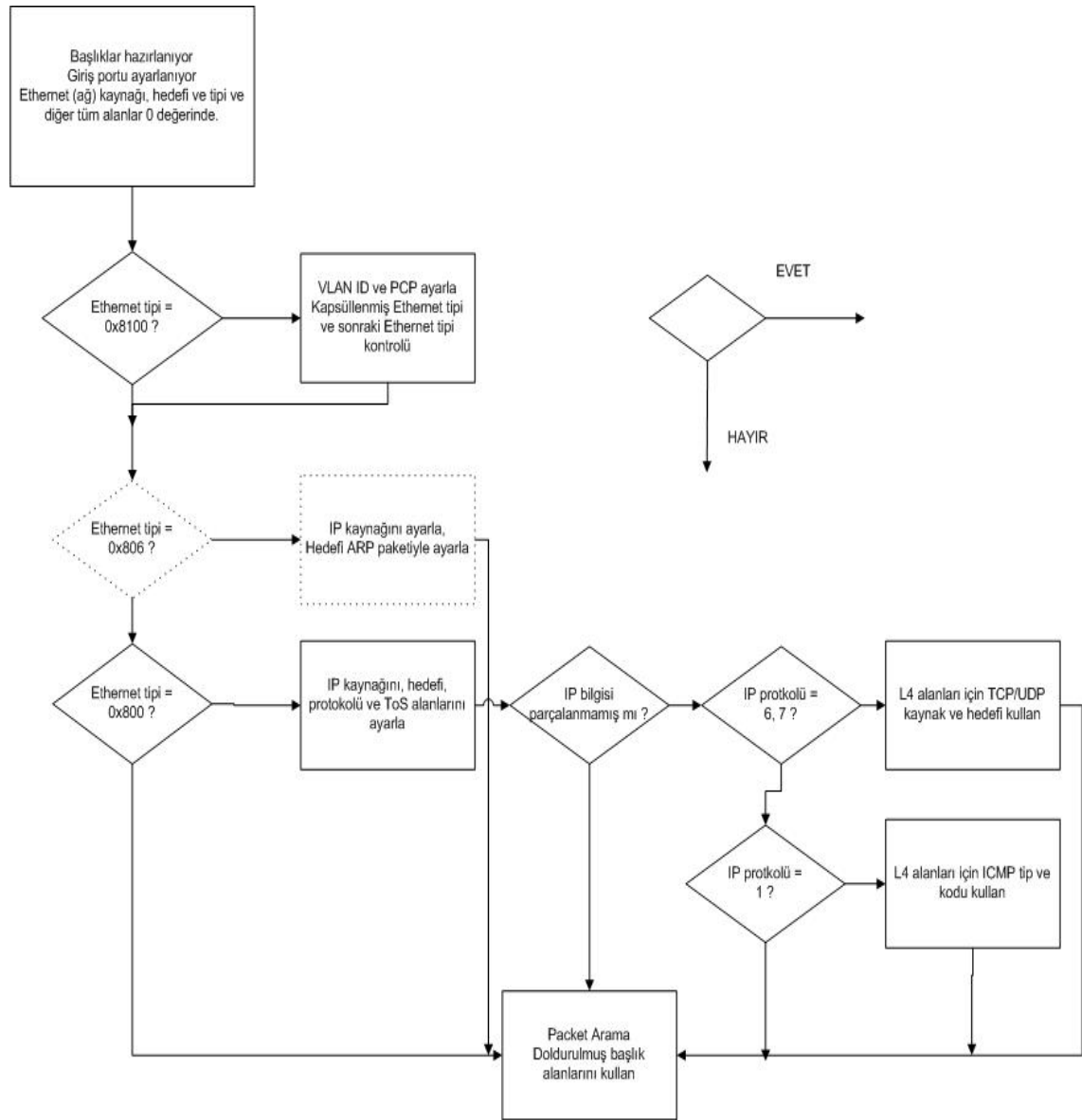
Bir paket OpenFlow anahtarına geldiğinde Şekil 3.7'deki üstbilgi biçimi kullanarak ayrıştırılır ve flow tablosundaki tüm kayıtların başlıkları ile karşılaştırılır. Eşleştirme algoritması Şekil 3.7'deki başlığın yapısını takip ederek Şekil 3.8'de gösterilen akış şemasına uygun şekilde işler.

Ayrıştırma sonrasında, ilk adım olarak, giriş portu, ağ tipi kaynak ve hedef adresleri ile ağ kaynağı bilgileri ayarlanır. Şekil 3.8'de gösterilen diğer tüm alanların değeri bu aşamada sıfırdır. Gelen ilk veride Ethernet tipi değerinin 0x8100 olup olmadığı kontrol edilir. Eğer beklenen değer bulunursa, *VLAN ID* ve *PCP* (Priority Code Point) değerleri arama işlemi için flow verisinin başlığına eklenir.

Bu durum yoksa Ethernet tipi, gelen *IP* paketinde Ethernet çerçevesiyle taşınıp taşınmadığı 0x800 değeri ile kontrol edilir. Eşleşme durumunda ise *IP* kaynak ve hedef

bilgisi flow tablosu aramasında kullanılmak üzere başlığa eklenir. Opsiyonel olarak Bu iki adım arasında *ARP* kontrolü yapılabilir. *IP* kontrolünden sonra yapılan kontrol *IP* paketinin parçalı olup olmadığı kontrolüdür. Eğer parçalanma yoksa paket normal olarak kabul edilir ve akış içerisinde “EVET” adımıdaki ekstra kontroller yapılır. İlk kontrol transport katmanı protokol kontrolü (*TCP* ve *UDP*) bundan sonra ise *ICMP* kontrolüdür. İlk işlemde *TCP* ve *UDP* kaynak ve hedef portları eklenir, daha sonraki işlemde ise *ICMP* tipi ve kodu başlığa eklenir.

Bu akış içerisinde oluşturulan başlık bilgisi flow tablosunun tüm kayıtları arasında kontrol edilir [38].



Şekil 1-8 Flow tablosundaki verilerle paketin karşılaştırılması. [36]

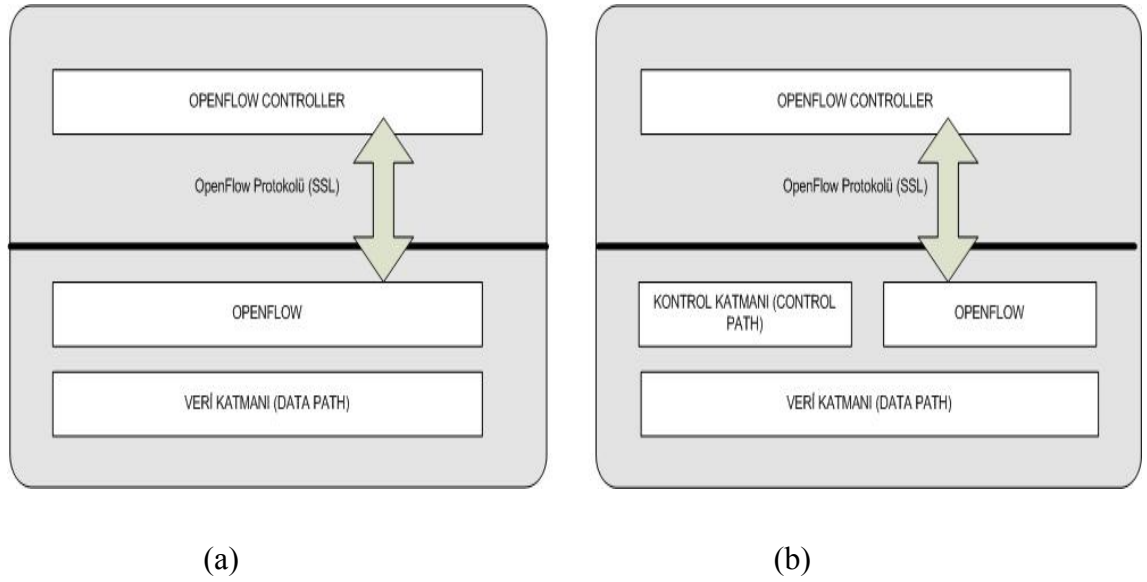
1.8.3 OpenFlow ağ anahtarı Eylemleri

Bahse konu OpenFlow ağ anahtarlarını eylemleri iki adet ana bölümde toplanır: zorunlu eylemler ve opsiyonel eylemler. Adlarından da anlaşılacağı gibi zorunlu eylemlerin tüm OpenFlow ağ anahtarlarında olması gerekmektedir. Opsiyonel eylemler ise sonradan geliştirebilir olanlardır.

