
GAZİ ÜNİVERSİTESİ
Gazi
1926

SANAL AĞ GÜVENLİĞİ FONKSİYONLARININ ENERJİ ETKİN YERLEŞTİRİLMESİ: YÖNTEM ÖNERİLERİ VE UYGULAMA

Merve Sedef DEMİRCİ

DOKTORA TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ ANA BİLİM DALI

GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

MART 2020

Merve Sedef DEMİRCİ tarafından hazırlanan "SANAL AĞ GÜVENLİĞİ FONKSİYONLARININ ENERJİ ETKİN YERLEŞTİRİLMESİ: YÖNTEM ÖNERİLERİ VE UYGULAMA" adlı tez çalışması aşağıdaki jüri tarafından OY BİRLİĞİ ile Gazi Üniversitesi Bilgisayar Mühendisliği Ana Bilim Dalında DOKTORA TEZİ olarak kabul edilmiştir.

Danışman: Prof. Dr. Şeref SAĞIROĞLU

Bilgisayar Mühendisliği Ana Bilim Dalı, Gazi Üniversitesi ……..……..

Bu tezin, kapsam ve kalite olarak Doktora Tezi olduğunu onaylıyorum.

Başkan: Prof. Dr. İlyas ÇİÇEKLİ

Bilgisayar Mühendisliği Ana Bilim Dalı, Hacettepe Üniversitesi ...……

Bu tezin, kapsam ve kalite olarak Doktora Tezi olduğunu onaylıyorum.

Üye: Prof. Dr. Suat ÖZDEMİR

Bilgisayar Mühendisliği Ana Bilim Dalı, Gazi Üniversitesi ...……

Bu tezin, kapsam ve kalite olarak Doktora Tezi olduğunu onaylıyorum.

Üye: Doç. Dr. Hacer KARACAN

Bilgisayar Mühendisliği Ana Bilim Dalı, Gazi Üniversitesi .....……

Bu tezin, kapsam ve kalite olarak Doktora Tezi olduğunu onaylıyorum.

Üye: Dr. Öğr. Üyesi Pelin ANGIN

Bilgisayar Mühendisliği Ana Bilim Dalı, Gazi Üniversitesi ...……

Bu tezin, kapsam ve kalite olarak Doktora Tezi olduğunu onaylıyorum.

Tez Savunma Tarihi: 05/03/2020

Jüri tarafından kabul edilen bu çalışmanın Doktora Tezi olması için gerekli şartları yerine getirdiğini onaylıyorum.

……..…….

Prof. Dr. Sena YAŞYERLİ
Fen Bilimleri Enstitüsü Müdürü

ETİK BEYAN

Gazi Üniversitesi Fen Bilimleri Enstitüsü Tez Yazım Kurallarına uygun olarak hazırladığım bu tez çalışmasında;

- Tez içinde sunduğum verileri, bilgileri ve dokümanları akademik ve etik kurallar çerçevesinde elde ettiğimi,
- Tüm bilgi, belge, değerlendirme ve sonuçları bilimsel etik ve ahlak kurallarına uygun olarak sunduğumu,
- Tez çalışmasında yararlandığım eserlerin tümüne uygun atıfta bulunarak kaynak gösterdiğimi,
- Kullanılan verilerde herhangi bir değişiklik yapmadığımı,
- Bu tezde sunduğum çalışmanın özgün olduğunu,

bildirir, aksi bir durumda aleyhime doğabilecek tüm hak kayıplarını kabullendiğimi beyan ederim.

……..
Merve Sedef DEMİRCİ
05/03/2020

SANAL AĞ GÜVENLİĞİ FONKSİYONLARININ ENERJİ ETKİN YERLEŞTİRİLMESİ: YÖNTEM ÖNERİLERİ VE UYGULAMA

(Doktora Tezi)

Merve Sedef DEMİRCİ

GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

Mart 2020

ÖZET

Yazılım Tanımlı Ağlar (Software Defined Networking - SDN) ve Ağ Fonksiyonlarını Sanallaştırma (Network Functions Virtualization – NFV) teknolojilerinin kullanıldığı ortamlarda güvenlik; saldırı tespit sistemi ve güvenlik duvarı gibi fonksiyonların sanallaştırılması ve birer yazılım haline getirilerek genel amaçlı sunucular üzerinde çalıştırılması ile sağlanmaktadır. Bir ağda, sanal güvenlik fonksiyonları (virtual security functions - VSF) tarafından işlenen trafik akışlarının farklı güvenlik gereksinimleri olabilmektedir. Öte yandan her ağda, güvenlik sağlanırken göz önünde bulundurulması gereken ve ağın ihtiyaçlarına göre farklılık gösteren bir takım operasyonel amaçlar (maliyetin ez aza indirgenmesi, yük dengeleme yapılması vb.) vardır. Bu nedenle, ağın güvenlik gereksinimlerini ve operasyonel amaçlarını göz önünde bulundurarak VSF'lerin yerleştirilmesi ele alınması gereken önemli bir problemdir. Bu doğrultuda, tez kapsamında yapılan çalışmalar üç başlık altında toplanmıştır. İlk çalışmada, literatürde ilk defa bir taksonomi geliştirilmiş ve literatürdeki mevcut VSF yerleştirme çözümleri bu taksonomi ışığında sekiz boyutta sınıflandırılarak tartışılmıştır. İkinci çalışmada, literatürde ilk defa, trafik akışı bazında güvenlik gereksinimleri göz önünde bulundurularak enerji etkin bir şekilde sanal ağ güvenliği fonksiyonlarının yerleştirilmesi ve trafiğin ihtiyaçları doğrultusunda bu fonksiyonlar üzerinden yönlendirilmesi problemi modellenmiş ve çözülmüştür. Bu amaçla, bir Tam Sayılı Doğrusal Programlama (Integer Linear Programming – ILP) modeli geliştirilmiş ve bu modelin ölçeklenebilirlik sorununu çözmek için enerji tüketimini en aza indirgeyen çözümleri kısa sürede üreten yeni bir sezgisel algoritma önerilmiştir. Son olarak üçüncü çalışmada, geliştirilen yaklaşımın yazılım tanımlı ağ altyapısında uygulanabilirliğini göstermek için sezgisel algoritmanın aldığı kararlar doğrultusunda ağa gelen trafik akışlarını yönlendiren SDN kontrolcüsü geliştirilmiştir. Sonuç olarak, geliştirilen yöntemlerin, enerji tüketimini %48'e kadar azalttığı ve geliştirilen SDN kontrolcüsünün ağa gelen trafiği toplam bant genişliği, uçtan uca gecikme, paket kayıp oranı ve jitter gibi hizmet kalitesi parametreleri açısından etkin bir şekilde yönlendirebildiği görülmüştür.

Bilim Kodu	:	92407
Anahtar Kelimeler	:	Enerji verimliliği, Ağ güvenliği, VSF yerleştirme, Ağ fonksiyonlarını sanallaştırma, Yazılım tanımlı ağlar, Optimizasyon
Sayfa Adedi	:	109
Danışman	:	Prof. Dr. Şeref SAĞIROĞLU

ENERGY EFFICIENT PLACEMENT OF VIRTUAL SECURITY FUNCTIONS: METHOD PROPOSALS AND APPLICATION

(Ph. D. Thesis)

Merve Sedef DEMİRCİ

GAZİ UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

March 2020

ABSTRACT

In Software Defined Networking (SDN) and Network Functions Virtualization (NFV) enabled environments, network security is provided by virtualizing security functions such as intrusion detection system and deep packet inspection etc., and deploying them on general-purpose servers as software. The traffic handled by virtual security functions (VSF) may have different security requirements. On the other hand, there are some operational objectives that differ according to the needs of the network such as minimizing cost and load balancing. Therefore, the issue of placing VSFs considering both security requirements and operational objectives is an important research challenge. In this regard, the studies conducted in the scope of this thesis is grouped under three parts. In the first study, a novel thematic taxonomy for classifying current VSF placement solutions is developed for the first time in the literature. State-of-the-art studies are reviewed and categorized based upon the eight dimensions of the developed taxonomy. In the second study, the problem of (i) VSF placement with the objective of minimizing energy consumption while meeting security requirements of the traffic at the flow-level, (ii) and routing flows through the VSFs according to their needs is defined and modelled for the first time in the literature. To this end, an Integer Linear Programming (ILP) model is developed to minimize server energy consumption. In addition, a novel heuristic algorithm that produces solutions minimizing energy consumption within practical time limits is proposed to solve the scalability issue of the ILP model for larger scale network instances. Finally in the third study, an SDN controller that forwards network traffic according to the results of the proposed energy-efficient VSF placement heuristic is developed in order to demonstrate the practicality of our placement approach in SDN infrastructures. As a result, it is concluded that the developed methods can reduce energy consumption by up to 48%, and network traffic can be effectively routed in terms of the QoS parameters including throughput, end-to-end latency, packet loss rate, and jitter.

Science Code	:	92407
Key Words	:	Energy efficiency, Network security, VSF placement, Network functions virtualization, Software defined networks, Optimization
Page Number	:	109
Supervisor	:	Prof. Dr. Şeref SAĞIROĞLU

TEŞEKKÜR

Akademik gelişimime büyük katkı sağlayarak doktora çalışmalarım süresince kıymetli görüş ve yardımlarıyla beni yönlendiren danışman hocam Sayın Prof. Dr. Şeref SAĞIROĞLU'na; Çalışmalarım boyunca görüşlerini esirgemeyerek değerli fikirleriyle beni yönlendiren komite üyesi hocalarım Sayın Prof. Dr. İlyas ÇİÇEKLİ ve Sayın Doç. Dr. Hacer KARACAN'a;

Kıymetli zamanlarını ayırarak görüşlerini paylaşan ve değerli fikirleriyle çalışmama katkılar sağlayan jüri üyesi hocalarım Sayın Prof. Dr. Suat ÖZDEMİR ve Sayın Dr. Öğr. Üyesi Pelin ANGIN'a;

Bugünlere gelmemde büyük pay sahibi olan, maddi manevi destekleriyle bana her zaman güç veren ve hayattaki en büyük şansım olan çok değerli aileme;

Varlığıyla hayatımı güzelleştiren ve çalışmalarım süresince desteğini hiçbir zaman esirgemeyerek değerli görüş ve yardımlarıyla ihtiyacım olan her an yanımda olan, birlikte oğlumuzu büyütmekten gurur duyduğum sevgili eşim Mehmet DEMİRCİ'ye;

Ve son olarak, hayatımıza anlam katarak bana mutlulukların en güzelini yaşatan ve varlığıyla bana en güzel güç ve motivasyon kaynağı olan canım oğlum Alptuğ'a sonsuz teşekkürlerimle.

İÇİNDEKİLER

Sayfa

Sayfa

Sayfa

Sayfa

ÇİZELGELERİN LİSTESİ

Çizelge **Sayfa**

ŞEKİLLERİN LİSTESİ

Şekil **Sayfa**

Şekil **Sayfa**

SİMGELER VE KISALTMALAR

Bu çalışmada kullanılmış simgeler ve kısaltmalar, açıklamaları ile birlikte aşağıda sunulmuştur.

Simgeler	**Açıklamalar**
GBytes	Gigabytes
Mbps	Megabit per second
Ms	Milisaniye
W	Watt

Kısaltmalar	**Açıklamalar**
5G	Fifth Generation (Beşinci Nesil)
ARP	Address Resolution Protokol (Adres Çözümleme Protokolü)
BSS	Business Support System (İş Destek Sistemi)
CAPEX	Capital Expenditures (Yatırım Giderleri)
DDoS	Distributed Denial of Service (Dağıtık Hizmeti Engelleme Saldırıları)
DPI	Deep Packet Inspection (Derin Paket İnceleme)
DR	Dynamic Ratio (Dinamik Oran)
ETSI	European Telecommunications Standards Institute (Avrupa Telekomünikasyon Standartları Enstitüsü)
IDS	Intrusion Detection System (Saldırı Tespit Sistemi)

Kısaltmalar	Açıklamalar
ILP	Integer Linear Programming (Tam Sayılı Doğrusal Programlama)
IoT	Internet of Things (Nesnelerin İnterneti)
IP	Integer Programming (Tam Sayılı Programlama)
IPS	Intrusion Prevention System (Saldırı Engelleme Sistemi)
ISP	Internet Service Provider (İnternet Servis Sağlayıcısı)
	(Karışık Tam Sayılı Kuadratik Kısıtlı Programlama)
NAT	Network Address Translator (Ağ Adresi Dönüştürücü)
NFV	Network Functions Virtualization (Ağ Fonksiyonlarını Sanallaştırma)
NFVI	Network Functions Virtualization Infrastructure (NFV Altyapısı)
ONF	Open Networking Foundation (Açık Ağ Kurumu)
OPEX	Operational Expenditures (İşletim Giderleri)
OSS	Operations Support System (Operasyon Destek Sistemi)
SDN	Software Defined Networking (Yazılım Tanımlı Ağlar)
VANET	Vehicular Ad Hoc Network (Araçsal Tasarsız Ağ)
VIM	Virtualized Infrastructure Manager (Sanallaştırılmış Altyapı Yöneticisi)
VNF	Virtual Network Function (Sanal Ağ Fonksiyonu)

Kısaltmalar	Açıklamalar
VPN	Virtual Private Network (Sanal Özel Ağ)
VSF	Virtual Security Function (Sanal Güvenlik Fonksiyonu)

1. GİRİŞ

Geleneksel ağlarda ağ güvenliği, her biri orta kutu (middlebox) olarak adlandırılan donanım cihazları üzerinde yer alan güvenlik duvarı, derin paket inceleme, saldırı tespit sistemi vb. ağ fonksiyonları ile sağlanmaktadır. Her biri kendi üreticisine özgü olan (vendor specific) bu cihazlar, yüksek seviyeli (high-level) kuralların gerçeklenebilmesi için düşük seviyeli komutlar ile yeniden yapılandırılmak zorundadır. Bu tür yapılandırmalar ise karmaşıklığı artırırken ağı hatalara daha açık bir hale getirmektedir. Bunların yanı sıra, cihaz ve cihazı kontrol eden yazılımın bir arada bulunması ve aynı firma tarafından üretilmesi olarak tanımlanan dikey entegrasyon (vertical integration), ağda herhangi bir değişiklik yapılmasını zorlaştırmakta ve yeniliklerin yavaşlamasına neden olmaktadır. Ayrıca, bu orta kutuları üreten firmaların sayısı arttıkça, ağ altyapısında yer alan cihazların ve uygulamaların birbirleriyle uyumlu bir şekilde çalışmaları zorlaşmaktadır. Bunlara ek olarak, marka bağımlı olan bu cihazların bakımı ve güncellenmesi oldukça büyük boyutlarda bütçe ve iş gücü gerektirmektedir [1].

Ağ fonksiyonlarını sanallaştırma (Network Functions Virtualization - NFV), fonksiyonları donanımdan ayırarak birer yazılım haline getiren ve bu yazılımları genel amaçlı sunucular üzerinde çalıştırarak yukarıdaki problemlere çözümler sunan yeni bir ağ teknolojisidir [1, 2]. NFV ile ağ fonksiyonları sanallaştırılmakta ve her bir servis için özel bir donanım kullanımına ihtiyaç duymadan ağda uygun yerlerde konumlandırılan sunucular üzerinde kolaylıkla çalıştırılabilmektedir. Böylelikle donanım bağımlılığı ile yatırım ve işletim maliyeti azalmaktadır.

Hem endüstrinin hem de akademinin ilgisini çekmiş olan yazılım tanımlı ağlar (Software Defined Networking - SDN) da bazı ortak amaçlarla ortaya çıkmış yeni bir ağ teknolojisidir. SDN'in temel felsefesi kontrol katmanını veri katmanından ayırmaktır [2, 3]. Ağdaki davranışı kontrol eden mantığı, paketlerin yönlendirilmesini sağlayan ağ cihazlarından alarak bu görevi merkezi bir kontrolcü yazılımına vermekte ve böylelikle ağı programlanabilir hale getirmektedir. Böylelikle SDN de, NFV teknolojisinde olduğu gibi, donanımdan bağımsız gelişimin önünü açmakta ve marka bağımlılığını ortadan kaldırmaktadır. Ayrıca, kontrolcünün mantıksal olarak merkezi olması, ağ davranışının izlenmesini ve analiz edilmesini basitleştirmekte ve ağ yönetimini kolaylaştırmaktadır.

SDN ve NFV teknolojileri birbirlerinden bağımsız uygulanabilir olsalar da birbirlerini tamamlayıcı ve destekleyici niteliğe sahiptirler. Her ikisi de donanım bağımlılığını ortadan kaldırma, esnekliği artırma, maliyeti azaltma ve yazılımlaştırma hedeflerine hizmet ettiği için NFV tabanlı teknolojilerin uygulanması için en uygun ağ mimarisi SDN tarafından sağlanmaktadır [2, 3].

SDN ve NFV teknolojilerinin kullanıldığı ağlarda güvenlik; saldırı tespit/engelleme sistemi, derin paket inceleme ve güvenlik duvarı gibi ağ fonksiyonlarının sanallaştırılması ve genel amaçlı sunucular üzerinde birer yazılım halinde çalıştırılması ile sağlanmaktadır. Böylelikle (i) son kullanıcıların ihtiyaçları doğrultusunda güvenlik hizmetleri oldukça özelleştirilebilir hale gelmekte, (ii) bu hizmetlerle ilişkili olan yatırım giderleri (capital expenditures - CAPEX) ve işletim giderleri (operational expenditures - OPEX) azaltılmakta ve (iii) herhangi bir saldırı durumunda güvenlik fonksiyonlarının yerlerinin değiştirilmesi veya trafiğin bu fonksiyonlar üzerinden yeniden yönlendirilmesi çok daha kolay olmaktadır [1, 2, 3].

Tez kapsamında ele alınan problem ve araştırma amacı

Bir ağda, sanal ağ güvenliği fonksiyonları (virtual security function - VSF) tarafından işlenen trafiğin farklı güvenlik gereksinimleri olabilmektedir. Örneğin, şüpheli bir kaynaktan gelen trafiğin derinlemesine incelenmesi gerekirken, başka bir kaynaktan gelen trafiğe yalnızca güvenlik duvarı kurallarının uygulanması yeterli olabilir. Ya da kullanıcıların farklı güvenlik hizmeti aboneliklerine sahip olduğu bir servis sağlayıcısı ağında, trafiğin kullanıcıya özgü VSF kümelerinden geçmesi gerekebilir. Bu nedenle, VSF'lerin ağda nerelere yerleştirileceği ve trafiğin güvenlik gereksinimleri doğrultusunda bu VSF'ler üzerinden nasıl yönlendirileceği ele alınması gereken önemli bir problemdir. Öte yandan ağda, güvenlik sağlanırken göz önünde bulundurulması gereken ve ağın ihtiyaçlarına göre farklılık gösteren bir takım operasyonel amaçlar (maliyetin ez aza indirgenmesi, yük dengeleme yapılması veya enerji tüketiminin en aza indirgenmesi vb.) vardır. Dolayısıyla, VSF yerleştirme yapılırken, trafiğin güvenlik gereksinimlerinin yanı sıra; ağ topolojisi, link kapasiteleri, sunucu özellikleri gibi ağ karakteristiklerinin ve aynı zamanda yazılım ve enerji giderleri gibi bütçe kısıtları ile kullanıcı deneyiminin kalitesi gibi parametrelerin de göz önünde bulundurulması gerekmektedir.

Literatürdeki çalışmalar incelendiğinde, araştırmacıların VSF yerleştirme problemi için maliyetin, gecikmenin ve ağ yükünün en aza indirgenmesi amaçlarını göz önünde bulunduran çözümler geliştirdikleri görülmektedir [1-13]. Çalışmaların büyük çoğunluğunda ise; (i) yerleştirilen VSF sayısı, (ii) donanımsal ekipman sayısı ve (iii) ağ kaynaklar kullanımının (bant genişliği, linklerin sayısı, CPU kullanımı vb.) bir kombinasyonu olarak tanımlanan maliyetin en aza indirgenmesi hedeflenmiştir [1-4, 6-13]. Fakat bu çalışmaların hiçbirinde, ağ maliyetinin en önemli bileşenlerinden biri olan ve sürdürülebilirlik ile yeşil bilişim (green computing) hedeflerine ulaşmak için kritik bir önlem olduğu bilinen enerji tüketimi ele alınmamıştır. Fakat sanal güvenlik fonksiyonu yerleştirme probleminde ele alınması gereken en önemli operasyonel amaçlardan biri, ağdaki enerji tüketiminin minimize edilmesidir. Çünkü VSF'lerin üzerine kurulduğu sunucular muazzam miktarda enerji tüketmektedir [4]. Tipik veri merkezlerinde ve telekom ağlarında [5], sunucular tarafından tüketilen enerji miktarının, güç yönetimi teknikleri uygulansa bile, toplam enerji tüketiminin %50'sini oluşturduğu bilinmektedir [6]. Bu doğrultuda ETSI de, sanal ağ fonksiyonları yerleştirilirken kullanılan kaynak sayısını azaltarak enerji tüketimini optimize eden NFV çerçevelerine olan ihtiyacı vurgulamıştır [8].

Bu amaçla, bu tez kapsamında, literatürde ilk defa, ağa gelen trafik akışlarının güvenlik gereksinimlerini karşılayacak şekilde sunucuların enerji tüketiminin en aza indirgenmesi amacıyla sanal ağ güvenliği fonksiyonlarının yerleştirilmesi problemi modellenmiş ve çözülmüştür. Bu doğrultuda, tez çalışmasının temel amaçları aşağıdaki gibi tanımlanmaktadır.

1. SDN ve NFV tabanlı ağlardaki trafiğin güvenlik gereksinimlerini, trafik akışı seviyesinde karşılayacak şekilde, enerji tüketimini en aza indirgeyen VSF yerleştirme yaklaşımlarının geliştirilmesi
2. Trafiğin güvenlik ihtiyaçları doğrultusunda yapılan enerji etkin VSF yerleştirme ile uyumlu ve ağa gelen trafik akışlarını ağ işleyişi bakımından etkin bir şekilde yönlendirecek SDN kontrolcü tasarımı ve gerçekleştirilmesi

Tezin katkıları

Bu hedeflere ulaşmak için tez kapsamında yapılan işler üç başlık altında toplanarak çalışmanın literatüre katkısı aşağıda özetlenmiştir.

1. İlk çalışmada, SDN ve NFV teknolojileriyle yapılandırılmış ağlarda sanal güvenlik fonksiyonlarının yerleştirilmesi alanındaki çalışmaların sınıflandırılması için literatürde ilk defa bir taksonomi geliştirilmiştir. Geliştirilen taksonomiye göre literatürdeki VSF yerleştirme çözümleri; geliştirilen yerleştirme yaklaşımları, ele alınan operasyonel amaçlar, kullanılan optimizasyon yöntemleri, akış yönetimi stratejileri, uygulama alanları, kullanılan veri, optimizasyon araçları ve simülasyon/emülasyon ortamları olmak üzere 8 boyutta sınıflandırılmıştır.
2. İkinci çalışmada, ağdaki operasyonel ve güvenlik kısıtlarının yanı sıra trafiğin akış bazında güvenlik gereksinimlerini de göz önünde bulundurarak enerji etkin yerleştirme çözümleri üreten bir ILP modeli geliştirilmiştir. Geliştirilen ILP modelinin ölçeklenebilirlik sorununu çözmek için enerji tüketimini en aza indirgeyen yerleştirme çözümlerini ILP modeline göre çok daha kısa sürede üreten yeni bir sezgisel algoritma geliştirilmiştir. Elde edilen bulgular; geliştirilen yöntemlerin enerji tüketimini %48'e kadar azalttığını göstermiştir.
3. Son olarak üçüncü çalışmada, geliştirilen yaklaşımın uygulanabilirliğini göstermek için enerji etkin VSF yerleştirme algoritmasının aldığı kararlar doğrultusunda ağa gelen trafik akışlarını yönlendirecek SDN kontrolcüsü tasarlanmış ve gerçekleştirilmiştir. Elde edilen bulgular, geliştirilen SDN kontrolcüsünün ağa gelen trafiği toplam bant genişliği, ortalama gecikme ve paket kayıp oranı gibi QoS (quality of service - hizmet kalitesi) parametreleri açısından etkin bir şekilde yönlendirebildiğini göstermiştir.

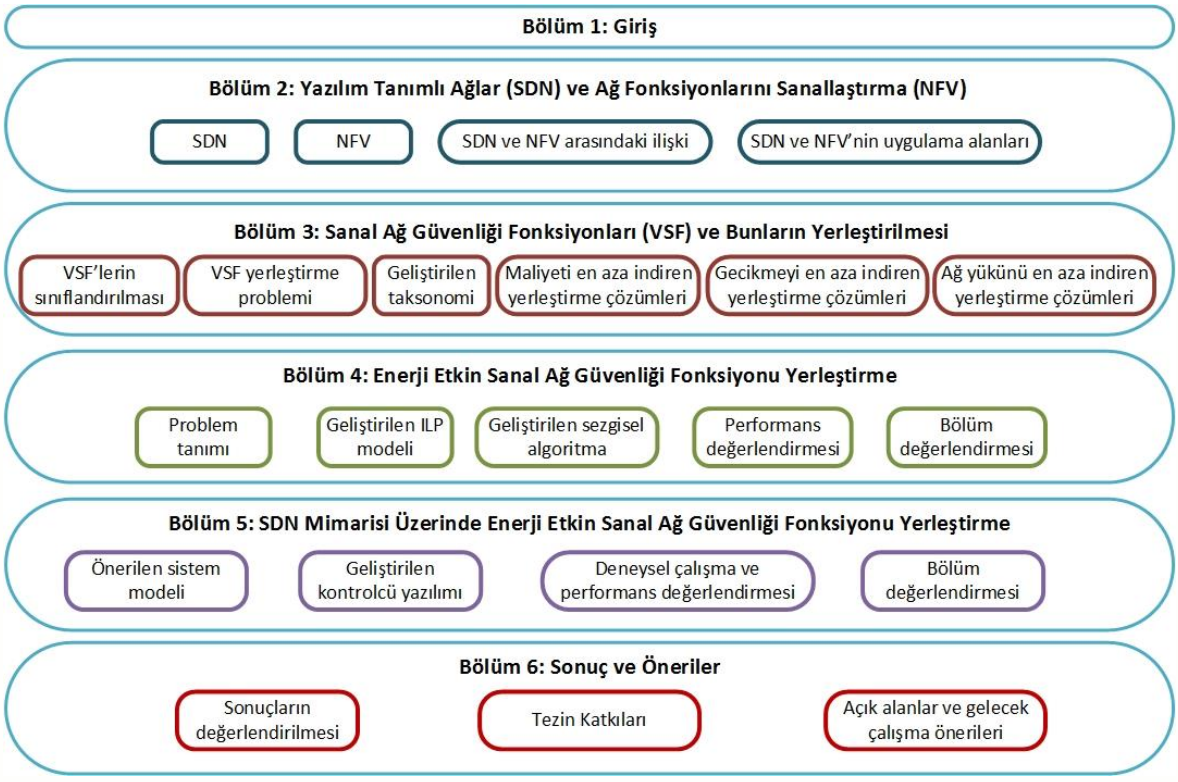
Ayrıca, VSF yerleştirme probleminde ele alınması gereken açık araştırma alanları belirlenerek bu alanlarda neler yapılabileceği gelecek çalışma önerileri olarak sunulmuştur.

Tezin organizasyonu

Tezin organizasyonu Şekil 1.1'de verilmiş olup bölüm içerikleri aşağıda özetlenmiştir:

Bölüm 2'de, SDN ve NFV kavramları tanıtılmış ve bu iki teknoloji arasındaki ilişki; benzer yönleri ve farklılıklarıyla birlikte incelenmiştir. Ayrıca, bu teknolojilerin hayata geçirilebileceği uygulama ortamları ve ağ türleri detaylı bir şekilde açıklanmıştır.

Bölüm 3'te, NFV ile sanallaştırılmış ağ güvenliği fonksiyonları sınıflandırılarak belirlenen kategorilere ait sanal güvenlik fonksiyonları tanıtılmış ve her bir sanal güvenlik fonksiyonu alanında yapılan son çalışmalara genel bir bakış sunulmuştur. Sonrasında sanal ağ güvenliği fonksiyonlarının yerleştirilmesi problemi tanımlanmış ve örnek senaryolar üzerinden açıklanmıştır. Daha sonra bu problemi ele alan çalışmaların sınıflandırılması için geliştirilen taksonomi sunulmuş ve bu taksonomiye ait alt başlıklar açıklanmıştır. Son olarak, literatürdeki mevcut VSF yerleştirme çözümleri göz önünde bulundurdukları operasyonel amaçlar doğrultusunda özetlenmiş ve karşılaştırılmıştır.



Şekil 1.1. Tezin organizasyonu

Bölüm 4’te, sanal ağ güvenliği fonksiyonlarının enerji etkin yerleştirilmesi problemi modellenmiş ve bu problemin çözümü için geliştirilen ILP modeli ve sezgisel algoritma sunulmuştur. Sonrasında, tez kapsamında geliştirilen bu çözümler hem birbirleriyle hem de literatürdeki referans yerleştirme çözümleri ile kıyaslanarak performans değerlendirmeleri yapılmıştır.

Bölüm 5’te, geliştirilen yaklaşımın SDN ortamlarında uygulanabilirliğini göstermek için önerilen sistem mimarisi sunulmuş ve bu amaçla geliştirilen SDN kontrolcü yazılımının performans değerlendirmeleri yapılmıştır.

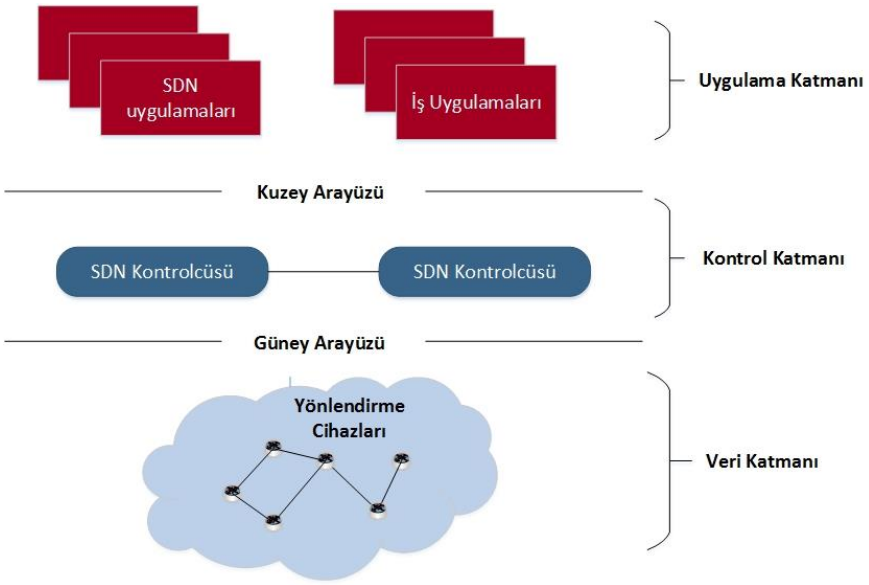
Son olarak Bölüm 6’da, bu tez çalışması kapsamında ele alınan problem, bu problemin çözümü için geliştirilen yöntemler ve elde edilen bulgular özetlenmiş ve yorumlanmıştır. Ayrıca çalışmanın bilime katkısı açıklanmış ve belirlenen açık araştırma konuları ışığında geleceğe yönelik çalışma önerileri sunulmuştur.

2. YAZILIM TANIMLI AĞLAR VE AĞ FONKSİYONLARINI SANALLAŞTIRMA

Bu bölümde, tez kapsamında ele alınan SDN ve NFV teknolojileri tanıtılmış, bu iki teknoloji arasındaki ilişki ele alınmış ve bu teknolojilerin uygulandıkları ağ türleri açıklanmıştır.

2.1. Yazılım Tanımlı Ağlar (Software Defined Networking - SDN)

Yazılım Tanımlı Ağlar, ağdaki kontrol katmanını veri katmanından ayıran yeni bir teknolojidir [14]. SDN ile ağdaki kontrol mekanizması, yönlendirme cihazlarından ayrılarak kontrolcü adındaki merkezi bir kontrol yazılımına verilmektedir. Kontrolcü yazılımı, yönlendirmenin nasıl yapılacağını belirleyen kuralların tanımlamasından sorumluyken, veri katmanındaki programlanabilir anahtarlar, kontrolcü tarafından belirlenen kurallar doğrultusunda trafiğin yönlendirilmesinden sorumludur [15]. Ayrıca, SDN mimarisinde, çeşitli ağ uygulamalarının bulunduğu bir de uygulama katmanı bulunmaktadır. Bu ağ uygulamaları, izleme, yük dengeleme, güvenlik duvarı ve saldırı tespit sistemi gibi fonksiyonları içermektedir. Şekil 2.1'de ONF (Open Networking Foundation) tarafından tanımlanan SDN mimarisi sunulmaktadır [16]. Sunulan mimarideki katmanlar ve bu katmanlar arasındaki ilişki aşağıdaki alt başlıklarda detaylı bir şekilde açıklanmıştır [17].



Şekil 2.1. SDN mimarisi

2.1.1. Veri katmanı (data layer)

Altyapı katmanı olarak da bilinen veri katmanında, paketlerin yönlendirilmesinden sorumlu olan programlanabilir yönlendirme cihazları bulunmaktadır. Geleneksel ağ mimarisindeki yönlendirme cihazlarının aksine bu ağ elemanları, yönlendirmenin nasıl olacağına karar veren gömülü bir yazılım içermemektedirler. Yani, yönlendirme kurallarını kendileri belirleyemezler. Onun yerine, kontrol katmanında yer alan kontrolcü yazılımı tarafından belirlenen kurallar doğrultusunda yönlendirme yaparlar [18, 19]. Bu kurallar, anahtarların akış tablolarında tutulmaktadır ve Güney ara yüzü (Southbound API) olarak adlandırılan ara yüz aracılığıyla kontrolcü tarafından güncellenebilmektedir. Böylelikle, veri katmanındaki ağ elemanları basitleşir ve dikey entegrasyon ortadan kaldırılmış olur. Switch Light, Open vSwitch, OpenFlow Reference ve Pica8 en çok kullanılan programlanabilir anahtar örnekleridir [15].

