DOKUZ EYLÜL ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

SÜRDÜRÜLEBİLİR BİR BİLGİ VE İLETİŞİM TEKNOLOJİLERİ ALTYAPISI TASARIMI

İbrahim Özden DUMAN

Eylül, 2020

İZMİR

SÜRDÜRÜLEBİLİR BİR BİLGİ VE İLETİŞİM TEKNOLOJİLERİ ALTYAPISI TASARIMI

Dokuz Eylül Üniversitesi Fen Bilimleri Enstitüsü

Yüksek Lisans Tezi

Bilgisayar Bilimleri Anabilim Dalı

İbrahim Özden DUMAN

Eylül, 2020

İZMİR

YÜKSEK LİSANS TEZİ SINAV SONUÇ FORMU

İBRAHİM ÖZDEN DUMAN, tarafından **DR. ÖĞR. ÜYESİ UĞUR ELİİYİ** yönetiminde hazırlanan "**SÜRDÜRÜLEBİLİR BİR BİLGİ VE İLETİŞİM TEKNOLOJİLERİ ALTYAPISI TASARIMI** " başlıklı tez tarafımızdan okunmuş, kapsamı ve niteliği açısından bir Yüksek Lisans tezi olarak kabul edilmiştir.

Dr.Öğr.Üyesi Uğur ELİİYİ

Yönetici

Dr. Öğr. Üyesi Serhat PEKER

Jüri Üyesi

Dr. Öğr. Üyesi Mete EMİNAĞAOĞLU

Jüri Üyesi

Prof. Dr. Özgür ÖZÇELİK
Müdür
Fen Bilimleri Enstitüsü

TEŞEKKÜR

Yüksek lisans eğitimim ve tez çalışmamın her aşamasında gösterdiği anlayış ve ilgiyle desteğini esirgemeyen ve her zaman bana zaman ayırmaktan imtina etmeyen tez danışmanım Dr. Öğr. Üyesi Uğur ELİİYİ'ye teşekkürü bir borç bilirim.

Hayatım ve eğitim hayatım boyunca bir an dahi maddi, manevi desteklerini eksik etmeyen ve bana sabır gösterip kendilerinden çok beni düşünen, iyi günümde kötü günümde yanımda olup binbir cefa ile beni okutan sevgili anneme, babama ve abime sonsuz minnettarım.

Yüksek lisans eğitimim için beni cesaretlendiren ve motivasyonum düştüğünde moral vererek desteğini esirgemeyip 2 yıl boyunca ders ve tez aşamasında sabrını eksik etmeyerek eğitim hayatımı kendinden ödün vererek destekleyen sevgili eşime ve bu süreçte varlığı ve gülüşü ile neşe kaynağım olan biricik kızıma sonsuz teşekkür ederim.

İbrahim Özden DUMAN

SÜRDÜRÜLEBİLİR BİR BİLGİ VE İLETİŞİM TEKNOLOJİLERİ ALTYAPISI TASARIMI

ÖZ

Birden fazla bilgisayar veya benzeri elektronik cihaz sistemlerinin üzerinde sürekli iletişim halinde bulunduğu, merkezi yönetim ve destek kolaylığı gibi çeşitli sebeplerden dolayı birbirine bağlanmasını sağlayan yapılara ağ altyapıları denir. Günümüzde hayatın her noktasına giren altyapıları doğru şekilde planlayıp, yönetmek ve altyapı üzerinde sunulan bir hizmetin performansını etkileyen faktörleri doğru şekilde ölçerek altyapının sürekliliğini, verimliliğini ve kullanıcı memnuniyetini sağlamak en önemli problemlerden biridir. Birçok altyapı tasarım hatalarından ve takip edilememelerinden kaynaklı sorunlardan dolayı ya kısa sürede atıl duruma düşmekte ya da verimlilikleri kullanıcılarını memnun etmemektedir. Bu yüzden bir altyapının tasarımından, onu yönetecek ekibin oluşturulmasına kadar dikkat edilmesi gereken birçok parametre vardır. Bu parametrelerin ölçümü ve takibine katkı sağlayacak örnek temel performans göstergeler için kapsamlı bir derleme çalışması yapılmıştır. Büyük ağ altyapılarının tasarımı, protokol seçimi ve yedekliği gibi tasarım seçenekleri ile ilgili örnek tasarım çalışmaları sunulmuştur. Hiyerarşik tasarımlarda sanal özel ağların varsayılan ağ geçitlerinin konumlandırıldığı yere göre ağ altyapısı üzerindeki performans etkilerini araştırmak için simülasyon test ortamları oluşturularak elde edilen sonuçlar, önerilen temel performans göstergeleri ile ilişkilendirilerek tasarım farklılıklarının altyapılara etkileri incelenmiştir.

Anahtar kelimeler: İletişim ağı tasarımı, ağ simülasyonu, hiyerarşik tasarım, iletişim teknolojileri standartları, ağ performans ölçümü, temel performans göstergeleri, ağ altyapı sürekliliği

DESIGN OF A SUSTAINABLE INFORMATION AND COMMUNICATIONS TECHNOLOGIES INFRASTRUCTURE

ABSTRACT

Structures, on which there is constant communication between computers or multiple electronic device systems, enabling them to be connected to each other for various reasons such as ease of central administration and support, are called network infrastructures. One of the most important problems of today is to plan and manage these infrastructures correctly that have entered every point of life, and to ensure the continuity, efficiency, and user satisfaction of the infrastructure by accurately measuring the factors affecting the performance of a service offered on itself. Many network infrastructures either become unusable in a short time due to design errors and inability to measure, or efficiency does not satisfy users. Therefore, there are many parameters to be considered from the design of infrastructure to the creation of a team for managing it. A substantial list of key performance indicators was compiled and proposed for monitoring and measuring these parameters. Design studies were carried out for large network infrastructure designs, protocol selection and redundancy. In the hierarchical designs, a simulation test environment was created to measure the performance effects of the virtual private networks on the network infrastructure according to the location of the default gateways. The effects of design differences on the infrastructure were examined by associating the results obtained from these tests with the proposed key performance indicators.

Keywords: Communications network design, network simulation, hierarchical design, communication technologies standards, network performance measurement, key performance indicators, network infrastructure continuity

İÇİNDEKİLER

Sayfa

ŞEKİLLER LİSTESİ

Sayfa

TABLOLAR LİSTESİ

Sayfa

BÖLÜM BİR
GİRİŞ

Birden fazla bilgisayar veya benzeri elektronik cihaz sistemlerinin üzerinde sürekli iletişim halinde oldukları, merkezi yönetim, destek kolaylığı, izleme kolaylığı ve daha kısa sürede sorun çözümü gibi çok çeşitli sebeplerden dolayı birbirine bağlanmasını sağlayan yapılara ağ altyapıları denir. Bu çalışmadaki ana amaç ağ altyapılarının tasarımı, sürekliliği ve verimliliği ile ilgili ölçümlere yönelik araştırmalar yapılması ve bu ölçümler sonucunda kullanıcı memnuniyetini sağlayan altyapı ve tasarımlardan örnekler sunulmasıdır.

Bilişim sistemleri altyapıları başlığı altında birçok başlık değerlendirilebilir fakat çalışma kapsamında daha çok ağ altyapıları ele alınmaktadır. BİT (Bilgi ve iletişim teknolojileri) sistemlerinde ağ altyapıları yönetimi bilgi, tecrübe gerektiren sistemlerdir. Bir ağ altyapısı hizmetinin sürekliliğini sağlamak için doğru tasarlanmış bir altyapı, performansı ölçülebilir teknik ekip, sürekli teknolojik gelişim, izleme, doğru insan gücü, doğru yatırım, eğitim ve tecrübe gerekmektedir. Bu bileşen ve süreçler özellikle farklı noktalarda yaygınlaşmış ve farklı sistemlerle bütünleşik çalışan büyük ağ sistemleri için geçerlidir. Bu tip büyük ağ sistemlerinin genel olarak görülebileceği yerlerin en yaygınları şu şekildedir:

- İnternet servis sağlayıcıları,
- Bankalar,
- Devlet kurumları,
- Farklı noktalara yaygınlaşmış şirketler,
- Belediyeler.

Araştırmada yapılan çalışmalar ve örnekler, akıllı şehir bağlamı ile ilişkilendirilmeye çalışılarak akıllı şehir ağ altyapıları tasarım çalışmalarına katkı sağlanmaya çalışılacaktır. Bu çalışma yerel ağları, kablosuz ağları, yazılım tabanlı

ağları ve geniş alan ağlarını kapsamaktadır. Tasarlanan ağ altyapılarının sürekliliği ve verimliliğiyle ilgili teknoloji standartları ve kütüphaneleri başta olmak üzere, ağ tasarımlarının değerlendirilmesi ve geliştirilmesi için yapılan akademik çalışmalar, ticari ürünlere ait teknoloji rehberleri, gerçek ortamlarda kullanılan ürün, topoloji, yöntem ve protokoller incelenmiştir. Bu incelemeler sonucunda gerçek ağ ürünlerinin çalışma prensiplerine uygun ve gerçek tasarımlarda kullanılabilecek ağ topoloji tasarım çalışmaları yapılmıştır. Bu çalışmalar protokollerin verimliliğine göre özelleştirilerek farklı topoloji çalışmaları sunulmuştur.

Tasarlanan ağ altyapılarının yatırıma dönüştürülmeden önce üzerinde oluşabilecek trafik yükünü taşıyıp taşıyamayacağı veya istenen verime ulaşıp ulaşılamayacağını önceden simüle ederek görmek mümkündür. Bu tür testleri sağlayabilecek simülasyon, emülasyon yazılım araçları ile gerçek ortam test yazılımları çalışma kapsamında ticari, açık kaynaklı ve akademik başlıkları altında incelenmiştir. Bu incelenen araçlar arasından akademik çalışmalar içinde kullanılan Mininet test yazılımı çalışma kapsamındaki testler için kullanılmıştır. Bu yazılım üzerinde farklı tasarım modellerinin ağ performansına etkileri araştırılmıştır.

Tasarlanan ve aktif kullanılan bir ağ altyapısının ölçülebilmesi, izlenebilmesi ve değerlendirilebilmesi için mevcutta bulunan veya çalışma kapsamında eklenen yeni göstergeler ile temel performans göstergeleri tablosu oluşturularak altyapı izlenebilirliğine katkı sağlanması amaçlanmıştır.

1.1 İlgili Literatür

Bilişim altyapılarının performansının ölçülebilmesi, takip edilmesi, verimliliklerinin artırılması ve daha iyi yönetilebilmesi için yapılmış birçok araştırma ve çalışma mevcuttur. Örneğin Al-Kuwaiti, Kyriakopoulos ve Hussein (2009), yaygın olarak kullanılan güvenilirlik, hata toleransı ve güvenlik gibi metrikleri incelemiş ve metriklerin tamamlayıcı özelliklerini belirlemek için sistematik bir yaklaşım sunmuşlardır. Altyapı düzeyindeki metrikleri ölçülebilir ve ölçülemez olarak

değerlendirerek tablo haline getirmişler ve bilgi güvenliği metriklerinin belirlenmesi için yöntem önerisinde bulunmuşlardır.

Bir diğer çalışmada metrik oluşturulması ile ilgili çerçeveler incelenmiş, metriklerin telekomünikasyon sistemlerinin tasarımı, analizi, işletimi ve yönetiminde önemli bir karar aracı olduğu belirlenmiştir (Al-Shehri, Loskot ve Mert, 2017). Aynı yazarlar metriklerin TPG (Temel Performans Göstergesi) olarak kullanıldığını ve Hizmet Seviyesi Anlaşmaları (SLA: Service Level Agreement) tanımlamak için hayati öneme sahip olduğunu belirtmişlerdir. TPG metrikleri genellikle müşteriler tarafından sistem veya hizmet benimsenme oranlarını artırmak için seçilirken, SLA metriklerinin hizmet sağlayıcıları ile son kullanıcılar arasındaki sözleşmeye dayalı anlaşmaları denetlemek üzere oluşturulduğunu vurgulamışlardır.

Bahnasse ve Kamoun (2015), farklı yönlendirme protokollerini test etmişler ve test edilen ağ altyapılarının davranışlarını inceleyerek, özellikle protokollerin yayılım süreleri, kuyruk gecikmesi ve işlem kapasitelerini karşılaştırmışlardır. Çalışma sonucunda özellikle geliştirilmiş iç ağ geçiti yönlendirme (EIGRP: Enhanced Interior Gateway Routing Protocol) protokolünün, dinamik çok noktalı sanal özel ağ çözümü üzerinde daha performanslı ve hızlı olduğu sonucuna varmışlardır.

Bholebawa ve Dalal (2016), YTA (Yazılım Tabanlı Ağlar) ile geleneksel ağ yapılarını karşılaştırarak OpenFlow tabanlı mimarileri incelemişlerdir. Ağ topolojisi performans analizlerinin bant genişliği kullanımı, paket iletim hızı ve uç düğümler arasındaki gidiş-dönüş yayılım gecikmesi gibi metriklerin karşılaştırılması ile yapılabileceğini belirtilmişlerdir. Chandan ve Utpat (2013), yönlendirme protokolleri arasında performans analizi yaparak EIGRP protokolünün daha hızlı olduğu sonucunu elde etmişlerdir.

Cicioğlu ve Çalhan (2016), YTA denetleyicilerinin birbirlerine göre farklılıklarına değinmiş ve YTA çözümlerinin geleneksel ağ mimarisinin ortaya çıkardığı birçok problem için çözümler sunabildiğini belirtmişlerdir. YTA'nın, yenilikçi ağ tasarımlarını karşılayabilen ve geleneksel ağ çözümlerine göre daha etkili

yapılandırmaya sahip, daha yüksek performans ve daha fazla esneklik sağlayan yeni bir yaklaşım olduğunu vurgulamışlardır.

Yönlendirme protokolleri ve uygulamaları üzerine yapılmış bir diğer çalışmada, bu protokollerin çeşitli özelliklere göre ne şekilde sınıflandırılabileceği açıklanmış ve kullanımı en yaygın yönlendirme protokolleri incelenmiştir (Karaman, 2006). Yaygın kullanılan yönlendirme protokollerinin ağ tasarımlarına etkilerini araştıran yazar, protokollerin ağ topolojilerine etki eden elverişli ve elverişsiz yönleri olabileceğini vurgulamıştır.

Masset ve Taburiaux (2018), YTA çözümlerini inceleyerek denetleyiciler ve OpenFlow protokolünü değerlendirmişlerdir. Özellikle POX denetleyicisi ve Mininet ile çalışmalar yürüterek farklı topolojiler üzerinde performans incelemeleri gerçekleştirmişlerdir. Ağ üzerinde istemci sayısı artıkça ağ üzerinde oluşan gecikmenin arttığını gözlemlemişlerdir. Nikolaev (2014), altyapı tasarımı, altyapılarda enerji tasarrufları ve YTA yapıları üzerine çalışmalar yapmıştır. YTA yapılarının ÇPEA (Çoklu Protokol Etiket Anahtarlama) çözümlerinde kullanımını değerlendirerek YTA yapılarına ait uygulama örnekleri vermiştir. Ayrıca çekirdek ağlar için kapasite planlamasının nasıl yapılacağı ve yönetilen ağ altyapılarında talep edilmesi olası trafik miktarının hesaplama yöntemlerinden örnekler sunmuştur.

McGill (2007), hizmet kalitesi parametreleri olarak bilinen ağ performans ölçütleri sınıfını araştırmış, kullanıcı memnuniyetinin ve uygulama performansının ölçüm zorluklarından bahsederek kullanıcı şikâyetine kadar uygulamada yaşanan performans problemlerinden genellikle haberdar olunamadığını belirtmiştir. Ağ performansının değerlendirilebilmesi için kullanıcı veya uygulama merkezli ölçümlere odaklanılması gerekliliğini vurgulayarak performans değerlendirmeleri için paket performanslarının ölçülebileceğini belirmiştir. Aynı çalışmada ayrıca, ağ problemlerinin öngörülebilmesi için takip edilen metriklerin birden fazla düzeyde izlenmesini önermiştir.

Rohani ve Roosta (2014), sistem erişilebilirliğinin hesaplanması konusundaki çalışmalarında, kritik bir uygulamada “Erişebilirlik” seviyesinin dikkate alınması

gereken ilk kriter olduğunu belirtmişlerdir. İlk arızaya kadar geçen zaman (MTTF: Mean Time to Failure), arızalar arasında geçen ortalama zaman (MTBF: Mean Time between Failures) ve ortalama onarım zamanı (MTTR: Mean Time to Repair) tanımlamalarını yaparak ölçüm yöntemlerinden örnekler vermişlerdir. Ayrıca bir sistemin arızalanma olasılığının genellikle MTTF ve/veya MTBF parametreleri ile ölçülebileceğini belirtmişler, erişilebilirlik hesaplama parametrelerine ilişkin aşağıdaki sonuçlara varmışlardır:

- MTBF değeri ne kadar yüksek olursa, sistemin güvenilirliği ve kullanılabilirliği de o kadar yüksek olur;
- MTTR kullanılabilirliği etkiler, diğer bir deyişle sistemin bir arızadan kurtarılmasının uzun zaman alması sistemin düşük bir kullanılabilirliğe sahip olacağı anlamına gelir;
- MTBF, MTTR'ye kıyasla çok büyükse yüksek kullanılabilirlik elde edilebilir.

Shorfuzzaman ve Masud (2015), iletim denetimi protokolünün (TCP: Transmission Control Protocol) uçtan uca gecikme performansını değerlendirmişler, gecikmeyi karakterize etmek doğrultusunda bir simülasyon modeli geliştirerek test etmişlerdir. Diğer bir çalışmada bulut veri merkezi ağlarına ait topolojik yapılar incelenmiş ve bu ağların merkezilik ve dirençlilikleriyle ilgili kritik düğüm ve bağlantıların belirlenmesinde kullanılacak topolojik metrik hesaplamaları için bir yöntem önerilmiştir (Qi, Liu, Gutierrez ve Narang, 2017). Veri merkezi ağının topyekün performansını ölçmeleri dışında, gerçekleştirdikleri simülasyon testleriyle ağ topolojisi ve trafik yükünün toplam enerji tüketimi başta diğer performans ölçütlerinde önemli bir etkisi olduğunu göstermişlerdir.

Ağ performansı odaklı mikro-teknolojik performans göstergeleri dışında, hem akıllı şehir bağlamında hem de yatırımların ekonomik etkileri genelinde BİT'i ilgilendiren makro düzeyde değişken ve parametrelerin incelendiği çalışmalar da mevcuttur (Akçura ve Avci, 2014; Bülbül, Topal, Büyükkeklik ve Demirer, 2016).

Tezin sonraki bölümlerinde incelenen çalışma başlıkları şu şekilde belirlenmiştir. İkinci bölümde ağ altyapılarıyla ilgili BİT standartları özetlenmiş, üçüncü bölümde ise ağ altyapı bileşen, tasarım ve süreçleriyle ilgili en önemli kavram ve terimler ayrıntılı olarak sunulmuştur. Literatürden derlenmiş ve önerilen temel performans göstergelerini içeren dördüncü bölümden sonra, tercih edilen güncel ve popüler ağ simülasyon araçlarıyla gerçekleştirilmiş örnek testlere ait sonuçlar beşinci bölümde performans göstergeleri bazında paylaşılmıştır. Altıncı ve son bölümde, tez çalışması kapsamında yapılan araştırmalar ışığında ağ altyapısı tasarımı ve performans ölçümleriyle ilgili tespitler sunulmuş ve olası gelecek çalışma konuları tartışılmıştır.

BÖLÜM İKİ
BİLGİ VE İLETİŞİM TEKNOLOJİLERİ STANDARTLARI

Bilgi ve iletişim teknolojileri (BİT) kısa bir zamanda günlük hayatın vazgeçilmez ayrılmaz bir parçası haline gelmiş ve yaşamın her alanına yerleşmiştir. Önceden kamu veya özel kuruluşlar tarafından sadece yerinde verilen birçok hizmet artık BİT ile uzaktan verilebilir hale gelmiştir. BİT altyapıları üzerinde verilen hizmetlerin gün geçtikçe artması ile bunların sürdürülebilirliğinin sağlanması önemli yatırım konularından biri haline gelmiştir. Sürdürülebilirlik, yönetim, güvenlik gibi ihtiyaçların ortaya çıkması ile BİT kütüphaneleri veya standartları geliştirilmiş ve kullanılmaya başlanmıştır. Tez çalışması özelinde bu bölümde yönetim ve güvenlik standartlarından başlıcaları ele alınmıştır.

Kurulacak ve yönetilecek bir altyapıdaki tüm süreçler için ITIL (Information Technologies Infrastructure Library - Bilgi Teknolojisi Altyapı Kütüphanesi), tüm BT (Bilgi Teknolojileri) hizmetlerinin yönetim seviyesinde takibi ve ölçülmesi için COBIT (Control Objectives for Information and Related Technology - Bilgi ve İlgili Teknolojiler İçin Kontrol Hedefleri) ve genel güvenlik süreçleri ve farkındalık için ISO (International Organization for Standardization - Uluslararası Standartlar Kurumu) 27001 standart ve/veya kütüphanelerinden yararlanılması BT hizmetlerine önemli katkı sağlayacaktır. Bu standart ve/veya kütüphanelerin yanı sıra literatürde bulunan diğer standartlar da uygulanabilir.

2.1 ITIL

1980'lerde İngiliz Merkezi Bilgisayar ve Telekomünikasyon Ajansı tarafından, BİT hizmetlerindeki problemlerin tespiti ve hizmet kalitesinin geliştirilmesi amacı ile temelleri atılmış ve farklı sürümler şeklinde günümüze kadar gelmiş olan ITIL, BT hizmetlerini belirli bir kalite seviyesinde yönetmek üzere geliştirilmiş hizmet yönetim metodolojisidir.

ITIL, iş-süreç yaklaşımı sayesinde müşteri, tedarikçi, BT bölümü ve kullanıcıları arasında başarılı bir iletişim kurulmasını sağlamaktadır. ITIL, "En iyi uygulamalar / deneyimler" üzerine yapılandırılmıştır ve BT hizmet yönetimi ve dağıtımı süreçleri ile dünyada BT yönetiminde yaygın olarak kullanılan ve kabul gören bir standarttır. ITIL sürüm 2 2001'de yayımlanmıştır. 2007 yılında ITIL'in üçüncü sürümü bir giriş ve beş temel kitap olarak; son sürümü olan ITIL v4 ise 2019 yılında yayımlanmıştır (Türkiye Bilişim Derneği, 2020).

ITIL, hizmet yönetimi süreçleri için kullanılan, hizmet yönetimini sürdürmek için yol gösteren ve hizmet sağlama süreçlerini ayrıntılı şekilde gösteren farklı kitap ve sürümlerden oluşan bir hizmet süreç yönetim kütüphanesidir. ITIL süreçlerinin nasıl uygulanacağı BT yöneticileri tarafından kurumun altyapı, teknoloji ve yönetim yapısına göre belirlenmelidir.

ITIL beş ana süreçten oluşmaktadır (Aydoğdu, 2016a):

- Hizmet stratejisi:

  İhtiyaç olunan teknoloji ile maliyeti arasındaki dengenin korunarak uygun maliyetli bir BT stratejisi oluşturulmasını sağlayan adımdır. Ne yapılacağını ve neden ihtiyaç duyulduğunu tanımlar. Bir hizmetin sürekliliğinin başarılı bir şekilde sağlanabilmesini ve kullanıcılara yeni hizmet verilmesi aşamasında stratejik düşünme ve strateji geliştirme kabiliyetlerinin kazandırılmasını amaçlar. Hizmet stratejisi, BT'nin hizmet verdiği kurumun hizmet beklentilerini daha iyi algılamasına ve sonucu olarak daha fazla memnuniyet elde etmesine olanak sağlar.
- Hizmet tasarımı:

  BT hizmetlerinin nasıl inşa edileceğini tanımlar. Verimli ve genel BT hizmetleri ile uyumlu, etkili hizmetlerin tasarlandığı adımdır. Strateji aşamasında alınan kararlara göre hizmet tasarımları oluşturulur.
- Hizmet dönüşümü:

  Hizmet tasarımı adımında hazırlandığı şekilde BT hizmetlerini inşa etmeyi ve testlerin yapıldığı, test sonuçlarına göre hizmetlerin uygulamaya alındığı adımdır.

- Hizmet uygulaması:

  Kullanıcılara kabul edilebilir seviyede hizmet sunmayı ve sunulan hizmetleri en iyi şekilde yönetmeyi amaçlar. Uygulamaya alınan hizmetlerin müşteriye/kullanıcıya sunulduğu ve işletildiği kısımdır.
- Sürekli hizmet geliştirme:

  BT süreçleri ve metriklerinin sürekli iyileştirilmesini amaçlar. Müşteriye sunulan hizmetler sürekli ölçülerek, bu adımda kötü giden noktaların tespit edilerek iyileştirilmesi amaçlanır.

ITIL ile sağlanan faydalar (Kaynak, 2011):

- Hizmet kalitesinde artış: BT hizmetlerinin müşteri odaklı olması ile sunulan hizmet kalitesinin sürekli artırılarak memnuniyet derecesinin yükseltilmesini sağlar.
- Maliyet takibi: BT maliyetlerinin ayrıntılı şekilde takibi ve yönetilmesini olanak sağlar.
- Etkin iletişim: Kullanıcı ile BT arasında veya BT ekipleri arasındaki iletişim ve süreçlerin geliştirilmesini sağlar.
- Süreç temelli yaklaşım: Hizmet kalitesinin TPG(Temel Performans Göstergesi)'leri ile izlenmesi, hizmet verimliliğin artmasını olanak sağlar.
- Güçlü BT organizasyonu: BT yapısındaki her bir parametrenin TPG'leri aracılığı ile izlenmesi ve değerlendirilmesi sonucu BT organizasyon verimliliğinin sürekli güçlendirilmesini sağlar.
- Kullanıcı memnuniyeti: BT organizasyonuna daha profesyonel bir yapı kazandırılması, verilen hizmetlerin kalitesinin yükselmesi sonucu kullanıcıların memnuniyetinin ve BT hizmetlerine olan güvenlerinin artmasını sağlar.

2.2 COBIT

1996 yılında BT yönetimi ve BT denetimi için ISACA (Information Systems Audit and Control Association - Bilgi Sistemleri Denetim ve Kontrol Derneği) tarafından

oluşturulmuş olan COBIT yöneticilerin kontrol gereksinimleri, teknik konular ve iş riskleri arasında köprü görevi gören bir çerçevedir (Aydoğdu, 2016b). COBIT, BT yönetiminde daha iyi ve verimli yönetim için ulaşılması gereken hedefleri ortaya koymaktadır. COBIT, ITIL gibi diğer standartlardan farklı olarak BT süreçlerine değil BT'nin yönetimine odaklanmaktadır (Tutu, 2010). COBIT kontrol hedeflerinden oluşmaktadır ve bu hedeflere nasıl ulaşılacağına dair yöntem veya çözüm önermez. Fakat denetim hedefleri ile ilgili örneklemelere ihtiyaç duyulabileceği için ISACA tarafından "en iyi uygulamalar" ek dokümanı yayımlanmıştır.

COBIT 1'in kapsamı "denetim" kavramı ile başlamıştır. Her yeni çıkan sürüm ile COBIT'in kapsamına farklı kavramlar eklenmiştir. COBIT 2 ile beraber "kontrol" kavramı, COBIT 3 ile birlikte "yönetim" kavramı çerçeve kapsamına girmiştir. Cobit 1,2 ve 3 sürümlerindeki kavramların birleşmesiyle COBIT bir BT yönetim çerçevesi haline gelmiştir. COBIT 4 ve COBIT 4.1 sürümlerinde "BT yönetişimi" çerçeve kapsamına girmiş ve son olarak COBIT 5 ile "kurumsal BT yönetişimi" kavramı çerçeveye dâhil edilerek günümüzde kullanılan sürümüne ulaşılmıştır.

COBIT yöneticilere, BT kullanıcılarına ve denetçilere çeşitli faydalar sağlamaktadır. COBIT'in yöneticilere sağladığı yarar, BT alanında daha etkili kararlar verme ve kuruluş için doğru yatırımlar yapma olanağı sağlaması olarak açıklanabilir. BT kullanıcılarının COBIT'ten yararlanma nedeni, kontrol, güvenlik ve süreç yönetimi güvencesi sağlanmasıdır. Denetçilere ise BT altyapısı içindeki sorunları kontrol esasları çerçevesinde belirlemelerinde yardımcı olmaktadır (Aydoğdu, 2016a).

COBIT 5, beş temel prensibe dayanır (Efe, 2019):

- Paydaş ihtiyaçlarının karşılanması,
- Kuruluşun baştan sona kapsanması,
- Tek bir bütünleşik çerçevenin uygulanması,
- Bütüncül bir yaklaşımın sağlanması,
- Yönetişimin yönetimden ayrılması.

COBIT5'in BT'ne yönelik amaçları şunlardır (Artinyan, 2007; Aydoğdu, 2016b):

- Varlıkların korunması,
- Bilginin doğruluk ve bütünlüğü,
- Etkinlik denetimi,
- Verimlilik denetimi,
- Güvenirliğin sağlanması,
- Gizliliğin sağlanması.

COBIT 5'in genel faydaları (Duygun, 2016):

- İşe yönelik kararların verilmesi noktasında sistematik bir yaklaşım, ortak dil ve anlayış sağlar.
- BT'ye yönelik yatırımlardan değer elde edilmesini sağlar.
- BT ile ilgili risklerin kabul edilebilir seviyede yönetilmesini sağlar.
- BT maliyetlerinin etkin şekilde yönetilmesini sağlar.
- Paydaşların ihtiyaçlarını karşılamaya yardımcı olur.
- Teknoloji ile inovasyonu mümkün kılan güven sağlar.

2.3 ISO 27001

1995 yılında İngiltere hükümetinin ve sanayi kuruluşlarının talepleri ile BSI (British Standards Institute - İngiliz Standartlar Enstitüsü) kuruma yazdırılmıştır. İlk BS 7799 kodu ile tanımlanmıştır. ISO ise BS 7799 kodlu bilgi güvenliği ile ilgili standardın revize edilmiş halinin birinci bölümünü 2000 yılında kabul ederek 17799 kodu ile yayımlamıştır (Bayram, 2016). Bilgi Güvenliği Yönetimi Sistemi (BGYS) deyimi ilk kez BSI tarafından 1998 yılında BS 7799 standardının revize ederek yayımlanan hali olan BS 7799-2 standardı bölüm 2 içinde kullanılmıştır. Daha sonra bölüm 2 2005 yılında ISO tarafından standart olarak kabul edilerek ISO/IEC 27001:2005 olarak yayımlanmıştır.

ISO/IEC 27001, BGYS gereksinimlerini tanımlayan tek uluslararası denetlenebilir standarttır (International Organization for Standardization, 2013). Yeterli ve orantılı

güvenlik denetimleri seçilmesini sağlamak için tasarlanan ISO/IEC 27001'in temel amacı hassas bilginin korunmasıdır. BGYS, kurumun çalışanları, iş süreçlerini ve BT sistemlerini kapsayan ve kurumun hassas bilgilerini yönetebilmek amacıyla benimsenen sistematik bir yaklaşımdır. BGYS sayesinde kurumlar BT altyapılarına yönelik olası tehlikeleri analiz ederek, BT altyapılarına yönelik risklerin oluşması durumunda hangi kontrollerin uygulanacağına önceden sistematik bir yaklaşımla belirlenir (ISO, 2013).

ISO 27001 BGYS kurmanın yararları:

- Kurumda hangi bilgi varlıklarının olduğunu tespit edilip önem sırası oluşturularak bilgi varlıkları farkındalığı oluşturulur.
- Önceden belirleyeceği kontrol yöntemleri ile risklere karşı korunma yöntemlerini belirlenip uygulanarak sahip olunan varlıklar korunur.
- Bir felaket senaryosunda, önceden kritikliğe göre belirlenen planlar ile iş sürekliliği sağlanır.
- Bilgiyi sistematik kontroller ile korur.
- Kurumun bütününe bilgi güvenliği farkındalığı kazandırılır.
- Çalışanların motivasyonunu arttırır.
- Yasal yükümlülüklere uyum kolaylaşır, yasal takip ve kovuşturmalar önlenir.
- Kurumun saygınlığını ve güvenirliğini artırır.
- Bilgi varlıklarına yönelik risklere karşı önceden alınacak önlemler belirlenir.
- Sürekli kontroller ile bilgi güvenliğine yönelik açıklar tespit edilir ve önceden önlem alınır.