OpenFlow ağ anahtarları iki kategoriye ayrılırlar:

1-Dedicate OpenFlow ağ anahtarları

2-OpenFlow etkin (enabled) ağ anahtarları [39]



Şekil 1-9 (a) Dedicated OpenFlow ağ anahtarı, (b) OpenFlow protokolü etkin ağ anahtarı

Openflow ağ Flow tablosunda bulunan flow'larla ilgili Decicated OpenFlow ağ anahtarlarında 3 adet işlem gerçekleştirir. Bir diğer dördüncü işlem ise OpenFlow protokolü etkin ağ anahtarlarında gerçekleşmektedir:

1. Flow paketlerinin belirtilen port veya portlara yönlendirilmesi. Bu da bir paketin ağ içinde hareketine olanak verir. Çoğu ağ anahtarında beklenen belli bir oranda bu işlemi gerçekleştirmesidir.
2. Flow paketlerini kapsüller ve denetleyiciye yönlendirir. Bu paketler güvenli bağlantı ile gönderilirler. Burada denetleyici, flow paketinin ne yapılacağına karar verir. Flow tablosuna ekleyebilir ya da işlem yapmadan ağ içinde yönlendirebilir.
3. Flow paketlerini siler. Örneğin ağa yönelik bir saldırı söz konusu ise, denetleyici bu paketi siler.
4. OpenFlow protokolü etkin ağ anahtarlarında, flow, ağ anahtarının normal işlem döngüsüne sokulur.

1.9 POX Denetleyicisi

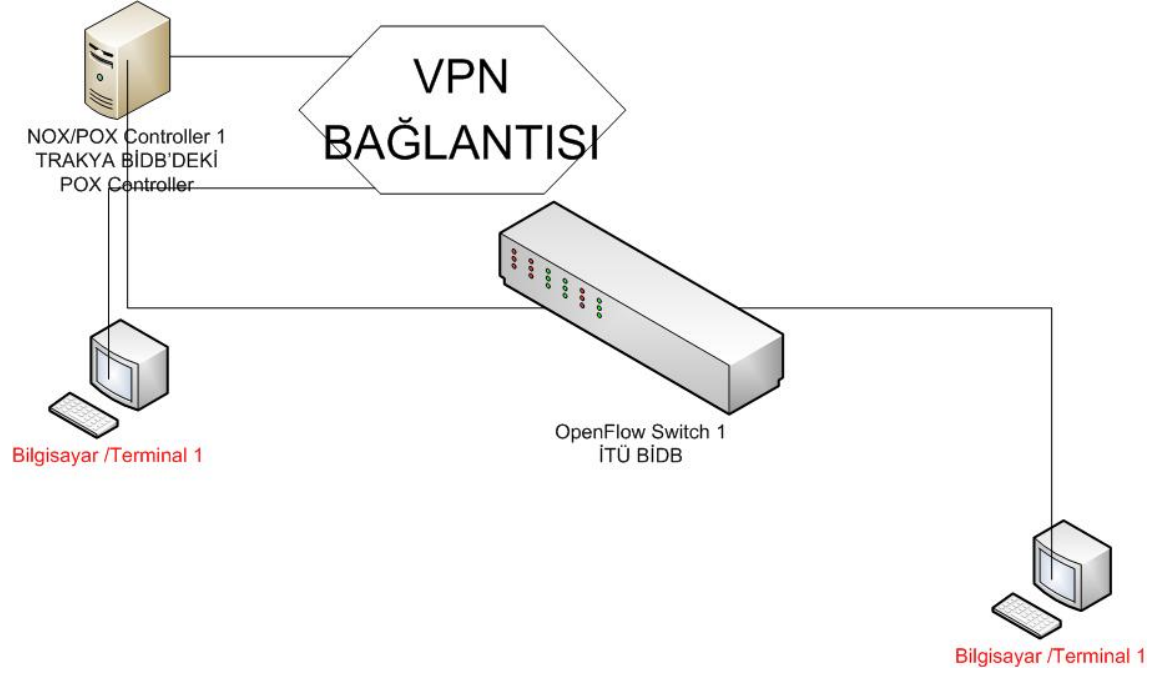
OpenFlow teknolojinin bir diğer önemli ayağı denetleyicidir. Denetleyici (Controller) tanımı kısaca Uzaktan Kontrol (kumanda) işlemi olarak tanımlanabilir [35]. OpenFlow ağ anahtarlarının flow tablolarının ağ anahtarı dışındaki bir bilgisayardan yönetilmesini sağlar. Anahtardaki flow tablosunun flow elemanları bu denetleyici sayesinde kolayca eklenip çıkarılabilir. Denetleyici sayesinde, anahtardaki ağ ile ilgili değişimler görülebilir ve anahtarın konfigürasyonundaki değişiklikler de yapılabilir.

OpenFlow Denetleyicisi ürünleri birden fazla olup bu çalışmada *POX* denetleyicisi kullanılmıştır. *POX*, python programlama diliyle geliştirilmiş olup. *LINUX*, *UNIX* ve *WINDOWS* işletim sistemlerinde çalışmaktadır. Python diliyle yazılmış olmasında ötürü, python runtime destekleyen her türlü işletim sisteminde çalışma olanağı sunmaktadır.

1.10 TEST ORTAMI

1.10.1 Ağ yapısı

Bu çalışmada Trakya Üniversitesi ve İstanbul Teknik Üniversitesi Bilgi İşlem Daire Başkanlıkları arasında oluşturulan bir sanal ağ kullanılmıştır. OpenFlow ağ anahtarı İstanbul Teknik Üniversitesi'nde olup, OpenFlow denetleyicisi ve ağ test yazılımı Trakya Üniversitesi'nde kuruludur. Genel yapı Şekil 3-10'da gösterilmiştir.



Şekil 1-10 Çalışmada kullanılan ağ yapısı.

Trakya Üniversitesi'nde ağ trafiği üretici, İstanbul Teknik Üniversitesi'nde ise, üreticinin hedef bölümü bulunmaktadır. POX denetleyicisi de Trakya Üniversitesi'ndedir. Trakya Üniversitesi'nde bulunan PC'nin özellikleri şunlardır:

Intel Core i3 İşlemci

4 GB RAM, 64 bit Windows 8

İstanbul Üniversitesi'nde bulunan PC ise sanal bir makinedir. Sadece aşağıda bahsedilen D-ITG yazılımı ölçüm istemcisi için kullanılmıştır:

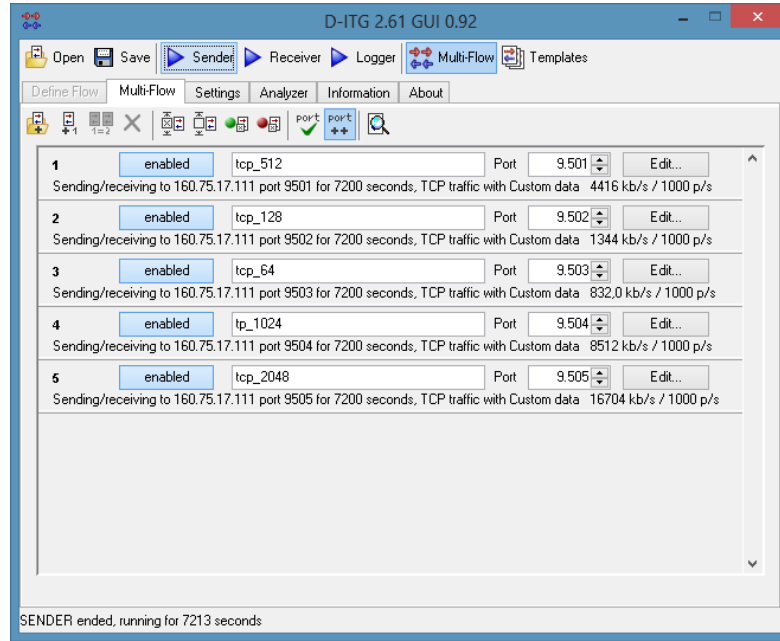
Pentium 4 2.40 GHz

128 MB RAM, Windows XP SP 3

Bu iki bilgisayar birbirlerine, CISCO VPN istemci uygulaması ile bağlıdır. POX denetleyicisi Trakya Üniversitesi'nde, ağ trafiği üretici bilgisayarda farklı bir bilgisayarda kuruludur.

1.10.2 Ağ Trafiği oluşturma yazılımı

Tez çalışmasında, *D-ITG, Distributed Internet Traffic Generator 2.8.0* yazılımı kullanılmıştır. Bu yazılım, belli parametreler vererek örneğin ağ üstündeki verilerin paket boyutlarını ayarlayarak istenilen zaman aralıklarında ağ üzerinde trafik oluşturmak için kullanılmaktadır. Açık kaynak kodlu olan uygulama, sunucu istemci mantığıyla çalışmakta olur, kaynak ve hedef bilgisayarlarda aynı anda çalıştırılarak, ağ üstündeki paketlerin gönderilme ve ulaşma zamanlarını tutarak analiz için uygun bir rapor çıkartmaktadır.