2.1.2. Kontrol katmanı (control layer)

Kontrol katmanında, yazılım tanımlı ağ mimarisinin en önemli bileşeni olan kontrolcü yazılımı bulunmaktadır. Birbiriyle uyumlu bir şekilde çalışan veya hiyerarşik düzende yapılandırılmış birden fazla kontrolcü de bulunabilmektedir. Ağın beyni olarak düşünülen kontrol katmanı, ağdaki yönlendirme davranışını yönetmektedir. Yönlendirme kurallarını belirleyen kontrolcü yazılımı, bu kuralları veri katmanındaki programlanabilir anahtarların akış tablolarına yazmak için Güney ara yüzünü kullanmaktadır [20]. NOX, POX, Floodlight, Beacon, DIFANE, OpenDaylight ve ONOS farklı amaç ve öncelikler doğrultusunda en çok kullanılan SDN kontrolcüleri arasında yer almaktadır [15, 21].

2.1.3. Güney arayüzü (southbound API)

Kontrol ve veri katmanları arasında bir ara yüz olarak görev yapan güney ara yüzü, kontrolcü ve yönlendirme cihazları arasındaki iletişimi sağlayan protokolleri kapsamaktadır. Günümüzde, güney ara yüzü olarak en çok kullanılan ve bir standart halini almış olan protokol OpenFlow'dur [17]. OpenFlow protokolü (i) kontrolcünün anahtarlar ile güvenli biçimde iletişim kurmasını sağlar, (ii) anahtarlara akış tablolarında değişiklik yapmaları gerektiğini söyler ve (iii) kontrolcü ile anahtarlar arasında akan mesajların şeklini belirler. 2008 yılında Stanford Üniversitesi'nde geliştirilmesinden bu yana, veri merkezleri, omurga

ağları ve kurumsal ağlar gibi birçok ortama adapte edilmiştir. OpenFlow mesajları; (i) kontrolcüden anahtarlara giden kurallar ve güncellemeler ve (ii) anahtarlardaki mevcut kurallarla eşleşmeyen paketler olmak üzere çift yönlü iletilmektedir [22, 23, 24].

2.1.4. Uygulama katmanı (application layer)

Uygulama katmanı, kontrol katmanının üzerinde yer alan ve yük dengeleme, trafik izleme, filtreleme, saldırı tespiti ve derin paket inceleme gibi ağ yönetimi ve güvenliğiyle ilgili fonksiyonların gerçekleştirilmesinden sorumlu ağ uygulamalarını içeren katmandır [17]. Bu katmanda bulunan her uygulama veya fonksiyon, kendi işleviyle ilgili kuralları belirlemektedir. Daha sonra bu kurallar, Kuzey ara yüzü (Northbound API) aracılığıyla kontrolcüye iletilmekte ve OpenFlow kurallarına dönüştürüldükten sonra kontrolcü tarafından programlanabilir anahtarların akış tablolarına yazılmaktadır [15].

2.1.5. Kuzey arayüzü (Northbound API)

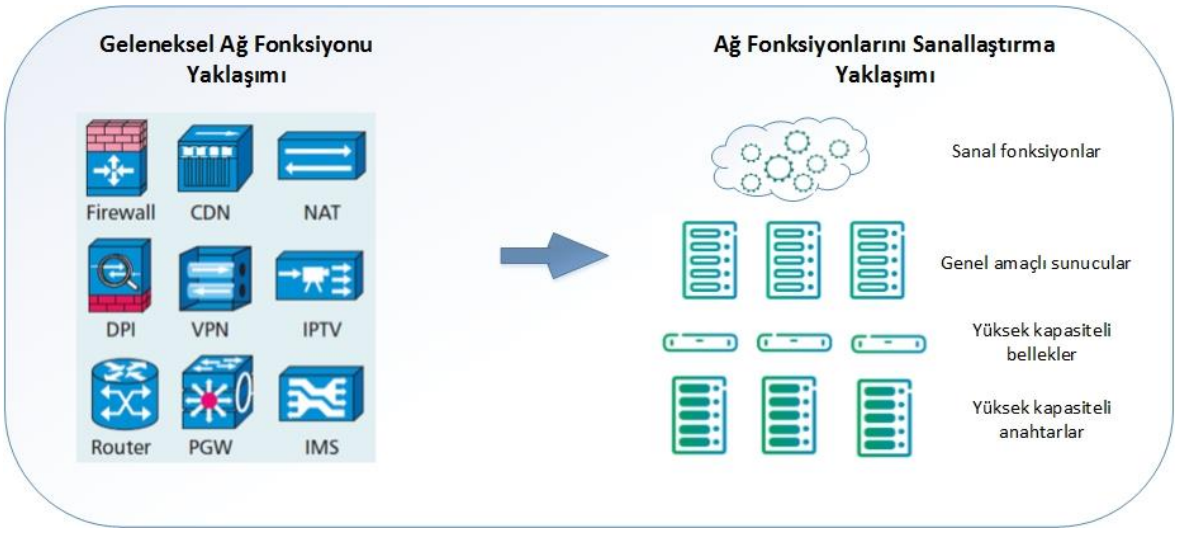
Kuzey ara yüzü, uygulama ve kontrol katmanları arasındaki iletişimi sağlamaktadır. Bir başka ifadeyle, uygulama katmanındaki uygulamaların ağı programlamasına izin vermektedir. Bu ara yüz aracılığıyla, Python, Java ve C++ gibi yüksek seviyeli diller kullanılarak ağ uygulamaları geliştirilebilmektedir. Kuzey ara yüzü, veri katmanı cihazlarının detaylarını soyutlayarak, ağ yöneticileri, servis sağlayıcıları ve araştırmacılar gibi birçok kullanıcıya, kontrol kurallarını ve ağın davranışını özelleştirme fırsatı sunmaktadır [25].

2.2. Ağ Fonksiyonlarını Sanallaştırma (Network Functions Virtualization - NFV)

Günümüzdeki geleneksel ağ mimarisinde, temel ağ işlemlerinin gerçekleştirilebilmesi için farklı türlerde özel donanım gereçleri kullanılmaktadır. Daha açık bir ifadeyle, ağ fonksiyonlarının her biri, geliştirildiği firmaya ait olan, ayrı bir donanım üzerinde çalıştırılmaktadır. Böyle bir mimariye yeni bir servis eklemek, (i) o servisi çalıştıracak cihaz için yeni bir konum bulunması, (ii) farklı firmalara ait cihazların birlikte çalışabilirliğinin sağlanması ve (iii) bir servis ekleyebilmek için tüm mimarinin yeniden tasarlanması açılarından oldukça zordur. Ayrıca, her bir fonksiyon için ayrı bir özel cihaz kullanılması

gerekliliği maliyet açısından da olumsuz sonuçlar doğurmaktadır [26].

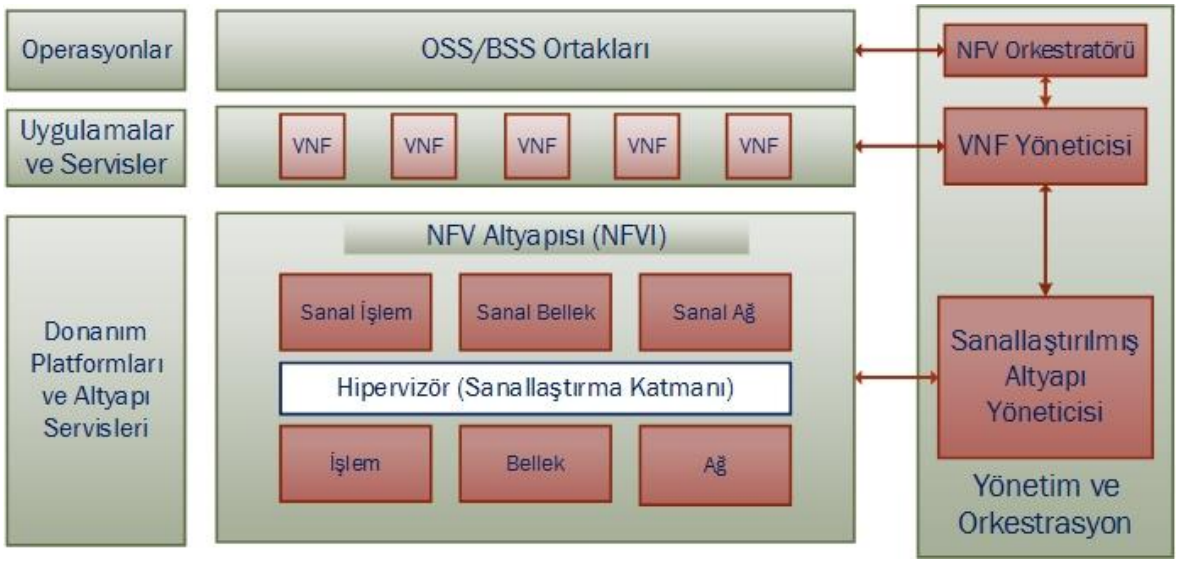
NFV, yukarıda bahsedilen problemlere çözüm sunmak amacıyla, ağ fonksiyonlarını özel donanımlardan ayırarak birer yazılım haline getiren ve bunların genel amaçlı sunucular üzerinde çalıştırılmasını sağlayan yeni bir ağ teknolojisidir [27]. Şekil 2.2'de NFV yaklaşımının nasıl çalıştığı gösterilmektedir [28].



Şekil 2.2. Geleneksel ağ fonksiyonu ve ağ fonksiyonlarını sanallaştırma yaklaşımları

Şekil 2.2'de görüldüğü gibi, NFV ile ağ fonksiyonları sanallaştırılmakta ve her bir servis için özel bir donanım kullanılmasına gerek kalmadan genel amaçlı sunucular üzerinde uygun konumlara yerleştirilebilmektedirler. NFV, SDN'de olduğu gibi donanım bağımlılığını ortadan kaldırmakta ve böylelikle donanım maliyeti ile enerji tüketimini azaltmaktadır. Böylelikle, yenilikler hızlanmakta ve ağ fonksiyonları yazılımlaştırıldıkları için çok daha kısa sürede hizmete sunulabilmektedir [28]. Bunlara ek olarak NFV ile fonksiyonlar arasındaki kaynaklar paylaşılabilmekte, insan gücü ve uzmana duyulan ihtiyaç azaltılmakta ve donanım kurulumlarına ihtiyaç duyulmadan fonksiyonlar ağ üzerinde yeniden konumlandırılabilmektedir.

Çalışma mantığını daha iyi ifade edebilmek için NFV'nin ETSI (European Telecommunications Standards Institute) tarafından belirlenen modüler mimarisi Şekil 2.3'te sunulmuştur [29]. Şekilden de görülebildiği gibi bu mimarinin, NFV altyapısı (NFV Infrastructure - NFVI), sanallaştırılmış altyapı yöneticisi (Virtualized Infrastructure Manager - VIM), VNF yöneticisi (VNF manager) ve orkestratör (orchestrator) olmak üzere



Şekil 2.3. Modüler NFV diyagramı

dört ana bileşeni bulunmaktadır. NFVI, sanal ağ fonksiyonlarının (virtual network functions - VNF) üzerinde çalıştığı ortamdır. Fiziksel kaynaklar, sanal kaynaklar ve sanallaştırma katmanı olarak bilinen hipervizörü içermektedir. Hipervizör; fiziksel bellek, ağ ve işlem kaynaklarının, VNF'ler tarafından kullanılabilecek şekilde sanallaştırılması ve sanal kaynaklara dönüştürülmesinden sorumludur. VNF'lerin altyapıdaki fiziksel kaynaklardan bağımsız bir şekilde çalışabilmesini ve sanal makinelerin işlevselliğinden faydalanabilmesini sağlamaktadır. VNF'ler, NFV'deki sanal kaynakları kullanarak çalışan güvenlik duvarı, ağ adresi dönüştürücüsü (network address translator - NAT) ve WAN iyileştirici (optimizer) gibi sanallaştırılmış ağ fonksiyonlarıdır [29].

Sanallaştırılmış altyapı yöneticisi, sanal makinelerdeki sanal kaynakların VNF'ler ile olan etkileşimini kontrol etmektedir. VNF yöneticisi; VNF'lerin başlatılması, ölçeklenmesi, güncellenmesi ve sonlandırılması gibi VNF yaşam döngüsünün tüm adımlarından sorumlu iken; orkestratör bileşeni, VNF'lerin yönetimi ve birbirleriyle uyumlu bir şekilde çalışmasından sorumludur. Son olarak, bu mimarideki OSS (operation support system) ve BSS (business support system), sırasıyla operasyon destek sistemi ve iş destek sistemi anlamına gelmektedir. OSS, hata yönetimi veya servis yönetimi gibi operasyonel gereksinimlerle ilgilenirken; BSS, müşteri yönetimi, sipariş yönetimi ve ürün yönetimi gibi faaliyetlerle ilgilenmektedir [29].

Önceki paragraflarda da açıklandığı gibi SDN ve NFV, birbirlerinden bağımsız fakat tamamlayıcı teknolojilerdir. Bir arada kullanılmak zorunda değillerdir, fakat ancak birlikte kullanıldıklarında her ikisi de amacına tam olarak ulaşmaktadır. Sonraki alt başlıkta, bu iki teknoloji arasındaki benzerlikler, farklılıklar açıklanacak ve SDN'in VSF yerleştirme çözümlerini nasıl kolaylaştırdığı değerlendirilecektir.

2.3. SDN ve NFV Arasındaki İlişki

SDN ve NFV teknolojileri birbirlerinden bağımsız uygulanabilir olsalar da birbirlerini tamamlayıcı ve destekleyici teknolojilerdir. Şekil 2.4'te ifade edildiği gibi, her ikisi de yazılımlaştırma, programlanabilir ağ alt yapısı, ağ işlemlerinin ve bakımının daha kolay yapılması gibi birçok ortak hedefe sahiptir.

Şekil 2.4. SDN ve NFV teknolojileri arasındaki ilişki

SDN programlanabilir yönlendirme cihazlarının avantajını kullanarak ağdaki karar verme mekanizmasını merkezileştirirken, NFV ağ fonksiyonlarını sanallaştırmakta ve genel amaçlı yüksek kapasiteli sunucular üzerinde birer yazılım olarak çalıştırmaktadır [26, 30]. Bir başka ifadeyle SDN, NFV ile sanallaştırılmış ağ fonksiyonlarının (virtual network functions - VNF) etkin bir şekilde kullanılabileceği programlanabilir ağ alt yapısını sunmaktadır [31]. SDN ve NFV, birlikte, VNF konumlarının ve operasyonel gereksinimlerin trafik yönlendirme kurallarını etkilediği esnek bir ortam sunmaktadır. VNF'ler ağda konumlandırılırken SDN'in sağladığı en büyük kolaylık, yönlendirme

kurallarının hızlı ve dinamik bir şekilde belirlenip çalıştırılabilmesidir. VNF'lerin ağdaki operasyonel amaçlar doğrultusunda ilgili konumlara yerleştirilmesi ve trafiğin bu fonksiyonlar üzerinden yönlendirilmesi, merkezi kontrol altındaki programlanabilir ağ alt yapısı tarafından sağlanan daha hızlı güncelleme yeteneği ve daha kolay bakım hizmeti sayesinde SDN'de çok daha kolay olmaktadır [32–34]. Detaylandırmak gerekirse, trafik akışlarının ihtiyaç duyduğu VNF'ler tarafından doğru bir şekilde işlenebilmesi için gerekli olan yönlendirme kuralları programlanabilir anahtarların akış tablolarına SDN kontrolcüsü tarafından hızlı bir şekilde yazılabilmektedir. Bu sebeplerden dolayı SDN, NFV teknolojilerinin uygulanması için en ideal platform olarak kabul edilmekte ve VNF tabanlı çözümler SDN ortamları için tasarlanmaktadır. Özellikle 5G gibi yeni nesil ağ çözümleri, SDN ve NFV'nin birlikte var olduğu mimarileri temel almaktadır.

2.4. SDN ve NFV Teknolojilerinin Uygulama Alanları

SDN ve NFV teknolojileri, sağladıkları esnek ağ yönetimi, ağ kaynaklarının etkin kullanımı, düşük maliyet ve yüksek ölçeklenebilirlik gibi özellikleri sayesinde, günümüzde birçok ağ türünde uygulanmaya başlamıştır. Bu bölümde, bu teknolojilerin en çok hâkim olduğu uygulama alanları açıklanarak bu alanlarda nasıl kullanıldıklarına değinilmiştir.

2.4.1. Veri merkezleri

SDN ve NFV'nin veri merkezleri için sunduğu en büyük avantajlardan biri ölçeklenebilirliği sağlamalarıdır. SDN ve NFV teknolojileriyle yapılandırılmış veri merkezilerinde, ağ uygulamalarının güncellenmesi veya iyileştirilmesi gerektiğinde, herhangi bir markaya ait yeni bir donanım almak yerine yalnızca yazılımsal değişimler yapmak yeterli olmaktadır [35, 36].

Ağın sanallaştırılması ve yazılımlaştırılmasını hedefleyen bu teknolojilerin veri merkezleri için sunduğu bir diğer avantaj ise yük dengeleme yapılmasını sağlamalarıdır. OpenFlow protokolü ile ağ yükünün arttığı zamanlarda trafiğin alternatif rotalar üzerinden yönlendirilmesi, geleneksel ağlara göre çok daha kolay ve hızlı bir şekilde gerçekleştirilebilmektedir [37, 38].

Bu teknolojiler, veri merkezlerinde enerji tasarrufu yapılmasını da kolaylaştırmaktadır. SDN kontrolcüsünün merkezi yapısı sayesinde, trafiğin durumu ve şartları kolaylıkla izlenebilmekte ve böylelikle aktif durumda olması gereken ağ elemanları belirlenerek diğerlerinin kapatılması sağlanabilmektedir. SDN ve NFV teknolojilerinin uygulandığı veri merkezlerinde, geleneksel ağ altyapısını kullanan veri merkezlerine göre %25 ile %52 oranları arasında enerji tasarrufu sağlandığı bilinmektedir [39].

2.4.2. İnternet servis sağlayıcısı (internet service provider - ISP) ağları

İnternet servis sağlayıcıları, kişilerin veya kurumların internete ve internetteki servislere erişimini sağlayan kuruluşlardır. İnternete bağlanan kişi ve cihaz sayısı arttıkça ISP ağları, kaynakların etkin kullanımını sağlayarak değişen ağ şartları altındaki dinamik trafik hacmine hizmet vermek durumunda kalmıştır. Dolayısıyla, ağ performansını ve kullanıcı kalitesi deneyimini artırmak ve karmaşıklık ile maliyeti azaltmak için SDN ve NFV teknolojilerinin kullanımı ISP ağları için kaçınılmaz hale gelmiştir. Fakat tüm altyapıyı SDN ve NFV teknolojileri ile yeniden inşa etmek, ISP ağları için veri merkezlerinde olduğu kadar kolay değildir, çünkü bu durum belirli protokollerin ve servislerin bozulmasına neden olarak ağın işleyemez hale gelmesine sebep olabilir. Bu nedenle ISP ağları, geriye dönük uyumluluğu koruyan, mevcut protokolleri ve donanımları tam olarak kullanırken SDN ve NFV'nin avantajlarından aşamalı bir biçimde faydalanan evrimsel (evolutionary) bir geçiş yaklaşımını benimsemektedirler [40, 41].

Bu doğrultuda, Google'ın B4 olarak adlandırılan omurga ağında hibrid bir yaklaşım geliştirilerek SDN mimarisi geleneksel ağ altyapısıyla birlikte kullanılmaktadır. Bu uygulama ile SDN'in sunduğu merkezi ağ yönetimi özelliği sayesinde, trafiğin ihtiyaçları doğrultusunda, ortalama bant genişliği değerlerinin maksimize edilebildiği ve hız sınırlandırma işleminin kolaylıkla gerçekleştirilebildiği görülmüştür [42].

2.4.3. Kurumsal ağlar

Kurumsal ağlarda ele alınması gereken en önemli meseleler ölçeklenebilirlik ve güvenliktir. SDN ve NFV altyapısına sahip kurum ağlarında, yük dengeleme yapılması veya herhangi bir saldırı durumunda güvenlik sistemlerinin genişletilmesi, yeniden yapılandırılması ve artırılması geleneksel altyapıya sahip kurum ağlarına göre çok daha kolay olmaktadır. Bu

teknolojilerin sağladığı ölçeklenebilirlik ve esneklik sayesinde, kurumun ihtiyaçları doğrultusunda ağa yeni servisler kolaylıkla eklenebilmekte veya çıkarılabilmektedir.

SDN ve NFV ile yapılandırılmış kurum ağlarında güvenliğin sağlanması da geleneksel ağlara göre pek çok açıdan daha kolay olmaktadır. Kurumlarda, birimlerin görev tanımları farklı olduğu için bunların sistemlere erişim politikaları da farklı yürütülmektedir. Örneğin, bir birimdeki yalnızca bir grup çalışana insan kaynakları verisine erişim hakkı tanımlanmıştır. Geleneksel ağ alt yapısında bu işlem, ağ seviyesinde fiziksel bölümleme ve uygulama katmanında erişim kontrolü mekanizmaları ile sağlanmaktadır. SDN'de ise dinamik olarak sanal ağlar tanımlanabilmekte ve böylelikle her sanal ağın belirli güvenlik politikalarını uygulaması ve yetkisiz erişimlerin engellenmesi sağlamaktadır. Bu mekanizma, "kendi cihazını getir" (bring your own device - BYOD) gibi uygulamalar için de güvenlik kurallarının katı bir şekilde uygulanmasını sağlamaktadır, çünkü daha önceden bilinmeyen bir cihaz sanal ağlardan birine bağlanmak istediğinde kurumsal ağa minimum erişim hakkı tanınmaktadır. Başarılı bir kimlik doğrulama işlemi sonrasında ise SDN ve NFV teknolojilerinden faydalanılarak sanal ağ güncellenmekte, kullanıcının güvenlik seviyesi tanımlanmakta ve ağdaki diğer servislere erişimi gerçekleştirilmektedir [43].

Öte yandan, ağın genel güvenlik politikasının getirdiği gereksinimlere bağlı olarak NFV, trafiğin farklı güvenlik fonksiyonlarından hizmet almasını mümkün kılmaktadır. Örneğin, trafik, sırasıyla güvenlik duvarı, saldırı tespit sistemi ve virüs taraması fonksiyonlarından dinamik bir şekilde yönlendirilebilmektedir. Bu esneklik sayesinde, akış bazında kararlar verilebilmekte, kurumun ihtiyaçlarına uyarlanmış yeni güvenlik çözümleri kolaylıkla oluşturulabilmekte ve böylelikle klasik, kablolu ve statik kurum ağlarına göre birçok üstünlük sağlanabilmektedir.

2.4.4. Nesnelerin interneti (internet of things - IoT)

Nesnelerin interneti, akıllı hizmetler sunmak amacıyla gerçek zamanlı veri toplamak ve iletmek için milyarlarca cihazın ağ bağlantısıyla birbirleriyle iletişim kurmalarını sağlayan bir yapıdır. Yeterli ağ altyapısının olduğu durumlarda, IoT ile birbirine bağlı cihazlara uzaktan erişim sağlanabilmekte ve bu cihazlar kontrol edilebilmektedir. Fakat klasik zaman aşımı tabanlı iletim protokollerini kullanan geleneksel ağlar, IoT'nin gereksinimlerini etkin, ölçeklenebilir, hatasız ve maliyet etkin bir biçimde karşılamakta yetersiz kalmaktadırlar.

Yazılımlaştırma tabanlı SDN ve NFV teknolojileri ise ağın merkezi görünümünden faydalanarak cihazların uzaktan kontrol edilmesini ve yönetilmesini sağlamaktadır. Bu nedenle, bu teknolojiler IoT ağlarının bel kemiğini oluşturmaktadır [44].

SDN tabanlı IoT çözümleri akıllı şehir uygulamalarında önem kazanmaktadır. Genişletilmiş algılama kapasitesine sahip IoT cihazları ile büyük şehirlerden altyapıyla ilgili ve çevresel veriler toplanarak etkili ve sürdürülebilir kentsel gelişim çözümleri geliştirilebilmektedir. Veri toplanması aşamasında, şehrin farklı bölgelerine kurulmuş olan IoT ortamlarının ortak bir fiziksel altyapıya sahip olması gerekmektedir. Bu altyapıda ise sensör düğümleri, farklı üreticiler tarafından geliştirilmiş olan birden çok uygulamayı desteklemeli ve ağ davranışı yalnızca yazılım aracılığıyla yönetilebilmeli ve özelleştirilebilmelidir. Bu nedenle, SDN ve NFV destekledikleri ara yüzler ve soyutlamalar ile IoT tabanlı akıllı şehir çözümlerinin geliştirilmesinde önemli rol oynamaktadır [45].

IoT'nin en önemli uygulamalarından biri de araçsal tasarsız ağlardır (Vehicular Ad Hoc Network - VANET). Bir araçsal ağda, araçlar hem birbirleriyle hem de sinyal alıcı-verici ve baz istasyonlarından oluşan sabit ağ altyapısıyla iletişim kurmaktadırlar. Böylelikle, karayolu güvenliği veya trafik yönetim sistemi gibi katma değer hizmetler sağlanabilmektedir. Böyle bir ağda geleneksel ağ altyapısı ve SDN-NFV teknolojileri bir arada kullanılmaktadır. Araçlar ve yol kenarı birimleri, yönlendirme işlemleri için gerekli olan kontrol mesajlarını SDN ile tanımlanmış olan merkezi kontrolcü biriminden almaktadır. Kontrolcünün herhangi bir sebepten erişilemez olduğu durumlarda ise düğümler geleneksel tasarsız yönlendirme protokollerine geri dönerek gerekli işlemleri gerçekleştirmektedirler. Bu tür bir altyapı, ağ şartlarının ve düğüm konumlarının oldukça hızlı değiştiği aşırı akışkanlık gösteren araçsal ağlar için oldukça önemlidir, çünkü ağ bağlantılarının sürekli olarak yeniden hesaplanması işlemini son derece pratik hale getirmektedir [45, 46].

2.4.5. 5G

5G ağları için en önemli gereksinimlerden biri esnekliktir. Esnekliğin sağlanması; sunucular, kablolar ve diğer ağ kaynaklarını içeren fiziksel ağın yeniden tasarlanmasına gerek kalmadan yeni servislerin başlatılabilmesi ve uygulanabilmesi demektir. Esneklik aynı zamanda, gün içinde ve gece yarısı veya hafta içi ve hafta sonlarındaki değişen trafik miktarı ve özelliklerine karşı ağın kolaylıkla uyarlanabilir olması anlamına gelmektedir. SDN ve

NFV teknolojileri de 5G için tam olarak bu noktada sahneye çıkmaktadır. Bu teknolojiler sayesinde, sanal linkler ile birbirine bağlı sanal ağ fonksiyonlarından oluşan farklı ve esnek yapılarda mantıksal ağlar oluşturulabilmekte ve bu ağlar programlanabilir alt yapının üzerine kurulabilmektedir. Bu nedenle, SDN ve NFV, 5G'nin bel kemiğini oluşturan iki temel teknolojidir [47].

Esneklik ve ölçeklenebilirliğin yanı sıra, ultra-yüksek kapasite, ultra-düşük gecikme ve çok sayıda eş zamanlı oturumu destekleme yeteneği de 5G'nin en önemli gereksinimleri arasındadır. Bu gereksinimleri karşılayabilmek için, 5G'nin sunduğu en önemli çözüm ağ dilimlemedir (network slicing). Ağ dilimleme, aynı fiziksel ağ altyapısında sanallaştırılmış ve birbirinden bağımsız mantıksal ağların oluşturulmasını ve çoğullanmasını (multiplexing) sağlayan bir ağ mimarisidir. Ağ dilimleme çözümün temelini ise SDN ve NFV teknolojileri oluşturmaktadır. NFV ile oluşturulan her bir dilim, belirli uygulamalar tarafından talep edilen çeşitli gereksinimleri yerine getirmek için tasarlanmış uçtan uca bir ağdır. Ayrıca NFV, 5G ağ dilimleri üzerinde çalışan sanal fonksiyonların kaynak tüketiminin optimize edilmesini, bu fonksiyonların ölçeklenmesini ve sürekli olarak düzgün bir biçimde hizmet vermelerini de garanti etmektedir [48].

SDN ise merkezi kontrol katmanı sayesinde 5G ağındaki veri akışının kolay bir şekilde sağlanmasını ve kontrol edilmesini sağlamaktadır. Merkezi kontrol katmanı sayesinde optimum veri akışları gerçek zamanlı olarak belirlenmekte ve böylelikle büyük veri kesintilerinin önüne geçilmektedir. Bu da 5G'de gecikmenin en aza indirgenmesinde büyük rol oynamaktadır [47, 48].

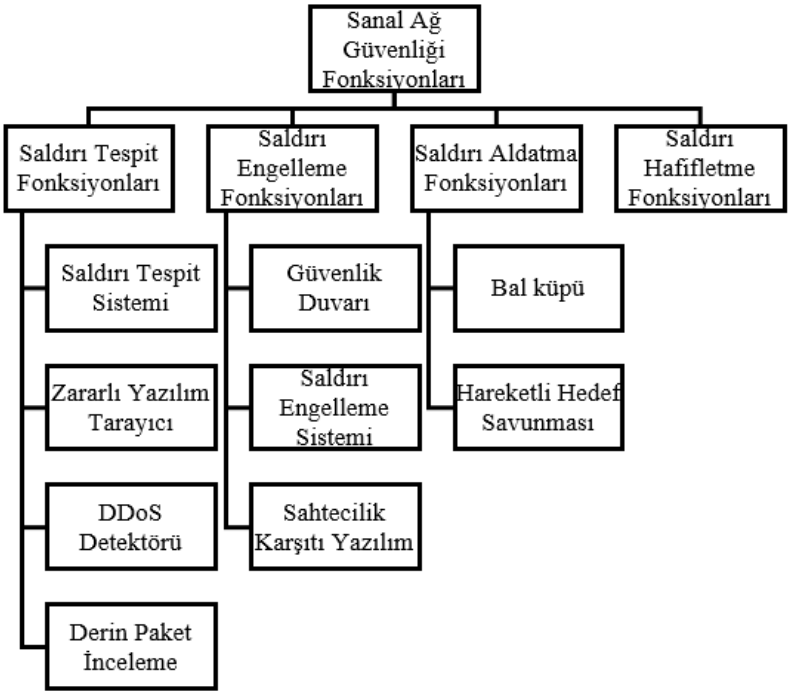
Teknolojinin her alanında olduğu gibi, SDN ve NFV teknolojileriyle yapılandırılmış ağlarda da güvenlik ele alınması gereken önemli bir problemdir. Bir sonraki bölümde bu tür ağlarda güvenliğin sağlanması için kullanılan sanallaştırılmış ağ güvenliği fonksiyonları anlatılacak ve bu fonksiyonların yerleştirilmesi problemi açıklanarak bu alanda yapılan çalışmaların sınıflandırılması için tez kapsamında geliştirilmiş olan taksonomi açıklanacaktır.

3. SANAL AĞ GÜVENLİĞİ FONKSİYONLARININ YERLEŞTİRİLMESİ ALANINDAKİ ÇALIŞMALARIN TAKSONOMİSİ

Geleneksel ağlarda; güvenlik duvarı, saldırı tespit/engelleme sistemi, derin paket inceleme vb. güvenlik fonksiyonları, ağda sabit noktalarda bulunan ve her biri üreticisine özgü olan (vendor-specific) donanımlar üzerinde çalışmaktadır [35]. SDN ve NFV teknolojileriyle yapılandırılmış ağlarda ise bu fonksiyonlar sanallaştırılmakta ve genel amaçlı sunucular üzerinde birer yazılım olarak çalıştırılmaktadır [11, 49]. Hu ve Ahn'a göre [35], sanallaştırılmış ağ güvenliği fonksiyonları; saldırı tespit (attack detection), saldırı engelleme (attack prevention) ve saldırı yakalama (attack capturing) fonksiyonları olmak üzere 3 başlık altında ele alınabilir. Bu tez çalışmasında ise, yeni bir sınıflandırma yaklaşımı önerilerek, Şekil 3.1'de de ifade edildiği gibi, sanal ağ güvenliği fonksiyonları; saldırı tespit (attack detection), saldırı engelleme (attack prevention), saldırı aldatma (attack deception) ve saldırı hafifletme (mitigation) fonksiyonları olacak şekilde 4 başlık altında ele alınmıştır. Önerilen sınıflandırmadaki saldırı aldatma fonksiyonları, saldırıyı engellemenin yanı sıra saldırganın kafasını karıştırarak yanlış yönlendirmeyi hedeflemektedir. Bu şekilde saldırganın kaynakları tüketilmekte ve aynı zamanda davranışı hakkında önemli bilgiler toplanabilmektedir. Saldırı hafifletme fonksiyonları ise; saldırının tam olarak engellenemediği durumlarda çeşitli fonksiyonları birlikte kullanarak saldırının etkisini azaltmaya çalışmaktadır. Bu nedenle önerilen sınıflandırmadaki bu iki kategorinin diğer sınıflardan ayrılabileceği görülmüştür. Devam eden alt başlıklarda, yukarıda belirtilen kategorilerin her biri açıklanarak bu kategorilere ait sanal güvenlik fonksiyonları tanıtılmış ve her bir sanal güvenlik fonksiyonu alanında yapılan son çalışmalara genel bir bakış sunulmuştur.