ISO 27001 BGYS kurma aşamaları (KAS, 2010):

- Varlıkların sınıflandırılması,
- Gizlilik, bütünlük ve erişebilirlik kriterlerine göre varlıkların değerlendirilmesi,
- Risk analizi,

- Risk analizi çıktılarına göre uygulanacak kontrollerin belirlenmesi,
- Dokümantasyonun oluşturulması,
- Kontrollerin uygulanması,
- Kurumsal farkındalık oluşturulması,
- Zaafiyet analizi,
- İç tetkik,
- Prosedürlerle belirtilmiş kayıtların tutulması,
- Belgelendirme.

Bu bölümde ana hatlarıyla özetlenmiş önemli BT hizmet standartlarını takiben, bir sonraki bölümde öncelikle BİT kapsamında iletişim ağ altyapılarındaki önemli bileşenler, ilgili kavram ve terimlerle beraber sunulmaktadır. Sonrasında ise ağ altyapı tasarımıyla ilgili başlıca topolojiler, bu tasarımları doğrudan veya dolaylı etkileyen belli başlı protokol türleri ve eşlik eden süreçler paylaşılmaktadır.

BÖLÜM ÜÇ
BİLGİ VE İLETİŞİM TEKNOLOJİLERİNDE AĞ ALTYAPILARI VE TASARIMI

Günümüzde BİT altyapıları neredeyse her kurumda yer edinerek birçok hizmetin sunulmasına olanak sağlamaktadır. Hızla gelişmekte olan BİT, düşük maliyetli, hızlı, yüksek kaliteli altyapı ihtiyacını da beraberinde getirmiştir. Artan kurumsal iletişim ihtiyaçları ile de telekomünikasyon sistemleri BT sistemlerinin ayrılmaz bir parçası olmuştur. Dijitalleşme ile altyapılar üzerinde sunulan hizmetler artması ile kaliteli altyapılara olan ihtiyaç da gün geçtikçe artmaktadır. Telekomünikasyon sistemlerinin kurum içindeki diğer iletişim sistemlerine entegrasyonu ise adeta bir zorunluluk haline gelmektedir. Bugün kurumsal bir firmada ağ altyapıları üzerinden masa telefonlarının, görüntülü konferans sistemlerinin, yazılımsal telefonların, kurumsal uygulamaların ve daha birçok kurumsal uygulamanın kullanıldığını görmek mümkündür. Kurumsal ihtiyaçlar ve teknolojinin hızlı bir şekilde değişmi ve gelişmi göz önüne alındığında ihtiyaçlara uzun vadede cevap verebilecek ve süreklilik arz eden bir altyapının tasarlanması ihtiyaçtan çok zorunluluğa dönüşmektedir. Bu dönüşüme en hızlı uyum sağlayabilen ve uzun vadeli ihtiyaçlara cevap verebilecek tasarımları yapan kuruluşlar, oluşabilecek farklı durumlara maksimum düzeyde hazırlıklı olarak en az iş kaybı ile çalışırlıklarını sürdürebileceklerdir. Bunun en iyi örneği günümüzde etkileri devam eden COVID-19 hastalığı salgınının olumsuz etkilerinden dolayı evden çalışma zorunluluğudur. Altyapısını oluşabilecek riskleri düşünerek ve her türlü ihtiyaca cevap verebilecek şekilde tasarlayan kurumlar görüntü, ses, uzak bağlantı, ağ trafiği kapasite ihtiyaçlarını karşılayarak bu son pandemi sürecinde hızlı bir şekilde uzaktan çalışmaya uyum sağlayabilmişlerdir.

İhtiyaçlara uzun vadede cevap verebilecek bir altyapı için ise en uç noktadan en tepeye kadar ayrıntılı şekilde bir tasarım yapılmalı ve oluşabilecek uzun vadeli ihtiyaçlar bu tasarım süreci aşamasında göz önünde bulundurulmalıdır.

3.1 Ağ Altyapıları ve Temelleri

İki veya daha fazla bilgisayarın veya cihazın kablolu veya kablosuz bir şekilde birbirine bağlanarak veri alışverişinde bulunmasına sağlayan yapıya ağ denir (İTÜ BİDB, 2013c). Ağ altyapıları ise, sistemlerin çalışırlıklarını sürdürebilmeleri için gerekli olan birimler arası veri iletimini sağlayan yapı olarak tanımlanabilir. Ağ altyapıları sistemin verimliliği ve kullanıcı memnuniyeti için en belirleyici unsurlardan biridir. Ağ altyapıları güvenlik, eğitim, devlet, sağlık ve iletişim sektörleri gibi birçok alanda kullanılmaktadır. Bu altyapıların kurulumları genellikle oldukça maliyetli olduğu için uzun ömürlü ve sürdürülebilir olarak tasarlanmaları gereklidir. Bu yüzden, kurulum ve planlama aşamasında oldukça dikkatli olunmalıdır.

Bir ağ altyapısı, bilgisayarları veya cihazları birbirine bağlayan aşağıda listelenmiş pasif veya aktif bileşenleri içerir:

Pasif bileşenler:

- Kablolar,
- Kabinler,
- Paneller,
- Prizler,
- Kablo Kanalları.

Aktif bileşenler:

- Bilgisayar ve sunucular,
- Kenar anahtar/Yönlendirici/Güvenlik duvarı,
- Kablosuz ağ cihazları.

İyi planlanmayan bir altyapı yatırımı kısa sürede istekleri cevaplayamayan bir yatırıma dönüşerek atıl hale gelebilir. Eğer uzak ofis gibi az sayıda ağ cihazından oluşan bir altyapı kuruluyorsa yanlış planlamanın etkisi hissedilmeyebilir. Ama örneğin akıllı şehir bağlamında büyük bir ağ altyapısı kurulacaksa, projenin önemli bir kısmı planlanmaya ayrılmalıdır. “Altyapı üzerinde hangi hizmetler verilecek; ne kadar

trafik oluşması bekleniyor; hangi teknolojiler kullanılacak; hangi cihazlar tercih edilecek; yedeklik nasıl olacak; yatırımın kaç yıl boyunca talepleri karşılaması bekleniyor; genişleyebilirlik ölçüsü ne olacak?" gibi birçok sorunun cevabının planlanma aşamasında sorularak bulunması gerekmektedir.

Kurulacak ağ altyapılarında kullanılacak teknolojiler projenin bütününü ve genişleyebilirliğini göz önünde bulundurarak seçilmelidir. Örneğin, "Omurga anahtar yedekliği için sadece seçilen marka cihazlarında çalışan bir protokol mü ya da sanal yönlendirici fazlalık protokolü (Virtual Router Redundancy Protocol: VRRP) gibi tüm cihazlarda desteklenmesi muhtemel standartlar mı tercih edilmeli?" sorularının cevapları tasarım öncesinde artı ve eksileriyle değerlendirilerek belirlenmelidir.

Sonuç olarak iyi planlanmış bir ağ altyapısı sürdürülebilir, genişleyebilir ve yönetimi kolay bir yapı oluşturacaktır.

3.2 Ağ Cihazları

Ağ cihazları uç sistem konumunda olan bilgisayar veya benzeri sayısal sistemlerin birbirleriyle karşılıklı çalışmalarını, iletişime geçmelerini sağlayan ara cihazlardır (MEB, 2011). Bu ağ cihazları, yerel ağlarda veya geniş alan ağları içerisindeki veri iletimini sağlayabilirler.

3.2.1 Yönlendiriciler

IP (Internet Protokolü) paketlerini, bir ağdan başka bir ağa ileten cihazlara yönlendirici denir (Yaşar, 2011). Yönlendirici cihazlar ağlar arasında adreslerin yönlendirilmesinde rol oynarlar ve bu yönlendirme işlemlerini IP adreslerine göre yaparlar. Bir yönlendirici en az iki ağ kartına bağlı olmalıdır, yani fiziksel olarak birbirinden farklı iki ağa bağlı olması gerekmektedir. Şekil 3.1'de bir yönlendirici örneği görülmektedir.

Temelde yönlendiriciler aşağıdaki bileşenlerden oluşur;

- Sadece okunabilir bellek,
- Rastgele erişim belleği,
- Ana kart,
- İşlemci.

Bunun dışında yönlendiriciler üzerinde yapılandırma yapılmasını sağlayan konsol ara yüzleri mevcuttur.

Şekil 3.1 Örnek yönlendirici

3.2.2 Kenar Anahtarlar

Kenar anahtarları ağ ortamını çakışma bölgelerine ayıran ve OSI (Open Systems Interconnection - Açık Sistem Arabağlantısı) modelinin Katman 2 veya Katman 3 katmanlarında çalışan cihazlardır. Katman 2 erirşimlerini üzerlerinde yer alan MAC (Media Access Control - Ortam Erişim Kontrolü) adres tabloları aracılığı ile yaparken Katman 3 erişimlerini IP adresleri üzerinden yönlendirme tabloları aracılığı ile ağ içerisindeki diğer cihazlarla iletişim sağlarlar. Yerel ağlar içinde iletişim ve bağlantılar bu cihazlar üzerinden yapılır. Üzerlerinde bulunan konsol arayüzü aracılığı ile ya da web arayüzünden yapılandırılabilirler. Kenar anahtarlar Şekil 3.2'de örneklendirildiği gibidir.

Şekil 3.2 Örnek kenar anahtar

3.2.3 Kablosuz Ağ Denetleyicileri

Kablosuz erişim noktalarını tek bir noktadan kontrol etmek için kullanılan cihazlardır. Cihazların üzerlerinde çalışan yazılım yönlendirici ve kenar anahtarlardan farklılık gösterir. Genellikle Web ara yüzleri kullanılarak programlanır. Özellikle erişim noktalarının sinyal yayım güçleri otomatik olarak yapılandırılarak ve erişim noktalarının birbirlerini etkilemeleri engellenerek kablosuz altyapının verimli çalışması sağlanır. Kablosuz denetleyici ve erişim noktası örnekleri Şekil 3.3 ve Şekil 3.4'te paylaşılmıştır.

Şekil 3.3 Örnek kablosuz denetleyici

Şekil 3.4 Örnek erişim noktası

3.2.4 Güvenlik Duvarları

Güvenlik duvarları, önceden belirlenmiş güvenlik kurallarına göre hem giden hem de gelen ağ trafiğini izleyerek değerlendiren ve bu değerlendirme sonucuna göre karar verip yöneten güvenlik cihazı veya güvenlik yazılımlarıdır. Güvenlik duvarlarının temel çalışma prensibi, gelen veri paketlerinin incelenmesi ve kurallara uymayanların

geçişlerinin yasaklanması şeklinde özetlenebilir (Berqnet, 2019). Sahip olunan lisanslara göre web filtreleme, virüs kontrolü, uygulama kontrolü gibi güvenlik kontrollerini gerçekleştirebilirler.

Şekil 3.5 Örnek güvenlik duvarı

Şekil 3.5'te bir güvenlik duvarı örneğine yer verilmiştir.

3.2.5 Büyük Ölçekli Ağ Altyapılarının Tasarımı

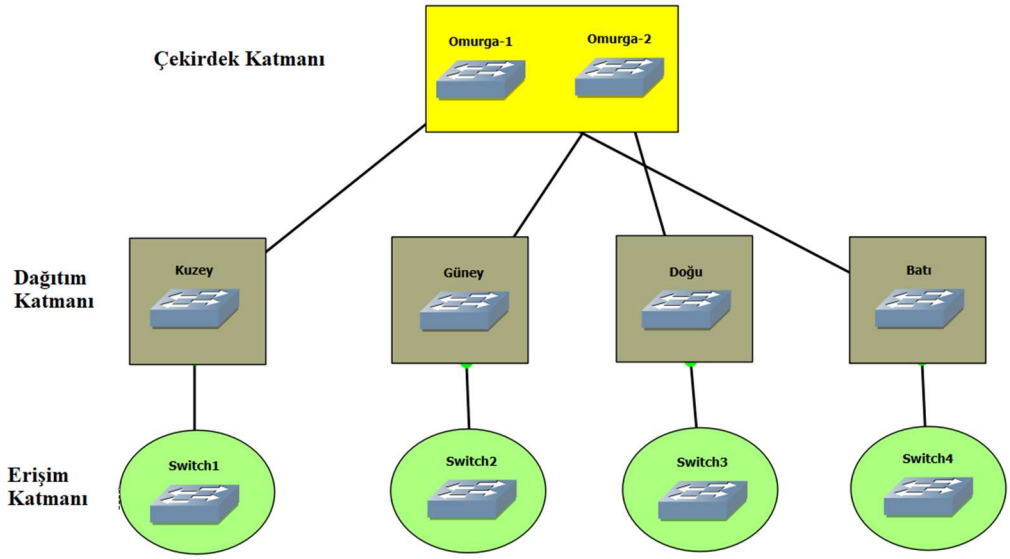
Ağ tasarımı, bir kurumun BT altyapısının kurulumdan önce ağ üzerinde yer alan tüm öğelerin istenen verimlilikte birlikte çalışırlığının proje öncesi planlanması aşamasıdır. Büyük ölçekli ağ altyapılarında süreklilik önemlidir ve kesintilere tahammül süreleri düşüktür. Doğru bir tasarım büyük ölçekli ağ altyapıları üzerinde daha yüksek işletim verimliliği sağlayacaktır. Ağ tasarımı genellikle ağ mimarları, BT yöneticileri ve diğer ilgili çalışanlar tarafından gerçekleştirilen bir görevdir.

Ağ planlama ve tasarımı, tekrarlayan bir süreçtir. Süreç her yeni ağa veya hizmete göre yeniden uyarlanabilir. Tasarım, genişleyebilirlik, sürdürebilirlik ve verimlilik ihtiyaçları göz önünde bulundurularak yapılmalıdır. En pahalı ürünler kullanılarak yapılan bir tasarım doğru tasarım yapıldığı anlamına gelmez. Önemli olan ihtiyaçlara uygun en az maliyetle hedeflenen verimliliği sağlayabilen bir altyapının tasarlanmasıdır. Örneğin, sadece bir kişinin çalıştığı ve genişleme beklenmeyen bir bölgeye 48 bağlantı ara yüzüne sahip Katman 3 bir kenar anahtar konumlandırılması doğru bir yatırım olmayacaktır. Bu yüzden ihtiyaca uygun tasarım yapılması proje maliyetlerini düşürecektir.

3.3 Hiyerarşik Ağ Altyapı Tasarımı

Hiyerarşik tasarım bir ağ altyapısının daha kolay yönetilmesini, trafiğin doğru cihazlar üzerinden ilerlemesine ve gereksiz yere cihaz kapasitelerinin tüketilmesini engelleyerek genişleyebilirlik, erişebilirlik ve sürdürülebilirliği artırır. Hiyerarşik tasarım üç katmana ayrılarak tasarlanır. Ağ cihazları görevlerine ve işlevlerine göre üç farklı katmanda konumlandırılırlar. Üç katmanlı yapı çekirdek, dağıtım ve erişim katmanlarından oluşmaktadır.

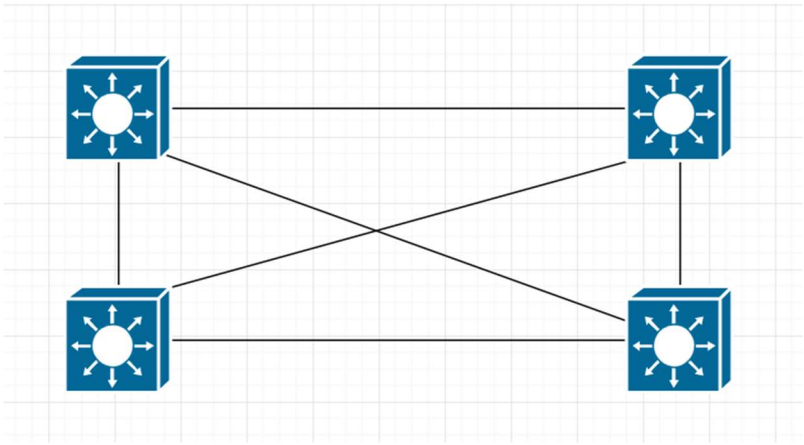
Örneğin, bir futbol sahasının dört bölgeye ayrıldığı, bu bölgelerin kuzey, güney, doğu, batı şeklinde isimlendirildiği ve bölgelerin her birinde dağıtım katmanı cihazlar konumlandırarak toplama noktaları oluşturulduğu bir senaryoda, bu dört toplama noktası için merkez sistem odasında yedekli çekirdek katman omurga kenar anahtarları konumlandırıp toplama noktaları ile bağlantıların sağlandığı varsayılsın. Bu örnek kâğıt üzerinde tasarlandığında ağ topolojisi Şekil 3.6'ya benzer olacaktır. Bu şekilde tasarlanmış bir ağ altyapısında her bölgenin trafiği kendi içinde sadece kendi toplama noktası içinde dönecektir. Örneğin; Doğu toplama noktasının trafiği sadece bu bölgede ilerleyecek, batı toplama noktasına veya omurga noktasına ulaşıp gereksiz yere kapasite tüketmeyecektir. Ama Doğu toplama noktasından bir Batı toplama noktasındaki bir cihaza erişilecekse o zaman trafik omurga kenar anahtara kadar çıkıp Batı toplama noktasına ulaştırılacaktır. Ya da bir toplama noktasındaki bir problem sadece bu toplama noktasını etkileyecektir. Mevcut yapıya yeni erişim kenar anahtarı eklenmek istendiğinde sadece ilgili toplama noktasında işlem yapılması ile zamandan tasarruf sağlanacaktır.



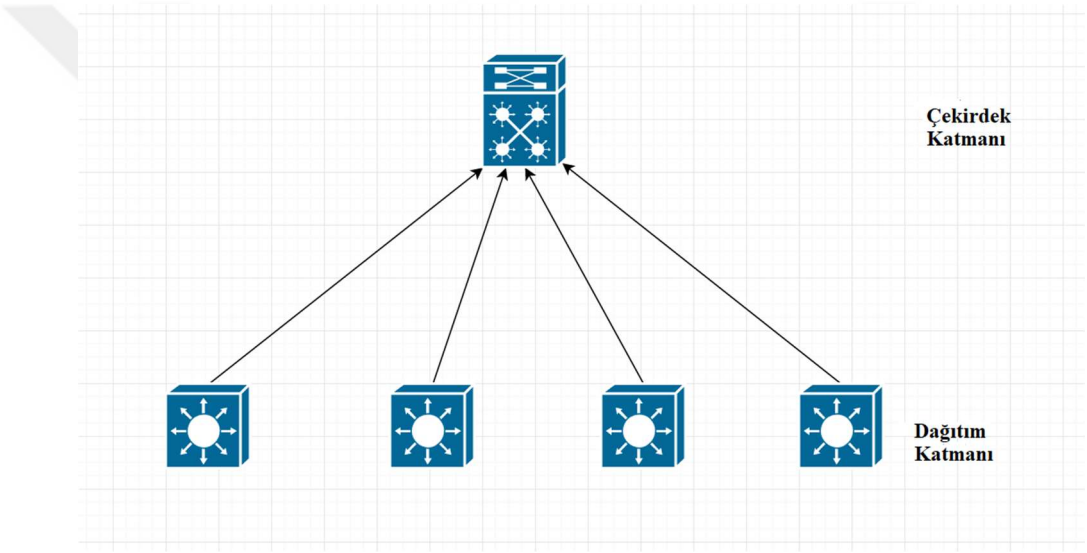
Şekil 3.6 Hiyerarşik katmanlı topoloji tasarım örneği

Hiyerarşik tasarımda kullanıcı aynı SÖA (Sanal Özel Ağ) içerisindeki bir kullanıcı veya cihaza erişecekse trafik aynı erişim katmanı kenar anahtarı üzerinde ilerler ve trafik yukarı katmanlara ulaşmadan iletişim sağlanmış olur. Kullanıcı farklı kenar anahtarda aynı veya farklı SÖA'daki bir kullanıcı veya cihaza ulaşacaksa o zaman trafik dağıtım katmanına kadar çıkar ve ilgili erişim katmanı kenar anahtarına trafik anahtarlanır. Eğer trafik farklı bölgedeki bir cihaza ulaşacaksa o zaman trafik çekirdek katmanı cihazına kadar ilerleyerek, bu cihaz üzerinden ilgili cihaza yönlendirilir.

Çekirdek katmanı, karmaşıklığı ve yönetim zorluğunu azaltarak yönetim kolaylığı ve performans sağlar. Çekirdek katmanındaki cihazlar genellikle merkez sistem odasında konumlandırılır. Bu katman ana bağlantıların yapıldığı, verilen hizmet kaynaklarının konumlandırıldığı ve ağ altyapısının asıl yükünü taşıyan katmandır. Çekirdek katmanının olmadığı bir tasarımda, dağıtım kenar anahtarlarını arasında bağlantı oluşturulmak istendiğinde N tane dağıtım noktası için $N*(N-1)/2$ link gerekirken çekirdek katmanının dâhil edildiği bir tasarımda $N*2$ link gerekir (Cisco, 2020). Bu kural örneklendirilirse, çekirdek katmana yer vermeden tasarlanan bir topolojide oluşan bağlantı sayıları Şekil 3.7'deki gibi, çekirdek katman ile yapılan bir tasarımda oluşan bağlantı sayıları Şekil 3.8'de olduğu gibi şekillenmektedir.



Şekil 3.7 Çekirdek katman olmadan dağıtım katmanı kenar anahtarları arası bağlantı örneği



Şekil 3.8 Çekirdek katmanı ile dağıtım katmanı kenar anahtarları arası bağlantı örneği

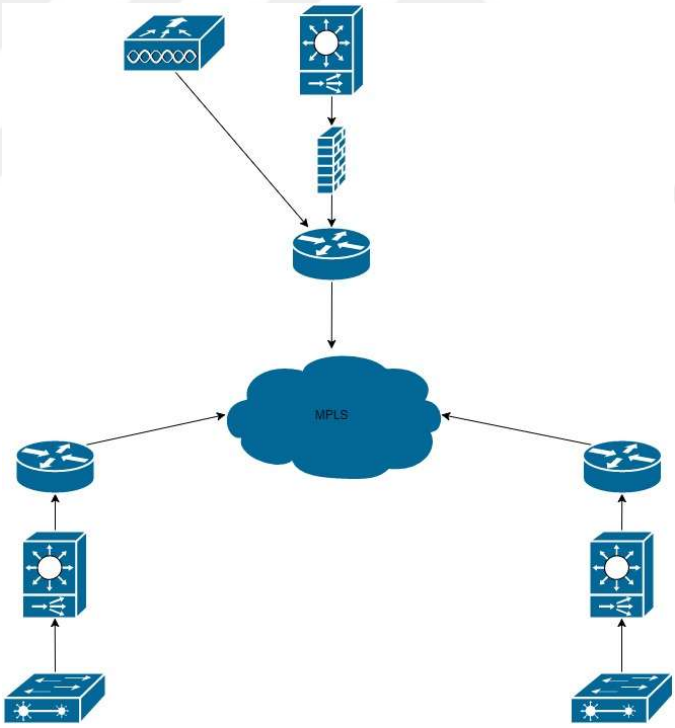
Çekirdek katman, sunulan hizmetlerde 24x365 saat kesintisiz iletişime olanak sağlayan en kritik katmandır. Amaçlanan kesintisizlik süresine uygun cihaz ve teknolojiler seçilmeli, çekirdek katmandaki iletişim mümkün oldukça Katman 3 olarak tasarlanmalıdır.

Dağıtım katmanı, erişim katmanlarının bir noktada toplanmasını sağlayan, çekirdek katmandaki hizmetlere ulaşım noktası veya erişim katmanından gelen trafiği filtrelemek için veya kullanıcı sanal ağlarının yükünü taşıyan katmandır. Bu katmanın görevi sınır tanımlarını sağlamaktır ve paket manipülasyonunun gerçekleştirilebileceği

yerdir. Dağıtım katmanı, altyapıdaki genişleyebilirliği ve esnekliği artırır, karmaşıklığı azaltır. İstenilen dağıtım noktasına erişim katmanı kenar anahtarı eklenerek kolayca genişleme sağlanabilir. Aynı ağ altyapısında farklı bölge tanımlamaları yapılmasına olanak sağlar. Böylece ağ altyapısında oluşabilecek bir hatanın sadece belirli bölgede sınırlı kalması sağlanarak diğer noktalara etki oranı düşer ve erişilebilirlik artar.

Büyük ölçekli ağ altyapılarında, dağıtım katmanı aşağıdaki gibi çeşitli işlevler için kullanılabilir (Schalley, 2013):

- Adres veya alan toplama noktası,
- Departman veya çalışma grubu erişim geçiş noktası,
- SÖA Katman 3 yönlendirme işlevi,
- Güvenlik.



Şekil 3.9 Şubelerde dağıtım katmanı kullanım örneği

Şekil 3.9'daki topolojide de görüldüğü gibi dağıtım katmanı şubelere de konumlandırılarak merkezdeki çekirdek katmanı ile kurumsal bağlantı hizmetleri üzerinden haberleşilir.

Erişim katmanı, kurulan ağ altyapısının ve sağlanan kalitenin kullanıcıya açılan kapısıdır. Çünkü uç noktada yüksek çözünürlüklü film izleyen bir kullanıcı veya yüksek çözünürlükle izlenen bir kamera ya da 1000 kişinin yer aldığı bir konferans salonunda kablosuz hizmet bekleyen kullanıcılar olabilir. Bu yüzden ağ altyapısını uç noktalarda tasarlamadan önce aşağıdaki soruların cevapları aranmalıdır:

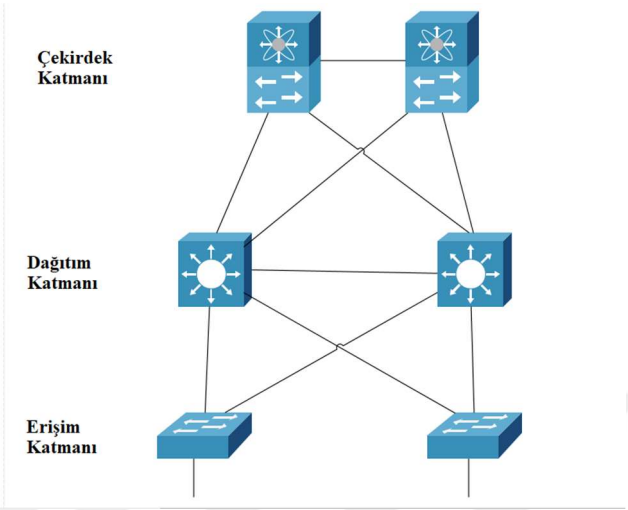
- Hangi hizmetler verilecek?
- Hizmetin önem/kritiklik seviyesi nedir?
- Sunulacak hizmetin ne kadar bant genişliği tüketmesi beklenmektedir?
- Cihazların konumlandırılacağı yerlerde risk seviyesi nedir?

Bu soruların cevabına göre altyapı tasarımı yapılması verimli, müşteri memnuniyeti yüksek ve doğru yatırım yapılmasını sağlar. Çünkü tasarlanmadan yapılan bir yatırım, istenilen verim sağlanamadan atıl olmasına veya gereksiz yüksek proje bedellerinin ödenmesine sebep olacaktır. Örneğin; sadece iki kişinin çalıştığı ve sadece ofis metin uygulamalarının kullanıldığı bir bölgeye üst seviye ağ cihazlarının konumlandırılması yanlış yatırıma yol açacaktır. Verilecek hizmete göre tasarlanmış bir ağ altyapısı bütçenin çok daha verimli kullanılmasını sağlayacaktır.

Genellikle kullanıcı veya cihazlar erişim katmanı üzerinden ağ yapısına dâhil olur. Erişim katmanı üzerinde bilgisayarlar, cep telefonları, el terminalleri, IoT cihazları, IP telefonlar, kameralar, tabletler gibi kablolu veya kablosuz birçok cihaz bağlı olabilir. Erişim katmanını hizmeti ve altyapıyı sağlayan servis sağlayıcı ile kullanıcı arasındaki sınır olarak da nitelendirebiliriz. İnsandan kaynaklanan hataları veya gelebilecek siber saldırıları bu katman üzerinde engellemek veya şiddetini azaltmak için tasarım ve yapılandırma aşamalarında uygun yöntemler kullanılabilir. Örneğin; bir kullanıcının BT ekibini bilgilendirmeden odasında bulunan bir kabloyu Hub ile çoklaması ve Hub üzerinde döngüye sebep olacak bir bağlantı yapması ile tüm ağ altyapınızın veya belirli bölgenin çalışırlığı veya performansı etkilenebilir. Çünkü Hub cihazlarında döngü önleyici algoritmalar yoktur.

Erişim katmanı, son kullanıcıların ağa girmesine izin verilen noktadır. Bu katman da belirli bir kullanıcı kümesinin ihtiyaçlarını optimize etmek için erişim listeleri veya

filtreleri de kullanılabilir. Hiyerarşik tasarımdaki katmanların sade görünümü Şekil 3.10'daki gibidir.



Şekil 3.10 Hiyerarşik ağ altyapısı tasarım katmanları

Ürün seçimi, üründen beklentilere, cihazın yapılandırma yeteneklerine, sahip olduğu teknolojilere ve diğer özelliklerine göre yapılmalıdır. Burada en önemli unsur ağ altyapısını yönetecek teknik ekiptir. Seçilen cihazlar, BT ekibinin bunları yönetebilecek bilgi ve tecrübeye sahip olup olmadığı ve personelin gelişime açıklığı ile beraber değerlendirilmelidir. Çünkü ne kadar iyi tasarlanırsa tasarlansın kurulan yapı yönetilemiyorsa o projeden beklenilen verim alınamaz. Maalesef birçok BT projesinde insan yerine cihaza yatırım yapılmaktadır. Bu anlayışla yönetilen bir ağ altyapısı dış paydaşlara bağımlı olmaktan kurtulamayacaktır. Başarılı bir ağ altyapının ve dış paydaşlara en az bağımlılığın yolu sürekli eğitilen ve teknolojiye uyum sağlaması desteklenen bir BT ekibinden geçmektedir. Bu yüzden bir projenin değerlendirilmesi aşamasında altyapıyı yönetecek BT ekibinin eğitimi de değerlendirilerek plana dâhil edilmelidir.

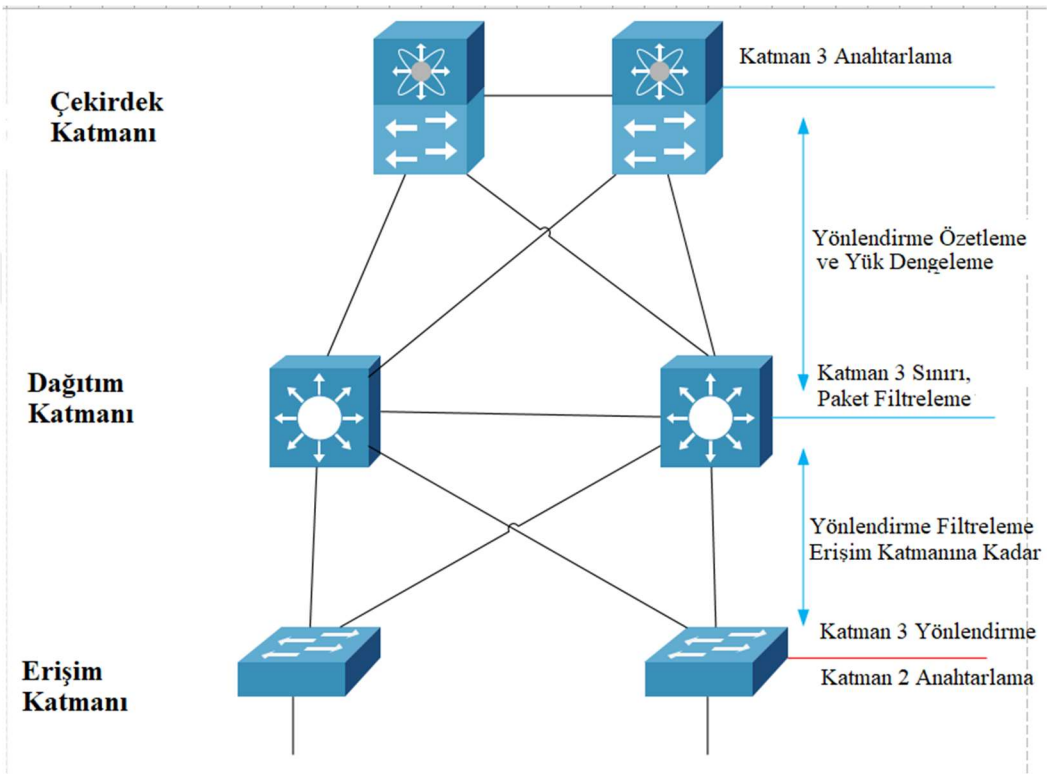
Önceki tasarım yaklaşımlarında katmanlar arası kenar anahtar bağlantıları "Trunk" moda alınır ve kenar anahtarlar arası bağlantılar Katman 2'ye dönüştürülürdü. Bu yaklaşımda Kapsayan Ağaç Protokolünün (STP: Spanning Tree Protocol) çalışma prensipleri gereği bazı bağlantıların kapatılmasına sebep olmaktadır. Bu dezavantajı ortadan kaldırmak için bağlantı noktası kanalı (port-channel) yapılandırması gibi

yöntemler tercih edilebiliyor, fakat port-channel eşit şekilde yük dengeleme yeteneğine sahip olmadığı için bağlantılar istenen verimde kullanılamıyordu. Önceki tasarım yöntemlerinde tüm sanal ağ Katman 3 arayüzlerinin çekirdek katmanda yer alan kenar anahtar üzerinde sonlandırılması tercih edilmekteydi. Bu yöntem ile sanal ağlar arası olan tüm trafik yükü çekirdek katman üzerinde yer alıyordu (KnowledgeClub, 2020b).