Şekil 1-11 D-ITG uygulaması arayüzü

D-ITG uygulamasında kullanılan, ITGSend.exe bileşeninin parametreleri ve gönderilen ağ paketlerinin özellikleri ise aşağıdaki gibidir.

```
-a 160.75.17.111 -m RTTM -T TCP -rp 9501 -t 28800000 -c 64
```

```
-a 160.75.17.111 -m RTTM -T TCP -rp 9501 -t 28800000 -c 128
```

```
-a 160.75.17.111 -m RTTM -T TCP -rp 9501 -t 28800000 -c 512
```

```
-a 160.75.17.111 -m RTTM -T TCP -rp 9501 -t 28800000 -c 1024
```

-a 160.75.17.111 -m RTTM -T TCP -rp 9501 -t 28800000 -c 2048

Yapılan işlemin tam komutu ise şöyledir:

ITGSend.exe itg_multi_list.txt -l 8_saat_loglama.txt

1.10.3 HP E5406 z1 Ağ Anahtarı



Şekil 1-12 OpenFlow destekli ağ anahtarı cihazı [40].

Ağ anahtarına ulaşmak için, *SSH* portu üzerinden bağlanabilen, ücretsiz bir yazılım olan *putty* tercih edilmiştir. Ağ anahtarına bağlandıktan sonra,

conf t

komutu ile, çalışmada kullanılan pcerin birbirine bağlı olduğu 111. *VLAN*'ın ayarlarına girilebilir, bunun için

vlan 111

komutu yazılır, bu aşamada, OpenFlow protokolünü devre dışı bırakmak veya devreye almak için

openflow enable/disable

komutları kullanılır. Aynı *VLAN* için controller atama için ise

openflow controller tcp:CONTROLLER_IP_ADRESİ:PORT_NUMARASI

komutu kullanılır.

Çalışma esnasında, 3.10.3 teki ağ anahtarının 3 durumu; OpenFlow protokolü devre dışı, OpenFlow protokolü etkin ve POX Denetleyicisi etkin durumu için, 5 adet sırasıyla, 64, 128, 512, 1024, 2048 byte'lık veri paketleri ilke 8 saat boyunca ağ trafiği oluşturulmuştur.

1.10.4 PYTHON

Çalışmadan kullanılan POX Denetleyicisi uygulaması, python dilinde yazılmış olup, pypy isimli, alternatif bir python runtime'ı üzerinde çalışmaktadır. Denetleyici uygulamasını başlatmak için

```
python.exe pox/pox.py openflow.of_01 --address=160.75.17.13 --port=6633
```

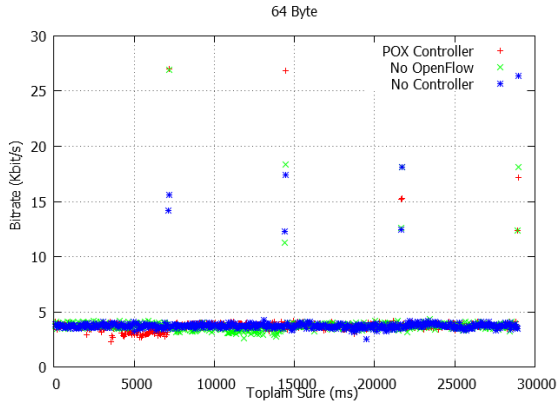
komutu kullanılmıştır.

BÖLÜM 4

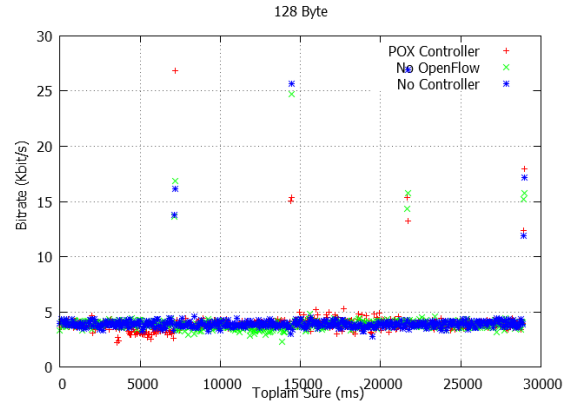
2 TARTIŞMA VE SONUÇ

Yapılan tez çalışmasında, test ortamında 8 saat boyunca 5 adet farklı boyutta ağ paketi gönderilerek bir trafik oluşturulmuştur. Alınan değerlerin bit oranı yani saniye başına bit değerinden grafik değerleri aşağıya çıkarılmıştır.

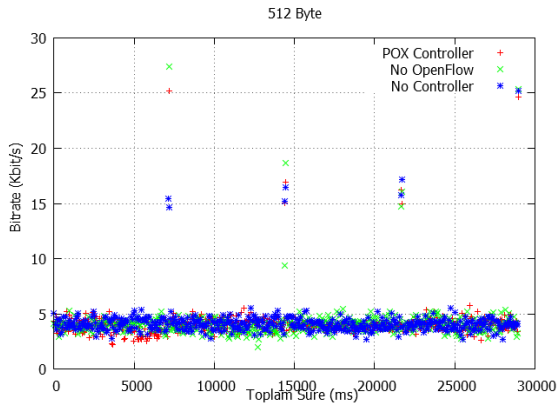
2.1 Bit Oranı (Bitrate) sonuçları



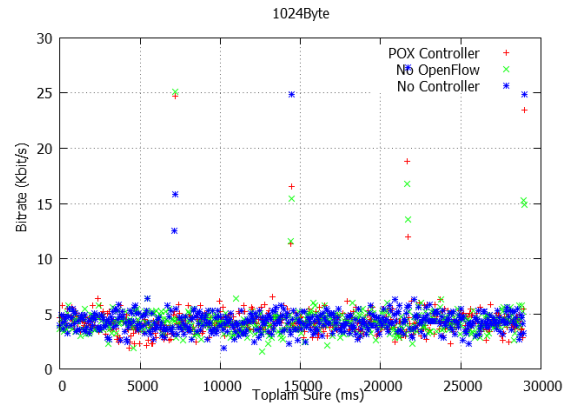
(a) 64 byte



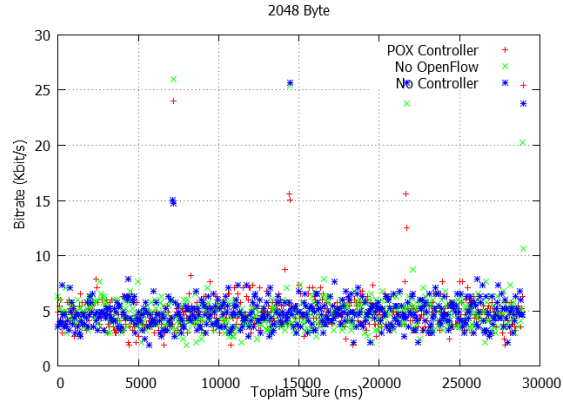
(b) 128 byte



(c) 512 byte



(d) 1024 byte



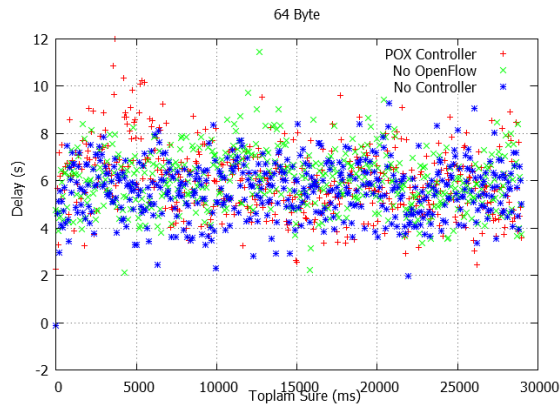
(e) 2048 byte

Şekil 2-1 Bit oranı değerleri.