3.1. Sanal Güvenlik Fonksiyonları ve Sınıflandırılması

Bu bölümde, tez kapsamında saldırı tespit, saldırı engelleme, saldırı aldatma ve saldırı hafifletme fonksiyonları olacak şekilde 4 başlık altında sınıflandırılan sanal güvenlik fonksiyonları açıklanmaktadır.



Şekil 3.1. Sanal ağ güvenliği fonksiyonlarının sınıflandırılması amacıyla önerilen yaklaşım

3.1.1. Saldırı tespit (attack detection) fonksiyonları

Bu tez çalışması kapsamında saldırı tespit fonksiyonları; saldırı tespit sistemi, zararlı yazılım tarayıcı, DDoS detektörü ve derin paket inceleme olarak dört başlık altında ele alınmaktadır. Devam eden alt başlıklarda bu fonksiyonlar tanıtılmış ve bu alanda yapılan son çalışmalara genel bir bakış sunulmuştur.

Saldırı tespit sistemi (intrusion detection system - IDS)

Saldırı engelleme sistemleri, ağı izleyerek zararlı aktiviteleri tespit eden ve sistem yöneticilerine raporlayan pasif güvenlik fonksiyonlarıdır [50]. Geleneksel ağlardaki üzerindeki saldırı tespit sistemleri, imza-tabanlı, istatistik tabanlı ve protokol analizi yaklaşımlarını kullanırlar. Fakat bu tür sistemlerin kullandıkları veri; sistem kayıtları, servisler ve düğüm mesajlarından alındığı için sanal makinelere yapılan saldırıları tespit etmeleri oldukça zor ve maliyetlidir [51]. Ayrıca ağ büyüdükçe ve karmaşıklığı arttıkça yapılan analiz işlemi ve sonrasında trafiğin nasıl yönlendirileceği kararı performans açısından olumsuz sonuçlar doğurmaktadır [40]. SDN üzerinde yapılandırılmış IDS'lerde ise istatistikleri toplamak, anlamlandırmak ve trafiği uygun şekilde yönlendirmek yazılım

tabanlı mimariden dolayı çok daha kolay olmaktadır.

SDN'in bu avantajlarından faydalanarak Van Adrichem ve arkadaşları tarafından yapılan bir çalışmada [52], POX kontrolcü yazılımı üzerinde çalışan OpenNetMon adında bir saldırı tespit sistemi geliştirilmiştir. OpenNetMon ağdaki veri hacmi, paket kaybı ve gecikme metriklerine bakarak saldırı tespiti yapabilmektedir. SDN kontrolcüsü, merkezi yapısı sayesinde, büyük miktardaki analiz edilecek ve yorumlanacak trafik verisine erişim imkânına sahiptir. Tang ve arkadaşları tarafından yapılan bir çalışmada da [53] bu mantığı kullanarak kontrol katmanında çalışan derin öğrenme temelli bir saldırı tespit sistemi geliştirilmiştir.

Zararlı yazılım tarayıcı (malware scanner)

Zararlı yazılım tarayıcılar, yerel ağı internetteki virüs, solucan, truva atı vb. kötü amaçlı yazılımlardan koruyan ağ güvenliği fonksiyonlarıdır.

Ceron ve arkadaşları [54], SDN'in esnek yapısından faydalanarak bir zararlı yazılım analizi mimarisi önermişlerdir. Bu kapsamda, SDN kontrolcüsünün üzerinde bir uygulama olarak çalıştırılan denetleme birimi ile önceden tanımlanmış örüntüler kullanılarak ağ akışları analiz edilmektedir. Kötü amaçlı trafik tespit edildiğinde, bariyer modülü kötü amaçlı yazılımın diğer sistem elemanlarıyla iletişim kurmasını önlemekte ve yapılandırma yöneticisi ağ topolojisini değişen trafik özelliklerine göre dinamik olarak değiştirmektedir. Deneysel sonuçlar, geliştirilen bu sistemin ileri düzey kötü amaçlı yazılımları dinamik bir şekilde analiz edebildiğini ve geleneksel çözümlere kıyasla daha fazla zararlı yazılım tespit edebildiğini göstermektedir.

DDoS detektörü

Ağ güvenliğini tehdit eden en önemli saldırı türlerinden biri de dağıtık hizmeti engelleme (Distributed Denial of Service - DDoS) saldırılarıdır.

DDoS saldırılarında, hedef bilgisayara işleyemeyeceği boyutta sahte istek gönderilerek hizmetini engellemek ve böylelikle gerçek istekleri de işleyemeyecek hale getirilerek servis dışı bırakılması amaçlanmaktadır. Geleneksel ağlarda DDoS saldırılarını tespit edebilmek

için çok çeşitli çözümler mevcuttur, fakat bunların çoğunda, çok sayıda paketin analiz edilmesi gerektiğinden yanıt süresi arttıkça doğruluk seviyesi düşmektedir [55, 56].

SDN'de ise çok sayıda anahtar, kontrolcü yazılımı ile çok sayıda anahtar eş zamanlı olarak yönetilebildiğinden DDoS tespiti çok daha kısa sürede daha etkin bir şekilde yapılabilmektedir. Bu kapsamda, Braga ve arkadaşları [56], NOX kontrolcüsü üzerinde çalışan bir DDoS detektörü geliştirmişlerdir. SDN'in esnek yapısından faydalanılarak geliştirilen bu fonksiyon ile ağda DDoS tespitinden sorumlu anahtarlar kolay bir şekilde eklenip çıkarılabilmektedir. Ayrıca, geliştirdikleri sınıflandırıcı ile yeni saldırı türleri de sisteme kolaylıkla dâhil edilebilmektedir.

Derin paket inceleme (deep packet inspection - DPI)

DPI, trafik akışlarını ve kullanıcı aktivitelerini ayrıntılı ve gerçek zamanlı olarak analiz edebilmek için kullanılan gelişmiş bir ağ güvenliği fonksiyonudur. Bu fonksiyonlar ile paket başlıklarının yanı sıra veri kısmı da trafik türü, protokol uygunluğu ve zararlı yazılım içerip içermediği açısından kontrol edilmektedir. Paket içeriğinde şüpheli bir durum veya saldırı tehdidi algılandığında, paket başka bir yere yönlendirilmekte veya durum başka bir güvenlik fonksiyonuna raporlanmaktadır [1]. Böylelikle, ağın saldırılardan korunması, performansının artırılması, bant genişliği maliyetinin azaltılması, tıkanıklık denetiminin yapılması ve hizmet kalitesinin artırılması amaçlanmaktadır [57].

3.1.2. Saldırı engelleme (attack prevention) fonksiyonları

Bu tez çalışması kapsamında saldırı engelleme fonksiyonları; güvenlik duvarı, saldırı engelleme sistemi ve sahtecilik karşıtı yazılım olacak şekilde üç başlık altında ele alınmaktadır. Devam eden alt başlıklarda bu fonksiyonlar tanıtılmış ve bu alanda yapılan son çalışmalara genel bir bakış sunulmuştur.

Güvenlik duvarı (firewall)

Erişim kontrolünün ilk seviyesi olan güvenlik duvarı, yerel ağ ile internet arasına yerleştirilerek yerel ağı dışarıdaki güvenilir olmayan cihazlardan korumaktadır. Geleneksel ağlarda güvenlik duvarının yerleştirildiği sabit konumu dolayısıyla iç trafik görülememekte

ve denetlenememektedir. SDN'de ise bu güvenlik fonksiyonu, merkezi kontrolcü üzerinde bir uygulama olarak yapılandırıldığı için iç trafiğin tamamı görülür ve denetlenir. Paket filtreleme, yazılım tanımlı ağlarda güvenlik duvarı tasarımının sadece bir parçasıdır. Ağ şartları, yapılandırmalar ve akış kuralları dinamik bir şekilde değiştiği için güvenlik duvarı kurallarının akış kuralları ile olan uygunluğu da ayrıca denetlenmektedir. Bunların yanı sıra, güvenlik duvarı yerleştirme mimarisi (merkezi veya dağıtık) ve ağda hangi noktalara konumlandırılması gerektiği de dikkatli bir şekilde tespit edilmelidir [58, 59].

Geleneksel ağlarda, durum denetlemesiz (stateless) ve durum denetlemeli (stateful) olmak üzere iki tür güvenlik duvarı fonksiyonu kullanılmaktadır [59]. Durum denetlemesiz olanlarda paketler yalnızca paket başlığındaki IP adresi ve port numarasına bakılarak filtrelenirken, durum denetlemeli olanlarda bağlantının durumuna, iletişim protokollerine ve kaynak/hedef portlarına da bakılmaktadır. Bir başka ifadeyle, durum denetlemeli güvenlik duvarları, yalnızca öncesinde bilinen ve bağlantı kurulmuş bir akış üzerinden gelen paketleri kabul etmektedir [60]. Fakat SDN'deki güvenlik duvarı fonksiyonları durum denetlemesiz filtreleme yapmaktadırlar, çünkü OpenFlow kontrolcüye bağlantı hakkında oldukça kısıtlı bilgi sunmaktadır. Bu nedenle, SDN güvenlik duvarlarının tamamen durum denetlemeli hale getirilmesi ele alınması gereken önemli bir problemdir [58].

Literatürde, SDN için güvenlik duvarı fonksiyonu öneren birçok çalışma mevcuttur [58, 61, 62]. Bu doğrultuda Hu ve arkadaşları tarafından geliştirilen FlowGuard isimli güvenlik duvarı, ağ şartları değiştikçe, kural ihlallerini gerçek zamanlı ve otomatik bir şekilde tespit edebilmektedir [58]. Deng ve arkadaşları ise [27], geleneksel güvenlik duvarı yaklaşımının SDN'de etkin bir şekilde kullanılamayacağını ifade ederek sanal güvenlik duvarlarının NFV ile yönetilebilmesi için VNGuard adında bir çerçeve önermişlerdir. VNGuard, sanal güvenlik duvarlarının yüksek seviyeli kurallar üretebilmesi için bir dil tanımlamakta ve üretilen bu kurallar için optimum konumu bulan bir modül içermektedir.

Saldırı engelleme sistemi (intrusion prevention system - IPS)

Saldırı tespit sistemleri, saldırıyı yalnızca tespit etmekte ve raporlamaktadırlar, yani, saldırıyı başlangıç aşamasında veya başlamadan önce engelleyecek şekilde proaktif bir yapıya sahip değillerdir. Bu nedenle, şüpheli ağ aktivitelerine karşı hızlı bir şekilde harekete geçmek için saldırı engelleme sistemlerine ihtiyaç duyulmaktadır. Geleneksel IPS'ler,

saldırının önceden saptanmasına ihtiyaç duydukları için IDS fonksiyonları üzerine kurulmaktadırlar. Fakat böyle bir mimari esnek bir şekilde genişletilebilir değildir. Çoğu açık kaynak kodlu yazılımlar olsa da farklı kodlama teknikleri, geliştirme ortamları ve ara yüzleri bu sistemlerin etkin bir şekilde kullanımını zorlaştırmaktadır. Ayrıca geleneksel ağ altyapısı üzerinde kurulu IPS'lerin değişen ağ şartlarına dinamik çözümler üretmesi de oldukça zordur [51].

Yazılım tanımlı ağlarda ise, IPS fonksiyonları saldırıları çok daha çevik, dinamik ve düşük maliyetle tespit edebilmekte ve engelleyebilmektedirler. Literatürde SDN için IPS geliştirme üzerine yapılmış çalışmalar diğer güvenlik fonksiyonlarına göre sayıca daha azdır. Zhang ve arkadaşları [63], SDN tabanlı bir IPS ve yük dengeleme fonksiyonu geliştirmişlerdir. Deneysel sonuçlar, geliştirilen IPS'in saldırıları daha kısa sürede engelleyebildiğini ve yük dengeleme yaparak gecikmeyi azalttığını göstermektedir.

Sahtecilik karşıtı yazılım

Farklı ağ protokollerinde sahtecilik, saldırı tespitinin ve hafifletilmesinin engellenmesi amacıyla kötü niyetli kişiler tarafından sıklıkla kullanılan bir tekniktir. IP sahteciliğine (IP spoofing) karşı geliştirilmiş çözümler, genellikle, SDN kontrolcü uygulamaları tarafından kullanılan filtreleme tablolarının yanı sıra doğrulama kodları ve özetleme fonksiyonları gibi şifreleme yöntemlerinin kullanımını içermektedir [64, 65]. Benzer şekilde, ARP (Adress Resolution Protocol) sahteciliği saldırılarını engellemek için dinamik MAC-IP ilişkilendirme listelerini kullanarak gerçek zamanlı doğrulama yapan [66] ve ARP istek paketlerinin başlıklarını güvenli sahte değerlerle değiştirerek anahtarları önbellek zehirlenmesinden koruyan [67] SDN uygulama modülleri geliştirilmiştir.

3.1.3. Saldırı aldatma (attack deception) fonksiyonları

Tez çalışması kapsamında saldırı aldatma fonksiyonları; bal küpü ve hareketli hedef savunması olacak şekilde iki başlık altında ele alınmaktadır. Devam eden alt başlıklarda bu fonksiyonlar tanıtılmış ve bu alanda yapılan son çalışmalara genel bir bakış sunulmuştur.

Bal küpü

Bal küpü, bir ağa yapılan sızmaları ve saldırıları izleyebilmek, saldırı yöntemleri hakkında bilgi sahibi olmak ve yeni saldırı türlerini önceden tespit edebilmek amacıyla içerisine kasıtlı olarak açıklıkların yerleştirildiği ağ güvenliği fonksiyonudur. SDN'de ağ alt yapısı yazılım ile kontrol edildiğinden bal küpü gibi dinamik öğrenme ve trafik yönlendirmesi gereken sistemlerde geleneksel ağlara göre daha iyi çözümler üretilebilmektedir [68].

Bu doğrultuda Shin ve arkadaşları tarafından bal küpü fonksiyonunun da bulunduğu birçok güvenlik modülünden oluşan Fresco isimli bir çerçeve önerilmiştir. Fresco, zararlı bir bağlantı isteği tespit ettiğinde trafiği bal küpüne yönlendirmektedir. Böylelikle, saldırganlar gerçek hedefi ile haberleştiğini sanarak sisteme zarar verememektedir [69]. HoneyMix isimli bir diğer bal küpü fonksiyonu da SDN'in programlanabilirlik özelliğinden faydalanarak ağ trafiği üzerinde detaylı kontroller yapmakta ve saldırganların değerli hedefleri tehdit etmelerini engellemek için onları olabildiğinde uzun süre kendi üzerinde tutmaktadır.

Hareketli hedef savunması (moving target defense)

Saldırganlar, bir saldırıyı planlamadan ve gerçekleştirmeden önce ağ hakkında bilgi toplamaktadırlar. Hareketli hedef savunması, ağın yapısını gizleyerek veya saldırganı kasıtlı bir biçimde yanlış bilgilerle aldatarak bu keşif çabasının başarısını azaltmak için geliştirilmiş bir stratejidir. Bu amaçla, ana bilgisayarların gerçek IP adreslerini gizlemek ve yalnızca sıklıkla değiştirilen sanal IP adreslerini görünür hale getirmek için IP adresi mutasyon yöntemleri önerilmiştir [70, 71]. Yazılım tanımlı ağlarda, bu tür teknikler DNS çözümleyicileri ve programlanabilir anahtarlarla uyum içinde çalışan bir kontrolcü modülü olarak kolaylıkla uygulanabilmektedir. Bunların yanı sıra, sanal topolojilerden yararlanarak kullanıcıların ağa dair yanlış bilgilere sahip olmasını sağlayan farklı yöntemler de mevcuttur [72–74]. Bu tür kapsamlı saldırı aldatma çözümlerinin, NAT, DHCP, ARP ve DNS de dâhil olmak üzere birçok ağ hizmetinin dinamik olarak yapılandırılması yoluyla kaynak gizleme, adres mutasyonu, bal küpü ve sahte ağ bileşeni oluşturma gibi yaklaşımları kullanmaları gerekmektedir.

3.1.4. Saldırı hafifletme (attack mitigation) fonksiyonları

Saldırı hafifletme fonksiyonları, genel olarak, hedeflenen ağın hizmet vermesini engellemek amacıyla gerçekleştirilen DDoS saldırılarına karşı bir savunma katmanı olarak geliştirilmiş fonksiyonlardır. LFADefender [75], SDN'de link taşması (link flooding) saldırılarına karşı esnek ve maliyet etkin savunma mekanizması sunan bu tür bir fonksiyondur. Link taşması saldırıları, az sayıda kritik linki tıkamak için çok sayıda düşük hızlı akış kullanmaktadır. Bu saldırıları tespit etmek ve durdurmak zor olduğu için saldırılan ağ için maliyet açısından olumsuz sonuçlar doğurmaktadır.

LFADefender, akış yoğunluğunu ve tıkanıklık seviyelerini inceleyerek saldırganlar tarafından hedef alınan linkleri tam olarak belirlemek için sürekli bir biçimde linkleri izlemektedir. Bir linkte tıkanıklık algılandığında, trafiğin bir kısmını o linkten alıp diğer linklere yönlendirerek saldırıyı hafifletmektedir. Bu aşamadan sonra ise saldırıyı gerçekleştiren bot yazılımlarını tespit etmek ve tamamen engellemek için ileri düzeyde analiz yapmaktadır.

Çapraz ateş saldırısı (crossfire attacks) gibi [76] çoğu link taşması saldırısı, traceroute vb. araçları kullanarak linklerin kritiklik düzeyini tespit edebilmek amacıyla bir keşif aşamasıyla başlamaktadır. Aydeger ve arkadaşları [77], yazılım tanımlı ağları bu tür saldırılara karşı korumak amacıyla linklerin profillerini çıkaran ve saldırı durumunda trafiğin yeniden yönlendirilmesini sağlayan bir çözüm geliştirmişlerdir. Bu hafifletme yaklaşımı, öncelikle hangi linklerin saldırganlar tarafından hedef alındığını tespit etmekte ve daha sonra saldırganın kafasını karıştırmak için ICMP paketlerini rastgele seçilen diğer rotalara yönlendirmektedir.

3.2. Sanal Güvenlik Fonksiyonlarının Yerleştirilmesi Problemi

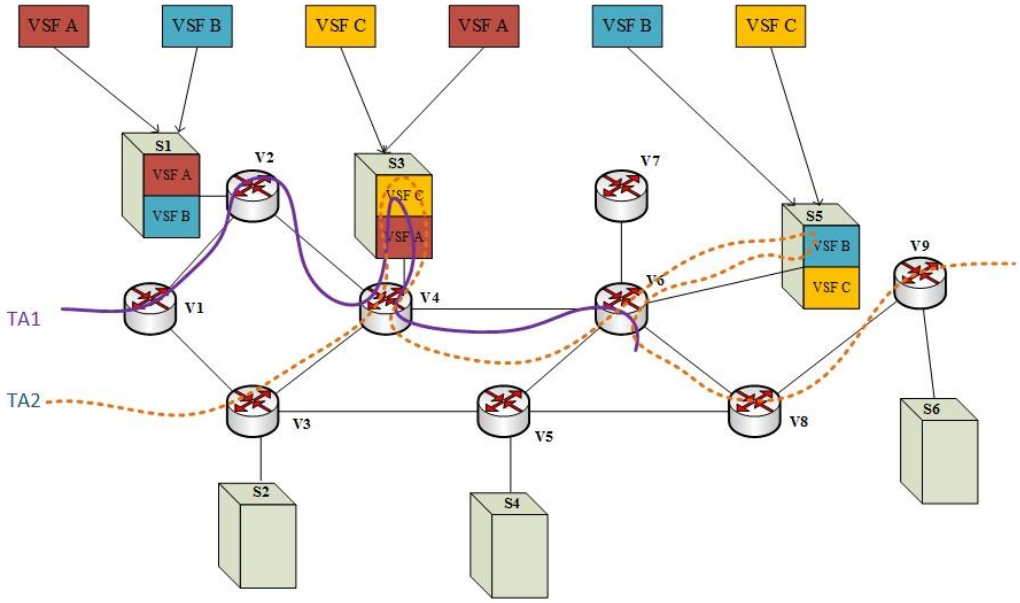
Daha önceki bölümlerde de bahsedildiği gibi, SDN ve NFV teknolojileriyle yapılandırılmış ağlarda ağ güvenliği; IDS, IPS, DPI, güvenlik duvarı vb. güvenlik fonksiyonlarının sanallaştırılması ve genel amaçlı sunucular üzerinde birer yazılım olarak çalıştırılmasıyla sağlanmaktadır. Fakat bu fonksiyonların ağda nerelere yerleştirileceği ele alınması gereken önemli problemlerden biridir. Çünkü, yerleştirme yapılırken güvenlik zafiyetine sebebiyet vermemenin yanı sıra, o ağ için erişilmesi gereken, enerji tüketiminin minimize edilmesi

maliyet etkin bir yerleştirme yapılması veya ağ yükünün ve gecikmenin en aza indirgenmesi gibi farklı operasyonel amaçların da göz önünde bulundurulması gerekmektedir [78]. Bu kapsamda VSF yerleştirme problemi, ağ kaynaklarını etkin ve etkili bir şekilde kullanarak VSF'ler için en uygun yerlerin bulunması ve trafik akışlarının yönlendirme gereksinimleri (forwarding requirements) karşılanacak şekilde bu VSF'lere en uygun şekilde atanması olarak tanımlanmaktadır. Problemi ağ güvenliği açısından daha açık bir şekilde ifade etmek gerekirse, bir adet DPI ve bir adet güvenlik duvarının yerleştirilmesi gereken bir ağda; DPI güvenlik duvarının önüne yerleştirilirse, ağa gelen tüm trafik akışları güvenlik duvarı tarafından filtrelenmeden derinlemesine incelemeye tabii tutulacaktır. Dolayısıyla, güvenlik duvarı tarafından ağa girmesine izin verilmeyecek olan akışlar da derinlemesine incelenerek DPI fonksiyonunun kaynakları israf edilecektir. Güvenlik duvarı, DPI fonksiyonunun önüne yerleştirildiğinde ise ağa girmesine izin verilmeyen akışlar elimine edilecek, yalnızca izin verilen akışlar derinlemesine incelenecektir.

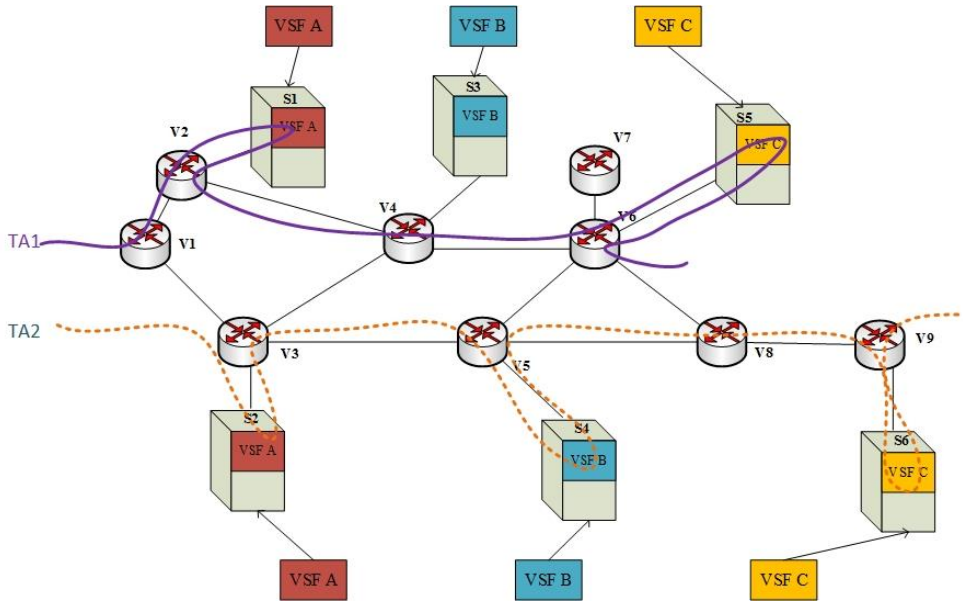
Problemi ağdaki operasyonel amaçlar açısından daha iyi açıklamak amacıyla, Şekil 3.2.'de enerji tüketimi ve ağ yükü arasındaki ödünleşimi (trade-off) gösteren iki senaryo verilmiştir. Şekilde S sunucuları, V anahtarları, TA ise trafik akışlarını temsil etmektedir. TA1, A ve C türündeki VSF'lerden geçecek şekilde V1'den V6'ya; TA2 ise A, B ve C türündeki VSF'lerden geçecek şekilde V3'ten V9'a varmak istemektedir.

Şekil 3.2 (a)'da verilen Senaryo 1'de gösterildiği gibi, bir sunucunun kapasitesi dolana kadar VSF'leri yerleştirerek dolduğunda yeni bir sunucu açılması ile kullanılan sunucu sayısı minimize edilecektir. Fakat bu senaryoda her ne kadar enerji tüketimi en aza indirgense de, link ve fonksiyon kapasiteleri açısından sunucuların üzerindeki yük artmış olacaktır. Öte yandan, Şekil 3.2 (b)'de verilen Senaryo 2'de ifade edildiği üzere, VSF'ler sunuculara dağıtıldığında ağ yükü Senaryo 1'e göre daha dengeli olacaktır, fakat bu durumda da daha çok sunucu aktif hale getirileceğinden enerji tüketimi artacaktır.

Görüldüğü gibi, VSF yerleştirme işleminde her duruma uygun tek bir optimum çözüm yoktur. Ağın yapısı, sunucuların kapasitesi, ağ giderleri için harcanabilecek bütçe ve kullanıcı deneyimi ile ilgili beklentiler gibi çözümün uygunluğunu etkileyen birçok faktör bulunmaktadır. Yukarıda verilen senaryolarda yalnızca enerji tüketimi ve yük dengeleme amaçları arasındaki ödünleşim verilse de, maliyet ve gecikmenin minimize edilmesi gibi farklı operasyonel amaçlar için farklı senaryolar ortaya çıkacaktır.



(a)



(b)

Şekil 3.2. VSF yerleştirme senaryoları (a) Senaryo 1, (b) Senaryo 2

Bu nedenle, sanal güvenlik fonksiyonu yerleştirilmesi probleminde ağa gelen trafiğin güvenlik gereksinimlerinin karşılanmasının yanı sıra, ağın operasyonel ihtiyaçlarına da cevap veren çözümlerin geliştirilmesi hayati öneme sahiptir [3, 79–85].

3.3. Geliştirilen Taksonomi

Bu bölümde, SDN ve NFV teknolojileriyle yapılandırılmış ağlarda sanal güvenlik fonksiyonlarının yerleştirilmesi alanındaki çalışmaların sınıflandırılması için geliştirilen taksonomi sunulmuştur. Şekil 3.3'te gösterilen taksonomiye göre literatürdeki VSF yerleştirme çözümleri; geliştirilen yerleştirme yaklaşımları, ele alınan operasyonel amaçlar, kullanılan optimizasyon yöntemleri, akış yönetimi stratejileri, uygulama alanları, kullanılan veri, optimizasyon araçları ve simülasyon/emülasyon ortamları olmak üzere 8 boyutta sınıflandırılmıştır. Aşağıdaki alt başlıklarda, bu boyutların her biri detaylı bir biçimde açıklanmaktadır.

3.3.1. Yerleştirme yaklaşımları

Yerleştirme yaklaşımı, fonksiyonları yerleştirirken trafik matrisinin göz önünde bulundurulup bulundurulmadığını ifade etmektedir. Koordinesiz (uncoordinated) yerleştirme yaklaşımında, yerleştirme problemi trafik akışlarının rotalarından ve yönlendirilmesinden bağımsız olarak ele alınmaktadır [1–4, 11–13]. Daha açık bir ifadeyle, önce fonksiyonlar yerleştirilmekte, sonrasında ise trafik akışları, konumları belirlenmiş olan fonksiyonlar üzerinden kendi ihtiyaçları doğrultusunda yönlendirilmektedir. Ancak bazı çalışmalar, trafikten bağımsız bir şekilde yerleştirme yapmanın etkin olmayan sonuçlar doğuracağını ifade etmektedir. Bu nedenle, bu tez kapsamında koordineli yerleştirme olarak isimlendirilmiş olan, trafik matrisini de göz önünde bulundurarak ağın operasyonel amaçlarına uygun şekilde yerleştirme yapan çözümler geliştirilmiştir [6, 7, 9]. Literatürdeki mevcut VSF yerleştirme çözümlerinin yerleştirme yaklaşımı açısından dağılımı incelendiğinde de, çalışmaların çoğunda koordineli yerleştirme çözümlerinin geliştirildiği görülmektedir.

3.3.2. Operasyonel amaçlar

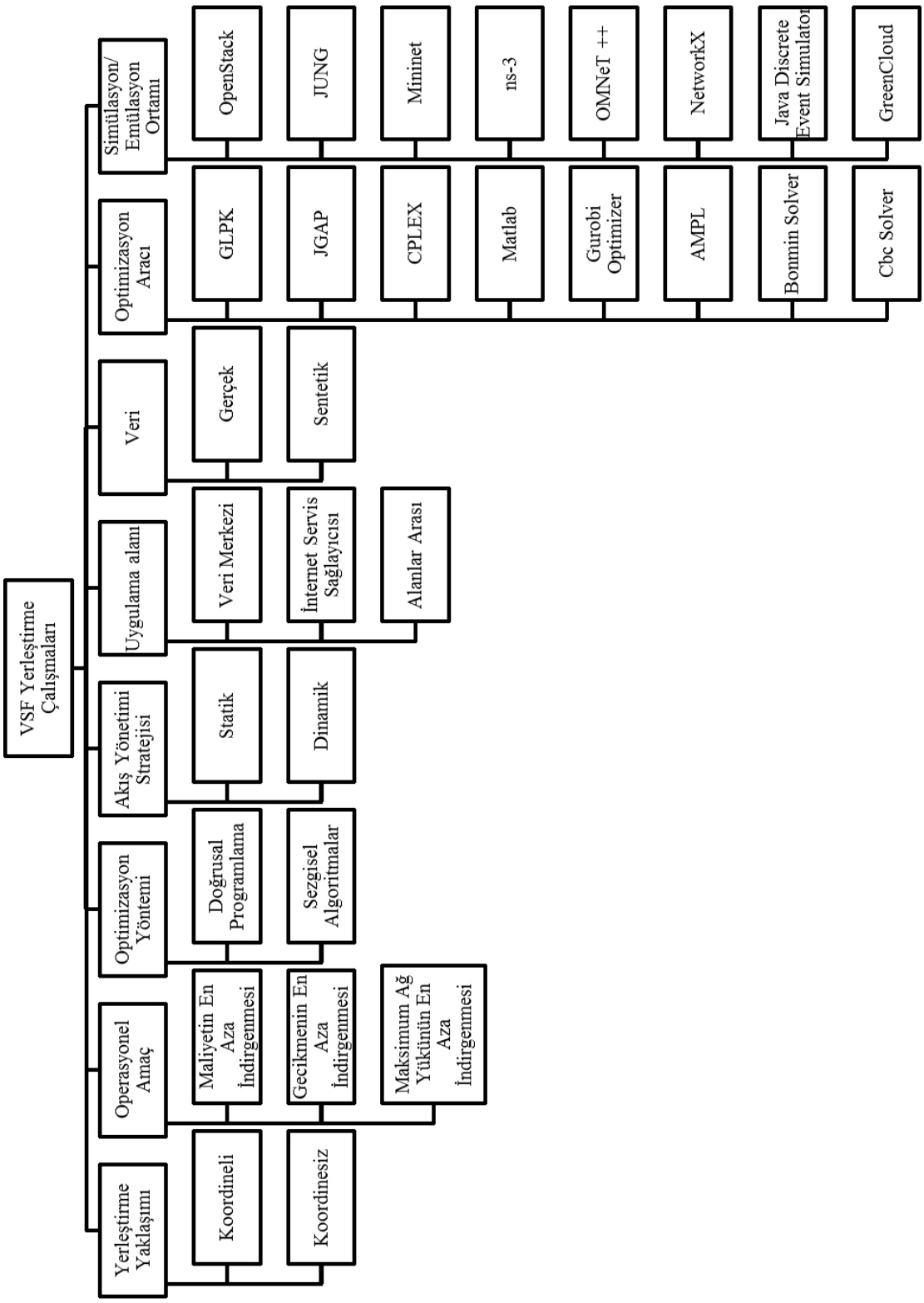
VSF yerleştirme çalışmalarında ele alınan operasyonel amaçlar, maliyetin, [1-4, 6-13], gecikmenin [4, 7, 12] ve maksimum ağ yükünün [1-3] en aza indirgenmesi olarak bu tez kapsamında üç başlık altında sınıflandırılmaktadır. Literatürdeki çalışmaların büyük çoğunluğunda maliyetin en aza indirgenmesi ele alınırken, nispeten daha az sayıda çalışmada gecikmenin ve maksimum ağ yükünün en aza indirgendiği görülmüştür. Bu

operasyonel amaçları tanımlarken kullanılan parametreler de bu tez kapsamında sınıflandırılarak Şekil 3.4'te sunulmuştur. Bölüm 3.4, 3.5 ve 3.6'da tartışılacak olan çalışmalar da bu sınıflandırma doğrultusunda ele alınacaktır.

