KOCAELİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

YÜKSEK LİSANS TEZİ

MPLS YÖNLENDİRME KULLANAN OMURGA AĞ YAPISININ PERFORMANS DEĞERLENDİRMESİ

KORHAN SÖZEN

KOCAELİ 2013

KOCAELİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

YÜKSEK LİSANS TEZİ

MPLS YÖNLENDİRME KULLANAN OMURGA AĞ YAPISININ PERFORMANS DEĞERLENDİRMESİ

KORHAN SÖZEN

Prof.Dr. Adnan KAVAK
Danışman, Kocaeli Üniv. .....

Yrd.Doç.Dr. Suhap ŞAHİN
Jüri Üyesi, Kocaeli Üniv. .....

Yrd.Doç.Dr. Necla BANDIRMALI
Jüri Üyesi, Yalova Üniv. .....

Tezin Savunulduğu Tarih: 09.07.2013

ÖNSÖZ VE TEŞEKKÜR

Bu çalışmada MPLS yönlendirme kullanan omurga ağ yapısının performans değerlendirmesi yapılmıştır. MPLS ile uyumlu çalışan OSPF ve IS-IS protokolleri yardımıyla omurga ağ yapısı üzerinde çeşitli trafik türleri gönderilerek çeşitli sonuçlar elde edilmiştir. Elde edilen bu veriler ile MPLS kullanıldığında ve kullanılmadığında omurga ağ yapısının performansında ne gibi değişikliklerin göründüğü tespit edilmiştir.

Yüksek lisans eğitimim süresince değerli birikimlerini benimle paylaşan, tezimin her aşamasında sorunlarımı dinleyerek, çalışmalarıma yön veren ve yoğun akademik yaşamında değerli zamanını her türlü problemimi çözmeye ayıran tez danışmanım saygı değer hocam Prof. Dr. Adnan KAVAK'a, desteklerini benden hiçbir zaman esirgemeyen ve yüksek lisan eğitimimi her zaman destekleyen Kocaeli Türk Telekom Network Müdürü Kenan BAYRAK'a, Kocaeli Üniversitesi Bilgisayar Mühendisliği Bölümünde Araştırma Görevlisi olarak çalışan arkadaşlarım Fidan KAYA ve Süleyman EKEN'e ve tüm mesai arkadaşlarıma teşekkürlerimi sunarım.

Ayrıca maddi ve manevi desteklerini tüm hayatı boyunca esirgemeyen annem Mürüvet SÖZEN'e, babam Ali SÖZEN'e ve ağabeyim Gürkan SÖZEN'e şükranlarımı sunarım.

Mayıs – 2013 Korhan SÖZEN

İÇİNDEKİLER

ÖNSÖZ VE TEŞEKKÜR i
İÇİNDEKİLER ii
ŞEKİLLER DİZİN iv
TABLOLAR DİZİNİ ivi
SİMGELER DİZİNİ ve KISALTMALAR vii
ÖZET viii
ABSTRACT x
GİRİŞ 1
1. MPLS 2
1.1. Tarihçe 2
1.2. MPLS Teknolojisi 3
1.3. MPLS Yapısında Kullanılan Terimler 3
1.3.1. Yönlendirme (Routing) 3
1.3.2. Yönlendirici (Router) 3
1.3.3. Anahtarlama (Switching) 4
1.3.4. Ağ Anahtarı (Switch) 4
1.3.5. Etiket 4
1.3.6. Etiket Anahtarlama 5
1.3.7. Etiket Kenar Yönlendiricisi (LER – Lable Edge Router) 5
1.3.8. Etiket Anahtar Yönlendiricisi (LSR – Lable Switching Router) 5
1.3.9. Etiket Anahtar Yolu (LSP – Lable Switching Path) 5
1.3.10. Aynı Yönlendirme Sınıfı (FEC- Forwarding Equivalence Class) 5
1.3.11. Etiket Dağıtım Protokolü (LDP – Label Distribution Protocol) 5
1.4. MPLS Paket Yapısı 6
1.5. MPLS Çalışma Prensibi 9
2. YÖNLENDİRME PROTOKOLLERİ 11
2.1. OSPF (Open Shortest Path First) 11
2.1.1. OSPF Çalışma Prensibi 12
2.1.2. OSPF Paket Tipleri 13
2.1.3. OSPF Bölge Kavramı 14
2.2. IS-IS (Intermediate System to Intermediate System) 15
2.2.1. IS-IS Protokolünde Bölge Mantığı 16
2.2.2. IS-IS Protokolü Çalışma Prensibi 17
2.3. BGP (Boarder Gateway Protocol) 18
2.3.1. BGP Mesaj Tipleri: 19
2.3.2. BGP Protokolü Çalışma Prensibi 19
3. BENZETİM ARAÇLARI 25
3.1. GNS3 25
3.2. Wireshark 30
3.3. Oracle VM Virtual Box 33
3.4. JPERF 2.0 34
4. BENZETİM SONUÇLARI 40
4.1. Ağ Topolojileri 41

4.1.1. OSPF Topolojisi 43
4.1.2. IS-IS Topoloji 49
4.2. Analiz Sonuçları 56
4.2.1. Bant Genişliği (Bandwidth) 56
4.2.2. Seğirme (Jitter) 58
4.2.3. Aktarım (Throughput) 60
5. SONUÇLAR ve ÖNERİLER 63
KAYNAKLAR 65
EKLER 67
ÖZGEÇMİŞ 90

ŞEKİLLER DİZİN

Şekil 1.1. Etiket Dağıtım Protokolü 6
Şekil 1.2. MPLS Paket Yapısı 8
Şekil 1.3. MPLS Çalışma Prensibi 10
Şekil 2.1. OSPF Bölgeleri 15
Şekil 2.2. IS-IS Protokolü Bölge Mimarisi 17
Şekil 2.3. IBGP ve EBGP Alanları 22
Şekil 2.4. IBGP Protokolündeki Yansıtıcılar ve İstemciler 24
Şekil 3.1. GNS3 Genel Görünüm 26
Şekil 3.2. GNS3 Menü Çubuğu 26
Şekil 3.3. GNS3 IOS image and Hypervisors Ekranı 27
Şekil 3.4. GNS3 Node Types Paneli 28
Şekil 3.5. GNS3 Topology Summary Paneli 28
Şekil 3.6. GNS3 Capture Paneli 29
Şekil 3.7. GNS3 Console Paneli 29
Şekil 3.8. Wireshark Genel Görünüm 30
Şekil 3.9. Wireshark Main Tool Bar 31
Şekil 3.10. Wireshark Filter Tool Bar 31
Şekil 3.11. Wireshark Packet List 31
Şekil 3.12. Wireshark Packet Details 32
Şekil 3.13. Wireshark Packet Bytes 32
Şekil 3.14. Wireshark IO Graph 33
Şekil 3.15. Oracle VM Virtual Box 34
Şekil 3.16. Jperf 2.0 35
Şekil 3.17. Jperf Settings 36
Şekil 3.18. Jperf Application Layer Options 37
Şekil 3.19. Jperf Transport Layer Options 37
Şekil 3.20. Jperf IP Layer Options 38
Şekil 3.21. Jperf Graphic Screen 38
Şekil 3.22. Jperf Output 39
Şekil 4.1. Cisco 7200 Service Provider Router 40
Şekil 4.2. OSPF Ağ Topolojisinin Türkiye Haritası Üzerinde Gösterimi 41
Şekil 4.3. IS-IS Ağ Topolojisinin Türkiye Haritası Üzerinde Gösterimi 42
Şekil 4.4. OSPF Ağ Topolojisinin GNS3 Ortamına Yansıtılmış Hali 42
Şekil 4.5. IS-IS Ağ Topolojisinin GNS3 Ortamına Yansıtılmış Hali 43
Şekil 4.6. GNS3 OSPF Topoloji 44
Şekil 4.7. Ankara Yönlendiricisi İçin OSPF Komşuluk Tablosu 44
Şekil 4.8. Wireshark OSPF Konfigürasyonunda Yakalanan Paketler 45
Şekil 4.9. Ankara Yönlendiricisi İçin Yönlendirme Tablosu 46
Şekil 4.10. İstanbul Yönlendiricisinden, Van yönlendiricisine Doğru Yapılan Ping ve Trace Sonuçları 47
Şekil 4.11. OSPF MPLS Konfigürasyonu İle İstanbul Yönlendiricisinden Van Yönlendiricisine Doğru Yapılan Trace Sonuçları 47

Şekil 4.12. Wireshark OSPF MPLS Konfigürasyonunda Yakalanan Paketler..... 48
Şekil 4.13. Ankara Yönlendiricisi İçin MPLS Yönlendirme Tablosu 49
Şekil 4.14. GNS3 IS-IS Topoloji 50
Şekil 4.15. Ankara Yönlendiricisi İçin IS-IS Komşuluk Tablosu 51
Şekil 4.16. Samsun Yönlendiricisi İçin IS-IS Komşuluk Tablosu 51
Şekil 4.17. Wireshark IS-IS Konfigürasyonunda Yakalanan Paketler 52
Şekil 4.18. Ankara Yönlendiricisi İçin Yönlendirme Tablosu 53
Şekil 4.19. İstanbul Yönlendiricisinden, Van yönlendiricisine Doğru Yapılan Ping ve Trace Sonuçları 53
Şekil 4.20. IS-IS MPLS Konfigürasyonu İle Kocaeli Yönlendiricisinden Van Yönlendiricisine Doğru Yapılan Trace Sonuçları 54
Şekil 4.21. Wireshark IS-IS MPLS Konfigürasyonunda Yakalanan Paketler..... 55
Şekil 4.22. Kocaeli Yönlendiricisi İçin MPLS Yönlendirme Tablosu 55
Şekil 4.23. OSPF/OSPF MPLS TCP Bandwidth Grafiği 57
Şekil 4.24. IS-IS/IS-IS MPLS TCP Bandwidth Grafiği 57
Şekil 4.25. OSPF/OSPF MPLS UDP Bandwidth Grafiği 58
Şekil 4.26. IS-IS/IS-IS MPLS UDP Bandwidth Grafiği 58
Şekil 4.27. OSPF/OSPF MPLS UDP Jitter Grafiği 59
Şekil 4.28. IS-IS/IS-IS MPLS UDP Jitter Grafiği 59
Şekil 4.29. OSPF/OSPF MPLS TCP Transfer Grafiği 60
Şekil 4.30. IS-IS/IS-IS MPLS TCP Transfer Grafiği 61
Şekil 4.31. OSPF/OSPF MPLS UDP Transfer Grafiği 61
Şekil 4.32. IS-IS/IS-IS MPLS UDP Transfer Grafiği 62

TABLOLAR DİZİNİ

Tablo 1.1. MPLS ‘in Gelişimi 2

SİMGELER DİZİNİ VE KISALTMALAR

Kısaltmalar

ABR : Area Border Router (Bölge Sınır Yönlendiricisi)
ARIS : Aggregate Route-Based Switching (Birleştirilmiş Yol Temelli IP Anahtarlama)
ATM : Asynchronous Transfer Mode (Eşzamansız Aktarım Modu)
BGP : Border Gateway Protocol (Sınır Geçit Protokolü)
CLNP : Connectionless-mode Network Protokol (Bağlantısız Ağ Protokolü)
CLNS : Connectionless-mode Network Service (Bağlantısız Ağ Servisi)
CoS : Class of Service (Servis Sınıfı)
DBD : Database Description Packet (Veri Tabanı Tanım Paketi)
DEC : Digital Equipment Corporation (Dijital Ekipman Ortaklığı)
DLCI : Data Link Control Identifier (Veri Bağlantı Denetimi Tanımlayıcısı)
EBGP : External Border Gateway Protocol (Dış Sınır Geçit Protokolü)
FEC : Forwarding Equivalence Class (Denklik Sınıf Yönlendirmesi)
GNS : Graphical Network Simulator (Grafik Network Simülasyonu)
IANA : İnternet Assigned Numbers Authority (İnternet Atanmış Numaralar Yetkilisi)
IBGP : Internal Border Gateway Protocol (İç Sınır Geçit Protokolü)
IETF : The Internet Engineering Task Force (İnternet Mühendisliği Görev Gücü)
IOS : Internetworking Operating System (Ağlararası İletişim İşletim Sistemi)
IP : İnternet Protocol (İnternet Protokolü)
IS-IS : Intermediate Systems to Intermediate Systems (Orta Sistemden Orta Sisteme)
LDP : Label Distribution Protocol (Etiket Dağıtım Protokolü)
LER : Lable Edge Router (Etiket Kenar Yönlendiricisi)
LFIB : Label Forwarding Information Base (Etiket Yönlendirme Bilgi Tabanı)
LS Ack : Link State Acknowledgement (Bağlantı Durum Onay Paketi)
LSA : Link State Advertisemets (Bağlantı Durum Anonsu Paketi)
LSP : Lable Switching Path (Etiket Anahtar Yolu)
LSR : Lable Switching Router (Etiket Anahtar Yönendiricisi)
LSU : Link-State Update Packet (Bağlantı Durum Güncelleme Paketi)
MED : Multi-Exit Discriminator (Çoklu Çıkış Ayrımı Devresi)
MPLS : Multi Protocol Lable Switching (Çoklu Protokol Etiket Anahtarlama)
NNSA : Not-So-Stubby Area (Çok Kısa Olmayan Alan)
OSI : Open System Interconnection (Açık Sistem Bağlantı)
OSPF : Open Shortest Path First (Açık Kısa Yöne Öncelik)
PPP : Point-to-Point Protocol (Noktadan Noktaya Protokol)
QoS : Quality of Services (Servis Kalitesi)
RIP : Routher Information Protocol (Yönlendirme Bilgisi Protokolü)

SPF	:	Short Path First (Kısa Yöne Öncelik)
TCP	:	Transmission Control Protocol (İletim Kontrol Protokolü)
ToS	:	Type of Service (Hizmet Tipi)
TTL	:	Time To Live (Yaşam Süresi)
UDP	:	User Datagram Protocol (Kullanıcı Datagram Protokolü)
VCI	:	Virtual Channel Identifier (Sanal Kanal Tanımlayıcı)
VM	:	Virtual Machine (Sanal Makine)
VPI	:	Virtual Path Identifier (Sanal Yol Tanımlayıcı)
VPN	:	Vircual Privet Network (Sanal Özel Ağ)

MPLS YÖNLENDİRME KULLANAN OMURGA AĞ YAPISININ PERFORMANS DEĞERLENDİRMESİ

ÖZET

Günümüzün servis sağlayıcıları omurga ağlarında ağırlıklı olarak MPLS protokolünü kullanmayı tercih etmektedirler. Bunun sebebi MPLS protokolünün diğer protokollere göre yüksek hızda veri transferine olanak sağlamasıdır. Bu sayede servis sağlayıcıları artan trafik yükünü kolaylıkla kontrol edebilmektedirler. Bu çalışma MPLS protokolü kullanan omurga ağların performans değerlendirmelerini içermektedir.

Tez çalışmasında servis sağlayıcılarının Türkiye'de kullandığı tipik omurga ağ yapısının OSPF ve IS-IS protokolleri kullanılarak oluşturulan topolojiler üzerinden MPLS'in kullanıldığı ve kullanılmadığı durumlarda farklı trafikler geçirilerek benzetimi yapılarak etiketleme teknolojisinin ağ trafiğine etkisinin analizi yapılmıştır.

Elde edilen sonuçlarda MPLS kullanıldığında omurga ağ yapısının performansının arttığı, seğirmenin azaldığı ve veri aktarım hızının arttığı gözlemlenmiştir.

Anahtar Kelimeler: Benzetim Sonuçları, BGP, IS-IS, MPLS, OSPF.

PERFORMANCE ANALYSIS OF BACKBONE NETWORK STRUCTURE USING MPLS ROUTING

ABSTRACT

Today's service providers mainly prefer using MPLS protocol in their backbone networks. MPLS protocol allows high speed data transfer, compared to other protocols. In this way service providers can easily control increasing traffic loads. This thesis is focused on performance analysis of backbone networks using MPLS protocol.

In this thesis, by conducting simulation of backbone network topology in Turkey, on which OSPF and IS-IS protocols are considered, performance analysis of this network has been done in terms of bandwidth, jitter and data transfer rate when MPLS protocol is applied on top of these protocols.

As a result of this analysis, it has been observed that the performance of backbone network structure has been improved by reducing jitter and increasing data transfer rate.

Keywords: Simulation Results, BGP, IS-IS, MPLS, OSPF.

GİRİŞ

Ağ teknolojilerinin hızla geliştiği günümüzde internet, her yerden erişilebilirliği ile kurumsal ve son kullanıcı pazarı için birçok yeni uygulamanın geliştirilmesine ilham olmuştur. Geliştirilen bu yeni uygulamalar network omurgası için garanti edilen bant genişliğine olan ihtiyacı arttırmışlardır. Günümüzde Internet üzerinde sağlanan klasik veri uygulamalarının yanı sıra yeni ses ve çoklu ortam uygulamaları geliştirilmiş ve kullanıcılara sunulmuştur. Ancak bu yeni nesil uygulamalar hız ve bant genişliği açısından internet omurgasındaki kaynakları zorlamaktadırlar.

Gelişen teknoloji ile gelecek yıllarda internet kullanıcılarının çok daha fazla artacağı düşünülürse, servis sağlayıcıların trafiği taşıyabilmek için altyapılarını ve network omurgalarını ihtiyaç doğrultusunda geliştirmek zorunda kalmaları kaçınılmazdır. Bu nedenle network omurgalarında çok daha hızlı anahtarlama (switching) ve yönlendirme (routing) yapabilen teknolojik cihazlar tercih edilmektedir.

Günümüzde servis sağlayıcıları altyapılarında hali hazırda hızlı işlem yapabilen yönlendiriciler kullanmaktadırlar. Ancak halen altyapılarının son kullanıcılara bakan belirli bir kısmında hız ve kalitesi düşük, gecikmesi yüksek olan çerçeve aktarıcı (Frame Relay) ve eş zamansız aktarım modu (ATM - Asynchronous Transfer Mode) altyapı teknolojileri kullanmaktadırlar. Ayrıca bu altyapıların gecikmeleri de hesaba katıldığında ses ve görüntü trafiklerinde ciddi problemlerle karşılaşılmaktadır [1].

Bu tarz problemlerin giderilebilmesi için Çok Protokollü Etiket Anahtarlama (MPLS – Multi Protocol Lable Switching) teknolojisi kullanılmaktadır. MPLS teknolojisi halihazırda kullanılmakta olan, standart varış adresine göre düğümden düğüme aktarma yöntemi yerine, etiket değiştirerek aktarma yöntemini getirmiştir. Bu sayede paket iletimi işlemlerini sadeleştirmekte, veri iletimini oldukça hızlandırmakta ve ölçeklenebilirliği arttırmaktadır.

1. MPLS

1.1. Tarihçe

Gelişen MPLS teknolojisi 1990'ların ortalarında geliştirilen ağ teknolojilerinden esinlenilerek ortaya çıkarılan bir teknolojidir. MPLS'in yönlendirme için etiketlerin kullanılması kavramı 1980'li yıllara dayanmaktadır. Bu tarihte mühendisler cihazlar üzerindeki paket yönlendirme oranlarını OC-3 (155,52 Mbps) ve OC-12 (622,08 Mbps)'ye yükseltmeye çalışmaktaydılar. Çalışan yönlendirme yazılımı, paketlerin başındaki adres bilgilerini okuyor ve buna göre karar veriyordu. Bu yüzden karar verme işlemi cihazın işlemci gücüne bağlıydı. Bu probleme alternatif bir çözüm olarak paketlerin IP adreslerine bakılarak karar verme aşamasının hızlandırılması yerine adreslemede etiket kullanımı gündeme geldi. Tasarım açısından etiket anahtarlama yöntemi IP yönlendirmeye göre daha kolay bir işlem gerektiriyor ve önceki yazılımsal yönlendirmenin aksine donanımsal olarakta gerçekleştirilebildiği için daha hızlı karar verme şansı sunuyordu. 1990'ların ortasında etiket anahtarlama dikkatleri üzerine çekti ve bu yöntem için çeşitli teknolojiler geliştirilmeye başlandı. Sırasıyla; Toshiba 'nın hücre anahtarlamalı yönlendirmesi (1995), Ipsilon'un IP anahtarlaması (1996), IBM'in birleştirilmiş yol tabanlı IP anahtarlaması (1996) ve Cisco'nun etiket anahtarlaması (1996) temel alınarak 1997'de The Internet Engineering Task Force (IETF) adlı örgüt etiket anahtarlama için standartları belirledi [2].

Tablo 1.1. MPLS 'in Gelişimi

ÜRETİCİ	ÜRÜN	YIL
TOSHİBA	Cell Switching Router	1995
IPSILON	IP Switch	1996
CISCO	Tag Switching	1996
IBM	Aggregate Route-Based Switching (ARIS)	1996
IETF	MPLS	1997

1.2. MPLS Teknolojisi

MPLS, birçok fikrin sentezinden oluşan bir protokoldür. IP omurgasında hızlı ve kontrol edilebilir bir yapı oluşturabilmenin yanında IP dışındaki protokolleri yönlendirebilme becerisine sahiptir. MPLS'in Frame relay ve ATM trafiklerini yönlendirebilmek için konfigürasyonları bulunmakta ve kullanılmaktadır [3].

MPLS, IP şebekelerini sanal devre modunda çalıştırmaya yarayan bir teknolojidir. Kullanılan IP omurgasına etiket değişim özelliğini eklemiştir. Bu özellik sayesinde datagram tabanlı IP şebekeleri sanal devre şebekelerinin olumlu özelliklerine kavuşmuşlardır.

MPLS paketleri Frame relay, ATM, PPP, Ethernet gibi layer 2 teknolojileri üzerinden iletilebilir. Çerçeve tabanlı MPLS'te etiketler layer 2 ve layer 3 başlıkları arasına yerleştirilirler. Bu etiketleme işlemine Shim Header (Pul başlığı) adı verilir. Hücre tabanlı MPLS'te ise ATM çerçevenin VPI/VCI bölümündeki değerler etiket olarak kullanılır. Layer 2 (veri) anahtarlama teknolojileri ile layer 3 yönlendirme teknolojilerini birleştirir. MPLS mimarisi esnek bir mimari olduğundan her tür layer 2 teknolojisi ile birlikte çalışabilir [4].

1.3. MPLS Yapısında Kullanılan Terimler

MPLS'in çalışma prensibini anlatmadan önce MPLS'in temel protokollerinde, fonksiyonlarında ve terminolojisinde kullanılan terimlerden bahsetmek gereklidir. Kullanılan bu terimleri aşağıdaki gibi sıralayabiliriz.

1.3.1. Yönlendirme (Routing)

Yönlendirme, iletilecek paketlerin ağ mimarisi içerisinde kaynaktan hedefe gönderilmesi olayına verilen isimdir. Yönlendirme ile paketlerin "a" dan "b" noktasına veya ağlar arasında aktarımı ifade edilmektedir.

1.3.2. Yönlendirici (Router)

Yönlendiriciler, yönlendirme işlemini yapan elektronik cihazlardır. Genellikle farklı ağları haberleştirmek için kullanılırlar. Yönlendirmelerini IP adreslerine göre yaparlar. Bir omurgada pek çok yönlendirici çeşitli şekillerde ağa bağlanabilmektedir. Yönlendiriciler OSPF (Open shortest path first), RIP (Router

information protocol) gibi yönlendirme protokollerini kullanan yönlendiriciler, kendilerine gelen paketleri hangi hedefe yollayacaklarına bu protokoller sayesinde karar verirler. Yönlendiriciler bu ve benzeri protokolleri kullanarak yönlendirme tablolarını oluştururlar. Yönlendiriciler, tutulan bu tablolar sayesinde gelen paketin daha sonra yönüne uygun olarak gideceği sonraki noktanın (next hop) belirlenmesini sağlarlar. Tabloların oluşturulması ve paketin gönderileceği yönün belirlenmesi için harcanan zaman mantıksal işlemler için ayrılan zamandan farklıdır.

1.3.3. Anahtarlama (Switching)

Anahtarlama, OSI ikinci katman (Layer 2) temel alınarak verinin giriş kapısından çıkış kapısına yönlendirilmesini ifade etmektedir. Basit şekliyle gelen veri her hangi bir işleme veya sorguya tabi tutulmadan yönlendirilir.

1.3.4. Ağ Anahtarı (Switch)

Ağ anahtarları, anahtarlama işlemini yapan elektronik cihazlardır. Aynı ağ içerisindeki cihazları haberleştirmek için kullanılırlar. Yönlendirmelerini MAC adreslerine göre yaparlar. Ağ anahtarının her bir kapısı (port) diğerlerinden bağımsız veri alış-verişinde bulunabilir. Bir veri paketi kapılardan her hangi birine ulaştığında ağ anahtarı gönderen cihazın MAC adresini ve gönderilen kapıyı adres tablosuna kaydeder. MAC adres tablosunda bulunan kayıtlar incelenerek hedef MAC adresinin bağlı olduğu kapıyı bulmaya çalışır. Eğer MAC adresi tabloda mevcutsa veri paketi sadece hedef kapısına gönderilir. Mevcut değilse veri paketi gelen kapı hariç bütün kapılara gönderilir.

1.3.5. Etiket

Etiketler, oldukça kısa ve sabit uzunlukta olan, yönlendirme işlemine yardımcı olan başlıklardır. Etiketler kullanım itibariyle FEC'lerle (Forwarding Equivalence Class) ilişkidedirler. Etiketler aslında bir adres değillerdir. Onlar yerel ve tek bir veri-bağlantı içindedirler ve geniş çapta hiç bir anlamları yoktur. Etiketler, Çerçeve Aktarımındaki DLCI (Data Link Control Identifier) veya ATM ağlarda kullanılan VPI/VCI'ya (Virtual Path Identifier / Virtual Channel Identifier) benzerler.

1.3.6. Etiket Anahtarlama

Etiket anahtarlama, geleneksel 3. Katman (Layer 3) yönlendirme verimsizliğine çözüm olarak geliştirilen bir yöntemdir. Etiketler 2. Katmanda (layer 2) iletilecek paketler atanarak yüksek hızlı iletime yardımcı olurlar. Etiket yönlendirme tablosundaki bir girdiyi işaretleyerek paketin iletileceği yeri belirtir. Etiket anahtarlama tekniği, her hangi bir yönlendirmeye karar vermeden önce paketleri incelemediği için geleneksel yönlendirme metotlarından çok daha hızlıdır [5].

1.3.7. Etiket Kenar Yönlendiricisi (LER – Lable Edge Router)

MPLS omurgasının kenar noktalarında (en dış kısmında) bulunan, etiketlemenin başladığı ve bittiği yönlendiricilerdir. MPLS omurgasında yapılan bir yönlendirme LER'de etiketleme ile başlar, omurga buyunca devam eder ve yine bir LER'de etiket bilgisiyle yapılan anahtarlama son bulur.

1.3.8. Etiket Anahtar Yönlendiricisi (LSR – Lable Switching Router)

MPLS omurgasının içinde yer alan yönlendiricilerdir. Temel işlevleri kendilerine gelen MPLS paketlerini anahtarlamaktır.

1.3.9. Etiket Anahtar Yolu (LSP – Lable Switching Path)

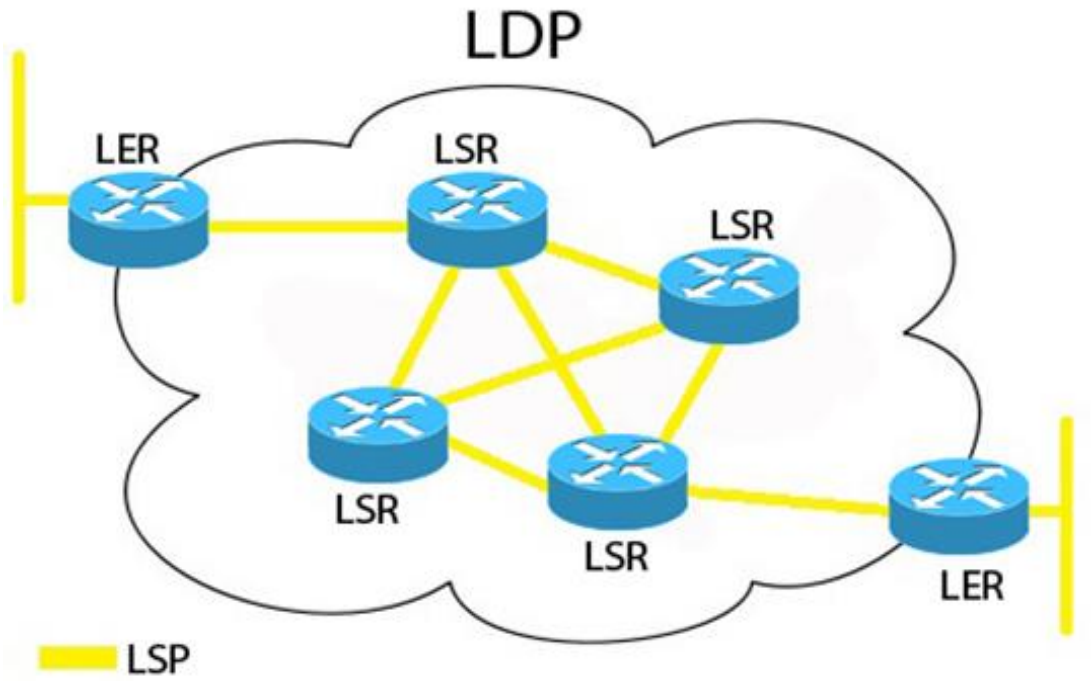
MPLS omurgası boyunca başlangıç ve bitiş LER'leri arasında oluşturulan tek yönlü yollardır. Manuel olarak ağ yöneticileri tarafından oluşturulabildikleri gibi, dinamik olarak etiket dağıtımı için kullanılan LDP'den alınan bilgi ile de oluşturulabilirler [3].