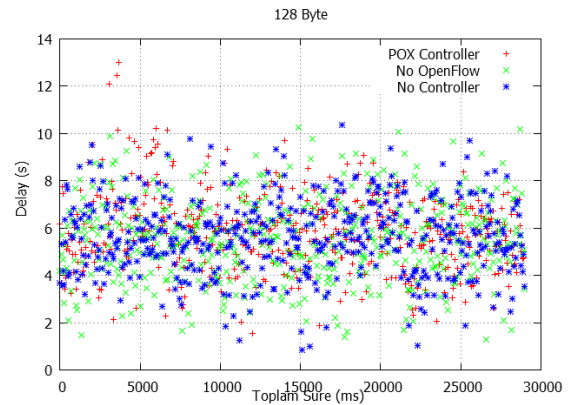
Bit oranı değerlendirmesinde, 64, 128, 512 byte'lık ağ paketlerinde, gerek ağ anahtarı üzerinde, OpenFlow etkin ve devre dışı iken sonuçlar ile ağ anahtarı POX Denetleyicisi ile denetlenirken, bit oranı dağılımının çok farklı bir değerde olmadığı gözlenmiştir.

2.2 Gecikme (Delay) sonuçları

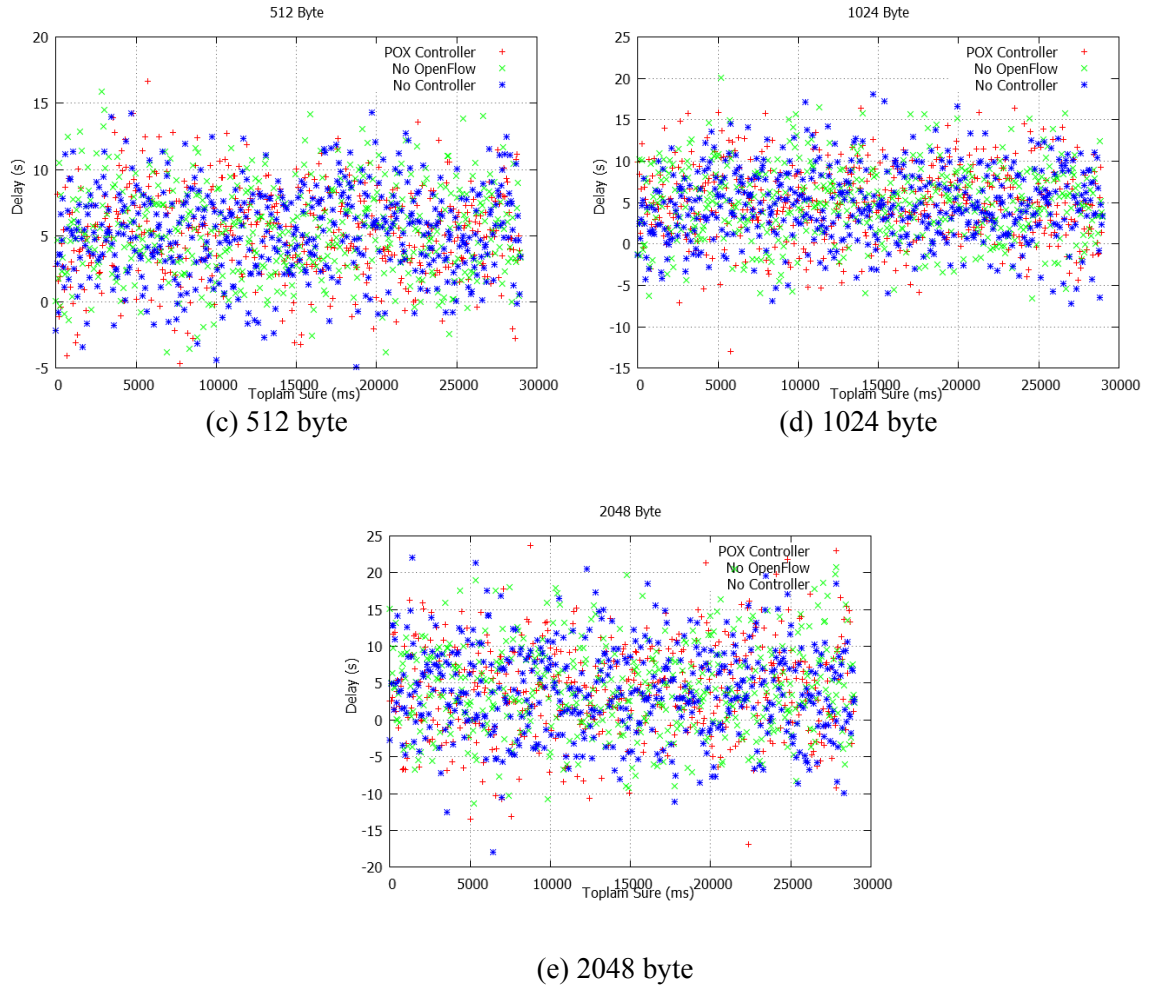
Ağ gecikmesi bir bilgisayar ağının önemli bir tasarım ve performans özelliğidir. Gecikme tek yönlü gecikme ya da gidiş-dönüş gecikmesi olarak da ölçülebilir [41]. Gidiş-dönüş gecikmesi (round-trip delay) olarak adlandırılır (*RTT*). *RTT* ölçümü, karşılıklı iki uygulama gerektirir. *D-ITG* uygulaması ağ gecikmesini hesaplarırken gidiş-dönüş gecikmesi hesaplanmıştır.



(a) 64 byte



(b) 128 byte

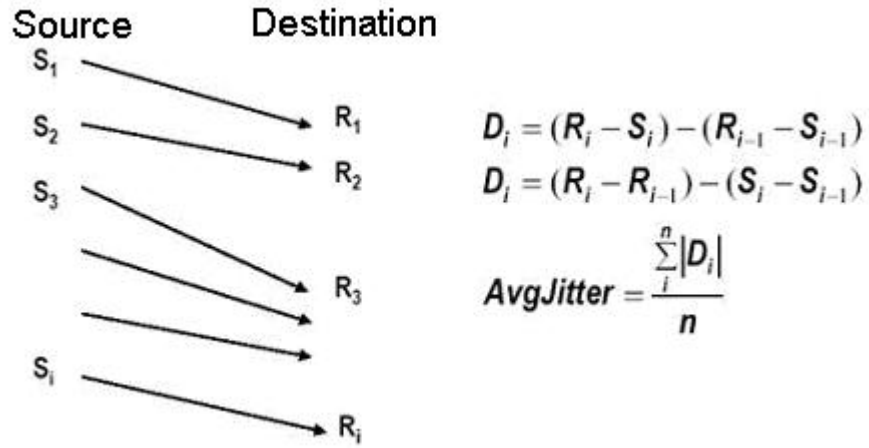


Şekil 2-2 Ağ gecikmesi değerleri.

Gidiş-dönüş gecikmesi süreleri ve değerleri incelendiğinde, her üç durumda ve 64, 128, 512, 1024, 2048 byte'lık paket boyutlarında, belirli bir fark görülememiştir.

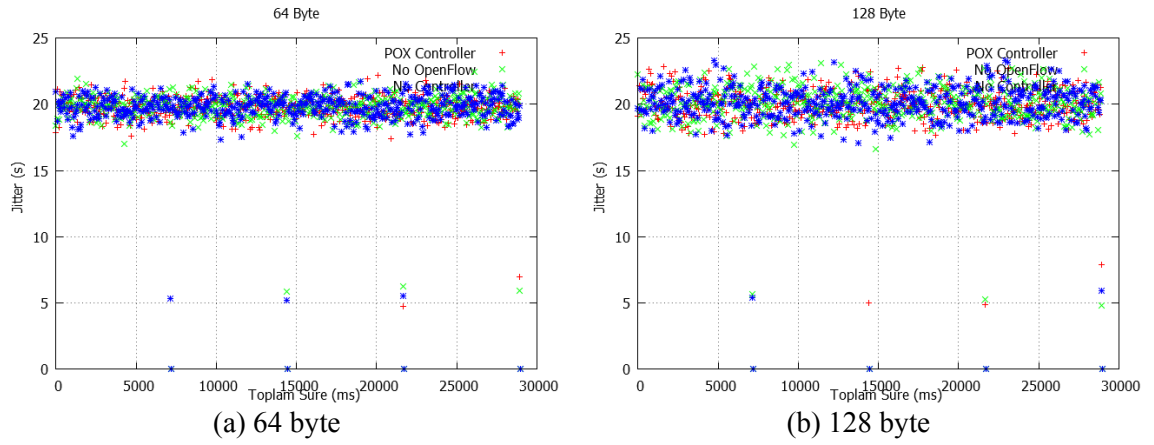
2.3 Ağ gecikmesi dağılımı (Jitter) sonuçları