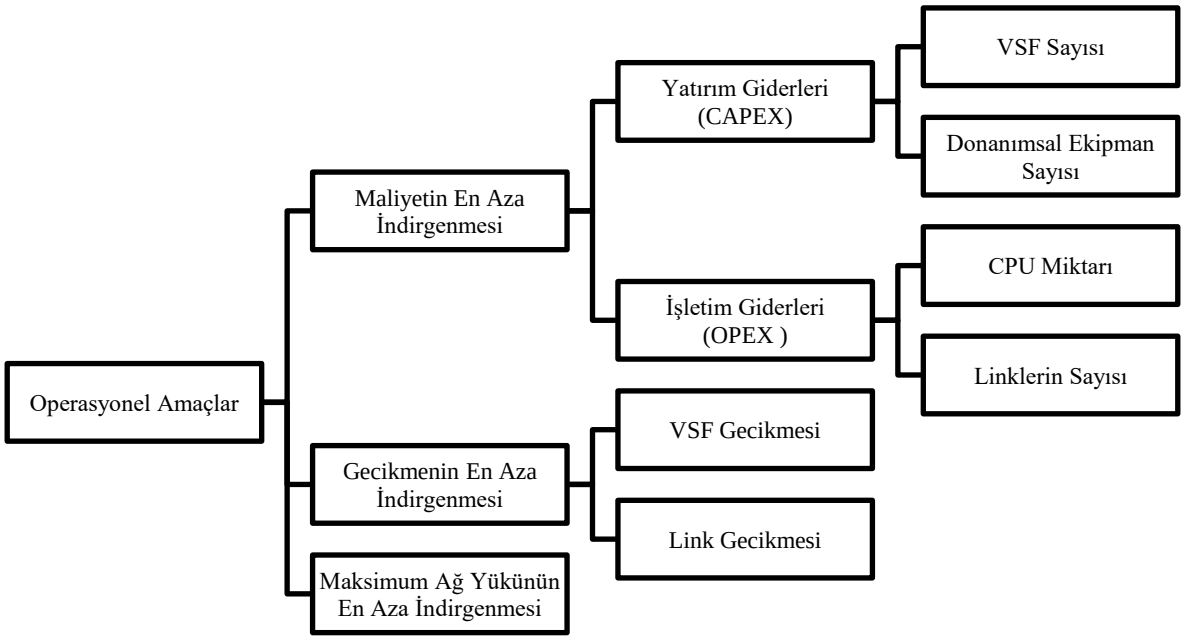
3.3.3. Optimizasyon yöntemleri

VSF yerleştirme çalışmalarında kullanılan optimizasyon yöntemleri bu tez kapsamında iki başlık altında sınıflandırılmıştır. Bunlar, doğrusal programlama [3-7, 10, 11, 13] ve sezgisel algoritmalardır [1-3, 6, 7, 10, 12, 13]. Doğrusal programlama, optimizasyon amacının doğrusal bir amaç fonksiyonu olarak tanımlandığı bir optimizasyon yöntemidir. Bu yöntemde kısıtlar da doğrusal eşitlikler veya eşitsizlikler olarak ifade edilmektedir. Tam sayılı doğrusal programlama (Integer Linear Programming - ILP), optimum çözümleri bulmak amacıyla en çok kullanılan doğrusal programlama yöntemidir. Bazı çalışmalarda tam sayılı programlama (Integer Programming - IP) olarak da anılan ILP modellerinde tüm karar değişkenlerinin tam sayı olarak ifade edilmektedir [86]. Literatürdeki çalışmalar incelendiğinde, VSF'lerin sunucularla eşleştirilmesi ve trafik akışlarının linklere atanması gibi kısıtların modellenmesinde ikili değişkenlerin yeterli olduğu görülmüştür.

Yukarıda bahsedilen matematiksel modelleme yöntemi optimum çözümleri bulmakta, fakat VSF problemi NP-zor sınıfında yer aldığı için büyük ölçekli ağlarda makul zaman aralığında çözüm üretememektedir. Bu nedenle araştırmacılar tarafından çözüm uzayındaki alternatifleri derecelendirerek her adımda daha iyi çözüme ulaşmaya çalışan sezgisel algoritmalar geliştirilmiştir. Bu algoritmalar optimum çözümü garanti etmeseler de makul zaman aralığında optimuma yakın çözümler üretmektedirler. Literatürdeki VSF yerleştirme çalışmalarının kullandıkları optimizasyon yöntemi bakımından dağılımı incelendiğinde, matematiksel modelleme ve sezgisel algoritma yöntemlerinin ikisinin de oldukça yaygın bir şekilde kullanıldığı görülmüştür.



Şekil 3.3. VSF yerleştirme çalışmalarının sınıflandırılması için geliştirilen taksonomi



Şekil 3.4. VSF yerleştirme çözümlerinde kullanılan operasyonel amaçların sınıflandırılması

3.3.4. Akış yönetimi stratejileri

Bu tez kapsamında, akış yönetimi stratejileri statik [1-5, 7-11] ve dinamik [6, 12, 13] olarak sınıflandırılmıştır. Statik olan akış yönetimi stratejisinde, VSF'ler optimum konumlarına yerleştirilir ve gelen trafik bu fonksiyonlar üzerinden yönlendirilir. Yani, VSF'lerin yerleri yeni trafik akışlarının gereksinimleri doğrultusunda değişmemektedir. Fakat trafik akışlarının zamana bağlı olarak önemli ölçüde farklılaştığı şartlarda bu strateji etkin bir biçimde kullanılamayabilir. Bu nedenle bazı çalışmalarda fonksiyon yerlerinin trafiğin ihtiyaçları doğrultusunda değiştirildiği dinamik akış yönetimi stratejisi önerilmiştir. Ayrıca bu stratejiye göre, trafiğin ihtiyaçlarına bağlı olarak yeni fonksiyonlar devreye alınabilmekte veya ağdan çıkarılabilmektedir. Literatürdeki VSF yerleştirme çalışmaları akış yönetimi stratejileri bakımından değerlendirildiğinde, yalnızca birkaç çalışmada dinamik strateji önerildiği görülmüştür [6, 12, 13].

3.3.5. Uygulama alanları

Taksonomideki bu boyut, literatürdeki VSF yerleştirme çözümlerinin ne tür ağlar için geliştirildiğini ifade etmektedir. Bu doğrultuda, geliştirilen çözümlerin uygulama alanları veri merkezleri [2, 4, 6, 9, 10, 11, 13], İnternet Servis Sağlayıcısı (Internet Service Provider - ISP) ağları [3, 5, 6, 8] ve alanlar arası (cross domain) [7] olarak sınıflandırılmıştır. Geliştirilen VSF yerleştirme çözümlerinin uygulama alanlarına göre dağılımı incelendiğinde, birincil uygulama alanının veri merkezleri olduğu ve bunu internet servis sağlayıcılarının takip ettiği görülmüştür. Diğer alanlarda uygulanan çözümlerin sayısı ise yok denecek kadar azdır.

3.3.6. Veri

Geliştirilen çözümleri test etmek amacıyla kullanılan trafik verisi, bu tez kapsamında, gerçek veri ve trafik üreteçleri tarafından üretilen sentetik veri olarak sınıflandırılmıştır. Internet2 [87] ve GEANT [88] en çok kullanılan gerçek veri setleri arasında yer almaktadır. Yapılan çalışmaların kullandıkları veri türü açısından dağılımı incelendiğinde, çalışmaların çoğunda sentetik veri kullanıldığı ve gerçek veri seti kullanılan çalışmaların oldukça az sayıda olduğu görülmüştür.

3.3.7. Optimizasyon araçları

Optimizasyon araçları, geliştirilen matematiksel modellerin çözümlenmesinde kullanılan geliştirme ortamlarını ifade etmektedir. CPLEX, GLPK, Gurobi Optimizer yapılan çalışmalarda en çok kullanılan model çözücüler arasında yer almaktadır [1–13].

3.3.8. Simülasyon/emülasyon ortamları

Son olarak, simülasyon/emülasyon araçları önerilen sezgisel algoritmaların performansını değerlendirmek ve sonuçları analiz ederek görselleştirmek amacıyla kullanılan ortamları ifade etmektedir. OpenStack, Mininet, ns-3, OMNET++ ve Java Discrete Event Simulator VSF yerleştirme çalışmalarında en çok tercih edilen simülasyon/emülasyon ortamları arasında yer almaktadır [1–13].

3.4. Maliyeti En Aza İndirgeyen Çözümler

Fonksiyonların üzerinde çalıştığı donanım maliyeti NFV ile elimine edilse bile, sanallaştırılmış fonksiyonların ağ üzerinde mantıklı ve makul bir şekilde konumlandırılması, maliyet etkin ağ çözümleri geliştirilmesinde büyük rol oynamaktadır. Örneğin, ağ operatörü performans ve bütçe gereksinimleriyle örtüşecek şekilde tam olarak ne kadar VSF'e ihtiyaç duyulduğunu ve bunların ağ üzerinde nerelere konumlandırılacağını planlarsa gereksiz VSF satın alma işlemi önlenir. Böylelikle, etkin bir yerleştirme stratejisi ile fazla kaynak tedariki engellenerek ağ yönetimi maliyeti düşürülmektedir [89].

Literatürde maliyet araştırmacılar tarafından farklı şekillerde tanımlanmıştır. Daha açık bir ifadeyle, maliyeti azaltmak için, bazı çalışmalarda yazılım lisanslama ücretlerinin minimize edilmesi amaçlanırken, bazı çalışmalarda kullanılan ağ kaynaklarının en aza indirgenmesi hedeflenmiştir. Bu doğrultuda, bu tez kapsamında, maliyeti tanımlamak için kullanılan parametreler, Şekil 3.4'te de ifade edildiği üzere, yatırım giderleri (capital expenditures - CAPEX) ve işletim giderleri (operational expenditures - OPEX) olmak üzere iki başlık altında toplanmıştır. Yatırım giderleri, satın alınan donanımlar, yazılım lisanslama ve kurulum ücretleri gibi altyapısal maliyetleri içerirken, işletim giderleri, planlama, yeniden yapılandırma ve ağ kaynaklarının kullanımı gibi ağın işleyişiyle ilgili maliyetleri içermektedir [90, 91].

Bu bölümde, maliyeti minimize etmeye çalışarak VSF yerleştirme yapan çalışmalar ele alınmaktadır. İlgili çalışmalar, yukarıda açıklanan maliyet parametreleri kapsamında sınıflandırılarak Şekil 3.3'teki geliştirilen taksonominin boyutları ışığında tartışılmış ve karşılaştırılmıştır. Çizelge 3.1, geliştirilen taksonomideki boyutlara göre karşılaştırılan çalışmaları listelemektedir.

3.4.1. Yatırım giderlerinin (CAPEX) en aza indirgenmesi

CAPEX'in en önemli bileşenlerinden biri, yazılım lisanslama ücretlerinden dolayı VSF'lerin sayısıdır. Kullanılan VSF sayısı minimize edilerek bu fonksiyonların satın alımlarına ve sunuculara kurulmasına harcanan bütçe azaltılmaktadır. Bu doğrultuda, Murukan ve Jamaluddine [2] fonksiyonlar arasındaki sıralama kısıtlarını da göz önünde bulundurarak genetik algoritma tabanlı bir yaklaşımla aktif hale getirilen VSF sayısını minimize

etmişlerdir. Fakat bu yaklaşımda, farklı VSF türlerinin güvenlik gereksinimleri ve kaynak tüketimi parametreleri gibi özel ihtiyaçları dikkate alınmamıştır.

3.4.2. İşletim giderlerinin (OPEX) en aza indirgenmesi

Shameli-Sendi ve arkadaşları [4], VSF'leri belirli bir sırada yerleştirirken hesaplama maliyeti ve gecikmeyi optimize etmeye çalışmışlardır. Trafiğin daha kısa sürede daha az düğüm üzerinden yönlendirilmesinin amaçlandığı bu çalışmada koordinesiz akış yönetimi stratejisi uygulanmıştır. Yani, yerleştirme problemi trafik akışlarının yönlendirilmesinden bağımsız olarak ele alınmıştır. Büyük ölçekli ağlarda makul sürelerde optimum yerleştirme yapılamamasına rağmen, önerilen algoritmanın OpenStack'te kullanılan mevcut yerleştirme algoritmasına göre daha iyi sonuçlar ürettiği görülmüştür.

Doriguzzi-Corin ve arkadaşları [5] VSF'leri yerleştirirken toplam bant genişliği, CPU ve hafıza gibi ağ kaynaklarının kullanımını minimize eden bir ILP modeli geliştirmişlerdir. Diğer çalışmalardan farklı olarak, maliyet optimizasyonu kısıtlarının yanı sıra güvenlik kısıtlarının da göz önünde bulundurulduğu çalışmada, ağdaki düğümler kısmen meşgulken ortalama gecikmenin 2-3 kat azaltıldığı gözlenmiştir. [7]'de ise yazarlar bu çalışmalarını genişleterek kabul edilebilir zaman aralığında optimuma yakın sonuçlar üretecek bir sezgisel algoritma önermişlerdir.

Dwiardhika ve Tachibana [8], VSF'leri maliyet etkin bir şekilde yerleştirirken sanal ağların güvenlik seviyesini artıracak bir yaklaşım önermişlerdir. Bu amaçla, her bir sanal düğüme bir güvenlik derecesi değeri atamış ve bunları istenilen güvenlik seviyesine ulaşılacak şekilde ağda konumlandırmışlardır.

Liu ve arkadaşları [12], ağdaki toplam kaynak kullanımını minimize ederken VSF'lerden kaynaklanan gecikmeyi de en aza indirgemişlerdir. Phan ve arkadaşları [10], maliyeti optimize edecek şekilde, değişken hacimdeki trafiğe hizmet verebilmek için yeni VSF'leri aktif hale getirebilecek proaktif bir yaklaşım önermişlerdir.

Çizelge 3.1. Maliyeti en aza indirgeyen sanal güvenlik fonksiyonu yerleştirme çalışmaları

Kaynak	Maliyet Parametreleri		Yerleştirme Yaklaşımı		Akış Yönetimi Stratejisi		Optimize Edilen Metrikler	Optimizasyon Yöntemi		Kullanılan Teknolojiler, Araçlar ve Veri	Uygulama Alanı
	CAPEX	OPEX	Koordineli	Koordinesiz	Statik	Dinamik		Matematiksel Model	Sezgisel Algoritma		
[1]	✓	✓		✓	✓		DPI sayısı, kullanılan bant genişliği		✓	JGAP [92]	-
[2]	✓	✓		✓	✓		DPI sayısı		✓	OpenStack [93], ODL [94]	Veri merkezi
[3]	✓	✓		✓	✓	✓	DPI sayısı, kullanılan bant genişliği	✓	✓	GEANT [88], GLPK [95], JUNG [96]	-
[4]		✓		✓	✓		Trafiğin linklerden geçmesi ve VSF' ler tarafından işlenmesi için gereken süre	✓		OpenStack [93], ODL [94], GLPK [95]	Veri merkezi
[5]		✓	-	-	✓		Kullanılan toplam bant genişliği, CPU ve hafıza	✓		-	Telekom servis sağlayıcısı
[6]		✓	✓			✓	Kullanılan toplam bant genişliği	✓	✓	Gurobi [97], Python [98], GARR [99]	Veri merkezi, Omurga ağı

Çizelge 3.1. (devam) Maliyeti en aza indirgeyen sanal güvenlik fonksiyonu yerleştirme çalışmaları

Kaynak	Maliyet Parametreleri		Yerleştirme Yaklaşımı		Akış Yönetimi Stratejisi		Optimize Edilen Metrikler	Optimizasyon Yöntemi		Kullanılan Teknolojiler, Araçlar ve Veri	Uygulama Alanı
	CAPEX	OPEX	Koordineli	Koordinesiz	Statik	Dinamik		Matematiksel Model	Sezgisel Algoritma		
[8]		✓	-	-	✓		Kullanılan toplam CPU ve bant genişliği	✓		-	-
[9]		✓	✓		✓		Kullanılan toplam CPU, hafıza ve disk	✓		-	Veri merkezi
[10]	✓	✓	-	-	✓		VSF sayısı ve kullanılan toplam bant genişliği	✓	✓	OpenStack [93], ODL [94], GLPK [95]	Veri merkezi
[11]	✓	✓	✓		✓		VSF ve linklerin sayısı	✓		OpenStack [93], ODL [94]	Veri merkezi
[12]		✓		✓		✓	Düğümler-deki maksimum kaynak bölünmesi	✓	✓	-	-
[13]		✓	✓			✓	Kullanılan toplam bant genişliği, CPU ve hafıza	✓	✓	GLPK [95]	Veri merkezi

Geliştirdikleri yaklaşımın kaynak kullanımını minimize edecek uygulanabilir çözümler ürettiğini gösterseler de, ağdaki kaynak kısıtlarını göz önünde bulundurmamışlardır.

[13]'de yazarlar kaynak tüketimini minimize etmeye çalışmış ve ağdaki güvenlik ve kaynak gereksinimi kısıtlarını da dikkate almışlardır. Gelen trafik akışlarının güvenlik seviyesini göz önünde bulundurmuşlar ve her bir VSF için bir güvenlik seviyesi tanımlamışlardır. Bir akışın VSF'lere atamasını yaparken, güvenlik seviyesi değerleri toplamı o akışın güvenlik seviyesine eşit olacak VSF'ler seçilmektedir. Dolayısıyla, trafik akışlarının geçmesi gereken VSF kümesi akışlara özgü bir biçimde değil de rastgele bir şekilde belirlenmektedir.

3.4.2. Yatırım ve işletim giderlerinin (OPEX) birlikte en aza indirgenmesi

Bouet ve arkadaşları [1], güvenlik kısıtlarını da karşılayacak şekilde DPI fonksiyonlarının maliyet etkin yerleştirilmesi üzerime çalışmışlardır. Bu amaçla yazarlar, DPI sayısını ve ağ yükünü aynı anda minimize edecek genetik algoritma tabanlı bir yaklaşım önermişlerdir. Fakat bu çözüm, geniş ölçekli ağlar için uygulanabilir olmadığından [1]'deki karmaşıklığı azaltarak çizge tabanlı yeni bir açgözlü sezgisel algoritma geliştirmişlerdir [3].

Jarraya ve arkadaşları [11], [2]'de ele alınan problem için koordineli yerleştirme yaklaşımını kullanan ve VSF'lerin önceliklerini de dikkate alarak uygun bir sırada yerleştirilmesini sağlayan OCDO (Ordered Cloud Defense Optimization) isminde bir çerçeve önermişlerdir. Geliştirdikleri bu yaklaşım ile büyük ölçekli veri merkezleri için bile maliyeti makul seviyede düşürebildikleri görülmüştür.

Shameli-Sendi ve arkadaşları tarafından yapılan son çalışmada [10], maliyet optimizasyonu kısıtlarının yanı sıra güvenlik kısıtlarını da göz önünde bulunduran SDPAP (Security Defense Patterns Aware Placement) isimli bir yaklaşım geliştirilmiştir. Bu çalışmanın diğer çalışmalardan temel farkı, ağ güvenliği uzmanları tarafından farklı yerleştirme çözümleri için tanımlanmış olan ağ güvenliği savunması örüntülerinin (network security defense patterns - NSDF) dikkate alınmasıdır. Böylelikle yazarlar, IDS'in çekirdek ağa yerleştirilmesi veya VPN'in (virtual private network) kenar (edge) ağlara yakın yerleştirilmesi gibi, VSF'lerin yanlış ve etkin olmayan bir biçimde konumlandırılmasını engellemektedirler.

3.5. Gecikmeyi En Aza İndirgeyen Çözümler

Günümüzde, internete bağlı cihaz sayısındaki artıştan dolayı, iletişim ağları üzerinden iletilen kullanıcı verilerinin önemli ölçüde artması gecikme problemini de beraberinde getirmiştir. Ağ fonksiyonlarının sanallaştırılmasıyla, donanım kaynaklarında görülen tek nokta arızası (single point of failure) ortadan kaldırılmakta, performans darboğazlarını azaltan ve ağ servislerinin hızlı bir şekilde etkinleştirilmesine olanak sağlayan daha ölçeklenebilir ağ mimarileri oluşturulmaktadır. Fakat bu tür mimarilerde de link gecikmelerinin ve sanal ağ fonksiyonlarının trafiği işlemelerinden kaynaklanan gecikmelerin en aza indirgenmesi gerekmektedir [91, 100].

Bu doğrultuda, bu tez kapsamında, gecikmeyi tanımlamak için kullanılan parametreler, Şekil 3.4'te de gösterildiği üzere, VSF gecikmesi ve link gecikmesi olmak üzere iki başlık altında sınıflandırılmıştır.

Bu bölümde, gecikmeyi minimize etmeye çalışarak VSF yerleştirme yapan çalışmalar ele alınmaktadır. İlgili çalışmalar, yukarıda açıklanan gecikme parametreleri kapsamında sınıflandırılarak Şekil 3.3'teki geliştirilen taksonominin boyutları ışığında tartışılmış ve karşılaştırılmıştır. Çizelge 3.2'de geliştirilen taksonomideki boyutlara göre karşılaştırılan çalışmalar listelenmiştir.

3.5.1. VSF gecikmelerinin en aza indirgenmesi

Liu ve arkadaşları [12], trafik akışları VSF'ler tarafından işlenirken meydana gelen gecikmeyi en aza indirgemişlerdir. Bu amaçla, akışları kaynak düğümünden varış düğümüne iletirken en az sayıda düğüm üzerinden geçirmeyi hedeflemişlerdir, çünkü akışların takip ettiği düğüm sayısı azaldıkça VSF'ler birbirlerine daha yakın noktalara yerleştirilecek ve böylelikle VSF'lerin trafiği işlemesinden doğan gecikme azaltılmış olacaktır.

Çizelge 3.2. Gecikmeyi en aza indirgeyen sanal güvenlik fonksiyonu yerleştirme çalışmaları

Kaynak	Gecikme Parametreleri		Yerleştirme Yaklaşımı		Akış Yönetimi Stratejisi		Optimize Edilen Metrikler	Optimizasyon Yöntemi		Kullanılan Teknolojiler, Araçlar ve Veri	Uygulama Alanı
	VSF Gecikmesi	Link Gecikmesi	Koordineli	Koordinesiz	Statik	Dinamik		Matematiksel Model	Sezgisel Algoritma		
[4]	✓	✓		✓	✓		Trafik akışlarının takip ettiği düğüm sayısı	✓		OpenStack [93], ODL [94], GLPK [95]	Veri merkezi
[7]		✓	✓		✓		Linklerdeki ortalama gecikme	✓	✓	GLPK [95], CERNET [101], GT-ITM [102]	Alanlar arası
[12]	✓			✓		✓	Trafik akışlarının takip ettiği düğüm sayısı	✓	✓	-	-

3.5.2. Link gecikmelerinin en aza indirgenmesi

Uçtan uca gecikmenin önemli bir bileşeni link gecikmesidir. VSF'lerin uygun olmayan bir biçimde yerleştirilmesi, kuyruk gecikmelerini artıran link tıkanıklıklarına yol açmakta ve aynı zamanda paket kaybından kaynaklanan yeniden iletim gecikmelerine sebep olmaktadır [100]. Bu doğrultuda Xu ve arkadaşları [7], VSF yerleştirme yaparken, trafiğin linklerden iletilmesi esnasında ortaya çıkan ve özellikle 5G ağları için oldukça önemli olan ortalama gecikme değerini en aza indirgemişlerdir.

3.5.3. VSF ve link gecikmelerinin birlikte en aza indirgenmesi

Shameli-Sendi ve arkadaşları [4], VSF'leri belirli bir sırada yerleştirirken gecikmeyi optimize etmeye çalışmışlardır. Trafik akışlarını daha kısa sürede daha az düğüm üzerinden yönlendirilerek hem VSF'lerden hem de trafiğin linklerde iletilmesinden kaynaklanan gecikme en aza indirgenmiştir. Geliştirilen algoritma, OpenStack'te kullanılan mevcut yerleştirme algoritmasına göre daha iyi sonuçlar üretirken, ağ topolojisi büyüdükçe optimum değerlere makul sürede ulaşılamamaktadır.

3.6. Ağ Yükünü En Aza İndirgeyen Çözümler

VSF'ler trafik akışlarını işlediğinden, ağda yük dengeleme yapmak için VSF'lerin trafiği işleme yükünün dengelenmesi gerekmektedir. Örneğin, akışların yolları üzerine bir tane IDS koymak yerine iki veya daha fazla yerleştirmek, bu fonksiyonların işlem yükünün dengelenmesini sağlayacaktır [81, 103]. Ayrıca, akışların VSF'ler arasında yeniden yönlendirilmesinden kaynaklanan link yükleri de dengelenmiş olacaktır [1]. Bu nedenle, VSF yerleştirme probleminde yük dengeleme ele alınması gereken önemli bir problemdir. Çizelge 3.3'te bu alanda yapılan çalışmalar listelenmiştir.

Bouet ve arkadaşları [1, 3], sanal DPI fonksiyonlarını maliyet etkin bir biçimde yerleştirmek için DPI sayısını en aza indirgemişlerdir. Fakat fonksiyon sayısı azaldıkça, trafiğin bu fonksiyonlar tarafından işlenmesi için geçmesi gereken yollarda yük arttığından, DPI'lar arasındaki mesafeyi kısaltarak yük dengeleme yapmayı hedeflemişlerdir. Murukan ve Jamaluddine de [2], bir ağda birden fazla güvenlik fonksiyonuna ihtiyaç duyulacağını

belirterek [1]'i genişletmişler ve yük dengeleme amacıyla ile birden fazla VSF türü için yerleştirme yapılması üzerine çalışmışlardır.

Sonuç olarak, literatürde mevcut olan VSF yerleştirme çalışmaları incelendiğinde, ağdaki operasyonel ihtiyaçlar doğrultusunda ele alınan optimizasyon amaçlarının maliyetin [1-4, 6-13], gecikmenin [4, 7, 12] ve ağ yükünün [1-3] en aza indirgenmesi olmak üzere üç başlık altında toplandığı görülmektedir. Çalışmaların büyük çoğunluğunda ise; (i) yerleştirilen VSF sayısı, (ii) donanımsal ekipman sayısı ve (iii) ağ kaynaklar kullanımının (bant genişliği, linklerin sayısı, CPU kullanımı vb.) bir kombinasyonu olarak tanımlanan maliyetin en aza indirgenmesi hedeflenmiştir. Fakat bu çalışmaların hiçbirinde, ağ maliyetinin en önemli bileşenlerinden biri olan ve sürdürülebilirlik ile yeşil bilişim (green computing) hedeflerine ulaşmak için kritik bir önlem olduğu bilinen enerji tüketimi ele alınmamıştır. Bunlara ek olarak, trafiği VSF'ler üzerinden yönlendirirken, akış bazında güvenlik gereksinimlerini göz önünde bulunduran bilgimiz dâhilinde herhangi bir çalışmaya rastlanmamıştır.

Bir sonraki bölümde, bu tez çalışması kapsamında modellenen enerji etkin sanal güvenlik fonksiyonu yerleştirme problemi açıklanmış, bu problemin çözümü için geliştirilen ILP modeli ve sezgisel algoritma açıklanmış ve yapılan deneysel çalışmalar sunulmuştur.

Çizelge 3.3. Ağ yükünü en aza indirgeyen sanal güvenlik fonksiyonu yerleştirme çalışmaları

Kaynak	Yerleştirme Yaklaşımı		Akış Yönetimi Stratejisi		Optimize Edilen Metrikler	Optimizasyon Yöntemi		Kullanılan Teknolojiler, Araçlar ve Veri	Uygulama Alanı
	Koordineli	Koordinesiz	Statik	Dinamik		Matematiksel Model	Sezgisel Algoritma		
[1]		✓	✓		VSF' ler arasındaki mesafe		✓	JGAP [92]	-
[2]		✓	✓		VSF' ler arasındaki mesafe		✓	OpenStack [93], ODL [94]	Veri merkezi
[3]		✓	✓		VSF' ler arasındaki mesafe	✓	✓	GEANT [88], GLPK [95], JUNG [96]	-

4. SANAL AĞ GÜVENLİĞİ FONKSİYONLARININ ENERJİ ETKİN YERLEŞTİRİLMESİ İÇİN MODEL VE ALGORİTMA GELİŞTİRME

Sanal güvenlik fonksiyonu yerleştirme probleminde ele alınması gereken en önemli operasyonel amaçlardan biri, ağdaki enerji tüketiminin minimize edilmesidir. Çünkü VSF'lerin üzerine kurulduğu sunucular ve trafiğin VSF'lerden yönlendirilmesini sağlayan ağ cihazları muazzam miktarda enerji tüketmektedir [104]. Tipik veri merkezlerinde ve Telekom ağlarında [105], sunucular tarafından tüketilen enerji miktarı, güç yönetimi teknikleri uygulansa bile, toplam enerji tüketiminin %50'sini oluşturmaktadır [106]. Özellikle büyük veri merkezleri için [107], sunucuların enerji tüketimini en aza indirgeyecek çözümlerin geliştirilmesi gerektiği aşikârdır.

Bunlara ek olarak, bazı ülkelerde çevresel sürdürülebilirliği sağlamak için çevreye yayılan karbondioksite vergi koyma uygulaması başlatılmıştır. Bu nedenle hizmet sağlayıcıların elektrik ve karbon giderlerini düşürmek amacıyla, bu tür ağlarda kaynak paylaşımı açısından etkin algoritmalar kullanarak tüketilen enerji miktarını minimize etmeleri gerekmektedir [34].

Sanal ağ fonksiyonlarının sunucular üzerinde çalıştırıldığı NFV teknolojisiyle yapılandırılmış ağlarda, bu fonksiyonların hangi sunuculara yerleştirileceği kaynakların etkin bir şekilde tüketilmesi ve enerji tüketiminin azaltılması açısından oldukça önemli bir etkiye sahiptir. Bu doğrultuda ETSI de, sanal ağ fonksiyonları yerleştirilirken kullanılan kaynak sayısını azaltarak enerji tüketimini optimize eden NFV çerçevelerine olan ihtiyacı vurgulamıştır [108]. Bu amaçla, sanal ağ fonksiyonlarının sayısı, nerelere konumlandırılacağı ve trafiğin bu fonksiyonlar üzerinden nasıl yönlendirileceği enerji etkin çözümler üretmek adına cevaplandırılması gereken sorulardır.

Bu doğrultuda, bu tez çalışmasında, ağa gelen trafik akışlarının güvenlik gereksinimlerini karşılayacak şekilde enerji tüketiminin en aza indirgenmesi amacıyla sanal ağ güvenliği fonksiyonlarının yerleştirilmesi problemi modellenmiştir. Daha açık bir ifadeyle; bu tez kapsamında ele alınan problem, verilen bir ağ topolojisi, VSF özellikleri (specifications) ve trafik matrisi için ağa gelen trafiğin güvenlik gereksinimlerini karşılarken sunucuların toplam enerji tüketimini en aza indirgemek amacıyla

- Her bir VSF türünden kaç tane kullanılacağı,
- Bu VSF'lerin hangi sunuculara yerleştirileceği,
- Her bir trafik akışının ihtiyaç duyduğu güvenlik hizmetini alacağı şekilde nasıl yönlendirileceği ve
- Her bir trafik akışının ihtiyaç duyduğu güvenlik hizmetini hangi konumlardaki VSF'lerden alacağının

belirlenmesi olarak tanımlanmaktadır.

Bu bölümün geri kalanında; enerji etkin sanal güvenlik fonksiyonu yerleştirme problemi için geliştirilen ILP modeli ve sezgisel algoritma sunulmuş ve önerilen çözümlerin performansını değerlendirebilmek için elde edilen deneysel sonuçlar verilmiştir.

4.1. Geliştirilen ILP Modeli

Bu tez çalışmasında ele alınan problem ILP ile modellenmiştir. Devam eden alt başlıklarda geliştirilen ILP modeli detaylandırılmaktadır.

4.1.1. Fiziksel ağ

$G = (V, E)$ şeklinde gösterilen fiziksel ağ, V ve E'nin sırasıyla anahtarları ve bunlar arasındaki linkleri gösterdiği yönsüz bir çizge (undirected graph) olarak ifade edilmektedir. Ağdaki her bir anahtar V'ye genel amaçlı bir sunucunun bağlanabileceği ve bu sunuculardan her birine bir veya daha fazla VSF'in yerleştirilebileceği bilinmektedir. S kümesi, sunucuları temsil etmekte ve aşağıda ifade edilen ikili değişken $h_{v,i} \in \{0,1\}$, S kümesinde yer alan sunucu i'nin V kümesinde yer alan anahtar v'ye bağlı olup olmadığını göstermektedir.

$$h_{v,i} = \begin{cases} 1, & eğer\ sunucu\ i \in S, anahtar\ v \in V'ye\ bağlı\ ise \\ 0, & değilse \end{cases} \quad (4.1)$$

4.1.2. Sanallaştırılmış ağ güvenliği fonksiyonları

Bir ağa, güvenlik duvarı, IDS, IPS, DPI ve zararlı yazılım tarayıcı gibi farklı türde sanal güvenlik fonksiyonları yerleştirilebilmektedir. Bu doğrultuda, A kümesi muhtemel VSF türlerini temsil etmektedir. t türündeki bir VSF, sırasıyla K_t^{CPU}, c_t ve uc_t olarak temsil

edilen, CPU gereksinimi, trafik işleme kapasitesi ve o VSF için yazılım lisanslama ücreti olarak tanımlanan birim maliyeti değerlerine sahiptir. Burada göz önünde bulundurulması gereken önemli noktalar şunlardır:

- Her sunucu her türdeki sanal güvenlik fonksiyonunu çalıştıramayabilir. Daha açık bir ifadeyle, belirli VSF türlerinin özel donanımsal gereksinimleri olabilmektedir (DPI için donanım hızlandırmalı şifreleme (hardware accelerated encryption) gerekliliği vb.)
- Öte yandan, ağ operatörlerinin belirli VSF'lerin belirli sunucularda çalıştırılması gibi özel tercihleri olabilmektedir (güvenlik duvarı fonksiyonunu kenar sunucularda çalıştırılması vb.)