1.3.10. Aynı Yönlendirme Sınıfı (FEC- Forwarding Equivalence Class)

Aynı rotadan yönlendirilen ve benzer işlem gören paketlerin bütününe verilen isimdir.

1.3.11. Etiket Dağıtım Protokolü (LDP – Label Distribution Protocol)

Etiket dağıtım protokolü MPLS ağlarında etiketlerin etiket anahtar yönlendiricilerine gönderilebilmesi için kullanılır. Etiket dağıtım protokolü sayesinde aynı yönlendirme sınıfları etiketler ile eşlenir ve etiket anahtar yolları oluşturulur. Etiket dağıtım protokolü çift yönlü çalışır. Her bağlantıda etiket anahtar yönlendirici çiftleri karşılıklı olarak birbirlerinin etiket eşleştirmesini öğrenebilirler [6].



Şekil 1.1. Etiket Dağıtım Protokolü

1.4. MPLS Paket Yapısı

MPLS‘in çalışma mantığını kavrayabilmemiz için öncelikle MPLS‘in paket yapısın hakkında bilgi sahibi olmamız gerekmektedir.

MPLS paket yapısındaki en önemli unsurlardan biri MPLS etiketleridir. MPLS yürütülen ortama göre dilim veya standart ikinci katman kodlaması şeklinde formları bulunur. Bu aşamada yönlendirme esnasında MPLS etiketlerinin geçirdiği adımlardan bahsetmek yerinde olacaktır. MPLS paketlerinin yönlendirilmesi esnasında paket başlıkları LSR‘ler arasında transfer edilirken; sıkıştırma, çıkarma ve takas işlemlerinden geçirilirler. Bu işlemleri özetleyecek olursak;

Sıkıştırma: Mevcut yığına bir etiket ekleme veya IP paketine etiket ekleme işlemidir. İşlem sonunda TTL değeri IP paketindeki TTL değerine eşlenir.

Çıkarma: LER tarafından yapılan bir işlemdir. Yığının en üstündeki etiket çıkartılır. Etiketin TTL değeri IP paket başlığında bulunan TTL alanına kopyalanır.

Takas: LSR tarafından yapılan bir işlemdir. Sıkıştırma ve takas işlemlerinin beraber yapıldığı işleme verilen addır. Alınan paketteki TTL değeri bir azaltıldıktan sonra oluşturulan yeni paketin MPLS başlığındaki TTL alanına kopyalanır.

Teoride $2^{20} - 1$ adet etiket kullanılabilmektedir. Ancak 0 ila 15 arasında kalan etiketler rezerve edildiği için kullanılmamaktadır. 4 ila 15 arasındaki değerler gelecekte kullanılmak üzere rezerve edilmişlerdir. 0 ila 3 arasında kalan değerlerin her birinin ise farklı bir görevi vardır. Bunlar;

- 0: Bu etiket "IPv4 Açık Sıfır Etiketi" (IPv4 Explicit NULL Label) olarak adlandırılır. Kullanılır ise etiket yığınının en sonunda yer alması gerekir. 0 etiketi, etiketin kaldırılması ve bundan sonra yönlendirilmenin IP başlığına göre devam etmesi gerektiğini bildirir. MPLS'e dayalı servis kalitesi (QoS) uygulamalarında kullanılır.

- 1: Bu etiket yönlendirici uyarı etiketidir. Yönlendirme 1 etiketinden sonraki etikete göre yapılmalıdır. Paket yönlendirilmesine devam edildiği sürece bu etiketle yönlendirilmeye devam eder.

- 2: Bu etiket "IPv6 Açık Sıfır Etiketi" (IPv6 Explicit NULL Label) olarak adlandırılır. 2 etiketi, etiketin kaldırılması ve bundan sonra yönlendirilmenin IPv6 adresine göre yapılması gerektiğini belirtir.

- 3: Bu etiket kapalı sıfır etiketidir. Bu etiketi yalnızca LSR'ler atayıp dağıtabilir. Ayrıca hiçbir zaman kapsüllemede yer almaz. 3 etiketi, LSR'nin en üstteki etiketi çıkartıp ilgili ara yüze, etiketli veya etiketsiz yönlendireceğini belirtir.

MPLS başlığında bulunan üç bitlik Exp alanı MPLS 'in taslaklarında servis sınıfı (CoS) veya tıkanıklık işareti olarak tanımlanmıştır. Bu alan servis tipi (ToS) fonksiyonları için kullanılabilir.

MPLS başlığında bulunan TTL bilgisi ise döngü oluşmasını engellemek amacıyla, IP ile uyumlu şekilde kullanılır. MPLS paketinin takip ettiği LSP yolu boyunca uğranan her LSR yönlendiricinin TTL değeri bir düşürülür. Frame Relay veya ATM gibi TTL desteği olmayan ağlarda ise, TTL paketin omurgaya gireceği yönlendirici üzerinde,

uğranacak yönlendirici kadar düşürülür. Eğer TTL değeri daha yönlendirmeye başlamadan sıfır değerine ulaşmışsa LER yönlendirme yapmaz.

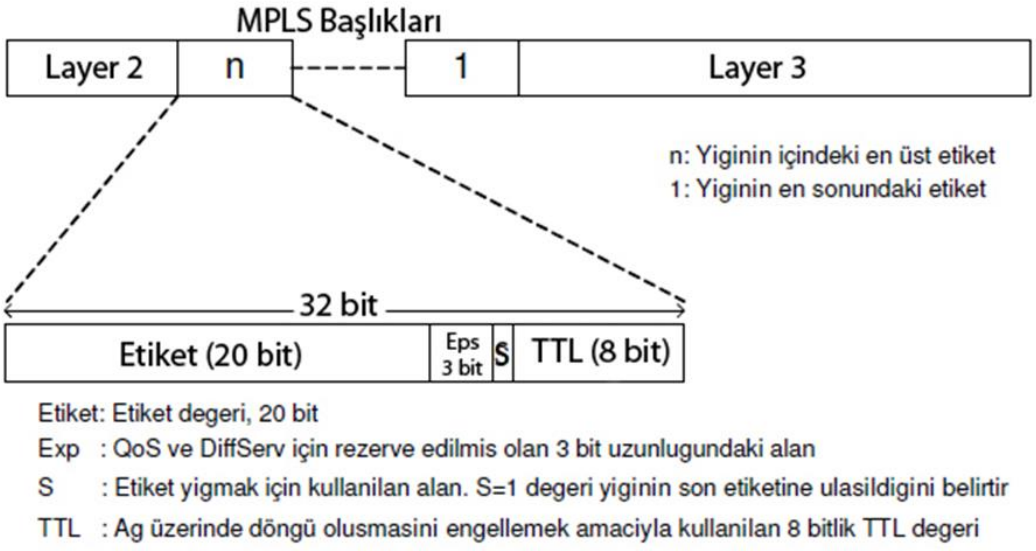
MPLS etiket yığını en son eklenen etiketin ilk önce çıkartılması esasına göre çalışır. Farklı bir ifadeyle X-1 sayıdaki etiketten oluşan bir etiket yığınına eklenen X ‘inci etiket, yönlendirme için ilk kullanılacak etiket olur. Birden fazla etiket bulunan etiket yığınlarında LSR‘ler sadece en üstteki etiketi okur ve buna göre yönlendirme yapar. Yönlendirilecek ara yüz belirlendikten sonra MPLS başlığı çıkarılır. Eğer yeni bir etiket çıkarsa bir sonraki adımda yapılacak yönlendirme için bu başlık kullanılır. Çıkmazsa açılan başlığın yerine yeni bir başlık eklenerek daha önce belirlenmiş olan arayüzden paket gönderilir.

Etiket yığını üç tip uygulamada kullanılabilmektedir;

Birincisi trafik mühendisliği yaparken daha küçük paket yönlendirme akışlarının daha büyük akışlarda toplanmasıdır.

İkincisi üstteki etiketin hedef yönlendiricisi, bir alttaki etiketin de bu yönlendirici üzerinde bulunan sanal özel ağı olabilir.

Son olarak etiket yığınları servis sağlayıcıların birbirlerinin ağlarında tünelleme yoluyla taşımasıdır [3].



Şekil 1.2. MPLS Paket Yapısı

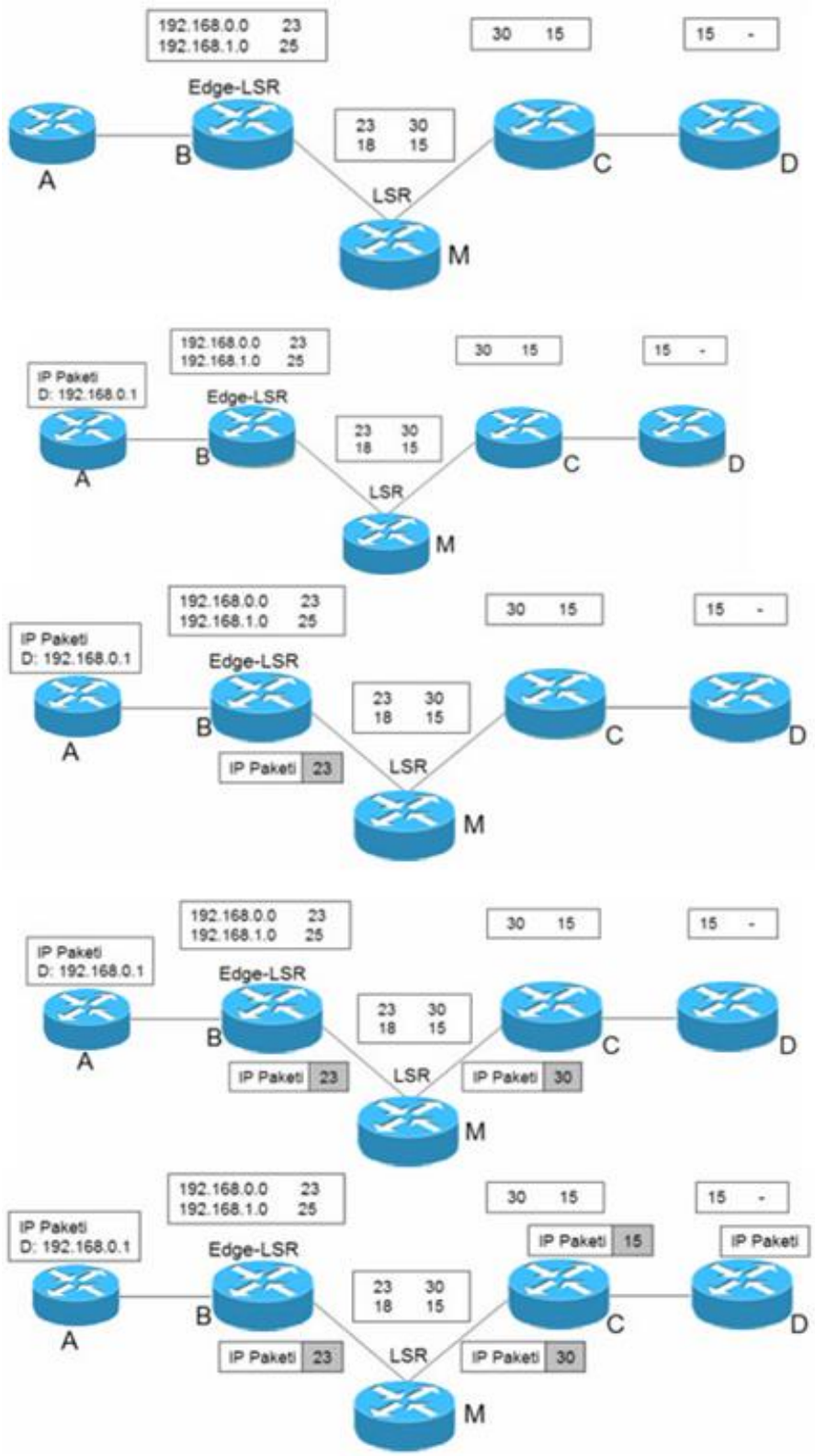
1.5. MPLS Çalışma Prensibi

Bilindiği gibi geleneksel IP yönlendirmelerinde yönlendirme tablosu, yönlendirme protokolleri tarafından veya yöneticilerin yaptığı statik tanımlarla oluşturulur. Her atlamada varış adresi yeniden kontrol edilerek yönlendirme kararı verilir. Paketin ağ omurgası içerisinde izleyeceği yol, dinamik olarak her atlama üzerinde, anlık olarak belirlenir.

MPLS ise layer 2 (switching) ve layer 3 (routing) yönlendirmenin olumlu yönlerini bir araya getirir. Yönlendiricinin yükünü azaltarak hız ve performans artışı sağlar. Uç noktalar arasında, esnek ve zahmetsiz idare edilebilen özel sanal ağ (VPN) tanımlamalarına izin verir. Uç noktalar arasında trafiğin istenen bir yol ve bunun alternatifleri üzerinden gitmesini sağlar (Trafik Mühendisliği). Ayrıca halen servis sağlayıcıların az da olsa kullandığı ATM cihazları üzerinde IP trafiği iletimini kolaylaştırır.

MPLS'in çalışması şu şekilde gerçekleşir;

Bir yönlendirme protokolü baz alınarak yönlendiriciler tarafından ağ yapısı ve yönlendirme tabloları oluşturulur. Tüm yönlendiriciler, oluşturulan yönlendirme tablolarındaki her ağ için etiket (lable) üretirler. LDP protokolü, bitişik aygıtlar arasında etiket değerleri oluşturmak üzere tablodaki yönlendirme topolojisini kullanır. MPLS operasyonuna katılacak yönlendiriciler arasında etiket bilgileri değiştirilir. Bu işlem sayesinde her yönlendiricide, paketler için gelen giden etiket çiftleri oluşturulur ve gidilecek son noktalar arasında LSP'lerin haritalarının oluşması sağlanır. Oluşturulan bu haritalar Etiket Yönlendirme Bilgi Tabanına (LFIB) eklenir. Hangi üçüncü katman servislerine gereksinimi olduğu ve varış adresine göre izleyeceği yola (FEC) uygun bir etiket eklenerek ilk paket belirlenmiş LER'e gönderilir. LER yönlendirme ve gereksinimleri dikkate alarak paket için bir etiket oluşturur ve paketi sonraki LSR'ye gönderir. Paketin iletildiği LRS paketin önündeki etiketi okur, tabloda listelenmiş yeni etiketle değiştirir (Lable switching) ve paketi bitişikteki düğüme iletir. Her çekirdekteki düğümde bu uygulama tekrarlanır. Paketin aktarıldığı son LRS etiketi çıkarır, paketin başlığını okur ve onu son noktaya iletir [7].



Şekil 1.3. MPLS Çalışma Prensibi

2. YÖNLENDİRME PROTOKOLLERİ

MPLS ile kullanılan üç adet yönlendirme protokolü bulunmaktadır. Bunlar;

- OSPF (Open Shortest Path First)
- IS-IS (Intermediate System to Intermediate System)
- BGP (Border Gateway Protocol)

2.1. OSPF (Open Shortest Path First)

Dahili bir yönlendirme protokolü olan OSPF Ekim 1989'da IETF tarafından standart kabul edilmiştir. Günümüze kadar birçok modernizasyondan geçen OSPF dinamik yönlendirme protokollerinden biridir. Marka bağımsız olarak çalışması OSPF'i en çok tercih edilen protokollerden bir tanesi haline getirmiştir [8].

OSPF sayesinde kullanılamaz hale gelen bir ağ hakkındaki bilgi, protokolün çalıştırıldığı bütün yönlendiriciler tarafından bilinir ve bu sayede ulaşılamayacak ağa gönderilen paketler boşu boşuna yönlendiricilerde ilerlemeyerek bağlantılı ağları meşgul etmemiş olurlar.

OSPF gibi dinamik yönlendirme yapabilen protokoller sayesinde statik yönlendirmeden doğan dezavantajlar giderilmiş olur. Örneğin OSPF sayesinde yeni eklenen bir ağ için bütün yönlendiricilere girilmesi gereken yönlendirme tanımlarına gerek kalmaz [2].

OSPF hedef ağa giderken Dijkstra Algoritmasını kullanır. Dijkstra Algoritması en kısa yolu bulmak üzere kullanılan bir algoritmadır [9]. OSPF protokolü sayesinde tüm yönlendiriciler içinde bulundukları topolojide gerçekleşen işlemlerden haberdar olup, topolojinin haritasına sahiptirler. Yönlendiriciler bu harita sayesinde ve dijkstra algoritmasından faydalanarak hedef ağlara ulaşabilecekleri en kısa yolu tespit ederler. OSPF, ağ topolojisinde meydana gelen değişiklikleri kısa sürede algılamak ve bu bilgileri yönlendiricilere en kısa sürede aktarabilmek için geliştirilmiştir [3].

Ağdaki Protokol trafiği sadece ağ üzerinde bir değişiklik olduğu zaman değişiklik bilgisini yönlendiricilere bildirmek üzere yapılır. Bu sayede gereksiz trafiklerle ağ meşgul edilmemiş olur.

OSPF'de yönlendirme kararı, otonom sistemde bulunan yönlendiriciler arası bağlantıların durumuna göre yapılır [2]. Yönlendiriciler, ulaşabildikleri ağlar ile gerekli bilgileri birbirleri ile değiştirirler. Her yönlendirici bulunduğu otonom sistem içinde bulunduğu alandaki bütün yönlendiricilere kendi üzerindeki ara yüzleri ve bu ara yüzlerden hangi ağlara erişimi olduğu bilgisini, "bağlantı durumu" bilgisini gönderir. OSPF bir ağa kaç adımda (hop) ulaşılabildiği bilgisine göre değil, ne kadar zamanda (veya hızda) ulaşılabileceğinin bilgisine göre işlem yapar.

2.1.1. OSPF Çalışma Prensibi

OSPF protokolü, yönlendiricinin ilk çalıştırıldığı anda broadcast yayınında bulunarak topoloji bilgisini tüm arabirimlerinden göndermez. Bunun yerine, ilk olarak komşularını öğrenebileceği bir süreç başlatır. Komşu yönlendiriciler de aynı işlemi yaparlar. Komşuluklar öğrenildikten sonra topoloji bilgilerinin değiş tokuşuna geçilir. Bu topoloji bilgilerinin içinde farklı paket tipleri (Link State Update, Databae Discription) mevcuttur. Değiş tokuş işlemi tamamlandıktan sonra yönlendiriciler SPF (Short Path First) algoritmasını çalıştırarak yeni yolları hesaplamaya başlarlar [10].

OSPF, topoloji bilgisini içeren paketlerin yanı sıra onlar kadar önemli olan Bağlantı Durumu Duyuru (Link State Advertisemets - LSA) paketlerini kullanmaktadır. Örneğin, tek bir bağlantı bilgisi içeren LSA paketi, o bağlantının ait olduğu alt ağın adres bilgisini, maskesini, maliyetini (metrik değeri) ve diğer bilgilerini içermektedir [10].

OSPF protokolü güzergahları şu şekilde belirler;

- Yönlendiriciler, bütün arabirimlerde yer alan komşuları belirler. Birbirlerini tanımak için "Hello" paketleri gönderirler. Komşuluk listeleri bir komşu tablosunda tutulur.

- Daha sonra yönlendiriciler öğrenilen topoloji bilgilerini (LSA) değiş tokuş ederler.

- Yönlendiriciler öğrendikleri topoloji bilgilerini veri tabanına yerleştirirler.

- Bu aşamadan sonra her yönlendirici, topoloji veri tabanları üzerinden SPF algoritması çalıştırarak veri tabanına kayıtlı olan ağlara giden en iyi yolları hesaplayarak belirler.

- Son olarak yönlendiriciler, belirlediği en iyi yolları yönlendirme tablosuna yerleştirirler.

2.1.2. OSPF Paket Tipleri

OSPF protokolü, otonom sistem içinde düzenli işleyişin devamı (yönlendiriciler arasında komşuluk kurulması, bağlantı-durum bilgilerinin senkronizasyonu, yönlendirme tablolarının güncellenmesi) için altı farklı paket kullanır. Bu paketler;

Merhaba Paketi (Hello Packet): OSPF protokolü kullanan yönlendiriciler arasında komşuluk kurulması sırasında ve kurulan komşuluğu devam ettirilmesi için kullanılırlar. İlk komşuluk kurulma anında 224.0.0.5 adresine çoklu yayın olarak gönderilirler, komşuluk kurulduktan sonra ise komsunun adresine unicast olarak 5 ya da 10 saniyede bir gönderilirler.

Veri Tabanı Tanım Paketi (DBD - Database Description Packet): Komşuluklar kurulurken kullanılan pakettir. Yönlendirici veri tabanı içeriğini taşımak için kullanılırlar. Tüm topoloji bilgileri bu pakette bulunur.

Bağlantı-Durum Anonsu Paketi (LSA- Link State Advertisement): Tüm topoloji bilgilerinin bulunduğu pakettir.

Bağlantı-Durum Güncelleme Paketi (LSU - Link-State Update Packet): Komşuluk kurulan yönlendiriciler arasındaki durum bilgilerini güncellemek amacıyla kullanılan pakettir. Topolojide her hangi bir değişiklik olduğunda bu paket kullanılır.

Bağlantı-Durum İstek Paketi (LSR - Link-State Request Packet): Komşuluk kurulan yönlendiricilerden farklı hedeflere ulaşabilme bilgilerini siteme iletme amacıyla kullanılan pakettir.

Bağlantı-Durum Onay Paketi (LS Ack - Link State Acknowledgement): LSU ve LSA paketlerinin alındığına dair karşıya teyit amaçlı gönderilen pakettir [11].

2.1.3. OSPF Bölge Kavramı

OSPF topolojisinin çalışma mantığı bölgeler (Area) üzerine kurulmuştur ve bu sayede bir hiyerarşi sağlanabilmektedir. Bölgeler sayesinde yönlendiriciler kendilerinin bulunduğu bölgeler dışında kalan alt ağların topolojileri hakkında temel bilgilere sahiptirler. Bölge dışında kalan yönlendiriciler hakkında ise hiçbir bilgiye sahip değildirler. Bu durum bellek kullanımını azalttığı gibi SPF çalıştırmak için işlemci gücüne gereksinimi de azaltmaktadır. OSPF'in merkezi 0. Bölgedir (Area 0). Bu bölge, omurga bölgesi (Backbone Area) olarak ta adlandırılır ve farklı bölgeler içinde omurga bölgesi ile iletişim halinde olan yönlendiricilerin mutlaka bir ara yüzü omurga bölgesi ile konuşmalıdır [12].

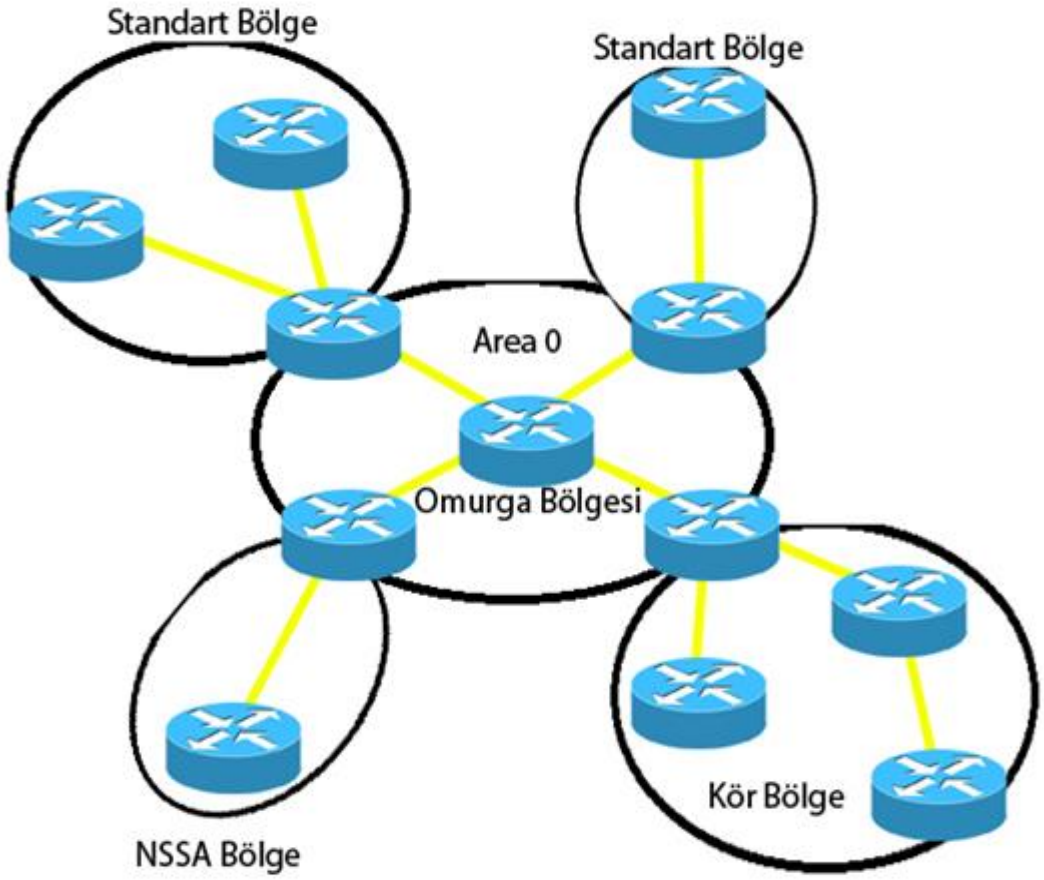
OSPF bölgelerini aşağıdaki gibi sıralayabiliriz;

Omurga Bölgesi (Backbone Area): Transit bölge olarak ta adlandırılır. Diğer bölgeler, bağlantı-durum bilgisi alış-verişi için, mutlaka bu bölge ile bağlantı kurmak zorundadır.

Standart Bölge (Standard Area): Standart bölge, yön bilgisi özetlerini, harici yön bilgilerini ve standart bağlantı durum güncellemelerini kabul eder.

Kör Bölge (Stub Area): Kör bölge harici yön bilgisi güncellemelerini kabul etmez. Bu bölgede bulunan yönlendiriciler harici yön bilgilerini saklamazlar. Kendilerini omurga bölgesine bağlayan bölge sınır yönlendiricilerine (ABR – Area Border Router) doğru varsayılan rotaları bulunmaktadır.

NSSA Bölge (Not-So-Stubby Area): Bu bölgeye de harici yön bilgileri bölge sınır yönlendiricileri tarafından gönderilmez. Ancak, bölgedeki yönlendiriciler tarafından öğrenilen rotalar bölgede bulunan diğer yönlendiriciler ile paylaşılır.



Şekil 2.1. OSPF Bölgeleri

2.2. IS-IS (Intermediate System to Intermediate System)

IS-IS, DEC (Digital Equipment Corporation) tarafından geliştirilmiş, orta sistemler arasındaki iletişimi sağlamak adına tasarlanmış ve 1992 yılında IOS (International Organization for Standardization) tarafından ISO/IEC 10589 kodu ile standartlaştırılmış bir protokoldür [13].

Tam adı; "Intermediate System to Intermediate System Intra-Domain Routeing Exchange Protocol for use in Conjuction with the Protocol for Providing Connectionless-mode Network Service" (Bağlantısız mod ağ servisi sağlamak amacıyla alan içi orta-sistemden orta-sisteme yönlendirme protokolü) dir [3]. Uzun adından da anlaşılacağı üzere IS-IS, CLNP(Connectionless-mode Network Protokol) için yönlendirme protokolü olarak dizayn edilmiştir.

IS-IS protokolü, orta sistemler (IS - İntermediate-systems) arasında hiyerarşik yönlendirme için geliştirilmiş olup, ISO CLNS (Connectionless-mode Network Service) ortamında, CLNP yönlendirmesi için kullanılan dinamik ve bağlantı durumuna göre yönlendirme yapan bir protokoldür. Orta sistem (IS) OSI terminolojisinde yönlendiricileri tanımlar. Uç sistem (ES – End system) ise yönlendirici ile aynı alt ağda bulunan cihazlardır [14].