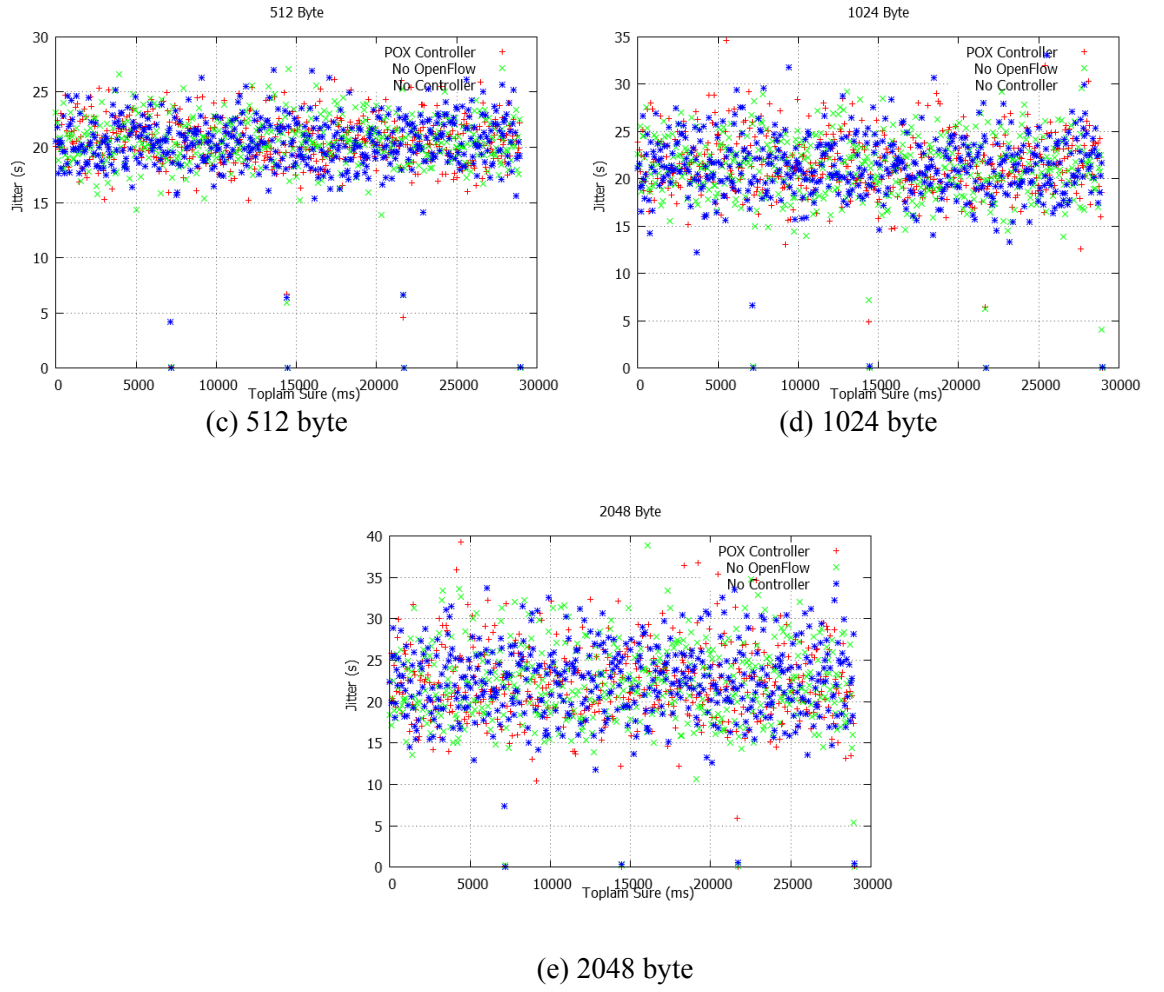
Basitçe söylemek gerekirse, jitter ağdaki gecikmenin varyasyonudur [42]. Ağ gecikmesi dağılımı verilerinin hesabı şekil 4.3'te gösterilmiştir.



Şekil 2-3 Ağ gecikmesi dağılımı değerlerinin hesaplanması [43].

Tüm çalışmanın ağ üzerindeki değerlerini görebilmek için, ağdaki gecikmenin varyasyonu üzerinde istatistikli bir çalışma yapılmıştır. Buradaki hipotezimiz, üç durumda da, POX denetleyicisinin kullandığımız ağ anahtarının çalışmasında belirli bir etkisinin olup olmadığının anlaşılmasına çalışılmasıdır.





Şekil 2-4 Ağ gecikmesi dağılımı değerleri.

2.4 SONUÇ

2.4.1 64 byte'lık paketler için sonuçların değerlendirilmesi

Yapılan çalışmada, 64 byte'lık paketler için, OpenFlow protokolü ve buna bağlı POX denetleyicisinin ağ anahtarının yönettiği durumlarda alınan sonuçlar incelendiğinde, POX denetleyicisi ile çalıştırıldığında, kullandığımız ağ anahtarının veri iletişimde, kayda değer bir değişiklik olmadığı görülmüştür. Şekil 4.5 incelendiğinde sadece 15 ms'lik fark oluştuğu görülebilir.

Descriptive Statistics

Dependent Variable: Jitter (s)

OpenFlow ve Controller Durumu	Mean	Std. Deviation	N
OpenFlow Devre Dışı.	19,72010252	1,320077647	484
OpenFlow Etkin No POX	19,55555622	2,256485732	484
POX	19,55710093	2,141973679	484
Total	19,61061828	1,952909405	1448

Tablo 2-1 64 byte'lık paketler için Ağ gecikmesi dağılımı değerleri istatistiği.

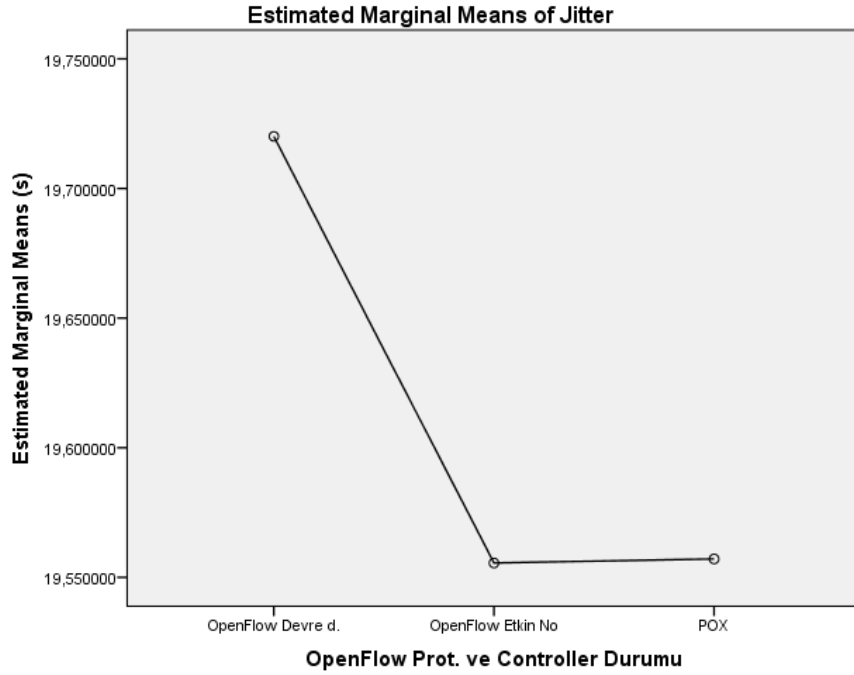
Tests of Between-Subjects Effects

Dependent Variable: Jitter (s)

Source	Type III Sum of Squares	df	Mean Square	F	Sig.
Corrected Model	8,607 ^a	2	4,304	1,129	,324
Intercept	556875,160	1	556875,160	146039,674	,000
OpenFlow Durumu ve Controller	8,607	2	4,304	1,129	,324
Error	5510,041	1445	3,813		
Total	562385,202	1448			
Corrected Total	5518,648	1447			

a. R Squared = ,002 (Adjusted R Squared = ,000)

Tablo 2-2 64 byte'lık paketler için Ağ gecikmesi dağılımı değerleri değerlendirilmesi.



Şekil 2-5 64 byte'lık Ağ gecikmesi dağılımı değerlerinin ölçümleri.

2.4.2 128 byte'lık paketler için sonuçların değerlendirilmesi

Üretilen ağ trafiğinde paket boyutu iki katına çıkarılarak, 128 byte'lık paketler için, OpenFlow protokolü ve buna bağlı POX denetleyicisinin ağ anahtarının yönettiği durumlarda alınan sonuçlar incelenmiş, POX denetleyicisi ile çalıştırıldığında, kullandığımız ağ anahtarının veri iletişiminde, 4.4.2 bölümünde gözlenen değişimden farklı bir sonuç vermediği görülmüştür. Şekil 4.6 incelendiğinde ortalama olarak bu üç durum arasında 12 ms'lik fark olduğu görülebilir

Descriptive Statistics

Dependent Variable: Jitter(s)

OpenFlow ve Controller Durumu	Mean	Std. Deviation	N
OpenFlow Devre Dışı	19,85799681	2,420571452	484
OpenFlow Etkin No	19,87413446	2,317742501	484
POX	19,74183942	2,371526112	484
Total	19,82465690	2,369418035	1452

Tablo 2-3 128 byte'lık paketler için Ağ gecikmesi dağılımı değerleri istatistiği.