Bu nedenle, sunucu i'nin t türündeki VSF'i çalıştırıp çalıştıramayacağını ifade etmek üzere $w_{i,t} \in \{0,1\}$ ikili değişkeni tanımlanmıştır.

$$w_{i,t} = \begin{cases} 1, & eğer\ sunucu\ i \in S, t \in \text{A türündeki VSF'i çalıştırabiliyorsa} \\ 0, & değilse \end{cases} \quad (4.2)$$

4.1.3. Trafik isteği

Bir akışın trafik isteği $f =< S^{\mathrm{f}},\ D^{\mathrm{f}},\ \Psi^{\mathrm{f}},\ B^{\mathrm{f}}>$ şeklinde dörtlü bir değişken olarak tanımlanmaktadır. Burada S^{f} ve D^{f} akışın kaynak ve varış düğümünü, Ψ^{f} geçmesi gereken VSF türleri kümesini ve B^{f} bant genişliği gereksinimini ifade etmektedir. Her trafik akışının güvenlik gereksinimi farklı olacağından Ψ^{f} her bir akış için farklı olabilmektedir.

4.1.4. ILP Formülasyonu

Geliştirilen ILP modelinde kullanılan değişkenler Çizelge 4.1'de verilmektedir.

Geliştirilen modelde kullanılan karar değişkenleri ise aşağıda tanımlandığı gibidir:

$$x_{it} = \begin{cases} 1, & \text{eğer } t \text{ türündeki VSF sunucu } i'\text{ye yerleştirildiyse} \\ 0, & \text{değilse} \end{cases} \quad (4.3)$$

$$y_i = \begin{cases} 1, & \text{eğer sunucu } i \text{ kullanıldıysa} \\ 0, & \text{değilse} \end{cases} \quad (4.4)$$

Çizelge 4.1. ILP modelinde kullanılan değişkenler

Değişken	Tanım
$\boldsymbol{S}$	Sunucuların kümesi
$\boldsymbol{F}$	Güvenlik gereksinimlerine göre sanal ağ fonksiyonlarından tarafından işlenerek ağdan geçmesi gereken trafik akışlarının kümesi
$\boldsymbol{A}$	Sanal ağ güvenliği fonksiyonu türlerinin kümesi
$\boldsymbol{V}$	Fiziksel ağdaki anahtarların kümesi
$\boldsymbol{e_i}$	Sunucu $\boldsymbol{i}$'nin enerji tüketimi
$\boldsymbol{e_{i,max}}$	Sunucu $\boldsymbol{i}$*'nin* maksimum enerji tüketimi
$\boldsymbol{e_{i,idle}}$	Sunucu $\boldsymbol{i}$*'nin* boş durumdaki (idle state) enerji tüketimi
$\boldsymbol{c_{u,v}}$	u-v linkinin trafik kapasitesini temsil eden pozitif tam sayı
$\boldsymbol{c_i^{CPU}}$	Sunucu $\boldsymbol{i}$*'nin* CPU kapasitesini temsil eden pozitif tam sayı
K_t^{CPU}	t türündeki sanal güvenlik fonksiyonunun CPU gereksinimini temsil eden pozitif tam sayı
c_t	t türündeki sanal güvenlik fonksiyonunun trafik işleme kapasitesini (maksimum akış sayısı) temsil eden pozitif tam sayı
uc_t	t türündeki sanal güvenlik fonksiyonunun birim maliyetini temsil eden pozitif tam sayı
$budget$	Ağ yöneticisi tarafından belirlenen fonksiyonlara ayrılacak toplam bütçe
$h_{v,i}$	Sunucu i'nin anahtar v*'ye* bağlı olup olmadığını gösteren ikili tam sayı
$w_{i,t}$	Sunucu i'nin t türündeki sanal güvenlik fonksiyonunu çalıştırıp çalıştıramayacağını gösteren ikili tam sayı
Ψ^f	Akış f*'nin* geçmesi gereken sanal ağ güvenliği fonksiyonları kümesi
$f =< S^f, D^f, \Psi^f, B^f>$	Kaynak/varış düğümleri, geçmesi gereken sanal güvenlik fonksiyonları kümesi ve bant genişliği gereksinimi ile tanımlanmış bir akış

$$r_{it}^{f} = \begin{cases} 1, & \text{eğer akış } f, \text{t türündeki VSF'i sunucu } i' \text{den alıyorsa} \\ 0, & \text{değilse} \end{cases} \tag{4.5}$$

$$z_f^{u,v} = \begin{cases} 1, & \text{eğer akış } f\ u, v \text{ linkinden geçiyorsa} \\ 0, & \text{değilse} \end{cases} \tag{4.6}$$

Daha önce de belirtildiği gibi, amaç ağa gelen tüm trafik akışlarının güvenlik gereksinimlerini karşılarken sunucuların toplam enerji tüketimini en aza indirgeyecek

şekilde sanal güvenlik fonksiyonları için optimum konumların bulunmasıdır. Bu doğrultuda, geliştirilen amaç fonksiyonu sunucuların toplam enerji tüketimini literatürdeki gibi formüle etmektedir [109–111].

Amaç fonksiyonu

En aza indirge

$$E = \sum_{i \in S}(y_i . e_{i,idle} + \sum_{t \in A} x_{it} . e_i(c_i^{CPU}, K_t^{CPU})) \quad (4.7)$$

$$e_i(c_i^{CPU}, K_t^{CPU}) = (e_{i,max} - e_{i,idle}) . K_t^{CPU} / c_i^{CPU} \quad (4.8)$$

Bu formülasyona göre bir sunucunun enerji tüketimi, çalışmaya başlamasıyla tükettiği boş durumdaki enerji tüketimi miktarı ile o sunucuya fonksiyon yerleştirilmesiyle ortaya çıkan enerji tüketiminin toplamına eşittir. Bir sunucuda bir fonksiyonun çalışmasına bağlı olarak ortaya çıkan enerji tüketimi ise, o fonksiyonun sunucunun CPU kapasitesinin ne kadarını kullandığı ile doğru orantılıdır.

Kısıtlar

Sunucu kapasitesi kısıtı

$$\sum_{t \in A} K_t^{CPU} . x_{it} \leq c_i^{CPU} , \qquad \forall i \in S \quad (4.9)$$

Eş. 4.9'da ifade edilen kısıt, bir fiziksel sunucunun CPU kapasitesinin, o sunucuya yerleştirilen VSF'lerin toplam CPU gereksinimleri tarafından aşılmadığını garanti etmektedir.

Link kapasitesi kısıtı

$$\sum_{f \in F} B_f . z_f^{u,v} \leq c_{u,v} , \forall u \in V, \forall v \in V \quad (4.10)$$

Eş. 4.10, bir linkin kapasitesinin o link üzerinden geçen trafik akışlarının toplam bant genişliği gereksinimleri tarafından aşılmaması gerektiğini ifade etmektedir.

Akış korunumu kısıtı

$$\sum_{u \in V} z_f^{u,v} + (\text{if } v = S^f \text{ then } 1) = \sum_{w \in V} z_f^{v,w} + (\text{if } v = D^f \text{ then } 1), \quad \forall v \in V, \ \forall f \in F \quad (4.11)$$

Eş. 4.11'de verilen kısıt akış korunumu (flow conservation) kısıtı olarak bilinmekte ve herhangi bir düğüm için, o düğüme ait giriş derecesi kenarlarının (in-degree edges) sayısının çıkış derecesi kenarlarının (out-degree edges) sayısına eşit olması gerekliliğini ifade etmektedir. Bu tez kapsamında ele alınan problem özelinde, verilen bir akışın kaynak ve varış düğümleri için, kaynak düğümün ekstra bir çıkış kenarına, varış düğümün ise ekstra bir giriş kenarına sahip olması gerektiği anlamına gelmektedir. Daha basit bir ifadeyle bu kısıt, bir düğüme giren tüm trafiğin çıkması gerekliliğini garanti etmektedir.

Bütçe kısıtı

$$\sum_{i \in S} \sum_{t \in A} x_{it} \, . uc_t \leq budget \quad (4.12)$$

Eş. 4.12, satın alınan sanal güvenlik fonksiyonlarının toplam maliyetinin ağ yöneticisi tarafından belirlenen bütçeyi aşamayacağını ifade etmektedir. Fonksiyonların toplam maliyeti ise o fonksiyon türünün birim maliyetinin o türde alınan fonksiyon sayısı ile çarpılmasıyla bulunmaktadır.

Fonksiyon türü kısıtı

$$x_{it} \leq w_{i,t} \qquad \forall i \in S, \ \forall t \in A \quad (4.13)$$

Eş. 4.13'te, bir sunucu belirli bir VSF türünü çalıştırabiliyorsa, o sunucuda o VSF türünden en fazla bir tane bulunabileceği ifade edilmektedir.

Döngü engelleme kısıtı

$$\sum_{v \in V} z_f^{u,v} \leq 1, \qquad \forall u \in V, \ \forall f \in F \quad (4.14)$$

$$\sum_{v \in V} z_f^{v,u} \leq 1, \qquad \forall u \in V, \ \forall f \in F \quad (4.15)$$

Eş. 4.14 ve 4.15'te verilen kısıtlar, akışlar tarafından takip edilen yollarda döngü oluşmasını engellemektedir.

Fonksiyon kapasitesi kısıtı

$$\sum_{f \in F} r_{it}^{f} \, . \, B^{f} \leq c_t \, . \, x_{it} \, , \quad \forall i \in S, \, \forall t \in A \tag{4.16}$$

Eş. 4.16, bir fonksiyondan hizmet alan akışların o fonksiyonun işleme kapasitesini aşmaması gerektiğini ifade etmektedir.

Fonksiyon kullanılabilirliği kısıtı

$$\sum_{u \in V} \sum_{v \in V} h_{u,i} \, . \, z_f^{u,v} - r_{it}^{f} \geq 0, \qquad \forall i \in S, \, \forall t \in A, \forall f \in F \tag{4.17}$$

Eş. 4.17'de verilen kısıt, bir fonksiyon bir sunucuda mevcutsa o sunucunun bağlı olduğu düğüm üzerinden geçen akışların o fonksiyon tarafından hizmet alabilmesini sağlamaktadır.

Güvenlik kısıtı

$$\sum_{i \in S} r_{it}^{f} = 1, \qquad \forall f \in F, \, \forall t \in \Psi^{f} \tag{4.18}$$

Eş. 4.18'de ifade edilen kısıt, trafik akışlarının güvenlik gereksinimlerin karşılanacağını garanti etmektedir. Spesifik olarak, her trafik akışının geçmesi gereken sanal güvenlik fonksiyonu kümesindeki fonksiyonların her birinden geçmesini sağlamaktadır.

Sunucu kullanılabilirliği kısıtı

$$\sum_{t \in A} x_{it} - y_i \geq 0 \, , \qquad \forall i \in S \tag{4.19}$$

$$x_{it} \leq y_i, \qquad \forall i \in S, \forall t \in A \tag{4.20}$$

Son olarak Eş. 4.19 ve 4.20'de verilen kısıtlar, i sunucusunun ancak ve ancak herhangi türdeki bir güvenlik fonksiyonunu bulundurduğu sürece kullanıldığı anlamına gelmektedir.

VSF yerleştirme probleminin NP-zor sınıfında yer aldığı bilinmektedir [1, 4, 27, 109, 112, 113]. ILP modeli ile optimum çözümün bulunması büyük ölçekli ağlar için uygulanabilir değildir, çünkü problemin boyutu büyüdükçe ILP modeli makul sürelerde çözüm üretememektedir. Bu nedenle, bu tez kapsamında, gerçek ağ senaryoları için ILP modelinin ölçeklenebilirlik problemini çözmek amacıyla yeni bir sezgisel algoritma geliştirilmiştir. Devam eden alt başlıklarda geliştirilen algoritma sunulmaktadır.

4.2. Geliştirilen Sezgisel Algoritma

Geliştirilen sezgisel algoritma; ILP modelinde olduğu gibi, verilen bir ağ topolojisi, trafik matrisi ve sanal güvenlik fonksiyonu türlerinin özellikleri için akış düzeyindeki güvenlik gereksinimlerini göz önünde bulundurarak enerji tüketimini en aza indirecek şekilde farklı türdeki VSF'lerin sayılarını ve hangi sunucuların üzerine konumlandırılacağını bulmaktadır. ILP modeli ile benzer şekilde, geliştirilen sezgisel algoritma (i) VSF-sunucu eşleşmelerini ve (ii) her bir trafik akışı tarafından izlenecek rotayı çıktı olarak vermektedir.

Geliştirilen algoritmanın fazlarını detaylandırmandan önce, bu fazlar arasındaki ilişkiyi ve algoritmanın genel çalışma mantığını açıklamak amacıyla, önerilen yaklaşım Şekil 4.1 ve 4.2'deki akış diyagramlarında genel hatlarıyla sunulmuştur. Şekil 4.1, verilen bir trafik matrisi için geliştirilen algoritma tarafından hesaplanan ilk yerleştirme aşamasını gösterirken, Şekil 4.2 trafik matrisinde bulunmayan yeni bir trafik akışının geliştirilen algoritma tarafından nasıl ele alındığını ve ihtiyaç halinde yeni VSF'lerin nasıl yerleştirildiğini göstermektedir.

Bu tez kapsamında geliştirilen sezgisel algoritma iki fazdan oluşmaktadır. Bu bölümün geri kalanında, geliştirilen sezgisel algoritmanın fazları açıklanmış, her bir faza ait sözde kod sunulmuş, karmaşıklık analizi yapılmış ve geliştirilen yöntemlerin performans analizi sonuçları tartışılmıştır.

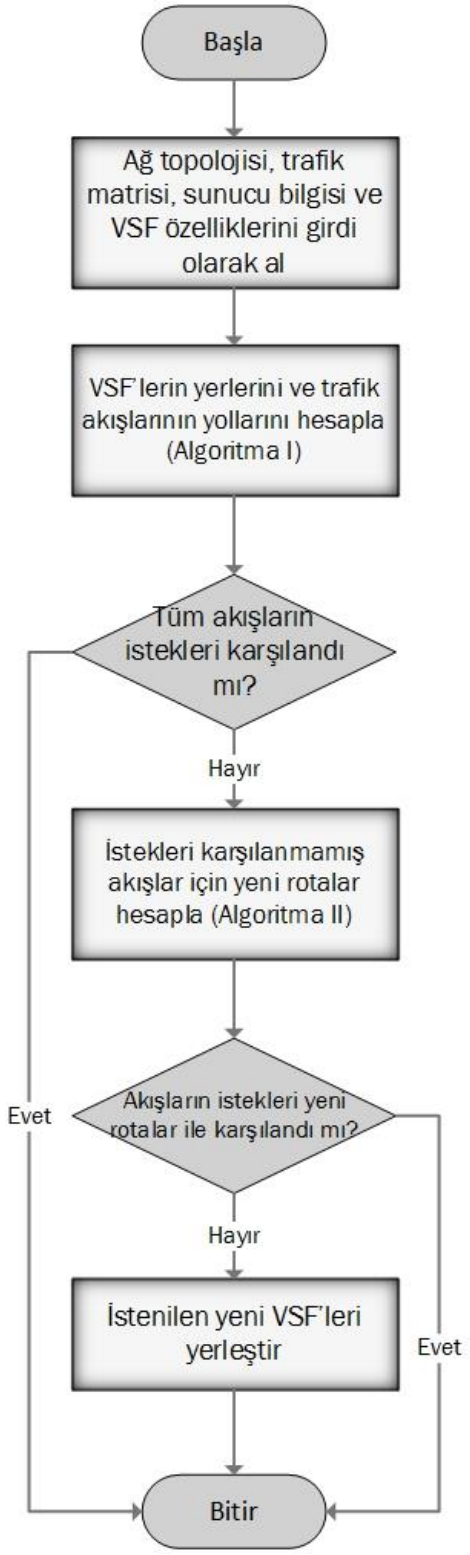
4.2.1. Faz I: arasındalık merkeziliği tabanlı VSF yerleştirme

Bu tez kapsamında yapılan çalışmanın temel amacı enerji tüketimini en aza indirgemek olduğu için VSF'lerin yerleştirildiği sunucuların sayısının minimize edilmesi hedeflenmiştir. Bu nedenle, VSF'lerin yerleştirileceği anahtar sunucuları tespit etmek için arasındalık

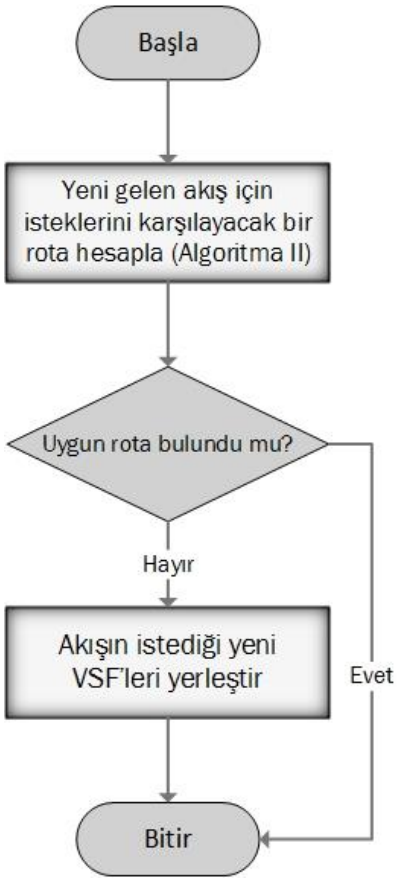
merkezliği tabanlı bir yaklaşım geliştirilmiştir. Arasındalık merkeziliği, bir çizgedeki en önemli düğümlerin tespit edilebilmesi için en kısa yol tabanlı bir merkezilik ölçüsüdür. Bir düğüm üzerinden geçen en kısa yolların sayısı olarak tanımlanmaktadır [1]. Ağ teorisinde, daha yüksek arasındalık merkeziliği değerine sahip olan düğüm, üzerinden daha çok bilgi aktarıldığı için o düğüm ağda daha büyük etkiye sahiptir.

Faz I'in ilk aşamasında, literatürde sıklıkla kullanılan Dijkstra algoritması ile her bir trafik akışı için en kısa yol hesaplaması yapılmaktadır. Daha sonra her bir düğüm için arasındalık merkeziliği değeri hesaplanarak düğümler bu metriğe göre azalan şekilde sıralanır. En yüksek arasındalık merkeziliği değerine sahip olan düğümün üzerinden geçen akışlar tarafından en çok istenen fonksiyon belirlenir. O düğüme bağlı olan sunucunun kapasitesi belirlenen fonksiyon için yeterliyse, fonksiyon o sunucuya yerleştirilmektedir. Fonksiyon yerleştirildikten sonra akışlar bu fonksiyona atanmaya başlanmaktadır. Algoritma bu noktada, bu atama işlemiyle hangi akışların isteklerinin karşılanıp karşılanmadığını kontrol etmekte ve tüm istekleri karşılanan akışları sonraki hesaplamalarda dikkate alınmaması için işlem sürecinden çıkarmaktadır. Bir akışın isteklerinin karşılanması ise hizmet almak istediği tüm ağ güvenliği fonksiyonlarından geçmesi şeklinde tanımlanmaktadır.

Sonrasında, bu düğüm üzerinden geçen akışlar tarafından en çok istenen ikinci fonksiyon belirlenir ve sunucunun kapasitesi dolana kadar yerleştirme işlemi bu şekilde yapılır. Sunucunun kapasitesi dolduğunda ise arasındalık merkeziliği değeri en yüksek olan ikinci düğüme geçilir ve bu işlem her VSF türünden ağ üzerinde en az bir tane konumlandırılıncaya kadar devam ettirilir. Yani, ağ üzerinde herhangi bir noktaya her VSF türünden en az bir tane yerleştirildiğinde Faz I sonlandırılır. Faz I'e ait sözde kod Şekil 4.3'te verilen Algoritma I'de sunulmuştur.



Şekil 4.1. Geliştirilen VSF yerleştirme çözümüne ait ilk yerleştirme aşaması



Şekil 4.2. Geliştirilen VSF yerleştirme çözümüne ait yeni gelen trafik akışının değerlendirilmesi aşaması

Bu adımların sonunda, ağa gelen tüm trafik akışlarının istekleri giderildiyse, yani, akışlar servis almak istedikleri VSF'lerden geçerek kaynak düğümlerinden varış düğümlerine ulaştılarsa, sezgisel algoritma sonlandırılmaktadır. Fakat Faz I'in sonunda istekleri karşılanmayan bir veya daha fazla trafik akışının kalması durumunda Faz II'ye geçilmektedir.

4.2.2. Faz II: akışların yeniden yönlendirilmesi ve yeni VSF'lerin yerleştirilmesi

Geliştirilen algoritmanın ikinci fazına geçilmesi, ilk fazda trafik akışlarının takip ettiği en kısa yollar üzerine her VSF türünden en az bir tane yerleştirilmesiyle tüm trafik akışlarının isteklerinin karşılanamadığı anlamına gelmektedir. Bir başka ifadeyle, Faz II'ye geçilmesi,

Algoritma I. Arasındalık merkeziliği tabanlı VSF yerleştirme (Faz I)

```
türSayısı ← VSF türlerinin sayısı
karşılanmamışAkışListesi← istediği tüm fonksiyon türlerinden hizmet alamamış trafik akışları
foreach akış in karşılanmamışAkışListesi
  Dijkstra algoritması ile akışın en kısa yolunu hesapla.
end for
her bir düğüm için arasındalık merkeziliği (BC) değerini hesapla.
sıralanmışDüğümListesi ← BC değerine göre azalan bir şekilde sıralanmış düğümler
foreach düğüm in sıralanmışDüğümListesi
   for i  in range (türSayısı)
      akışListesi ← düğüm üzerinden geçen gereksinimleri karşılanmamış akışlar
      f ← akışListesi'ndeki akışlar tarafından en çok talep edilen fonksiyon
      CPUGer ← f'e ait CPU gereksinimi
      if düğüm'e sunucu bağlı ise
         s ← düğüm'e bağlı olan sunucu
         sunucuKalanKap ← s sunucusunun kalan CPU kapasitesi
         if sunucuKalanKap >= CPUGer ve f türünde bir fonksiyon s sunucusuna
         yerleştirilmemiş ise
            s sunucusuna f fonksiyonunu yerleştir
            sunucuKalanKap ← sunucuKalanKap - CPUGer
         end if
         foreach akış in akışListesi
            bantTalep ← akış'ın bant genişliği talebi
            fonksiyonKalanKap ← f fonksiyonunun kalan trafik işleme kapasitesi
            if fonksiyonKalanKap >= bantTalep
               akış'ı f 'den hizmet alanlar listesine ekle
               akış'ın gereksinim listesini güncelle
               fonksiyonKalanKap ← fonksiyonKalanKap - bantTalep
            end if
            if akış'ın tüm gereksinimleri karşılandı ise
               akış'ı karşılanmamışAkışListesi'nden sil.
            end if
         end for
      end if
      if her fonksiyon türünden bir tane yerleştirildi ise
         tüm döngülerden çık ve Faz I'i sonlandır.
      end if
   end for
end for
```

Şekil 4.3. Geliştirilen yerleştirme algoritmasının Faz I'ine ait sözde kod

bazı akışların hizmet almak istediği VSF'lerin bu akışların en kısa yolları üzerindeki sunuculara yerleştirilmediğinden güvenlik isteklerinin karşılanamadığını göstermektedir. Bu nedenle, bu aşamada, Faz I'de güvenlik istekleri karşılanamayan her bir akış için alternatif rotalar hesaplamak amacıyla o akışların geçebileceği tüm basit yollar (all simple paths) hesaplanmaktadır. Fakat alternatif yol olarak değerlendirilecek basit yolların sayısı, ağın boyutuyla doğru orantılı olarak arttığı için ağdaki düğüm sayısı olan $|V|$ ile sınırlandırılmıştır. Böylelikle ağ boyutu büyüdükçe karmaşıklığın çok fazla artmaması amaçlanmıştır.

Bu fazda, yol uzunluklarını olabildiğince kısa tutabilmek amacıyla, bir akış için hesaplanan tüm basit yollar arasından en kısa alternatif rota seçilir. Daha sonra, bu akış seçilen yol üzerinden yönlendirildiğinde isteklerinin karşılanıp karşılanmadığı kontrol edilir. Eğer karşılanmıyorsa bir sonraki en kısa yol alternatif yol olarak seçilir. Bu şekilde bir akış için seçilen alternatif yolların hiç biri akışın güvenlik hizmeti isteklerini karşılamıyorsa, akışın orijinal en kısa yolu üzerindeki sunuculara ihtiyaç duyduğu fonksiyon türleri yerleştirilir. Bu durumda, eğer sunucuların kapasite kısıtından dolayı yerleştirme yapmak mümkün olamıyorsa, diğer alternatif rotalar denenir.

Faz I'de güvenlik hizmeti istekleri karşılanamayan her akış için, istekleri karşılanana kadar Faz II tekrar edilir. Ağa gelen bütün trafik akışlarının tüm güvenlik ihtiyaçları karşılandığında ise geliştirilen sezgisel algoritmanın çalışması sonlandırılır. Şekil 4.4'te verilen Algoritma II'de Faz II'ye ait sözde kod sunulmuştur.

4.2.3. Karmaşıklık analizi

Bu bölümde, geliştirilen ve iki fazdan oluşan algoritmanın karmaşıklık analizi sunulmuştur.

Faz I'de ilk olarak; $|F|$ akışların sayısını, $|V|$ de ağdaki düğümlerin sayısını temsil etmek üzere, her bir akış için en kısa yolu hesaplamanın karmaşıklığı $O(|F|.|V|^2)$'dir. Buradaki $O(|V|^2)$ en kısa yolu bulmak amacıyla kullanılan Dijkstra algoritmasının karmaşıklığından gelmektedir. İkinci olarak, her bir düğüm için arasındalık merkeziliği hesaplanmasının karmaşıklığı $O(|F|.|V|)$'dir, çünkü burada her bir düğümden geçen akışların sayısı hesaplanmaktadır.

```
Algoritma II. Akışların Yeniden Yönlendirilmesi ve Yeni VSF'lerin Yerleştirilmesi (Faz II)
karşılanmamışAkışListesi ← Faz I'den sonra istediği tüm fonksiyon türlerinden hizmet alamamış trafik akışları
repeat
  for akış in karşılanmamışAkışListesi
    tümBasitYollar ← akış için hesaplanan tüm basit yolların atlama sayısına göre artan bir şekilde sıralanmış hali
    alternatifYollar ← akış için hesaplanan tüm basit yollar arasından seçilmiş |V| tane en kısa yol
    for p in alternatifYollar
      akış'ın gereksinimlerinin p yolu üzerinde karşılanıp karşılanmadığını kontrol et
      if akış'ın gereksinimleri karşılandı ise
        akış.yol ← p
        break
      end if
    end for
    if akış'ın gereksinimleri karşılanmadı ise
      for p in alternatifYollar
        akış'ın ihtiyaç duyduğu fonksiyon türü örneklerini p yolu üzerindeki sunuculara best-fit yaklaşımına göre yerleştir
        if akış'ın gereksinimleri karşılandı ise
          akış.yol ← p
          break
        end if
        else
          bu aşamada yerleştirilmiş olan fonksiyonları sil
        end if
      end for
    end if
  end for
until ağdaki tüm akışların gereksinimleri karşılanana kadar
```

Şekil 4.4. Geliştirilen yerleştirme algoritmasının Faz II'sine ait sözde kod

Üçüncü olarak, düğümleri arasındalık merkeziliği değerine göre sıralamanın karmaşıklığı $O(|V|.\log(|V|))$'dir. Son olarak, fonksiyonları sunuculara yerleştirmenin ve trafik akışlarını bu fonksiyonlara atamanın karmaşıklığı; sunucu sayısı, genellikle küçük bir sabit değer olan fonksiyon türü sayısı ve trafik akışı sayısının çarpımıyla doğru orantılıdır ve $O(|V|.|F|)$ şeklinde ifade edilebilmektedir. Bu nedenle Faz I'e ait zaman karmaşıklığı en kötü durumda $O(|F|.|V|^2 + |F|.|V| + |V|.\log(|V|) + |V|.|F|)$ olarak hesaplanmakta ve bu da $O(|F|.|V|^2)$ olarak ifade edilebilmektedir.

İlk fazda istekleri karşılanamayan akışlar için çalıştırılan ve algoritmanın ikinci aşaması olan Faz II'de, ilk olarak bu akışlar için tüm basit yollar hesaplanmaktadır. Bu işlemin

karmaşıklığı; F akışları, V düğümleri ve E linkleri temsil etmek üzere $O(|F| . |V+E|)$'dir. İkinci olarak, akışlar seçilen alternatif yollar üzerinden yeniden yönlendirilmektedir. Bu işlemin karmaşıklığı, Algoritma II'de $|V|$ ile sınırlandırılmış olan her bir akış için mevcut alternatif yolların sayısına ve bu yolların uzunluğuna bağlıdır. Bu karmaşıklık $O(|F|.|V|^2)$ olarak ifade edilmektedir. Benzer şekilde, akışların istekleri yeniden yönlendirme ile karşılanmadığında bu akışlar için yeni VSF'lerin yerleştirilmesi işleminin karmaşıklığı da $O(|F|.|V|^2)$'dir.

Sonuç olarak, bu tez kapsamında geliştirilmiş olan iki fazlı algoritmanın zaman karmaşıklığı $O(|F|.|V|^2)$ şeklinde ifade edilmektedir.

4.3. Geliştirilen Yöntemlerin Performans Değerlendirmesi

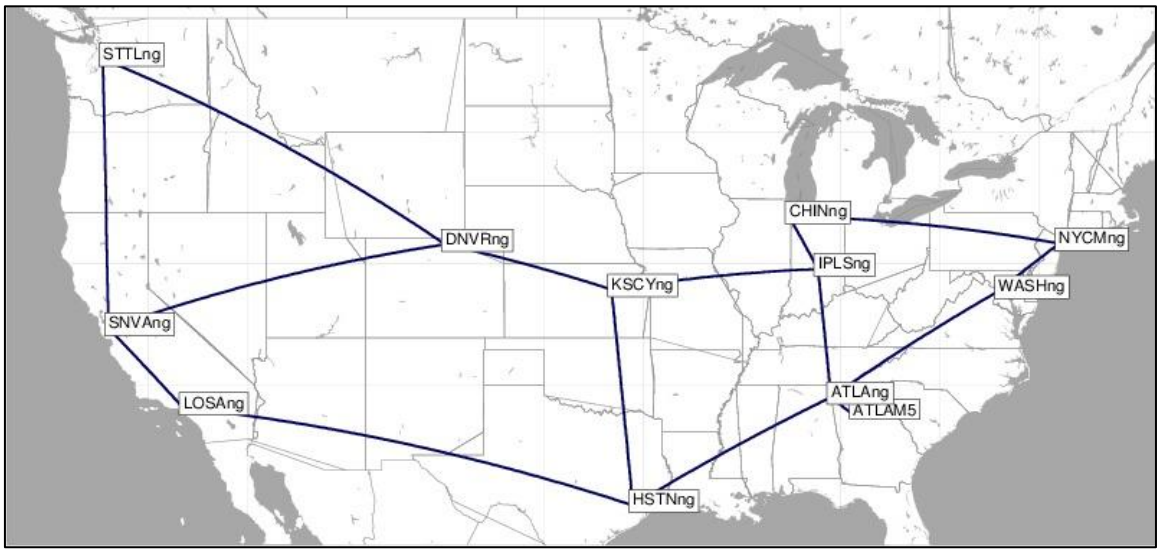
Bu tez kapsamında geliştirilmiş olan yöntemlerin performansını ve etkinliğini değerlendirmek için ILP tabanlı model ve geliştirilen algoritma gerçek ağ topolojisi ve trafik verisi kullanılarak test edilmiştir. Bu doğrultuda, yapılan deneyler aşağıdaki hedefler göz önünde bulundurularak tasarlanmıştır:

- Geliştirilen sezgisel algoritma tarafından ulaşılan enerji tüketimi değerleri, ILP tabanlı modelin ürettiği optimum değerler ile kıyaslanarak geliştirilen algoritmanın başarısının ölçülmesi
- Geliştirilen sezgisel algoritma tarafından ulaşılan enerji tüketimi değerleri, literatürde karşılaştırma ve değerlendirme amacıyla kullanılan referans yerleştirme çözümlerinin (benchmark placement solutions) ürettiği değerler ile kıyaslanarak geliştirilen algoritmanın başarısının ölçülmesi
- Geliştirilen sezgisel algoritma ve ILP modelinin, aktif sunucu sayısı ve akışların yol uzunluğu gibi enerji tüketimi haricindeki diğer metrikler üzerindeki etkisinin ölçülmesi
- Geliştirilen yaklaşımların farklı topolojiler ve trafik hacmi altındaki davranışlarının analiz edilmesi

Bu bölümün geri kalanında, deney ortamı, kullanılan veriler, teknolojiler ve araçlar hakkında bilgi verilmiş, değerlendirme metrikleri açıklanmış ve deneysel sonuçlar sunulmuştur.

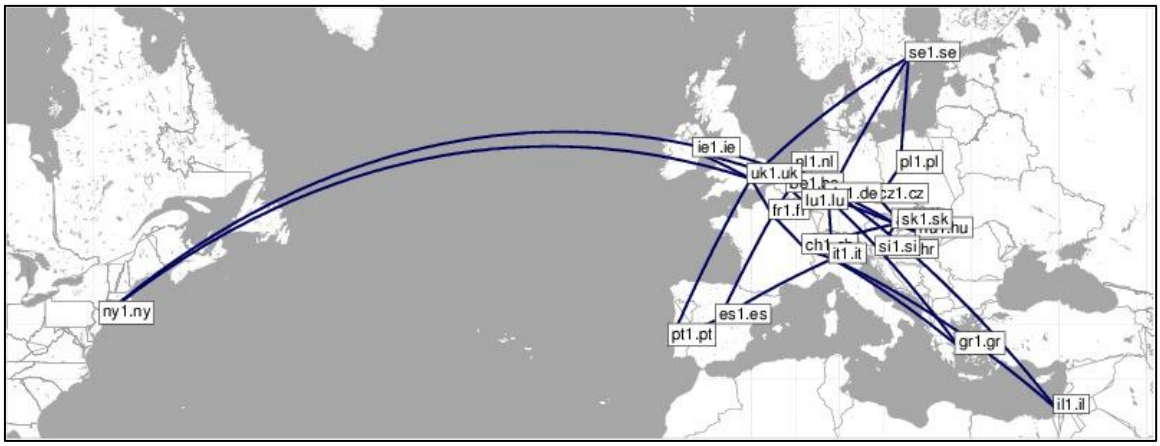
4.3.1. Deney ortamı ve kullanılan veriler

Bu tez kapsamında geliştirilen ILP modeli ve sezgisel algoritmanın performansını gerçekçi senaryolar altında değerlendirebilmek için iki gerçek dünya veri seti kullanılmıştır [114]. Bunlardan biri 12 anahtar, 15 çift yönlü link ve 132 trafik akışından oluşan Internet2 araştırma ağı veri setidir. Internet2 topolojisi Şekil 4.5'te gösterilmiştir.