Yeni tasarım yaklaşımı olan hiyerarşik tasarım ile büyük ağ altyapıları için dinamik yönlendirme protokollerinin kullanımı, ağ altyapı yönetimi, yük dağılımı ve yedeklik konularında eski tasarım yöntemlerinde olan birçok dezavantajı ortadan kaldırırak altyapı verimini artırmaktadır. Hiyerarşik tasarımda statik yönlendirme yerine dinamik yönlendirme protokolünün kullanılması yönetim açısından önemli kolaylıklar sağlamaktadır. Fakat dinamik protokol yapılandırması varsayılan ayarlarla uygulandığında tasarım hiyerarşik olsa dahi operasyonlar genişledikçe sorunlar ortaya çıkmaya başlayacaktır. Bu yüzden tasarım aşamasında "Hangi ağı kim, nerede kullanacak? Hangi ağa nereden, kim erişecek? Hangi ağ hangi katmanda sonlandırılacak?" gibi sorulara aranan cevaplara uyan bir taslak hazırlanıp bu taslağa göre tüm katmanlarda tercih edilen yönlendirme protokolü için kurallar ve erişim kontrol listeleri oluşturulmalıdır. Örneğin, yazıcılar ağına erişim sadece dağıtım katmanından sağlanacaksa bu ağın çekirdek katmanda sonlandırılmasına gerek yoktur. Dağıtım katmanında gerekli ağlar çekirdek katmana doğru elle dinamik yönlendirme özetlemeleri yapılmalıdır veya dağıtım listeleri ile dağıtım katmanında yönlendirme güncellemeleri filtrelenmelidir. Bu sayede cihazlar gerekli olan yönlendirme güncellemelerini alacak, cihaz kaynakları gereksiz olarak tüketilmeyecek ve özetle cihazların gereksiz işlem yapmalarının önüne geçilecektir. Bu şekilde tasarlanmış bir ağ altyapısında dinamik yönlendirmenin kontrolü ağ mühendisinin elinde olacak ve yönetimsel kolaylıklar sağlayacaktır.

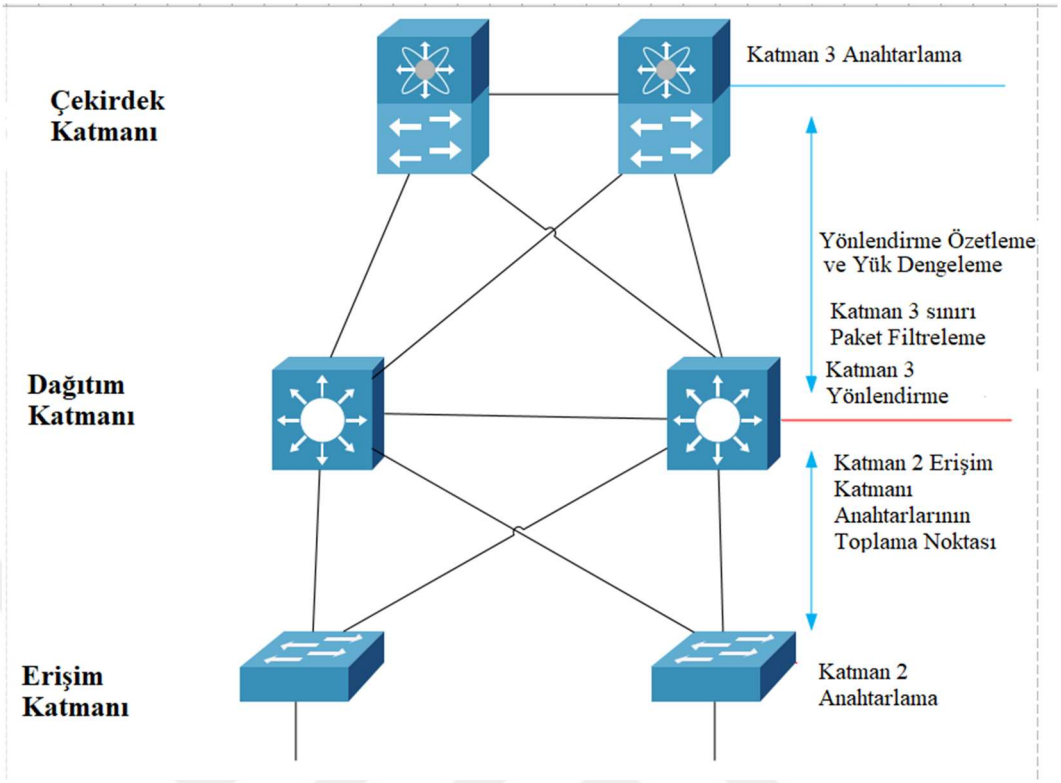
Yeni tasarım yaklaşımlarında ağ altyapıları mümkün oldukça Şekil 3.11'de örneklendirildiği gibi Katman 3 olarak tasarlanmaya çalışılmaktadır. Bu tasarım yönteminin benimsenmesindeki en önemli sebep STP protokolünün sebep olduğu dezavantajları ortadan kaldırması ve bu protokole olan bağımlılığın azaltılmasıdır

(Jordan ve Bruno, 2016). Çünkü Katman 2'de yapılabilecek bir bağlantı hatası ağ içinde döngülere sebep olarak, ağ altyapısında ciddi problemlere neden olacaktır. Örneğin, seri olarak ardı ardına takılmış Hub cihazı ile ağ altyapısında oluşabilecek bir döngünün etkileri tüm ağ altyapısında hissedilecektir.



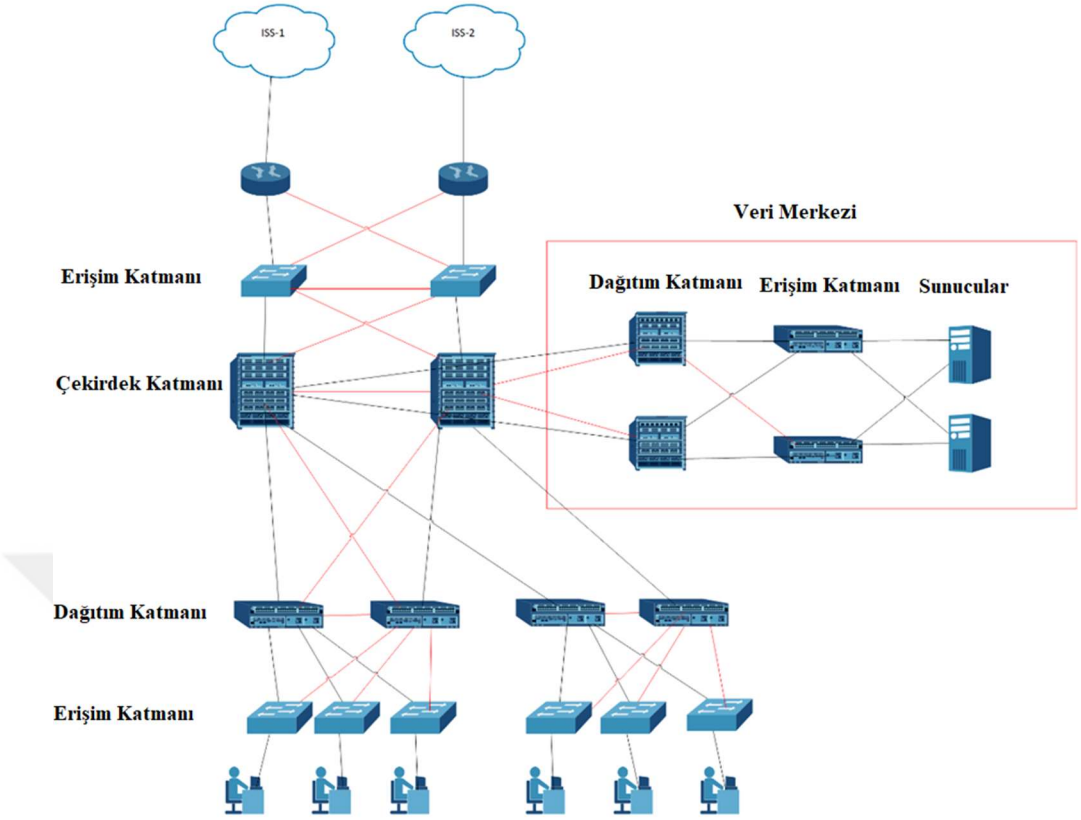
Şekil 3.11 Tüm katmanlar arası iletişimin Katman 3 olarak tasarlandığı topoloji örneği

Günümüzde tercih edilen yöntemlerden biri de tüm katmanların Katman 3 olarak tasarlanması ve Katman 2-3 sınır noktası olarak erişim katmanının belirlenmesidir. Ağ yayın trafiğini ve yerel trafiği kontrol altına almak için en etkili tasarım yöntemlerinden biridir. Fakat yönetimsel zorlukları vardır. Ağ mühendisleri üzerindeki iş yükünü artırmaktadır. Erişim katmanının Katman 3 olarak tasarlanması proje maliyetlerini de artıracaktır. Çünkü erişim katmanındaki cihazların Katman 3 olarak konumlandırılması, uygun yönlendirme yeteneklerine ve lisanslarına sahip olmaları gerekir.



Şekil 3.12 Erişim katmanın tamamen Katman 2 olarak tasarlandığı topoloji örneği

En çok tercih edilen bir diğer yöntem ise Şekil 3.12'de gösterildiği gibi Katman 2-3 sınır noktası olarak dağıtım katmanının belirlendiği tasarımdır. Bu tasarım yönteminde hem çekirdek katmanındaki yük hem de erişim katmanındaki değişikliklerden dolayı oluşan yapılandırma yükü azalmaktadır.



Şekil 3.13 Yedekli hiyerarşik topoloji örneği[1]

Katman temelli ve sürekliliği göz önünde bulundurarak yedekli tasarlanmış bir topoloji Şekil 3.13'teki gibi bir görünüme sahip olacaktır. Bu topolojide tam yedeklik senaryoları düşünülerek çekirdek, dağıtım ve internet erişim katmanlarındaki her bir cihaz yedekli olacak şekilde tasarlanmıştır. Herhangi bir olumsuz bir durum yaşanıp bir bağlantı koptuğunda yedek bağlantı üzerinden devam edilebilecek şekilde yedeklenmiştir. Çekirdek ve dağıtım katmanlarının Katman 3 olarak yapılandırılacağı varsayılarak tasarlanmıştır. Yerel ağ altyapısında ve veri merkezinde hiyerarşik tasarım uygulanarak çekirdek katman üzerindeki gereksiz trafik yükünün azaltılması amaçlanmıştır. Bu yapıda sanal ağların ağ geçitleri hem yerel ağlarda hem de veri merkezi bölgelerinde dağıtım katmanlarında sonlandırılarak özellikle Kuzey-Güney trafiğinin kontrol altına alınabilmesi amaçlanmıştır. Bu şekilde tasarlanmış bir

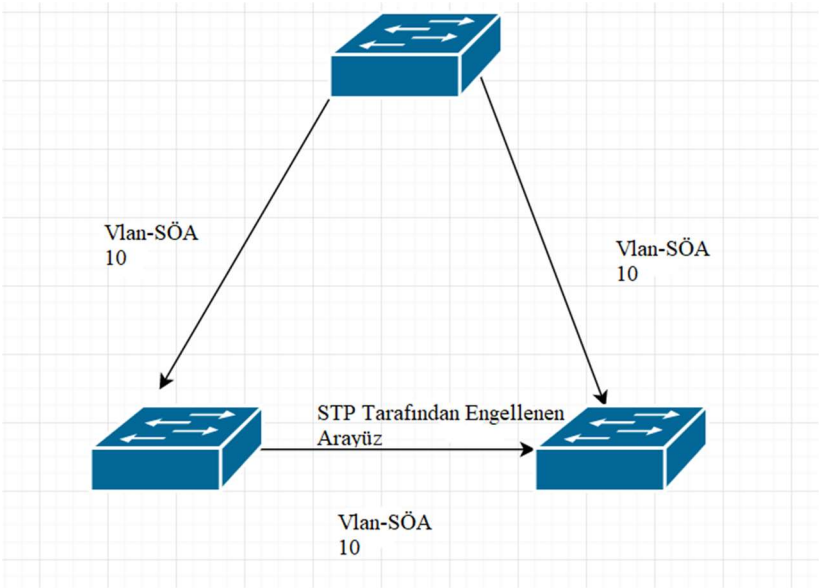
[1] Kırmızı çizgiler yedeklik bağlantılarını temsil etmektedir.

altyapıda trafik gerekmedikçe çekirdek katmanına ulaşmayacak ve dağıtım katmanı üzerinde kalacaktır. Ağ yayın trafiklerinin çekirdek katmanına erişmesi engellenerek ağ yapısının en önemli noktası olan omurga kenar anahtarları yani çekirdek katmanındaki cihazların, erişim katmanında oluşabilecek olumsuzluklardan etkilenme olasılığı minimize edilmiş olacaktır.

3.3.1 Kenar Anahtar Tasarımı

Kenar anahtarlarda oluşabilecek bir hata veya döngü tüm ağ altyapısını etkileyebilir. Bu hata ve döngü gibi problemleri dağıtım katmanında bazı tasarım yöntemleri ile önlenmesi mümkündür. En çok kullanılan tasarımlardan bir tanesi "Loop-Based" tasarımıdır. Bu tasarım modelinde döngüleri engellemek için STP protokolü kullanılır. STP protokolü döngüleri engellemesine rağmen aşağıdaki gibi dezavantajlara sebep olabilir:

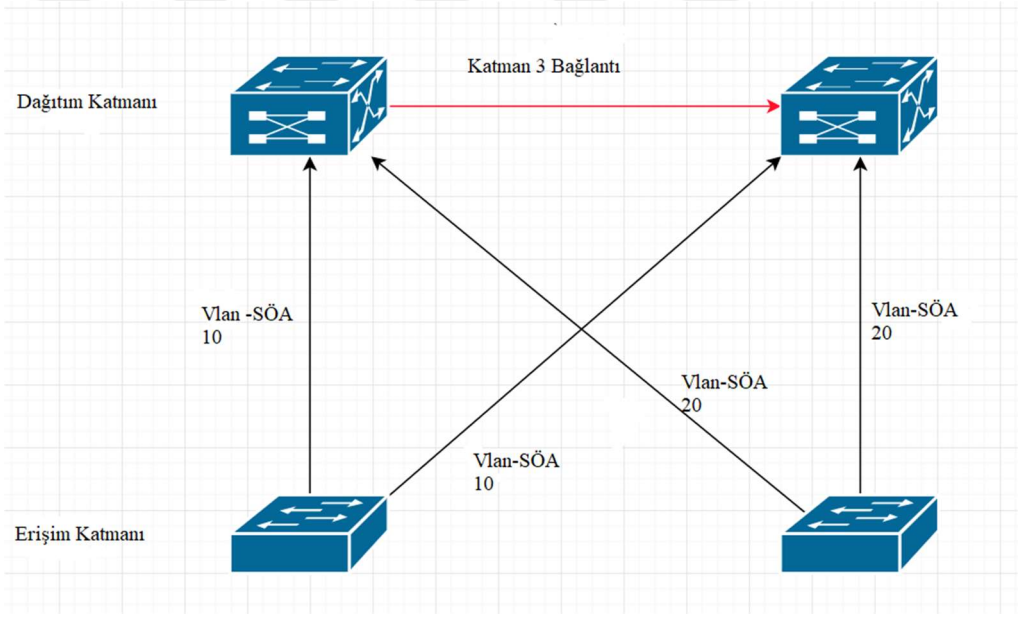
- STP'nin portları engellemesinden dolayı bant genişliğinin azalması,
- Ağ yayılım sürelerinin yavaş olması,
- Konfigürasyon hata olasılığının fazla olması.



Şekil 3.14 STP protokolünün port engelleme örneği

Kenar anahtarlar arasında oluşturulan ara bağlantıların genellikle “Trunk” olarak yapılandırıldığı görülür. Böyle yapılandırılmış bir altyapıda STP protokolü devreye girerek bazı bağlantı noktaları (portlar) engellenir ve ağ altyapısında döngü oluşması önlenir (Cisco, 2017a). Şekil 3.14 incelendiğinde SÖA 10’da olan 3 kenar anahtarın gönderdiği trafik başka ara yüzden gelecek şekilde bağlandığında, STP protokolü devreye girerek bir portu engelleyerek döngüyü önlediği görülmektedir. STP aktif olmadığı durumda, Şekil 3.14’teki gibi birbirine bağlanan cihazlar arasında döngü oluşarak ağ çalışırlığı olumsuz yönde etkilenecektir.

STP protokolünün oluşturduğu sorunları çözümlemek için Şekil 3.15’te gösterilen ve “loop-free” tasarım olarak adlandırılan model tercih edilmektedir.



Şekil 3.15 Loop-free tasarım topoloji örneği

Dağıtım katmanı veya çekirdek katmanı kenar anahtarları arasında Katman 3 bağlantı oluşturur. Böylelikle aradaki Katman 2 bağlantı engellenir ve gereksiz ağ yayın trafiği diğer kenar anahtara taşınmaz. Bu yapıda STP devreye girmediği için yedek bağlantılar engellenmez ve kullanılabilir duruma gelir. İki kenar anahtar arasındaki trafik yönlendirmesi Katman 3 yönlendirme protokolleri aracılığı ile

sağlanır. Bu tasarım modelinde de STP protokolünün her ihtimale karşı açık tutulması önerilmektedir.

3.3.2 Kablolama Tasarımı

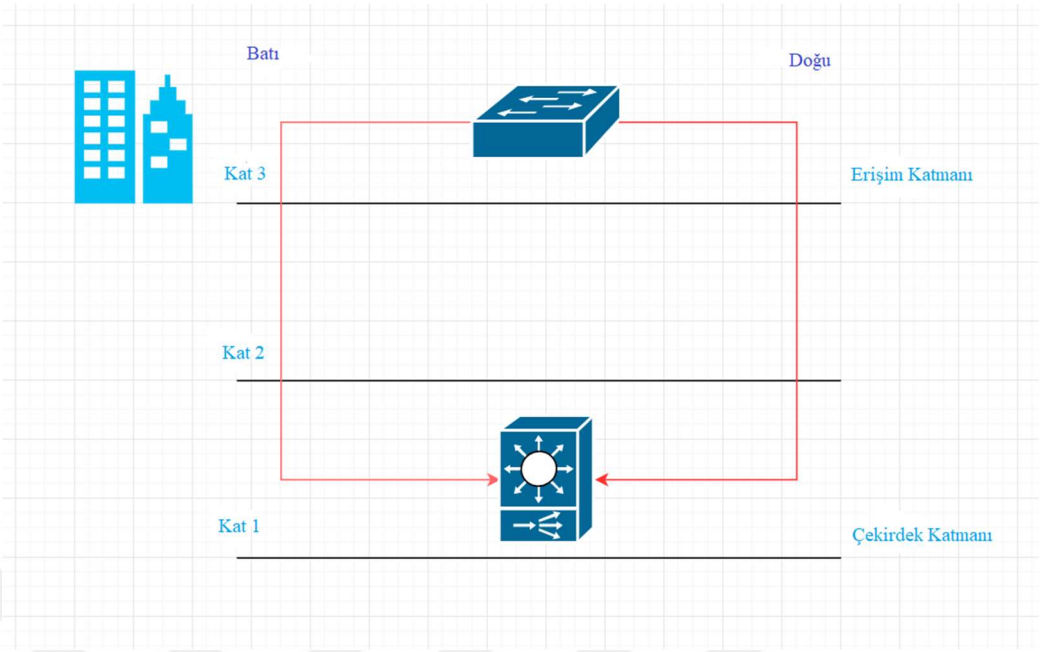
Kablolama tasarımı ağ altyapısının yedekliğine, esnekliğine, genişleyebilirliğine ve sürdürülebirliğine doğrudan etki eden bir faktördür. Bir ağ altyapısının uygulamasına geçilmeden önce iyi tasarlanmış bir kablolama altyapı tasarımı gereklidir. Planlanan ağ altyapısının genişliğine göre hangi tür kablonun seçileceği, hangi modülün kullanılacağı, yedekliğin nasıl sağlanacağı ve kablo altyapısının fiziksel güvenliği gibi konular üzerinde önceden düşünülüp planlama gerekir. Örneğin, 80 km uzunluğunda bir fiber kablonun ucuna maksimum 2 km aralığında çalışabilecek bir modül kullanılması tasarlanırsa, ağ altyapısı kurulumu sırasında problemlerle karşılaşılması kaçınılmaz olacak ve maliyet artışına sebep olacaktır.

Kablolama altyapısı tasarımı yaparken en uç noktadan merkeze doğru aşağıdaki plan adımları dikkate alınmalı veya uygulamaya konulmalıdır:

- Kablolamanın dış mekânda mı, yer altında mı, bina içinde mi veya veri merkezinde mi olacağı göz önünde bulundurulup, tasarım için örnek alınan kablolama standardı temel alınmalıdır.
- Uç noktada hangi cihazların yer alacağı, merkezden hangi uygulamalarla veya servislerle bağlantı kurulacağı ve cihaz başına ortalama ne kadar trafik oluşacağı tasarım öncesinde belirlenmelidir.
- Kullanılacak kablolar, fiziksel ortam şartları göz önünde bulundurularak belirlenmelidir. Örneğin, tek bilgisayar çalışacak ise kategori 6 kablo kullanılabilecekken, dış ortamda bulunan bir sensör için güneş, yağmur ve darbelere dayanıklı olabilecek kablo tipleri seçilmelidir.
- Sahadaki erişim kenar anahtarları için toplama noktaları belirlenmelidir. Toplama noktalarının fiziksel güvenliği, soğutma gereksinimleri ve enerji ihtiyaçları göz önünde bulundurulmalıdır.

- Kenar anahtar seçiminde özellikle cihazın konumlandırılacağı yerin fiziksel şartları göz önünde bulundurularak bina içi kenar anahtarın mı yoksa endüstriyel bir kenar anahtarın kullanılması gerektiği belirlenmelidir.
- Uç noktalar ile toplama noktaları arası, toplama noktaları ile merkez sistem odası arası en fazla ne kadar mesafe olacağı hesaplanarak fiber kablo ve modüller belirlenmelidir. Çünkü uzaklığa göre fiber modül tipleri, fiber kablo tipi ve maliyetleri değişmekte ve mesafe artıkça modül tipleri ve maliyetleri de değişmektedir. Örneğin, uç nokta ile toplama noktaları arası 80 km mesafe varsa, bu mesafe aralığında çalışabilecek bir modül seçilmelidir (KnowledgeClub, 2020b).
- Fiber kablo tipi belirlenirken genellikle yakın mesafede çok kanallı, uzak mesafelerde ise tek kanallı fiber kablo tercih edilmelidir.
- Bir fiber kablonun arada tekrarlayıcı olmadan gidebileceği ortalama mesafe 80-100 km arasındadır. Bu mesafelerin üzerindeki durumlarda araya tekrarlayıcı veya kenar anahtar koyulması gerekir (KnowledgeClub, 2020b).

Hiyerarşik tasarımdaki üç ana katman olan erişim, dağıtım ve çekirdek katmanları arasındaki bağlantı yedekliği dikkatli tasarlanmalıdır. Örneğin, erişim katmanından dağıtım katmanına iki kablo hattı gidecekse bu iki hat uç noktadan çıkışından itibaren birbirinden tamamen farklı güzergâhlardan gitmelidir. Bu sayede hattın birinin bulunduğu bölge zarar görürse diğer hat üzerinden iletişim devam edebilecektir, altyapının güvenilirliği, erişilebilirliği ve sürdürülebilirliği artacaktır. Fakat bu yöntem yerine aynı fiber kablonun aynı güzergâhtan geçirilmesi durumunda, fiber kablonun zarar görmesi ile iki fiber hattında da problem yaşanacak ve yedeklik ortadan kalkacaktır. Eğer bina içi kablolama yapılıyorsa iki katman arasındaki yedekli gelen kablo Şekil 3.16'daki gibi tamamen farklı güzergâhlardan geçmelidir.



Şekil 3.16 Örnek bina içi yedek hat güzergâhı

Fiber uç planlaması ilgili bölgede beklenen genişleme ve katmanlar arası oluşması muhtemel trafiği bir üst katmana aktarmak için gereken fiber uç gereksinimlerini de dikkate alarak yapılmalıdır. Burada uygun cihaz, uygun kablo kararı aşamasında taşınacak veri ve beklenen trafik gecikmesinin tahmini kritik rol oynamaktadır. Kablolama ve ağ altyapı ürünleri seçiminden önce mutlaka hangi verilerin taşınacağı belirlenerek, taşınacak olan veri miktarı tahminleri oluşturulmalı ve uçtan merkeze doğru ne kadar bant genişliği ihtiyacı olacağı ve seçilen kablo tipinde tahmini gecikme değerleri hesaplanmalıdır. Örneğin, ideal şartlarda dümdüz şekilde serilmiş 100 km'lik trafik olmayan bir fiber hatta ortalama 500 ms gecikme olması beklenir (KnowledgeClub, 2020b).

Özellikle akıllı şehir ağ altyapıları gibi uzak noktalara yayılan büyük altyapıların tasarımında her bir seçenek önceden değerlendirilmeli ve hesaplanarak karar verilmelidir. Cihazların üzerinden geçecek trafiğin boyutu, kullanılacak protokoller ve katmanlar arası oluşacak trafik miktarı göz önünde bulundurularak cihaz seçimi ve kablolama tasarımı yapılmalıdır. Trafik akışının yoğun olduğu katmanlı tasarım yapısında katmanlar arası cihaz bağlantılarında aşağıdaki gibi bir bağlantı modeli tercihe göre uygulanabilir (KnowledgeClub, 2020b):

- Erişim katmanı ile Dağıtım katmanı arası bağlantı: 1 Gigabit (Gb)'lik bağlantı uçları için her 48 bağlantı ucu için bir 1 Gb'lik uçtan bir üst katmana bağlantı kablosu tasarlanmalıdır.
- Dağıtım katmanı ile Çekirdek katmanı arası bağlantı: 10 Gb'lik bağlantı uçları için 4'e 1 oranında çekirdek katmanına bağlantı yapılmalıdır. Örneğin; 48 bağlantı ucu için 12 adet bağlantı oluşturulmalıdır. Bu yöntem daha çok veri merkezi trafiğinde darboğazların oluşmaması için tercih edilir.
- Çekirdek katmanı bağlantıları: Çekirdek katmanı üzerinde yapılan bağlantılar 1'e 1 olarak yapılmalıdır.

Ağ altyapılarında sürekliliği artırmak için mümkün oldukça "Full-Meshed" veya "Hub-and-Spoke" topolojileri tercih edilmelidir. Fakat maliyet ve kurulum kolaylığından dolayı projelerde genellikle "Ring" (halka) topolojileri seçilmektedir. Ring topolojilerinin maliyet ve kurulum yönünden sağladığı avantajlara rağmen süreklilikle ilgili dezavantajlara sebep olmaktadır. Ring topolojilerinde cihaz seçimi esnasında cihazın donanım ve yazılım olarak en fazla kaç ring noktasını desteklediği ve üzerinde hangi uygulamaların çalışacağı göz önünde bulundurularak tercih edilmelidir. Örneğin, kamera trafiğinin taşınacağı ağlarda zorunlu olmadıkça ring topoloji tercih edilmemelidir. Çünkü ring topolojilerinde çalışan kamera görüntülerinde donma gibi sorunlarla karşılaşılma olasılığı vardır. Ring topolojileri yerine full-meshed veya hub-and-spoke gibi topolojilerin kullanılması bize dar alanda avantajlar sağladığı gibi geniş alanlara yaygınlaştığında aşağıdaki gibi bazı dezavantajlara sebep olabilmektedir (KnowledgeClub, 2020b):

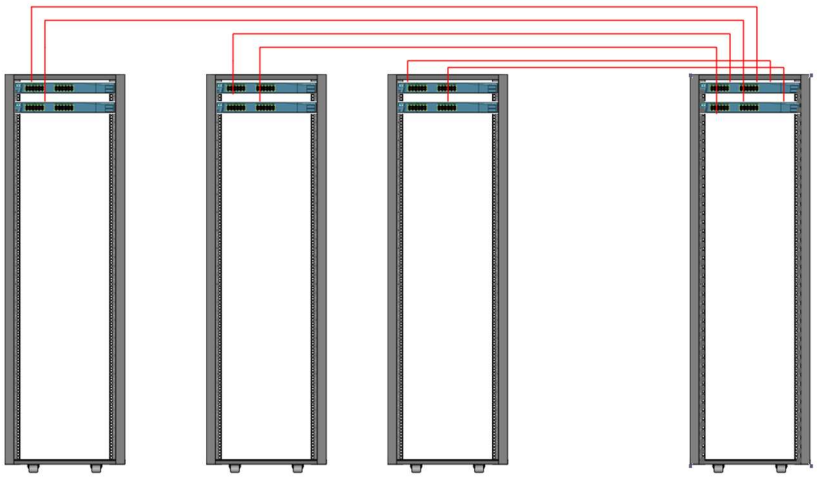
- Maliyet artışı,
- Altyapı genişlemesinde daha fazla kazı ve kablolama ihtiyacı,
- Altyapı genişlemesinde daha fazla iş yükü.

Diğer taraftan bir sistem odası ve kabin tasarımları yaparken genel olarak aşağıdaki hususlar dikkate alınmalıdır:

- Bina içine konumlandırılmış kenar anahtarların bulunduğu kattan bir alt ve bir üst kata aynı kabinden bağlantı sağlanabilir, fakat bir kabinden

birden fazla bağlantı ile üst kata veya alt kata bağlantı sağlanarak diğer katlara bağlantı sağlanması önerilmemektedir. Eğer zorunluluk var ise ve bağlantı verilmesi gerekiyorsa çözüm olarak kat BT odası konumlandırılması önerilmektedir.