2.2.1. IS-IS Protokolünde Bölge Mantığı

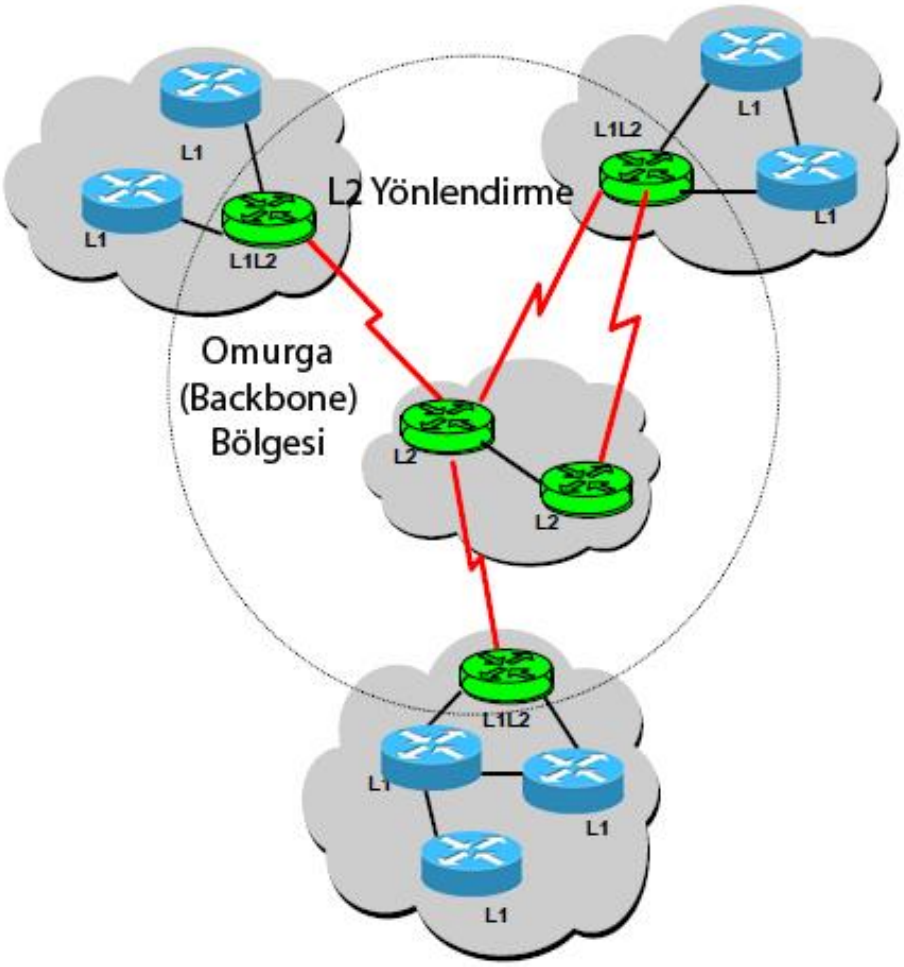
IS-IS protokolü aynen OSPF protokolünde olduğu gibi bölge mantığını kullanır. Farklı bölgeler oluşturularak yönlendirilmenin kolaylaştırılması hedeflenmektedir. IS-IS protokolünde bulunan bölgelerde 2 farklı yönlendirme yapılır. Bu yönlendirmeler seviye 1 (Level 1) ve seviye 2 (Level 2) yönlendirme olarak isimlendirilir. Seviye 2 yönlendirme yapılan bölge omurgayı (backbone) temsil eder. Seviye 1 yönlendirme ise omurga haricinde kalan bölgelerde yapılan yönlendirme şeklidir.

IS-IS protokolünde bulunan yönlendiriciler üç farklı şekilde isimlendirilirler. Bunlar;

- Seviye 1 Yönlendirici (Level 1 Router): Sadece bölge içi yönlendirme yapan yönlendiricilerdir. Başka bir bölgeye paket gönderileceği zaman kendisine en yakın seviye 1/2 yönlendiriciyi kullanarak işlemi gerçekleştirir.

- Seviye 2 Yönlendirici (Level 2 Router): Bu yönlendiriciler IS-IS protokolünde bulunan bölgeler arasındaki yönlendirmeyi sağlarlar. Omurga bölgesinde bulunan yönlendiricilerdir.

- Seviye 1/2 Yönlendiriciler (Level 1/2 Router): Bu yönlendiriciler OSPF protokolünde bulunan ABR‘ler ile benzerlikler gösterir. Seviye 1 ve seviye 2 veri tabanlarını kendi veri tabanlarında tutarlar [15].



Şekil 2.2. IS-IS Protokolü Bölge Mimarisi

2.2.2. IS-IS Protokolü Çalışma Prensibi

IS-IS protokolünde yönlendiriciler ara sistem olarak çalışmakta ev topolojideki diğer ara sistemler ile erişilebilirlik bilgisini paylaşırlar. Bu sürecin gerçekleşme şekli aşağıdaki gibidir.

- Her uç sistem (ES) farklı bölgelerde bulunurlar. OSI yönlendirmesi uç sistemler arasındaki merhaba (Hello) paketlerini dinlemeye başlar. Bu sayede en yakın ara sistem tespit edilir. Bir uç sistem, diğer bir uç sisteme paket yollamak istediği zaman, bu paketi kendisi ile aynı ağda bulunan ara sistemlerden birine yollar.

- Yönlendirici hedef adrese bakarak, paketi yollayabilecek en iyi yoldan gönderir. Eğer hedef uç sistem aynı alt ağda bulunuyor ise ara sistem bunu uç sistem merhaba paketlerinden öğrenir ve gerektiği şekilde paketi bu uç sisteme gönderir. Buna ek

olarak, kendisi üzerinden geçen yoldan daha uygun bir yol bulduğunu belirtmek amacıyla "tekrar yönlendir" (redirect) mesajı da gönderir.

- Eğer yönlendirme yapılacak uç sistem aynı bölgede farklı bir ağda bulunuyor ise yönlendirici yine bu uç sistemin nerde olduğunu bilir ve paketi uygun şekilde yönlendirir.

- Eğer hedef adres farklı bir bölgede yer alıyorsa seviye 1 yönlendirici (Level 1 IS) paketi en yakın ikinci seviye yönlendiriciye gönderir. Paket hedef bölgeye ulaşıncaya kadar seviye 2 yönlendirme devam eder. Eğer hedef adres farklı bir yönlendirme alanında yer alıyorsa, bu durumda üçüncü seviye yönlendirme yapılır [3].

- Hedef yönlendirme alanına ve ardından uygun bölgeye ulaşıldıktan sonra uç sisteme kadar yönlendirme devam eder.

2.3. BGP (Boarder Gateway Protocol)

Sınır geçit protokolü (BGP – Border Gateway), IETF (The Internet Engineering Task Force) tarafından Ocak 2006'da RCF 4276 ve RCF 4277 standardı ile tanınmış, otonom sistemler arasında yönlendirme için en yaygın olarak kullanılan protokoldür [3].

BGP protokolünde yönlendiricilere otonom sistem numaraları atanır. Otonom sistemlerde amaç IP networklerinin coğrafi konumlarına göre gruplara bölünmesi ve IP networklerinin yönetiminin kolaylaştırılarak sınırların çizilmesidir. Otonom sistem numaraları 1 ile 65535 arasında değişir ve 16 bitten oluşurlar. Otonom sistem numaralarında 64512 ile 65535 arası özel otonom sistem numaralarına ayrılmıştır. BGP otonom sistemleri bu numaralara göre ayırt eder [16]. Otonom sistem numaraları IANA (İnternet Assigned Numbers Authority) tarafından kontrollü olarak dağıtılır [3]. BGP, yönlendirme tablosunu oluşturmak için metrik hesaplamalarında, hedefe giderken üzerinden geçilen otonom sistem sayısını göz önüne alır.

2.3.1. BGP Mesaj Tipleri:

BGP topolojisinde toplam dört farklı mesaj kullanılmaktadır. Bu mesajlar yöneticiler arasında komşuluk kurulması ve komşuluğun devam ettirilmesi amacıyla kullanılırlar. Kullanılan bu mesaj tiplerini aşağıdaki gibi özetleyebiliriz.

Açılış (Open): Komşuluğun kurulması esnasında ilk gönderilen mesajdır. Açılış mesajının içeriğinde gönderen yönlendiriciye ait; versiyon numarası, otonom sistem numarası, kurulan TCP oturumunun askıda kalma süresi (Hold time), BGP yönetici kimliği (Router ID) ve isteğe bağlı parametreler bulunmaktadır.

Canlı Tutma (Keep alive): Her yönlendirici, diğerine oturum süresince aktif (canlı) olduğunu bildirmekle yükümlüdür. Bu durumun kontrolü canlı tutma mesajları ile gerçekleştirilir. Her 60 saniyede bir güncelleme mesajları yollarlar. 3 kere, 60 sn mesaj içeriği tamamlanırsa ardından bir daha "keep alive" mesajı yeniden güncelleme olmadığı sürece yollamazlar.

Güncelleme (Update): Komşu yönlendiricilere gönderilecek, yönlendirme bilgilerini içeren mesajdır. Her mesajda sadece bir patikaya ait bilgi gönderilir. Eğer bir komşuya birden fazla patika hakkında bilgi yollanacaksa her patika için ayrı güncelleme mesajı gönderilmesi gerekir. Network bilgisinin erişilebilirliğini kontrol eder. Yol erişim bilgisini tutar. Yönlendirmeleri yeniden yapılandırır.

Uyarı (Notification): Güncellemeler esnasında görülen hataları göndermek için kullanılan mesajlardır. Gönderildikten hemen sonra komşuluk sona erdirilir. Mesajla birlikte hata kodu ve hata ile ilgili bilgi gönderilir [17].

2.3.2. BGP Protokolü Çalışma Prensibi

BGP protokolü trafiği güvenli bir şekilde taşır. Trafik alışverişi gerçekleştirilmeden önce bir oturum açılarak TCP ile kontrol edilir ve aktif olarak çalıştığı gözlenir. TCP'nin 179 no'lu portundan bağlantı sağlanıp teyit alındıktan sonra trafik akışı sağlanır. Bu sayede TCP oturumu ile yönlendiriciler arası yönlendirme tabloları karşılıklı olarak paylaşılır. Tabloların değiştirilmesi sonlandırıldıktan sonra sadece güncelleme anlarında anons numaraları bir arttırılarak güncelleme yapılır. Bu aşamalar geçildikten sonra yalnızca değişiklik halinde yönlendirme bilgisi

güncellemeleri yapılır. Bunun dışında komşuluğun ilk kurulduğu andaki paylaşılan yönlendirme tablolarındaki bilgiler kullanılır.

BGP protokolü kullanılan ağlarda varılmak istenilen ağa gitmek için farklı yollar bulunabilir. Bu yollarda en verimlileri seçilerek yönlendirme tablosuna işlenir. Bu seçimin yapılmasında göze alınan bir çok kriter bulunmaktadır. BGP protokolü seçenekleri tek bir yola düşürünceye kadar eleme işlemine devam eder. BGP protokolünde seçim bant genişliği baz alınarak yapılmamaktadır. Aynı şekilde hedef ağa giden farklı yollardan yük dağılımı yapılarak da yönlendirme yapılmaz. BGP protokolünde en iyi yol hesabı aşağıdaki ölçütlere dikkat edilerek yapılır. Yollardan biri tercih edilinceye kadar bütün ölçütler sırasıyla karşılaştırılır. Eğer hedefe ulaşılamıyorsa değerlendirilmeye alınmaz.

1. En yüksek yerel tercih değerine (local preference) sahip olan yolun önceliği vardır.

2. Hedef ağa ulaşabilmek için en az otonom sistem geçilmesini sağlayan yolun önceliği bulunmaktadır.

3. Kaynağı düşük olan yolun önceliği bulunmaktadır. Eğer bir yol dahili yönlendirme protokolünden öğrenildiyse 0, harici yönlendirme protokolünden öğrenildiyse 1, tam değil (incomplete) ise 2 değeri verilir. Eğer IBGP komşuluğundan öğrenilen bir yol bilgisi dahili yönlendirme protokolünden de öğrenildiyse, bu bilgi EBGP komşuluğuyla paylaşılmaz ve bu bilgi kullanılarak bir EBGP yönlendiricisine paket gönderilmez.

4. Düşük MED (Multi-Exit Discriminator) metriğine sahip olan yol tercih edilir.

5. EBGP komşuluğu kurulmuş olan yönlendiricilerden öğrenilen yolların, IBGP‘den öğrenilen yollara göre önceliği mevcuttur.

6. Dahili yönlendirme protokolü metriği daha düşük olan komşudan öğrenilen yol tercih edilir.

7. BGP kimlik numarası daha düşük olan yönlendirici tercih edilir.

BGP toplulukları (Community) aynı özellikteki yollar kümesi olarak tanımlanabilir. Aynı yönlendirme politikalarını aynı özellikteki yollara uygulanmasını sağlar. BGP topluluk kimlikleri iki adet 16 bitlik sayıyı birleştirilmesiyle oluşan 32 bitlik sayı ile ifade edilirler. Yazım şekli; Lokal otonom sistem:XX şeklindedir. Lokal otonom sistem değeri, topluluğun hangi otonom sistem içinde dolaştığını belirtir ve 0 ile 65535 arasında değer alır. XX değeri ise topluluk değeridir. O da 0 ile 65535 arası bir değer alır. 0:0 – 0:65535 ile 65535:0 – 65535:65535 arasında kalan BGP topluluk kimlik numaraları rezerve edilmiştir ve internette kullanılmazlar.

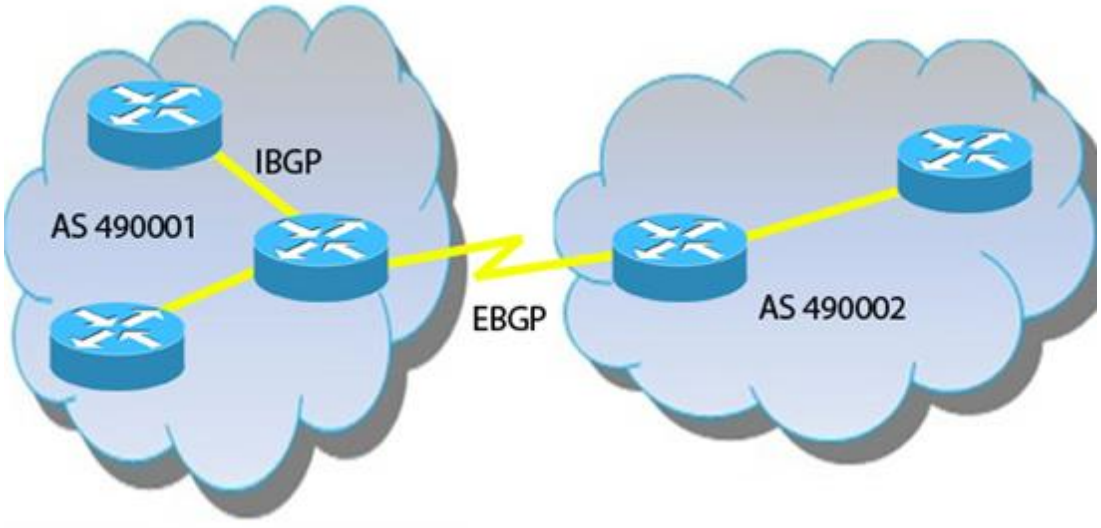
Bazı topluluk numaraları ise iyi bilinen (well known) olarak adlandırılırlar. Yönlendiriciler bu numaralara sahip yollar için nasıl davranmaları gerektiğini bilirler.

Bu toplulukları aşağıdaki gibi sıralanabilir;

- No-export (65535:65281): Hiçbir EBGP komsuna anons etme.
- No advertise (65535:65282): Hiçbir BGP komsuna anons etme.
- Local-as (65535:65283): Lokal otonom sistemin dışına anons etme.

Aynı özellikleri taşıyan ağlara, aynı politikaları uygulamak amacıyla tasarlanan ve büyük kolaylıklar getiren BGP toplulukları otonom sistem içinde yapılan operasyonel işlemleri de azaltmaktadır.

IBGP & EBGP: BGP'de bulunulan otonom sisteme göre iki şekilde komşuluk kurulmaktadır. Bunlar; dahili komşuluk ve harici komşuluk olarak adlandırılmaktadır. Komşuluk kurulacak yönlendirici aynı otonom sistemde yer alıyorsa dahili, farklı otonom sistemde yer alıyorsa harici komşuluk kurulmaktadır. Kurulan komşuluğa göre de oturum dahili BGP (IBGP - Internal Border Gateway Protocol) ve harici BGP (EBGP – External Border Gateway Protocol) olarak adlandırılır.



Şekil 2.3. IBGP ve EBGP Alanları

EBGP komşuluklarında yönlendiriciler birbirlerine direkt bağlı olmak zorundadırlar ancak aynı otonom sistemde bulunmazlar. Herhangi bir yönlendirme protokolü komşuluk kurulacak yönlendiriciyi aramak zorunda olmamalıdır. İlk yapılandırma sırasında yönlendiricilere, komşuluk kuracakları yönlendiricilerin IP adresleri belirtilir.

IBGP komşuluklarında ise durum farklıdır. Komşuluk kurulacak yönlendiricinin direkt bağlı olmasına gerek yoktur. Bunun nedeni yönlendiricilerin aynı otonom sistemde bulunmalarıdır. Yönlendiriciler, komşuluk kuracakları yönlendiricinin yerini otonom sistemde çalışmakta olan dahili yönlendirme protokolü yardımıyla, statik rotalarla veya direkt bağlı oldukları ara yüzlerle bulabilirler.

Genellikle otonom sistemin transit ağ olarak kullanılması (Otonom sistemin taşıyıcı olarak görev alması) durumlarında IBGP'nin omurgada kullanılması tavsiye edilmemektedir. Bunun dışındaki durumlarda IBGP kullanılmasından kaçınılmalıdır. Bu gibi durumlarda IBGP kullanılmazsa otonom sistemler arasında yönlendirme bilgilerinin taşınması için EBGP yönlendirme protokolü tablolarının tamamının, dahili yönlendirme protokolü tablolarına aktarılması gereklidir. Böyle bir durum otonom sistem içindeki yönlendiricilerin performansında ciddi şekilde düşüşe sebebiyet verecektir.

BGP protokolünde yönlendirme hedefi anonsların yapıldığı yöne doğrudur. Dahili yönlendirme protokolündeki gibi IP adreslerini görerek yönlendirme yapmaz. IBGP ile çalışan yönlendiricinin diğer yönlendiriciler ile komşuluk kurması BGP protokolünün çalışma prensibidir. Bu sayede yönlendirme döngülerinin ve kara deliklerin (Black holes) oluşmasının önüne geçilir. IBGP, yönlendiricilerin tümüyle komşuluk kurduktan sonra farklı bir IBGP yönlendiricisinden alınan yönlendirme güncellemesi, diğer bir IBGP yönlendiricisine anons edilmez. Bunun haricinde bir yönlendirici diğerlerinin tümüyle komşuluk kurmazsa yönlendirme tabloları tam olarak senkron olmayacak ve bu durum yönlendirmede kayıplara sebebiyet verecektir.

Rota Yansıtma (Router Reflection) Mekanizması: IBGP protokolü kullanırken yönlendiricilerin aralarında komşuluk kurmalarının getirdiği zorlukları aşmak amacıyla Rota Yansıtma (Route Reflection) mekanizması geliştirilmiştir. Rota yansıtma mekanizmasına göre hiyerarşik tasarım ve yönlendirici grupları oluşturulur. Genel olarak çalışma şeklini aşağıdaki gibi anlatabiliriz;

- Yansıtıcı vasfındaki yönlendiriciler diğer bütün yansıtıcı yönlendiricilerle IBGP komşuluğu kurarlar.

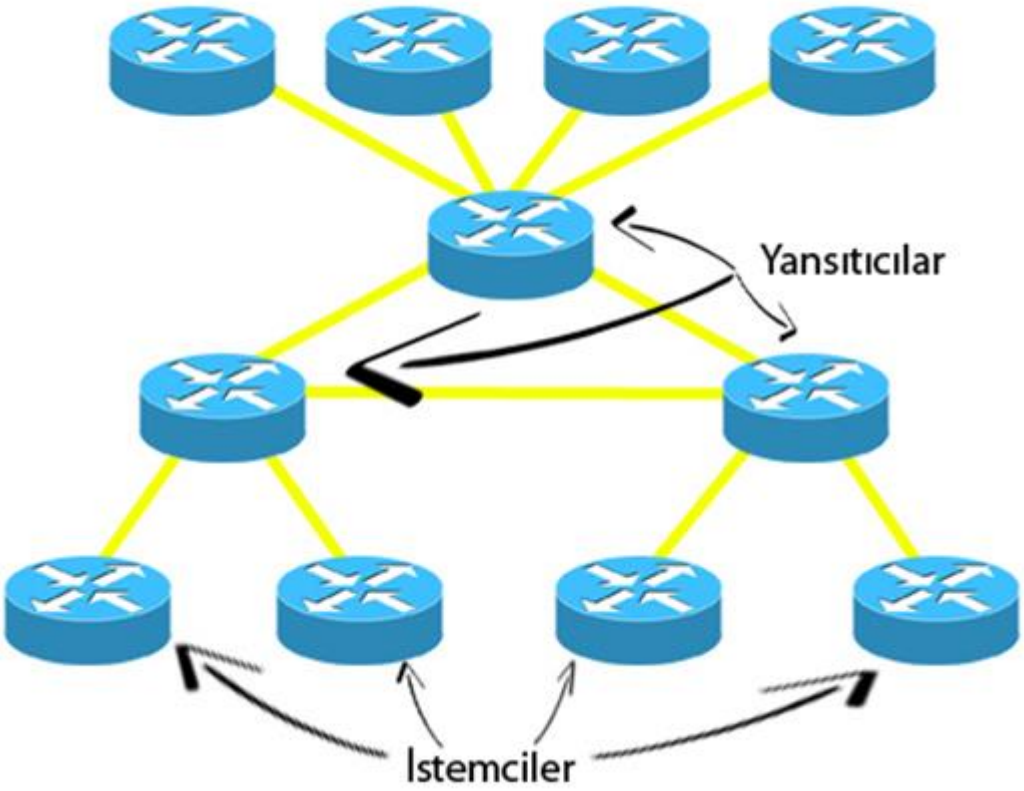
- Yansıtıcılar istemcilerden ve istemci olmayanlardan rota bilgilerini alırlar.

- En iyi yol seçimi yapılır.

- Eğer yol istemci olan yönlendirici üzerinde ise bu bilgi tüm yönlendiricilerle paylaşılır.

- Eğer yol istemci olan yönetici üstünde değil ise, bu bilgi istemci olan yönlendiriciye bildirilir.

Otonom sistem içerisinde birden çok yansıtıcı kullanılabilir. Yedeklilik düşünülüyorsa iki tane kullanılması yeterli olacaktır. Bununla beraber bir yönlendirici aynı anda bu iki yansıtıcının istemcisi olması yine yedekliliği sağlamak açısından önem taşımaktadır. Rota yansıtıcıları kullanımının avantajları arasında uygulanmasının kolay olması ve paket yönlendirmesini etkilememesi gösterilebilir. Rota yansıtıcıları BGP'yi ölçeklenebilir kılan önemli bir özelliktir.



Şekil 2.4. IBGP Protokolündeki Yansıtıcılar ve İstemciler

BGP Protokolü farklı otonom sistemler arasında kullanıldığı için bilgi amaçlı verilmiş olup bu projede daha fazla yer almayacaktır.

Yukarıda anlatılan bu üç protokol servis sağlayıcıları tarafından sıklıkla kullanılan protokoller arasındadır. Artan trafik miktarlarına getirdiği çözümler ve ağ mimarilerinin işletilmesinde sağladığı kolaylıktan ötürü servis sağlayıcıları MPLS teknolojisini topolojilerinin çok büyük bir çoğunluğunda kullanmaya başlamışlardır. MPLS ile kullanılan mimarilerin detaylı şekilde anlatılmasının ardından, MPLS kullanan topolojilerin, MPLS siz topolojilerle kıyaslamasına geçilecektir.

3. BENZETİM ARAÇLARI

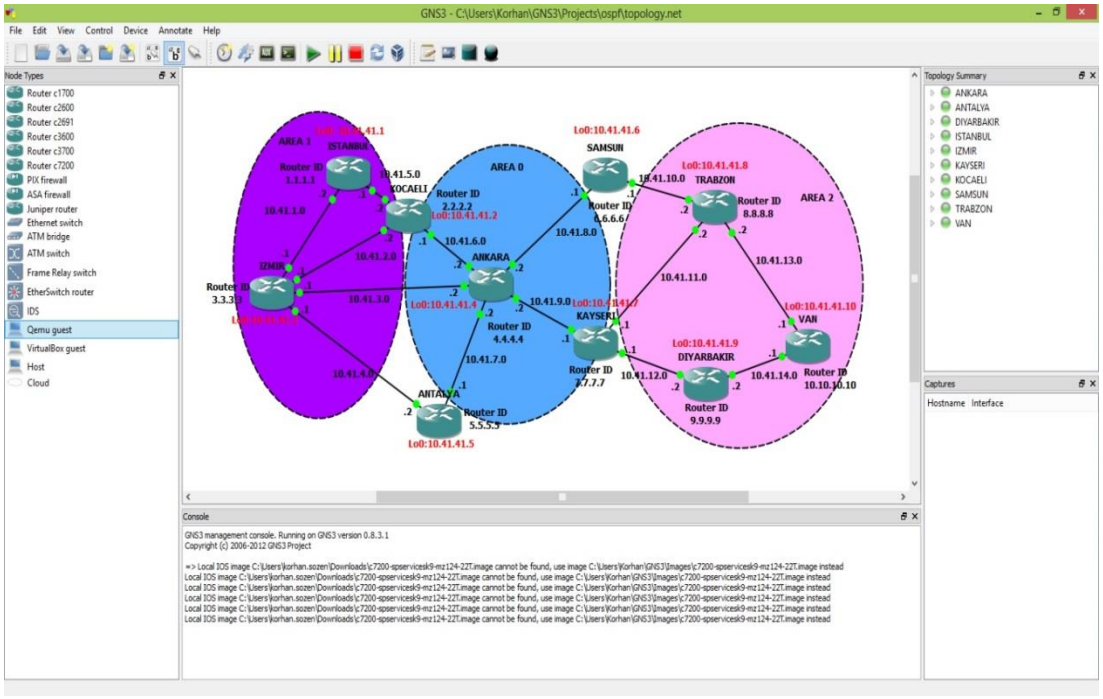
Projeye başlamadan önce, projede kullanılan programlara detaylı bir şekilde değinmek projenin anlaşılabilirliği açısından önem taşımaktadır. Bu bölümde projede kullanılan GNS3, Wireshark, Oracle VM Virtual Box ve Jperf isimli dört adet benzetim aracına değinilecektir.

3.1. GNS3

İşbirl GNS3 (Graphical Network Simulator 3) gelişmiş ağları simüle edebilen açık kaynaklı (open sorce) bir simülatör programıdır. Dynamips altyapısı sayesinde Cisco ve Juniper gibi Dünyanın önde gelen yönlendirici markalarının IOS'larını (İnternetwork Operating System) yükleyerek gerçeğe en yakın verilerin elde edilmesini ve gerçek zamanlı çalıştırılıp test edilmesini sağlar [18].

Geliştirilmiş yapısıyla karmaşık mimarilerin gerçeğe en yakın şekliyle simüle edilmesine yardımcı olan GNS3 farklı açık kaynak programlarla bütünleşik çalışabilmesiyle de tercih edilmektedir.

En düşük fiyatlı yönlendiricinin dahi 1000$ değerinde olduğu düşünüldüğünde network mühendislerinin deney amaçlı fiziki bir ağ mimarisi kurmaları çok zordur. Bu nedenle GNS3, dünya üzerinde açık kaynaklılığı ve direkt firmaların IOS'larını kullandığından ötürü gerçeğe en yakın sonuçları veren ve birçok kişi tarafından rağbet gören bir network programdır.



Şekil 3.1. GNS3 Genel Görünüm

GNS3'ün ana sayfasında 5 adet yönetim paneliyle beraber üst bölümde çeşitli işlemleri kolay yoldan yapmamıza yardımcı olan bir menü çubuğu bulunmaktadır ve bu menü üç kısımdan oluşmaktadır. Birinci kısım proje ve topolojilerle ilgili olan kısımdır. Bu kısımda yeni bir proje/topoloji yaratmak, var olan proje/topolojiyi açmak, projeleri/topolojileri kaydetmek ve tasarlanan projeler/topolojiler içerisindeki bağlantıların sağlanması için gerekli olan kablo seçimlerini yapabildiğimiz butonlar bulunmaktadır.

İkinci kısım IOS'u girilmiş yönlendiriciler ile ilgili olan kısımdır. Yönlendiricilerin çalışması, durdurulması, yeniden başlatılması, yönlendiricilerin console ekranının açılması gibi işlemler için ayırılan kısımdır.

Üçüncü ve son kısım ise çeşitli çizim ve yazı işlemlerinin yapılmasına yardımcı olan kısımdır. Çeşitli notların alınması, şekillerin çizilerek ağ alanlarının belirlenmesi gibi işlemler burada bulunan kısaltmalar sayesinde yapılır.