Şekil 4.5. Internet2 topolojisi [135]

Diğeri ise 22 düğüm, 36 çift yönlü link ve farklı zamanlarda ölçülmüş 462 trafik akışından oluşan ve Şekil 4.6'da gösterilen GEANT araştırma ve eğitim omurga ağıdır.



Şekil 4.6. GEANT topolojisi [135]

Yakın zamanda Berkeley Laboratuvarı tarafından yayınlanan bir veri raporuna (data sheet) göre bir sunucunun maksimum enerji tüketimi 330W olarak alınmıştır [110]. Bu rapora göre, bir sunucunun boş durumdaki enerji tüketiminin maksimum enerji tüketimine oranı dinamik oran (dynamic ratio - DR) olarak adlandırılmaktadır. Bu değer, sunucunun enerji etkinliğine bağlı olarak 0,1 ile 0,5 arasında değişebilmektedir. Bu tez çalışması kapsamında yapılan deneylerde, DR değeri spektrumun her iki ucuna da eşit uzaklıkta olan 0,3 olarak seçilmiştir. Ağdaki tüm sunucuların CPU kapasiteleri ve enerji tüketimi özellikleri birbirleriyle aynıdır.

Yapılan deneylerde, güvenlik duvarı, IDS/IPS, DPI ve zararlı yazılım tarayıcı olmak üzere ağa yerleştirilecek olan dört farklı türde sanal ağ güvenliği fonksiyonu ile çalışılmıştır. Bu fonksiyonlar için CPU gereksinimi veya trafik işleme kapasitesi gibi veriler de literatürdeki diğer çalışmalardan ve fonksiyon üreticilerin yayınladığı veri raporlarından elde edilmiştir [113, 115]. Her bir fonksiyonun trafiği hat hızında işleyebildiği bilinmektedir.

Değerlendirme amacıyla, akış bazında güvenliği temsil edecek şekilde kurumsal ağ güvenliğindeki seviyeler de dikkate alınarak üç farklı güvenlik paketi tasarlanmıştır [116]. Ağa gelen trafiğin, trafik sınıflandırıcısı tarafından bu güvenlik paketi sınıflarına kabaca eşit bir şekilde dağıtıldığı varsayılmaktadır. Bu güvenlik paketleri aşağıda açıklanmıştır:

Temel seviye güvenlik: Bu seviyede güvenlik gereksinimi olan akışlar için yalnızca güvenlik duvarından geçmek yeterli olacaktır [116].

İleri düzey güvenlik: İleri düzey güvenlik gereksinimi olan trafik akışları güvenlik duvarı ile birlikte zararlı yazılım tarayıcı fonksiyonu tarafından da analiz edilmelidirler [116].

Kritik güvenlik: Kritik düzeyde güvenlik gereksinimi olan akışlar, tüm güvenlik fonksiyonlarından bir kere geçmelidirler [116].

Geliştirilen ILP modeli, IBM tarafından geliştirilmiş olan yüksek performanslı optimizasyon paketi CPLEX [117] ortamında gerçeklenmiştir (implemented). CPLEX, doğrusal, karışık tam sayılı veya kuadratik olarak ifade edilen matematiksel modelleri çözmek için simpleks ve bariyer algoritmaları gibi teknikleri kullanmaktadır. Geliştirilen sezgisel algoritma ise Python dilinde gerçeklenmiştir. Deneyler Intel Core i7-4700Q CPU @ 2.40 GHz 15.8 GB RAM özelliklerine sahip dizüstü bir bilgisayar ile yürütülmüştür.

4.3.2. Değerlendirme metrikleri

Geliştirilen modelleri test etmek için kullanılan değerlendirme parametreleri; toplam enerji tüketimi, aktif sunucu sayısı, ortalama akış yol uzunluğu ve ortalama çalışma süresi (average execution time)'dir. Bu metrikler aşağıda kısaca açıklanmıştır.

Toplam enerji tüketimi: Ağdaki toplam enerji tüketimi, Bölüm 4.2'de verilmiş olan Amaç Fonksiyonu (Eş. 4.7. ve Eş. 4.8)'na göre ölçülmektedir.

Aktif sunucu sayısı: VSF'leri yerleştirmek için gerekli olan toplam sunucu sayısıdır.

Ortalama akış yolu uzunluğu: Trafik akışları tarafından takip edilen yollardaki ortalama atlama (hop) sayısı olarak ifade edilmektedir. Bu metriğin, bazı ağlar için ağdaki gecikmeyi etkileyen en önemli faktörlerden biri olduğu literatürde görülmüştür [118].

Ortalama çalışma süresi: Verilen bir ağ topolojisi ve trafik matrisi için sanal ağ güvenliği fonksiyonlarının optimum konumlarının bulunması işleminde geçen süre olarak tanımlanmaktadır.

4.3.3. Deneysel sonuçlar ve tartışma

Bu tez çalışmasında, literatürde ilk defa olarak, trafik akışı bazında güvenlik gereksinimleri göz önünde bulundurularak enerji etkin bir şekilde sanal ağ güvenliği fonksiyonlarının yerleştirilmesi ve trafiğin ihtiyaçları doğrultusunda bu fonksiyonlar üzerinden yönlendirilmesi problemi modellenmiş ve çözülmüştür. Belirtilen bu problemi aynı amaç ve kısıtlar doğrultusunda çözen herhangi bir çalışma olmadığından, bu tez kapsamında geliştirilen yaklaşımlar, literatürde değerlendirme amacıyla sıklıkla kullanılan ve referans olarak kabul edilen yerleştirme çözümleri ile karşılaştırılmıştır [32, 79, 119-121]. Bu yerleştirme çözümleri aşağıda kısaca açıklanmaktadır.

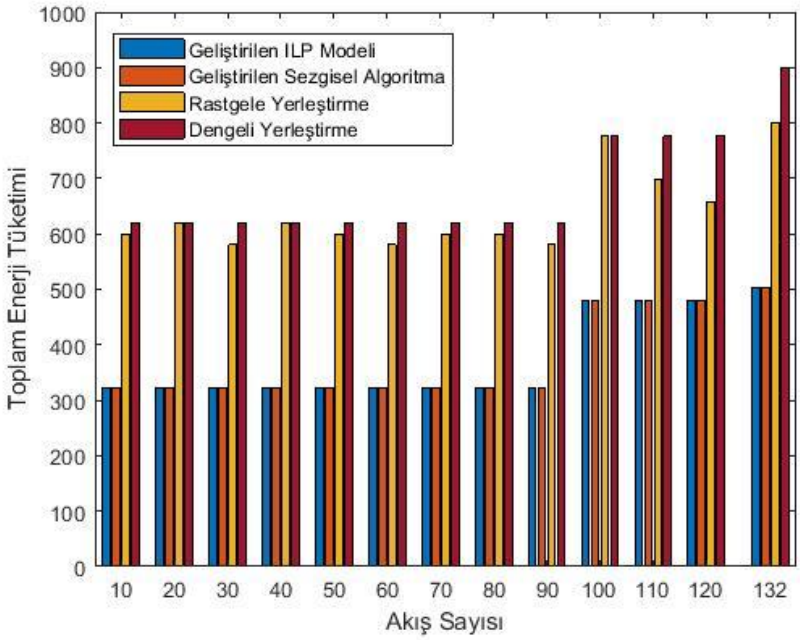
Rastgele yerleştirme (random-fit): Bu strateji ile VSF'ler rastgele bir biçimde seçilen uygun sunuculara yerleştirilmektedir [32, 79, 119-121].

Dengeli yerleştirme: Bu yaklaşım; her sunucuya birer tane VSF yerleştirerek, her birinde en az bir tane VSF olana kadar birden fazla VSF yerleştirmekten kaçınan ve böylelikle VSF'leri sunucular arasında mümkün olduğunca eşit bir şekilde dağıtmayı amaçlayan bir stratejidir. Bu tür bir yük dengeleme yaklaşımı, tek arıza noktası (single point of failure) problemi açısından iyidir [120].

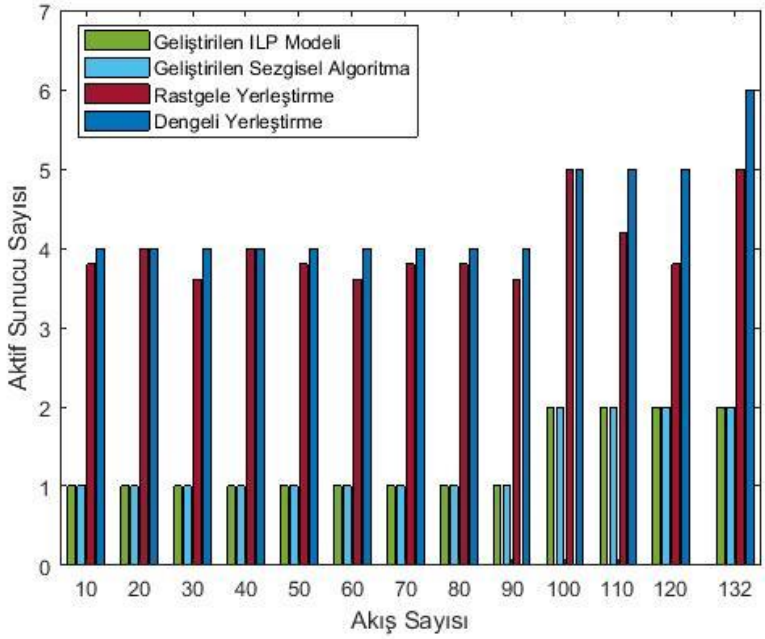
Her bir modelin, Internet2 ve GEANT topolojileri üzerinde değişen miktardaki trafik akışı sayısı için toplam enerji tüketimi ve aktif sunucu sayısı açısından ürettiği sonuçlar Şekil 4.7, 4.8, 4.9, 4.10 ve 4.11'de sunulmuştur. Rastgele yerleştirme modeli, her defasında farklı bir yerleştirme çözümü ürettiği için, beş kere çalıştırılarak bu deneylerin ortalaması alınmıştır. Elde edilen deneysel sonuçlar aşağıda detaylandırılmaktadır.

Internet2 topolojisi için, Şekil 4.7 (a) ve (b)'de gösterildiği gibi, geliştirilen sezgisel algoritma toplam enerji tüketimi ve aktif sunucu sayısı açısından optimum çözümleri bulmaktadır. Çünkü ağdaki merkezi düğümleri bularak VSF'leri yerleştirmekte ve böylelikle en az sayıda sunucuyu aktif hale getirmektedir. Ayrıca geliştirilen algoritma, ürettiği daha az enerji tüketimi ve sunucu sayısı sonuçları ile rastgele yerleştirme ve dengeli yerleştirme yöntemlerinden daha iyi performans sergilemektedir. Daha detaylı bir şekilde açıklamak gerekirse, Internet2 topolojisinde 132 trafik akışına hizmet verebilmek için aktif hale getirilmesi gereken minimum sunucu sayısı, ILP modeli tarafından bulunduğu üzere 2'dir. Geliştirilen sezgisel algoritma da 2 sunucuyu aktif hale getirerek optimum çözümü bulurken, dengeli yerleştirme yaklaşımı, Şekil 4.7 (b)'de gösterildiği üzere aynı durum için 6 adet sunucuyu aktif hale getirmektedir. Bu beklenen bir sonuçtur çünkü geliştirilen ILP modeli ve sezgisel algoritma VSF'leri en az sayıda sunucu üzerinde gruplarken, dengeli yerleştirme yöntemi bu fonksiyonları sunucular arasında dağıtmaktadır. Öte yandan, rastgele yerleştirme yaklaşımı, dengeli yerleştirmeye göre daha iyi bir çözüm üreterek ortalama 5 adet sunucuyu aktif hale getirmektedir. Bu da beklenen bir durumdur, çünkü rastgele yerleştirme stratejisi VSF'leri sunucular arasında kasıtlı bir şekilde dağıtmak yerine rastgele bir biçimde uygun sunuculara yerleştirmektedir. Bunlara ek olarak, geliştirilen sezgisel algoritmanın 120 ve 130 adet trafik akışı için aktif hale getirdiği sunucu sayısı birbiriyle aynıyken, 132 adet trafik akışına hizmet verebilmek için daha çok enerji tüketildiği görülmüştür. Bu durum; 132 adet akış için fazladan bir tane daha VSF'e ihtiyaç duyulmasından, fakat bu VSF'in yeni bir sunucuya ihtiyaç kalmadan mevcut sunuculardan biri üzerinde çalıştırılabilmesinden kaynaklanmaktadır. Son olarak, geliştirilen algoritma,

dengeli yerleştirme çözümüne göre %38 ila %48 enerji tasarrufu sağlarken, rastgele yerleştirme yaklaşımına göre %27 ila %48 enerji tasarrufu sağlamaktadır. Bu değerler, trafik akışı sayısı arttıkça yükselmektedir ki bu da gerçekçi ağ koşulları için istenen bir sonuçtur.



(a)

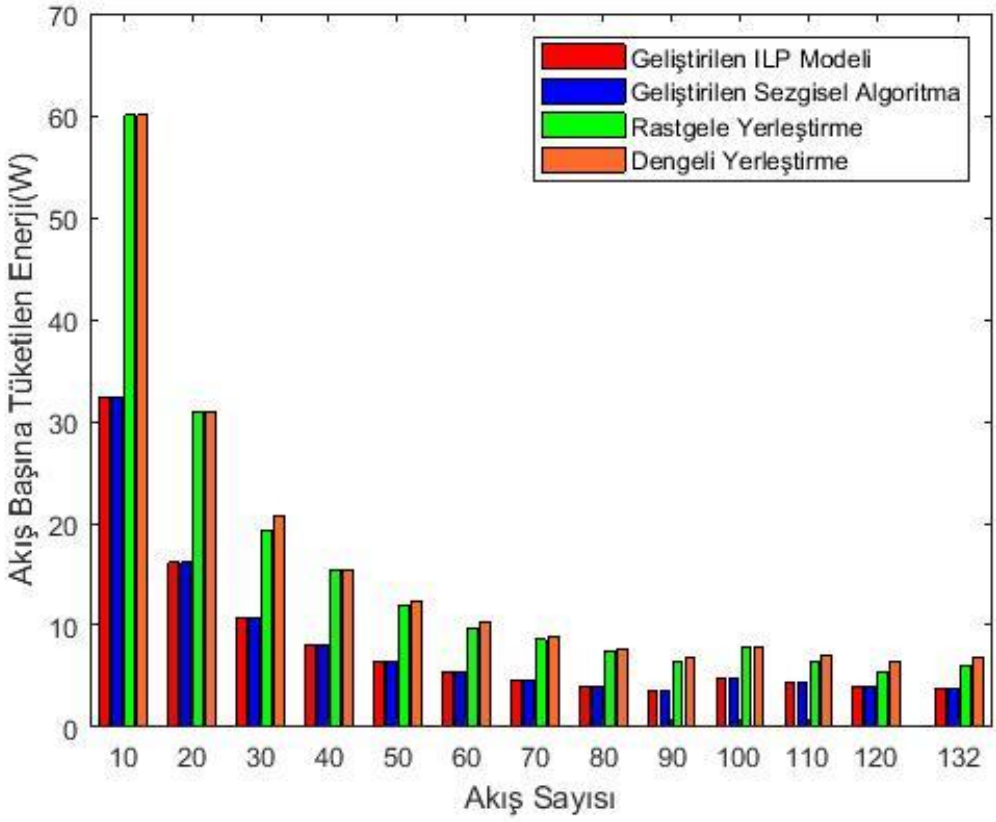


(b)

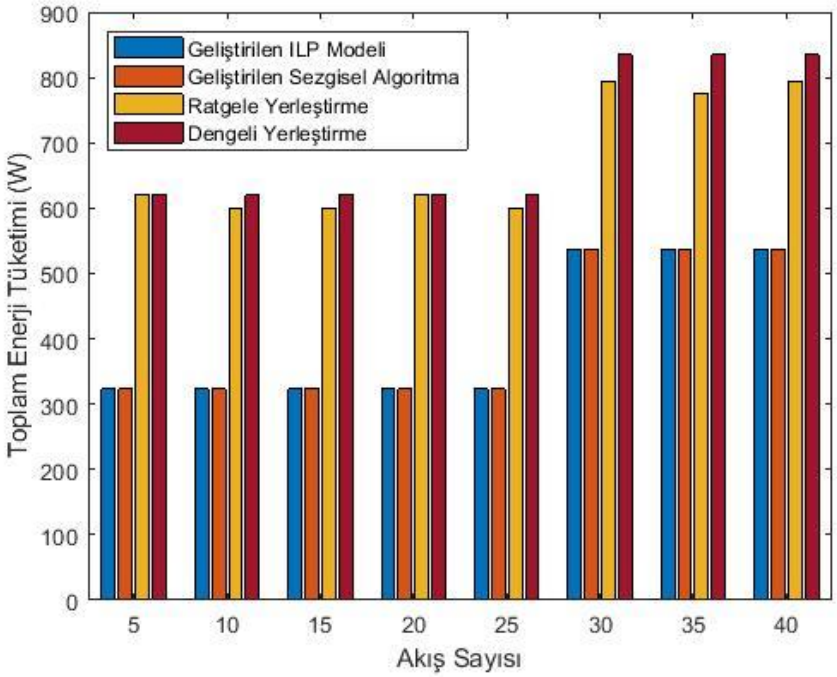
Şekil 4.7. Internet2 topolojisinde değişen sayıda trafik akışı için elde edilen değerler (a) toplam enerji tüketimi, (b) aktif sunucu sayısı

Şekil 4.8'de Internet2 topolojisi için, akış başına tüketilen enerji miktarı gösterilmektedir. Bu grafiğe bakıldığında, trafik miktarı yeni sunucuların aktif hale getirilmesini gerektirinceye kadar VSF'leri birleştirmenin etkinliği artırdığı ve akış başına tüketilen enerji miktarını azalttığı gözlenmektedir. Yeni sunucuların etkinleştirilmesi gerektiği durumda ise bu pozitif eğilim kaybolmaktadır.

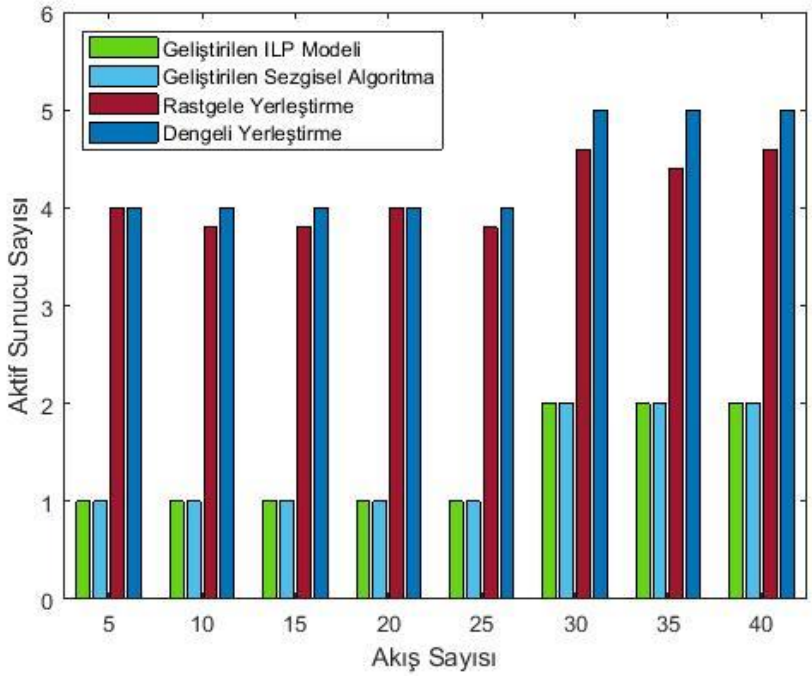
GEANT topolojisi için de benzer sonuçlar elde edilmiştir. Şekil 4.9'da görüldüğü gibi, geliştirilen sezgisel algoritma, bu veri setindeki 40 akış için hem toplam enerji tüketimi hem de aktif hale getirilen sunucu sayısı açısından rasgele yerleştirme ve dengeli yerleştirme çözümlerinden daha iyi sonuçlar üreterek ILP modelinin ürettiği optimum sonuçları bulmuştur. Fakat geliştirilen ILP modeli, GEANT veri setinde 40 akıştan fazlası için 24 saat içinde çözüm üretememiştir. Bu nedenle, Şekil 4.9'da gösterildiği gibi, kullanılan dört farklı yerleştirme modeli için GEANT topolojisi üzerinde yalnızca 40 akışa kadar kıyaslama yapılabilmiştir.



Şekil 4.8. Internet2 topolojisi için akış başına tüketilen enerji miktarı



(a)



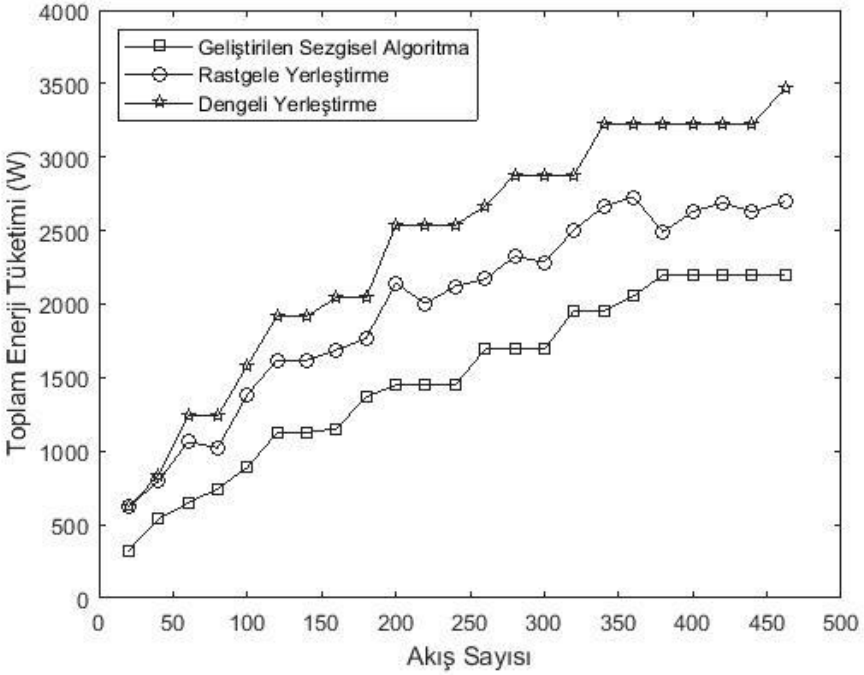
(b)

Şekil 4.9. GEANT topolojisinde 40 tane trafik akışı için elde edilen değerler (a) toplam enerji tüketimi, (b) aktif sunucu sayısı

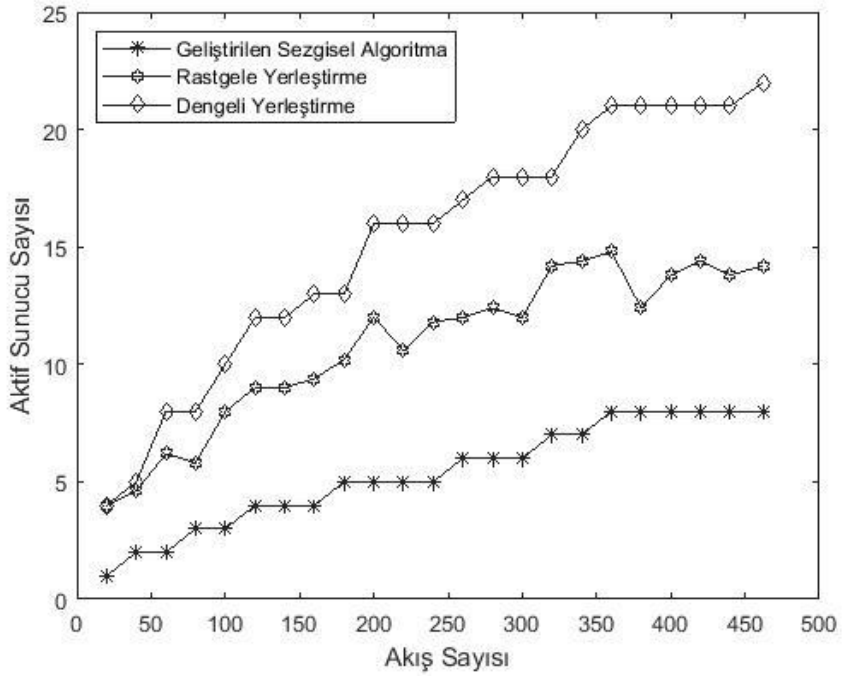
462 trafik akışı içeren GEANT veri setinin geri kalanı için ise geliştirilen sezgisel algoritmanın performansı, Şekil 4.10'da gösterildiği gibi, rastgele yerleştirme ve dengeli yerleştirme çözümleri ile kıyaslanmıştır. Bu veri seti için, geliştirilen algoritma, dengeli yerleştirme çözümüne göre %32 ila %48 arasında enerji tasarrufu sağlarken, rastgele yerleştirme yaklaşımına göre %11 ila %48 arasında enerji tasarrufu sağlamaktadır.Internet2 topolojisi ile kıyaslandığında, bu oranların alt sınırları arasındaki değişimin veri seti ile ilgili olduğu görülmektedir. GEANT veri setindeki akışların en kısa yolları üzerindeki ortak düğümlerin sayısı, Internet2'dekinden daha az olduğu için sezgisel algoritmanın ilk fazında istekleri karşılanmayan akışların sayısı daha fazladır. Dolayısıyla, geliştirilen sezgisel algoritma, bu akışlara hizmet verebilmek için ikinci fazda yeni sunucuları aktif hale getirerek yeni VSF'ler yerleştirmektedir. Bu da veri setine bağlı olan bir varyasyona sebep olmaktadır. Fakat yine de, Şekil 4.10 (b)'de görüldüğü gibi, geliştirilen algoritma tarafından aktif hale getirilen sunucu sayısı, dengeli yerleştirme çözümüne göre oldukça azdır.

Şekil 4.11'de GEANT topolojisi için, akış başına tüketilen enerji miktarı gösterilmektedir. Akış sayısına bağlı olarak daha yavaş bir düşüş görülse de, bu topoloji için de Internet2'de olduğu gibi, akış sayısı arttıkça akış başına tüketilen enerji miktarının azaldığı görülmektedir.

Sonuç olarak, geliştirilen sezgisel algoritmanın ILP modeline göre çok daha küçük bir çözüm uzayını tarayarak optimum yerleştirme çözümlerini bulduğu görülmüştür. Geliştirilen ILP modeli, her durumda optimum çözümü bulmaktadır, fakat çok daha yüksek çalışma zamanına sahiptir. Çizelge 4.2'de, geliştirilen sezgisel algoritma ve ILP modelinin her iki topoloji için de ortalama çalışma zamanları gösterilmektedir. Sezgisel algoritma, ILP modeline göre 228 kat daha hızlı bir şekilde sonuç üretmektedir. Ayrıca, ILP modelinin 22 düğüm içeren GEANT topolojisi için 40'dan fazla trafik akışı olduğu durumlarda makul sürelerde sonuç üretemediği görülmektedir. Fakat geliştirilen sezgisel algoritmanın, GEANT topolojisinde 462 trafik akışı olduğu durumda dahi ortalama 5 saniye gibi kısa bir sürede sonuç üretebildiği gözlenmiştir.

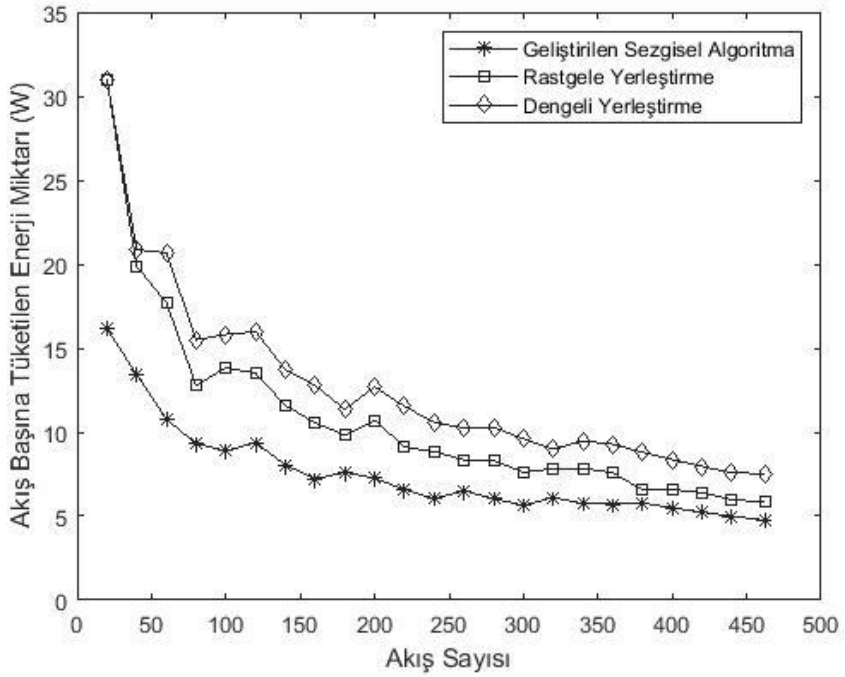


(a)



(b)

Şekil 4.10. GEANT topolojisinde geriye kalan trafik akışları için elde edilen değerler (a) toplam enerji tüketimi, (b) aktif sunucu sayısı

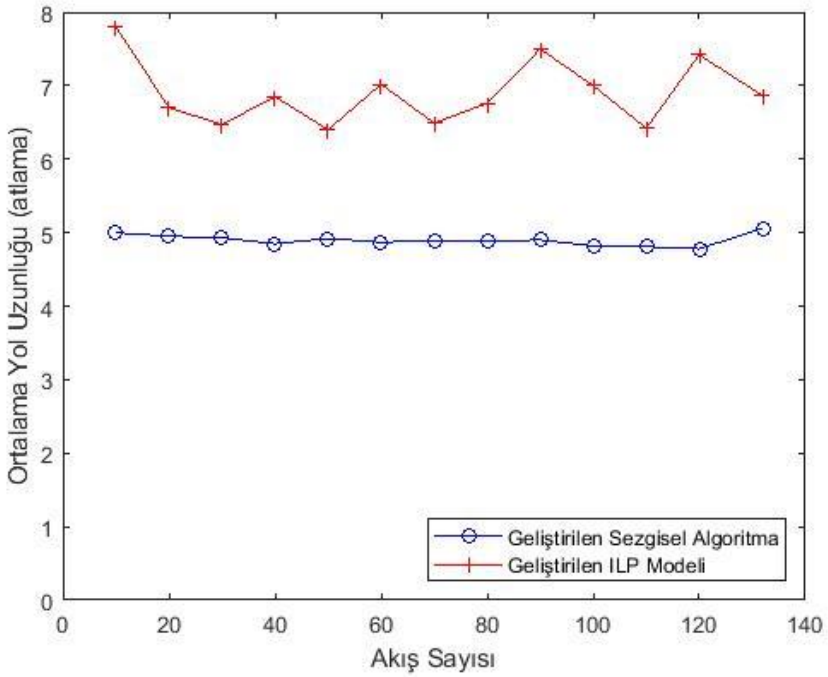


Şekil 4.11. GEANT topolojisi için akış başına tüketilen enerji miktarı

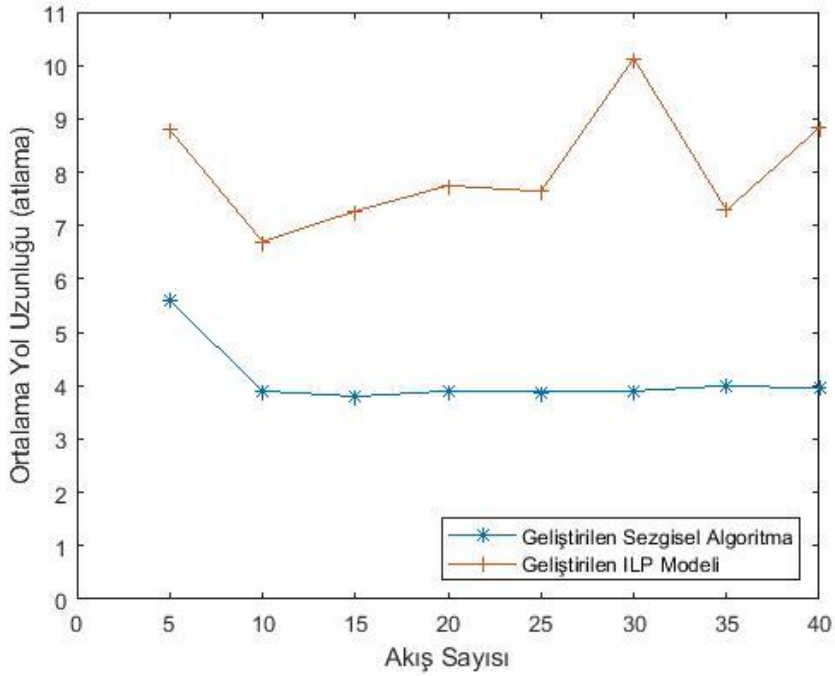
Çizelge 4.2. Ortalama çalışma süreleri

Topoloji	Geliştirilen ILP Modeli	Geliştirilen Sezgisel Algoritma
Internet2 (12 düğüm, 15 link, 132 trafik akışı)	38,77s	0,17s
GEANT (22 düğüm, 36 link, 462 trafik akışı)	∞	5,02s

Geliştirilen sezgisel algoritmanın ILP modeline üstünlük sağladığı bir diğer metrik ise trafik akışların ortalama yol uzunluğudur. Çünkü daha önce de bahsedildiği gibi, sezgisel algoritma en kısa yol tabanlı arasındalık merkeziliği metriğini kullanmaktadır. Şekil 4.12 (a)'da görüldüğü gibi, Internet2 topolojisi için, ILP modelinin ürettiği ortalama yol uzunluğu 6,9 iken sezgisel algoritmanın ürettiği yol uzunluğu 4,9'dur. Bu da sezgisel algoritmanın, ILP modeline göre bu topoloji için yolları ortalama %28,99 oranında azalttığını göstermektedir. Şekil 4.12 (b)'de gösterildiği üzere, bu oran GEANT topolojisi için % 51,3'e yükselmiştir. Bunun sebebi ise şu şekilde açıklanmaktadır: Ağ topolojisi büyüdükçe ILP modeli tarafından bulunan rotalar uzamakta fakat sezgisel algoritma hem Faz I'de hem de Faz II'de olabildiğince çok sayıda trafiği en kısa yollardan yönlendirmeye çalıştığı için sezgisel algoritma için bu fark fazla olmamaktadır. Ağ topolojisinin büyümesine ve trafik akışı sayısının artmasına rağmen yol uzunluklarının kısa tutulmaya çalışılması ise gerçekçi ağ koşulları için istenen bir durumdur.