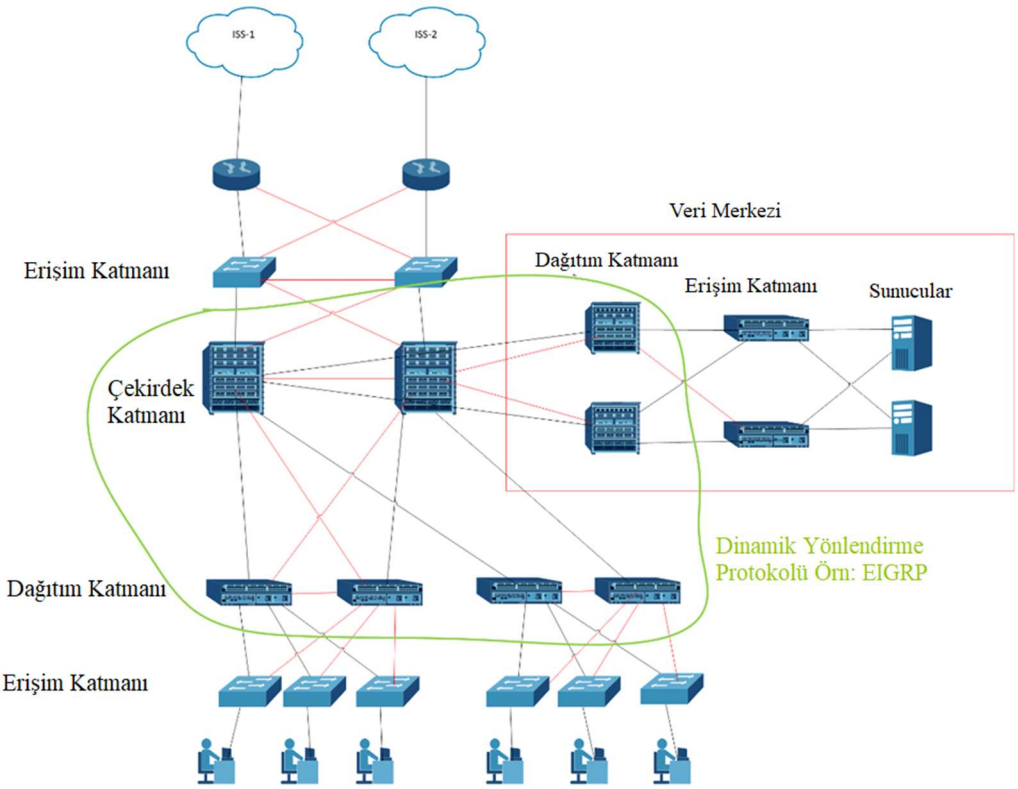
- Kabinlere taşıyabileceği kadar kablo sonlandırılmalıdır.
- Kenar anahtarların çalışacağı sıcaklık aralıkları ve soğutma gereksinimleri gibi çevresel şartlar dikkate alınmalıdır. Örneğin; Ortalama bir dış ortam kenar anahtarın çalışma sıcaklığı 35-60 derece aralığındadır.
- Kat kabinlerinden verilen bakır kablo bağlantıları sayıları önceden hesaplanmalı ve mesafeleri 90 metreyi geçmemelidir.
- Dağıtım katmanı kenar anahtarları ile erişim katmanı anahtarları aynı kabinde sonlandırılmamalıdır. Dağıtım katmanı ayrı bir kabin içinde ve ayrı enerji ile beslenmelidir.
- Bina içinde veya sahada tasarım katmanları arası planlan fiber kablolama yapısı tasarlanırken ana hat ve yedek hat birbirinden ayrı güzergâhlardan planlanmalıdır.
- Çekirdek Katmandaki cihazların yedekleri farklı bir kabinde konumlandırılmalı ve farklı kesintisiz güç kaynağı (UPS: Uninterruptible Power Supply) ve elektrik panolarından enerji alınmalıdır.
- Veri merkezleri için yangın planlaması yapılmalı ve birden fazla veri merkezi olması durumunda her bir veri merkezinin farklı yangın bölgelerinde bulunmasına dikkat edilmelidir.
- Her kabine özel kenar anahtar konumlandırılmalı, bir kabinden diğer kabindeki bir cihaza bağlantı oluşturulmamalıdır. Kabinlerdeki kenar anahtarlar da başka bir kabinde farklı ve ayrılmış kenar anahtar üzerinde sonlandırılmalı ve yedeklikleri oluşturulmalıdır. Bu bağlantı kuralları Şekil 3.17'de örneklendirilmiştir. Şekil 3.17'deki gibi hazırlanan bir kabin ve bağlantı tasarımı sistem odası sürekliliğine ve yedekliğine önemli katkı sağlayacaktır.



Şekil 3.17 Kabin kenar anahtar tasarımı

3.3.3 Ağ Altyapılarında Kullanılabilecek Dinamik Yönlendirme Protokolleri

Uçtan uca veri paketinin izleyeceği yolun belirlenmesi ve doğru yere iletilmesine yönlendirme denir (Boyan, 2017). Göndericiden alıcıya doğru yola çıkmış bir paket, hedefe ulaşırken IP adreslerini kullanarak yolunu bulur. Yönlendirme işlemleri dinamik veya statik olarak gerçekleştirilebilir. Yönlendirme işlemleri OSI Katman 3'de gerçekleştirilir. Dinamik yönlendirme protokolleri yapılan tanımlamalara göre cihaz üzerinden erişilebilecek veya cihaza doğrudan bağlı ağ adreslerini diğer cihazlara seçilen protokolün güncelleme paketleri ile bildirerek yönlendirme tablolarının güncellenmesini sağlar. Dinamik yönlendirme protokolleri gelen paketleri seçilen yönlendirme protokolünün en iyi yol seçimi algoritması ile uygun yolu belirleyip yönlendirme işlemlerini gerçekleştirirler.



Şekil 3.18 Hiyerarşik topoloji tasarımında dinamik yönlendirme ile tasarlanmış örnek topoloji

Şekil 3.18'deki tasarımdaki gibi bir ağ altyapısında dağıtım ve çekirdek katmanı arası trafik statik yönlendirme ile yönetilebilir. Fakat EIGRP, OSPF gibi dinamik yönlendirme protokollerinin kullanılması ile katmanlar arası elle işlem gereksinimlerinde azalma olacaktır. Hatta dinamik yönlendirmede "port-channel" kullanmak yerine EIGRP gibi dinamik yönlendirme protokolleri üzerinde yük dengeleme yapılarak katmanlar arası bağlantılar aktif kullanılabilir.

3.3.3.1 EIGRP Protokolü

EIGRP protokolü, Cisco Systems firması tarafından geliştirilmiş bir yönlendirme protokolüdür. Önceki yıllarda sadece Cisco firmasına ait cihazlarda çalışmasına rağmen protokolün kullanımı diğer firmalara da açılmıştır. Hızlı ve kolay yönetim için ideal bir yönlendirme protokolüdür. EIGRP, uzaklık vektörü protokolü ve hat-durum protokollerinin özelliklerini alan karma bir protokol olarak tasarlanmıştır. EIGRP, komşulara giden en iyi yolu ve en iyi ikinci yolu önceden hesaplayarak topoloji

tablosunda tutar. Bu tablodaki veriler yardımıyla da bir bağlantı koptuğunda, ortalama 10 saniye gibi kısa bir süre içerisinde yedek bağlantıyı aktif eder.

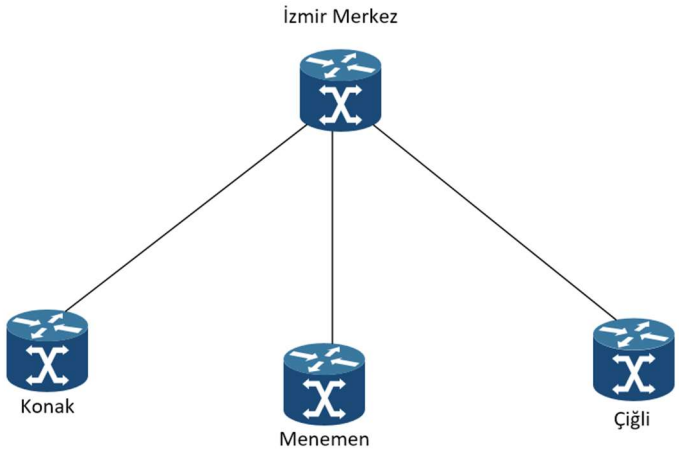
EIGRP'nin Özellikleri (İTÜ BİDB, 2013a):

- Yönetimsel uzunluk değeri 90'dır.
- VLSM (Variable Lenght SubnetMask - Değişken Uzunluklu Alt Ağ Maskeleri) desteği vardır.
- IP, IPX, AppleTalk protokollerini desteklemektedir.
- Otomatik ve elle özetleme özelliğine sahiptir.
- EIGRP protokolü sınıfsız çalışır.
- DUAL (Diffusing Update Algorithm - Difüzyon Güncelleme Algoritması) algoritmasını kullanır.
- Cisco tarafında geliştirilmiş RTP (Reliable Transport Protocol) protokolü kullanılır.
- Hello paketleri kullanılır.

Ağ altyapı tasarımlarında EIGRP protokolünün kullanımında mümkün olduğu kadar doğrusal topoloji tasarımlarından kaçınılmalıdır. Çünkü aşağıdaki şekildeki gibi doğrusal ağlarda "Query" paketlerine, "Reply" paketlerinin geç dönmesi durumunda EIGRP protokolü üzerinde SIA (Stuck In Active) problemine neden olacaktır. Eğer Şekil 3.19'daki gibi doğrusal tasarım yapılması gerekiyorsa, problem olasılığını azaltmak için uç noktaların stub olarak tasarlanması problem olasılığını ve etkilerini azaltmada yardımcı olacaktır. Çünkü stub, tip 4 ve 5 hat durumu ilanı (Link-State Advertisement: LSA) denilen yayın paketlerinin ilgili alana girişini engeller (Knight, 2015).

Şekil 3.19 Doğrusal ağ altyapı tasarım örneği

EIGRP tabanlı ağlarda önerilen tasarım Şekil 3.20'de örneklendirilen yıldız topoloji tasarımıdır. Merkez-Şube yapılandırmasında özellikle yıldız topoloji tercih edilmektedir. Şubeler "stub" olarak yapılandırılmalıdır.

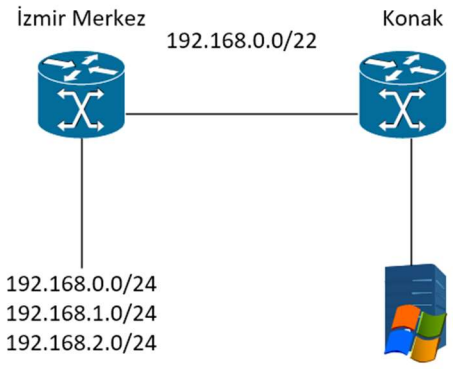


Şekil 3.20 Merkez-Şube yıldız topoloji örneği

EIGRP protokolü kullanılarak merkez-şube tasarımlarında 800 komşu seviyesine kadar gerçek ortamda çalışılabilmektedir. Fakat bu tip büyük ağ tasarımlarında yönlendirme güncellemelerinin ağ mühendisinin kontrolünde olması ve uç noktaların "stub" olarak yapılandırılması önemlidir. Çünkü "stub" olarak yapılandırılmayan bir ağ altyapısında herhangi bir değişikliğin cihazlar tarafından algılanma süresi komşu sayısına paralel olarak artacak ve ağ sistemi hantal bir yapıya dönüşecektir.

EIGRP ile tasarlanmış büyük bir ağ altyapısında cihazlar arası EIGRP güncelleme paketleri kontrol altında tutulamaz ise, bu durum yönlendirme tablolarının büyümesine sebep olacak ve ağ altyapısının performansını olumsuz etkileyecektir. Bu yüzden büyük ağ altyapılarında gereksiz yönlendirme bilgilerinin ağın ihtiyaç duyulmayan noktalarına gereksiz yere ulaştırılarak bant genişliği ve cihaz kaynaklarının gereksiz yere tüketilmesini engellemek için altyapının belirli noktalarında elle özetleme işlemleri yapılmalıdır. Ağ altyapıları üzerinde etkin ve kullanışlı özetlemeler yapabilmek için IP ve ağ maskesi planlaması aşamasında özetleme faktörü düşünülerek tasarlanmalıdır. Tasarım aşamasında özetleme faktörü düşünülmeden oluşturulan bir ağ altyapısında yapılacak özetleme ve yönlendirme manipülasyonları

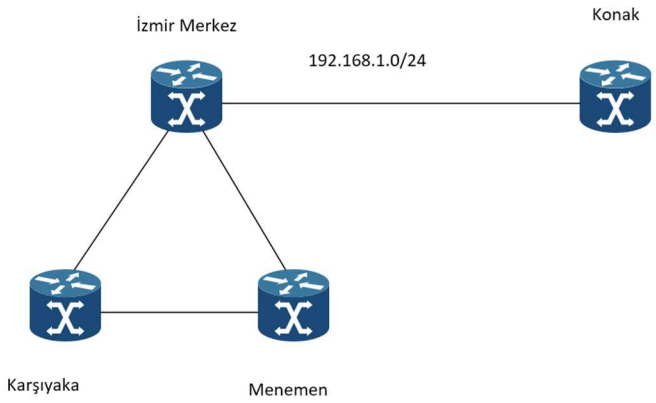
tasarım hatasından dolayı kısıtlı olacaktır. Ağ cihazlarında otomatik özetleme seçeneği mevcuttur. Fakat büyük ağ altyapılarında bu faydadan çok zarara dönüşebilir. Bu yüzden büyük ağ altyapılarında bu seçeneğin kapatılarak elle özetleme yöntemi ile yönlendirme bilgilerinin ağ mühendisinin kontrolü altında tutulması önerilmektedir.



Şekil 3.21 Örnek özetleme sorunu

Şekil 3.21'deki topoloji örneği incelendiğinde, İzmir merkez yönlendiricisinin arkasında yer alan 192.168.0.0/24, 192.168.1.0/24, 192.168.2.0/24 ağları otomatik özetleme ile 192.168.0.0/22 olarak ayarlanır. Fakat 192.168.1.0/24 ağında bir problem olduğunda Konak yönlendiricisi bunu fark edemeyecek ve kullanıcıyı yine İzmir merkez yönlendiricisine yönlendirmeye devam edecektir. Bu tip bir problemi çözmek için tasarım aşamasında doğru IP planlamasıyla elle özetleme kullanılmalıdır.

EIGRP protokolünde tasarım hatalarından dolayı karşılaşılabilecek bir diğer problem de "query-reply" paketlerinin hatalı tasarım sonucu ağ üzerinde döngülere sebep olmasıdır. Örneğin, Şekil 3.22'deki gibi kurulmuş bir ağ altyapısında "Hello, Update, Acknowledge" paketleri ağ ilk kurulduğunda yönlendiriciler arasında gelip giderek ağ yapısı aktif olur.



Şekil 3.22 Hatalı EIGRP tasarım örneği

192.168.1.0/24 ağını Karşıyaka yönlendiricisi hem İzmir merkez yönlendiricisinden hem de Menemen yönlendiricisinden öğrenir. Menemen ise hem İzmir Merkez yönlendiricisinden hem de Karşıyaka yönlendiricisinden öğrenir. Bütün bağlantılar ayakta iken problem oluşmazken İzmir Merkez ve Konak yönlendiricileri arasındaki bağlantıda problem oluşursa İzmir merkez yönlendiricisi Karşıyaka ve Menemen yönlendiricilerine 192.168.1.0/24 ağını bilip bilmediklerini öğrenmek için "Query" paketi atar. Karşıyaka bilgi istenen ağı daha önce Menemen yönlendiricisinden öğrendiği için İzmir merkez yönlendiricisine bu ağa Menemen yönlendiricisi üzerinden ulaşabileceği bilgisini döner. Menemen yönlendiricisi de İzmir merkez yönlendiricisine bu ağa Karşıyaka yönlendiricisi üzerinden ulaşabileceği bilgisini "Reply" paketi ile döner. Bu ise ağ altyapısında döngüye sebep olur. Bu problem EIGRP protokolünde SIA problemi olarak bilinir. Bu tip tasarım hatalarını engellemek için uç noktalarda "EIGRP stub" yapılandırılması uygulanmalıdır. Yukarıdaki örnekte Karşıyaka ve Menemen yönlendiricilerine "stub" yapılandırılması ile benzer problemlerin oluşması engellenecektir.

3.3.3.2 OSPF Protokolü

IETF (Internet Engineering Task Force) tarafından geliştirilmiş bir protokoldür. Birçok ağ üreticisi tarafından desteklenir. "Önce En Kısa Açık Yol" (OSPF: Open Shortest Path First), hat-durum protokolü olarak tasarlanmıştır. Ağ değişikliklerinde

tetiklenmiş güncelleme göndererek ağdaki komşuları bilgilendirir. OSPF protokolünde, yönlendiriciler ağdaki iki nokta arasındaki tüm yol seçenekleri arasında en kısa yol algoritması ile hedefe giden en iyi yolun hangisi olduğu karar verilir. (İTÜ BİDB, 2013c). OSPF, hedefe giden tüm yol bilgisini hızlı bir şekilde öğrenme özelliğiyle büyük ve karmaşık ağlarda daha iyi çalışabilir ve oldukça güvenilirdir. Yönlendiriciler komşularına her 10 saniyede bir "Hello" paketi atarak ağda bir değişiklik olup olmadığını kontrol eder. OSPF protokolü uzaklık vektörü protokolleri gibi atlama noktası sınırlandırma metriği kullanmaz. Metrik olarak maliyet hesaplama (cost calculation) yöntemindeki bant genişliği ile ters orantılı olarak tanımlanan maliyet (cost) değerleri kullanılır. En iyi yol en düşük maliyete sahip olan yoldur ve en iyi yollar maliyet bilgileri ile beraber yönlendirme tablosuna yazılır.

OSPF'de bir bağlantının durumu değiştiğinde, değişikliği fark eden yönlendirici hat durumu ilanı (LSA) denilen yayın paketini tüm komşularına yayımlar. Her yönlendirici cihaz gelen LSA paketinin bir kopyasını alır, LSDB (Link-State Database) hat durumu veri tabanını günceller ve gelen LSA paketini kendi komşu yönlendiricilerine iletir. Gönderilen bu LSA paketlerinden elde edilen bilgi ile bütün ağ, ağ altyapısındaki değişikliği algılayarak değişikliği tüm ağ altyapısına yansıtır. LSA paketleri içerisinde yönlendiricilerin bağlantı, ara yüz ve hat durum bilgileri yer almaktadır. OSPF çalışan her yönlendirici kendisine ait bir LSA tablosu bulundurur ve bu LSA tablosu diğer yönlendiricilere gönderilerek ağ içerisindeki bütün yönlendiricilerde aynı olması zorunlu olan bir LSDB veri tabanı oluşturulması sağlanır. Oluşturulan bu veri tabanı ağ içerisindeki yol bilgisi ve mesafe hesaplanmasında kullanılır. LSDB veri tabanı hedef ağa giden en iyi yolu hesaplamak için kullanılır. En kısa yol algoritması yardımıyla ağ topolojisi oluşturulur ve bu işlem 30 dakikada bir yeniden gerçekleşir. Ağ üzerinde tespit edilen herhangi bir değişiklik yoksa güncelleme gönderilmez ve "Hello" paketleri dışında ağda herhangi bir protokol trafiği oluşturulmaz.

OSPF protokolünde bütün trafiği yönetecek ve haberleşmeyi sağlayabilecek DR (Designated Router - Belirlenmiş Yönlendirici) olarak adlandırılan bir yönlendirici seçilir. Buna ek olarak DR yedeği olan BDR (Backup Designated Router - Belirlenmiş

Yedek Yönlendirici) olarak adlandırılan yedek bir yönlendirici daha seçilir. DR ve BDR seçimleri yönlendirici ID (Identification Number - Kimlik Numarası)'si olarak adlandırılan yönlendirici kimliklerine göre ile yapılır. Yönlendirici kimliği, bir yönlendiricinin aktif olan arayüzlerindeki en yüksek IP (Internet Protokol) adresidir. Ancak bu ağ içerisinde herhangi bir "loopback" adresi tanımlanmışsa o yönlendiricinin ID'si loopback IP'sidir. OSPF protokolünün yönetimi EIGRP protokolüne göre daha zordur fakat EIGRP protokolüne göre daha geniş ağ yapılarını yönetebilecek kapasiteye sahiptir. Farklı OSPF bölgeleri arası iletişimde alan sınırı yönlendiricisi (Area Border Router: ABR), farklı yönlendirme protokolleri arasında iletişimde otonom sistem sınırı yönlendiricisi (Autonomous-System Border Router: ASBR) tipleri kullanılır.

OSPF protokolünde üç farklı tablo kullanılmaktadır (Kırcı, Toka ve Erdem, 2015):

- Komşuluk tablosu: Yönlendiricinin komşularının listesi tutulmaktadır.
- Topoloji tablosu: Ağ içerisindeki bütün yönlendiriciler ve onların bağlantıları tutulmaktadır. Bu tablo LSDB olarak adlandırılmaktadır.
- Yönlendirme tablosu: Yönlendirme veri tabanı olarak da adlandırılır. Hedef ağlara giden en kısa yolların bilgisi bu tabloda tutulur.

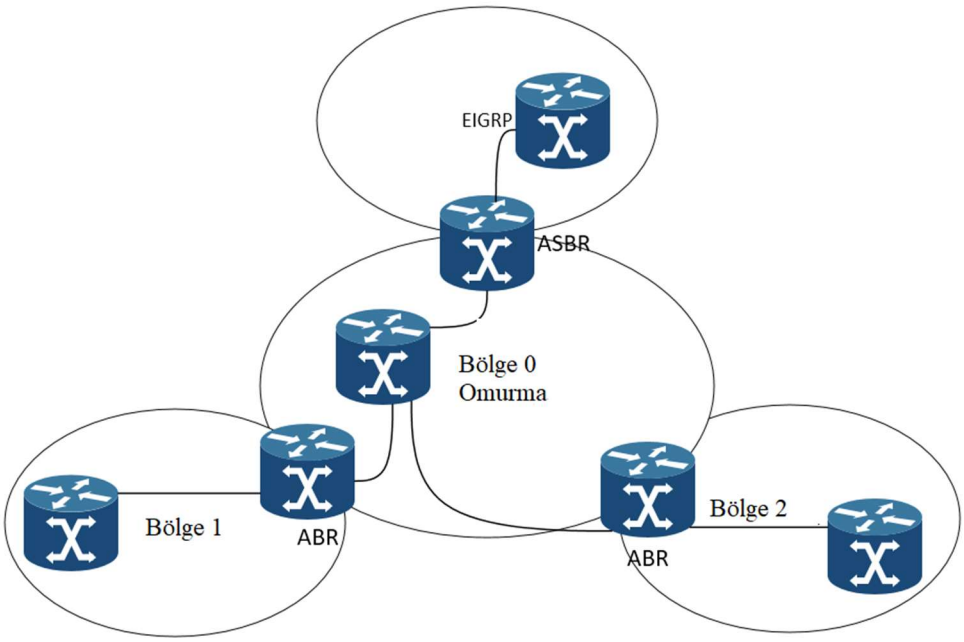
OSPF protokolünün özellikleri:

- VLSM desteği vardır.
- OSPF protokolü sınıfsız çalışabilir.
- Ağ üzerinde yük dağılımı yapabilir.
- Yönetimsel uzunluk değeri 110'dur.
- Bölge yapısını kullanır.
- Metrik olarak "cost calculation" yöntemi kullanılır.
- Tetiklenmiş güncellemeler kullanır.
- En kısa yol algoritması kullanılır ve bu algoritma ile tüm ağ altyapısının topolojisi çıkartılabilir.
- "Hello" paketleri kullanılır.
- Komşular arası haberleşmede LSA1, 2, 3, 4, 5, 6, 7 paketleri kullanılır.

- Güncelleme mesajları için LSU (Link State Updates - Hat Durum Güncellemeleri) paketleri kullanılır.

OSPF protokolünün ağ üzerinde kullanımının ölçeklendirilmesi kullanılan cihazların bellek, işlemci ve port bant genişliklerine bağlıdır. OSPF protokolü üzerinde oluşan kullanım yüküne ağların sayısı, komşu sayısı ve bölge sayısı gibi faktörler etki etmektedir. OSPF protokolü tasarlanmadan önce bu faktörler göz önünde bulundurularak yapılandırma ve cihaz seçimi yapılmalıdır. Tasarım aşamasında ortalama 50 cihazda bir yeni bir bölge ataması planlanmalıdır. Böylece daha istikrarlı ve yönetilebilir bir OSPF ağ yapısı oluşturulacaktır.

Tasarım aşamasında ABR ve ASBR yönlendiricileri için, OSPF ağ yapısındaki diğer cihazlardan daha yüksek performans değerlerine sahip cihazlar seçilmelidir. Çünkü ABR ve ASBR yönlendiricileri bölge içinde yer alan diğer yönlendiricilere göre daha fazla işlem yapacaktır. OSPF ağ yapısında ABR ve ASBR yönlendiricilerine ait LSA paketlerine filtrelemeler uygulanarak bu paketler kontrol altına alınmalı, elle adres özetleme yoluyla yönlendirme tabloları ve güncellemeler kontrol altına alınarak cihazlar üzerinde gereksiz işlemler engellenmelidir. Oluşturulan her bölge içindeki yerel yönlendiriciler “stub” olarak yapılandırılmalıdır. ABR ve ASBR yönlendiriciler OSPF ağ tasarımlarında Şekil 3.23’de görüldüğü gibi farklı bölge sınırlarında OSPF protokol özelliklerine uygun olarak sınır yönlendirici tipinde belirlenmeli ve konumlandırılmalıdır.

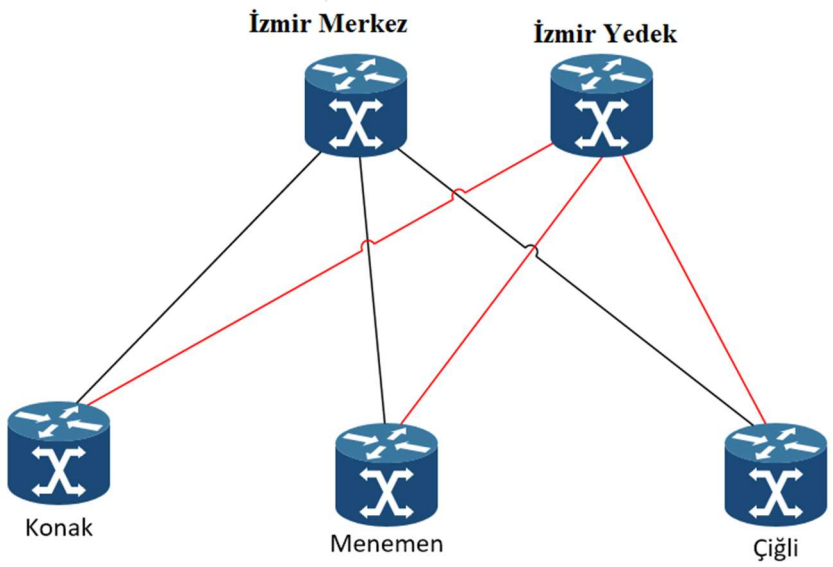


Şekil 3.23 OSPF topoloji örneği

3.3.4 Merkez Uç Nokta Ağ Altyapılarının Tasarımı

Hizmet verilen bölgede kuruma ait kablolama altyapısı tüm hizmet noktalarına ulaştırılamayabilir. Bu durumda ISS (İnternet Servis Sağlayıcılardan) ek altyapı hizmeti almak gerekebilir. Günümüzde en çok kullanılan altyapı hizmeti Katman 2 SÖA veya Katman 3 SÖA'dır. Maliyetsel ve bazı tasarımsal avantajlarından dolayı Katman 3 SÖA hizmeti daha çok tercih edilmektedir. Uzak noktalara altyapı sağlamak için Katman 3 SÖA hizmeti ISS'den alınabilir. ISS belirttiğimiz noktalara kendi altyapısı üzerinden bir bağlantı sağlar ve bağlantı götürdüğü bu noktaları kendi tarafında bir ÇPEA bulutu içinde toparlayarak müşteriye sunar. Böylece ağ altyapı hizmetlerine uç noktalar da dâhil edilmiş olur. Merkez ile şube trafiklerini haberleştirmek istediğimizde sadece ÇPEA altyapıları çözüm olmayacaktır. Çünkü EIGRP veya OSPF gibi yönlendirme protokollerinin merkez ile uzak noktalar arasında çalışması için yönlendiriciler arası kurulan komşulukların noktadan noktaya olması gerekmektedir. Bu yüzden ÇPEA ağ yapılarında doğrudan dinamik yönlendirme kullanımında problemler yaşanacaktır. Bu sorunu aşmak için merkez ile uç noktalar arasındaki bağlantının noktadan noktaya haline getirilmesi gerekmektedir. Bunun için GRE (Generic Routing Encapsulation - Genel Yönlendirme Kapsülleme), mGRE

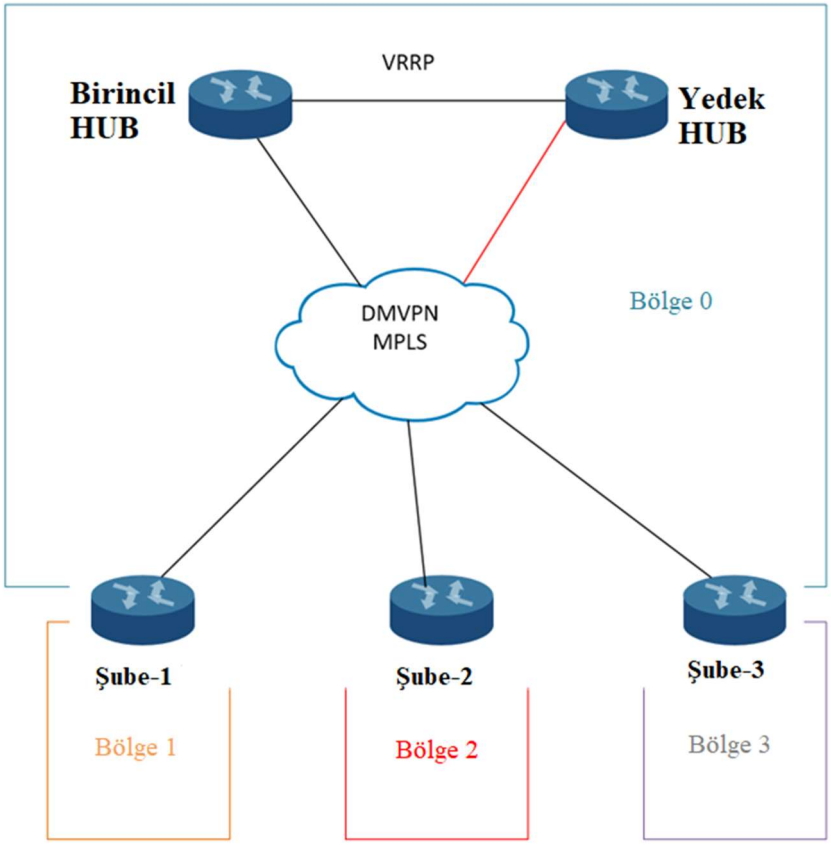
(Multipoint Generic Routing Encapsulation - Çoklu Genel Yönlendirme Kapsülleme) veya DMVPN (Dynamic Multipoint Virtual Private Network - Dinamik Çok Noktalı Sanal Özel Ağ) protokollerinden biri kullanılarak çözüme kavuşulabilir. Bu yöntemlerin içinden merkez-uç nokta topolojileri için şubelerin birbirleriyle de iletişim kuracağı varsayılırsa en iyi yöntem DMVPN faz 3 olacaktır. DMVPN'de uç noktalar hem merkez hem de şubeler ile NHRP (Next Hop Resolution Protocol - Sonraki Atlama Çözünürlüğü Protokolü) protokolünün yardımı ile iletişim sağlamaktadır. Burada uç nokta diğer uç noktaya bağlanırken sadece ilk gidişinde merkez üzerinden gideceği şubenin bilgilerini öğrenir, sonraki trafik akışında NHRP protokolü aracılığı ile direk uç noktadan uç noktaya bağlantı kurulur. Böylece merkez yönlendirme cihazlarının üzerindeki yük hafiflemiş olur. Tüm ağ cihazı üreticilerinde DMVPN desteği vardır. EIGRP DMVPN ağ yapılarının faz 3'de yayılım süreleri hızlıdır. Fakat her 600 uç noktada bir merkezde yeni bir yönlendirici konumlandırılması gerekir (KnowledgeClub, 2020b). Şekil 3.24'de merkez-şube DMVPN ağ topolojileri için mantıksal bağlantılar gösterilmiştir.