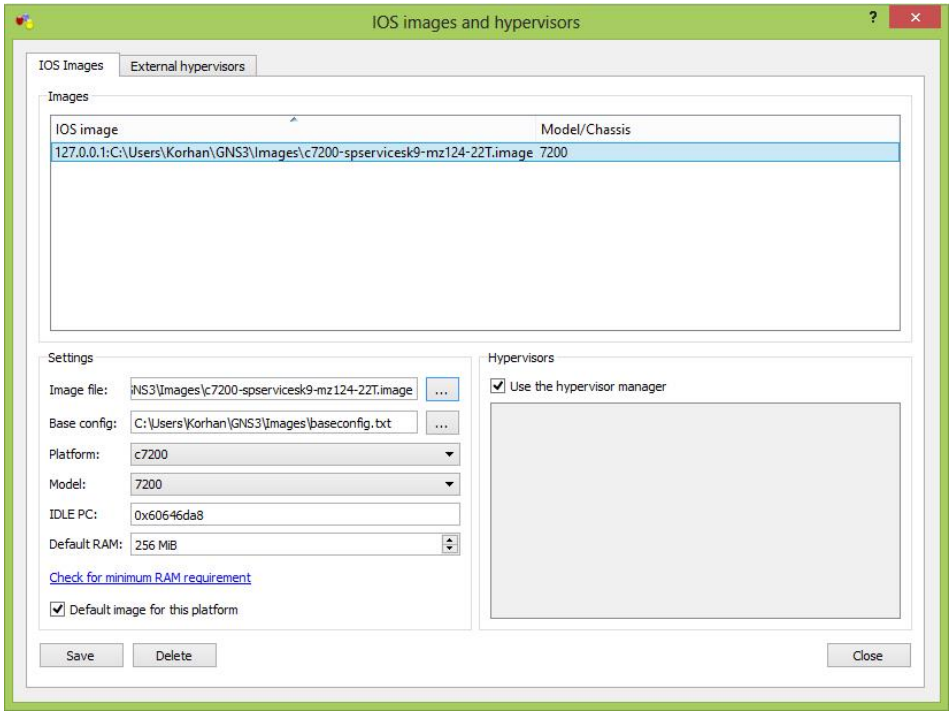


Şekil 3.2. GNS3 Menü Çubuğu

Menü çubuğunun hemen aşağısında bulunan yönetim panellerinin isimleri şu şekildedir;

- Main Screen (Ana Ekran)
- Node Types (Düğüm Çeşitleri)
- Topology Summary (Topoloji Özeti)
- Captures (Yakalananlar)
- Console (Konsol)

Main Screen (Ana Ekran): Projenin hazırlandığı bölümdür. GNS3'e "IOS İmage and Hypervisors" ekranıyla entegre edilen yönlendirici IOS'larının ağ çalışmasına dahil edildiği yerdir.

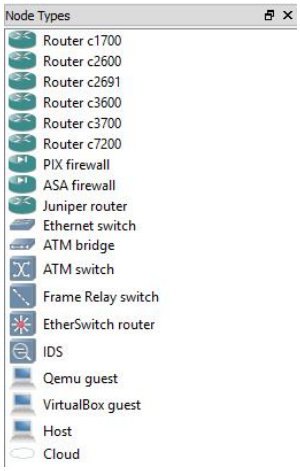


Şekil 3.3. GNS3 IOS image and Hypervisors Ekranı

IOS'ları indirilen yönlendiriciler "IOS İmage and Hypervisors" ekranıyla, GNS3 tarafından desteklenen yönlendirici modellerine göre imajları programa dahil

edilerek kullanılır. Sorunsuz olarak kullanılabilir hale getirilen yönlendirici imajları ana ekrana atılarak istenilen topolojinin oluşturulmasına başlanır.

Node Types (Düğüm Çeşitleri): İmajı GNS3'e entegre edilen yönlendiricilerin listesinin bulunduğu kısımdır. Bu kısımdan ilgili yönlendiriciler sürükle bırak yöntemiyle projeye dahil edilirler.



Şekil 3.4. GNS3 Node Types Paneli

Topology Summary (Topoloji Özeti): Oluşturulan topolojide bulunan sanal makinelerin listesinin bulunduğu kısımdır. Bu panelin sağladığı en büyük kolaylık karışık topolojilerde rahatlıkla sanal makinelerin seçiminin yapılabilmektedir. Kullanılan sanal makineler liste halinde bu kısımda bulunmaktadır.



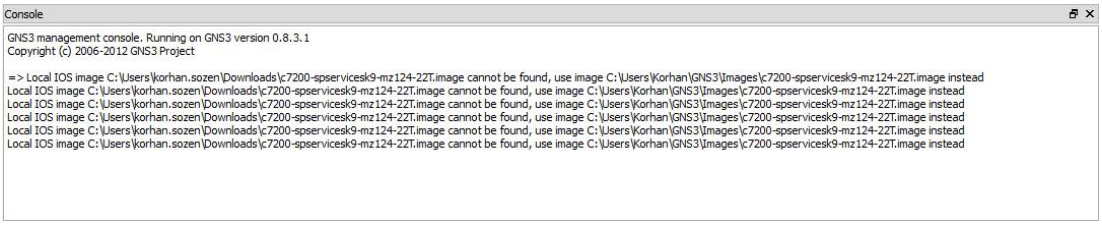
Şekil 3.5. GNS3 Topology Summary Paneli

Captures (Yakalananlar): Captures paneli, GNS3 ile entegre çalışabilen Wireshark programının yönlendirici ara yüzlerindeki paket alışverişlerini izlemek için çalıştırılmış wireshark uygulamalarının listelendiği kısımdır. Bu kısımda istenilen izlenime kolay şekilde ulaşılabilir.



Şekil 3.6. GNS3 Capture Paneli

Console (Konsol): Console paneli program üzerinde yapılan değişikliklerin loglarının gösterildiği yerdir. Örnek olarak; projenin ilk açılışında yönlendiricilerin doğru yüklenip yüklenemediğini veya kayıt edilecek topolojin kaydının doğru şekilde yapılıp yapılamadığını göstermesini verebiliriz.



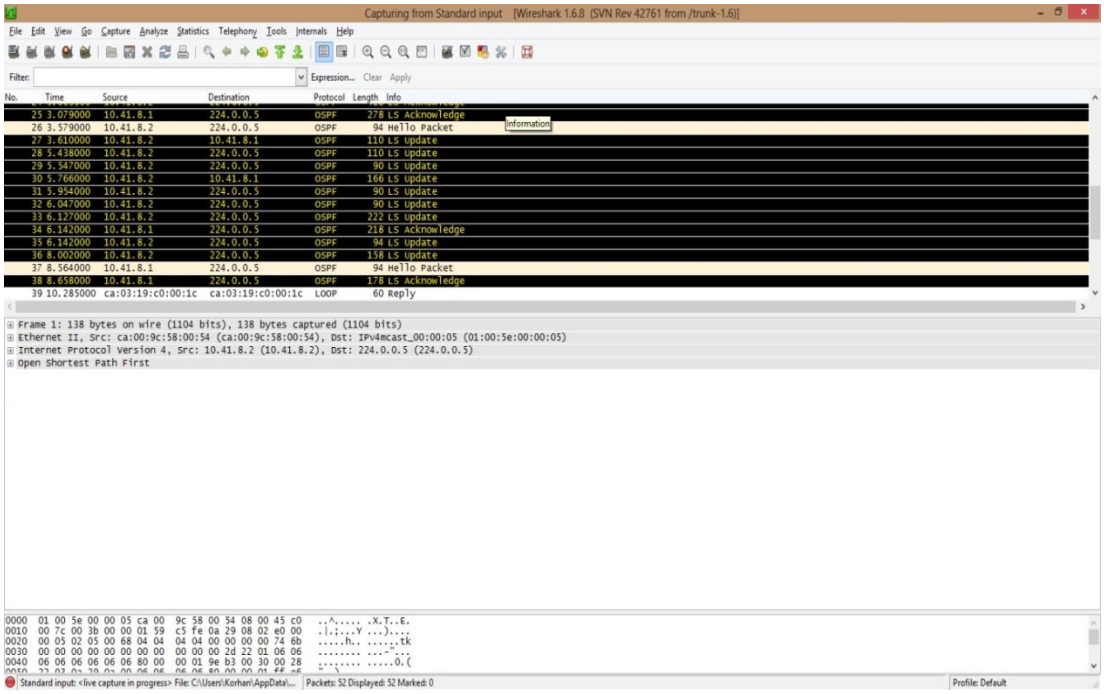
Şekil 3.7. GNS3 Console Paneli

GNS3 geliştirilmiş yapısıyla karmaşık topolojilerin oluşturulmasına ve gerçeğe yakın şekilde çalıştırılmasına imkan vermektedir. Ayrıca tercih edilmesinin en büyük nedenlerinden birisi de bu projede de kullanılan wireshark ve virtual box gibi programlarla entegre çalışabilmesidir.

3.2. Wireshark

Wireshark, 1998 yılında çalışmalarına başlanarak ağ protokol analizi yapabilen ve birçok kişi tarafından tercih edilen paket izleme programıdır [19]. Wireshark, bilgisayarımıza ulaşan paketleri yakalayabilir ve bu paketleri görüntülememize olanak sunar. Başka bir deyişle, her türlü ethernet kartlarındaki TCP/IP mesajlarını analiz edebilen bir programdır [20].

Wireshark'ın tercih edilmesinin en önemli sebeplerinden biri açık kaynaklı yapıya sahip olması nedeniyle dünyanın çeşitli yerlerindeki ağ uzmanlarının her geçen gün programa yenilikler ekleyebilmesidir [21].



Şekil 3.8. Wireshark Genel Görünüm.

Wireshark'ın ana ekranına baktığımızda beş adet menü görünmektedir. Bunlar;

- Main Tool Bar (Ana Araç çubuğu)
- Filter Tool Bar (Filtre Araç Çubuğu)
- Packet List (Paket Listesi)
- Packets Details (Paket Detayları)

- Packet Bytes (Paket Byteları)

Olarak adlandırılırlar.

Main Tool Bar (Ana Araç Çubuğu): Wireshark'ın sık kullanılan fonksiyonlarının kısa yollarının bulunduğu ve kullanıcıya göre ayarlanabilen kısımdır. Genel olarak üç gruba ayrılmıştır. Birinci grup izlenecek ara yüzler ile ilgili olan kısımdır. Ara yüzlerden geçen paketlerin izlenimine başlanması, durdurulması, geri yüklenmesi gibi aktivitelerle beraber izlenilen paketlerin kaydedilmesi, geri yüklenmesi gibi aktiviteler de bu bölümden yapılmaktadır. İkinci grupta paket listesi ile ilişkili kısa yollar bulunmaktadır. Farklı paketlerin renklendirilmesi, paketler arası geçiş, ilk paket, son paket seçimi gibi işlemler bu gruptan yapılmaktadır. Son grupta ise programın görünümü ve biçimlendirilmesi ile ilgili kısa yollar bulunmaktadır.



Şekil 3.9. Wireshark Main Tool Bar

Filter Tool Bat (Filtre Araç Çubuğu): Yakalanan paketlerin istenilen şekilde ayırılmasına yardımcı olan bölümdür. Listelenmek istenilen paketler arama kısmına yazılarak filtreleme gerçekleştirilir.



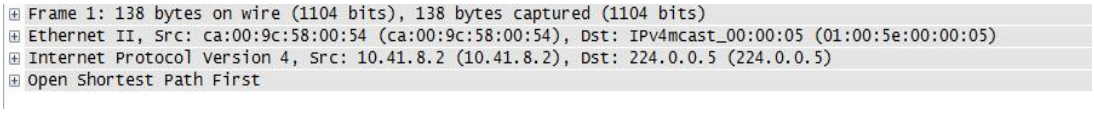
Şekil 3.10. Wireshark Filter Tool Bar

Packet List (Paket Listesi): Yakalanan paketlerin görüldüğü yerdir. Yakalanan paketlerin her biri için, bir satırlık özet bilgi sunar. Farklı paketleri farklı renklerle göstermektedir. Yakalanan paketlerin içine girildiğinde paket detayları kısmına detaylı bilgiyi aktarır.



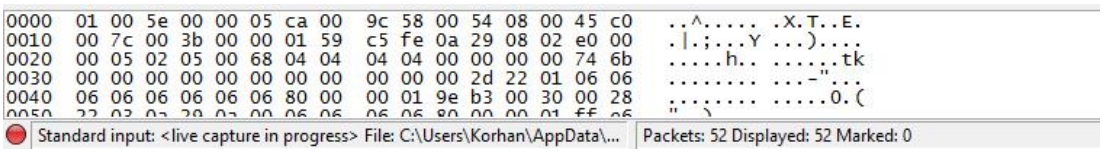
Şekil 3.11. Wireshark Packet List

Packet Details (Paket Detayları): Paket listesi kısmında seçili olan paket hakkında detaylı bilgilerin bulunduğu alandır. Paket listesi kısmında çerçeve, kullanılan protokol, adres bilgisi gibi yakalanan paket hakkında detaylı bilgiler bulunmaktadır.



Şekil 3.12. Wireshark Packet Details

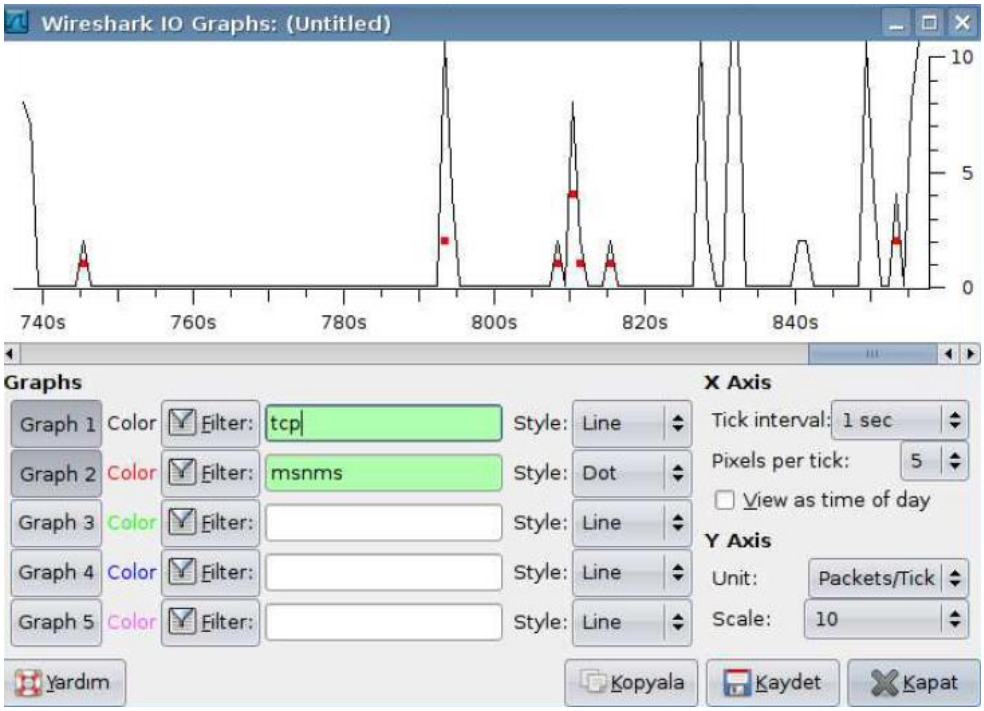
Packet Bytes (Paket Byteları): Paket listesi kısmında seçili olan paketin verisi hakkında bilgi veren bölümdür. Yakalanan paketlerin verilerini Hexa decimal olarak gösterir.



Şekil 3.13. Wireshark Packet Bytes

Bu menülere ek olarak bir de yakalanan paketlerin grafik şeklinde görülmesini sağlayan Giriş/Çıkış Grafiği (IO Graphs) seçeneği mevcuttur.

IO Graphs (Giriş/Çıkış Grafiği): IO Graphs yakalanan paketlerin (zamana göre) grafik olarak görünmesine yardımcı olur. Ağda durum kontrolü için oldukça faydalı bir özelliktir. Grafik sayesinde ağda olan anormallikler anında tespit edilebilir. Grafiği çıkarırken ister filtre (filter) kısmından hazır filtreleri, ister bizim seçtiğimiz paketlere göre filtrelemeleri yapabilir. Grafik çiziminin nasıl olacağını (Çizgi, nokta v.b) belirleyebilir, farklı paketleri renklendirebiliriz.

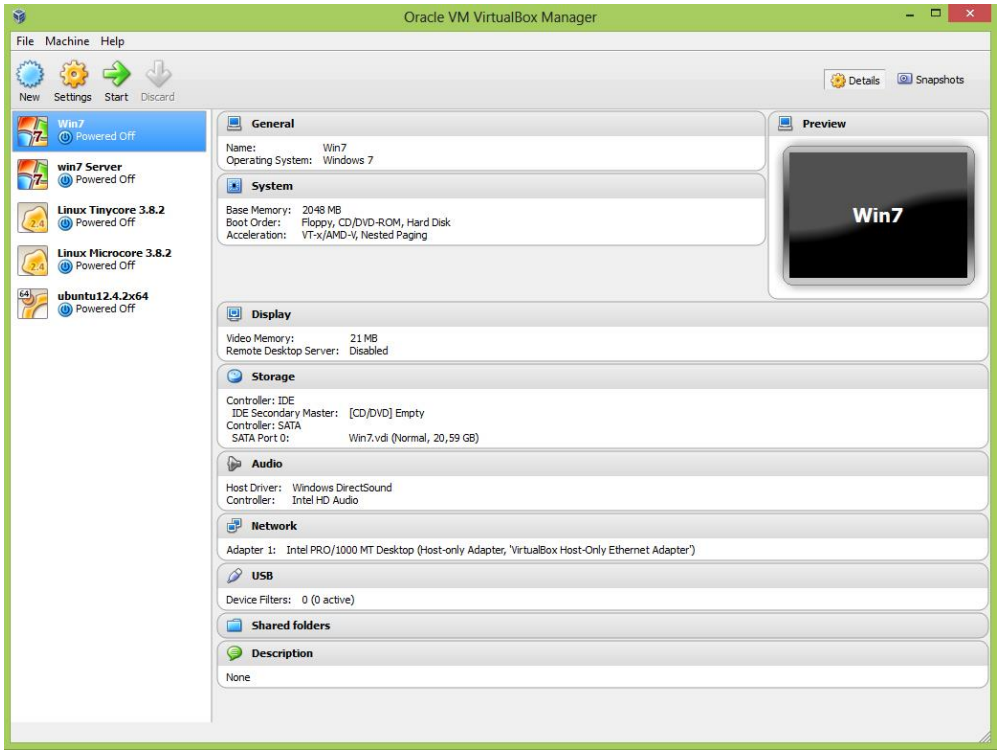


Şekil 3.14. Wireshark IO Graph

3.3. Oracle VM Virtual Box

Virtual Box, kişisel kullanım için Sun Microsystems tarafından geliştirilmiş olan açık kaynak sanal makine oluşturma yazılımıdır. Virtual Box'un en önemli özelliği farklı işletim sistemlerinde çalışabilmesi ve farklı misafir işletim sistemlerini desteklemesidir [22].

Kullanılacak sanal makinenin 32-64 bit olması veya eski veya yeni bir işletim sistemi olması fark etmez. Bu durum Virtual Box'un en önemli tercih sebeplerinden birisidir.

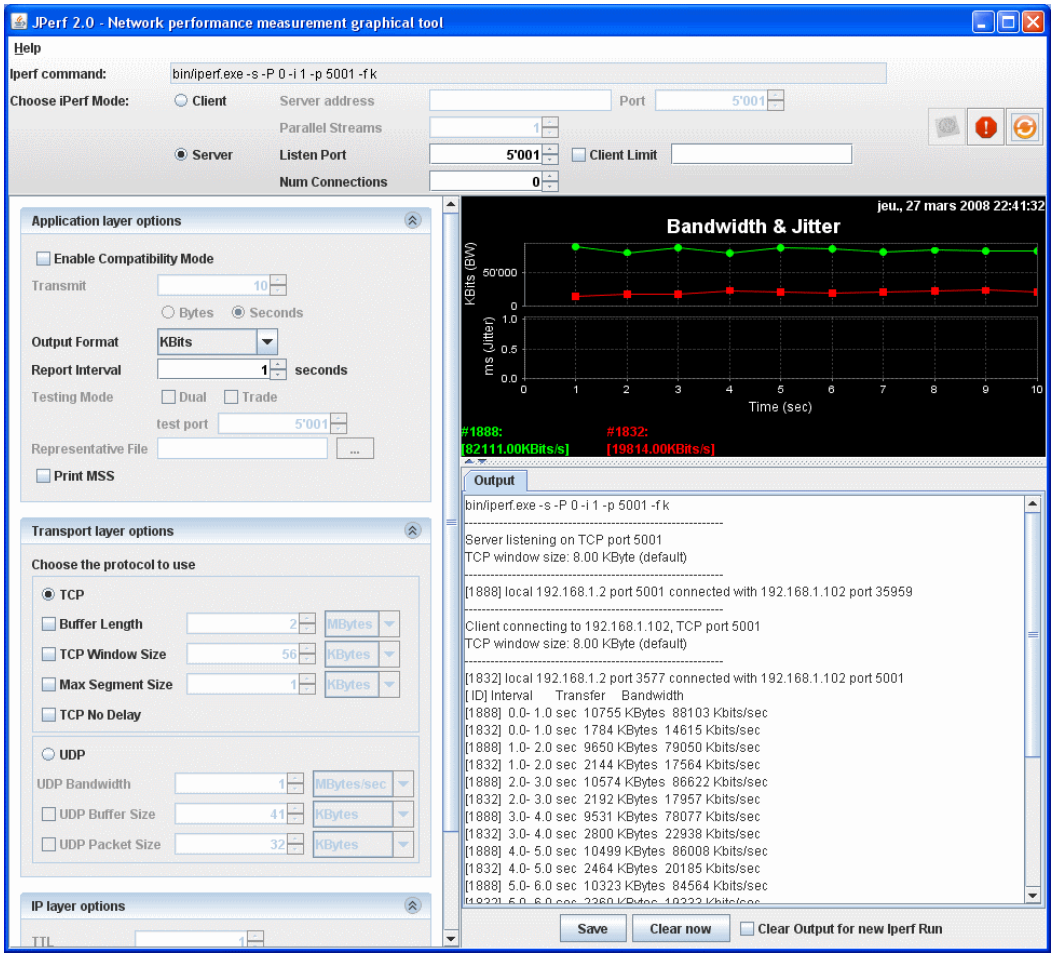


Şekil 3.15. Oracle VM Virtual Box

Oracle VM Virtual Box'ın ana ekranına baktığımızda üstte yeni sanal makine kurulumu, sanal makinenin ayarları, sanal makinenin başlatılması ve sanal makinenin silinmesi için gerekli olan butonlar bulunmaktadır. Ekranın sol tarafı yaratılan sanal makinelerin listesine ayrılmıştır. Bu listeden yaratılan sanal makineler seçilerek işletilmeye başlanabilir. Ekranın geriye kalan kısmında ise seçilen sanal makinenin ayrıntılarını gösteren kısım bulunmaktadır. Bu kısımda makinenin işletim sistemi, sistem özellikleri, hafızası, ses ayarları, network ayarları, USB ayarları, paylaşım klasörleri hakkında bilgi bulunmaktadır. Ayrıca yine bu kısımda bulunan ayarların içine girilerek istenilen değişiklikler yapılabilir.

3.4. JPERF 2.0

Jperf, bant genişliğini ve ağ kalitesini ölçmek için kullanılan grafik ara yüzlü, java tabanlı trafik ölçüm programıdır. Jperf birçok network uzmanı tarafından kullanılmaktadır. Bunun en büyük nedenlerinden biri açık kaynaklı bir uygulama olmasıdır [23].



Şekil 3.16. Jperf 2.0

Jperf ile bağlantı kalitesi dört şekilde test edilebilir;

- Gecikme (Latency): Tepki süresi veya RTT (Round Trip Time) ping komutu ile ölçülebilmektedir.

- Seğirme (Jitter): Paketlerin gecikme sürelerindeki varyasyon. Gecikme varyasyonu. UDP testi ile ölçülebilmektedir.

- Datagram Kaybı (Datagram Lost): UDP testi ile ölçülebilmektedir.

- Bant Genişliği (Bandwidth): TCP testi ile ölçülebilir.

Jperf ağ bağlantıları ile ilgili istatistiksel verileri çıkarırken TCP ve UDP'nin farklı kapasitelerini kullanır [23]. Jper programını kullanabilmek için birbirini ağ üzerinde

görebilen 2 ayrı makinada server (sunucu) ve client (alıcı) olarak çalıştırılması gerekmektedir.

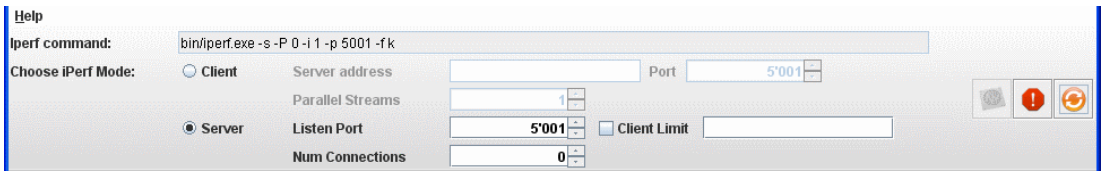
Jperf ana ekranı 6 bölümden oluşmaktadır. Bunlar;

- Ayarlar (Settings)
- Uygulama Katmanı Seçenekleri (Application Layer Options)
- Ulaşım Katmanı Seçenekleri (Transport Layer Options)
- IP Katmanı Seçenekleri (IP Layer Options)
- Grafik Ekranı (Graphic Screen)
- Çıktı (Output)

dır.

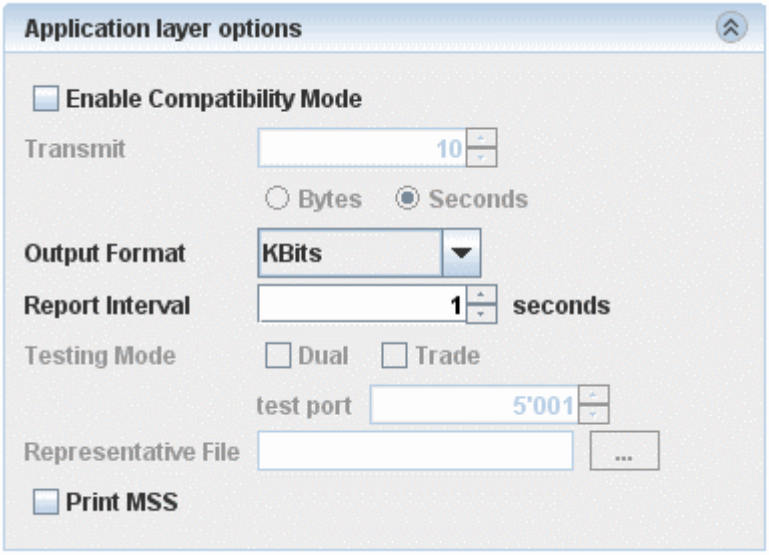
Ayarlar (Settings): Ana ayarların yapıldığı bölümdür. Jperf'in sunucu olarak mı, alıcı olarak mı kullanılacağı bu bölüm ile netleştirilir. Jperf alıcı olarak kullanılacaksa; sunucu adresi ve portu bu kısımdan girilmektedir. Şayet Jperf sunucu olarak kullanılacaksa; dinleme portu, alıcı limiti ve bağlantı sayısı yine bu kısımdan belirlenmektedir.

Uygulamanın başlatılması, sonlandırılması ve başlangıç ayarlarına dönülmesi bu bölümden gerçekleştirilmektedir.



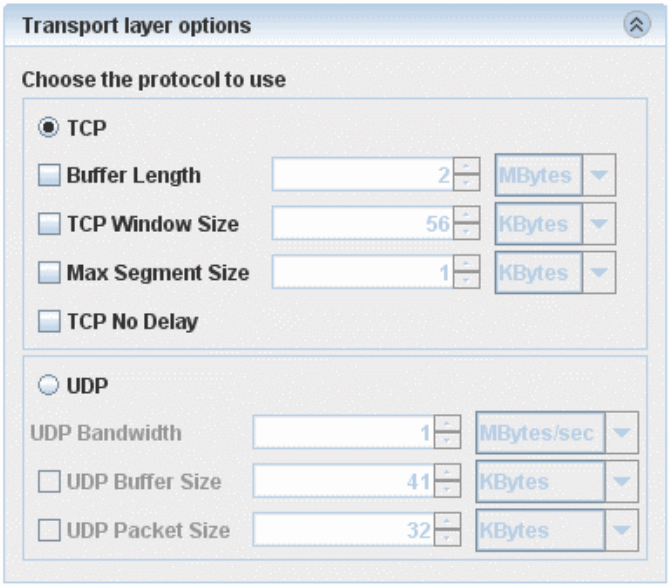
Şekil 3.17. Jperf Settings

Uygulama Katmanı Seçenekleri (Application Layer Options): Bu bölümde çıktı ekranına gönderilen paketler ile ilgili genel ayarlamalar bulunmaktadır. İletimin ne kadar süreceği, gönderilen paketlerden kaç saniyede veya byteda örnek alacağı, alınacak raporun formatı gibi seçenekler buradan belirlenir.



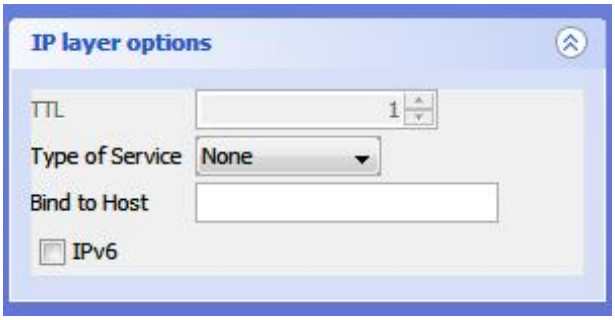
Şekil 3.18. Jperf Application Layer Options

Ulaşım Katmanı Seçenekleri (Transport Layer Options): Bu bölümde testin yapılacağı protokol (TCP/UDP) ile ilgili bilgiler seçilmektedir. Arabellek Uzunluğu (Buffer Length), Pencere Uzunluğu (Window Size), Segment Boyutu (Segment Size), Paket Boyutu (Packet Size) gibi seçeneklerin belirlendiği bölümdür.