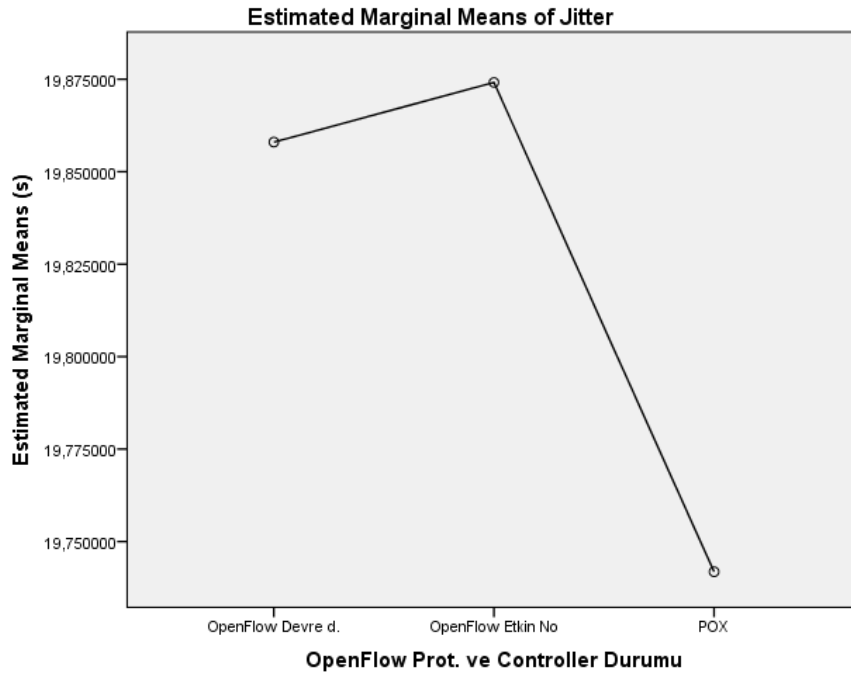
Tests of Between-Subjects Effects

Dependent Variable: Jitter (s)

Source	Type III Sum of Squares	df	Mean Square	F	Sig.
Corrected Model	5,042 ^a	2	2,521	,449	,639
Intercept	570660,715	1	570660,715	101569,773	,000
OpenFlow ve Controler Durumu	5,042	2	2,521	,449	,639
Error	8141,077	1449	5,618		
Total	578806,834	1452			
Corrected Total	8146,120	1451			

a. R Squared = ,001 (Adjusted R Squared = -,001)

Tablo 2-4 128 byte'lık paketler için Ağ gecikmesi dağılımı değerleri değerlendirmesi.



Şekil 2-6 128 byte'lık paketler için Ağ gecikmesi dağılımı değerlerinin ölçümleri.

2.4.3 512 byte'lık paketler için sonuçların değerlendirilmesi

512 byte'lık ağ paketleri için, OpenFlow protokolü ve buna bağlı POX denetleyicisinin ağ anahtarının yönettiği durumlarda alınan sonuçlar incelendiğinde, POX Denetleyicisi ile çalıştırıldığında, kullandığımız ağ anahtarının veri iletişimde,

kayda değer bir değişiklik olmadığı görülmüştür. Şekil 4.7 incelendiğinde ortalama olarak birbirleri arasında 20 ms'lik fark oluştuğu görülebilir

Descriptive Statistics

Dependent Variable: Jitter (s)

OpenFlow ve Controller Durumu	Mean	Std. Deviation	N
OpenFlow Devre Dışı.	20,37354376	2,838713071	484
OpenFlow Etkin No POX	20,25169131	2,970903050	484
POX	20,44275731	2,891020274	484
Total	20,35599746	2,899798243	1452

Tablo 2-5 512 byte'lık paketler için Ağ gecikmesi dağılımı değerleri istatistikleri.

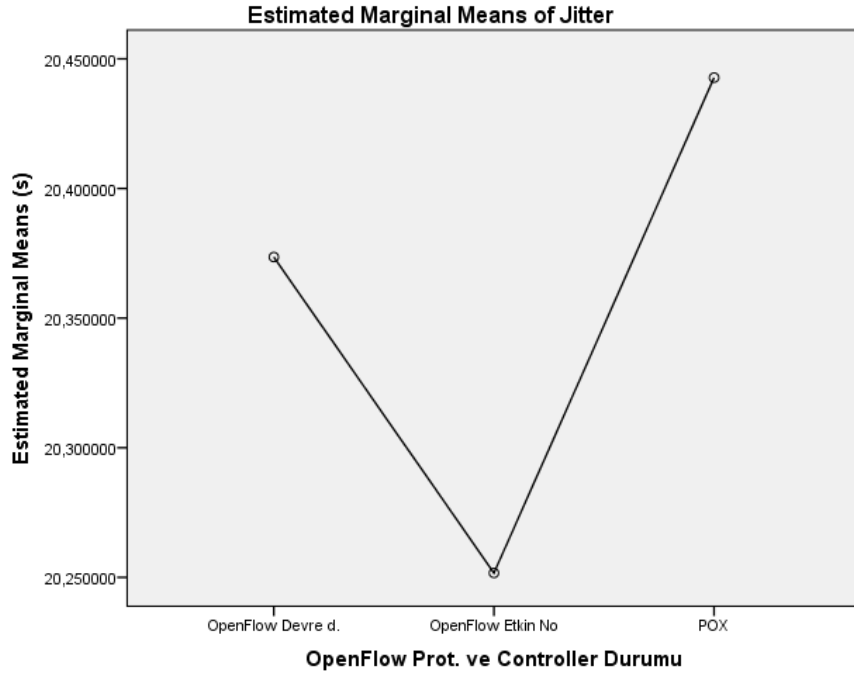
Tests of Between-Subjects Effects

Dependent Variable: Jitter (s)

Source	Type III Sum of Squares	df	Mean Square	F	Sig.
Corrected Model	9,058 ^a	2	4,529	,538	,584
Intercept	601660,351	1	601660,351	71505,481	,000
OpenFlow ve Controller Durumu	9,058	2	4,529	,538	,584
Error	12192,154	1449	8,414		
Total	613861,563	1452			
Corrected Total	12201,212	1451			

a. R Squared = ,001 (Adjusted R Squared = -,001)

Tablo 2-6 512 byte'lık paketler için Ağ gecikmesi dağılımı değerleri değerlendirmesi.



Şekil 2-7 512 byte'lık paketler için Ağ gecikmesi dağılımı değerlerinin ölçümleri.

2.4.4 1024 byte'lık paketler için sonuçların değerlendirilmesi

Yapılan çalışmada, 1024 byte'lık ağ paketleri için, OpenFlow protokolü ve buna bağlı POX denetleyicisinin ağ anahtarının yönettiği durumlarda alınan sonuçlar incelendiğinde, POX denetleyicisi ile çalıştırıldığında, kullandığımız ağ anahtarının veri iletişimde, Şekil 4.8 incelendiğinde ortalama olarak üç durum arasında yaklaşık 35 ms'lik fark olduğu görülmüştür. Paket boyutunun artmasıyla, böyle bir farkın olduğu düşünülmektedir..

Descriptive Statistics

Dependent Variable: Jitter (s)

OpenFlow ve Controller Durumu	Mean	Std. Deviation	N
OpenFlow Devre Dışı.	21,02074278	3,706450559	484
OpenFlow Etkin No POX	21,16146350	3,582484118	484
POX	21,39098719	3,736156773	484
Total	21,19106449	3,676269509	1452

Tablo 2-7 1024 byte'lık paketler için Ağ gecikmesi dağılımı değerleri istatistikleri.