Şekil 3.24 Örnek DMVPN Topolojisi

Farklı üreticilerin ürünlerinin konumlandırıldığı bir ağ altyapısının tasarımında OSPF gibi açık standart kullanılması gerekir. OSPF protokolü büyük ağları kontrol

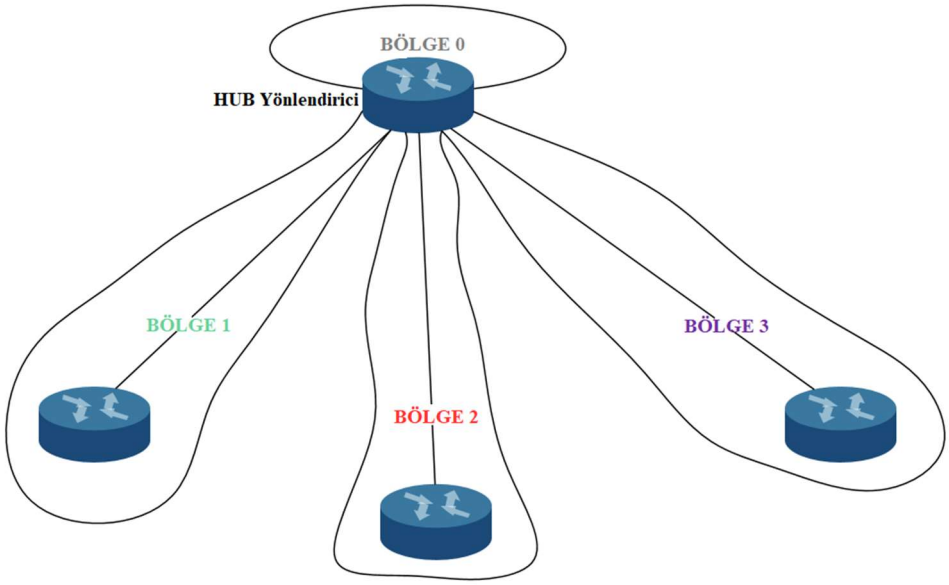
etmede daha başarılı olmasına rağmen DMVPN ağlarında bazı dezavantajları vardır. DMVPN tasarımında topolojideki tüm yönlendiricileri Bölge 0 (Area 0) üzerinde konumlandırma ya da farklı bölgelere bölme gibi farklı OSPF bölge topolojileri uygulanabilir. Kullanılabilecek en iyi tasarım yöntemlerinden biri de aşağıdaki gibi uç noktaları farklı bölgelere ayırma yöntemidir. Fakat bu yöntemin seçimi tasarlanacak yapıdaki cihaz sayısına göre değişebileceği için kurulacak yapıya göre değerlendirilmelidir. Büyük ağ yapıları için aşağıdaki gibi bir topoloji kullanılabilir.



Şekil 3.25 DMVPN OSPF topoloji örneği

Şekil 3.25'deki gibi bir DMVPN ağ yapısında uç noktalardaki yönlendiricilerin ABR olarak yapılandırılması tercih edilebilir. Fakat bu tip DMVPN yapılandırılmalarında uç noktalardaki GAA (Geniş Alan Ağı) hattı istikrarının bozularak hattın sürekli aktif pasif olmaya başlaması ile Bölge 0 (Area 0) bölgesine doğru bağlantı durumu güncelleme seli (LSU flood) problemi oluşması ihtimali

olacaktır (Knight, 2015). Bu tip bir problemin oluşmasını önlemek için Şekil 3.26'daki gibi bir topoloji tasarlanmalı veya LSA süreleri üzerinde değişiklikler yapılmalıdır.



Şekil 3.26 DMVPN OSPF topoloji örneği 2

Şekil 3.26'daki topolojide merkez yönlendiricisi ABR olarak yapılandırılmakta ve uç nokta bölge sınırları, her bölge için merkez yönlendirici arayüzüne kadar uzatılmaktadır. Böylece uç noktalardan gelen hat istikrar problemlerinin sebep olabileceği problemler en aza indirgenmiş olur. Tabii ki böyle bir yapıda merkez yönlendiricisi için ağ yapısının büyüklüğüne ve ihtiyaçlarına uygun güçlü bir cihaz belirlenmelidir.

OSPF protokolünün DMVPN ağlarında kullanılmasının oluşturduğu bazı dezavantajlar aşağıdaki gibidir:

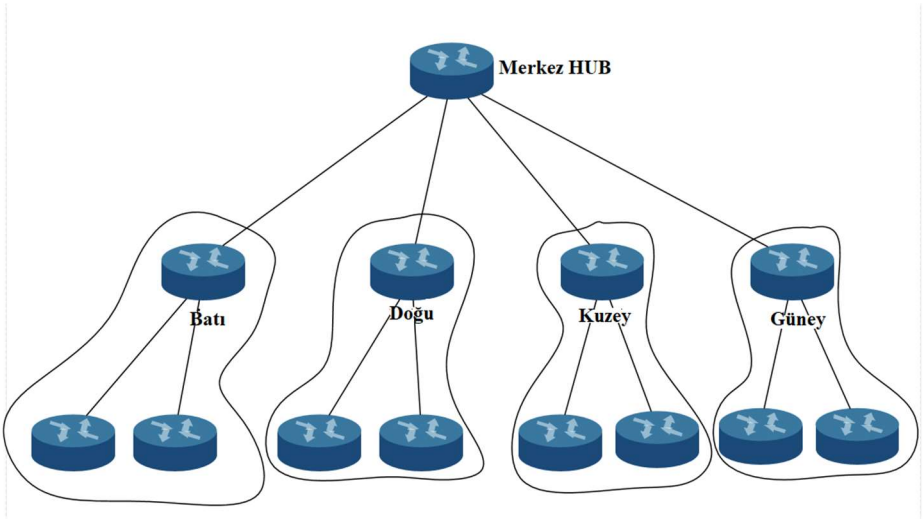
- Yönlendirme güncelleme paketlerinin filtrelenmesinde yaşanan problemler,
- Uç noktalarda hatların sürekli aktif pasif olması ile hat istikrar problemlerin yaşanması sonucunda merkeze doğru oluşacak "LSU Flood" problemi,

- Topoloji değişikliği ile ortaya çıkabilecek olumsuz etkilerin özetleme veya yönlendirme tablosu filtreleme yöntemleriyle kontrol altına alınamaması,
- Yönlendiricilerin kontrol katmanında daha fazla yük oluşması.

Diğer taraftan, DMVPN ağlarında EIGRP protokolü kullanımının OSPF'ye kıyasla sağladığı bazı avantajlar aşağıdaki gibidir:

- Yönlendiricilerin kontrol katmanında oluşan yükün OSPF'ye göre daha istikrarlı ve daha az olması,
- EIGRP protokolünün daha esnek trafik mühendisliği sağlaması,
- Yönlendirme güncellemelerinin, özetleme ve yönlendirme filtreleme yöntemleri ile kontrol altına alınabilmesini sağlaması,
- Topoloji değişikliğinde ortaya çıkabilecek etkilerin OSPF protokolüne oranla daha az olması ve gerektiğinde filtrelerle kontrol altına alınabilmesini sağlaması.

OSPF protokolü kullanıldığında ortamda çok fazla bölge oluştuysa ve protokolün taşıyabileceği sayıdan daha fazla uç nokta varsa, o zaman sınır ağ geçidi protokolü (Border Gateway Protocol: BGP) ağ tasarımında kullanılabilecek olası seçenekler arasında değerlendirilmelidir. Tasarlanacak DMVPN ağ yapısı binlerce noktadan oluşuyorsa aşağıdaki Şekil 3.27'de olduğu gibi uç noktaların ayrılıp her bölge için farklı bir DMVPN yönlendiricisi konumlandırılmalıdır. Böylelikle bölge yönlendiricilerinin merkez ile haberleştirilmesiyle daha istikrarlı bir ağ altyapısı tasarlanmış olacaktır.



Şekil 3.27 Bölgesel DMVPN topoloji tasarım örneği

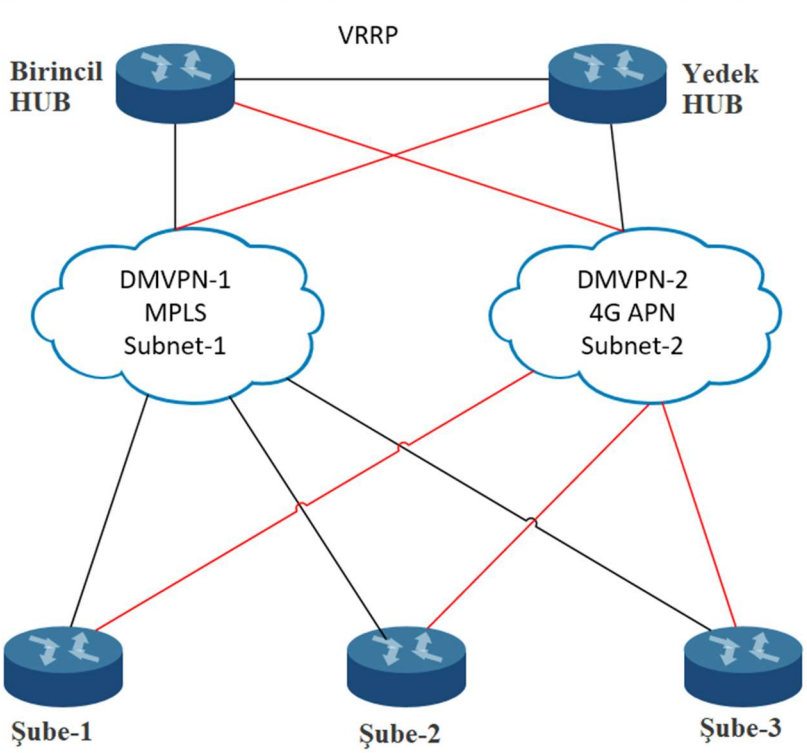
Yukarıdaki gibi bir tasarım ile bölgelerde oluşan trafik kendi bölgeleri içinde kalacak ve bu özel trafiklere has filtrelemeler uygulanarak bunların diğer bölgeleri ve merkezi daha az oranda etkilemesi sağlanacaktır. Bu yapıdaki bölgeler için IBGP (Internal Border Gateway Protocol - İç Sınır Geçit Protokolü) protokolü, bölgeler ve merkez arası trafik için EBGP (Exterior Border Gateway Protocol - Dış Sınır Geçit Protokolü) protokolü tercih edilirse daha sağlıklı bir altyapı oluşturulabilir. BGP protokolünün burada sağlayacağı en önemli faydalardan biri de ağ altyapısının sonradan istenilen kadar genişletilmesine olanak sağlamasıdır. BGP çok büyük ağ altyapılarını rahatlıkla yönetebilecek ve genişleyebilirlik ihtiyaçlarını karşılayacak yapıya sahiptir. Bu yüzden ISS alt yapılarında tercih edilmektedir. OSPF ve EIGRP protokolleri BGP protokolüne kıyasla bu tip çok geniş ağları yönetmede sorunlar yaşatabilecek ve ağ altyapısında oluşan bir değişikliğe bu protokollerin tepki süresi ağ büyüdükçe uzayacak, plansız hazırlanan bir ağ yapısındaki değişikliklerin algılanması için dakikalarca beklenmek zorunda kalınabilecektir (Bahnasse ve Kamoun, 2015).

3.3.5 Ağ Altyapılarında Yedeklik Tasarımı

3.3.5.1 Merkez Uç Nokta Ağ Altyapılarının Yedekliği

DMVPN ile merkez uç nokta ağ altyapısına dâhil edilen uç noktalar ve merkez için yedeklik planı yapılması sürdürülebilirlik yönünden oldukça önemlidir. DMVPN

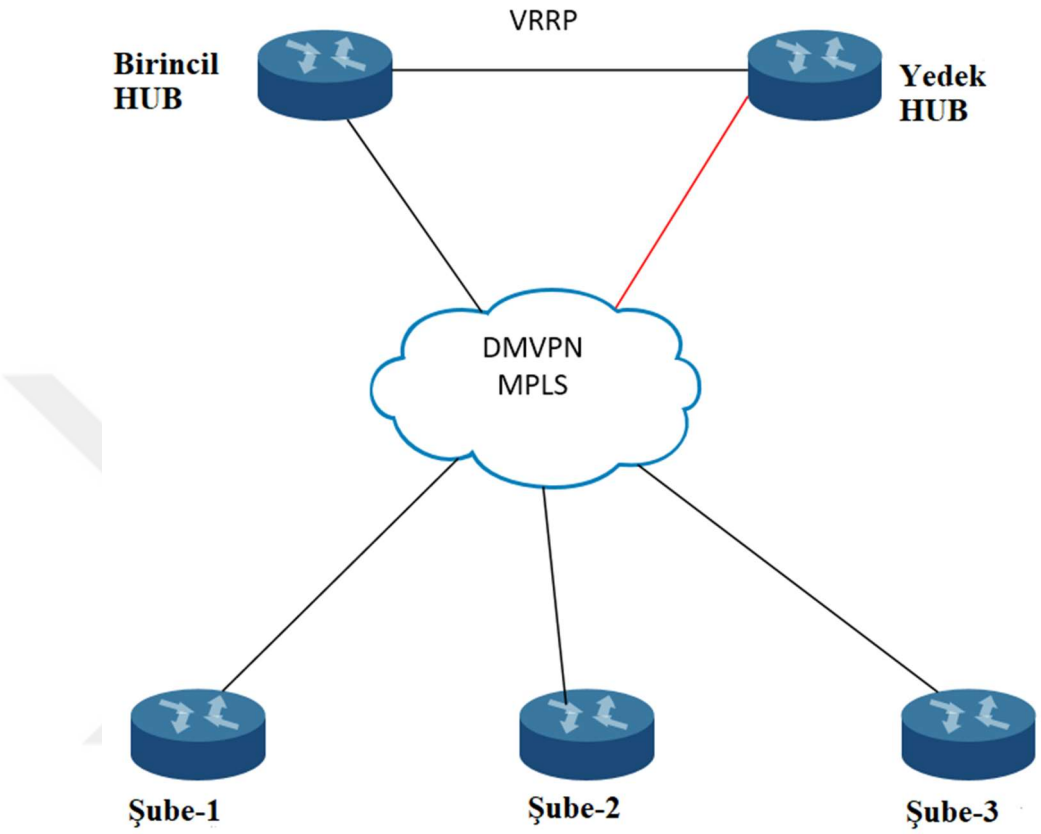
ağlarında uç noktalarda veya merkezde yedekleme çalışmaları yürütülebilir. Yedeklik seçenekleri arasında hat yedekliği oluşturulması maliyetleri artırırırken sürdürülebilirlik ve erişilebilirlik oranlarını da artıracaktır. Yedekleme ihtiyaçları, uç noktaların ve planlanan ağ yapılarının kritikliği ile ilişkilidir. Kesintiye direnci olmayan ağ altyapılarında para, zaman, itibar gibi kayıpları engellemek için yedekleme planlaması yapılması gerekebilir. Uç noktaların GAA bağlantılarının yedeklemesi için ana hat olarak ÇPEA seçilebilirken yedek hat olarak da mobil iletişimler için küresel sistem erişim noktası adı (Global System for Mobile Communications Access Point Name: GSM APN) hizmeti seçenekler arasında değerlendirilebilir.



Şekil 3.28 2 DMVPN bulutlu şube, merkez yedekliği

Şekil 3.28'deki topoloji örneğinde uç noktalarda ana hat olarak ÇPEA hattı tercih edilirken yedek hat olarak GSM APN hizmeti seçilmiştir. Uç nokta tarafında karasal hat koptuğunda şube GSM üzerinden çalışmaya başlayacaktır. Merkez tarafında yönlendirici yedekliği planlanmış ve yönlendiriciler arasındaki yedeklik için VRRP protokolü kullanılması öngörülmüştür.

Uç nokta tarafında yedek hat ihtiyacının olmadığı durumlarda ise Şekil 3.29'daki gibi bir yapı tasarlanabilir.



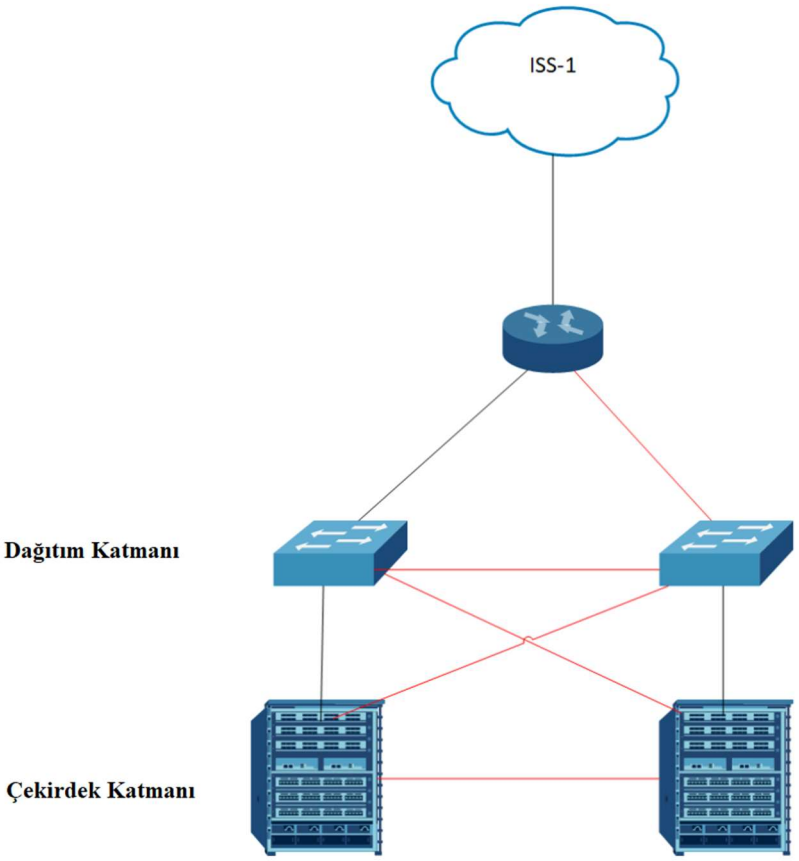
Şekil 3.29 Merkez yedekliği olan DMVPN topoloji örneği

Bu topolojide uç noktalarda yedek hat planlanmamışken merkezde hem hat hem de yönlendirici yedekliği planlanmıştır.

3.3.5.2 Hiyerarşik Altyapı Tasarımlarında Hat Yedekliği

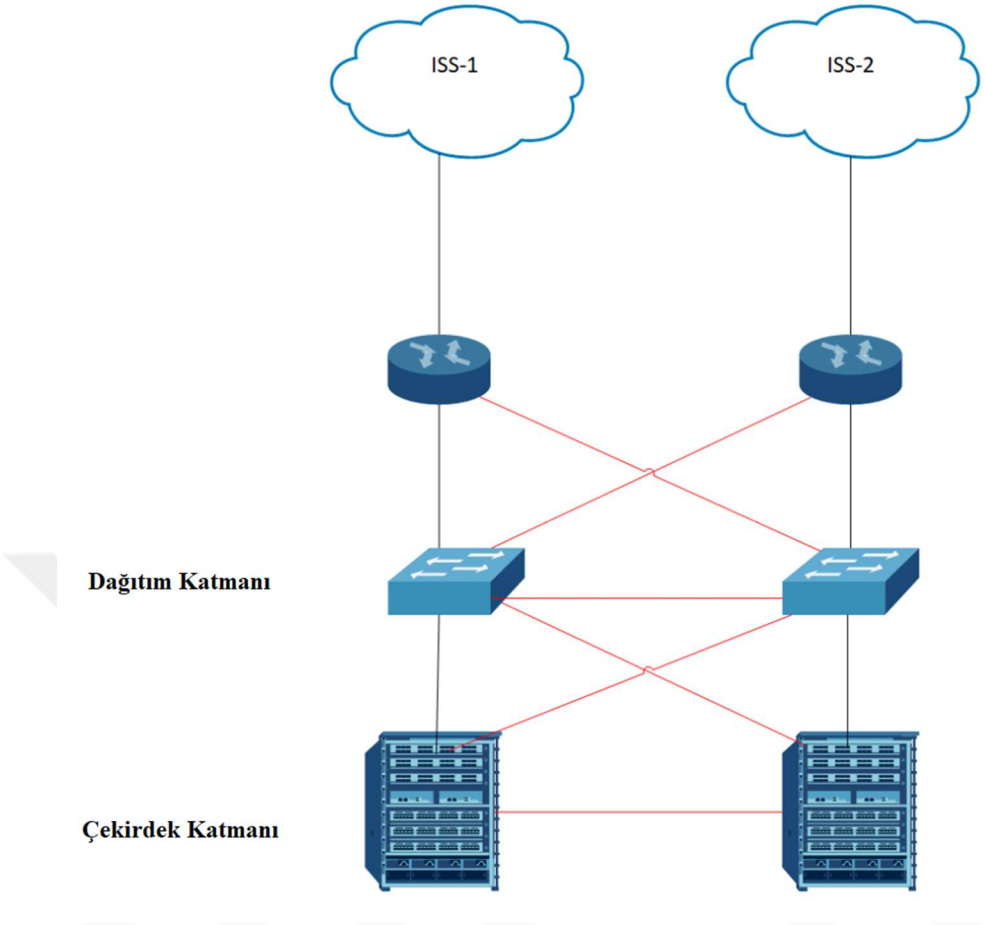
Ağ altyapılarının birbiri ile bağlantı sağlayabilmesi veya internet bağlantısı sağlanabilmesi için GAA altyapıları kritik rol oynamaktadır. İnternet bağlantıları, kurumlar arası bağlantılar ve şubeler arası bağlantılar GAA altyapıları üzerinden gerçekleştirilir. GAA cihazları üzerinde oluşabilecek herhangi bir problem veri, para, itibar kaybına ve operasyonel süreçlerin olumsuz etkilenmesine sebep olabilmektedir. Bu risklerin azaltılması için GAA cihazları için süreklilik planları yapılması

gerekmektedir. Bu planlar ihtiyaç, kritiklik ve maliyet bazında değerlendirilmelidir. Bu kriterlere göre birçok farklı topoloji tasarlanıp kullanılabilir.



Şekil 3.30 Tek ISS bağlantılı GAA alt yapısının yedekleme örneği

Şekil 3.30'daki örnek incelendiğinde topoloji üzerinde tek bir ISS bağlantısı mevcuttur. Yedek GAA hattı olmayan tasarımlarda, örnekteki gibi ISS cihazlarına bağlantı sağlayan dağıtım katmanı kenar anahtarı yedeklenerek veya ikinci bir yönlendirici konumlandırılarak hem ISS cihazları hem de dağıtım kenar anahtarları için yedeklik ve süreklilik artışı sağlanabilecektir.



Şekil 3.31 İki ISS bağlantısına sahip GAA bağlantılarının yedekliği

GAA altyapılarının yedekliğinin sağlanması ve sürekliliklerinin artırılması için kritik yerlerde Şekil 3.31'deki örnekte olduğu gibi en az iki ISS bağlantısı bulunmalıdır. Gerçek yedeklik için ISS sağlayıcıları farklı olmalı ve servis sağlayıcı altyapılarının farklı güzergâhlardan geldiğinin teyidi alınmalıdır. Çünkü aynı altyapıyı kullanan iki servis sağlayıcıdan alınan bir hizmet, hizmet alınan altyapıda problem olması halinde her iki bağlantının da etkilenmesine ve kesintisine sebep olacaktır. Bu da en baştan yanlış yapılandırılmış bir yedeklemedir.

3.3.5.3 Felaket Kurtarma Merkezi (FKM) Tasarımı

Büyük ve kritik ağ altyapılarının yedeklenmesi kurumlar için önemli bir ihtiyaçtır. Özellikle ağ altyapısı üzerinden çok sayıda vatandaşa kamu hizmeti veriliyorsa, kesinti tahammülü yok ve kesinti anında telafisi zor kayıplara sebep olacaksa merkez

yapısının yedeklenmesi kaçınılmazdır. Özel kuruluş ve kamu kurumlarında FKM ile ilgili birçok çalışma yapılmaktadır. FKM çalışmalarında tasarım aşamasına geçilmeden önce FKM'nin konumu kararlaştırılmalıdır. Merkez ile aynı binada veya odada yapılan bir FKM yatırımı çok da doğru bir seçim olmayacaktır. FKM konumu ile ilgili binanın konumu, sağlamlığı, deprem fay hatlarına yakınlığı gibi birçok risk başlığı göz önünde bulundurulmalıdır.

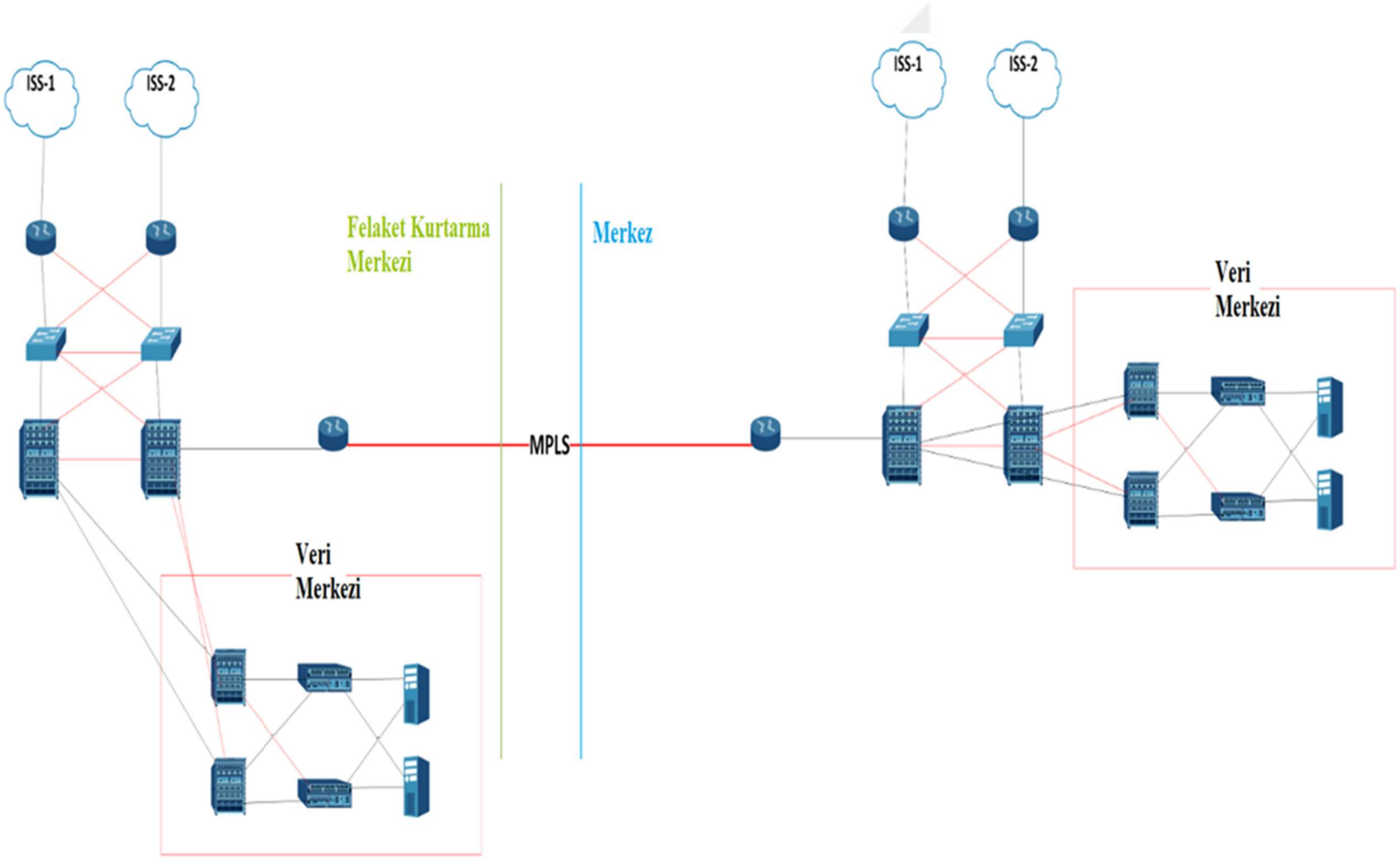
FKM tasarımında ana amaç merkezi altyapıda bir problem olduğunda FKM cihazlarının devreye girerek kesintisiz hizmet verilmesinin sağlanmasıdır. Tüm cihazların birebir FKM noktasında yedeklenmesi gerekmese de özellikle kritik cihazların FKM noktasında yedeklenmesine ihtiyaç vardır.

FKM altyapılarının oluşturulmasında geçmişten günümüze birçok protokol ve teknoloji kullanılmış ve teknolojinin gelişmesi ile tercihler değişmiştir. Önceki yıllarda L2TP (Layer 2 Tunneling Protocol - Katman 2 Tünel Protokolü) tünel, GRE tünel ve noktadan noktaya müşteriye özel fiber gibi çözümler tercih edilmekteydi. Fakat bu çözümlerin dezavantajı ağ yapısında cihazların yayımladığı paketlerinin diğer noktalara da iletilmesiyle oluşabilecek ağ problemlerinin tüm noktaları etkilemesiydi. Ama son yıllarda ağ terminolojilerine giren yeni teknolojiler ile tercih edilen çözümler de değişmiştir. Güncel tasarımlarda avantajlarından dolayı özellikle VXLAN (Virtual Extensible Local Area Network - Sanal Genişletilebilir Yerel Alan Ağı) ve OTV (Overlay Transport Virtualization - Kaplama Taşıma Sanallaştırması) protokolleri tercih edilmektedir. Bu protokollerin en önemli avantajlarından bir tanesi cihaz yayınlarının noktalar arasında taşınmamasıdır. Cihaz yayınlarını (Broadcast) taşıma gereksinimi oluştuğunda da söz konusu protokoller bu yetkinliğe sahiptir.

Şubeler haberleştirilecekse DMVPN ve GRE protokolleri tercih edilebilir. Ama veri merkezleri haberleşecekse aynı sanal ağda olmaları gerektiği için OTV veya VXLAN protokollerinin kullanılması daha iyi bir tercih olacaktır. Merkez sistem odası ile FKM arasında tasarlanacak bağlantının Katman 3 SÖA ÇPEA altyapı hizmeti üzerinde kurulması daha çok tercih edilmektedir. Katman 2 SÖA ÇPEA altyapı

hizmeti ile tasarlanırsa iki nokta arasındaki yayın bölgeleri birleştirmiş olur ve bu da altyapı için bir risk oluşturabilir.

Şekil 3.32'deki yedeklik tasarım örneğinde merkezde bulunan veri merkezi cihazları FKM bölgesinde yedeklenerek tasarlanmış ve ISS hat yedeklikleri de tasarıma dâhil edilmiştir. İki nokta arasında ISS'den ÇPEA altyapı hizmeti alındığı varsayılarak bir topoloji tasarlanmıştır. İstenirse aradaki bağlantı Katman 2 olarak çalıştırılarak merkez ve FKM bölgelerindeki yönlendiriciler arası VRRP protokolü çalıştırılarak yönlendiriciler arasında aktif pasif yedeklik sağlanabilir.



Şekil 3.32 FKM topoloji örneği

3.3.6 Altyapıların Geleceği Olarak Yazılım Tabanlı Ağlar

YTA, ağ cihazlarındaki kontrol katmanı ile veri katmanının birbirinden fiziksel olarak ayrıştırılması ile tüm cihazların yazılımsal bir kontrol katmanından yönetilmesi olarak tanımlanabilir. İlk önerildiğinde saf YTA olarak ortaya çıkmış ve saf YTA'daki tek kontrol katmanı fiziksel cihazlardaki veri katmanlarını yönetmek üzerine tasarlanmıştı. Fakat YTA merkez cihazında ve yedeğinde bir sorun meydana geldiğinde fiziksel cihazlar işleyişine devam edemiyor ve ağ altyapısı çalışamaz hale geliyordu. Bu problemi ortadan kaldırmak için melez yapıya geçildi (KnowledgeClub, 2020a). Melez yapıda YTA yönetici cihazında sorun olsa dahi ağ cihazları en son merkez cihazdan aldıkları yapılandırma ile işleyişine devam edebilecek hale dönüştürüldü.

YTA'ın yaygınlaşması ile artan geliştirme çalışmaları ile ağ mühendislerine bağımlılık azalacak, kullanıcıya özel yapılandırmalar kolayca kullanıcı ile beraber taşınabilecek ve programlama desteği ile yazılıma istenilen yeni özelliklerin kolayca eklenebilecektir. Ayrıca cihazlar üzerinde farklı yapılandırmalara gerek kalmadan tek yazılım üzerinden hat performansına göre uygulamaların yönlendirilebilmesi ve ağ ile ilgili tüm işlemler yazılım üzerinden takip edilebilmesi mümkün hale gelmiştir. YTA üzerinde çeşitli geliştirilmelere halen devam edilmektedir.

YTA ağ yapılarının bazı avantajları:

- YTA ağ yapıları programlanabilme özellikleri ile ağ altyapısına esneklik ve yeni özellikler eklenebilmesine olanak sağlar.
- Yönetim kolaylığı sağlar.
- Ağ trafiğinin daha esnek şekilde yönetilmesini sağlar.
- Ağ trafiğinin merkezi olarak optimize edilmesini sağlar.
- Python, xml, javascript, json kullanım desteklerini sağlar.