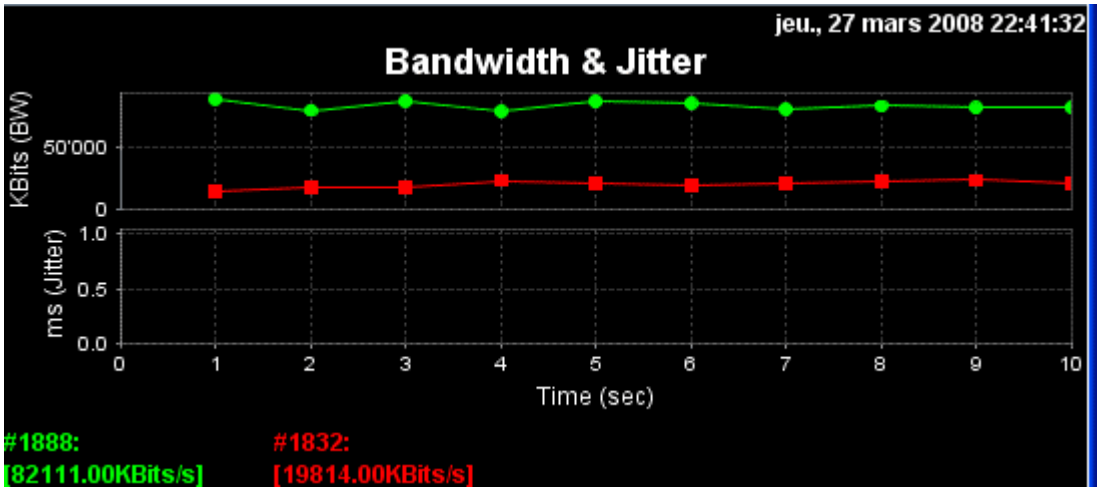
Şekil 3.19. Jperf Transport Layer Options

IP Katmanı Seçenekleri (IP Layer Options): Bu bölümde genel olarak IPv6 ile ilgili testler için seçenekler bulunmaktadır. Paketlerin yaşam süresi (TTL – Time To Live), servis tipi gibi seçenekler bu bölümde belirlenir.



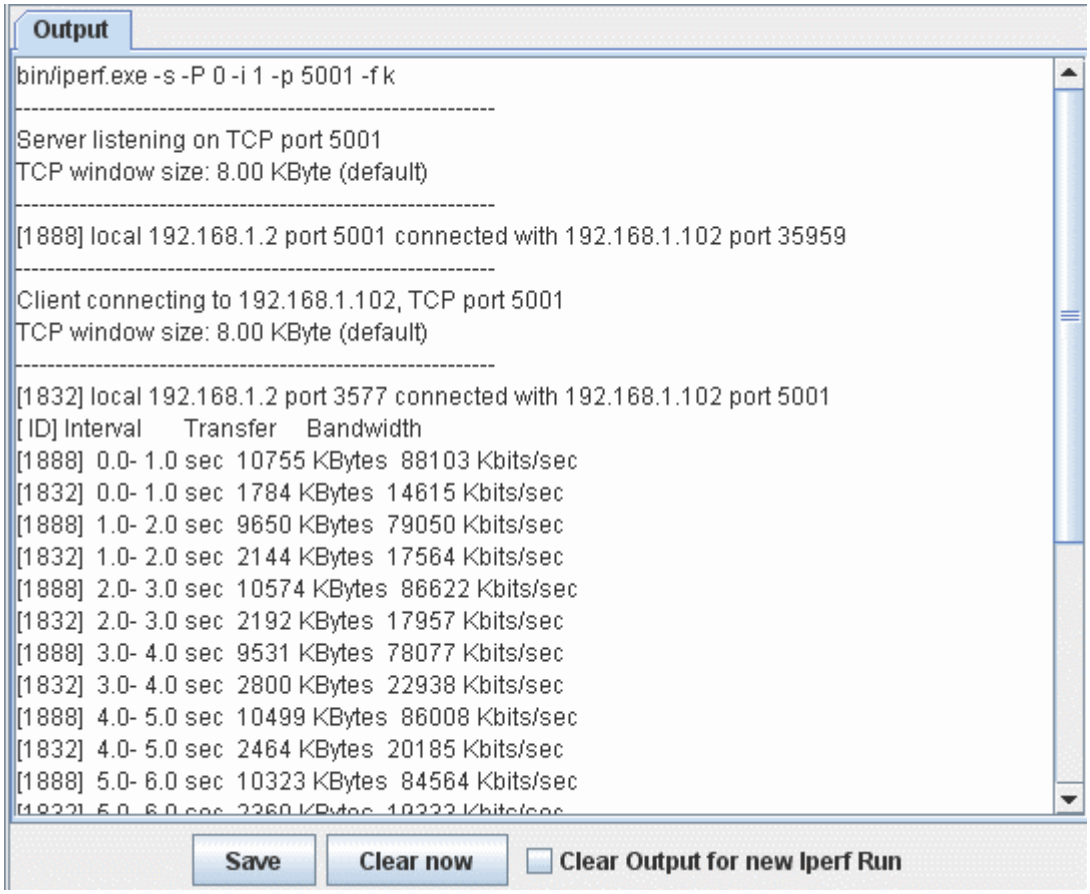
Şekil 3.20. Jperf IP Layer Options

Grafik Ekranı (Graphic Screen): Diğer bölümlerden girilen ayarlamalara göre gelen/gönderilen paketlerin grafiğinin çıkarıldığı kısımdır.



Şekil 3.21. Jperf Graphic Screen

Çıktı (Output): Diğer bölümlerden girilen ayarlamalara göre gelen/gönderilen paketlerin verileri hakkında bilgi veren bölümdür.



Şekil 3.22. Jperf Output

4. BENZETİM SONUÇLARI

Bu tez çalışmasında benzetim araçları yardımıyla, OSPF ve IS-IS protokolleri kullanılarak oluşturulan topolojiler üzerinden MPLS'in kullanıldığı ve kullanılmadığı durumlarda farklı trafikler geçirilerek etiketleme teknolojisinin ağ trafiğine etkisinin analizi yapılmıştır.

GNS 3 desteğiyle oluşturulan topolojide gerçek değerlere en yakın sonuca ulaşabilmek adına 10 adet Cisco 7200 Servis Sağlayıcı (Service Provider) yönlendirici kullanılmış ve bu yönlendiricilere Cisco PA-GE Kartları (Gigabit Ethernet destekleyen kartlar) tanımlanarak altyapının, bir servis sağlayıcı altyapısına benzetilmesi sağlanmıştır.



Şekil 4.1. Cisco 7200 Service Provider Router

Tüm yönlendiriciler birbirine fiber optik kablolar ile bağlanmıştır. Testin yapılabilmesine olanak sağlayan ve Oracle VM Virtual Box desteğiyle oluşturulan sanal makineler en uçta bulunan iki yönlendiriciye Cisco PA-2FE-TX kartları (Fast Ethernet destekleyen kartlar) tanımlanarak bağlanmıştır.

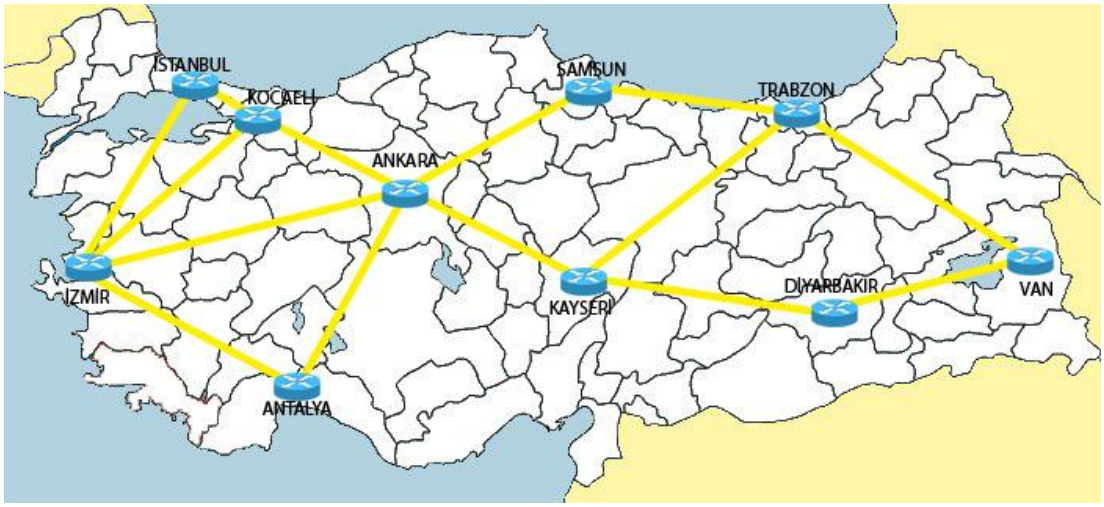
Son olarak analizin yapılabilmesi için kullanılan Jperf programı sanal makinelere kurularak, birinin sunucu olarak, diğerinin alıcı olarak ayarlamaları yapılmış ve çeşitli paketler gönderilerek analiz gerçekleştirilmiştir.

Ayrıca topoloji içinde gönderilen paketlerin içerik bilgisine ulaşmak için Wireshark programı kullanılmıştır.

4.1. Ağ Topolojileri

Analizde kullanılan ağ topolojisi Türkiye'nin şehirleri temel alınarak ayarlanmıştır. Servis sağlayıcıların yönlendiricilerinin bulundukları on büyük şehir (İstanbul, Kocaeli, Ankara, İzmir, Samsun, Kayseri, Antalya, Trabzon, Diyarbakır, Van) düşünülerek bu on şehrin her biri için bir Cisco 7200 servis sağlayıcı yönlendirici kullanılmıştır. Yönlendiriciler arasındaki bağlantı şekilleri olabilecek en verimli şekilde kurulmuştur. Topolojideki omurga yönlendirici olarak Ankara'da bulunan yönlendirici seçilmiştir. Aynı şekilde gerçek hayatta da Türkiye'nin önde gelen servis sağlayıcısı Türk Telekom A.Ş'nin de topolojisinde Ankara büyük önem taşımakta ve altyapısının kalbini oluşturmaktadır.

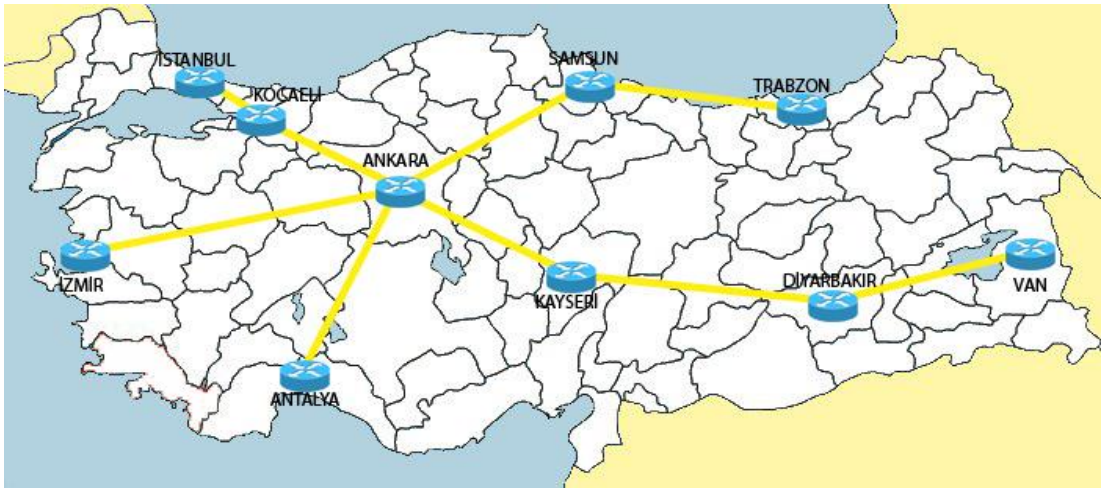
OSPF topolojisinde Ankara dışında bulunan tüm şehirlerdeki yönlendiriciler halka (ring) yapısıyla birbirlerine bağlanmışlardır. Bu durumda Ankara hariç herhangi iki şehirde bulunan yönlendiriciler arasında yaşanacak kesintide haberleşme susmayacak ve halka yapısı sayesinde trafik engellenmeyecektir. Ankara'da bulunan yönlendiricinin susması halinde ise doğu ve batı bloğu kendi içlerinde haberleşebilecek ancak iki blok aralarında haberleşemeyeceklerdir.



Şekil 4.2. OSPF Ağ Topolojisinin Türkiye Haritası Üzerinde Gösterimi

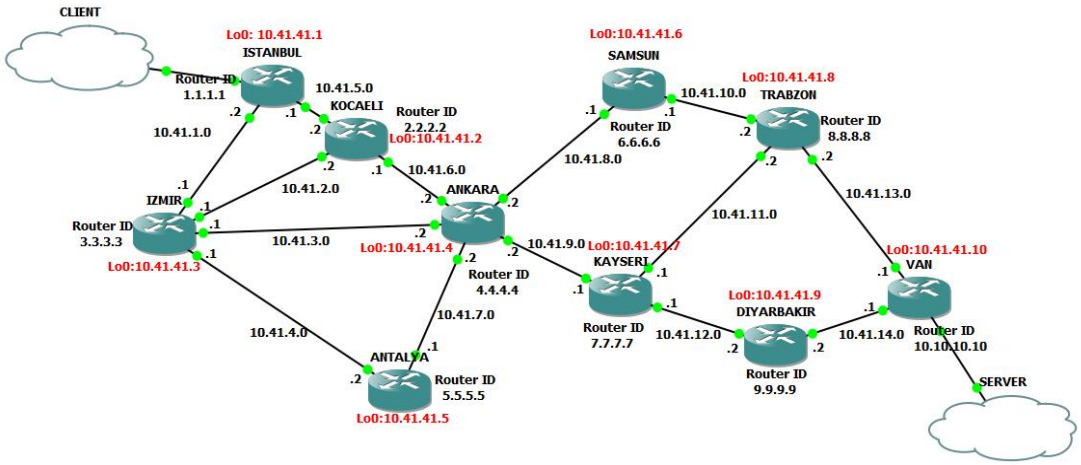
IS-IS topolojisinde ise orta sistemler arası yönlendirme gerçekleştirildiğinden ve konfigürasyonun daha rahat görülebilmesi için ring yapısı kullanılmamıştır. Bunun yerine seviye bir/iki çalışan yönlendiriciler (Kocaeli, İzmir, Kayseri ve Samsun) seviye iki omurga yönlendiricisine (Ankara) direkt bağlanmıştır. Geriye kalan seviye

bir yönlendiriciler (İstanbul, Trabzon, Diyarbakır ve Van) ise seviye bir/iki yönlendiricilere bağlanarak topoloji oluşturulmuştur.

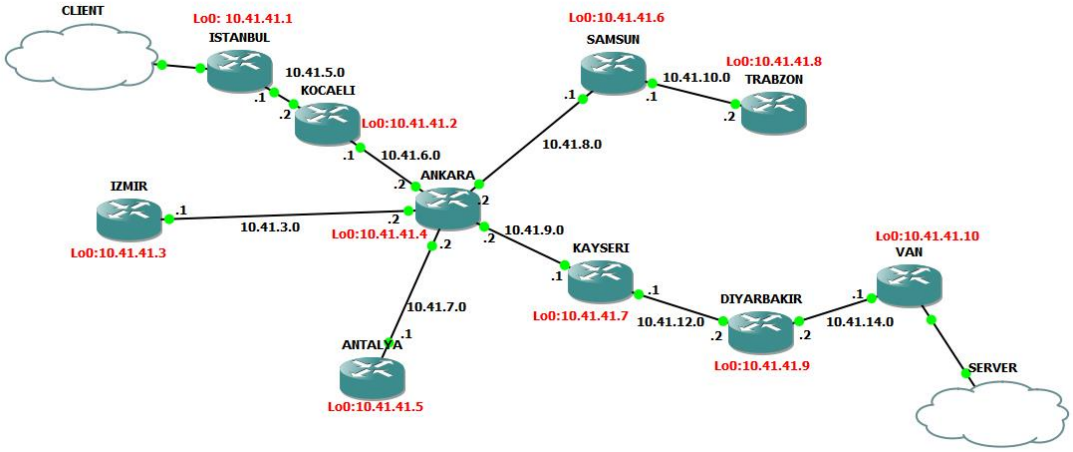


Şekil 4.3. IS-IS Ağ Topolojisinin Türkiye Haritası Üzerinde Gösterimi

Yönlendiricilere; 10.41.0.0/24'lü bloktan IP adresleri tahsis edilmiştir. Uç yönlendiriciler (İstanbul, Van) ile sanal makineler arasında kalan ağlara ise; 192.168.0.0/24'lü bloktan IP tahsis edilmiştir. Projelendirilen topolojilerin GNS3 ortamına yansıtılmış hali aşağıdaki şekildedir.



Şekil 4.4. OSPF Ağ Topolojisinin GNS3 Ortamına Yansıtılmış Hali



Şekil 4.5. IS-IS Ağ Topolojisinin GNS3 Ortamına Yansıtılmış Hali

Projede MPLS'i destekleyen iki farklı protokol kullanılmıştır. Bu protokoller OSPF ve IS-IS protokolleridir. BGP protokolünün de MPLS'i desteklemesine rağmen otonom sistemler arası çalışan bir protokol olduğu için projede sadece bilgi amaçlı bahsedilmiş olup konfigürasyonu yapılmamıştır. Yönlendiricilerin konumu değiştirilmeden OSPF ve IS-IS protokollerinin her biri için gerekli konfigürasyonlar yapılmıştır.

4.1.1. OSPF Topolojisi

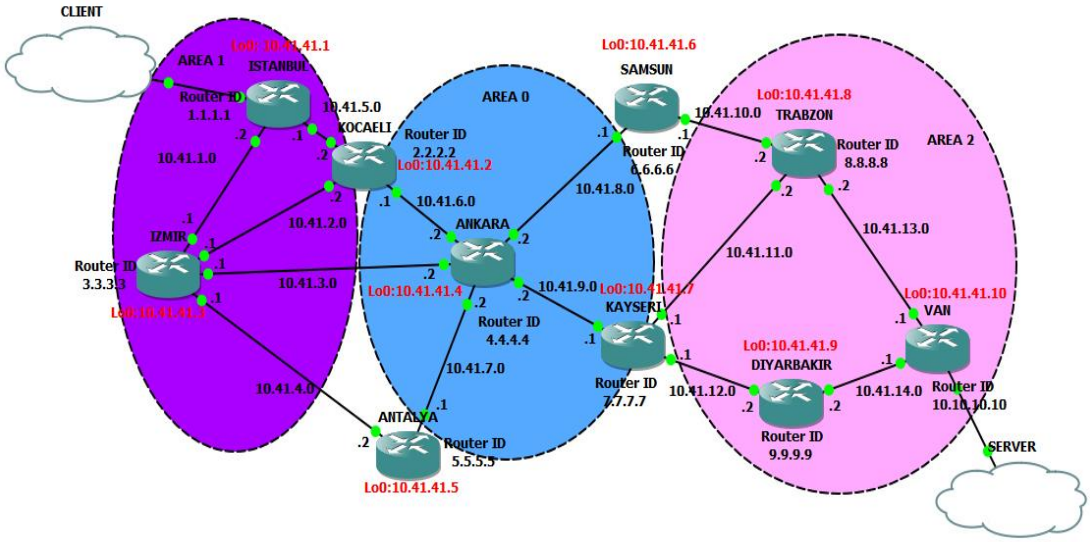
OSPF topolojisinde yönlendiriciler ikisi standart bölge, biri omurga bölgesi olmak üzere 3 bölgede (area) toplanmıştır. Bunlar;

- Omurga yönlendiricinin (Ankara) bulunduğu 0. Bölge (Omurga Bölgesi)

- Omurganın batısında kalan yönlendiriciler (İstanbul, İzmir, Kocaeli, Antalya) için 1. Bölge (Standart Bölge)

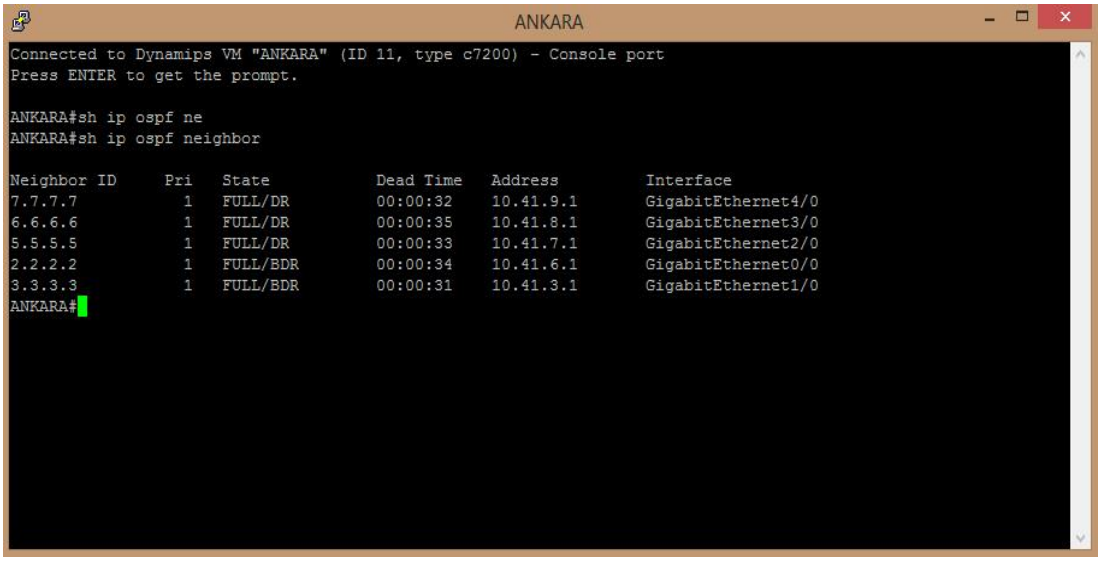
- Omurganın doğusunda kalan yönlendiriciler (Samsun, Kayseri, Trabzon, Diyarbakır, Van) için 2. Bölge (Standart Bölge)

Omurga bölgesi ile standart bölgeler arasında kalan sınır yönlendiricilerin (Kocaeli, Antalya, Samsun, Kayseri, İzmir) iç bacaklarına 0. Bölge tanımlanırken, dış bacaklarına 1. Ve 2. Bölge tanımlanmıştır.



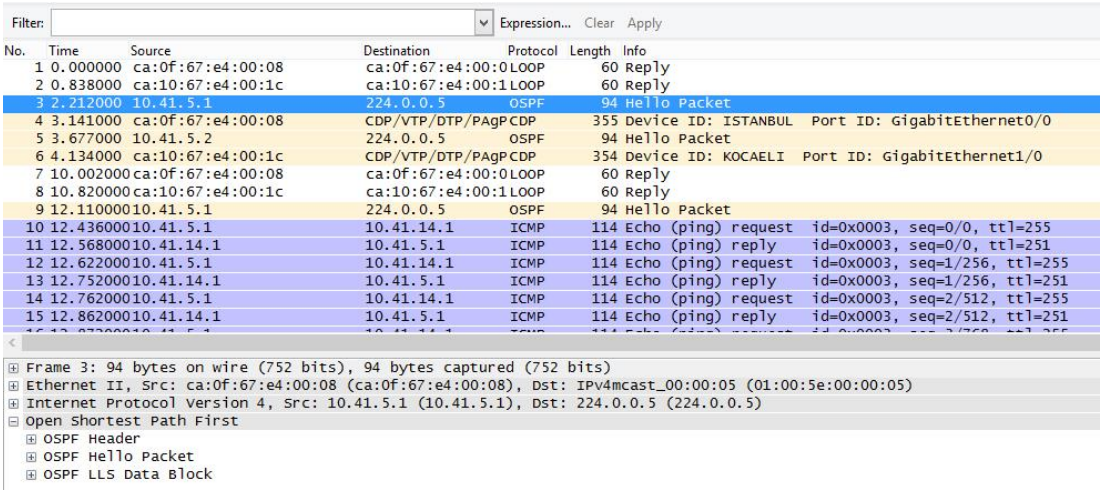
Şekil 4.6. GNS3 OSPF Topoloji

Gerekli planlamalar yapıldıktan sonra her bir yönlendirici için OSPF konfigürasyonu yapılmıştır. Tüm konfigürasyonlar tamamlandıktan sonra OSPF protokolünün doğru şekilde çalışıp çalışmadığı "Show" komutlarıyla test edilmiş. Şekil 4.7'de Ankara yönlendiricisi için OSPF komşuluk tabloları görünmektedir. Tablodan da anlaşılacağı üzere; Ankara yönlendiricisinin Kayseri (7.7.7.7), Samsun (6.6.6.6), Antalya (5.5.5.5), Kocaeli (2.2.2.2) ve İzmir (3.3.3.3) yönlendiricileriyle doğrudan komşu olduğu gözlemlenmektedir.



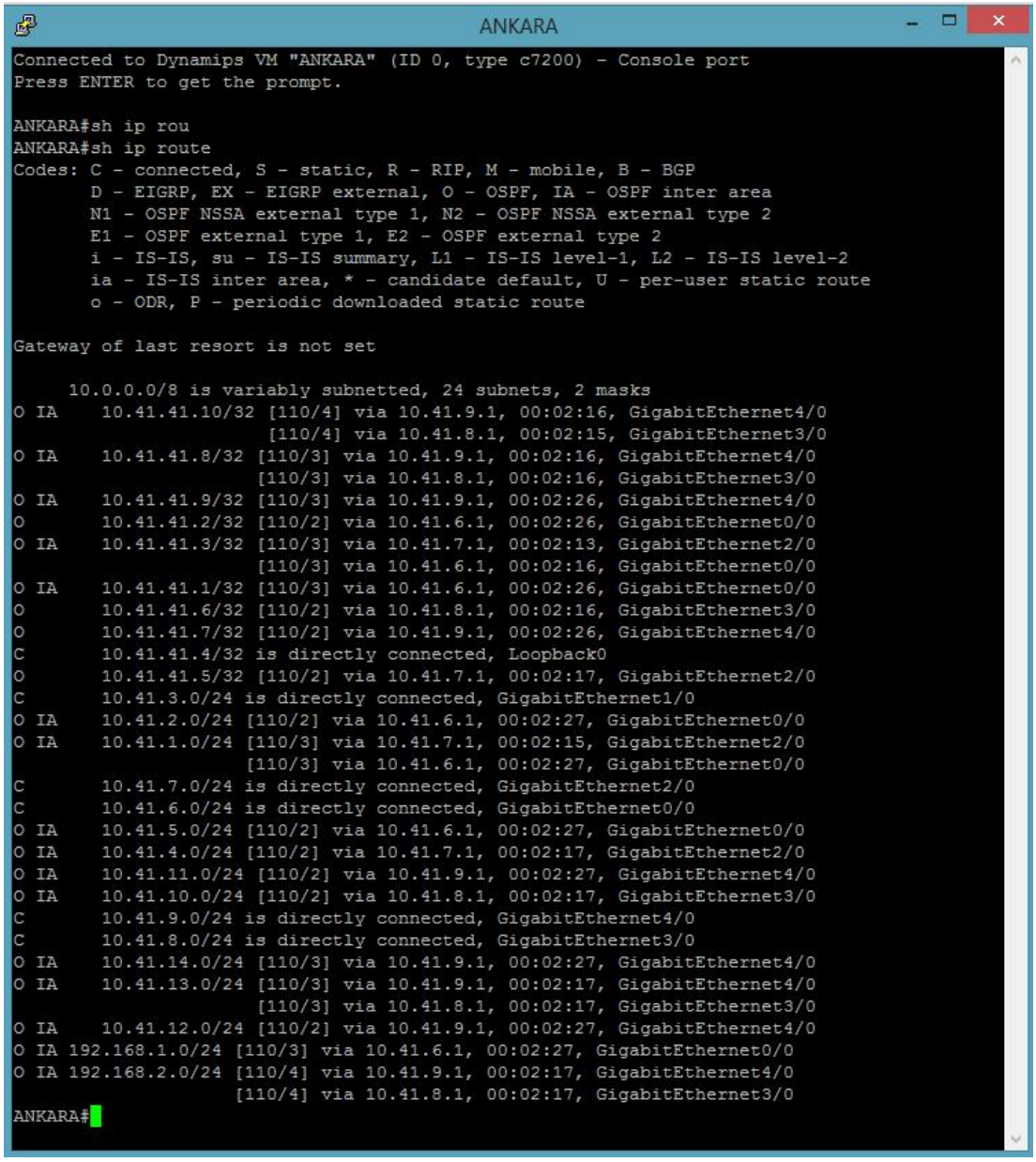
Şekil 4.7. Ankara Yönlendiricisi İçin OSPF Komşuluk Tablosu

OSPF protokolünün doğru çalışıp çalışmadığına Wireshark programı ile de bakabiliriz. Wireshark programında yakalanan paketleri incelediğimizde komşuluk kurmak için gönderilen merhaba (Hello) paketlerini görebiliriz. Şekil 4.8'de 10.41.5.1 adresinden komşuluk kurulumu için 224.0.0.5 adresine gönderilen merhaba paketlerinin içeriği görünmektedir.