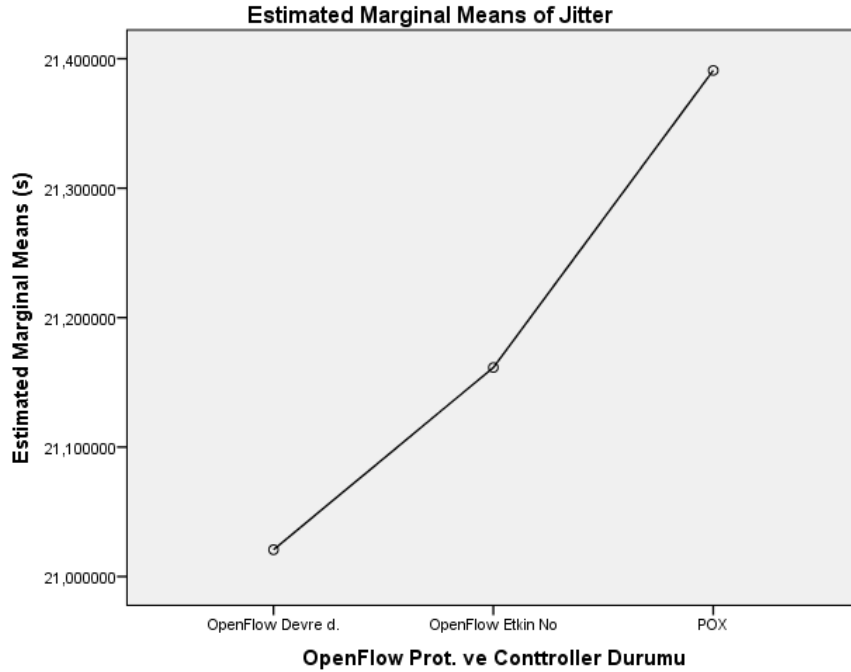
Tests of Between-Subjects Effects

Dependent Variable: Jitter (s)

Source	Type III Sum of Squares	df	Mean Square	F	Sig.
Corrected Model	33,810 ^a	2	16,905	1,251	,286
Intercept	652036,883	1	652036,883	48262,283	,000
OpenFlow ve Controler Durumu	33,810	2	16,905	1,251	,286
Error	19576,394	1449	13,510		
Total	671647,086	1452			
Corrected Total	19610,203	1451			

a. R Squared = ,002 (Adjusted R Squared = ,000)

Tablo 2-8 1024 byte'lık Ağ gecikmesi dağılımı değerleri değerlendirilmesi.



Şekil 2-8 1024 byte'lık Ağ gecikmesi dağılımı değerlerinin ölçümleri.

2.4.5 2048 byte'lık paketler için sonuçların değerlendirilmesi

Son olarak, 2048 byte'lık ağ paketleri için, OpenFlow protokolü ve buna bağlı POX denetleyicisinin ağ anahtarının yönettiği durumlarda alınan sonuçlar

incelendiğinde, POX denetleyicisi ile çalıştırıldığında, kullandığımız ağ anahtarının veri iletişimde, Şekil 4.9’da ortalama olarak birbirleri arasında yaklaşık 37 ms’lik fark oluştuğu görülmektedir. Bu değer de bir önceki değere oldukça yakın bir değerdir.

Descriptive Statistics

Dependent Variable: Jitter (s)

OpenFlow ve Controller Durumu	Mean	Std. Deviation	N
OpenFlow Devre Dışı	22,02414475	4,775860857	484
OpenFlow Etkin No POX	22,27211203	4,582169985	484
POX	22,06652740	4,798145473	484
Total	22,12092806	4,717712152	1452

Tablo 2-9 2048 byte’lık paketler için Ağ gecikmesi dağılımı değerleri istatistikleri.

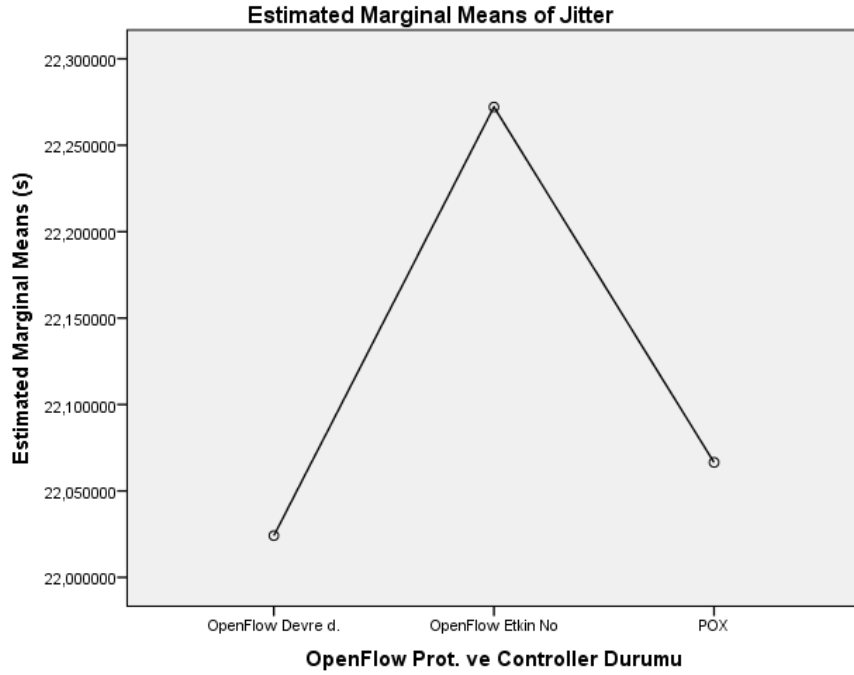
Tests of Between-Subjects Effects

Dependent Variable: Jitter (s)

Source	Type III Sum of Squares	df	Mean Square	F	Sig.
Corrected Model	17,029 ^a	2	8,514	,382	,682
Intercept	710515,085	1	710515,085	31896,311	,000
OpenFlow ve Controller Durumu	17,029	2	8,514	,382	,682
Error	32277,600	1449	22,276		
Total	742809,714	1452			
Corrected Total	32294,628	1451			

a. R Squared = ,001 (Adjusted R Squared = -,001)

Tablo 2-10 2048 byte’lık paketler için Ağ gecikmesi dağılımı değerleri değerlendirmesi.



Şekil 2-9 2048 byte'lık paketler için Ağ gecikmesi dağılımı değerlerinin dağılımı.

2.4.6 Genel Değerlendirme

Tüm sonuçlar değerlendirildiğinde, POX denetleyicisi ve ağ anahtarının OpenFlow protokolünün etkin olup olmaması durumlarında, kayda değer bir değişiklik görülmemiştir. Elde edilen farklar göz önüne alındığında, POX denetleyicisinin küçük ve orta işletmelerin sahip olduğu ağlarda, tercih edilebilecek de bir yazılım olduğu düşünülmektedir.

POX denetleyicisi, python dili ile geliştirildiğinden, platform bağımsız bir seçenek sunmaktadır. Python yürütüm (runtime) ortamının çalışabildiği her işletim sisteminde, denetleyiciyi çalışma imkanı bulunmaktadır. Yürütüm ortamının bu özelliği nedeniyle, mobil işletim sistemlerinden, gömülü sistemlere kadar geniş bir yelpaze araştırmacıların önüne açılmaktadır. Bu bağlamda, OpenFlow için “ÖzgürAkış” karşılığını kullanmak uygun olacaktır [44].

3 KAYNAKLAR

- [1] Jeremy M. Norman, From Gutenberg to the Internet: *A Sourcebook on the History of Information Technology 2. Cilt*, Jeremy Norman Co, s. 869, 2005.
- [2] İnternet Sitesi, Will.i.am song Reach For The Stars to debut on Mars; <http://www.bbc.co.uk/newsbeat/19399703>. Erişim Tarihi 11.11.2012.
- [3] İnternet Sitesi, First Teleportation from One Macroscopic Object to Another <http://www.technologyreview.com/view/507531/first-teleportation-from-one-macroscopic-object-to-another/> . Erişim tarihi 06.10.2012
- [4] Onur Bektaş, Murat Soysal, Serkan Orcan, *Türkiye İçin IPv6 Geçiş Zaman/Aşama Planı Önerisi*, Ulusal IPv6 Protokol altyapısı Tasarımı ve Geçiş Projesi, s. 13, 2011.
- [5] Martin Casado, Michael J. Freedman, Justin Pettit, Jianying Luo, Nick McKeown, Scott Shenker, ACM SIGCOMM Computer Communication Review Volume 37 Issue 4, s 1-12, 2007.
- [6] Jad Naous, David Erickson, G. Adam Covington, Guido Appenzeller, Nick Mckeown, Implementing an OpenFlow switch on the NetFPGA platform, Architecture for Networking and Communications Systems - ANCS , pp. 1-9, 2008.
- [7] Yüksek Lisans Tezi, Manuel Palacin Mateo, politecnico di torino, OpenFlow Switching Performance, http://www.openflow.org/downloads/technicalreports/MS_Thesis_Polito_2009_Manuel_Palacin_OpenFlow.pdf, 2009.
- [8] Rodrigo Braga, Lightweight DDoS Flooding Attack Detection Using NOX/OpenFlow, *35th Annual IEEE Conference on Local Computer Networks and Workshops*, 2010.
- [9] Andrea Bianco, Robert Birke, Manuel Palacin, Luca Giraudo, Openflow Switching: Data Plane Performance, In Proc. IEEE ICC 2010 (Next-Generation Networking and Internet Symposium), Cape Town, SA, 2010.
- [10] Nate Foster, Rob Harrison, Matthew L. Meola, Michael J. Freedman, Jennifer Rexford, and David Walker, Frenetic: A high-level language for OpenFlow