BÖLÜM DÖRT

BİT ALTYAPILARI İÇİN TEMEL PERFORMANS GÖSTERGELERİ

TPG modern iş sistemleri dünyasının her alanında sıklıkla kullanılan terimlerdendir. TPG ifadeleri ve tanımları çok yaygın olmasına rağmen, TPG'leri etkin bir şekilde kullanan işletmelerin aslında çok yaygın olmadığı görülür. TPG'leri doğru tanımlanıp düzgün kullanıldığında, bir işletmenin başarısında büyük bir fark yaratabilir.

4.1 TPG Nedir?

Basit bir ifadeyle TPG'ler şirketlerin, iş birimlerinin, projelerin veya bireylerin stratejik hedefleri ve hedeflerine göre ne kadar iyi performans gösterdiklerini ölçmenin bir yolunu sunar. TPG'ler, belirlenen hedefler doğrultusunda ilerlendiğinin anlaşılmasını sağlayan en önemli performans ölçütlerini sağlar. İyi tasarlanmış TPG'ler, mevcut performans seviyelerinin ve işletmenin bulunması gereken seviyede bulunup bulunmadığını net bir şekilde gösteren hayati navigasyon araçlarıdır. TPG'ler karar verme süreçlerini iyileştirmeye ve dolayısıyla toplam performansı geliştirmeye yardımcı olabilirler (Marr, 2016).

4.2 Etkili TPG'ler Nasıl Geliştirilir?

TPG'leri yapılan işe uygun olarak tasarlanıp belirlenirse ve TPG kullanılarak elde edilen veriler düzenli olarak analiz edildiğinde sistemin işleyişi hakkında kritik öneme sahip bilgi kaynağı sağlanır. Etkili gösterge tasarlamak için göstergenin adı, veri toplama yöntemi, değerlendirme, formül, ölçek kriterleri, hedefler ve performans eşikleri, veri kaynağı, veri toplama sıklığı ve veri raporlama sıklığı belirlenmelidir. BİT'deki planlama, uygulama, hizmet, altyapı ve destek gibi tüm süreçler için TPG tasarımı yapılabilir.

Amaca özel ve etkili TPG'leri tasarlamak ve tasarlanan TPG'den faydalanabilmek için sırasıyla aşağıdaki adımlar göz önünde bulundurulmalıdır (Marr, 2016).

1. Strateji belirlenmesi,
2. Cevaplanması gereken soruların tanımlanması,
3. Veri ihtiyaçlarını belirlenmesi,
4. Mevcut tüm verilerin değerlendirilmesi,
5. Uygun destekleyici verilerin bulunması,
6. Doğru ölçüm yöntem ve sıklıklarının belirlenmesi,
7. TPG'ler için sahiplik atanması,
8. TPG'lerin kuruluş çalışanları tarafından anlaşılması,
9. Performansı artırmaya yardımcı olduklarından emin olmak için TPG'lerin belirli periyotlarda analiz edilmesi.

4.3 Altyapılar İçin TPG Çalışması

Temel performans göstergeleri derlenirken Tablo 4.1'deki genel BİT TPG kategorileri temel alınmıştır. Bir BİT altyapısının sahip olması gereken ana özellikler genel kategoriler başlığı altında belirtilmiş, diğer fonksiyon (işlev/faaliyet) başlıkları altında yer alan TPG'leri doğal olarak bu kategorilerle ilişkilendirilmiştir. TPG'lerin ana amacı bir BİT altyapısının mümkün olan en az kesinti ile sürdürülebilirliğinin sağlanmasıdır. Bunun sağlanması için öncelikle altyapı performansının ölçülebilir ve değerlendirilebilir olması gerekmektedir. Ölçülemeyen ve değerlendirilmesi yapılmayan hiçbir altyapının uzun vadede hedeflenen verimlilikte çalışması mümkün görünmemektedir. Bu çalışma kapsamında var olan TPG'leri işlevsel başlıklar altında sınıflandırılmış ve var olan göstergelere yenileri eklenmiştir. Bu göstergeler istendiği takdirde performansı değerlendirilecek BİT altyapılarına göre özelleştirilebilir veya ağı yöneten mühendislerce genel kategorilerde tanıtılan ana göstergeler temel alınarak yeni göstergeler eklenebilir. Tablo 4.1'deki her göstergenin el ile hesaplanması veya takip edilmesi çok verimli bir seçenek değildir. Bu göstergelerin yazılımlar aracılığı ile takip edilmesi göstergelerden gelen önemli verilerin zamanında değerlendirilmesi

açısından önemlidir. Bu amaca yönelik kullanılan açık kaynaklı ağ performans izleme yazılımlarından bazıları aşağıda listelenmiştir:

- Zabbix (www.zabbix.com)
- Mrtg (www.mrtg.com)
- Cacti (www.cacti.net)
- Nagios (www.nagios.com)
- OpenNMS (www.opennms.com)

Tablo 4.1 TPG tablosunda yer alan göstergeler için bu yazılımlardan birkaçını kullanmak altyapının takibini ve verimliliğini ölçmeyi kolaylaştıracaktır. Yazılımların listelerinde olmayan yeni TPG tanımları için de yazılım üzerinde özel TPG girdileri oluşturmak gerekebilmektedir.

BT alanında TPG ile ilgili literatürdeki çalışmaları birleştirerek altyapı izlenebilirliği ve verimlilik artışı doğrultusunda temel alınacak bir kaynak oluşturulması amaçlanmıştır. Bunun için farklı alanlarda yapılmış TPG çalışmaları incelenmiş, bu çalışmalardaki önemli görülen göstergeler ve yeni belirlenen göstergeler bir havuzda toparlanmıştır. Belirlenen göstergeler farklı kategorilere ayrılarak Tablo 4.1 TPG tablosu oluşturulmuştur. Bu tablo için yararlanılan başlıca akademik ve diğer kaynaklar aşağıdaki gibidir:

- Personal, Guerrero, Garcia, Pena ve Leon (2014);
- Haizan, Alinda ve Rahman (2011);
- Adepetu, Arnautovic, Svetinovic ve Weck (2014);
- Jibril (2005);
- McGill (2007);
- Celebic ve Breu (2015);
- Rohani ve Roosta (2014);
- Raza, Bin Muhammad ve Majid (2016);
- KPI library (2020).

Tablo 4.1 Temel performans göstergeleri tablosu

BİT GENEL TPG KATEGORİLERİ	İnternet Bant Genişliği
Kullanılabilirlik	Toplam İnternet Bant Genişliği
Güvenilirlik	Gönderilen/Alınan Trafik
Genişleyebilirlik	Planlanmamış Kesintiler
Güvenlik	MTTD (Ortalama Tespit Süresi)
Birlikte Çalışabilirlik	En Yüksek İnternet Kullanımı
Bakım Yapılabilirlik	En Az İnternet Kullanımı
Geri Döndürülebilirlik	**BİT EKİPMAN TPG'LERİ**
Belgelenebilirlik	Kapasite Kullanımı
Sürdürülebilirlik	Bellek Kullanımı
BİT ALTYAPI TPG'LERİ	İşlemci Kullanımı
Kullanılabilirlik	Depolama Kullanımı
Güvenilirlik	Sunucu İşlemci Kullanımı
Bağlantı Kapasitesi	Sunucu RAM Kullanımı
Gecikme	Sunucu Depolama Alanı Kullanımı
Ağ İşlem Hacmi (Throughput)	Sunucu Ağı Kullanımı
Dalga Bozulumu	Ağ Kapasitesi
Kayıp Paket	Depolama Kapasitesi
RTT (Gidiş Dönüş Süresi)	Fiziksel Sunucu Sayısı
Yük	Katman 2 Anahtar Sayısı
BER(Bit Hata Oranı)	Katman 3 Anahtar Sayısı
Servis Kalitesi	Yönlendirici Sayısı
Aktarım Hızı	Güvenlik Duvarı Sayısı
Uygulama Yanıt Süresi	Tesislerin Hizmet Ömrü
Tampon	Donanım Hizmet Ömrü
Paket Hataları	Kablolama Hizmet Ömrü
Atılan Paket Sayısı	Maksimum Depolama Alanı Kullanımı
Yayılma Gecikmesi	Minimum Depolama Alanı Kullanımı
Kuyruk Gecikmesi	Kullanımdaki Fiziksel Bağlantı Sayısı
Serileştirme Gecikmesi	Ortalama Disk Okuma Süresi
Erişilebilirlik	Ortalama Disk Yazma Süresi
Engellenen Paketler	Ortalama Disk Aktarım Hızı
Paket Kuyruğu	Bakım Maliyeti
Uçtan Uca Gecikme	

Tablo 4.1 devamı

SES/GÖRÜNTÜ TPG'LERİ	Sunucu Ağ Talebi
MOS (Ortalama Görüş Puanı)	Sunucu İşlemci Talebi
Ses Ağı Kullanılabilirliği	Sunucu Bellek Talebi
Ses Ağı Performansı	Depolama Talebi
Ses Ağı Hizmet Dönüş Süresi	Günlük Depolama Talebi
Ses Ağı Kullanım Oranı	Uygulama Depolama Talebi
CBR (Sabit Bit Hızı)	Günlük İşlem Talebi
VBR (Değişken Bit Hızı)	**İNSAN KAYNAĞI TAKİP TPG'LERİ**
YEDEKLEME TPG'LERİ	Yıllık Zorunlu/İsteğe Bağlı Eğitimler
Yedekleme Kapasitesi	BT Eğitiminde Harcanan Saat Sayısı
Yedekleme İşlemlerinin % Başarı Oranı	Teknik Uzman Sayısı
Yedeklemenin Geri Yüklenmesi için Ortalama Süre	Teknik Ekip Sayısı
Ağ Ekipmanı Yapılandırma Yedeklemesi	**GÜVENİLİRLİK TPG'LERİ**
Kritik Veri Yedekleme	Tehlikeli Operasyonel Atıklar
SİSTEM ODASI TPG'LERİ	Varlık Kullanımı
Altyapı Maliyeti	Toplam Kesinti
Veri Merkezi Alan % Kullanımı	Toplam Çalışma Süresi
Veri Merkezi Dolap Alanının % Kullanımı	Hizmet Süresi (Her Hizmet İçin)
Toplam Veri Merkezi Alanı	Toplam Onarım Süresi
Toplam Veri Merkezi Kabin Alanı	Bakım Maliyeti (Her Cihaz İçin)
Ortalama Veri Merkezi Raf Kullanımı	Toplam Bakım Maliyeti
Uyumluluk Göstergeleri	**SİBER GÜVENLİK TPG'LERİ**
Soğutma Oranı	Enfekte Olan İstemci Sayısı
Soğutma Kapasitesi	Denetim Kaydı
Hava Kalitesi (İnsan Sağlığı İçin)	E-posta Spam İletilerinin Yüzdesi Durdurulan / Tespit Edilen
Güç Bağlantı Durumu	Algılanan Ağ Saldırı Sayısı
CO2 Oranı	Ağdaki Virüs Sayısı
BİT Odalarının Sıcaklığı	Güvenlik Denetim Sıklığı
Donanım Sıcaklığı (Cihaz Başına)	DNS İstek Sayısı
BİT Donanım Yazılım Sürümü	DNS İsteklerinde Kötü İstek Yüzdesi
BİT TALEP TAKİP TPG'LERİ	Algılanan İstismar Sayısı
En Yüksek Ağ İşlem Hacmi Talebi	Algılanan Güvenlik Açığı Sayısı
Toplam İnternet Bant Genişliği Talebi	Tespit Edilen Sıfır Gün Güvenlik Açığı Sayısı
Genel Ağ Talebi	Acil Değişiklik Sayısı

Tablo 4.1 devamı

Güvenlik Duvarındaki TCP Paketi Sayısı	Teknik Destek Kullanılabilirliği (Saat)
Güvenlik Duvarındaki UDP Paketi Sayısı	Olaylara Müdahale Performansı
Güvenlik Duvarındaki Bağlantı Sayısı	İş Emri Talebi Performansı
Güvenlik Duvarındaki Oturum Sayısı	Müşteri Destek Performansı
Güvenlik Duvarında Yeni Oturum Oranı	Birden Çok Bileti Açık Olan Müşteri Sayısı
Güvenlik Duvarı İşlem Hacmi	Sistem Başına Toplam Bilet Sayısı
Günlük Kayıt Miktarı	Ortalama Çağrı Sayısı
HTTPS / HTTPS Trafik Miktarı	Mesai Saati Sonrası Ortalama Çağrı Sayısı
Güvenlik İhlal Olayı Yanıt Süresi	Gelen Telefon Aramalarının Ortalama Oranı
En Sık Kullanılan Kaynaklar	Telefon Aramalarının Ortalama Yanıtlanma Hızı
En Sık Kullanılan Hedefler	Aktarılan Çağrıların Yüzdesi
En Sık Kullanılan Hizmetler	Memnuniyet Anketine Katılan %
En Sık Kullanılan Uygulamalar	Red Edilen Telefon Çağrı Yüzdesi
En Sık Kullanılan Bağlantı Noktaları	SLA Süresinde Çözülen Olay %
IPS / IDS Bağlantı Hızı	SLA'e Yanıt Süresi %
En Sık Bağlantı Yapılan Ülkeler	Süresi Geçen Kayıtlar
En Çok Bant Genişliği Tüketen İstemciler	Sorun/Olay Kuyruk Büyüklüğü
Engellenen Kritik Risk Sayısı	Ortalama Sorun/Olay Kapanış Süresi
Engellenen Yüksek Risk Sayısı	Ortalama Sorun/Olay Yanıt Süresi
Gerekli Güvenlik Güncelleme Sayısı	Yanlış Atanan Olaylar %
Güvenlik Riski Puanı	Yanlış Atanan Hizmet Talepleri %
IoT TPG'LERİ	Artan Hizmet Talepleri %
Sensör Sağlığı	Tekrarlanan Olaylar %
Sensör Kullanılabilirliği	Değişiklikler Nedeniyle Planlanmamış Kesinti Yüzdesi
Sensör Erişilebilirliği	Yeniden Açılan Hizmet Talepleri %
Sensör Sayısı	Olaylara Neden Olan Değişiklik %
YARDIM MASASI TPG'LERİ	Acil Değişiklikler %
Yardım Masası Bileti Sayısı (Yıllık)	Yönlendirme Olmadan Tamamlanan Olaylar %
Toplam Yardım Masası Bileti Sayısı	Uzaktan Çözülen Olaylar %
Yardım Masası % Kullanılabilirliği	30 Günden Eski Hizmet Taleplerinin Sayısı
Yardım Masası Yanıt Performansı	Bilinen Sorun Başına Olay Sayısı

Tablo 4.1 devamı

Kök Nedeni Belirlenen Sorunlar %	BİT Dışı Toplam Güç Değerlendirmesi
Problem Tespitine Kadar Geçen Süre	Pil Şarj Seviyesi
Güvenlik Olayı Yanıt Süresi	Aydınlatmada Tüketilen Enerji
Güvenlik Olaylarından Kaynaklanan Kesinti Süresi %	Ağda Enerji Tüketimi
Başarılı Yazılım Yükseltmeleri %	Watt Başına Mhz
Acil Sürüm Düzeltme Sayısı	Watt Başına Bant Genişliği
Kullanılan Yazılım Lisansları %	Watt Başına Kapasite
Kullanıcı Memnuniyeti	Elektrik Maliyeti
Şikâyet Sayısı	Güç Kullanımı
Bir Sorunu Çözmek için Ortalama Maliyet	UPS Kayıpları
Bir Olayı Çözmenin Ortalama Maliyeti	Kritik Yük Zirve Değeri
Paydaşlardan Yardım Alma Oranı %	Jeneratör Kapasitesi
ENERJİ TPG'LERİ	İhtiyaç Duyulan Jeneratör Kapasitesi
Güç Dağıtım Sisteminin Verimliliği	Jeneratör Soğutma
Güç Tüketimi Verimliliği	Sunucu Yıllık Enerji Kullanımı
En Yüksek Enerji Talebi	Depolama Yıllık Enerji Kullanımı
Ekipman Başına Güç Değeri	Ağ Ekipmanları Yıllık Enerji Kullanımı
BİT Ekipman Güç Değeri	Veri Merkezi Ağ Yıllık Enerji Kullanımı
BİT Dışı Ekipman Güç Değeri	Toplam BİT Ekipmanı Enerji Kullanımı
Planlanan BİT Cihazlarının Güç Değeri	Toplam BİT Enerji Kullanımı
UPS Enerji Kaybı	Soğutma Enerjisi Kullanımı
UPS Akü Kapasitesi	Işık Enerjisi Kullanımı
Aydınlatma Gücü Gereksinimi	Veri Merkezi Enerji Kullanımı
Beklenen En Yüksek Kritik Yük	Kabin Başına Enerji Kullanımı
Toplam Elektrik Gücü Gereksinimi	**HİZMET/SERMAYE HARCAMALARI TPG'LERİ**
Toplam Güç Gereksinimi	Sunucu İçin Operasyonel Harcama
Soğutma Gücü Kapasitesi	Sunucu İçin Sermaye Harcaması
Sunucu Güç Kullanımı	Veri Merkezi İçin Operasyonel Harcama
Toplam Depolama Güç Değerlendirmesi	Veri Merkezi İçin Sermaye Harcaması
Toplam Sunucu Güç Değerlendirmesi	Ağ Altyapısı İçin Operasyonel Harcama
Toplam Ağ Güç Değerlendirmesi	Ağ Altyapısı Sermaye Harcaması

Tablo 4.1 devamı

Ağ Altyapısı Yıllık Sermaye Harcaması
Ağ Altyapısı Yıllık Operasyonel Harcama
Sunucu Yıllık Operasyonel Harcama
Veri Merkezi Yıllık Sermaye Harcaması
YANGIN TPG'LERİ
Yangın Söndürme Gazı Dolum Tarihi
Yangın Söndürme Gaz Basıncı
Yangın Alanı Sıcaklığı
UYGULAMA/SERVİS TPG'LERİ
Kullanılabilirlik Uygulamaları
Uygulama Performansı Göstergesi
Uygulama Yanıt Süresi
Uygulama Talebi Sayısı
Servis Kullanılabilirliği
Tamamlanan Planlanmamış Hizmet Sayısı

BÖLÜM BEŞ
AĞ SİMULASYON ARAÇLARI İLE AĞ ALTYAPI TASARIMLARININ PERFORMANSININ ÖLÇÜLMESİ

Ağ simülasyonu (benzetim), bir ağ ortamının bilgisayar üzerinde modellenerek ağın fiziksel olarak kurulumu yapılmadan nasıl çalışacağının bilgisayar üzerinde fiziksel ağ yapısına benzer şekilde test edilmesi demektir. Ağ ortamında gerçekleşebilecek olayların önceden tahmin edilebilmesine ve olası risklerle ilgili öncelik değerlendirmesine olanak sağlar.

Tasarlanan ağ altyapı bileşenleri fiziksel olarak kurulup hayata geçirilmeden önce sanal ortamda test edilmekte, böylelikle öngörülmemiş hata ve eksiklikler sanal ortamda tespit edilip tasarım güncellenebilmektedir. Burada benzetim yazılımının sistemi gerçeğe yakın bir şekilde temsil edebilmesi önemlidir. Bu yüzden amaca uygun test aracı seçilmesi son derece önemlidir. Test araçları olarak matematiksel analiz yazılımları, ağ simülasyon yazılımları, ağ emülasyon yazılımları veya gerçek ortam test yazılımları kullanılabilir. Ağ test araçları ile bir ağ modeli oluşturulması, simülasyonun çalıştırılması, sonuçların analizi ile ağ performansına ait istatistiklerin toplanması gibi işlemler yapılabilir. Bilgisayar ağları testlerinde en sık kullanılanlar Omnet++, Opnet, Ns2, Ns3, Mininet, Fnss, GNS3, Packet Tracer, EVE-NG, VIRL gibi yazılım uygulamalarıdır. En çok tercih edilen simülasyon yazılımlarının listesi EK-A'da bir tablo olarak sunulmuştur. GNS3, EVE-NG ve VIRL uygulamaları ağ emülasyon yazılımları olarak adlandırılırlar. Bu tür yazılımların diğer test araçlarından farkı gerçek cihazlarda çalışan yazılımın birebir aynısını bu yazılımlar üzerinde çalıştırarak gerçek ağ altyapısının bir kopyasını bu yazılımlar üzerinde oluşturmak mümkündür. Böylelikle emülasyon üzerinde hazırlanan kodlar ve yapılandırmalar birebir gerçek ağ altyapısına aktarılabilmektedir.

Yazılım tabanlı ağların hayata girmesi ile ağ altyapıları üzerinde programlama kullanımı ve otomasyon ihtiyaçları artmış, artmaya da devam etmektedir. Gelecekte her ağ mühendisinin bir programlama dili bilmesi gerekeceği öngörülmektedir. Ağ

cihazlarına veri göndermek, veri çekmek veya belirli otomasyonlar oluşturmak için özellikle Python dili yaygın olarak tercih edilmektedir.

Ağ yapıları üzerinde test yapılması, ağ altyapısının mevcut durumunun takibi ve altyapıdaki cihazlara yapılandırma gönderilmesi gibi işlemler için kullanılan bazı araçlar şu şekildedir: Pyats, Ansible, Genie, Netmiko, Paramiko, Jinja, Robot. Bu araçlar ile yazılan betiklerin (script) becerilerine göre ağ altyapıları üzerinde otomatikleştirilmiş işlemler gerçekleştirilebilir, problemler önceden tahmin edilebilir ve genişleyebilirlik testleri yapılabilir (Preston, 2020).

5.1 Ağ Simülasyonunda Tercih Edilen Araçlar

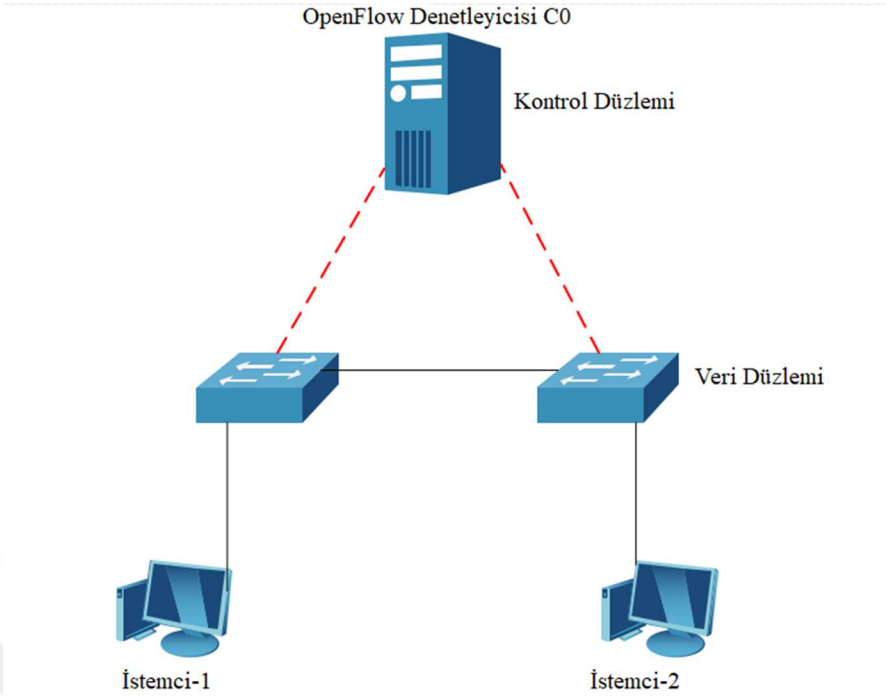
Bu çalışmada gerçekleştirilen testlerde geleceğin ağ yapıları olarak gösterilen YTA yapılarınının performansını gözlemleyebilmek amacı ile test yazılımı olarak hem YTA yapıları hem de geleneksel ağ yapılarının testinde kullanılan Mininet tercih edilmiştir. Mininet araştırma ve öğretim faaliyetlerinde kullanılmak üzeri Stanford Üniversitesinden bir grup akademisyen tarafından oluşturulmuştur (Lantz, Heller ve McKeown, 2010; Mininet, 2018). Mininet ağ emülasyon biçimidir. Emülasyonun önemli bir avantajı, herhangi bir katmandaki tüm ağ yazılımlarının “olduğu gibi” çalıştırılmasıdır. Bir benzetim ortamında ise uygulamalar ve protokol uygulamalarının simülatörde kullanılabilmeleri için simülatör ortamı içinde çalışacak şekilde özelleştirilmesi gerekir. Emülasyonun dezavantajı ise ağ büyüdükçe ve karmaşıklaştıkça emülasyonun da yavaşlayabilmesidir.

Emülasyon araçlarından biri olan Mininet, ağlara bağlanabilen hafif mantıksal düğümlerin oluşturulmasını destekleyen bir sistemdir. Bu düğümlere ağ isimuzayı (network namespace) denir. Mininet’te sanal makine teknolojisi kullanılmaz. Bunun yerine ağ isimuzayları ile tek bir dizüstü bilgisayarda çalışabilecek binden fazla düğümden oluşan ağların oluşturulmasını en az kaynak ile sağlar. Mininet’te denetleyici ve kenar anahtarlar kök isimuzayı (root namespace) üzerinde çalışırken sanal istemciler network namespace üzerinde çalışırlar. Mininet başlangıçta YTA için bir test ortamı olarak geliştirilirken aynı zamanda geleneksel ağ testleri için de

kullanılabilmektedir. Python desteği vardır ve Python uygulama programlama arayüzü (Application Programming Interface: API) kullanılarak özelleştirilmiş senaryolar oluşturmak mümkündür (Mininet, 2018).

Mininet uç noktaların donanım kulanımını birbirine göre izole ederek çalıştırır. Yani bir uç noktadaki işlemci kullanımı diğer noktayı etkilemez. Mininet kenar anahtar olarak Linux köprüsü (bridge) veya Open Vswitch (Open Vswitch, 2020) kullanmaktadır. Ayrı bir denetleyici kullanılmadığı sürece Mininet varsayılan olarak "OpenFlow Stanford Reference" denetleyicisini kullanır. Bu denetleyici ağ modelinde 16 kenar anahtara kadar destekler, fakat bu sayı çeşitli yöntemler ile teoride 4096 sayısına kadar yükseltilebilir. Uç noktalarda ve kenar anahtarlarda Linux işletim sistemi çekirdeği (kernel) çalışmakta ve ağ işlemleri bu Linux kerneli üzerinden gerçekleşmektedir (Mininet, 2018).

Simülasyonda denetleyici olarak Pox denetleyicisi tercih edilmiştir. Pox, Python tabanlı OpenFlow denetleyicisidir. Pox, Nox denetleyicisi temel alınarak tasarlanmıştır. Bu denetleyicinin varsayılan OpenFlow denetleyicisinden önemli farklarından bir tanesi hem kendi üzerinden hem de kenar anahtarlar üzerinden gönderilen paket istatistiklerinin elde edilebilmesi ve denetleyici üzerinden geçen trafiğin bir kopyasının kaydedilebilmesidir. Bu özellik ağ mühendisi ve araştırmacısına trafiği çeşitli yöntemlerle ve senaryolar altında analiz etme olanağı verir.



Şekil 5.1 Örnek Mininet YTA topolojisi

YTA ağ yapılarında geleneksel ağlardan farklı olarak kontrol paneli ile veri paneli Şekil 5.1'de gösterildiği gibi birbirinden ayrı tutulmuştur. Kontrol paneli, yönlendirme protokolleri ve diğer protokollerden topladığı bilgiler ile yönlendirme tablosuna bilgi sağlar. Veri katmanı, kontrol katmanından alınan bilgilerle oluşturulan yönlendirme tablosuna göre bir arayüzden diğer arayüze yönlendirme işlemlerini gerçekleştirir. Geleneksel ağlarda kontrol katmanı cihazlarla bütünleşik iken YTA yapılarında kontrol katmanı merkezi bir yazılım olarak konumlanır ve tüm cihazlar bu yazılım kontrol katmanından yönetilir. Ağ yapısındaki cihazlar üzerinde sadece veri paneli bulunmaktadır ve bu cihazlar kontrol paneli için YTA denetleyicisi ile iletişim halindedir. Cihazlar ile YTA denetleyicisi arasındaki bu iletişim OpenFlow protokolü ile sağlanmaktadır. Mininet de YTA ağlarının bu genel özelliklerine uygun çalışmaktadır.

Mininet'in tercih edilmesinin başlıca sebepleri aşağıdaki gibidir (Mininet GitHub, 2018):

- Hızlıdır,
- Özel topolojiler oluşturulabilir,
- Gerçek programların çalıştırılmasına olanak sağlar, örneğin Wireshark, tcpdump, vb.,
- OpenFlow protokolünü destekler,
- Kullanımı kolaydır,
- Açık kaynaktır.

Simülasyonların geliştirilmesinde tercih edilen diğer bir araç ise FNSS (Fast Network Simulation Setup)'dir. Ağ mühendisi ve araştırmacıları tarafından tercih edilen bir araçtır. Başlı başına bir simülasyon aracıdır. Python desteği vardır ve Mininet ile bütünleşik olarak çalışabilmektedir. Bu tez çalışması kapsamında tercih edilmesinin sebepleri şu şekildedir:

- Python desteğinin olması,
- Mininet ile uyumlu çalışabilmesi,
- Üzerinde kolaylıkla hiyerarşik topolojilerin oluşturulmasına olanak vermesi ve oluşturulan topolojilerin Mininet'e aktarılabilmesi
- Oluşturulan topolojiler üzerindeki bağlantılarda bant genişliği, gecikme gibi özelliklerin kolayca ayarlanmasına izin vermesidir.

Oluşturulan simülasyonda test aşamasında kullanılan araçlar aşağıdaki gibidir:

- Ping – Erişim test aracı (Ping, 2020).
- Wireshark – Paket yakalama aracı (Wireshark, 2020).
- TShark – Paket yakalama aracı (Tshark, 2020).
- Netstat – Ağ istatistik görüntüleme aracı (Netsat, 2020).
- Vmstat – Sistem izleme aracı (Vmstat, 2020).
- Iperf –Veri iletim test aracı (Iperf, 2020).
- ab(apache bench) – Web sunucu yük test aracı (Ab, 2020).
- Python ile yazılmış basit HTTP sunucu (Mininet GitHub, 2018)

- Curl – Metin tabanlı internet tarayıcısı (Curl, 2020).

Test için kullanılan bu araçların en önemli özelliği gerçek ortam testlerinde de kullanılan yazılımlar olmasıdır. Bu araçlar özellikle Linux tabanlı her işletim sisteminde çalışabilmektedir. Mininet'e üstünlük katan başlıca bu özelliğidir. Gerçek ortam testlerinde kullanılan araçların istenilen sürümü simülasyon ortamına özelleştirilmeden kullanılmasıdır. Bu araçlar Mininet ile birebir ilişkili araçlar değildir. Bu araştırma kapsamında kullanılması öngörülen Linux tabanlı araçlardır. Mininet Linux kernel yapısına sahip olması sebebiyle bu araçların simülasyon içinde kullanılmasına olanak sağlamaktadır. Farklı Linux tabanlı yazılım araçlarını da Mininet simülasyonları ile ilişkilendirmek mümkündür.

Test ortamı olarak birçok YTA denetleyicisini bünyesinde hazır bulunduran ve Mininet'in içinde çalışır bulunduğu SDNHUB (SDNHUB, 2020) sanal ortamı tercih edilmiştir. SDNHUB sanal ortamının çalıştırılması için VMware Workstation (Vmware Workstation, 2020) kullanılmıştır. Sanal ortama erişmek için ise Putty (Putty, 2020) istemcisi ve XmingServer (XmingServer, 2020) sunucusu kullanılmıştır. Test ortamının kurulması, testlerin yapılması ve değerlendirilmesi için yararlanılan kaynaklar ve kullanılan yazılımlar aşağıda listelenmiştir:

- SDNHUB (2020);
- Linkletter (2015);
- Sood ve Sharma (2014);
- POX Wiki (2018);
- Mininet GitHub (2018);
- Mininet (2018);
- O'Briain (2017);
- Mirantis (2016);
- Dordal (2016);
- Masset ve Taburiaux (2018);
- Ke (2020);
- KNET Solution (2020);

- Bholebawa ve Dalal (2016).