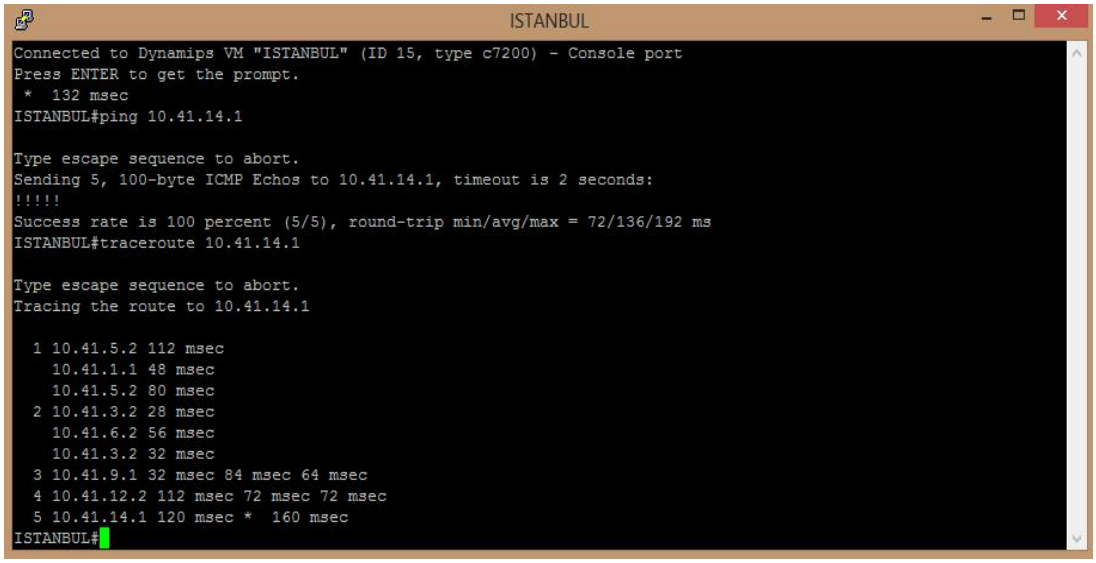
Şekil 4.8. Wireshark OSPF Konfigürasyonunda Yakalanan Paketler

Aynı şekilde OSPF protokolünün sorunsuz şekilde yapılandırıldığını yönlendirme tablolarından da anlayabilmekteyiz. Şekil 4.9'da görüldüğü üzere OSPF protokolü kullanılarak topoloji üzerinde bulunan her ağa erişim sağlanabildiği gözlemlenmektedir. Yine şekil 4.9'da Ankara yönlendiricisine direkt bağlı olmayan ağlar içinde yönlendirme tablosunda güzergahlar gösterilmiştir. Örneğin Van yönlendiricisinin döngü (loopback) adresi olan 10.41.41.10 adresine; 10.41.9.1 (Kayseri) veya 10.41.8.1 (Samsun) üzerinden gidilebildiği bilgisi görünmektedir.



Şekil 4.9. Ankara Yönlendiricisi İçin Yönlendirme Tablosu

Son olarak uç noktalarda bulunan yönlendiricilerden (İstanbul <> Van) birbirlerinin ara yüzlerine ping paketleri yollayarak yönlendirmenin doğruluğu teyit edilmiştir. Şekil 4.10'da İstanbul yönlendiricisinden Van yönlendiricisine yollanan ping paketlerinin durumu ve bu paketlerin hangi güzergahları kullanarak gidebilecekleri gözlemlenmektedir. Şekilden de anlaşılacağı üzere yollanan paketlerin tamamı başarılı şekilde hedef adrese ulaştırılmıştır.

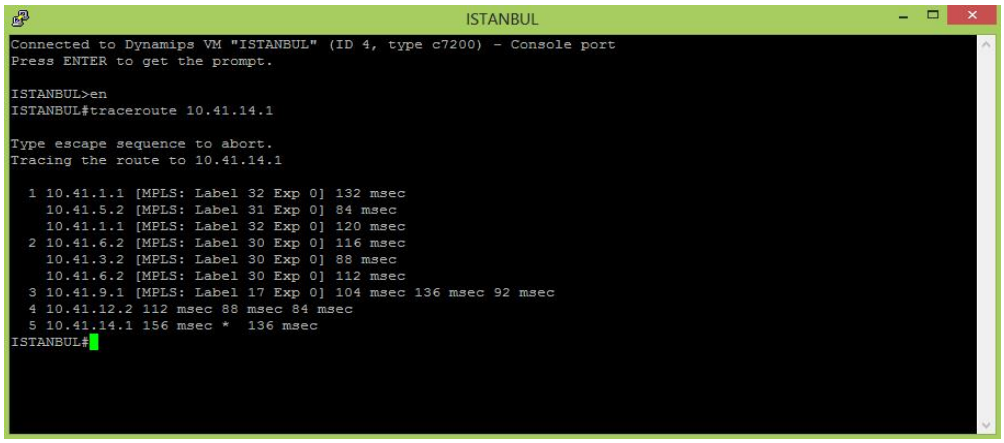


Şekil 4.10. İstanbul Yönlendiricisinden, Van yönlendiricisine Doğru Yapılan Ping ve Trace sonuçları.

Bu testler sayesinde konfigürasyonun doğru şeklide çalıştığından emin olmuş oluyoruz. Bu aşamadan sonra topolojinin MPLS'e uyumlu hale gelmesi için konfigürasyona MPLS komutları eklenmiştir.

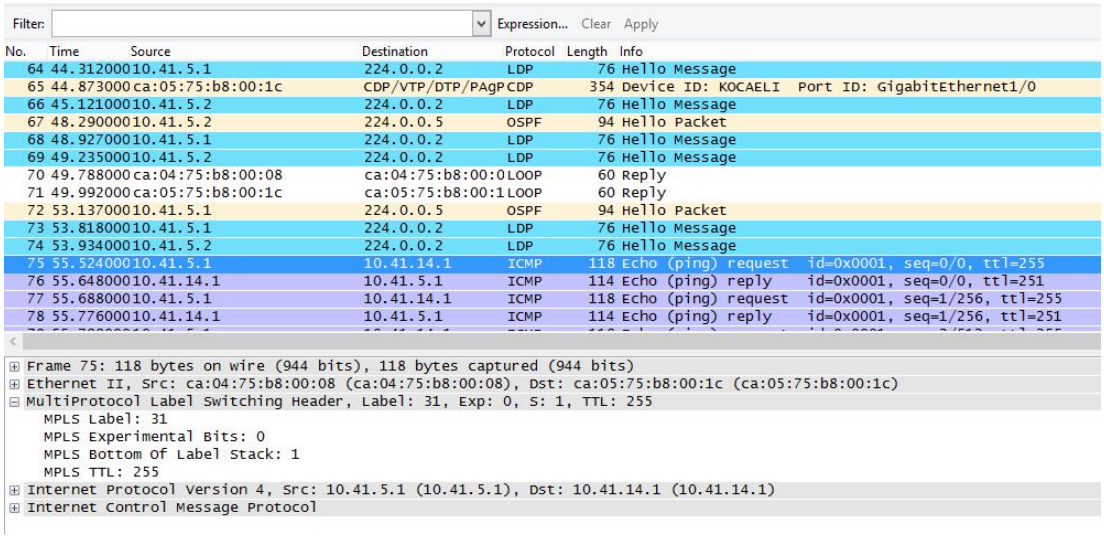
Gerekli komutlar eklendikten sonra taşınan paketlerin başına MPLS etiketlerinin yerleştirildiği gözlemlenmiştir.

Şekil 4.11'de görüldüğü üzere İstanbul yönlendiricisinin Van yönlendiricisine hangi güzergahlardan erişilebildiği bilgisine bakıldığında tabloya MPLS etiketlerinin geldiği görülmüştür.



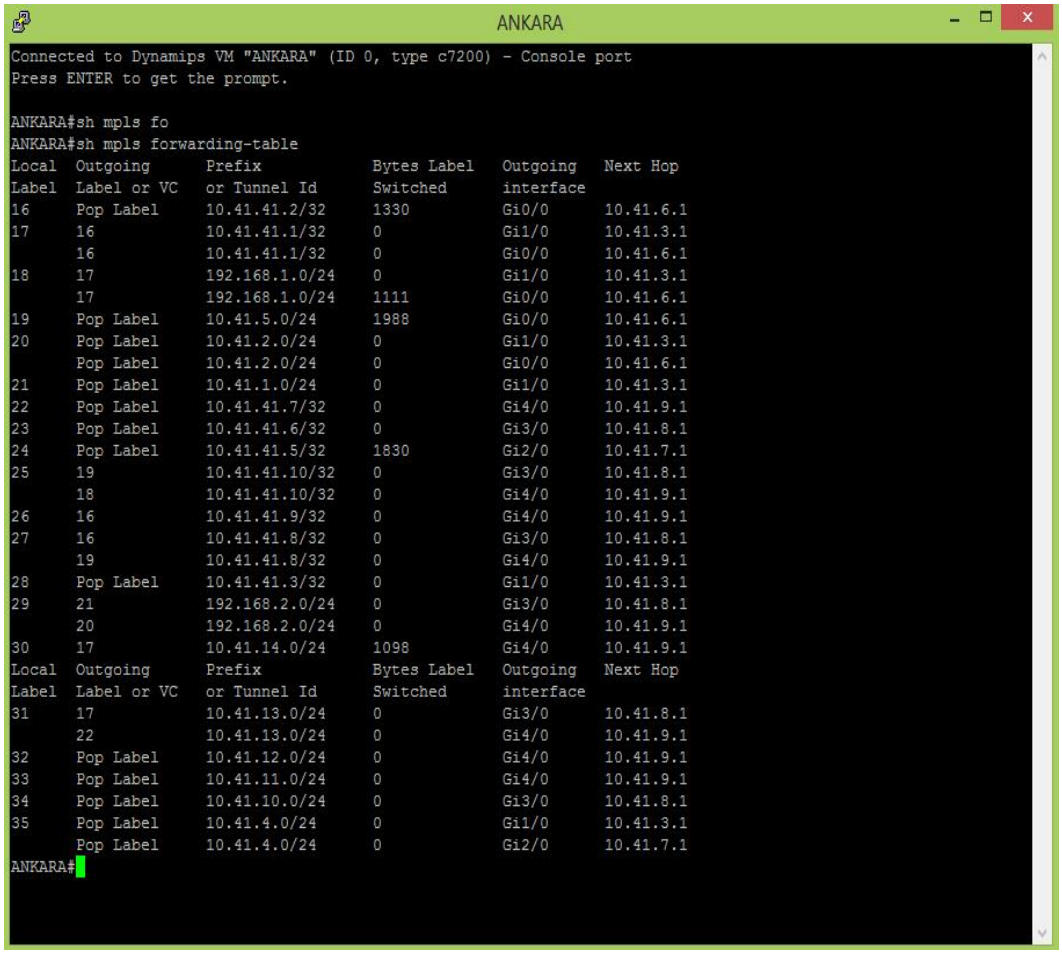
Şekil 4.11. OSPF MPLS Konfigürasyonu İle İstanbul Yönlendiricisinden Van Yönlendiricisine Doğru Yapılan trace sonuçları.

Yapılan konfigürasyondan sonra MPLS etiketlerinin paketlere eklendiklerini Wireshark programı aracılığıyla yakalanan paketlere bakarak daha net görebiliyoruz. Şekil 4.12'yi incelediğimizde İstanbul yönlendiricisinden Van yönlendiricisine gönderilen ping paketlerine MPLS etiketinin eklendiği bariz şekilde görülmektedir. Etiketin içeriği incelendiğinde 10.41.5.1 kaynağından gönderilen paketin 31 numaralı etiketle (Lable 31) iletildiği görünmektedir. Şekil 4.11'i tekrar incelediğimizde trace tablosunda 10.41.5.2 IP'sine 31 numaralı etiketle iletildiği bariz şekilde görülmektedir. Bu durum bize MPLS konfigürasyonunun doğru şeklide çalıştığını göstermektedir.



Şekil 4.12. Wireshark OSPF MPLS konfigürasyonunda yakalanan paketler.

MPLS konfigürasyonundan sonra her yönlendirme için farklı bir etiket ataması yapılmaktadır. Şekil 4.13'de Ankara yönlendiricisinin, öğrendiği ağlarla iletişime geçerken kullandığı MPLS etiketlerinin listesi görülmektedir.



Şekil 4.13. Ankara Yönlendiricisi İçin MPLS Yönlendirme Tablosu

Bu şekilde OSPF topolojisinin MPLS'li ve MPLS'siz olarak sorunsuz çalıştığı test edilmiştir. Bu aşamadan sonra sanal makinelerde kurulu olan Jperf programıyla analiz yapılmıştır.

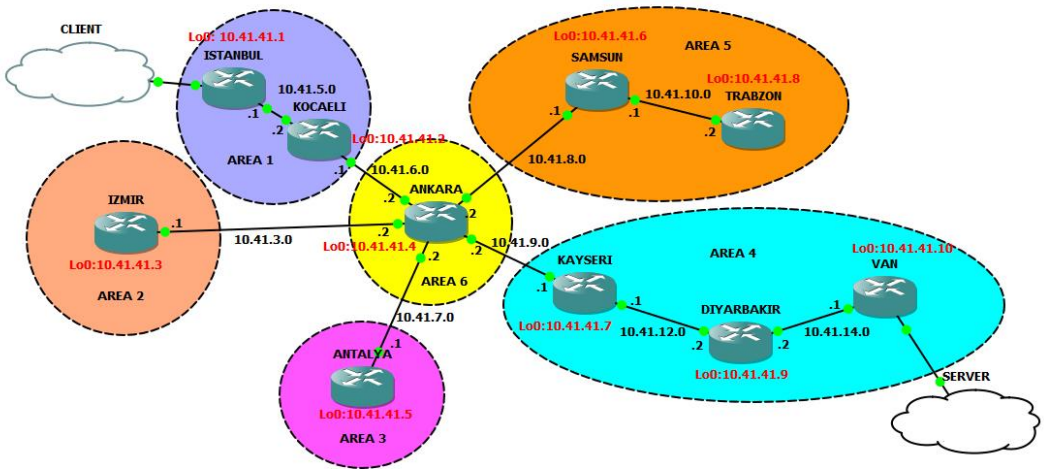
4.1.2. IS-IS Topoloji

IS-IS topolojisinde yönlendiriciler altı bölgede toplanmışlardır. Bunlar;

- Omurga yönlendiricinin (Ankara) bulunduğu ve ikinci seviye yönlendirmenin yapıldığı altıncı bölge

- Omurga yönlendiricinin (Ankara) ara yüzlerine bağlı olan yönlendiricilerin (Kocaeli, İzmir, Antalya, Samsun, Kayseri) bulunduğu ve birinci/ikinci seviye yönlendirmenin birlikte yapıldığı 5 bölge

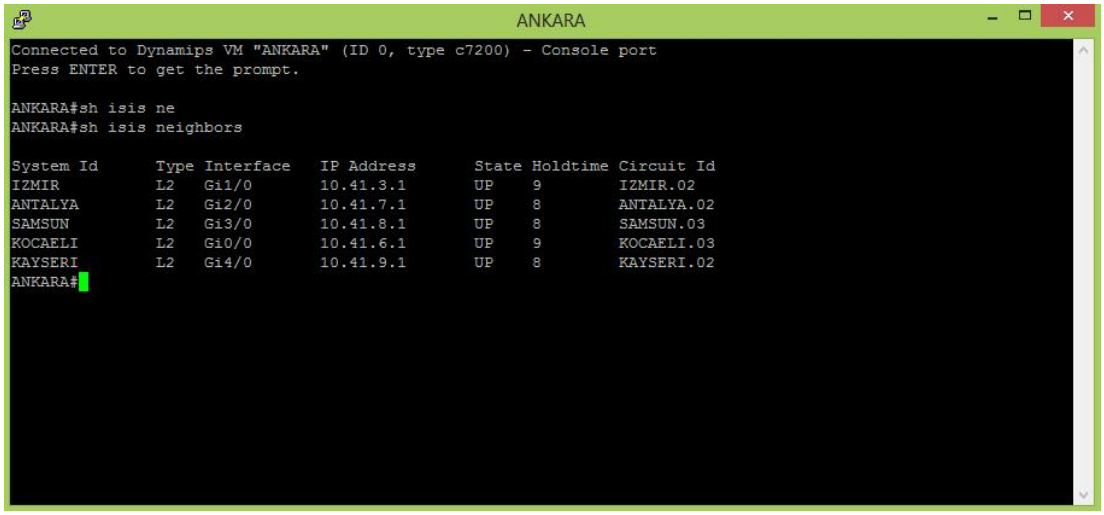
IS-IS topolojisinde Ankara yönlendiricisi ikinci seviye yönlendirici olarak tanımlanmıştır. Ankara yönlendiricisine direkt bağlı olan Kocaeli, İzmir, Antalya, Samsun ve Kayseri yönlendiricileri birinci ve ikinci seviye yönlendirici, geriye kalan İstanbul, Trabzon, Diyarbakır, Van yönlendiricileri de seviye bir yönlendirici olarak tanımlanmışlardır. Seviye bir ve iki yönlendirme yapan beş yönlendirici, omurga yönlendirici ile birinci seviye yönlendirme yapan dört yönlendirici arasındaki iletişimi sağlamak amacıyla konfigüre edilmiştir.



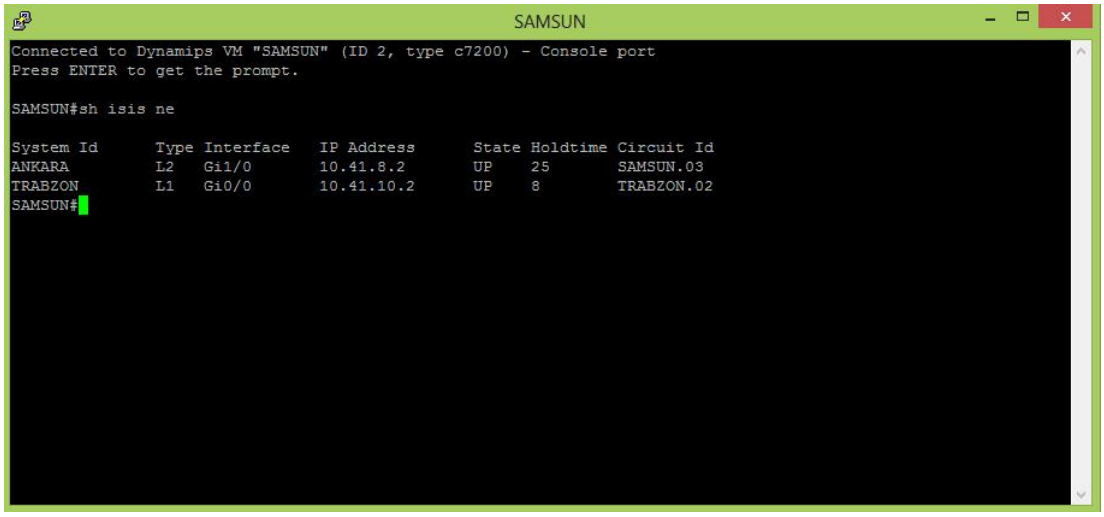
Şekil 4.14. GNS3 IS-IS Topoloji

Gerekli planlamalar yapıldıktan sonra her bir yönlendirici için IS-IS konfigürasyonu yapılmıştır. Tüm konfigürasyonlar tamamlandıktan sonra IS-IS protokolünün doğru şekilde çalışıp çalışmadığı "Show" komutlarıyla test edilmiş. Şekil 4.15'de Ankara yönlendiricisi için IS-IS komşuluk tablosu görünmektedir. Tabloyu incelediğimizde Ankara yönlendiricisinin komşuluk kurduğu; İzmir, Antalya, Samsun, Kocaeli ve Kayseri yönlendiricilerine seviye iki (L2) yönlendirme ile komşuluk kurduğu görünmektedir. Aynı şekilde Samsun yönlendiricisi için IS-IS komşuluk tablolarına şekil 4.16'da baktığımızda, Samsun yönlendiricisinin bağlı olduğu Ankara yönlendiricisiyle seviye 2 yönlendirme yaparak, Trabzon yönlendiricisiyle ise seviye bir (L1) yönlendirme yaparak komşuluk kurduğu gözlemlenmektedir. Seviye bir yönlendirmenin Samsun yönlendiricisinin komşuluk tablosunda görünüp, Ankara yönlendiricisinin komşuluk tablosunda görünmemesinin sebebi, Samsun yönlendiricisinin seviye bir/iki yönlendirmeye izin verip, Ankara'nın ise omurga

bölgesinde olup sadece seviye 2 yönlendirmeye izin vermesidir. Bu şekilde bütün yönlendiriciler birbirlerini görürler.

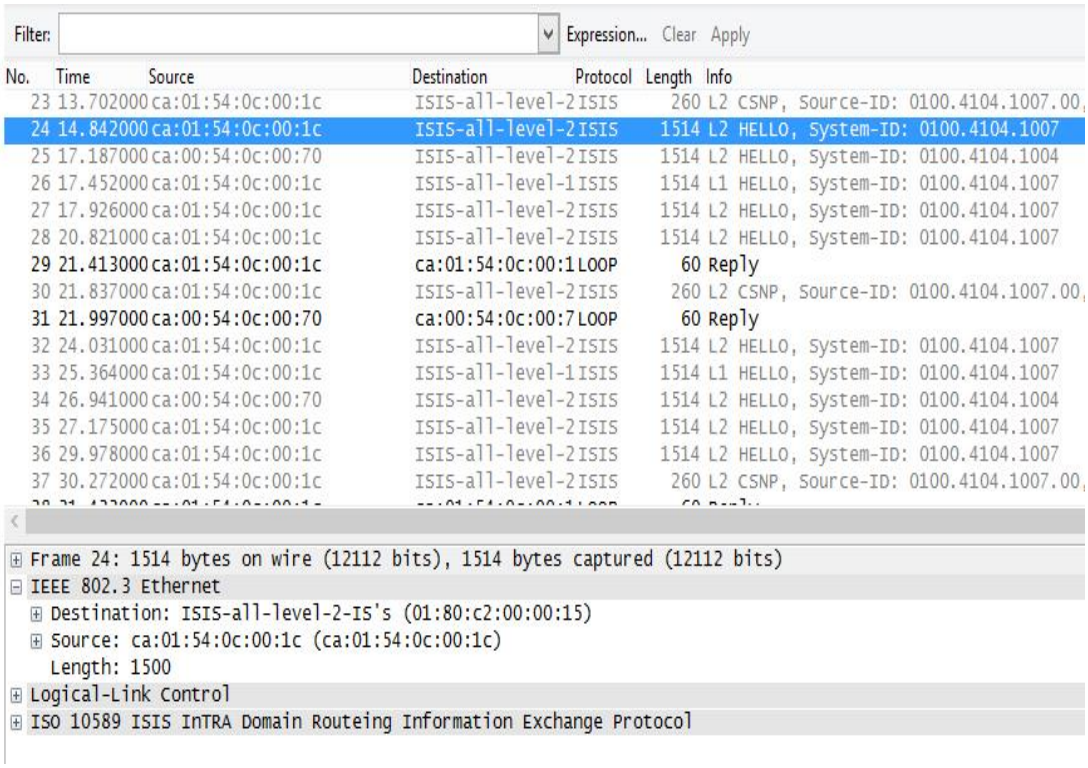


Şekil 4.15. Ankara Yönlendiricisi İçin IS-IS Komşuluk Tablosu



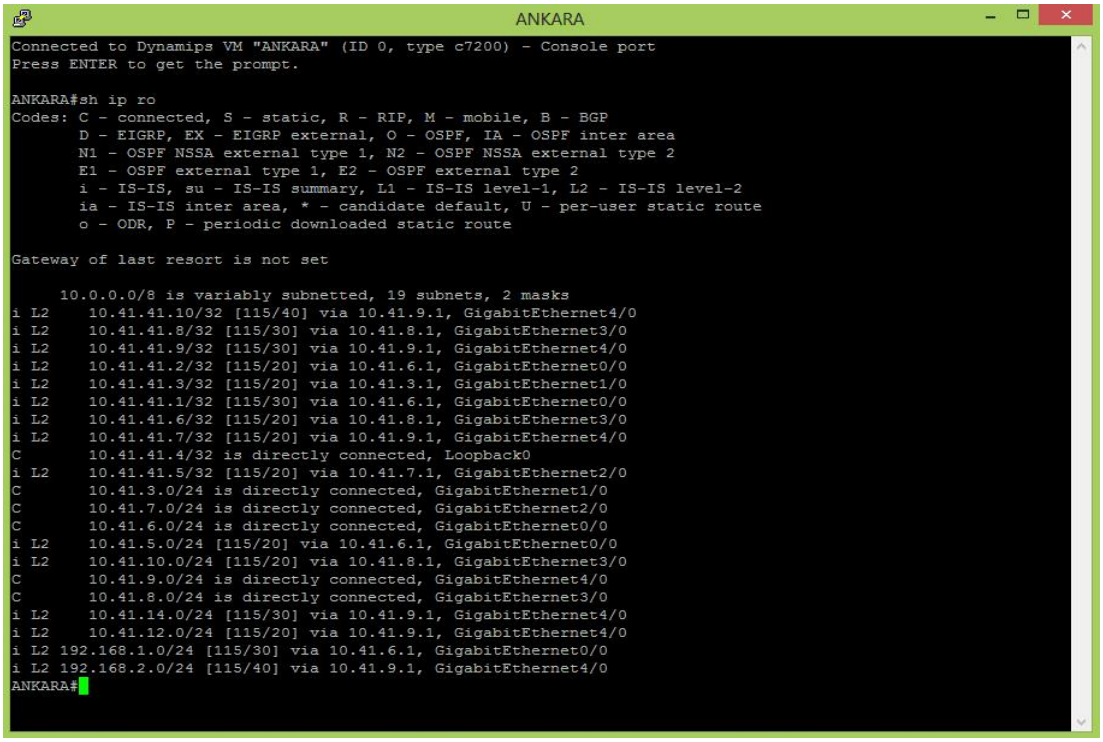
Şekil 4.16. Samsun Yönlendiricisi İçin IS-IS Komşuluk Tablosu

IS-IS protokolünün doğru çalışıp çalışmadığına Wireshark programı ile de bakabiliriz. Wireshark programında yakalanan paketleri incelediğimizde komşuluk kurmak için gönderilen merhaba (Hello) paketlerini görebiliriz. Şekil 4.17'de Ankara yönlendiricisinden tüm seviye 2 yönlendirme yapan yönlendiricilere gönderilen merhaba paketleri görünmektedir. Örneğin Şekilde seçili olan paket 0100.4104.1007 ID'li Kayseri yönlendiricisine aittir.



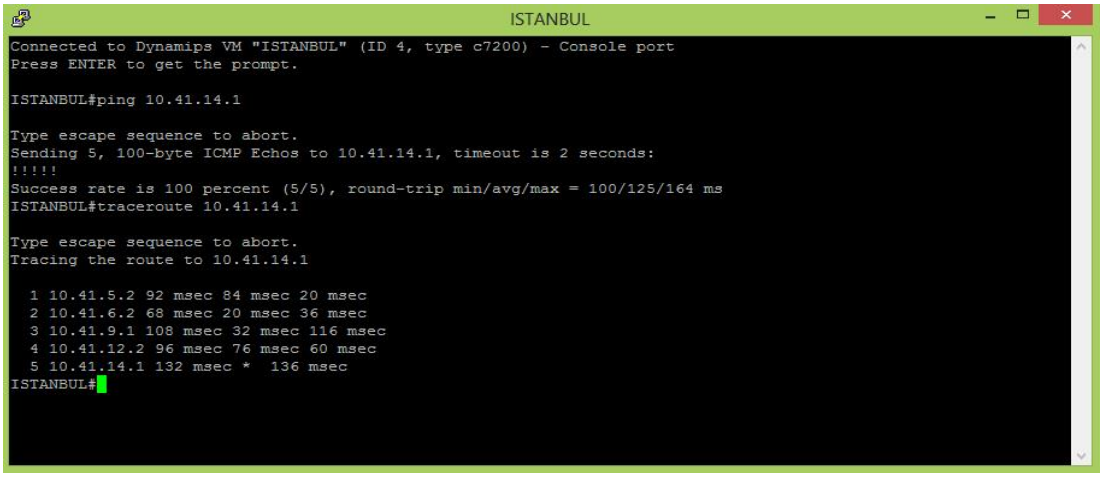
Şekil 4.17. Wireshark IS-IS Konfigürasyonunda Yakalanan Paketler

Aynı şekilde IS-IS protokolünün sorunsuz şekilde yapılandırıldığını yönlendirme tablolarından da anlayabilmekteyiz. Şekil 4.18'de görüldüğü üzere IS-IS protokolü kullanılarak topoloji üzerinde bulunan her ağa erişim sağlanabildiği gözlemlenmektedir. Yine şekil 4.16'da Ankara yönlendiricisine direkt bağlı olmayan ağlar içinde yönlendirme tablosunda güzergahlar gösterilmiştir. Örneğin Van yönlendiricisinin döngü (loopback) adresi olan 10.41.41.10 adresine; 10.41.9.1 (Kayseri) üzerinden gidilebildiği bilgisi görünmektedir.