(a)



(b)

Şekil 4.12. Internet2 ve GEANT topolojileri üzerinde geliştirilen ILP modeli ve sezgisel algoritma tarafından üretilen akış yolu uzunlukları (a) Internet2, (b) GEANT

4.4. Bölüm Değerlendirmesi

Tez çalışmasının bu bölümünde, ağa gelen trafiğin akış bazında güvenlik gereksinimleri göz önünde bulundurularak enerji etkin bir şekilde sanal ağ güvenliği fonksiyonu yerleştirme problemi literatürde ilk defa olarak modellenmiş ve çözülmüştür. Bu kapsamda, (i) optimum yerleştirme çözümleri üreten bir ILP modeli ve (ii) geliştirilen ILP modelinin ölçeklenebilirlik sorununu çözmek için bir sezgisel algoritma olmak üzere iki yeni yöntem geliştirilmiştir.

Deneysel sonuçlar; geliştirilen sezgisel algoritmanın, toplam enerji tüketimi ve aktif sunucu sayısı açısından ILP modeli tarafından üretilen optimum yerleştirme çözümlerini, ILP modelinden çok daha kısa sürede ürettiğini göstermektedir. Dolayısıyla, büyük ölçekli ağlar için optimum çözümlerin üretilmesini beklemenin kabul edilebilir olmadığı durumlarda, çoğu örnekte görüldüğü gibi uygulanabilir zaman aralıklarında optimum çözümleri ürettiği için geliştirilen sezgisel algoritmanın kullanılabileceği görülmektedir.

Geliştirilen sezgisel algoritmanın ILP modeline göre bir diğer avantajı ise; bazı ağlarda ağ gecikmesini etkileyen en önemli faktörlerden biri olan akışların yol uzunluğu önemli ölçüde azaltmasıdır. İki model tarafından üretilen akış yollarının uzunluğu arasındaki bu fark, ağ ölçeği büyüdükçe artmaktadır, çünkü ILP modeli tarafından üretilen yollar topoloji büyüdükçe uzamakta, fakat sezgisel algoritma trafiği en kısa yolları üzerinden veya aralarından en kısası seçilen alternatif yollar üzerinden yönlendirmektedir.

Ayrıca geliştirilen ILP modeli ve sezgisel algoritma, literatürde karşılaştırma ve değerlendirme amacıyla kullanılan referans yerleştirme çözümleri ile de kıyaslanarak performans değerlendirmeleri yapılmıştır. Elde edilen sonuçlar; geliştirilen sezgisel algoritmanın, literatürdeki referans yerleştirme çözümlerine göre enerji tüketimini %48 oranında azalttığını göstermektedir.

Literatürde, ele alınan bu problemi aynı amaç ve kısıtlar doğrultusunda çözen herhangi bir çalışma bulunmamaktadır. Fakat VSF yerleştirme problemi ile benzer olduğu bilinen fonksiyon/sanal makine/orta kutu/ vb. yerleştirme problemlerini enerji etkin bir biçimde çözmeyi hedefleyen çalışmalar [34, 104, 105, 108, 112, 113, 119] incelendiğinde, bu çalışmalarda elde edilen enerji tasarrufu değerlerinin %13,4 ile %46,1 arasında değiştiği

görülmektedir. Dolayısıyla, tez kapsamında geliştirilen algoritmanın ürettiği sonuçlar bu değer aralıklarında yer aldığı için kabul edilebilir olduğu yorumlanmaktadır.

5. SANAL AĞ GÜVENLİĞİ FONKSİYONLARININ ENERJİ ETKİN YERLEŞTİRİLMESİ İÇİN SDN KONTROLCÜSÜ TASARIMI VE GERÇEKLEŞTİRİLMESİ

Daha önceki bölümlerde de bahsedildiği gibi, SDN ve NFV teknolojileri, ortak amaçları paylaştıkları için çoğu durumda birlikte kullanılmaktadır. SDN, NFV ile sanallaştırılmış ağ fonksiyonlarının etkin bir şekilde kullanılabileceği programlanabilir ağ altyapısını sunmaktadır. Sanal ağ fonksiyonları ağda konumlandırılırken SDN'in sağladığı en büyük kolaylık, yönlendirme kurallarının hızlı ve dinamik bir şekilde belirlenip uygulanabilmesidir. Bu nedenle, bu tez çalışması kapsamında ele alınmış olan, VSF'lerin enerji etkin bir şekilde yerleştirilmesi ve trafiğin bu fonksiyonlar üzerinden yönlendirilmesi, merkezi kontrol altındaki programlanabilir ağ altyapısı tarafından sağlanan daha hızlı güncelleme yeteneği sayesinde SDN'de çok daha kolay olmaktadır. Bu nedenle, tez kapsamında, geliştirilen yaklaşımla uyumlu bir şekilde ağ yönetimini gerçekleştirecek bir SDN kontrolcü yazılımı geliştirilmiştir. Geliştirilen SDN kontrolcüsü, 4. Bölümde detaylı bir şekilde açıklanmış olan sezgisel algoritmanın enerji etkin sanal ağ güvenliği fonksiyonu yerleştirme kararları doğrultusunda trafik akışları için belirlenen yollardan bu akışları yönlendirmek amacıyla ağı programlayan bir yazılımdır. Daha açık bir ifadeyle, sezgisel algoritma tarafından belirlenen yönlendirme kuralları, geliştirilen SDN kontrolcü yazılımı tarafından, veri katmanındaki programlanabilir anahtarların akış tablolarına yazılmakta ve böylelikle akışlar dinamik bir şekilde yönlendirilmektedir.

Sonraki alt başlıklarda, geliştirilen yaklaşıma ait sistem modeli SDN ve NFV mimarisi üzerinde detaylı bir şekilde açıklanmış ve ağın işleyişi açısından elde edilen performans değerlendirmesi sonuçları sunulmuştur.

5.1. Önerilen Sistem Mimarisi

Daha önce de açıklandığı gibi, SDN'in temel felsefesi kontrol katmanını veri katmanından ayırmaktır. Bu tez çalışması kapsamında geliştirilen sezgisel algoritma, VSF yerlerinin belirlenmesi ve trafik akışlarının yönlendirilmesi için SDN kontrolcüsü üzerinde yönlendirme kararlarını belirleyen bir modül olarak gerçeklenmiştir. Önerilen sistem mimarisi Şekil 5.1'de özetlenmiş ve her bir adım aşağıda detaylı bir şekilde açıklanmıştır.

Adım 1: girdilerin alınması

İlk olarak, sanal ağ güvenliği fonksiyonları için optimum konumların bulunması ve trafik akışlarının güvenlik gereksinimleri doğrultusunda bu fonksiyonlardan yönlendirilirken takip etmeleri gereken yolların hesaplanabilmesi için ağ topolojisi, trafik matrisi, sunucu listesi ve sanal ağ güvenliği fonksiyonlarının özellikleri Enerji-Etkin VSF Yerleştirme ve Trafik Yönlendirme Modülü tarafından girdi olarak alınmaktadır.

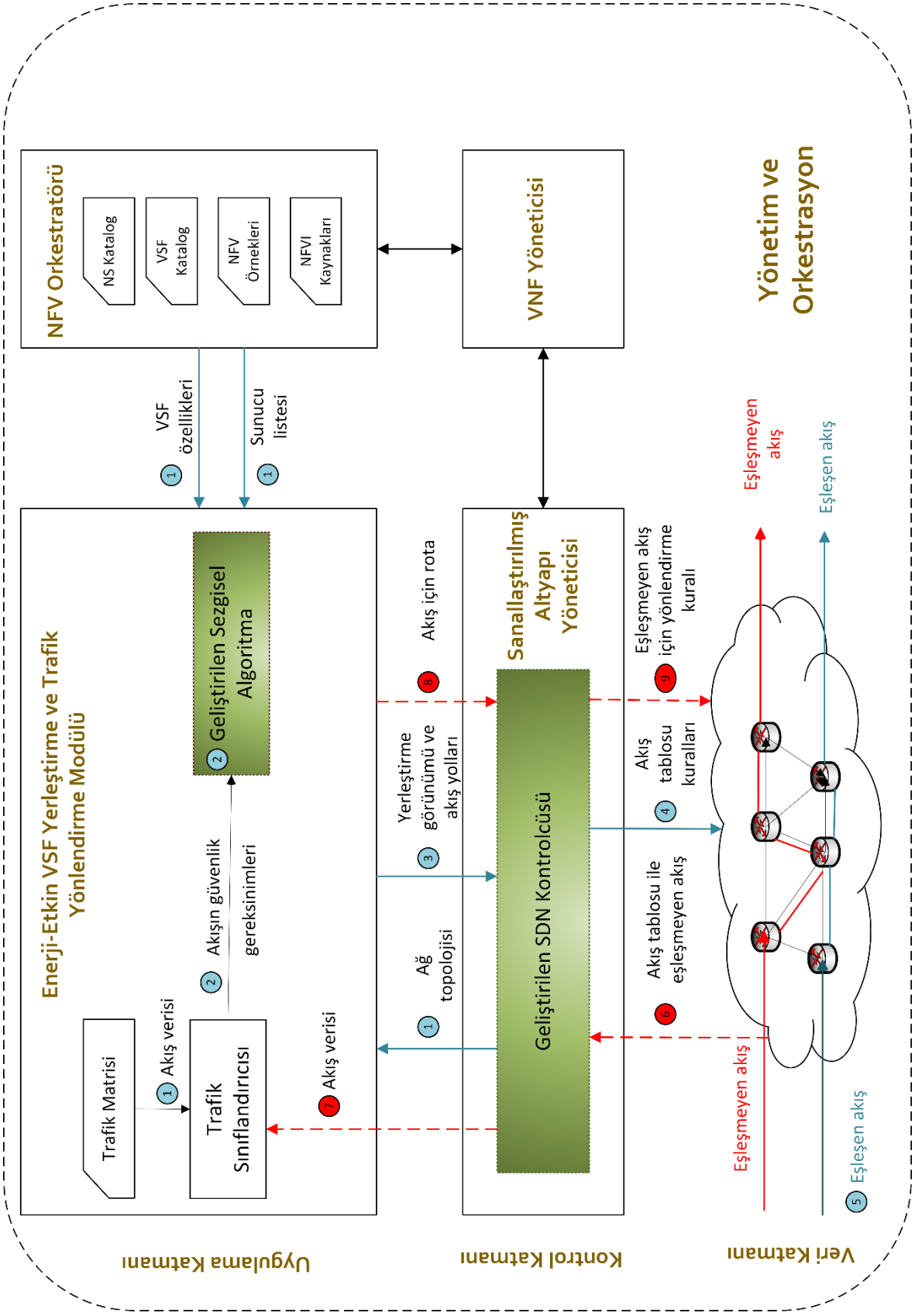
Bu girdilerden; farklı türdeki VSF'lerin (CPU gereksinimi, trafik işleme kapasitesi ve birim maliyeti vb.) özellikleri ve sunucu listesi (maksimum ve boş durumdaki enerji tüketimi, VSF'leri barındırmak için sahip olduğu CPU kapasitesi vb.) NFV orkestratörü modülü tarafından sağlanmaktadır. Ayrıca NFV orkestratörü, bu tür meta verilerin sağlanmasının yanı sıra VSF'lerin çalıştırılması için gerekli olan yazılım imajlarına da sahiptir.

Trafik matrisi, ağ operatörü tarafından istenen zaman aralıklarında (saatlik, günlük, haftalık vb.) sağlanmaktadır ve her bir trafik akışı için kaynak düğüm, varış düğümü, bant genişliği gereksinimi ve geçmesi gereken ağ güvenliği fonksiyonları kümesini içermektedir.

Son olarak ağ topolojisi, SDN mimarisinin merkezi kontrol yapısı sayesinde, tüm ağın görünümüne hâkim olan kontrolcü tarafından alınmaktadır. Topoloji verisi, ağ alt yapısındaki anahtarların birbirleriyle ve sunucularla olan bağlantıları ile anahtarları ve sunucuları birbirine bağlayan linklerin bant genişliği kapasitelerini içermektedir.

Adım 2: VSF konumlarının bulunması ve akış yollarının hesaplanması

Enerji-Etkin VSF Yerleştirme ve Trafik Yönlendirme Modülü, VSF konumlarının belirlenmesi ve trafik akışlarının takip edeceği yolların hesaplanmasından sorumludur. Bu doğrultuda, ilk olarak, trafik sınıflandırıcısı trafik matrisindeki akışların geçmesi gereken VSF'ler listesini değerlendirerek; akışların geçmesi gereken VSF türleri ve sayıları gibi akışlara ait güvenlik gereksinimlerini çıkarır ve organize eder. Daha sonra bu veriler işlenmek üzere, geliştirilen sezgisel algoritmaya girdi olarak verilir.



Şekil 5.1. SDN ve NFV mimarisi üzerinde önerilen sistem mimarisi

Sezgisel algoritma, trafik matrisindeki akışların güvenlik gereksinimlerini göz önünde bulundurarak enerji etkin bir biçimde VSF'lerin yerleştirilmesi ve trafik akışlarının ihtiyaçları doğrultusunda bu fonksiyonlar üzerinden yönlendirilmesinden sorumludur. Algoritmanın nasıl çalıştığı ve bu kararları nasıl verdiği Bölüm 4'te detaylı bir şekilde açıklanmıştır.

Adım 3: kontrolcünün bilgilendirilmesi

Enerji-Etkin VSF Yerleştirme ve Trafik Yönlendirme Modülü, trafik matrisindeki akışlar için hesaplanan rotaları kontrolcü yazılımına bildirir. Bu rotalar Kuzey ara yüzü aracılığıyla Python tabanlı POX kontrolcüsüne iletilmektedir. Bu yollar daha sonra kontrolcü tarafından OpenFlow kurallarına dönüştürülecek ve programlanabilir anahtarların akış tablolarına yazılacaktır.

Adım 4: kuralların akış tablolarına yazılması

Kontrolcü trafik akışları için hesaplanan yolları aldıktan sonra, Enerji-Etkin VSF Yerleştirme ve Trafik Yönlendirme Modülü tarafından alınan kararlar doğrultusunda akışların nasıl yönlendirileceğini anahtarlara bildirir ve veri katmanındaki anahtarların akış tablolarını günceller. Geliştirilen kontrolü yazılımı çalışmaya başladığında her bir anahtarla bağlantı kurulur ve belirlenen kurallar anahtarların akış tablolarına SDN mimarisindeki Güney ara yüzü için bir standart haline gelmiş olan OpenFlow protokolü üzerinden yazılır. Kontrolcü yazılımının bu yazma işlemini nasıl gerçekleştirdiği Şekil 5.2'de sözde kod şeklinde sunulmuştur.

Kontrolcü yazılımının çalışma prensibini daha iyi ifade edebilmek için Çizelge 5.1'de yönlendirme politikalarına küçük bir örnek sunulmuş ve kontrolcü tarafından 1 numaralı anahtara yazılması gereken kurallardan bir kesit gösterilmiştir. Bu politikalar doğrultusunda kontrolcü yazılımının yönlendirme işlemini nasıl gerçekleştirdiği aşağıda detaylı bir biçimde açıklanmıştır.

Algoritma III. Geliştirilen SDN kontrolcü yazılımı

```
Ağı keşfet ve topolojideki tüm anahtarlar ile bağlantıları kur.
foreach anahtar in topoloji
  dpid (data path id) değerlerini öğren.
end for
yönlendirmePolitikaları ← Enerji-Etkin VSF Yerleştirme ve Trafik Yönlendirme modülü
tarafından belirlenen akış yolları
msg ← new akışKuralı
foreach politika in yönlendirmePolitikaları
   msg.match.dl_src ← politika.src
   msg.match.dl_dst ← politika.dst
   anahtar ← politika.sw
   sonraki_düğüm ← politika.next_hop
   if event.connection.dpid == anahtar
      if anahtar == politika.dst
         msg.actions.append(outPort = 1)
         event.connection.send(msg)
         Sayaç değerlerini güncelle.
      end if
      else
          msg.actions.append(outPort = sonraki_düğüm)
          event.connection.send(msg)
          Sayaç değerlerini güncelle.
      end if
end for
```

Şekil 5.2. Geliştirilen kontrolcü yazılımının çalışma prensibine ait sözde kod

Çizelge 5.1. Örnek yönlendirme politikaları

Politika ID	Anahtar ID	Kaynak düğüm	Varış düğümü	Sonraki düğüm
100	1	*00:00:00:00:00:03*	*00:00:00:00:00:01*	*00:00:00:00:00:01*
101	1	*00:00:00:00:00:03*	*00:00:00:00:00:02*	*00:00:00:00:00:07*
102	1	*00:00:00:00:00:03*	*00:00:00:00:00:04*	*00:00:00:00:00:08*

Örnek olarak, Çizelge 5.1'de verilen yönlendirme politikalarına göre, 1 numaralı anahtar, *00:00:00:00:00:03* kaynağından gelip kendisine bağlı olan *00:00:00:00:00:01* varış noktasına gitmesi gereken bir paket aldığında, bu paketi *00:00:00:00:00:01* MAC adresli sunucuya (1 numaralı porttan) iletmektedir. Varış düğümü kendisi değilse ve *00:00:00:00:00:03* kaynağından gelip *00:00:00:00:00:02* MAC adresli sunucuya iletilmesi gereken bir paket aldıysa, bu paketi (2 numaralı porttan) *00:00:00:00:00:07* MAC adresli sunucuya; *00:00:00:00:00:03* kaynağından gelip *00:00:00:00:00:04* MAC adresli sunucuya

iletilmesi gereken bir paket aldıysa, bu paketi (3 numaralı porttan) *00:00:00:00:00:08* MAC adresli sunucuya iletecektir.

Adım 5: eşleşen akışın yönlendirilmesi

Ağa trafik matrisinde yer alan bir akış geldiğinde, anahtarlar Enerji-Etkin VSF Yerleştirme ve Trafik Yönlendirme Modülü belirlenen ve kontrolcü yazılımı tarafından kendilerine iletilen kurallar doğrultusunda akışı yönlendirerek kaynak düğümünden varış düğümüne ulaşmasını sağlar. Şekil 5.3'te OpenFlow anahtarlarının bu işlemi nasıl gerçekleştirdikleri gösterilmiştir. Öncelikle, gelen trafik akışı için akış tablosundaki kurallar arasında eşleşme aranır. Gelen paketin başlıkları (switch port, MAC src, MAC dst, vb.) tablodaki girdilerden biriyle eşleştiğinde işlem (action) aşamasına geçilir ve 1 numaralı işlem gerçekleştirilerek paket kontrolcü tarafından belirtilen port üzerinden bir sonraki anahtara iletilir. Tablonun istatistik bölümünde ise eşleşen paketlerin sayısı gibi veriler tutulmaktadır.

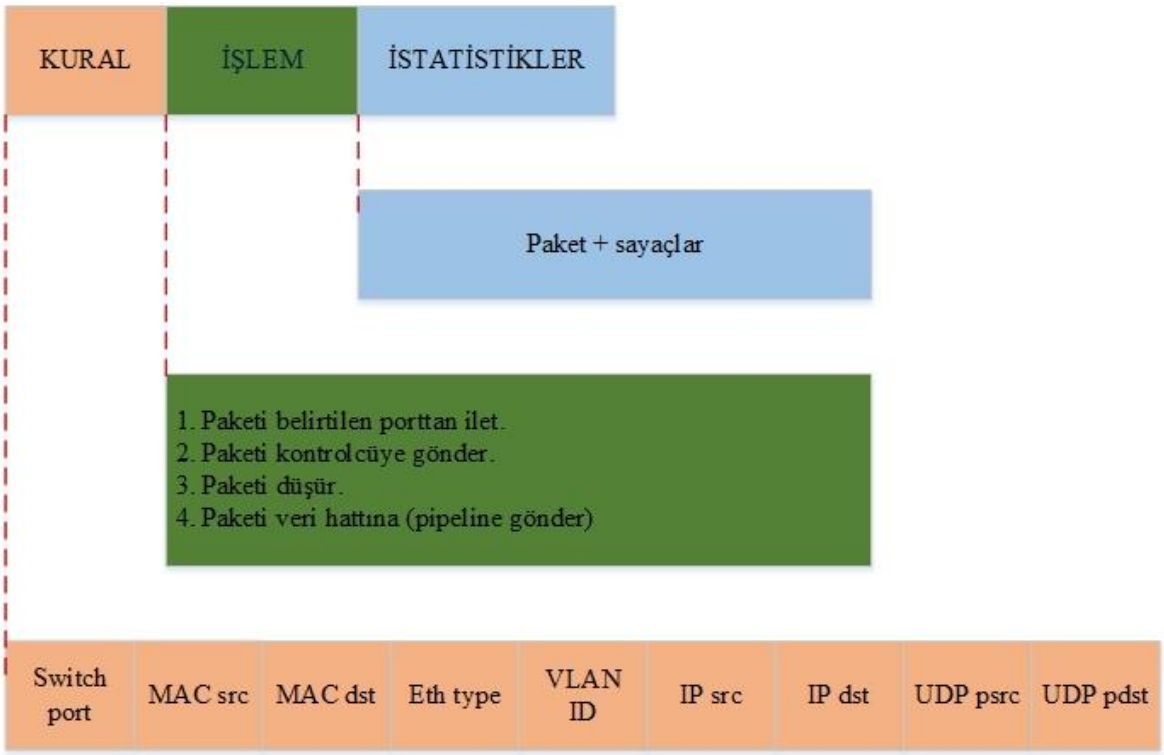
Ağa trafik matrisinde yer almayan yeni bir akış geldiğinde ise aşağıdaki adımlar verilen sırada uygulanır:

Adım 6: eşleşmeyen akışın kontrolcüye gönderilmesi

Ağdaki anahtarlardan biri, akış tablosundaki girdilerle eşleşmeyen bir akış algıladığında Şekil 5.3'te gösterilen 2 numaralı işlem gerçekleştirilir ve bu akış bir paket haline getirilerek, ne yapılacağını sormak için güvenli OpenFlow kanalı üzerinden kontrolcüye iletilir.

Adım 7: enerji-Etkin VSF Yerleştirme ve Trafik Yönlendirme Modülü 'nün bilgilendirilmesi

Kontrolcü paketi aldığında yeni akış hakkında bilgi sahibi olur ve akış için gerekli hesaplamaların başlatılması amacıyla Enerji-Etkin VSF Yerleştirme ve Trafik Yönlendirme Modülü ile iletişim kurar. Bu modül Adım 2'de açıklandığı üzere, trafik sınıflandırıcısı tarafından bu akış için güvenlik gereksinimleri belirlenir ve bu gereksinimler doğrultusunda, geliştirilen sezgisel algoritma tarafından akışın yönlendirilmesi gereken rota hesaplanır. Akışın güvenlik gereksinimlerini karşılayacak herhangi bir rota bulunamazsa, akış tarafından istenilen VSF türlerinin yenileri oluşturulur. Bu işlemin sezgisel algoritma tarafından nasıl yapıldığı Bölüm 4'te detaylı bir şekilde açıklanmıştır.



Şekil 5.3. OpenFlow anahtarlarının akış tablosunda gerçekleşen işlemler

Adım 8: eşleşmeyen akış için hesaplanan rotanın kontrolcüye iletilmesi:

Kontrolcü, eşleşmeyen akış için Enerji-Etkin VSF Yerleştirme ve Trafik Yönlendirme Modülü tarafından hesaplanan rota hakkında Kuzey ara yüzü aracılığıyla bilgilendirilir.

Adım 9: eşleşmeyen akış için belirlenen kuralların akış tablolarına yazılması:

Kontrolcü eşleşmeyen akış için hesaplanan rotayı öğrendikten sonra, anahtarların akış tablolarını günceller ve bu akışın nasıl yönlendirileceği konusunda veri katmanındaki anahtarları bilgilendirir. Bu işlem sırasında kontrolcünün *_handle_PacketIn (event)* metodu çalıştırılır ve eşleşmeyen akış için hesaplanan yönlendirme kuralı ağ topolojisindeki ilgili anahtarların akış tablosuna yazılır. Yazma işleminin detayları Adım 4'te detaylı bir şekilde açıklanmıştır.

Sonuç olarak, SDN ve NFV teknolojilerinin bir arada kullanılmasıyla geliştirilen böyle bir yaklaşım ile:

- Trafik akışları için belirlenen rotalar ağ altyapısına hızlı bir şekilde iletilebilmekte,

- Kurallar dinamik bir şekilde hemen uygulanabilmekte,
- Sanal ağ güvenliği fonksiyonları ağda istenilen yere kolay bir şekilde yerleştirilebilmekte
- Ağ koşulları değiştiğinde şekilde VSF'ler otomatik bir başlatılıp durdurulabilmekte,
- Ağ koşullarına bağlı olarak VSF'ler için yeniden yerleştirme yapılabilmekte ve
- Ağ koşullarına bağlı olarak trafik için yeni rotalar hesaplanabilmektedir.

5.2. Deneysel Çalışma

Bu tez kapsamında geliştirilmiş olan SDN kontrolcü yazılımının performansını ve etkinliğini değerlendirmek için yapılan deneyler aşağıdaki hedefler göz önünde bulundurularak tasarlanmıştır:

- Geliştirilen sezgisel algoritma tarafından belirlenen yolların ulaşılan toplam bant genişliği, paket kayıp oranı ve ortalama gecikme gibi ağ parametreleri üzerindeki etkisinin ölçülmesi,
- Geliştirilen sezgisel algoritma tarafından belirlenen yolların değişen trafik hacmi altındaki davranışlarının analiz edilmesi

Bu bölümün geri kalanında, deney ortamı, kullanılan veriler, teknolojiler ve araçlar hakkında bilgi verilmiş, değerlendirme metrikleri açıklanmış ve deneysel sonuçlar sunulmuştur.

5.2.1. Emülasyon ortamı ve kullanılan veriler

Bu tez çalışması kapsamında, SDN ortamında yapılan deneyler açık kaynaklı bir ağ emülatörü olan Mininet üzerinde gerçekleştirilmiştir [122]. Mininet, yazılım tanımlı ağlar için geliştirilen uygulamaların test edilmesini sağlayan bir geliştirme ortamıdır. Sanal bilgisayarlar, anahtarlar, kontrolcü yazılımı ve linklerden oluşan bir ağın benzetiminin yapılmasını sağlayan bir ağ emülatörüdür. Mininet ile herhangi bir bilgisayar üzerinde SDN uygulamaları geliştirilebilmekte ve bu uygulamalar herhangi bir değişikliğe veya güncellemeye ihtiyaç duyulmadan gerçek donanımlar üzerine sorunsuz bir şekilde aktarılabilmektedir.

Tez çalışmasının bu bölümünde Mininet emülatörü ile yapılan deneylerde, kontrol katmanında modüler POX kontrolcüsü [123] kullanılmış ve geliştirilen sezgisel algoritma VSF yerlerinin belirlenmesi ve trafik akışlarının yönlendirilmesi için bu kontrolcü üzerinde yeni bir modül olarak gerçeklenmiştir. Veri katmanında ise Mininet emülatörünün desteklediği SDN tabanlı Open vSwitch anahtarlar kullanılmıştır. Ayrıca Mininet ile ağ topolojisindeki her anahtar için bir NFV sunucusu oluşturulmuştur. Geliştirilen SDN kontrolcü yazılımının performansı da, Bölüm 4'te yapılan ölçümlerle karşılaştırılabilmesi için, bir önceki bölümde kullanılan ve 12 anahtar, 15 çift yönlü link ve 132 trafik akışından oluşan Internet2 veri seti üzerinde değerlendirilmiştir. Bu veri setindeki trafik verisinin Mininet ortamında üretilebilmesi için ise literatürde sıklıkla tercih edilen Iperf aracı kullanılmıştır [79].

5.2.2. Değerlendirme metrikleri

Geliştirilen SDN kontrolcü yazılımını test etmek amacıyla kullanılan değerlendirme parametreleri; Iperf tarafından üretilen çıktıda yer alan ve literatürdeki çalışmalarda performans değerlendirmesi amacıyla kullanılan, toplam bant genişliği, paket kayıp oranı, ortalama gecikme ve ortalama akış yol uzunluğudur [79]. Bu metrikler aşağıda kısaca açıklanmıştır.

Toplam bant genişliği: Değişen trafik akışı sayısı için ağdaki toplam veri aktarım hızıdır.

Transfer edilen toplam veri: Deney süresince ağda iletilen toplam veri miktarıdır.
Ortalama jitter: Jitter, paket iletiminde meydana gelen gecikmedeki değişim oranıdır. Bir akışın geçtiği toplam düğüm sayısı N ve i. sıradaki paketin n düğümünde yaşadığı gecikme $T_i^{(n)}$ şeklinde ifade edilecek olursa, bu akış için Iperf tarafından hesaplanan uçtan uca jitter değeri Eş. (5.1)'de ifade edilen beklenen değere eşittir [124].

$$J^{(N)} = E\left[\left|\sum_{n=1}^{N} T_{i+1}^{(n)} - T_i^{(n)}\right|\right] \tag{5.1}$$

Ortalama jitter ise her bir akış için hesaplanan jitter değerlerinin aritmetik ortalamasıdır.

Ortalama paket kayıp oranı: Ağda gönderilen paketlerin ne kadarının kaybolduğunu ifade

etmektedir. Daha açık bir ifadeyle, kaybolan paket sayısının gönderilen toplam paket sayısına oranıdır. Ortalama paket kayıp oranı ise her bir akış için hesaplanan kayıp oranı değerlerinin aritmetik ortalamasıdır.

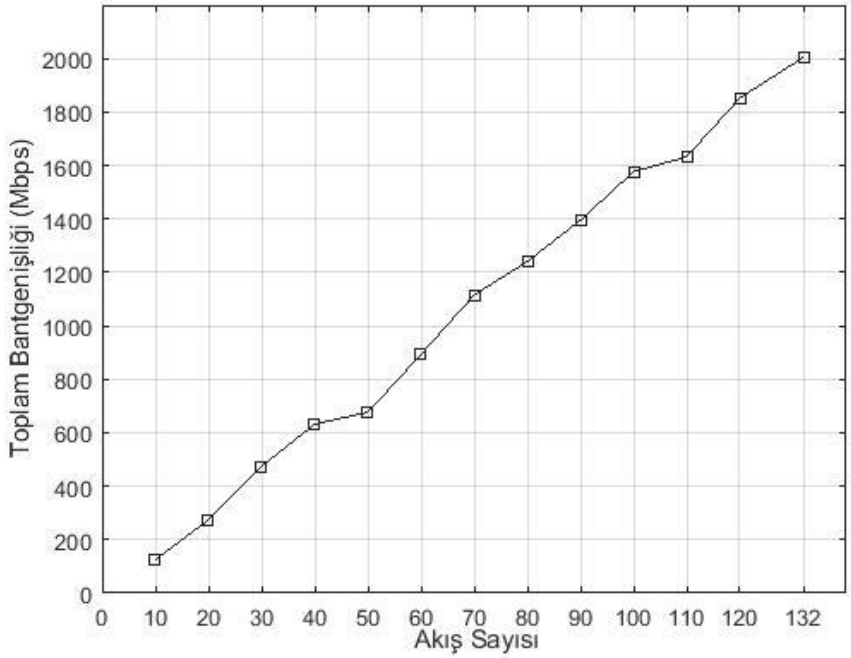
Ortalama gecikme: Trafik akışlarının kaynak düğümünden varış düğümüne gitmesi için geçen süredir.

Ortalama akış yolu uzunluğu: Trafik akışları tarafından takip edilen yollardaki ortalama atlama (hop) sayısı olarak ifade edilmektedir. Bu metriğin genel olarak ağdaki gecikmeyi etkileyen en önemli faktörlerden biri olduğu bilinmektedir çünkü ağda iletilen paketler kaynak düğümünden varış düğümüne yönlendirilirken geçtiği yol üzerindeki her düğümde çeşitli gecikme türlerine maruz kalmaktadır. Dolayısıyla akışların yol uzunluklarının artması, akışları oluşturan paketlerin işleneceği düğüm sayısının artması anlamına gelmekte ve toplam gecikmenin de artmasına yol açabilmektedir [118].