5.2 Simülasyon Testinin Amacı

Yapılan test ve araştırmalarda ana amaç büyük ağ yapılarında SÖA varsayılan ağ geçitlerinin konumlandırıldığı katmana göre genel ağ performansına etkisinin incelenmesi ve bu etkilerin bu çalışma içinde belirlenmiş TPG'leri ile ölçülmesidir. Bu ana amacın altında aşağıdaki gibi iki alt amaç belirlenmiştir:

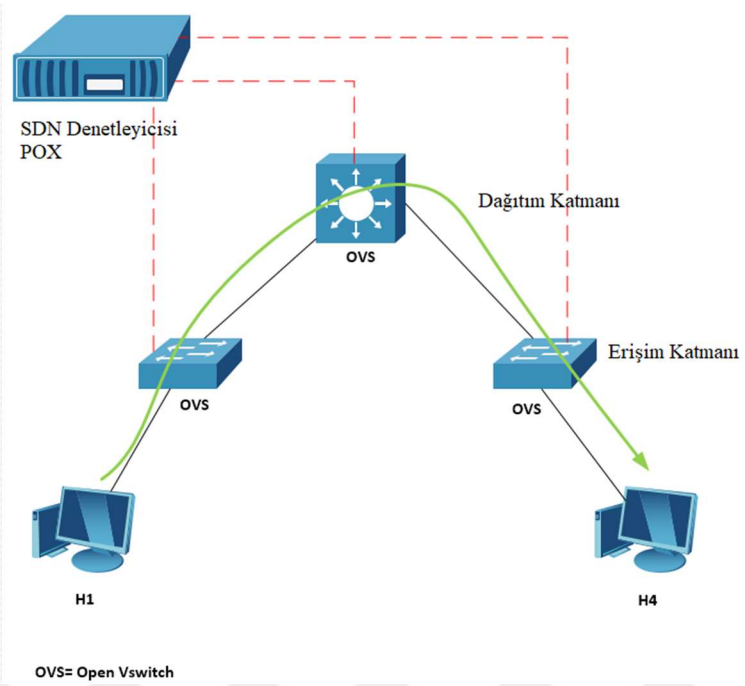
1. SÖA Katman 3 varsayılan ağ geçitlerinin dağıtım katmanında veya çekirdek katmanında konumlandırılması ile ilgili iki farklı topoloji üzerinde iki ayrı senaryo oluşturulması ve senaryolardaki ağ iletişim performans farklarının araştırılması;
2. YTA ağ yapılarındaki SÖA Katman 3 ara yüzlerinin dağıtım katmanında veya çekirdek katmanda konumlandırılmasına göre ağ iletişimindeki performans farklarının araştırılması.

Bu iki amacı gerçekleştirmek için Mininet üzerinde SÖA Katman 3 varsayılan ağ geçitlerinin dağıtım katmanında ve çekirdek katmanında sonlandırıldığı iki ayrı topoloji hazırlanarak hazırlanan test prosedürleri doğrultusunda test edilmesi amaçlanmıştır.

5.3 Simülasyon Test Ortamı ve Parametreleri

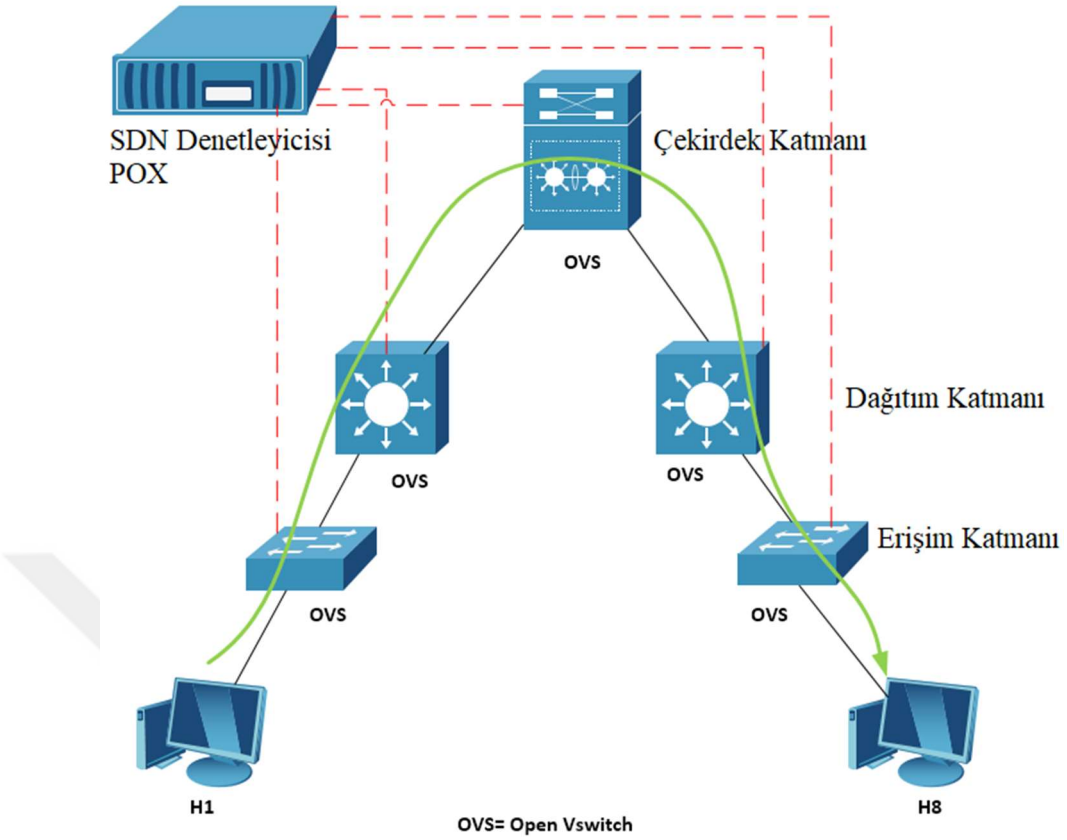
Test ortamı bir dizüstü bilgisayara kurulan VMware Workstation üzerinde hazırlanmıştır. VMware üzerine SDNHUB sanal ortamı oluşturulmuştur. Bu ortamı çalıştırmak için 2.5 GB RAM, 2 çekirdek CPU ve 20 GB disk alanı ayrılmıştır. SDNHUB platformu Ubuntu Linux işletim sistemi üzerinde çalışmaktadır. Bu ortamın üzerine FNSS, TShark, Vmstat paketleri kurularak, Mininet ile bütünleşik olarak çalışabilecek duruma getirilmiştir. 5.2'de belirtilen problemleri araştırmak için iki katmanlı ve üç katmanlı olarak FNSS ile bütünleşik çalışan daha önce yapılmış çalışmalar temel alınmıştır (van Adrichem, Doerr ve Kuipers, 2014; Ke, 2016).

Bahsedilen çalışmalardakine benzer şekilde Mininet ve FNSS Python API aracılığı ile kullanılmış, gerekli değişikliklerle bu tez çalışmasına uygun hale getirilmiştir.



Şekil 5.2 İki katmanlı test topolojisi

Şekil 5.2'deki topoloji, Katman 3 varsayılan ağ geçitlerinin dağıtım katmanında sonlandırılması ile ağ performansına etkisini araştırmak için kullanılmıştır. Trafik Şekil 5.2 üzerinde gösterilen yeşil ok yönünde ilerlemektedir.



Şekil 5.3 Üç katmanlı test topolojisi

Şekil 5.3'teki topoloji, Katman 3 varsayılan ağ geçitlerinin çekirdek katmanında sonlandırılması ile ağ performansına etkisini araştırmak için kullanılmıştır ve trafik şekil üzerinde gösterilen yeşil ok yönünde ilerlemektedir.

Topolojiler FNSS yardımı ile oluşturulup Python API yardımı ile Mininet ortamına aktarılarak çalışır hale getirilmiştir. Çalışır hale getirilen test ortamında Tablo 5.1'de parametreleri verilmiş olan Test 1, 2, 3 simülasyonları oynatılmıştır.

Tablo 5.1 Simülasyon test parametreleri

Test 1	**Test 2**	**Test 3**
iperf -c Hedef_Ip -i 1 -P 10 -t 300	**iperf -c -u Hedef_Ip -i 1 -b 10m -P 10 -t 300**	**ab -n 5000 http://Hedef_Ip**
TCP Performans Testi	UDP Performans Testi	HTTP Sunucu Yük Testi
i 1 : 1 saniyelik aralıklarla raporlanacak	**i 1 :** 1 saniyelik aralıklarla raporlanacak	**n 5000 :** 5000 istek sayısı
P 10 : Aynı anda 10 ayrı işlem başlayacak	**b 10m :** Bant genişliği 10 Mbit olarak sınırlanacak	
t 300 : Test 5 dakika sürecek	**P 10 :** Aynı anda 10 ayrı işlem başlayacak	
	t 300 : Test 5 dakika sürecek	
c : İstemci	**u :** Udp protokolü	**n :** İstek sayısı
i : Raporlama aralığı	**c :** İstemci	
P : Aynı anda açılacak bağlantı sayısı	**i :** Raporlama aralığı	
t : zaman	**b :** Bant genişliği	
	P : Aynı anda açılacak bağlantı sayısı	
	t : zaman	
YTA Denetleyici Ayarları		
./pox.py forwarding.13_learning –fakeways=10.0.0.1,192.168.0.1		
FNSS Bağlantı Ayarları		
Bağlantı Kapasitesi	**Bağlantı Gecikmesi**	**Bağlantı Tampon Belleği**
10 Mbit	2 ms	50 Paket

Her test Mininet üzerinde ayrı ayrı çalıştırılmış ve her testin bitişinde Mininet üzerindeki kalıntıları temizlenerek her test yeniden başlatılmıştır. Test 1 ve Test 2'de 3.49 GB veri, birbirinden bağımsız 10 bağlantı ile ağ üzerinde yoğunluk oluşturacak şekilde transfer edilmiştir. 10 bağlantı toplamda anlık 100 MBit'lik veri transferi başlatmaktadır. Test 1'de iperf aracı ile TCP performans testi, Test 2 iperf aracı ile UDP performans testi ve Test 3'te ab aracı ile HTTP sunucu stres ve yük testi gerçek ağ ortamlarında kullanılan araçlar ile gerçekleştirilmiştir. Test 3'te HTTP sunucu oluşturulması için Mininet üzerinde yazılmış Python kodu kullanılmıştır. Mininet, üzerinde çalışan istemcilere sınırlı kaynak ayırdığı ve belirli eşikler aşıldığındığında cevap veremez duruma geldiği için farklı testler sonucunda test süreleri TCP ve UDP testleri için 5 dakika ve HTTP testi için tek bağlantı üzerinden 5000 istek olarak belirlenmiştir. Testler iki katmanlı test senaryosunda H1 istemcisi ile H4 sunucusu arasında, üç katmanlı test senaryosunda ise H1 istemcisi ile H8 sunucusu arasında yapılmıştır. H4 ve H8 test sunucusu olarak, H1 ise test istemcisi olarak kullanılmıştır. Test esnasında cihazlar arası bağlantılar hakkında bilgi toplamak için ping, netstat ve curl kullanılmıştır. Test sırasındaki trafiği kaydetmek için TShark ve Wireshark kullanılırken, işlemci ve bellek kullanımını takip etmek için ise Vmstat kullanılmıştır. Denetleyici verilerini almak için Mininet ana makinesi üzerinden ve istemci-sunucu trafiğini kaydetmek için H1 ve H4/H8 üzerinden trafik örnekleri kaydedilmiştir.

Denetleyici olarak POX "./pox.py forwarding.l3_learning --fakeways=10.0.0.1,192.168.0.1" parametreleri ile çalıştırılmıştır. Varsayılan olarak denetleyici kenar anahtarları Katman 2 olarak yapılandırılmaktadır. Fakat ticari YTA ürünlerinde yeni yaklaşım olarak tüm ağ altyapısı Katman 3 olarak çalıştırılmaktadır. Buradaki amaç STP protokolünün sebep olduğu problemlere veya yayın fırtınası (broadcast storm) gibi problemlere çözüm oluşturmak için kullanılan tasarım yaklaşımıdır. Bu çalışmada ticari ürünlerde olan bu yaklaşım da simülasyon modeline dâhil edilerek kullanımdaki YTA altyapılarına benzer gerçekçi bir simülasyon ortamı oluşturulması amaçlanmıştır. Mininet'in varsayılan özelliklerinde yönlendirici test ortamı olmadığı ve ticari tasarım yaklaşımını daha iyi simüle etmek için POX denetleyicisi "forwarding.l3_learning" parametresi ile çalıştırılmıştır. Bu parametre kullanımı cihazlara tam olarak yönlendirici özelliği vermese de Katman 2 kenar anahtar ile Katman 3 arası bir kenar anahtar elde edilmesini sağlamaktadır. Böylelikle kenar anahtarlar MAC bilgilerinin yanında IP adreslerini de çözümleyebilmektedir. Bu parametre özellikle ARP (Address Resolution Protocol - Adres Çözümleme Protokolü) istek ve cevaplarını araştırmak için kullanılmaktadır (POX Wiki, 2018). İletişimin ağ geçitleri üzerinden sağlanması için "fakeways=10.0.0.1,192.168.0.1" parametresi ile sahte varsayılan ağ geçitleri oluşturularak Katman 3 iletişiminin sağlanması amaçlanmıştır. Mininet'te varsayılan olarak yönlendirici özelliği bulunmadığı için bu özellik kullanılmıştır.

5.4 Simülasyon Sonuçları

Simülasyondan toplanan veriler Wireshark yazılımı üzerinde üzerinde incelenmiştir. Elde edilen veriler tablo ve grafikler haline getirilerek, Tablo 4.1 TPG tablosundaki göstergeler ile eşleşenler özelinde yorumlanmaya çalışılmıştır. Elde edilen sonuçlar üzerinde bir çıkarım yaparken TCP, UDP vb. protokollerinin çalışma prensipleri, gerçek ağ sistemlerinde ortaya çıkan sorunlar ve bu sorunlar için geçerli çözüm yaklaşımları göz önünde bulundurulmuş ve anlamlı analiz sonuçları elde edilmesi amaçlanmıştır.

Tablo 5.2 İki katmanlı topoloji simülasyonu test 2 sonuç değerlendirme tablosu

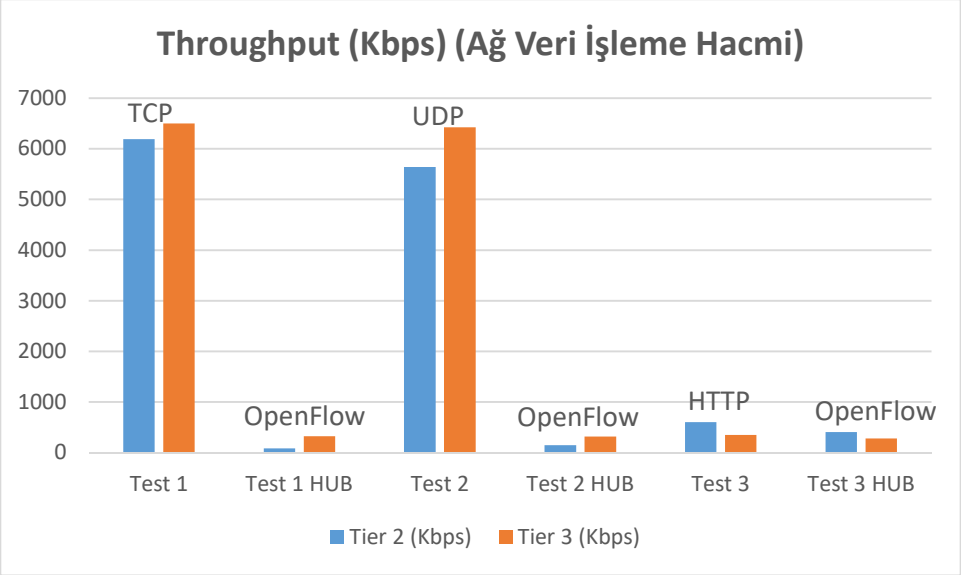
Oturum No	Transfer	Bant Genişliği	Dalga Bozulumu	Kayıp Toplam Veri Paketi	Bozuk Veri Paketi
1	27,3 Mbytes	761 Kbits/sn	1,515 ms	92%	4
2	64,1 Mbytes	1790 Kbits/sn	0,557 ms	82%	2
3	26,4 Mbytes	735 Kbits/sn	0,796 ms	93%	3
4	39,7 Mbytes	1110 Kbits/sn	0,218 ms	89%	3
5	39,0 Mbytes	1090 Kbits/sn	0,719 ms	89%	4
6	30,5 Mbytes	851 Kbits/sn	0,552 ms	91%	2
7	27,5 Mbytes	766 Kbits/sn	0,982 ms	92%	2
8	26,7 Mbytes	744 Kbits/sn	0,526 ms	93%	1
9	34,3 Mbytes	955 Kbits/sn	1,095 ms	90%	2
10	32,8 Mbytes	913 Kbits/sn	1,098 ms	91%	4
Ortalama		**971,5 Kbits/sn**	**0.8058 ms**	**90%**	**2,7**
Min. Boş İşlemci	**80/100**				
Boş Bellek	**67696/2566016 Kbit**				

Tablo 5.3 Üç katmanlı topoloji simülasyonu test 2 sonuç değerlendirme tablosu

Oturum No	Transfer	Bant Genişliği	Dalga Bozulumu	Kayıp Toplam Veri Paketi	Bozuk Veri Paketi
1	33,2 Mbytes	926 Kbits/sn	0,526 ms	91%	11
2	34,1 Mbytes	951 Kbits/sn	0,668 ms	90%	33
3	35,9 Mbytes	1.00 Mbits/sn	0,305 ms	90%	31
4	35,6 Mbytes	993 Kbits/sn	0,998 ms	90%	29
5	36,2 Mbytes	1.01 Mbits/sn	0,530 ms	90%	37
6	35,6 Mbytes	992 Kbits/sn	1,366 ms	90%	48
7	33,6 Mbytes	937 Kbits/sn	1,166 ms	91%	30
8	34,0 Mbytes	948 Kbits/sn	1,340 ms	90%	19
9	34,8 Mbytes	969 Kbits/sn	0,772 ms	90%	22
10	34,1 Mbytes	950 Kbits/sn	0,692 ms	90%	20
Ortalama		**967,6 Kbits/sn**	**0,8363 ms**	**90%**	**28**
Min. Boş İşlemci	**40/100**				
Boş Bellek	**67204/2566016 Kbit**				

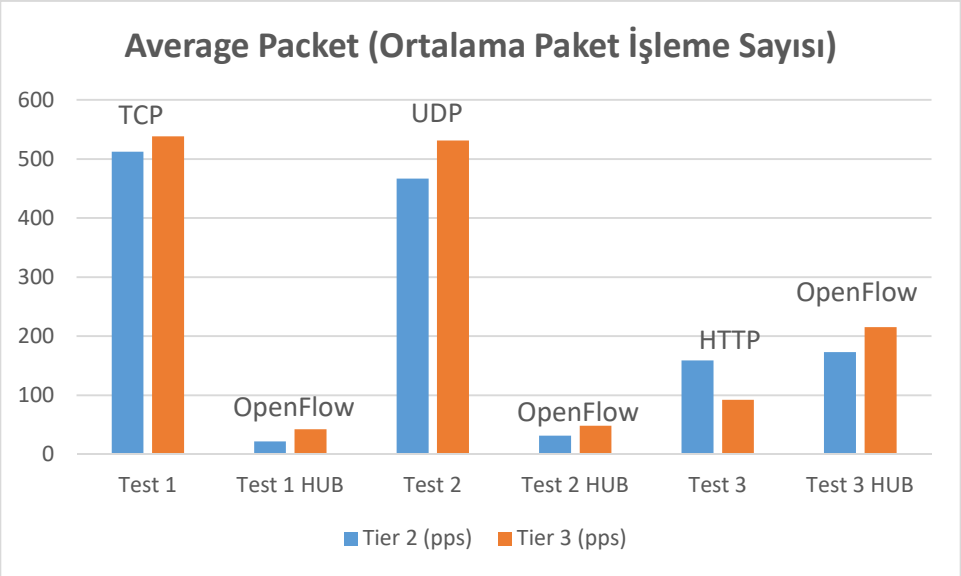
Test 2 simülasyon sonuçlarından oluşturulan Tablo 5.2 ve Tablo 5.3 sonuçları değerlendirildiğinde, iki katmanlı topoloji sonuçlarının yani SÖA varsayılan ağ geçitlerinin dağıtım katmanında sonlandırıldığı simülasyonun üç katmanlı topoloji yani SÖA varsayılan ağ geçitlerinin çekirdek katmanında sonlandırıldığı simülasyon sonuçlarına göre daha iyi performans sonuçları verdiği gözlenmiştir. Tablolara göre varsayılan ağ geçitleri dağıtım katmanında sonlandırıldığında dalga bozulumu (jitter) TPG değerinin ortalaması, çekirdek katmanında sonlandırıldığı test sonuçlarına göre daha düşüktür. Veri iletimindeki dalga bozulumu olarak tanımlanan jitter TPG'si özellikle ses ve görüntü ağları için kritik öneme sahiptir. Bu değerin yüksek olması ses kesintilerine, görüntü kayıplarına ve sonuç olarak bozuk bir veri akışına sebep olmaktadır. "Kayıp Toplam Veri Paketi %" TPG'si ağ üzerinde kayıp paket yüzdesini ölçmek için kullanılan göstergedir. Bu gösterge test sonuçları üzerinde değerlendirildiğinde iki topoloji simülasyonunda da kayıp oranının çok yüksek olduğu

görülmüştür. Verimli çalışan bir ağ altyapısında %3'ten yukarı paket kayıp oranı beklenmez. Fakat simülasyon sonuçlarındaki kayıp oranlarının %90'lara kadar yükselmesinin sebebinin test ortamındaki bir problemden mi kaynaklandığı veya testte kullanılan paket yoğunluğundan mı kaynaklandığını anlamak için ayrı bir test yapılmıştır. Bu testte tek bağlantı üzerinden Test 2 tekrar edilmiş ve sonucunda kayıp oranının %7'ye düştüğü görülmüş ve kayıp oranının %90 olmasının sebebi test yoğunluğunun protokoller ve istemci işlemcileri üzerinde oluşturduğu etki, protokollerin çalışma prensipleri ve Mininet'in her istemci için ayırdığı sınırlı kaynaklar göz önünde bulundurularak değerlendirildiğinde beklenen bir durum olduğu ve simülasyon ortamından kaynaklanmadığı sonucuna varılmıştır. Ayrıca bu TPG için her iki topoloji sonuçlarındaki ortalama kayıp oranının %90 olduğu görülmüş ve kayıp oranlarının aynı test için farklı topolojilerde değişim göstermediği görülmüştür. Düzensiz sıralı paket geldiğinde oluşan "Bozuk Veri Paketi" paketlerinin iki topoloji arasındaki ortalama değerleri incelendiğinde ağ geçitlerinin dağıtım katmanında sonlandırıldığında paketlerin daha sıralı geldiği görülmüş ve daha az paket sıra bozulumu görülmüştür. Test istemcilerinin işlemci kullanımları incelendiğinde ise iki katmanlı topolojide yer alan istemcinin daha az işlemci kullandığı gözlemlenmiştir. Bellek kullanımları ise hemen hemen iki topoloji testinde de aynı seviyelerdedir.



Şekil 5.4 Ağ Veri işleme hacmi değerlendirme sonuçları

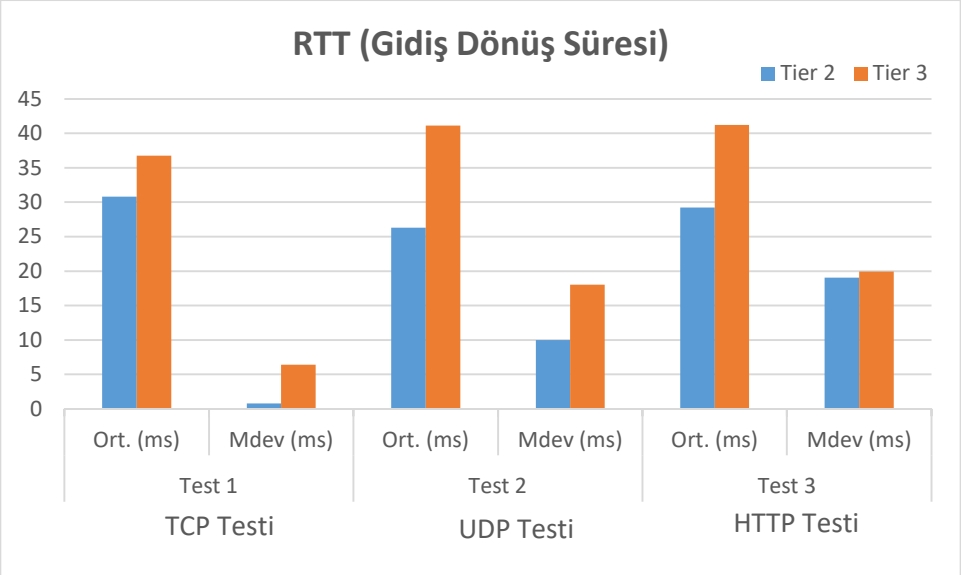
Şekil 5.4'te, birim zamanda işlenen veriye karşılık gelen ağ işlem hacmi (throughput) TPG için değerlendirme sonuçları yer almaktadır. Testlerdeki "Tier 2", iki katmanlı topoloji testine yani SÖA varsayılan ağ geçitlerinin hiyerarşik tasarımda dağıtım katmanında sonlandırılmasına karşılık gelirken "Tier 3", üç katmanlı topoloji testine SÖA varsayılan ağ geçitlerinin hiyerarşik tasarımda çekirdek katmanında sonlandırılmasına karşılık gelmektedir. Grafikte "HUB" olarak belirtilen sütunlar OpenFlow denetleyicisi verilerine karşılık gelmektedir. "Throughput" ağ altyapı performans ölçümlerinde kullanılan en önemli performans göstergelerinden biridir. Test 1 ve Test 2'de dağıtım katmanı ve çekirdek katmanı testleri arasında birim zamanda işlenen veri farkı en fazla %14 fark ile Katman 3'ün daha fazla veri işlediği gözlemlenmiştir. Test 3'te ise tam tersine en fazla %71 fark ile Katman 2 testinde daha fazla veri işleme değerlerine ulaştığı görülmüştür.



Şekil 5.5 Birim zamanda işlenilen ortalama paket sayısı

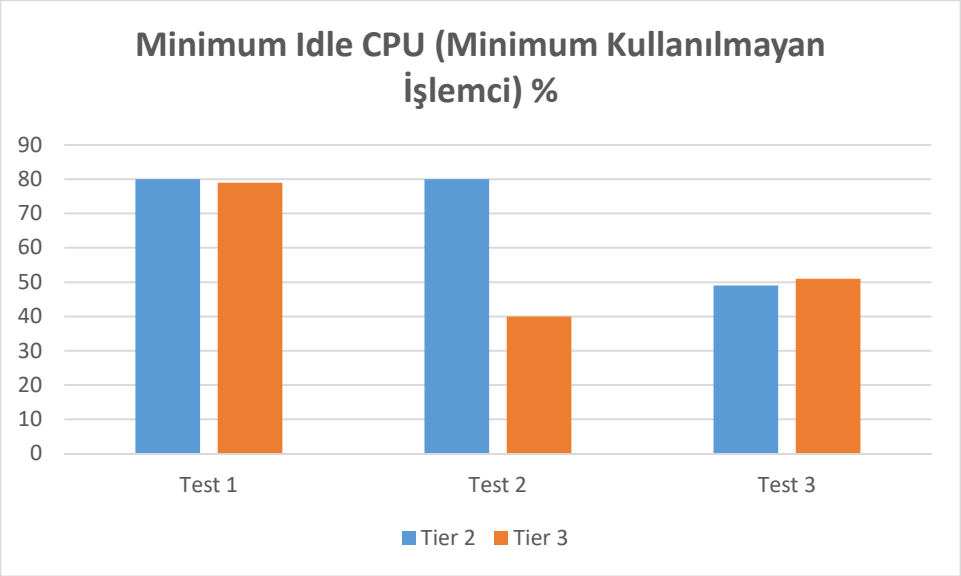
Şekil 5.5'te simülasyonda gerçekleştiren testlerdeki istemcilerin saniye başına ortalama işleyebildikleri paket sayıları gösterilmektedir. Grafik, Şekil 5.4'teki "throughput" grafiği ile benzerdir. Test 1 ve 2'de ağ geçitlerinin çekirdek katmanında sonlandırıldığında %14'e kadar daha çok paket işlenebildiği görülürken, Test 3'te tam tersi olarak dağıtım katmanında sonlandırıldığında %71'e kadar daha çok paket işlendiği gözlemlenmiştir. Ayrıca Test 3'te çekirdek katmanında sonlandırıldığında daha az paket işlendiği sonucu elde edilmesine rağmen test denetleyicisinin %25 daha

fazla paket işlediği gözlemlenmiştir. Grafik genel olarak değerlendirildiğinde çekirdek katmanı testlerinde Pox denetleyicisi üzerinde daha fazla paket işlendiği ve daha fazla yük oluştuğu gözlemlenmiştir.



Şekil 5.6 Simülasyon RTT (Gidiş dönüş süresi) değerlendirme sonuçları

Bir paketin hedefe gidiş-dönüş süresi olarak tanımlanan RTT (Gidiş Dönüş Süresi) TPG grafiği değerlendirildiğinde Şekil 5.6'da görüldüğü gibi iki katmanlı topolojinin ortalama RTT sürelerinin ve standart sapma değerlerinin daha düşük olduğu gözlemlenmiştir. Ayrıntılı incelendiğinde iki katmanlı topoloji testlerinde RTT değerlerinin daha düşük olduğu gözlemlenmiştir. Bu beklenilen bir sonuçtur. Çünkü trafik ne kadar fazla cihazdan geçerse her geçtiği cihazda oluşan gecikmelerden dolayı RTT değeri artacaktır. Bu bilgiden yola çıkıldığında dağıtım katmanında yapılan testlerin çekirdek katmanı üzerindeki testlere göre daha düşük RTT değerine sahip olacağı teste gerek olmadan tahmin edilebilmektedir. O yüzden Şekil 5.6'daki test sonuçlarında asıl odaklanılacak nokta RTT değişkenliğini ifade eden "Mdev" standart sapma sonuçlarıdır ve grafikte de görüldüğü gibi iki katmanlı topolojiye ait testlerde Mdev değerleri daha düşüktür. Test sonuçlarına göre varsayılan ağ geçitlerinin 2 katmanlı yapıda yani dağıtım katmanında sonlandırılması ile ağ üzerinde gerçekleşen iletişimde daha düşük gecikmeler olacağı sonucuna varılmıştır. Ayrıca elde edilen RTT göstergesinden yola çıkılarak noktadan noktaya gecikme veya bant genişliği gibi göstergeler de hesaplanabilmektedir (Wang, Zhou ve Li, 2004).



Şekil 5.7 Sunucu işlemcisinin kullanılmama durumu

Testlerde kullanılan istemcilerin boştaki işlemci kullanım verilerinin yer aldığı Şekil 5.7 incelenerek H4 ve H8 iperf sunucusu görevindeki istemcilerin işlemci kullanım oranları değerlendirildiğinde, Test 1 ve Test 3'te her iki topolojinin kullanımları birbirine yakın iken Test 2'de yarıya yarıya kullanım farkı oluşmuş ve üç katmanlı topolojide işlemcinin daha yoğun kullanılarak kaynak tükettiği gözlemlenmiştir. Normal şartlarda UDP protokolünün işlemciye TCP'ye göre daha az etki etmesi beklenmektedir. Fakat ortamda yoğun paket kaybı oluşuyorsa istemciler üzerinde artan kayıp paket işleme yükü ile bu durum tersine dönebilmektedir (Gu ve Grossman, 2005). UDP prokolü soket üzerinden bağlantı sağlamaktadır ve çalıştığı cihaz üzerinde yük işleyebileceğinden daha fazla geldiği zaman tampon bölge (buffer) kullanımı artmaktadır. Tampon bölgenin alanı azaldıkça işlemci kullanımı artmaktadır. Sunucu tarafında kayıp paket oranı arttıkça, istemci tarafından gönderilen paketlerin yeniden gönderme isteklerini işlemek için işlemci kullanım oranı artarken istemci tarafında kayıp paketlerin yeniden istenme taleplerinden dolayı tampon bölge kullanımı artmasıyla sunucu işlemci kullanımı artmaktadır (Tkaczewski, 2016). İşlemci üzerindeki yük, kayıp oranının artması, iletim hattının yüksek hızlı ve uzun mesafeli olması ile daha da yüksek seviyede artmaktadır. (Gu ve Grossman, 2005).