- networks, in Proc. Workshop on Programmable Routers for the Extensible Services of Tomorrow, 2010.
- [11] Minlan Yu, Jennifer Rexford, Michael J. Freedman, and Jia Wang Scalable flow-based networking with DIFANE, ACM SIGCOMM, Yeni Delhi, 2010.
- [12] Amin Tootoonchian, Yashar Ganjali, **A Distributed Control Plane for OpenFlow Networks**, NSDI Internet Network Management Çalıştay/Enterprise Networking (INM/WREN) Çalıştay, San Jose, CA, 2010.
- [13] M. Ribeiro Nascimento, C. Esteve Rothenberg, M. R. Salvador, Carlos Corrêa, Sidney Lucena and M. F. Magalhães. **Virtual Routers as a Service: The RouteFlow Approach Leveraging Software-Defined Networks**, 6. Future Internet Technologies 2011 (CFI 11), 2011.
- [14] H.E. Egilmez, B. Gorkemli, A.M. Tekalp, S. Civanlar. **Scalable Video Streaming Over OpenFlow Networks: An Optimization Framework For QoS Routing**, IEEE International Conference on Image Processing (ICIP 2011), 2011.
- [15] Michael Jarschel, Simon Oechsner, Daniel Schlosser, Rastin Pries, Sebastian Goll, Phuoc Tran-Gia. **Modeling and Performance Evaluation of an OpenFlow Architecture**, 23rd International Teletraffic Congress (ITC 2011), 2011.
- [16] Bennesby, R. ; Fonseca, P. ; Mota, E. ; Passito, A. **An inter-AS routing component for software-defined networks**, Network Operations and Management Symposium 2012 (NOMS), 2012.
- [17] Vestin J.; Dely, P.; Kassler, A.; Bayer, N.; Einsiedler, H. J.; Peylo C. **CloudMAC - Towards Software Defined WLANs** , ACM Mobicom, 2012.
- [18] İnternet Sitesi, <http://www.webmastersitesi.com/donanim-genel/675424-bilgisayar-agi-network-nedir.htm>. Erişim Tarihi 13.11.2012.
- [19] Mark Dye, Rick McDonald, Antoon Ruffi , Cisco Networking Academy, CCNA network fundamentals, OSI and TCP/IP model, Chapter 3, Cisco Press, s. 44, 2010.
- [20] İnternet Sitesi, http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=20269, Erişim Tarihi 17.11.2012.

- [21] International Telecommunication Union , ITU-T Rec. X.200 (1994 E), **Detailed description of the resulting OSI architecture**, Application Layer, s 32. 1994.
- [22] Mark Dye, Rick McDonald, Antoon Ruffo Cisco Networking Academy, CCNA Network Fundamentals ,OSI and TCP/IP Modeli, Chapter 3 , Cisco Press, s. 44, 2010.
- [23] Çölkesen R., Örencik B., Bilgisayar Haberleşmesi ve Ağ Teknolojileri, Papatya Yayıncılık, İstanbul, s. 39, 1999.
- [24] İnternet Sitesi, Sema Aktuğ. Dr. , Bilgisayar Haberleşmesi ders notları, OSI Referans Modeli, <http://web.itu.edu.tr/oktug/BH/notlar/bolum8.pdf>. Erişim Tarihi 16.12.2012
- [25] İnternet Sitesi, <http://computernetworkingnotes.com/osi-layer-modals/advantage-of-osi-layer.html>. Erişim Tarihi, 17.11.2012.
- [26] Leon Shklar, Richard Rosen, *Web Application Architecture: Principles, Protocols and Practices*, Wiley, s. 13, 2005.
- [27] Gökhan Atlı, internet sitesi, <http://tr.scribd.com/doc/68929040/25/TCP-IP-Modeli>, Academytech CCNA Eğitim Notları. Erişim Tarihi 15.10.2012
- [28] RFC 1122, Requirements for Internet Hosts – Communication Layers, R. Braden (ed.), 1989.
- [29] RFC 1123, Requirements for Internet Hosts – Application and Support, R. Braden (ed.), 1989.
- [30] Nick McKeown, Tom Anderson, Hari Balakrishnan, Guru M. Parulkar, Larry L. Peterson, Jennifer Rexford, Scott Shenker, Jonathan S. Turner, ACM SIGCOMM Computer Communication Review Volume 38, Number 2, April, s 70, 2008.
- [31] Thaar Al_taiy, *4 in 1: The Complete One-Week Preparation for the Cisco Ccent/CCNA*, Thaar Technology Publishing, s 365, 2011.
- [32] İnternet Sitesi, What is OpenFlow, <http://www.openflow.org/wp/learnmore/>. Erişim Tarihi 13.01.2013
- [33] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J.Rexford, S.Shenker, J.Turner, OpenFlow: Enabling Innovation in Campus Networks, <http://www.openflow.org/documents/openflow-wp-latest.pdf>, 2008.

- [34] İnternet Sitesi OpenFlow: The Next Generation Network interoperability, <http://www.bladenetwork.net/userfiles/file/OpenFlow-WP.pdf>, 2011.
- [35] Question & answer: openflow and software-defined networking, http://h17007.www1.hp.com/docs/openflow/openflow_faq.pdf.
- [36] Doktora Tezi, Yüksek Hızlı Ağlar İçin Zamanlama ve Anahtarlama Mimarilerinin Tasarımı ve Gerçeklenmesi, <http://www.mcu-turkey.com/wp-content/uploads/2011/11/cpu-turkey-3.pdf>. s. 3, 2011.
- [37] Jun Xu, Member, IEEE, and Mukesh Singhal, Fellow, **Cost-Effective Flow Table Designs for High-Speed Routers: Architecture and Performance Evaluation**, IEEE TRANSACTIONS ON COMPUTERS, VOL. 51, NO. 9, s. 1090, 2002.
- [38] İnternet Sitesi, OpenFlow Switch Specification v1.0. <http://www.openflowswitch.org>. Erişim Tarihi 16.12.2012.
- [39] Nick McKeown, Tom Anderson, Hari Balakrishnan, Guru M. Parulkar, Larry L. Peterson, Jennifer Rexford, Scott Shenker, Jonathan S. Turner, ACM SIGCOMM Computer Communication Review Volume 38, Number 2, s. 71, 2008.
- [40] İnternet Sitesi, HP 5400 zl Switch Series, Data Sheet.Hp Invent, 2009,<http://h20195.www2.hp.com/v2/GetPDF.aspx/4AA2-6511ENW.pdf>, Erişim Tarihi, 17.12.2012.
- [41] Abeck, Bryskin, Evans, Farrel, Filsfils, Hegering, McCabe, Network *Management Know It All*, Morgan Kaufmann, s. 135, 2009.
- [42] Tim Szigeti, Kevin McMenamy, Roland Saville, *Cisco TelePresence Fundamentals*, Cisco Press, s 79, 2009.
- [43] Stefano Avallone, Alessio Botta, Alberto Dainotti, Walter de Donato, and Antonio Pescap'e , D-ITG V. 2.7.0-Beta2 Manual University of Napoli Federico II May 24, s 12, 2009.
- [44] Yüksek Lisans Tezi, Hilmi Enes Eğilmez, Koç Üniversitesi, s. V, İstanbul 2012.

4 ÖZGEÇMİŞ

Adı Soyadı: Selçuk YAZAR

Doğum Tarihi: 06 Haziran1976

Öğrenim Durumu:

Derece	Bölüm/Program	Üniversite	Yıl
Lisans	Fizik Mühendisliği.	Ankara Üniversitesi	1993-1997
Y. Lisans	Bilgisayar Mühendisliği	Trakya Üniversitesi	2010-2013

Yüksek Lisans Tez Başlığı ve Tez Danışmanı:

Ağ anahtarlarında Openflow protokolü ve POX denetleyicisi kullanımı / Doç.
Dr. Erdem UÇAR

Görevler:

Görev Unvanı	Görev Yeri	Yıl
Sistem Destek Uzmanı	Saysis Bilgisayar	1998-1999
Teknik Destek Şefi	GISMAP-INFOTECH	1999-2001
Uygulama Mimarı	Turkuaz Bilgisayar	2002-2004
Araştırmacı	TÜBİTAK-UME	2004-2010