Şekil 4.18. Ankara Yönlendiricisi İçin Yönlendirme Tablosu

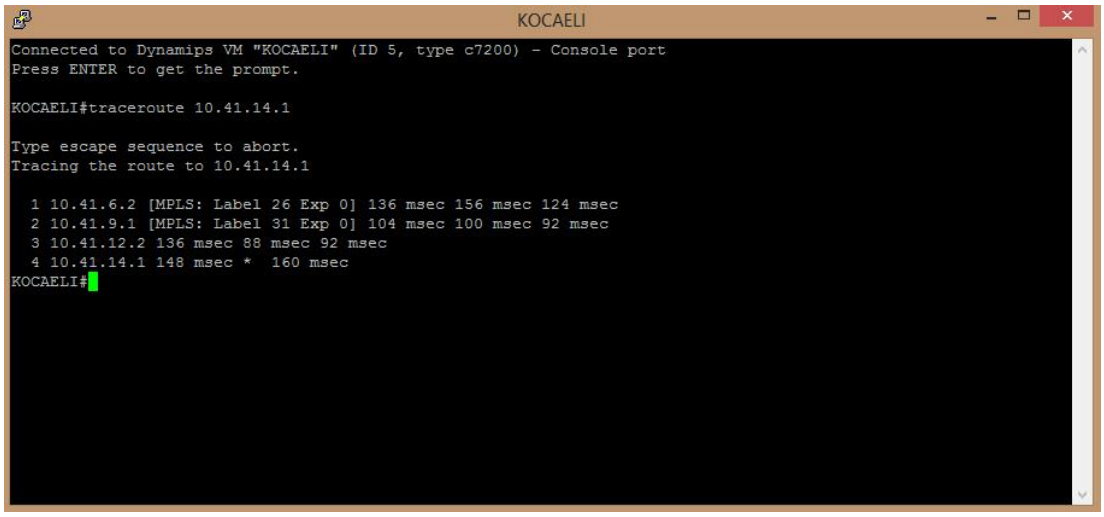
Son olarak uç noktalarda bulunan yönlendiricilerden (İstanbul <> Van) birbirlerinin ara yüzlerine ping paketleri yollayarak yönlendirmenin doğruluğu teyit edilmiştir. Şekil 4.19'da İstanbul yönlendiricisinden Van yönlendiricisine yollanan ping paketlerinin durumu ve bu paketlerin hangi güzergahları kullanarak gidebilecekleri gözlemlenmektedir. Şekilden de anlaşılacağı üzere yollanan paketlerin tamamı başarılı şekilde hedef adrese ulaştırılmıştır.



Şekil 4.19. İstanbul Yönlendiricisinden, Van yönlendiricisine Doğru Yapılan Ping ve Trace sonuçları.

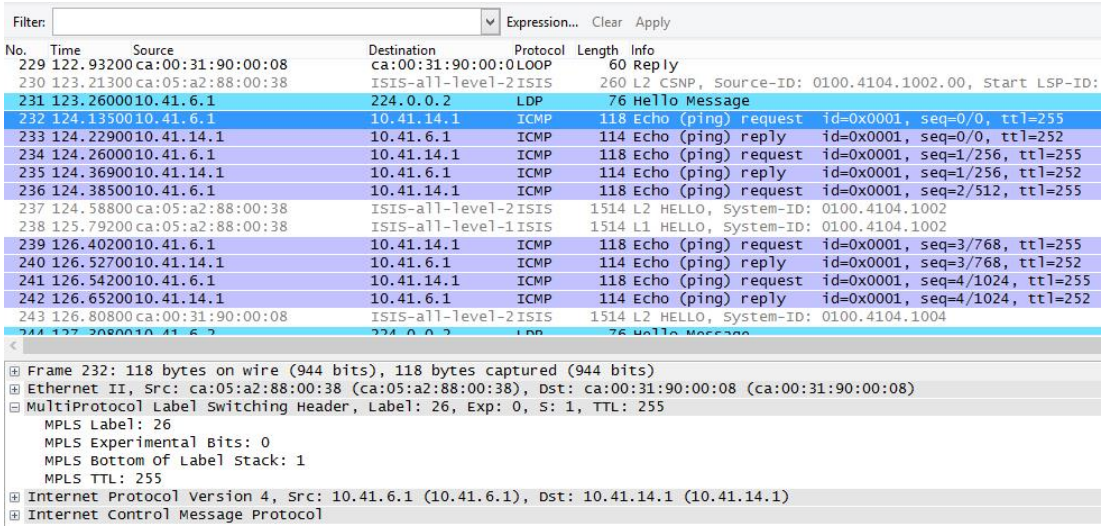
Bu testler sayesinde konfigürasyonun doğru şeklide çalıştığından emin olmuş oluyoruz. Bu aşamadan sonra topolojinin MPLS‘e uyumlu hale gelmesi için OSPF topolojisinde olduğu gibi konfigürasyona MPLS komutları eklenmiştir.

Gerekli komutlar eklendikten sonra taşınan paketlerin başına MPLS etiketlerinin yerleştirildiği gözlemlenmiştir. Şekil 4.20’de bu değişiklik gözlemlenmektedir.



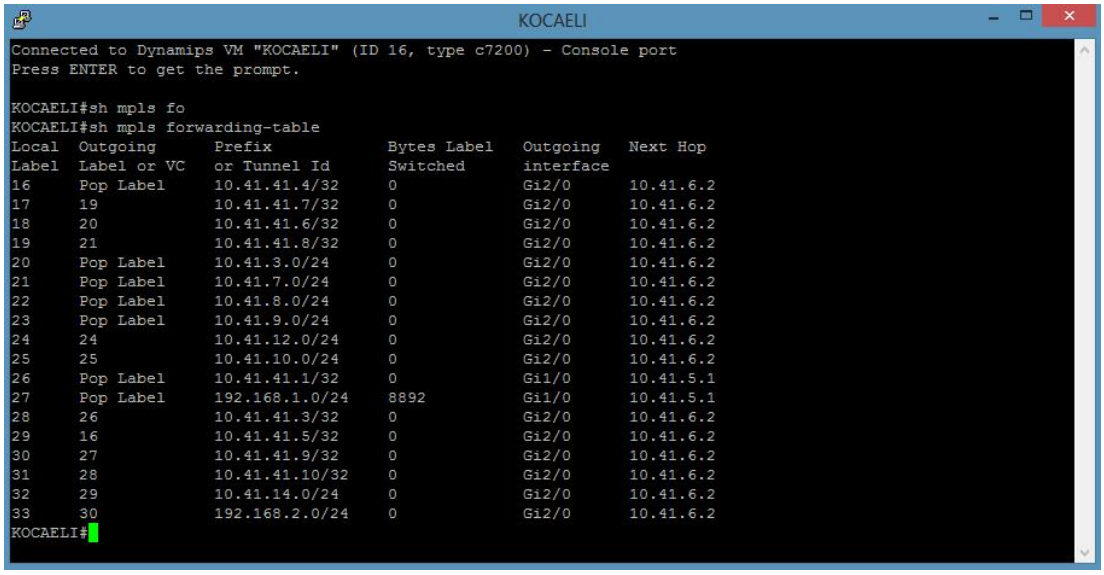
Şekil 4.20. IS-IS MPLS Konfigürasyonu İle Kocaeli Yönlendiricisinden Van Yönlendiricisine Doğru Yapılan trace sonuçları.

Yapılan konfigürasyondan sonra MPLS etiketlerinin paketlere eklendiklerini Wireshark programı aracılığıyla yakalanan paketlere bakarak daha net görebiliyoruz. Şekil 4.21‘i incelediğimizde Kocaeli yönlendiricisinden Van yönlendiricisine gönderilen ping paketlerine MPLS etiketinin eklendiği bariz şekilde görülmektedir. Etiketin içeriği incelendiğinde 10.41.6.1 kaynağından gönderilen paketin 26 numaralı etiketle (Lable 26) iletildiği görünmektedir. Şekil 4.20‘yi tekrar incelediğimizde trace tablosunda 10.41.6.2 IP‘sine 26 numaralı etiketle iletildiği bariz şekilde görülmektedir. Bu durum bize MPLS konfigürasyonunun doğru şeklide çalıştığını göstermektedir.



Şekil 4.21. Wireshark IS-IS MPLS Konfigürasyonunda Yakalanan Paketler.

IS-IS MPLS konfigürasyonundan sonra her yönlendirme için farklı bir etiket ataması yapılmaktadır. Şekil 4.22'de Kocaeli yönlendiricisinin, öğrendiği ağlarla iletişime geçerken kullandığı MPLS etiketlerinin listesi görülmektedir.



Şekil 4.22. Kocaeli Yönlendiricisi İçin MPLS Yönlendirme Tablosu

Bu şekilde IS-IS topolojisinin MPLS'li ve MPLS'siz olarak sorunsuz çalıştığı test edilmiştir. Bu aşamadan sonra sanal makinelerde kurulu olan Jperf programıyla analiz yapılmıştır.

4.2. Analiz Sonuçları

OSPF ve IS-IS protokolleri için ilgili topolojiler hazırlanıp, MPLS'li ve MPLS'siz yönlendirici konfigürasyonları yapıldıktan sonra en uçta bulunan yönlendiricilere sanal makineler bağlanarak Jperf trafik ölçüm programı yardımıyla;

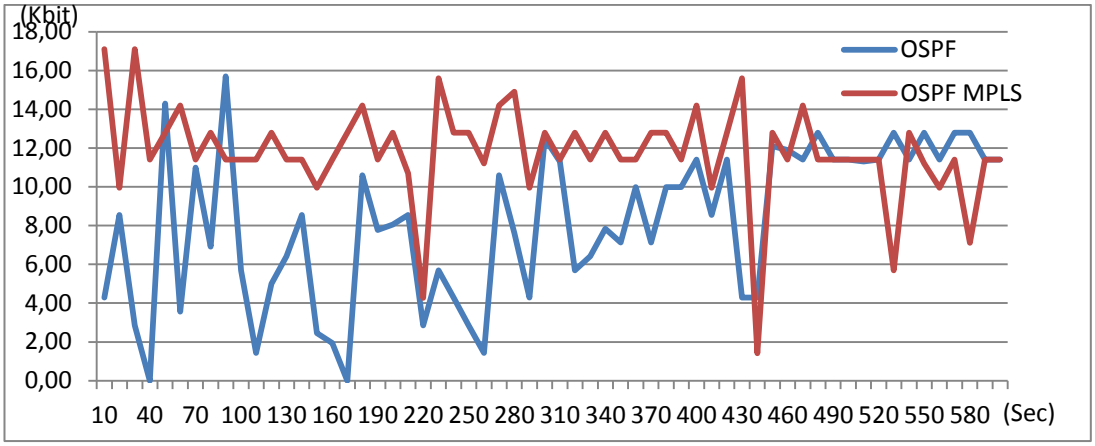
- Bant Genişliği (Bandwidth)
- Seğirme (Jitter)
- Aktarım (Throughput)

TCP ve UDP paketleri için ölçülerek kıyaslamalar yapılmıştır. Bu kıyaslamalardan elde edilen sonuçlar aşağıdaki gibidir.

4.2.1. Bant Genişliği (Bandwidth)

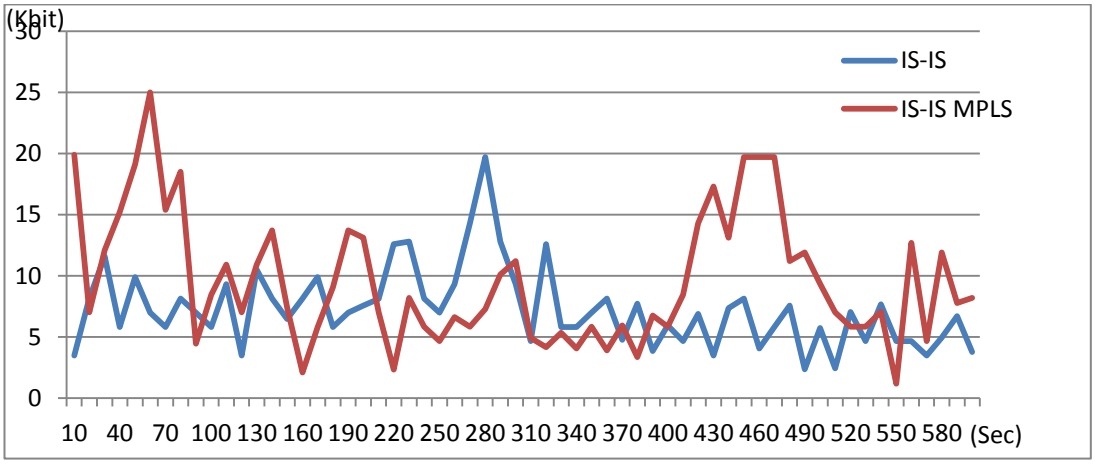
Bant genişliği analizi için on dakika boyunca Jperf alıcıdan sunucuya TCP ve UDP paketleri gönderilmiştir. Gönderilen TCP paketleri için pencere boyutu (TCP Window Size) 8.00 Kbyte olarak ayarlanmıştır. UDP ara bellek uzunluğu (UDP Buffer Size) 8.00 Kbyte olarak ayarlanmıştır. Yapılan analiz için 5001 numaralı port kullanılmıştır. Analiz; OSPF, OSPF MPLS, IS-IS, IS-IS MPLS konfigürasyonları için sırayla yapılıp, aynı paket kullanılarak (TCP veya UDP) yapılan MPLS'li ve MPLS'siz test sonuçları karşılaştırılmıştır. Buna göre;

Yollanan TCP paketleri için OSPF konfigürasyonu kullanılarak elde edilen grafik şekil 4.23'deki gibidir. Yapılan bu analizden OSPF konfigürasyonu ile yollanan TCP paketlerinde ortalama bant genişliği değeri 6,72 Kbit/sec olarak, OSPF MPLS konfigürasyonu ile yollanan TCP paketlerinde ise bant genişliği ortalama değeri 9,65 Kbit/sec olarak bulunmuştur.



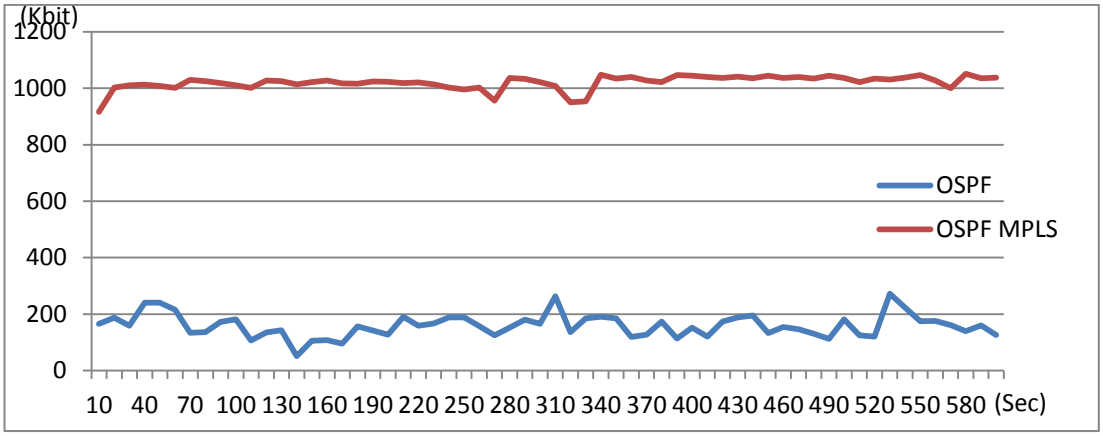
Şekil 4.23. OSPF/OSPF MPLS TCP Bandwidth Grafiği

Yollanan TCP paketleri için IS-IS konfigürasyonu kullanılarak elde edilen grafik şekil 4.24'deki gibidir. Yapılan bu analizden IS-IS konfigürasyonu ile yollanan TCP paketlerinde ortalama bant genişliği değeri 6,25 Kbit/sec olarak, IS-IS MPLS konfigürasyonu ile yollanan TCP paketlerinde ise bant genişliği ortalama değeri 9,57 Kbit/sec olarak bulunmuştur.



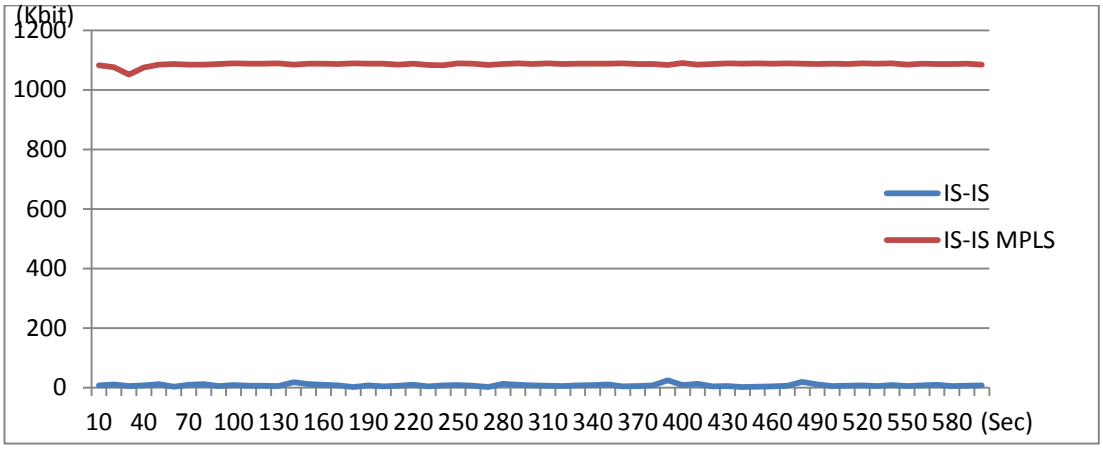
Şekil 4.24. IS-IS/IS-IS MPLS TCP Bandwidth Grafiği

Yollanan UDP paketleri için OSPF konfigürasyonu kullanılarak elde edilen grafik şekil 4.25'deki gibidir. Yapılan bu analizden OSPF konfigürasyonu ile yollanan UDP paketlerinde ortalama bant genişliği değeri 158,88 Kbit/sec olarak, OSPF MPLS konfigürasyonu ile yollanan UDP paketlerinde ise bant genişliği ortalama değeri 1020,35 Kbit/sec olarak bulunmuştur.



Şekil 4.25. OSPF/OSPF MPLS UDP Bandwidth Grafiği

Yollanan UDP paketleri için IS-IS konfigürasyonu kullanılarak elde edilen grafik şekil 4.26'daki gibidir. Yapılan bu analizden IS-IS konfigürasyonu ile yollanan UDP paketlerinde ortalama bant genişliği değeri 7,34 Kbit/sec olarak, IS-IS MPLS konfigürasyonu ile yollanan UDP paketlerinde ise bant genişliği ortalama değeri 1086,17 Kbit/sec olarak bulunmuştur.

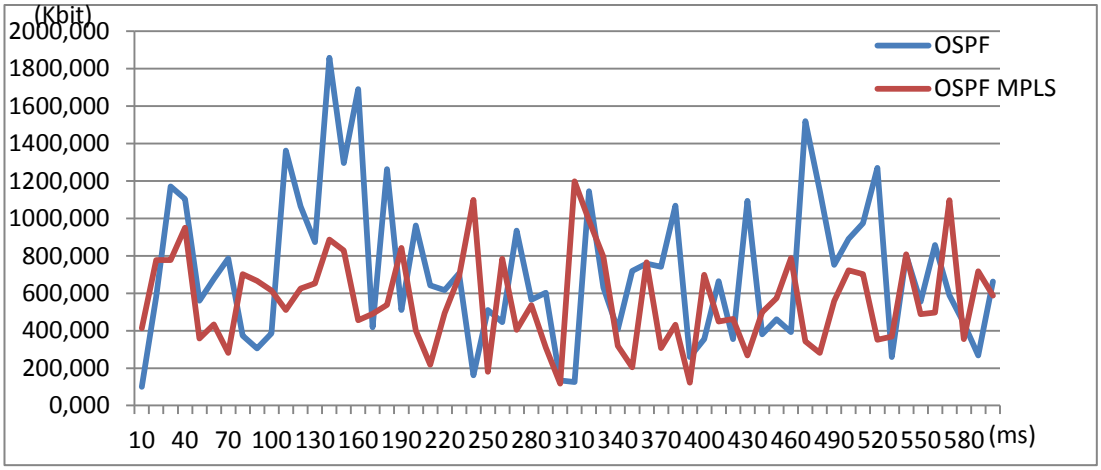


Şekil 4.26. IS-IS/IS-IS MPLS UDP Bandwidth Grafiği

4.2.2. Seğirme (Jitter)

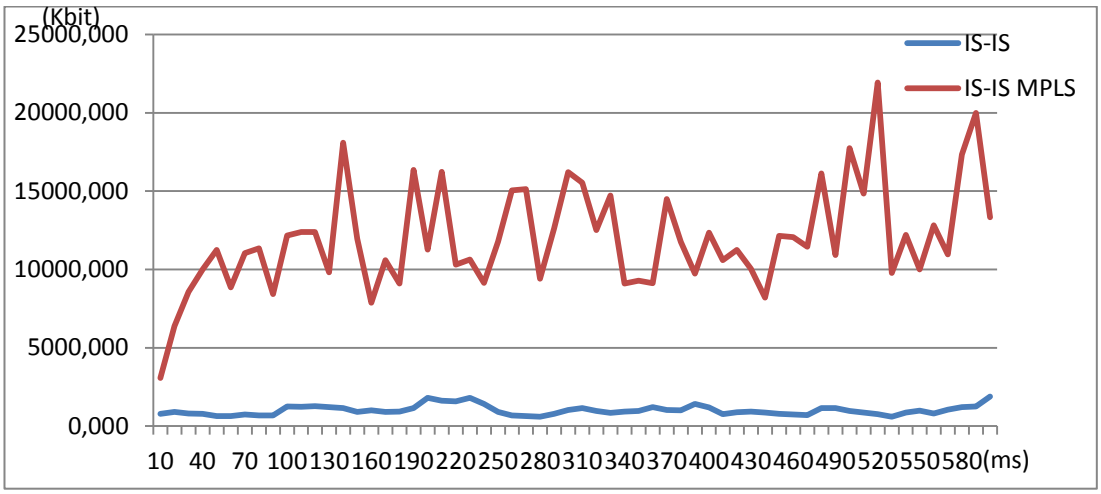
Seğirtme analizi için on dakika boyunca Jperf alıcıdan sunucuya UDP paketleri gönderilmiştir. Gönderilen UDP paketleri için UDP ara bellek uzunluğu (UDP Buffer Size) 8.00 Kbyte olarak ayarlanmıştır. Yapılan analiz için 5001 numaralı port kullanılmıştır. Analiz; OSPF, OSPF MPLS, IS-IS, IS-IS MPLS konfigürasyonları için sırayla yapılıp, MPLS'li ve MPLS'siz test sonuçları karşılaştırılmıştır. Buna göre;

Yollanan UDP paketleri için OSPF konfigürasyonu kullanılarak elde edilen grafik şekil 4.27'deki gibidir. Yapılan bu analizden OSPF konfigürasyonu ile yollanan UDP paketlerinde ortalama jitter değeri 720,28 ms olarak, OSPF MPLS konfigürasyonu ile yollanan UDP paketlerinde ise jitter ortalama değeri 563,13 ms olarak bulunmuştur.



Şekil 4.27. OSPF/OSPF MPLS UDP Jitter Grafiği

Yollanan UDP paketleri için IS-IS konfigürasyonu kullanılarak elde edilen grafik şekil 4.28'deki gibidir. Yapılan bu analizden IS-IS konfigürasyonu ile yollanan UDP paketlerinde ortalama jitter değeri 1011,12 ms olarak, IS-IS MPLS konfigürasyonu ile yollanan UDP paketlerinde ise jitter ortalama değeri 11996,57 ms olarak bulunmuştur.

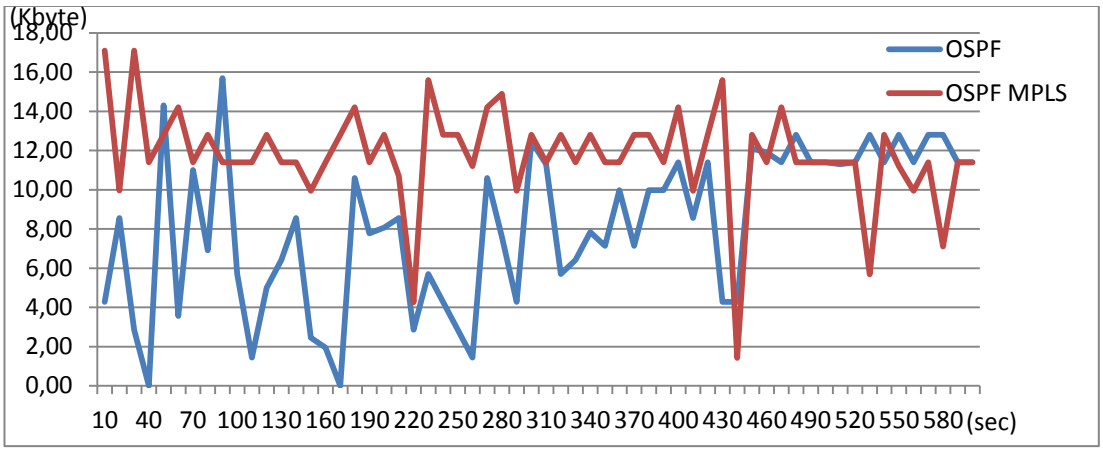


Şekil 4.28. IS-IS/IS-IS MPLS UDP Jitter Grafiği

4.2.3. Aktarım (Throughput)

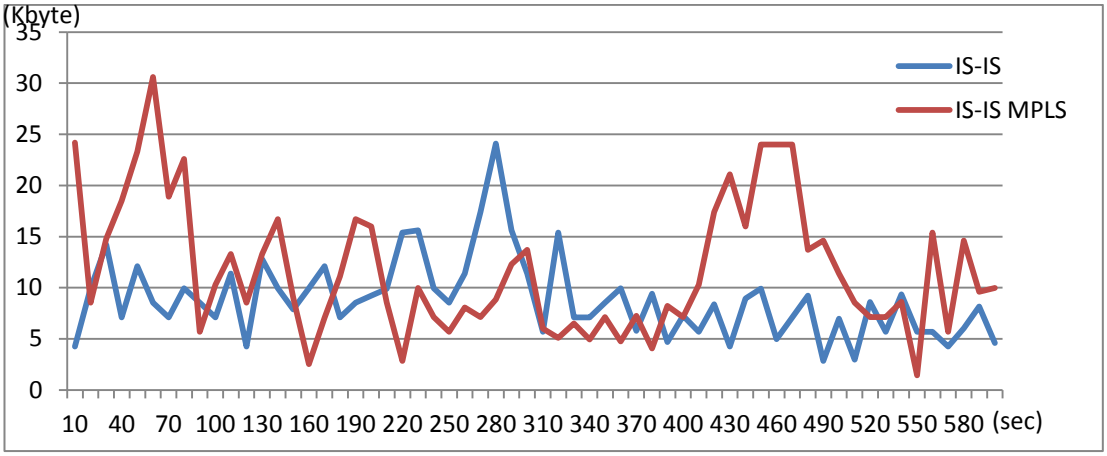
Aktarım analizi için on dakika boyunca Jperf alıcıdan sunucuya TCP ve UDP paketleri gönderilmiştir. Gönderilen TCP paketleri için pencere boyutu (TCP Window Size) 8.00 Kbyte olarak ayarlanmıştır. UDP ara bellek uzunluğu (UDP Buffer Size) 8.00 Kbyte olarak ayarlanmıştır. Yapılan analiz için 5001 numaralı port kullanılmıştır. Analiz; OSPF, OSPF MPLS, IS-IS, IS-IS MPLS konfigürasyonları için sırayla yapılıp aynı paket kullanılarak (TCP veya UDP) yapılan MPLS'li ve MPLS'siz test sonuçları karşılaştırılmıştır. Buna göre;

Yollanan TCP paketleri için OSPF konfigürasyonu kullanılarak elde edilen grafik şekil 4.29'daki gibidir. Yapılan bu analizden OSPF konfigürasyonu ile yollanan TCP paketlerinde ortalama veri transfer değeri 8,19 Kbyte, toplam veri transfer değeri ise 342,28 Kbyte olarak bulunmuştur. OSPF MPLS konfigürasyonu ile yollanan TCP paketlerinde ise ortalama veri transfer değeri 11,78 Kbyte, toplam veri transfer değeri ise 401,57 Kbyte olarak bulunmuştur.