Bu bilgilerden yola çıkarak Test 2'de yapılan 2 farklı topolojideki UDP testlerinde işlemci kullanım oranlarının birbirinden farklı olması, iletim ortamında yoğun paket

kaybı yaşanmasından dolayı üç katmanlı topolojide iletim mesafesinin daha uzun olması, daha çok cihazdan geçmesi, RTT değerlerinin daha yüksek olması ile istemci tarafından gelen kayıp paketlerin yeniden istek taleplerinin işlenme süresinin artması ile işlemci kullanımının iki katmanlı topolojiye göre daha fazla olduğu sonucuna varılmıştır.

Tablo 5.4 HTTP yük testi sonuçları

	Toplam İstek Sayısı	Testler İçin Geçen Süre	Saniye Başına İstek	Talep Başına Süre	Aktarım Hızı (Alınan)	Bağlantı (mean[+/-sd])	Bekleme (mean[+/-sd])
Tier 2	5000	415,838 sn	12,02 [t/sn](mean)	83,168 [ms](mean)	66,52 [Kbytes/sn]	55 20,9 ms	25 7,4 ms
Tier 3	5000	618,274 sn	8,09 [t/sn](mean)	123,655 [ms](mean)	46,87 [Kbytes/sn]	84 12,4 ms	34 3,1 ms

HTTP yük testi olan Test 3 sonuçlarından Tablo 5.4 elde edilmiştir. İki katmanlı topolojinin testinde, 5000 isteğin tamamlanması 415,84 saniye (sn) sürerken tek bir isteğe cevap verilmesi ortalama 83,17 milisaniye (ms) zaman almıştır. Saniyede ortalama 12,02 isteğe cevap verilebilmiş ve transfer hızı 66,52 Kbytes/sn değerine ulaşmıştır. Ayrıca bu topolojide sunucuya bağlantı kurma ortalama 55 ms sürerken, sunucuya gönderilen istekle alınan yanıt arasında geçen süre ortalama 25 ms olarak ölçülmüştür. Bu göstergeler üç katmanlı topoloji için değerlendirildiğinde daha düşük performans ölçümleri elde edildiği görülmüştür.

Genel olarak simülasyon sonuçları değerlendirildiğinde çalışmanın tasarım kısmında da bahis edilen "SÖA'ların varsayılan ağ geçitlerinin çekirdek katmanında yapılandırmak yerine dağıtım katmanı tercih edildiğinde ağ altyapılarının iletişim performansına olumlu etki eder" savı, ağ veri işleme hacmi ve ortalama paket işleme sayısı TPG'leri için uyuşmazken özellikle ağ performansı değerlendirilmelerinde önemli bir gösterge olan RTT TPG'ne göre ve gerçek ortam http veri kullanımını simüle etmeyi amaçlayan http yük testleri sonuçlarına göre kısmi olarak uyuşmaktadır. Bazı TPG'lerin Test 1 ve 2'de dağıtım katmanında, Test 3'te ise çekirdek katmanında daha iyi performans değerlerine ulaşılmasının sebepleri protokollerin topoloji farklılıklarına göre performans değerlendirmeleri gibi yeni çalışma konuları olarak incelenebilir. Bu tasarım şeklinin büyük ağ altyapıları için uyarlandığında sağlayabileceği performans iyileştirmelerinin ne düzeyde olacağının isabetli bir şekilde tahmini için, daha detaylı simülasyon modelleri geliştirilmesi ve daha kapsamlı testlerin yapılması gerektiği kaçınılmazdır.

BÖLÜM ALTI
SONUÇ

Bir ağ altyapısının sürdürülebilirliği ve verimliliğini sağlamak için proje ve tasarım aşaması dikkatli hazırlanmalıdır. Ağ altyapıları üzerinde kullanıcı memnuniyeti sağlayabilmek için sadece iyi tasarım yeterli değildir. Kurulumundan yönetimine kadar tüm aşamalar bir bütün olarak değerlendirilmeli ve takip edilmelidir. İlgili proje tasarımları ve cihaz yapılandırmaları yeterli bilgi ve tecrübeye sahip mühendisler tarafından yapılmalıdır. Büyük ağ yapılarında kullanılan kurumsal ağ cihazları tak çalıştır şeklinde verimli çalışmayacaktır. Bu yüzden projelendirme kısmında ağ üzerinde verilecek hizmetlere göre ihtiyaç duyulan erişilebilirlik ve süreklilik seviyeleri dikkate alınmalıdır. Bu faktörler göz önünde bulundurulmadan tasarlanan ağ altyapıları yatırımlarının karşılığını veremeyecek, iş, para ve zaman kaybına sebep olacaktır.

Bu tez çalışmasında öncelikle güncel BT standartları, ağ altyapı tasarımları ve bunların bileşenlerine ait teknoloji ve kavramların tanımları sunularak, modern iletişim ağlarının performanslarının ölçümünde kullanılabilecek göstergeler BİT süreç ve fonksiyonları bazında kategorilere ayrılarak belirlenmiştir. Çalışma kapsamında gerçekleştirilen iki örnek ağ topolojisine ait ve üç farklı senaryo altındaki performans testlerine ait simülasyon modelleri öncelikle parametreleriyle paylaşılmış, sonrasında elde edilen simülasyon test sonuçları tanımlanan başlıca temel performans göstergeleri ışığında yorumlanmıştır.

Simülasyon sonuçlarında da görüldüğü gibi aynı tip cihazların kullanıldığı ağlar üzerindeki tasarımsal farklılıklar ağ altyapı performansını etkilemektedir. Bu tasarımsal etkiler 20 kullanıcıya hizmet veren küçük ağlar üzerinde verilen hizmet bazında fark edilebilir şekilde hissedilmesede onlarca farklı noktası olan ve anlık binlerce kullanıcıya hizmet vermesi beklenen büyük bir akıllı şehir ağ altyapısı tasarımında önemli performans dalgalanmalarına hatta ağ yatırım projesinin daha en baştan başarısızlığına sebep olabilecektir. Bu sebeple, ağ altyapılarında kullanılan cihazların ihtiyaç duyulan yetkinliklere sahip olmasına rağmen ağdaki diğer cihazlarla

bağlantılarının doğru tasarlanıp yapılandırılmadığı zaman verimsiz, istekleri cevaplayamayan ve uzun vadede sürdürülemeyecek bir altyapıya dönüşmesi riskini önceden değerlendirebilmek için güncel ağ simülasyon ve emülasyon yazılım araçlarının kullanılması doğal bir gereklilik olarak ortaya çıkmaktadır. Belki bu tür yazılımların BİT sisteminde kullanım seviyeleri veya bunları kullanabilecek yetkin teknik personel sayısı da yeni performans göstergeleri olarak tanımlanabilir.

Diğer yandan kurulumu tamamlanan bir altyapının, bu çalışmada ana hatlarıyla özetlenen ITIL, COBIT ve ISO27001 standartları süreçlerine uygun hale getirilmesi BT sisteminin verimliliğini artıracaktır. Bu doğrultuda, teknik personelin güncel bilişim kütüphaneleri ve standart süreçlerine hâkim olması ve yönetilen cihazlar ile ilgili uzmanlaşmaları sağlanarak dış kaynaklara bağımlılık en aza indirgenerek yetkinlik seviyeleri artırılmalıdır. Teknik ekibin yönettiği altyapının sürekliliğini sağlamak için tasarıma, kritik noktalara göre belirlenen temel performans göstergeleriyle altyapı izlenebilir hale getirilmelidir ve tercihen insan hatasını en aza indirgemek için yazılımlar aracılığı ile takip edilmelidir. Böylelikle ağ performansına ait değişimler izlenebilir ve kritik noktalarda gerekli önlemlerin zamanında alınmasıyla ağın sürekliliği sağlanabilir. Ayrıca teknik ekibin güçlü ve eksik olduğu yönleri görmek, en çok emek ve zaman harcanan problemleri saptamak gibi analizler için de makine öğrenimi yöntemlerini de kullanan problem takip yazılımlarıyla BİT faaliyetleri tamamıyla ölçülebilir hale getirilebilir.

Ülkemizde birçok altyapı yatırımında bilişim projelerine ve çoğunlukla ithal cihazların alımı bazında çok büyük bütçeler ayrılmasına rağmen daha en başta bu projeleri tasarlayan kişilerin yetkinlik, beceri ve eğitim seviyeleri sorgulanmamakta, tasarım sonrasında kurulan altyapıyı yönetecek teknik ekibin istihdamı ve sonrasında yetkinliklerinin artırılması için yeterli bütçeler ayrılmamaktadır. Bu durum sonucunda ortaya doğru tasarlanmamış, yönetilemeyen ve verimsiz birçok altyapı çıkmakta ve yeterince yararlanılamayan yatırım harcamalarına sebep olunmaktadır. Bilişim projelerinde en önemli faktör, bilişim sözcüğünün kökenindeki bilmek fiiline hem özne hem de nesne olan insandır. Bu yüzden özellikle dövize endeksli büyük bütçeler gerektiren BİT ağ altyapı proje tasarımlarında yetkin ve ilgili eğitim geçmişlerine

sahip mühendis, uzman ve bilim adamlarının yer alması ulusal açıdan stratejik düzeyde bir önem arz etmektedir. En başta yanlış tasarlanan bir altyapının, kurulumu sonrasındaki dönemde ortaya çıkan problemlere karşılık önerilen kısa vadeli çözümler veya günü kurtaran cihaz bileşen alımlarıyla, sürdürülebilir ve kullanıcı memnuniyeti yüksek bir hale dönüşmesini beklemek pek gerçekçi olmayacaktır. Bu sebeple altyapı projeleri, yetkin mühendislik, somut ölçütler ışığında seçilen doğru tasarım yaklaşımı, yetkin teknik ekip ve en az dışa bağımlılık ilkeleri ile tasarlanmalıdır. Proje yönetimi ilgili eğitim, tecrübe ve sertifikasyona sahip uzman proje ekibi tarafından yapılmalı ve ekipler arası koordinasyon sağlanarak projenin planlanan şekilde ilerleyişi yönetim ekibinin sürekli kontrolü altında olmalıdır.

Akıllı şehir ve Endüstri 4.0 gibi uygulamaları gibi Nesnelerin İnterneti (IoT) ağ tabanlı altyapılar sensörlerin fiziksel kabiliyetlerine göre çok geniş alanlara yayılabilmektedir. Sahadaki cihazlardan toplanan verilerin anlık merkeze taşınması için her noktaya kablo çekmek her zaman mümkün olmamakta ya da maliyetler ve iş yükünden dolayı tercih edilmemektedir. Sahadaki IoT cihazları için kablolama altyapısı kullanmadan GSM APN bulutu üzerinden izole bir ağ altyapısı oluşturarak verilerin merkeze taşınabileceği gibi, bu çalışmada ağ tasarımı olarak örneklendirilen DMVPN tasarımı da kullanılabilir. Böylece uzak mesafedeki IoT cihazlarının merkeze erişimi kolaylıkla sağlanarak, örneğin akıllı ulaşım sistemleri gibi bütünleşik uygulama ağları uzaktan daha verimli ve etkin yönetilebilir hale getirilebilir.

Diğer yandan özellikle Endüstri 4.0 uygulamalarına uyum çalışmaları başlatan üretim tesislerinde siber-fiziksel cihazlar ve altyapıları için kablolama yapısı, kenar anahtar, erişim noktası gibi faktörler ısı, nem, su, kimyasallar gibi dış etkilere göre ve saha şartlarına uygun olarak belirlenmelidir. Özellikle siber-fiziksel cihazlardan toparlanan verilerin kritik olduğu bu üretim alanlarında ağ yedekliği projelendirme aşamasında tasarlanmalı ve bağlantılar zarar gördüğünde ağın kesintisizliğinin nasıl sağlanacağı ortaya çıkabilecek muhtemel maliyetleriyle beraber önceden planlanmalıdır. Kullanılan çeşitli sensör cihazlarına ve altyapıya özel tanımlanan TPG'ler optimal olarak belirlenmiş periyotlarda takip edilmelidir.

KAYNAKLAR

Ab (2020). 15 Mayıs 2020, https://httpd.apache.org/docs/2.4/tr/programs/ab.html

Adepetu, A., Arnautovic, E., Svetinovic, D. ve L. de Weck, O. (2014). Complex urban systems ICT infrastructure modeling: A sustainable city case study. *IEEE Transactions On Systems, Man, And Cybernetics*, *44* (3), 363-374.

Akçura, M.T. ve Avci, S.B. (2014). How to make global cities: Information communication technologies and macro-level variables. *Technological Forecasting & Social Change, 89*, 68-79.

Al-Shehri, S. M., Loskot, P., Numanoglu, T. ve Mert, M. (2017). *Common metrics for analyzing, developing and managing telecommunication networks*. *arXiv:1707.03290,* 1-51.

Al-Shehri, S. M., Loskot, P., Numanoglu, T. ve Mert, M. (2017). Common metrics for analyzing, developing and managing telecommunication networks. *arXiv:1707.03290.*

Artinyan, E. N. (2007). *CobiT Çerçevesi*. İstanbul: Deloitte.

Aydoğdu, K. (2016a). *ITIL, COBIT ve ISO27001 arasındaki ilişkiler*. 6 Şubat 2020, https://itilnotlari.wordpress.com/2016/04/08/itilde-strategy-design-surecleri/

Aydoğdu, K. (2016b). *ITIL'de strategy-design süreçleri*. 6 Şubat 2020, https://itilnotlari.wordpress.com/2016/03/04/itil-cobit-ve-iso27001-arasindaki-iliskiler/

Axelos, (2019). *ITIL Foundation V4 Edition*. Londra: TSO.

Bahnasse, A. ve Kamoun, N. E. (2015). Study and analysis of a dynamic routing protocols' scalability over a dynamic multi-point virtual private network. *International Journal of Computer Applications*, *123* (2), 26-31.

Bayram, M. (2016). *ITIL ve COBIT ve ISO 27001*. 13 Nisan 2020, https://mustafabayramblog.wordpress.com/2016/03/04/itil-cobit-iso-27001/

Berqnet (2019). *Firewall cihazı nedir? Ne işe yarar?*. 15 Şubat 2020, https://berqnet.com/blog/firewall-cihazi-nedir

Bholebawa, I. Z. ve Dalal, U. D. (2016). Design and performance analysis of OpenFlow-Enabled network topologies using Mininet. *International Journal of Computer and Communication Engineering*, *5* (6), 419-429.

Boyan, F. (2017). *Dinamik yönlendirme nedir*. 14 Nisan 2020, https://www.firatboyan.com/dynamic-routing-protocol-dinamik-yonlendirme-protokolu.aspx

Bülbül, H., Topal, A., Büyükkeklik, A. ve Demirer, Ö. (2016). Bilişim teknolojilerinin çeşitli performans göstergeleri ile ilişkisi, *Selçuk Üniversitesi Sosyal ve Teknik Araştırmalar Dergisi, 11*, 18-33.

Cacti (2020). 15 Şubat 2020, https://www.cacti.net/

Celebic, B. ve Breu, R. (2015). Using green KPIs for large IT infrastructures' energy and cost optimization. *3rd International Conference on Future Internet of Things and Cloud,* 645-650.

Chandan, Y. N. ve Utpat, B. A. (2013). Performance analysis of OSPF and EIGRP routing protocols for greener internetworking. *International Conference in Distributed Computing & Internet Technology (ICDCIT-2013) Proceedings published in International Journal of Computer Applications,* 1-5.

Cicioğlu, M. ve Çalhan, A. (2016). Yazilim tanimli ağlarda denetleyiciler. *International Academic Research Congress,* 3030-3034.

Cisco (2017). *Spanning tree protocol problems and related design considerations*. 20 Ocak 2020, https://www.cisco.com/c/en/us/support/docs/lan-switching/spanning-tree-protocol/10556-16.html

Cisco (2020). *Campus LAN and wireless LAN design guide*. 01 Eylül 2020, https://www.cisco.com/c/en/us/td/docs/solutions/CVD/Campus/cisco-campus-lan-wlan-design-guide.pdf

Curl (2020). 15 Mayıs 2020, https://curl.haxx.se/

Dordal, P.L. (2016). *Mininet*. 2 Haziran 2020, http://intronetworks.cs.luc.edu/current2/html/mininet.html

Duygun, H. C. (2016). *Genel hatlarıyla COBIT5.* 15 Ocak 2020*, http://hcduygun.blogspot.com/2016/02/genel-hatlaryla-cobit-5.html*

Efe, A. (2019). Kurumsal uygulamalarda tekli çerçeve ile kurumun baştan sona kapsanması: Türk kamu sektörü örneği için kalkınma ajansları üzerinden bir analiz. *Pamukkale İşletme ve Bilişim Yönetim Dergisi, 6* (1), 1-19.

Grimson, E. (2019). *Introduction to computer science and programming using Python*. Uzaktan Eğitim, Massachusetts Institute of Technology, Massachusetts.

Gu, Y. ve Grossman, R. (2005). Optimizing UDP-based Protocol Implementations. *In the Proceedings of the Third International Workshop on Protocols for Fast Long-Distance Networks (PFLDNet 2005), (Lyon, France, 2005)*, 3-4.

Haizan, R. N., Alinda, R. ve Rahman, A. (2011). The KPI development framework for ICTSQ measurement. 3rd *International Conference on Information and Financial Engineering 12,* 274-279.

International Organization for Standardization (ISO) (2013). *ISO/IEC 27001 Information security management.* 18 Ocak 2020, https://www.iso.org/isoiec-27001-information-security.html

Iperf (2020). 15 Mayıs 2020, https://iperf.fr

ISACA (2020). *Cobit.* 15 Ocak 2020, https://www.isaca.org/resources/cobit

İTÜ BİDB (İstanbul Teknik Üniversitesi Bilgi İşlem Daire Başkanlığı) *(2013a). IGRP ve EIGRP protokolleri*. 15 Şubat 2020, https://bidb.itu.edu.tr/seyir-defteri/blog/2013/09/07/igrp-ve-eigrp-protokolleri

İTÜ BİDB (İstanbul Teknik Üniversitesi Bilgi İşlem Daire Başkanlığı) (2013b). *OSPF protokolü.* 18 Şubat 2020, https://bidb.itu.edu.tr/seyir-defteri/blog/2013/09/07/ospf-(open-shortest-path-first---ilk-açık-yöne-öncelik)-protokolü

İTÜ BİDB (İstanbul Teknik Üniversitesi Bilgi İşlem Daire Başkanlığı) (2013c). Temel Ağ Cihazları. 18 Şubat 2020, https://bidb.itu.edu.tr/seyir-defteri/blog/2013/09/07/temel-a%C4%9F-cihazlar%C4%B1

Jibril, M. (2005). *Information & communications technology service - Review of KPI's*. Breckland: Breckland Council.

Jordan, S. ve Bruno, A. (2016). *Network design models*. 6 Şubat 2020, https://www.ciscopress.com/articles/article.asp?p=2698000

Karaman, Ö. (2006). *IP şebekelerinde yönlendirme protokolleri*. Yüksek Lisans Tezi, İstanbul Teknik Üniversitesi, İstanbul.

KAS (2010). *ISO 27001:2005 Bilgi güvenliği yönetim sistemi*. 9 Nisan 2020, http://www.kascert.com/goster.aspx?metin_id=352

Kaynak, A. (2011). *ITIL ve Kazanımları*. 15 Şubat 2020, https://www.mshowto.org/itil-ve-kazanimlari.html

Ke, C. H. (2016). *Using FNSS (Fast Network Simulation Setup) to build network topology for Mininet emulation*. 27 Mayıs 2020, http://csie.nqu.edu.tw/smallko/sdn/mininet_fnss.htm

Kırcı, P., Toka, Ç. Ö. ve Erdem, F. (2015). IP/MPLS ağlar üzerinde sanal-yerel servisler ve yönlendirici konfigürasyonları. *Uludağ Üniversitesi Mühendislik Fakültesi Dergisi, 20* (2), 155-165.

KNET Solution (2020). *Mininet tutorial*. 17 Mayıs 2020, http://knetsolutions.in/sdn-tutorials/mininet-tutorial/

Knight, J. (2015). *OSPF vs EIGRP for DMVPN*. 20 Şubat 2020, https://www.packetmischief.ca/2015/12/01/ospf-vs-eigrp-for-dmvpn/#ospftune

KnowledgeClub (2020a). *Cisco ARCH Eğitimi,* Uzaktan Eğitim.

KnowledgeClub (2020b). *Cisco DESIGN Eğitimi,* Uzaktan Eğitim.

KPI library (2020). 10 Mayıs 2020, http://kpilibrary.com/categories/itman

Lantz, B., Heller, B. ve McKeown, N. (2010). A network in a laptop: rapid prototyping for software-defined networks. *Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks (Hotnets-IX)* içinde *19* (1-6). New York: ACM.

Linkletter, B. (2015). *Using the POX SDN controller*. 30 Mayıs 2020, http://www.brianlinkletter.com/using-the-pox-sdn-controller

Marr, B. (2016). *What is a KPI,* 10 Mayıs 2020, http://www.bernardmarr.com.

Masset, B. ve Taburiaux, O. (2018). *Simulating industrial control systems using mininet*. Yüksek Lisans Tezi, Louvain-la-Neuve Üniversitesi, Belçika.

McGill, S. (2007). *Network performance management using application-centric key performance indicators.* Doktora Tezi, Central Florida Üniversitesi, Florida.

MEB (Milli Eğitim Bakanlığı) (2011). *Bilişim teknolojileri ağ temelleri*. Ankara: MEB, http://www.megep.meb.gov.tr/mte_program_modul/moduller_pdf/A%C4%9F%20Temelleri.pdf

Mininet GitHub (2018). 29 Mayıs 2020, https://github.com/mininet/mininet/wiki/Introduction-to-Mininet

Mininet (2018). *Resmi web sitesi*. 3 Haziran 2020, http://mininet.org/

Mirantis (2016). *Software Defined Networking Fundamentals (SDN50), Chapter 2 Mininet*. 17 Mayıs 2020, http://trainer.edu.mirantis.com/SDN50/mininet.html#lab-mininet

Mrtg (2020). 15 Şubat 2020, https://www.mrtg.com/

Nagios (2020). 15 Şubat 2020, https://www.nagios.org/

Netstat (2020). 15 Mayıs 2020, https://linux.die.net/man/8/netstat

O'Briain, D. (2017). *Building a network training emulator (NTE) software defned networking (SDN) virtual machine (VM)*. 01 Mayıs 2020, http://www.obriain.com/training/TEL3214/odt/TEL3214-Appendix_02-Build_SDN_VM_odt.pdf

OpenFlow (2020). 15 Mayıs 2020, https://www.opennetworking.org/

OpenNMS (2020). 15 Şubat 2020, https://www.opennms.com/

Open Vswitch (2020). 15 Mayıs 2020, https://www.openvswitch.org/

Personal, E., Guerrero, J. I., Garcia, A., Pena, M. ve Leon, C. (2014). Key performance indicators: A useful tool to assess Smart Grid goals. *Energy, 76,* 976-988.

Ping (2020). 15 Mayıs 2020, https://linux.die.net/man/8/ping

POX Wiki (2018). 3 Haziran 2020, https://openflow.stanford.edu/display/ONL/POX+Wiki.html.

Preston, H. (2020). *Introduction to writing network tests with pyATS.* 14 Nisan 2020*,* Uzak Konferans, https://us.pycon.org/2020/schedule/presentation/134.

Putty (2020). 15 Mayıs 2020, https://www.putty.org/

Qi, R., Liu, W., Gutierrez, J. ve Narang, M. (2017). Sustainable and resilient network infrastructure design for cloud data centers. J. Marx Gómez vd. (Ed.) *Engineering and Management of Data Centers* içinde (227-259). Cham: Springer

Raza, T., Bin Muhammad, M. ve Majid, M. A. A. (2016). A comprehensive framework and key performance indicators for maintenance performance measurement. *ARPN Journal of Engineering and Applied Sciences*, *11* (20).

Rohani, H. ve Roosta, A. K. (2014). *Calculating total system availability*. Amsterdam: Information Services Organization KLM-Air France.

Schalley, A. (2013). *Hierarchical design model – core – distribution – access layer*. 13 Nisan 2020, https://www.schalley.eu/2013/05/08/hierarchical-design-model-core-distribution-access-layer/.

SDNHUB (2020). 3 Haziran 2020, http://sdnhub.org/.

Shorfuzzaman, M. ve Rahman, M. (2016). Characterizing end-to-end delay performance of randomized TCP using an analytical model. *(IJACSA) International Journal of Advanced Computer Science and Applications*, *7*(3), 406-412.

Sood, M. ve Sharma, K. K. (2014). Mininet as a container based emulator for software defined networks. *International Journal of Advanced Research in Computer Science and Software Engineering Research Paper*, *4* (12), 681-685.

Tshark (2020). 15 Mayıs 2020, https://www.wireshark.org/docs/man-pages/tshark.html

Tutu, İ. (2010). Cobit nedir. 8 Nisan 2020, https://www.cozumpark.com/cobit-nedir/

Türkiye Bilişim Derneği (2020). *Bilgi Teknolojisi Altyapı Kütüphanesi.* 8 Mart 2020*,* https://tr.wikipedia.org/wiki/Bilgi_Teknolojisi_Altyap%C4%B1_K%C3%BCt%C3%BCphanesi

Tkaczewski, J. (2016). *How does UDP buffer sizing affect performance of FileCatalyst*. 2 Ağustos 2020, https://support.filecatalyst.com/index.php?/Knowledgebase/Article/View/14/1/how-does-udp-buffer-sizing-affect-performance-of-filecatalyst#:~:text=Too%20little%20UDP%20buffer%20space,kernel%20to%20discard%20UDP%20packets.&text=Scheduling%20and%20processing%20such%20timers,(%22user%20time%22).

Van Adrichem, N.L.M., Doerr, C., ve Kuipers, F.A. (2014). OpenNetMon: Network monitoring in OpenFlow Software-Defined Networks. J. Janusz Filipiak (Ed.) *Proceedings 2014 IEEE Network Operations and Management Symposium (NOMS)* içinde (1-8). Krakow: IEEE Society.

Vmstat (2020). 15 Mayıs 2020, https://linux.die.net/man/8/vmstat

Vmware Workstation (2020). *Versiyon: 15 pro*. 15 Mayıs 2020, https://www.vmware.com/products/workstation-pro.html

Wang, J., Zhou, M. ve Li, Y. (2004). Survey on the end-to-end internet delay measurements. *7th IEEE International Conference*, 155-166.

Wireshark (2020). 15 Haziran 2020, https://www.wireshark.org/

XmingServer (2020). 5 Haziran 2020, https://sourceforge.net/projects/xming/

Yaşar, F. (2011). *Router nedir? Nasıl çalışır? Ne amaçla kullanılır. 19 Ocak 2020,* https://www.mshowto.org/router-nedir-nasil-calisir-ne-amacla-kullanilir.html

Zabbix (2020). 15 Şubat 2020, https://www.zabbix.com/

EKLER

EK-A: Ağ simülasyon araçları

Ağ Simulasyonu Yazılım İsmi	İnternet Adresi	Dokümantasyon
OMNeT++	https://omnetpp.org/	https://omnetpp.org/documentation/
OPNET	https://www.riverbed.com/sg/products/steelcentral/	https://www.riverbed.com/sg/products/steelcentral/s/artide/DOC-4763
NS3	https://www.nsnam.org/	https://www.nsnam.org/documentation/
NS2	https://www.isi.edu/nsnam/ns/	https://www.isi.edu/nsnam/ns/ns-documentation.html
FNSS	https://fnss.github.io/	https://fnss.readthedocs.io/en/latest/#
PYNSIM	https://pypi.org/project/PyNSim/	http://umwrg.github.io/pynsim/
MININET	https://mininet.org/	https://github.com/mininet/mininet/wiki/Documentation
SIM2NET	https://pypi.org/project/sim2net/	https://sim2net.readthedocs.io/en/latest/
PYATS	https://pypi.org/project/pyats/	https://developer.cisco.com/docs/pyats/api/
GENIE	https://pypi.org/project/genie/	https://developer.cisco.com/docs/genie-docs/
ANSIBLE	https://www.ansible.com/	https://docs.ansible.com/
GNS3	https://www.gns3.com/	https://docs.gns3.com/
VIRL	http://virl.cisco.com/	https://learningnetwork.cisco.com/s/article/virl-documentation-index
EVE-NG	https://www.eve-ng.net/	https://www.eve-ng.net/index.php/documentation/

EK-B: Kısaltmalar

ABR Alan Sınırı Yönlendiricisi
API Uygulama Programlama Arabirimi
ARP Adres Çözümleme Protokolü
ASBR Otonom Sistem Sınır Yönlendiricisi
Avg Ortalama
BDR Belirlenmiş Yedek Yönlendirici
BGP Sınır Geçit Protokolü
BGYS Bilgi Güvenliği Yönetimi Sistemi
BİT Bilgi ve İletişim teknolojileri
BSI İngiliz Standartları Enstitüsü
BT Bilgi Teknolojileri
COBIT Bilgi ve İlgili Teknolojiler İçin Kontrol Hedefleri
CPU Merkezi İşlem Birimi
ÇPEA Çoklu Protokol Etiket Anahtarlama
Data Center Veri Merkezi
DMVPN Dinamik Çok Noktalı Sanal Özel Ağ
DNS Alan İsimlendirme Sistemi
DR Belirlenmiş Yönlendirici
DUAL Difüzyon Güncelleme Algoritması
EBGP Dış Sınır Geçit Protokolü
EIGRP Gelişmiş İç Ağ Geçidi Yönlendirme Protokolü
FKM Felaket Kurtarma Merkezi
GAA Geniş Alan Ağı
GB Gigabit
GRE Genel Yönlendirme Kapsülleme
GSM APN Mobil İletişim için Global Sistemler Erişim Noktası Adı
HTTP Zengin-Metin Transfer Protokolü
IBGP İç Sınır Geçit Protokolü
ID Kimlik Numarası
IETF İnternet Mühendisliği Görev Grubu
IoT Nesnelerin interneti
IP Internet Protokolü
ISACA Bilgi Sistemleri Denetim ve Kontrol Derneği
ISO Uluslararası Standartlar Kurumu
ISS İnternet Servis Sağlayıcı
ITIL Bilgi Teknolojisi Altyapı Kütüphanesi
Kbit Kilobit
Kbps Kilobit per Second
Kbytes Kilobytes
L2TP Katman 2 Tünel Protokolü
LSA Hat Durumu İlanı
LSDB Hat Durumu Veritabanı
LSU Hat Durum Güncellemeleri
MAC Ortam Erişim Kontrolü
Mbit Megabit

Mbytes Megabytes
Mdev Hareketli standart sapmadır. Her Ping denemesi sonucu oluşan RTT değerinin ortalama RTT'den ne kadar uzakta olduğunun ortalamasıdır. Mdev ne kadar yüksek olursa RTT göstergesi o kadar değişkendir.
mGRE Çoklu Genel Yönlendirme Kapsülleme
MHZ Megahertz
MPLS Çoklu Protokol Etiket Anahtarlama
Ms Milisaniye
MTBF Arızalar Arasındaki Ortalama Süre
MTTF Arızaya Kadar Ortalama Süre
MTTR Ortalama Tamir Süresi
NHRP Sonraki Atlama Çözünürlüğü Protokolü
OSI Açık Sistem Arabağlantısı
OSPF İlk Açık Yöne Öncelik
OTV Kaplama Taşıma Sanallaştırması
PPS Saniye Başına Paket
RAM Rastgele Erişimli Hafıza
RTP Güvenilir Taşıma Protokolü
RTT Gidiş-Dönüş Gecikme Süresi
SIA Bir yönlendirme rotasının aktif durumda kalabileceği maksimum süre dolduktan sonra EIGRP protokolünde oluşan durum.
SLA Hizmet Seviyesi Anlaşması
Sn Saniye
SÖA Sanal Özel Ağ
STP Kapsama Ağacı Protokolü
TCP İletim Kontrol Protokolü
TPG Temel Performans Göstergesi
UDP Kullanıcı Veri Bloğu Protokolü
UPS Kesintisiz Güç Kaynağı
VLSM Değişken Uzunluklu Alt Ağ Maskeleri
VRRP Sanal Yönlendirici Yedekliliği Protokolü
VXLAN Sanal Genişletilebilir Yerel Alan Ağı
YTA Yazılım Tabanlı Ağlar