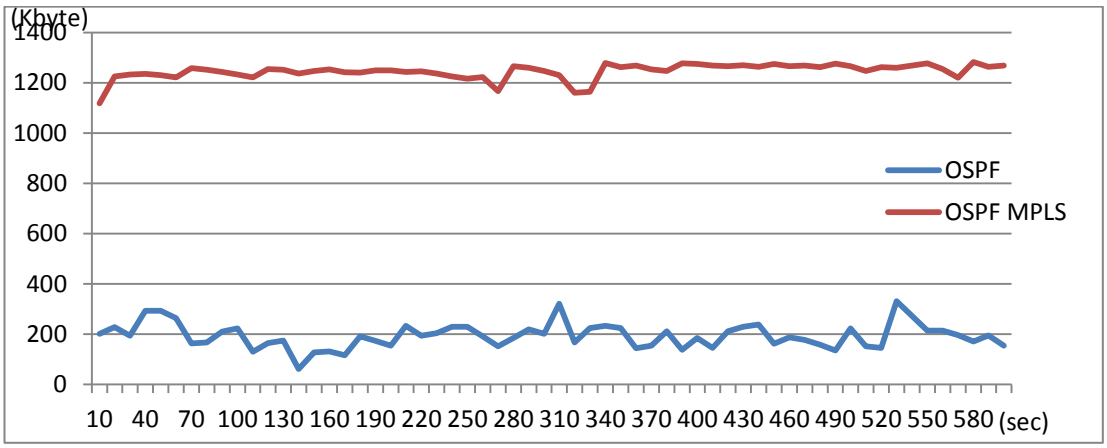
Şekil 4.29. OSPF/OSPF MPLS TCP Throughput Grafiği

Yollanan TCP paketleri için IS-IS konfigürasyonu kullanılarak elde edilen grafik şekil 4.30'daki gibidir. Yapılan bu analizden IS-IS konfigürasyonu ile yollanan TCP paketlerinde ortalama veri transfer değeri 8,86 Kbyte, toplam veri transfer değeri ise 531,81 Kbyte olarak bulunmuştur. IS-IS MPLS konfigürasyonu ile yollanan TCP paketlerinde ise ortalama veri transfer değeri 11,69 Kbyte, toplam veri transfer değeri ise 701,85 Kbyte olarak bulunmuştur.



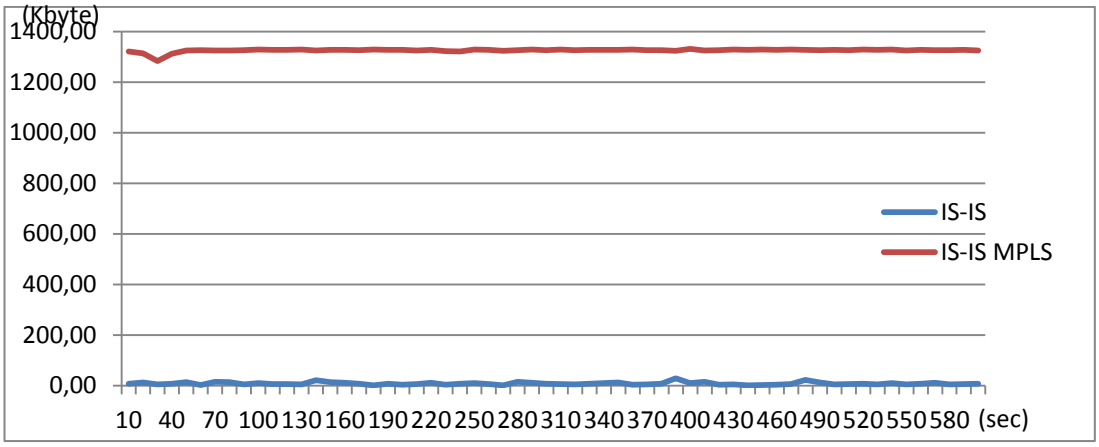
Şekil 4.30. IS-IS/IS-IS MPLS TCP Throughput Grafiği

Yollanan UDP paketleri için OSPF konfigürasyonu kullanılarak elde edilen grafik şekil 4.31'deki gibidir. Yapılan bu analizden OSPF konfigürasyonu ile yollanan UDP paketlerinde ortalama veri transfer değeri 193,95 Kbyte, toplam veri transfer değeri ise 11637 Kbyte olarak bulunmuştur. OSPF MPLS konfigürasyonu ile yollanan UDP paketlerinde ise ortalama veri transfer değeri 1245,48 Kbyte, toplam veri transfer değeri ise 74729 Kbyte olarak bulunmuştur.



Şekil 4.31. OSPF/OSPF MPLS UDP Throughput Grafiği

Yollanan UDP paketleri için IS-IS konfigürasyonu kullanılarak elde edilen grafik şekil 4.32'deki gibidir. Yapılan bu analizden IS-IS konfigürasyonu ile yollanan UDP paketlerinde ortalama veri transfer değeri 9,01 Kbyte, toplam veri transfer değeri ise 540,31 Kbyte olarak bulunmuştur. IS-IS MPLS konfigürasyonu ile yollanan UDP paketlerinde ise ortalama veri transfer değeri 1325,70 Kbyte, toplam veri transfer değeri ise 79542 Kbyte olarak bulunmuştur.



Şekil 4.32. IS-IS/IS-IS MPLS UDP Throughput Grafiği

5. SONUÇLAR VE ÖNERİLER

Bireysel ve kurumsal kullanıcıların her geçen gün değişen kullanım alışkanlıkları daha hızlı iletişim ihtiyacı doğurmaktadır. Hız ile beraber daha fazla veri akışı ve bant genişliği ihtiyacı ortaya çıkmıştır. Teknolojinin hızla gelişmesi, altyapının ve network omurgalarının geliştirilmesine zemin hazırlamıştır. Bu nedenle MPLS teknolojisi, servis sağlayıcıları tarafından, günümüzde tercih edilen yegane yönlendirme teknolojisidir. Yüksek hızda anahtarlama ve yönlendirmeye elverişli olması tercih sebeplerinin başında gelmektedir.

Yapılan bu tez çalışmada MPLS'in etiketleme özelliğinin trafik üzerindeki etkisi sanal yönlendirici ve makineler üzerinde test edilmiştir. MPLS'li ve MPLS'siz şekilde OSPF ve IS-IS protokolleri konfigürasyonu yapılarak, alıcı ve sunucuya TCP ve UDP paketleri yollanmış ve bu paketlerin bant genişliği, seğirme ve aktarım sonuçlarına göre analizi yapılmıştır.

Alınan sonuçlar değerlendirildiğinde aşağıdaki sonuçlara ulaşılmıştır.

Bant Genişliği analizinde;

- MPLS OSPF ve MPLS IS-IS konfigürasyonları üzerinden gönderilen TCP paketlerinin bant genişliği, MPLS'siz konfigürasyonlar üzerinden gönderilenlerin bant genişliklerine göre daha yüksek olduğu gözlemlenmiştir.

- Aynı şekilde MPLS OSPF ve MPLS IS-IS konfigürasyonları üzerinden gönderilen UDP paketlerinin bant genişliği, MPLS'siz konfigürasyonlar üzerinden gönderilenlerin bant genişliklerine göre daha yüksektir.

Bu durum bize MPLS kullanıldığında bant genişliğinin daha verimli kullanılabildiğini kanıtlamıştır.

Jitter analizinde;

- MPLS OSPF ve MPLS IS-IS konfigürasyonları üzerinden gönderilen UDP paketlerinin jitter değerlerinin, MPLS‘siz konfigürasyonlar üzerinden gönderilenlerin jitter değerlerine göre daha düşük olduğu gözlemlenmiştir.

Bu durum bize MPLS kullanıldığında gönderilen paketler arasında süre farkının daha düşük olduğunu kanıtlamaktadır.

Throughput analizinde;

- MPLS OSPF ve MPLS IS-IS konfigürasyonları üzerinden gönderilen TCP paketlerinin veri transferi ortalamasının ve toplamının, MPLS‘siz konfigürasyonlar üzerinden gönderilen veri transfer ortalaması ve toplamına göre daha yüksek olduğu gözlemlenmiştir.

- Aynı şekilde MPLS OSPF ve MPLS IS-IS konfigürasyonları üzerinden gönderilen UDP paketlerinin veri transferi ortalamasının ve toplamının, MPLS‘siz konfigürasyonlar üzerinden gönderilen veri transfer ortalaması ve toplamına göre daha yüksek olduğu gözlemlenmiştir.

Bu durum bize MPLS‘in daha fazla veri transferine olanak tanıdığını kanıtlamaktadır.

Yukarıdaki bilgiler ışığında; MPLS kullanılan konfigürasyonların, kullanılmayan konfigürasyonlara göre daha verimli çalıştığı gözlemlenmektedir. MPLS‘in daha fazla bant genişliğine ve daha fazla veri aktarımına olanak sağladığı görülmüştür. MPLS konfigürasyonu kullanılırken gönderilen paketler arasındaki iletim süresi farklılıklarının azaldığı görülmektedir. Tüm bu verilerin ışığında MPLS‘in yönlendirme protokollerini daha verimli hale getirdiği söylenebilir.

Bu çalışmayı geliştirmek adına farklı uygulama katmanı, trafik türleri ile çalışılarak, MPLS‘li ve MPLS‘siz konfigürasyonlarda davranış şekilleri kıyaslanabilir. Sanal yönlendiriciler yerine gerçek yönlendiriciler kullanılarak çalışmanın tutarlılığı arttırılabilir.

KAYNAKLAR

[1] Kaplan Y., MPLS, http://www.yasinkaplan.com/tr/docs/MPLS.pdf (Ziyaret Tarihi: 09 Mayıs 2013).

[2] Göktaş S., Çok protokollü etiket anahtarlama (MPLS) teknolojisi ve performans analizi, Yüksek Lisans Tezi, Marmara Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul, 2010, 275892.

[3] Başulaş T., MPLS Omurga Tasarımı, Yüksek Lisans Tezi, Beykent Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul, 2006, 201956.

[4] Söğüt İ., Çok protokollü etiket anahtarlama (MPLS) kullanılarak çok alanlı sanal özel ağ (VPN) kurulumu, Yüksek Lisans Tezi, Fen Bilimleri Enstitüsü, Muğla, 2009, 245010.

[5] http://www.linktionary.com/l/label_switching.html (Ziyaret Tarihi: 09 Mayıs 2013).

[6] http://www.cisco.com/en/US/docs/ios/12_4t/12_4t2/ftldp41.html (Ziyaret Tarihi: 09 Mayıs 2013).

[7] Yücel Ç., Çoklu protokollü etiket anahtarlama (MPLS) ve MPLS özel sanal ağlar (MPL - VPN), Türkiye'de İnternet (INET-TR), İstanbul, Türkiye, 19 -21 Aralık 2002.

[8] Moy J. T., OSPF Anatomy Of An Internet Routing Protocol, Addison-Wesley Publishing Company, Massachusetts, 1998.

[9] http://www.dummies.com/how-to/content/working-with-open-shortest-path-first-ospf-routing.html (Ziyaret Tarihi: 10 Mayıs 2013).

[10] http://www.fatih.edu.tr/~bguler/bilg121/Ynlendrme-protokoller.pdf (Ziyaret Tarihi: 11 Mayıs 2013).

[11] http://pic.dhe.ibm.com/infocenter/iseries/v7r1m0/index.jsp?topic=%2Frzal6%2Frzal6ospfpackettypes.htm (Ziyaret Tarihi: 11 Mayıs 2013).

[12] http://tr.scribd.com/doc/84382794/135/OSPF-Area (Ziyaret Tarihi: 11 Mayıs 2013).

[13] http://en.wikipedia.org/wiki/IS-IS (Ziyaret Tarihi: 12 Mayıs 2013).

[14] http://www.cisco.com/en/US/products/ps6599/products_white_paper09186a00800a3e6f.shtml (Ziyaret Tarihi: 12 Mayıs 2013).

[15] http://www.interlab.ait.ac.th/tein2/Presentations/Basic%20Routing/IS-IS%20-%20Routing%20Protocol.pdf (Ziyaret Tarihi: 12 Mayıs 2013).

[16] http://www.bidb.itu.edu.tr/?d=1115 (Ziyaret Tarihi: 13 Mayıs 2013).

[17] Tatar G., BGP Protokolü Temelleri Bölüm 2, http://www.cozumpark.com/blogs/network/archive/2013/03/31/bgp-protokol-temelleri-b-l-m-2.aspx (Ziyaret Tarihi: 18 Mayıs 2013).

[18] http://www.gns3.net/ (Ziyaret Tarihi:20 mayıs 2013).

[19] http://www.wireshark.org/ (Ziyaret Tarihi: 20 Mayıs 2013).

[20] Akpınar H., Wireshark Kullanıcı Rehberi, http://www.enderunix.org/docs/wireshark.pdf (Ziyaret Tarihi: 20 Mayıs 2013).

[21] http://utkuulusahin.wordpress.com/tag/turkce-wireshark-kullanim-kilavuzu/ (Ziyaret Tarihi: 20 mayıs 2013).

[22] https://www.virtualbox.org/ (Ziyaret Tarihi: 20 Mayıs 2013).

[23] http://openmaniak.com/iperf.php (Ziyaret Tarihi: 21 Mayıs 2013).

EKLER

Ek−A YÖNLENDİRİCİ KONFİGÜRASYONLARI

İstanbul:

```
!
!
upgrade fpd auto
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname ISTANBUL
!
boot-start-marker
boot-end-marker
!
logging message-counter syslog
!
no aaa new-model
ip source-route
ip cef
!
!
no ip domain lookup
no ipv6 cef
multilink bundle-name authenticated
!
!
voice dsp waitstate 0
!
!
!
memory-size iomem 0
archive
 log config
  hidekeys
!
!
!
```

```
interface Loopback0
 ip address 10.41.41.1 255.255.255.255
!
interface Ethernet0/0
 no ip address
 shutdown
 duplex auto
!
interface GigabitEthernet0/0
 ip address 10.41.5.1 255.255.255.0
 duplex full
 speed 1000
 media-type gbic
 negotiation auto
 mpls ip
!
interface GigabitEthernet1/0
 ip address 10.41.1.2 255.255.255.0
 negotiation auto
 mpls ip
!
interface FastEthernet2/0
 ip address 192.168.1.1 255.255.255.0
 duplex auto
 speed auto
 mpls ip
!
interface FastEthernet2/1
 no ip address
 shutdown
 duplex auto
 speed auto
!
router ospf 100
 router-id 1.1.1.1
 log-adjacency-changes
 network 10.41.1.0 0.0.0.255 area 1
 network 10.41.5.0 0.0.0.255 area 1
 network 10.41.41.1 0.0.0.0 area 1
 network 192.168.1.0 0.0.0.255 area 1
!
ip forward-protocol nd
no ip http server
no ip http secure-server
!
!
logging alarm informational
!
!
```

```
control-plane
!
!
mgcp fax t38 ecm
!
!
!
!
gatekeeper
 shutdown
!
!
line con 0
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
!
end
```

Ankara:

```
!

!
upgrade fpd auto
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname ANKARA
!
boot-start-marker
boot-end-marker
!
logging message-counter syslog
!
no aaa new-model
ip source-route
ip cef
!
!
no ip domain lookup
ip domain name lab.local
no ipv6 cef
```

```
multilink bundle-name authenticated
!
!
voice dsp waitstate 0
!
!
memory-size iomem 0
archive
 log config
  hidekeys
!
!
interface Loopback0
 ip address 10.41.41.4 255.255.255.255
!
interface Ethernet0/0
 no ip address
 shutdown
 duplex auto
!
interface GigabitEthernet0/0
 ip address 10.41.6.2 255.255.255.0
 duplex full
 speed 1000
 media-type gbic
 negotiation auto
 mpls ip
!
interface GigabitEthernet1/0
 ip address 10.41.3.2 255.255.255.0
 negotiation auto
 mpls ip
!
interface GigabitEthernet2/0
 ip address 10.41.7.2 255.255.255.0
 negotiation auto
 mpls ip
!
interface GigabitEthernet3/0
 ip address 10.41.8.2 255.255.255.0
 negotiation auto
 mpls ip
!
interface GigabitEthernet4/0
 ip address 10.41.9.2 255.255.255.0
 negotiation auto
 mpls ip
!
router ospf 100
```

```
 router-id 4.4.4.4
 log-adjacency-changes
 network 10.41.3.0 0.0.0.255 area 0
 network 10.41.6.0 0.0.0.255 area 0
 network 10.41.7.0 0.0.0.255 area 0
 network 10.41.8.0 0.0.0.255 area 0
 network 10.41.9.0 0.0.0.255 area 0
 network 10.41.41.4 0.0.0.0 area 0
!
ip forward-protocol nd
no ip http server
no ip http secure-server
!
!
logging alarm informational
!
!
control-plane
!
!
mgcp fax t38 ecm
!
!
gatekeeper
 shutdown
!
!
line con 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
 stopbits 1
line aux 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
 stopbits 1
line vty 0 4
 login
!
End
```

Antalya:

```
!

!
upgrade fpd auto
version 12.4
```

```
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname ANTALYA
!
boot-start-marker
boot-end-marker
!
logging message-counter syslog
!
no aaa new-model
ip source-route
ip cef
!
!
no ip domain lookup
ip domain name lab.local
no ipv6 cef
multilink bundle-name authenticated
!
!
voice dsp waitstate 0
!
!
!
!
!
memory-size iomem 0
archive
 log config
  hidekeys
!
!
interface Loopback0
 ip address 10.41.41.5 255.255.255.255
!
interface Ethernet0/0
 no ip address
 shutdown
 duplex auto
!
interface GigabitEthernet0/0
 ip address 10.41.7.1 255.255.255.0
 duplex full
 speed 1000
 media-type gbic
 negotiation auto
 mpls ip
```

```
!
interface GigabitEthernet1/0
 ip address 10.41.4.2 255.255.255.0
 negotiation auto
 mpls ip
!
router ospf 100
 router-id 5.5.5.5
 log-adjacency-changes
 network 10.41.4.0 0.0.0.255 area 1
 network 10.41.7.0 0.0.0.255 area 0
 network 10.41.41.5 0.0.0.0 area 0
!
ip forward-protocol nd
no ip http server
no ip http secure-server
!
!
logging alarm informational
!
!
control-plane
!
!
!
mgcp fax t38 ecm
!
!
gatekeeper
 shutdown
!
!
line con 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
 stopbits 1
line aux 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
 stopbits 1
line vty 0 4
 login
!
End
```

Diyarbakır:

```
!

!
upgrade fpd auto
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname DIYARBAKIR
!
boot-start-marker
boot-end-marker
!
logging message-counter syslog
!
no aaa new-model
ip source-route
ip cef
!
!
no ip domain lookup
no ipv6 cef
multilink bundle-name authenticated
!
!
voice dsp waitstate 0
!
!
memory-size iomem 0
archive
 log config
  hidekeys
!
!
interface Loopback0
 ip address 10.41.41.9 255.255.255.255
!
interface Ethernet0/0
 no ip address
 shutdown
 duplex auto
!
interface GigabitEthernet0/0
 ip address 10.41.14.2 255.255.255.0
 duplex full
 speed 1000
 media-type gbic
 negotiation auto
```

```
 mpls ip
!
interface GigabitEthernet1/0
 ip address 10.41.12.2 255.255.255.0
 negotiation auto
 mpls ip
!
router ospf 100
 router-id 9.9.9.9
 log-adjacency-changes
 network 10.41.12.0 0.0.0.255 area 2
 network 10.41.14.0 0.0.0.255 area 2
 network 10.41.41.9 0.0.0.0 area 2
!
ip forward-protocol nd
no ip http server
no ip http secure-server
!
!
logging alarm informational
!
!
control-plane
!
!
mgcp fax t38 ecm
!
!
gatekeeper
 shutdown
!
!
line con 0
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
!
End
```

İzmir:

```
!

!
upgrade fpd auto
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
```

```
no service password-encryption
!
hostname IZMIR
!
boot-start-marker
boot-end-marker
!
logging message-counter syslog
!
no aaa new-model
ip source-route
ip cef
!
!
no ip domain lookup
ip domain name lab.local
no ipv6 cef
multilink bundle-name authenticated
!
!
voice dsp waitstate 0
!
!
memory-size iomem 0
archive
 log config
  hidekeys
!
!
interface Loopback0
 ip address 10.41.41.3 255.255.255.255
!
interface FastEthernet0/0
 no ip address
 shutdown
 duplex half
!
interface GigabitEthernet1/0
 ip address 10.41.1.1 255.255.255.0
 negotiation auto
 mpls ip
!
interface GigabitEthernet2/0
 ip address 10.41.2.1 255.255.255.0
 negotiation auto
 mpls ip
!
interface GigabitEthernet3/0
 ip address 10.41.3.1 255.255.255.0
```

```
 negotiation auto
 mpls ip
!
interface GigabitEthernet4/0
 ip address 10.41.4.1 255.255.255.0
 negotiation auto
 mpls ip
!
router ospf 100
 router-id 3.3.3.3
 log-adjacency-changes
 network 10.41.1.0 0.0.0.255 area 1
 network 10.41.2.0 0.0.0.255 area 1
 network 10.41.3.0 0.0.0.255 area 0
 network 10.41.4.0 0.0.0.255 area 1
 network 10.41.41.3 0.0.0.0 area 1
!
ip forward-protocol nd
no ip http server
no ip http secure-server
!
logging alarm informational
!
!
control-plane
!
mgcp fax t38 ecm
!
gatekeeper
 shutdown
!
!
line con 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
 stopbits 1
line aux 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
 stopbits 1
line vty 0 4
 login
!
End
```

Kayseri:

```
!
```

```
upgrade fpd auto
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname KAYSERI
!
boot-start-marker
boot-end-marker
!
logging message-counter syslog
!
no aaa new-model
ip source-route
ip cef
!
!
no ip domain lookup
no ipv6 cef
multilink bundle-name authenticated
!
!
voice dsp waitstate 0
!
!
memory-size iomem 0
archive
 log config
  hidekeys
!
!
interface Loopback0
 ip address 10.41.41.7 255.255.255.255
!
interface Ethernet0/0
 no ip address
 shutdown
 duplex auto
!
interface GigabitEthernet0/0
 ip address 10.41.11.1 255.255.255.0
 duplex full
 speed 1000
 media-type gbic
 negotiation auto
 mpls ip
!
interface GigabitEthernet1/0
```

```
 ip address 10.41.9.1 255.255.255.0
 negotiation auto
 mpls ip
!
interface GigabitEthernet2/0
 ip address 10.41.12.1 255.255.255.0
 negotiation auto
 mpls ip
!
router ospf 100
 router-id 7.7.7.7
 log-adjacency-changes
 network 10.41.9.0 0.0.0.255 area 0
 network 10.41.11.0 0.0.0.255 area 2
 network 10.41.12.0 0.0.0.255 area 2
 network 10.41.41.7 0.0.0.0 area 0
!
ip forward-protocol nd
no ip http server
no ip http secure-server
!
!
logging alarm informational
!
!
control-plane
!
mgcp fax t38 ecm
!
gatekeeper
 shutdown
!
!
line con 0
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
!
End
```

Kocaeli:

```
!
upgrade fpd auto
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
```

```
!
hostname KOCAELI
!
boot-start-marker
boot-end-marker
!
logging message-counter syslog
!
no aaa new-model
ip source-route
ip cef
!
!
no ip domain lookup
no ipv6 cef
multilink bundle-name authenticated
!
!
voice dsp waitstate 0
!
!
memory-size iomem 0
archive
 log config
  hidekeys
!
!
interface Loopback0
 ip address 10.41.41.2 255.255.255.255
!
interface Ethernet0/0
 no ip address
 shutdown
 duplex auto
!
interface GigabitEthernet0/0
 ip address 10.41.2.2 255.255.255.0
 duplex full
 speed 1000
 media-type gbic
 negotiation auto
 mpls ip
!
interface GigabitEthernet1/0
 ip address 10.41.5.2 255.255.255.0
 negotiation auto
 mpls ip
!
interface GigabitEthernet2/0
```

```
 ip address 10.41.6.1 255.255.255.0
 negotiation auto
 mpls ip
!
router ospf 100
 router-id 2.2.2.2
 log-adjacency-changes
 network 10.41.2.0 0.0.0.255 area 1
 network 10.41.5.0 0.0.0.255 area 1
 network 10.41.6.0 0.0.0.255 area 0
 network 10.41.41.2 0.0.0.0 area 0
!
ip forward-protocol nd
no ip http server
no ip http secure-server
!
!
logging alarm informational
!
!
control-plane
!
!
mgcp fax t38 ecm
!
!
gatekeeper
 shutdown
!
!
line con 0
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
!
End
```

Samsun:

```
!
upgrade fpd auto
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname SAMSUN
!
```

```
boot-start-marker
boot-end-marker
!
logging message-counter syslog
!
no aaa new-model
ip source-route
ip cef
!
!
no ip domain lookup
ip domain name lab.local
no ipv6 cef
multilink bundle-name authenticated
!
!
voice dsp waitstate 0
!
!
memory-size iomem 0
archive
 log config
  hidekeys
!
!
interface Loopback0
 ip address 10.41.41.6 255.255.255.255
!
interface Ethernet0/0
 no ip address
 shutdown
 duplex auto
!
interface GigabitEthernet0/0
 ip address 10.41.10.1 255.255.255.0
 duplex full
 speed 1000
 media-type gbic
 negotiation auto
 mpls ip
!
interface GigabitEthernet1/0
 ip address 10.41.8.1 255.255.255.0
 negotiation auto
 mpls ip
!
router ospf 100
 router-id 6.6.6.6
 log-adjacency-changes
```

```
 network 10.41.8.0 0.0.0.255 area 0
 network 10.41.10.0 0.0.0.255 area 2
 network 10.41.41.6 0.0.0.0 area 0
!
ip forward-protocol nd
no ip http server
no ip http secure-server
!
!
logging alarm informational
!
!
control-plane
!
!
mgcp fax t38 ecm
!
!
gatekeeper
 shutdown
!
!
line con 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
 stopbits 1
line aux 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
 stopbits 1
line vty 0 4
 login
!
end
```

Trabzon:

```
!
upgrade fpd auto
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname TRABZON
!
boot-start-marker
boot-end-marker
```

```
!
logging message-counter syslog
!
no aaa new-model
ip source-route
ip cef
!
!
no ip domain lookup
ip domain name lab.local
no ipv6 cef
multilink bundle-name authenticated
!
!
voice dsp waitstate 0
!
!
memory-size iomem 0
archive
 log config
  hidekeys
!
!
interface Loopback0
 ip address 10.41.41.8 255.255.255.255
!
interface Ethernet0/0
 no ip address
 shutdown
 duplex auto
!
interface GigabitEthernet0/0
 ip address 10.41.11.2 255.255.255.0
 duplex full
 speed 1000
 media-type gbic
 negotiation auto
 mpls ip
!
interface GigabitEthernet1/0
 ip address 10.41.10.2 255.255.255.0
 negotiation auto
 mpls ip
!
interface GigabitEthernet2/0
 ip address 10.41.13.2 255.255.255.0
 negotiation auto
 mpls ip
!
```

```
router ospf 100
 router-id 8.8.8.8
 log-adjacency-changes
 network 10.41.10.0 0.0.0.255 area 2
 network 10.41.11.0 0.0.0.255 area 2
 network 10.41.13.0 0.0.0.255 area 2
 network 10.41.41.8 0.0.0.0 area 2
!
ip forward-protocol nd
no ip http server
no ip http secure-server
!
logging alarm informational
!
!
control-plane
!
!
mgcp fax t38 ecm
!
!
gatekeeper
 shutdown
!
!
line con 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
 stopbits 1
line aux 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
 stopbits 1
line vty 0 4
 login
!
End
```

Van:

```
!
!
upgrade fpd auto
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
```

```
hostname VAN
!
boot-start-marker
boot-end-marker
!
logging message-counter syslog
!
no aaa new-model
ip source-route
ip cef
!
!
no ip domain lookup
ip domain name lab.local
no ipv6 cef
multilink bundle-name authenticated
!
!
voice dsp waitstate 0
!
!
memory-size iomem 0
archive
 log config
  hidekeys
!
!
interface Loopback0
 ip address 10.41.41.10 255.255.255.255
!
interface Ethernet0/0
 no ip address
 shutdown
 duplex auto
!
interface GigabitEthernet0/0
 ip address 10.41.13.1 255.255.255.0
 duplex full
 speed 1000
 media-type gbic
 negotiation auto
 mpls ip
!
interface GigabitEthernet1/0
 ip address 10.41.14.1 255.255.255.0
 negotiation auto
 mpls ip
!
interface FastEthernet2/0
```

```
 ip address 192.168.2.1 255.255.255.0
 duplex auto
 speed auto
 mpls ip
!
interface FastEthernet2/1
 no ip address
 shutdown
 duplex auto
 speed auto
!
router ospf 100
 router-id 10.10.10.10
 log-adjacency-changes
 network 10.41.13.0 0.0.0.255 area 2
 network 10.41.14.0 0.0.0.255 area 2
 network 10.41.41.10 0.0.0.0 area 2
 network 192.168.2.0 0.0.0.255 area 2
!
ip forward-protocol nd
no ip http server
no ip http secure-server
!
!
logging alarm informational
!
!
control-plane
!
!
mgcp fax t38 ecm
!
!
gatekeeper
 shutdown
!
!
line con 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
 stopbits 1
line aux 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
 stopbits 1
line vty 0 4
 login
```

!
end

ÖZGEÇMİŞ

1984 yılında Rize'de doğdu. Lise öğrenimini İstanbul Adnan Menderes Anadolu Lisesinde tamamladı. 2004 yılında girdiği İstanbul Ticaret Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği'nden 2009 yılında mezun oldu. 2010 yılı güz döneminde başladığı Kocaeli Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Bilgisayar Bilimleri Anabilim Dalı'ndaki Yüksek Lisans eğitimine devam etmektedir. 2013 yılından itibaren Kocaeli Büyükşehir Belediyesi Strateji Geliştirme Dairesi Başkanlığında Bilgisayar Mühendisi olarak çalışmaktadır.