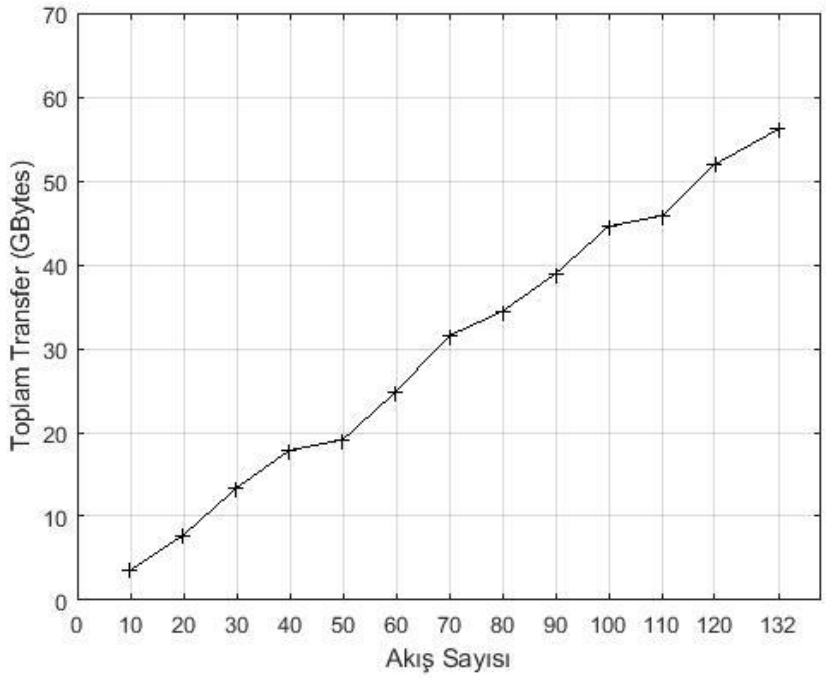
5.2.3. Sonuçlar ve tartışma

Geliştirilen kontrolcü yazılımı Internet2 topolojisi üzerinde değişen sayıda trafik akışı için yukarıda açıklanmış olan parametreler açısından değerlendirilmiştir. Deneyler, her bir akış sayısı için beş kere tekrarlanarak elde edilen sonuçların ortalaması alınmıştır. Elde edilen bulgular aşağıda detaylı bir şekilde açıklanmaktadır.

Internet2 topolojisi için, Şekil 5.4'te de gösterildiği gibi, toplam bant genişliğinin 132 trafik akışına kadar sürekli artış eğiliminde olması ağ kapasitesinin aşılmadığını ve ağın daha çok trafik akışına hizmet verebilecek durumda olduğunu göstermektedir. Bu da geliştirilen sezgisel algoritma tarafından belirlenen yolların ağ kapasitesi açısından herhangi bir probleme yol açmadığını göstermektedir. Şekil 5.5'te akış sayısına bağlı olarak transfer edilen toplam veri miktarı gösterilmiştir. Beklendiği üzere, akış sayısı arttıkça toplam bant genişliği gibi transfer edilen toplam veri miktarı da sürekli biçimde artmaktadır.



Şekil 5.4. Internet2 topolojisi üzerinde değişen trafik akışı sayısı için ulaşılan toplam bant genişliği



Şekil 5.5. Internet2 topolojisi üzerinde değişen trafik akışı sayısı için transfer edilen toplam veri miktarı

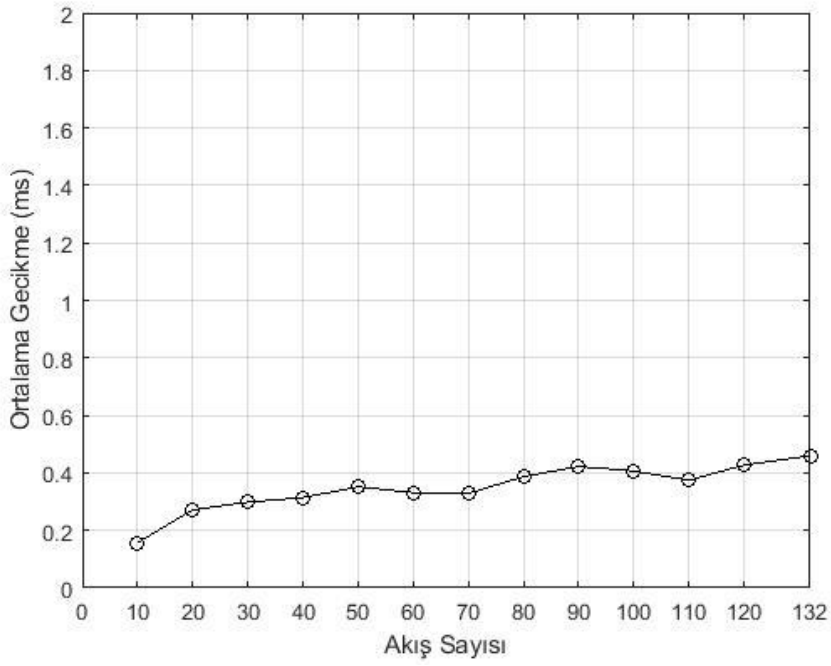
Bir diğer değerlendirme metriği olan jitter açısından bakıldığında ise paket iletimindeki gecikmede meydana gelen değişimin Internet2 topolojisi için 0,005 ms ile 0,008 ms arasında değiştiği görülmektedir. Ölçülen bu değerler oldukça düşük olup kabul edilebilir jitter aralığında yer almaktadır [125]. Bu da ağda tıkanıklıktan kaynaklanan herhangi bir gecikme olmadığını göstermektedir, çünkü yüksek jitter değerlerinin teme sebeplerinden biri ağda tıkanıklık olmasıdır. Ağda tıkanıklık varsa paketlerin yönlendiricilerdeki kuyruk yapılarında bekleme süreleri artacak bu da paketlerin gönderim hızıyla alım hızı arasındaki dalgalanmaya sebebiyet vererek jitter değerini artıracaktır [126, 127].

Bunlara ek olarak, paket kayıp oranı açısından bir değerlendirme yapıldığında, 50 trafik akışına kadar ağda herhangi bir kayıp olmadığı gözlenmiştir. 50 akıştan sonra kayıplar görülse de, ulaşılan maksimum kayıp oranının %0,4 olduğu gözlenmiştir. Bu da yukarıdaki bulguları destekler nitelikte olup ağda herhangi bir tıkanıklık olmadığını göstermektedir. Dolayısıyla bu bulgular birlikte değerlendirildiğinde, geliştirilen algoritma tarafından enerji etkin bir mantıkla belirlenen yolların ağ performansında herhangi bir bozulmaya neden olmadığı sonucuna varılmaktadır.

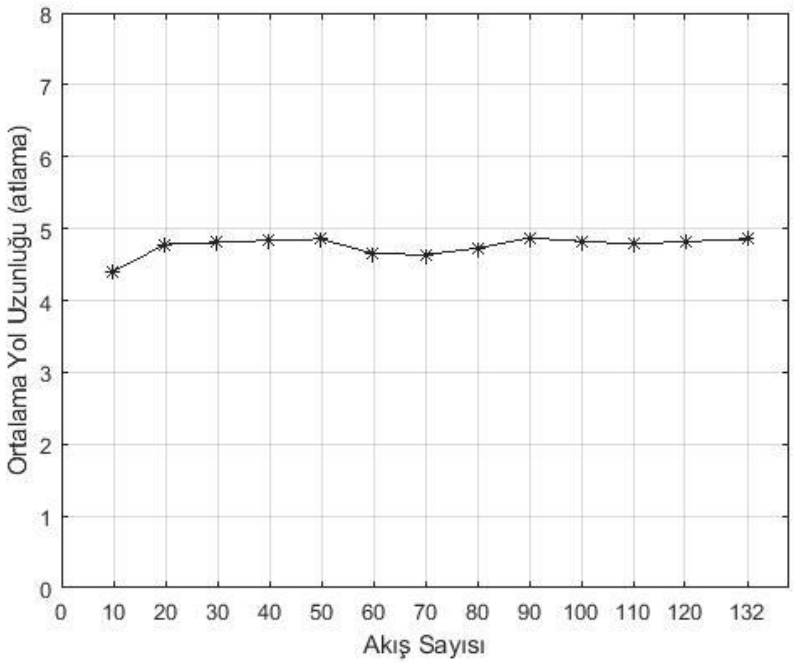
Geliştirilen SDN kontrolcüsünün performansını değerlendirme amacıyla ölçümlenen bir diğer metrik de ortalama gecikme değeridir. Bölüm 4'te de açıklandığı üzere, geliştirilen sezgisel algoritma ILP modelinden çok daha kısa sürede optimum yerleştirme çözümlerini üretmekte ve bunu yaparken ağdaki gecikmeyi etkilediği bilinen ortalama akış yolu uzunluğunu ciddi oranda düşürmektedir. Internet2 topolojisi için ise geliştirilen algoritma ILP modeline yolları ortalama %29 oranında kısaltmaktadır. Çizelge 5.2'de bu bölümde yapılan deneylerde elde edilen ortalama gecikme ve ortalama yol uzunluğu değerleri değişen trafik akışı sayısına bağlı olarak gösterilmektedir. Görüldüğü gibi, ortalama yol uzunluğunun arttığı yerde ortalama gecikme de artmakta ve ortalama yol uzunluğunun azaldığı yerde ortalama gecikme de azalmaktadır. Ayrıca, Şekil 5.6 ve Şekil 5.7'de bu değerlerin artış ve azalış eğilimlerinin değişen trafik akışı sayıları için aynı yönde olduğu daha net bir şekilde gösterilmektedir.

Çizelge 5.2. Değişen trafik akışı sayısına bağlı olarak elde edilen ortalama gecikme ve ortalama yol uzunluğu değerleri

Akış Sayısı	Ortalama Gecikme (ms)	Ortalama Yol Uzunluğu (atlama)
10	0.157	4.406
20	0.2717	4.78
30	0.2981	4.806
40	0.3143	4.836
50	0.3513	4.854
60	0.3316	4.65
70	0.3282	4.636
80	0.3871	4.7265
90	0.4223	4.869
100	0.4069	4.824
110	0.3743	4.783
120	0.4271	4.82
132	0.4596	4.852



Şekil 5.6. Internet2 topolojisi üzerinde değişen trafik akışı sayısı için elde edilen ortalama gecikme değerleri



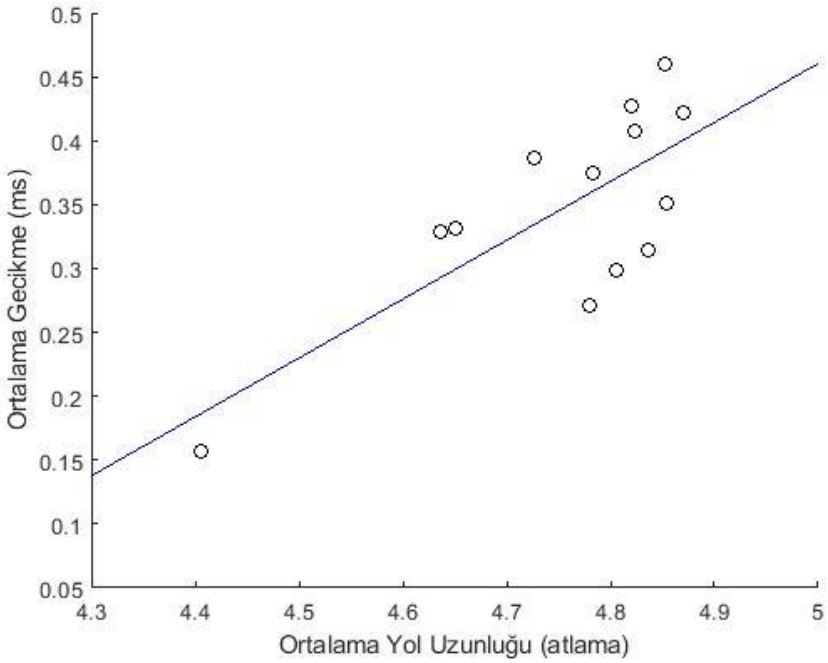
Şekil 5.7. Internet2 topolojisi üzerinde değişen trafik akışı sayısı için elde edilen ortalama yol uzunluğu değerleri

Şekil 5.8'de ise ortalama yol uzunluğu ile ortalama gecikme değerleri arasındaki ilişki gösterilmiştir. Şekilden de görülebildiği gibi, ortalama yol uzunluğu arttıkça ortalama gecikme değeri de genel olarak artış gösterme eğilimindedir. Dolayısıyla, bu değerlerin eğilimlerinin aynı yönde olması, sezgisel algoritmanın yol uzunluklarını kısaltarak gecikmeyi de azalttığını göstermektedir. Bu da geliştirilen algoritma tarafından belirlenen yolların hizmet kalitesi (quality of service) parametreleri açısından etkin yönlendirme yapabildiği ve sorunsuz performans ile çalıştığı şeklinde yorumlanmaktadır.

5.3. Bölüm Değerlendirmesi

Tez çalışmasının bu bölümünde, Bölüm 4'te geliştirilmiş olan enerji etkin VSF yerleştirme algoritmasının aldığı kararlar doğrultusunda ağa gelen trafik akışlarını yönlendiren SDN kontrolcü yazılımı geliştirilmiştir.

Geliştirilen kontrolcü, literatürde performans değerlendirmesi amacıyla kullanılan toplam bant genişliği, transfer edilen toplam veri miktarı, jitter değeri, paket kayıp oranı, ortalama gecikme ve ortalama akış yol uzunluğu gibi ağ parametreleri açısından test edilmiş ve performans ölçümleri yapılmıştır.



Şekil 5.8. Ortalama yol uzunluğu ve ortalama gecikme değerleri arasındaki ilişki

Ölçümler sonucunda; trafik hacmine bağlı olarak transfer edilen toplam veri miktarı ve hızı artmaya devam etse de kontrolcünün trafik akışlarını etkin bir şekilde yönlendirebildiğini göstermiştir. Bu da ağ kapasitesinin aşılmadığını ve ağın daha çok trafik akışına hizmet verebilecek durumda olduğunu göstermektedir.

Ayrıca, deneylerin çoğunda paket kaybının hiç yaşanmaması ve gözlenen maksimum kayıp oranının %0,4 olması sezgisel algoritma tarafından belirlenen yolların ağda herhangi bir tıkanıklığa sebep olmadığını göstermektedir. Tez kapsamında geliştirilen yöntemlerin temel amacı olan enerji tüketimini azaltmak için aktif sunucu sayısını minimize etme, ağdaki belirli noktalarda trafik yükünün artması olasılığını artırsa da, geliştirilen sezgisel algoritma tarafından belirlenen yolların ağ performansında herhangi bir bozulmaya neden olmadığı gösterilmiştir.

Son olarak, ortalama yol uzunluğu ve ortalama gecikme değerlerinin, değişen sayıda trafik akışı için genellikle aynı yönde hareket ettiği gösterilmiştir. Bu da geliştirilen sezgisel algoritmanın yol uzunluklarını geliştirilen ILP modeline kıyasla önemli ölçüde azaltmasının ağdaki uçtan uca gecikme üzerinde etkili olduğunu ve ortalama gecikme değerlerini azalttığını göstermektedir.

Literatürdeki VSF yerleştirme çalışmaları incelendiğinde, bu çalışmalarda ele alınan amaçlar doğrultusunda ağa gelen trafik akışlarını yönlendirme amacıyla geliştirilmiş herhangi bir kontrolcü yazılımına bilgimiz dâhilinde rastlanmamıştır. Fakat VSF yerleştirme problemi ile benzer olduğu bilinen orta kutu yerleştirme problemini ağ yükünü dengeleme amacıyla ele alan ve bu doğrultuda bir kontrolcü yazılımı geliştiren bir çalışmada [79], kontrolcü yazılımının performansı paket kaybı ve uçtan uca gecikme değerleri bakımından değerlendirilmiştir. Ma ve arkadaşları tarafından yapılan bu çalışmada [79], farklı topoloji ve trafik kümesi kullanıldığı için bu tez kapsamında elde edilen sonuçlar ile birebir kıyaslanamamaktadır. Fakat yine de belirtilen performans metrikleri açısından yorumlanacak olursa [79]'da elde edilen paket kayıp oranı değerlerinin %0 ile %0,8 arasında değiştiği ifade edilmiştir. Dolayısıyla, tez çalışması kapsamında elde edilen paket kayıp oranı bu aralıkta yer aldığından kabul edilebilir olduğu yorumlanmaktadır.

6. SONUÇ VE ÖNERİLER

SDN ve NFV teknolojileriyle yapılandırılmış ağlarda ağ güvenliği; IDS, IPS, DPI, güvenlik duvarı vb. güvenlik fonksiyonlarının sanallaştırılması ve genel amaçlı sunucular üzerinde birer yazılım olarak çalıştırılmasıyla sağlanmaktadır. Bu fonksiyonların ağda hangi sunucular üzerine yerleştirileceği ise ele alınması gereken önemli bir problemdir. Çünkü yerleştirme yapılırken ağa gelen trafik akışlarının güvenlik gereksinimlerinin karşılanmasının yanı sıra, o ağ için erişilmesi gereken operasyonel amaçların (maliyetin en aza indirgenmesi, gecikmenin en aza indirgenmesi, maksimum ağ yükünün en aza indirgenmesi vb.) da göz önünde bulundurulması gerekmektedir. Bu tez çalışması kapsamında, ağa gelen trafiğin güvenlik gereksinimleri göz önünde bulundurularak enerji etkin bir şekilde sanal ağ güvenliği fonksiyonlarının yerleştirilmesi ve trafik akışlarının güvenlik gereksinimleri karşılanacak şekilde bu VSF'ler üzerinden yönlendirilmesi problemi modellenmiş ve çözülmüştür. Aşağıda verilen alt başlıklarda, tez kapsamında yapılanlar özetlenerek çalışmanın literatüre katkısı değerlendirilmiş ve gelecek çalışma önerileri sunulmuştur.

6.1. Sonuçlar ve Değerlendirme

Tez kapsamında yapılan çalışmalar ve elde edilen sonuçlar aşağıda üç başlık halinde özetlenmiş olup, literatüre yapılan katkılar Şekil 6.1'de sunulmuştur.

Geliştirilen taksonomiler ve literatür çalışması

- Sanal ağ güvenliği fonksiyonlarının sınıflandırılması için yeni bir yaklaşım önerilmiş (Bölüm 3.1) ve bu fonksiyonlar literatürde ilk defa; saldırı tespit, saldırı engelleme, saldırı aldatma ve saldırı hafifletme fonksiyonları olacak şekilde 4 başlık altında sınıflandırılmıştır. Bu kategorilerin her biri açıklanmış ve ağ uygulamalarında aktif bir şekilde kullanılan ve literatürde yer alan yeni yaklaşımlar incelenerek bu kategorilere ait sanal güvenlik fonksiyonları tanıtılmıştır. Ayrıca, her bir sanal güvenlik fonksiyonu alanında yapılan son çalışmalara genel bir bakış sunulmuştur.

- Bu bağlamda elde edilen bulgular değerlendirildiğinde, SDN'de ağ alt yapısı yazılım ile kontrol edildiğinden, bu ortamlar için geliştirilen sanal ağ güvenliği fonksiyonu çözümlerinin geleneksel ağlara göre daha iyi çözümler ürettiği değerlendirilmiştir.

Şekil 6.1. Tez kapsamında literatüre yapılan katkılar

- SDN ve NFV teknolojileriyle yapılandırılmış ağlarda sanal güvenlik fonksiyonlarının yerleştirilmesi alanındaki çalışmaların sınıflandırılması için literatürde ilk defa bir taksonomi geliştirilmiştir. Geliştirilen taksonomiye göre literatürdeki VSF yerleştirme çözümleri; geliştirilen yerleştirme yaklaşımları, ele alınan operasyonel amaçlar, kullanılan optimizasyon yöntemleri, akış yönetimi stratejileri, uygulama alanları, kullanılan veri, optimizasyon araçları ve simülasyon/emülasyon ortamları olmak üzere 8 boyutta sınıflandırılmıştır. Mevcut VSF yerleştirme çözümlerinde ele alınan operasyonel amaçlar da sınıflandırılarak bu amaçları formüle ederken kullanılan

parametreler belirlenmiştir. Literatürdeki mevcut çalışmalar ele aldıkları bu parametreler doğrultusunda tartışılmış ve özetlenmiştir. Son olarak, bu çalışmalarda önerilen matematiksel ve algoritmik yaklaşımlar tartışılarak karşılaştırılmıştır. Bu bağlamda elde edilen bulgular değerlendirildiğinde, VSF yerleştirme çalışmalarında ele alınan operasyonel amaçların; maliyetin, gecikmenin ve maksimum ağ yükünün en aza indirgenmesi olarak üç başlık altında toplandığı görülmüştür. Çalışmaların büyük çoğunluğunda; (i) yerleştirilen VSF sayısı, (ii) donanımsal ekipman sayısı ve (iii) ağ kaynaklar kullanımının (bant genişliği, linklerin sayısı, CPU kullanımı vb.) bir kombinasyonu olarak tanımlanan maliyetin en aza indirgenmesi hedeflenmiştir. Fakat bu çalışmaların hiçbirinde, ağ maliyetinin en önemli bileşenlerinden biri olan enerji tüketiminin ele alınmadığı görülmüştür. Bunlara ek olarak, trafiği VSF'ler üzerinden yönlendirirken, akış bazında güvenlik gereksinimlerini göz önünde bulunduran herhangi bir çalışmaya rastlanmamıştır.

Geliştirilen çözümler

- Bu tez kapsamında literatürde ilk defa, ağa gelen trafik akışlarının güvenlik gereksinimlerini karşılayacak şekilde enerji tüketiminin en aza indirgenmesi amacıyla sanal ağ güvenliği fonksiyonlarının yerleştirilmesi problemi modellenmiş ve çözülmüştür.
- Ele alınan problem doğrultusunda, geliştirilen ILP modeli ile ağdaki operasyonel ve güvenlik kısıtların yanı sıra trafiğin akış bazında güvenlik gereksinimleri de göz önünde bulundurularak enerji etkin yerleştirme çözümleri üretilmiştir.
- Ele alınan problem NP-zor sınıfında yer aldığından, geliştirilen ILP modelinin ölçeklenebilirlik sorununu çözmek amacıyla geliştirilen sezgisel algoritma ile enerji tüketimini en aza indirgeyen yerleştirme çözümleri ILP modeline göre çok daha kısa sürede üretilmiştir.
- Geliştirilen ILP modeli ve sezgisel algoritma, gerçek dünya topolojisi üzerinde gerçek trafik verisi ile test edilerek performans değerlendirmeleri yapılmıştır. Ayrıca geliştirilen çözümler, literatürde karşılaştırma ve değerlendirme amacıyla kullanılan referans yerleştirme çözümleri ile de kıyaslanarak geliştirilen algoritmanın başarısı ölçülmüştür.
- Elde edilen sonuçlar; geliştirilen sezgisel algoritmanın, literatürdeki referans yerleştirme çözümlerine göre enerji tüketimini %48 oranında azalttığını göstermiştir.

- VSF yerleştirme problemi ile benzer olduğu bilinen fonksiyon/sanal makine/orta kutu/ vb. yerleştirme problemlerini enerji etkin bir biçimde çözmeyi hedefleyen çalışmalar [34, 104, 105, 108, 112, 113, 119] ile kıyaslandığında, bu çalışmalarda elde edilen enerji tasarrufu değerlerinin %13,4 ile %46,1 arasında değiştiği ve dolayısıyla tez kapsamında elde edilen değerin kabul edilebilir aralıkta olduğu yorumlanmıştır.
- Bunlara ek olarak, geliştirilen sezgisel algoritmanın ILP modeline göre bazı ağlarda ağ gecikmesini etkileyen en önemli faktörlerden biri olan akışların yol uzunluğunu %51'e kadar azalttığı görülmüştür.

SDN uygulaması

- Enerji etkin VSF yerleştirme algoritmasının aldığı kararlar doğrultusunda ağa gelen trafik akışlarını yönlendirecek SDN kontrolcü yazılımı geliştirilerek, önerilen yaklaşımın yazılım tanımlı ağ altyapısında uygulanabilirliği gösterilmiştir.
- Yazılım tanımlı ağ altyapısında uygulanabilirliğini göstermek için enerji etkin VSF yerleştirme algoritmasının aldığı kararlar doğrultusunda ağa gelen trafik akışlarını yönlendirecek SDN kontrolcü yazılımı geliştirilmiştir.
- Geliştirilen kontrolcü, toplam bant genişliği, paket kayıp oranı, ortalama gecikme ve ortalama akış yol uzunluğu gibi hizmet kalitesi parametreleri açısından test edilmiş ve performans ölçümleri yapılmıştır.
- Elde edilen bulgular; trafik hacmine bağlı olarak transfer edilen toplam veri miktarı ve hızı artmaya devam etse de kontrolcünün trafik akışlarını etkin bir şekilde yönlendirebildiğini göstermiştir.
- Ayrıca sezgisel algoritma tarafından belirlenen yolların ağda herhangi bir tıkanıklığa sebep olmadığı ve paket kayıp oranının çoğu deneyde 0 olduğu elde edilen sonuçlar arasındadır.
- Bunlara ek olarak, geliştirilen sezgisel algoritmanın yol uzunluklarını %51'e kadar azaltmasının ağdaki gecikme üzerinde de önemli bir etkisi olduğu ve yol uzunlukları azaldıkça ağdaki uçtan uca gecikmenin de azaldığı görülmüştür.

6.2. Karşılaşılan Zorluklar

Bu tez kapsamında karşılaşılan temel zorluklar;

- Bir SDN kontrolcüsünün nasıl geliştirileceği, belirlenen kuralları veri katmanındaki anahtarların akış tablolarına nasıl yazacağı, her bir işlem adımında hangi fonksiyonların çalıştırılması gerektiği vb. ile ilgili yeterli sayıda dokümanın mevcut olmaması,
- Mininet emülatörü hipervizör yazılımı üzerinde çalıştırıldığında Iperf ile üretilen trafiğin istenen bant genişliği değerlerine ulaşamaması,
- İstenen değerlerle trafik akışlarının üretilebilmesi amacıyla Mininet'in çalıştırılabilmesi için tamamen bu amaca yönelik Linux işletim sistemli bir bilgisayara ihtiyaç duyulması,
- Ve son olarak, tez kapsamında ele alınan VSF yerleştirme problemi literatürde net bir şekilde tanımlanmış olsa da bu problemi çözmek amacıyla izlenmesi gereken metodolojiyi, geliştirilen yaklaşımları ve alandaki eksiklikleri değerlendiren herhangi bir çalışma, taksonomi, alan değerlendirmesi vb. olmaması

olarak sıralanabilir.

6.3. Gelecek Çalışma Önerileri

Bu tez çalışması kapsamında, VSF yerleştirme probleminde ele alınması gereken açık araştırma alanları belirlenerek bu alanlarda neler yapılabileceği devam eden alt başlıklarda gelecek çalışma önerileri olarak sunulmuştur.

6.3.1. Yeni nesil ağ ortamları

Literatürdeki mevcut VSF yerleştirme çözümleri incelendiğinde, Bölüm 3.3.5'te de ifade edildiği üzere, çoğunluğun uygulama alanının veri merkezi ve internet servis sağlayıcısı ağları olduğu görülmüştür. Fakat NFV teknolojisinin en motive edici özelliklerinden biri, yazılımlaştırmayı destekleyerek ağ fonksiyonlarının donanım bağımsız hale getirilmesini sağladığı için çok çeşitli ağ ortamlarında uygulanabilir olmasıdır. Dolayısıyla bu fonksiyonların, özellikle 5G başta olmak üzere, mobil ortamlarda nerelere yerleştirileceği ele alınması gereken önemli problemler arasındadır.

Bu tez kapsamında yapılan çalışma 5G gibi yeni nesil ortamları için uygulanabilir olsa da, VSF yerleştirme yapılırken bu ortamlara özel olan hareketlilik yönetimi varlıkları (mobility management entities - MME), hizmet geçitleri (serving gateways - SGW) ve paket veri ağı geçitleri (packet data network gateways - PGW) gibi mobil fonksiyonların, yerleştirilecek olan sanal ağ güvenliği fonksiyonları ile olan ilişkisinin göz önünde bulundurulması önemli gelecek çalışma hedefleri arasında yer almaktadır.

6.3.2. Gerçek dünya uygulamaları

VSF yerleştirme problemi NP-Zor sınıfında yer aldığından herhangi bir amaç doğrultusunda VSF'ler için optimum konumların bulunması hesaplama açısından oldukça kompleks bir işlemdir. Dolayısıyla, doğrusal programlama gibi yöntemlerle optimum çözümün bulunması büyük ölçekli ağlar için uygulanabilir değildir. Bu nedenle araştırmacılar, matematiksel modeller ile optimum çözümlerin belirlenmesini beklemenin imkânsız olduğu durumlarda kullanılması amacıyla sezgisel algoritmalar geliştirmektedirler. Geliştirilen çözümlerin gerçek dünya problemleri üzerinde uygulanabilirliğini test etmek için ise emülasyon ortamlarının yanı sıra bu algoritmaların gerçek OpenFlow anahtarları ve bunlara bağlı sunuculardan oluşan gerçek laboratuvar ortamlarında hayata geçirilmesi gerekmektedir. Bu nedenle bu çalışmanın gelecek hedefleri arasında, sektörle yapılan anlaşmalar doğrultusunda, geliştirilen SDN kontrolcü yazılımının 5G test merkezinde gerçek bir laboratuvar ortamında uygulamaya geçirilmesi yer almaktadır.

6.3.3. Büyük veri analitiği ve yapay zekâ uygulamaları

Literatürdeki mevcut VSF yerleştirme yaklaşımları SDN ile yapılandırılmış ağlarda hayata geçirilirken farklı istatistiksel yaklaşımların kullanılması birçok açıdan yararlı olacaktır. Büyük veri çözümleri, bu yaklaşımlardan en önemlileri olarak düşünülmektedir. SDN kontrolcüsü tarafından işlenen trafik verisi büyük hacimlere ulaştığında, büyük veri yaklaşımlarının geliştirilmesi gerekliliği yeni bir bakış açısı olarak karşımıza çıkmaktadır. Büyük veri analitiği ile büyük veri setlerindeki gizli örüntüler ve bilinmeyen korelasyonlar çıkarılarak veriden değer üretilebilmektedir. Fakat bilgimiz dâhilinde, VSF yerleştirme problemiyle büyük veri bakış açısını bir araya getiren herhangi bir çalışma mevcut değildir. Bölüm 3.3.6'da da bahsedildiği gibi, gerçek veri seti kullanan çalışmalar bile oldukça az

sayıdadır. Bu nedenle, büyük veri analitiği yöntemleri ile elde edilmiş istatistiksel sonuçları kullanarak VSF yerleştirme yapılması oldukça ilgi çekici bir problem olup gelecek çalışma hedefleri arasında yer almaktadır. Böylelikle, geliştirilen sezgisel algoritma yapay zekâ tabanlı bir yerleştirme çözümüne dönüştürülebilecektir.

KAYNAKLAR

1. Bouet, M., Leguay, J. and Conan, V. (2013). *Cost-based placement of virtualized deep packet inspection functions in SDN*. IEEE Military Communications Conference, San Dieago, USA.

2. İnternet: A Cost-based Placement Algorithm for Multiple Virtual Security Appliances in Cloud using SDN: MO-UFLP (Multi-Ordered Uncapacitated Facility Location Problem). URL: https://arxiv.org/pdf/1602.08155.pdf, Son Erişim Tarihi: 06.03.2019.

3. Bouet, M., Leguay, J., Combe, T. and Conan, V. (2015). Cost-based placement of vDPI functions in NFV infrastructures. *International Journal of Network Management*, 25(6), 490–506.

4. Shameli-Sendi, A., Jarraya, Y., Fekih-Ahmed, M., Pourzandi, M., Talhi, C. and Cheriet, M. (2015). *Optimal placement of sequentially ordered virtual security appliances in the cloud*. IFIP/IEEE International Symposium on Integrated Network Management, Ottawa, Canada.

5. Doriguzzi-Corin, R., Scott-Hayward, S., Siracusa, D. and Salvadori, E. (2017). *Application-centric provisioning of virtual security network functions*. IEEE Conference on Network Function Virtualization and Software Defined Networks, Berlin, Germany.

6. Doriguzzi-Corin, R., Scott-Hayward, S., Siracusa, D., Savi, M. and Salvadori, E. (2019). Dynamic and Application-Aware Provisioning of Chained Virtual Security Network Functions. *IEEE Transactions on Network and Service Management*, Early Access, 1-14.

7. Xu, Q., Gao, D., Li, T. and Zhang, H. (2018). Low Latency Security Function Chain Embedding Across Multiple Domains. *IEEE Access*, 6, 14474–14484.

8. Dwiardhika, D. and Tachibana, T. (2018). *Cost Efficient VNF Placement with Optimization Problem for Security-Aware Virtual Networks*. IEEE 7th International Conference on Cloud Networking, Tokyo, Japan.

9. Phan, T. V., Bao, N. K., Kim, Y., Lee, H. J. and Park, M. (2017). *Optimizing resource allocation for elastic security VNFs in the SDNFV-enabled cloud computing*. International Conference on Information Networking, Barcelona, Spain.

10. Shameli Sendi, A., Jarraya, Y., Pourzandi, M. and Cheriet, M. (2016). Efficient Provisioning of Security Service Function Chaining Using Network Security Defense Patterns. *IEEE Transactions on Services Com*puting, 12(4), 534–549.

11. Jarraya, Y., Shameli-Sendi, A., Pourzandi, M. and Cheriet, M. (2015). *Multistage OCDO: Scalable Security Provisioning Optimization in SDN-Based Cloud*. IEEE 8th International Conference on Cloud Computing, New York, USA.

12. Liu, Y., Liu, H. Q. and Yang, Y. J. (2017). A new approach for delivering customized security everywhere: Security service chain. *Security and Communication Networks*, 1-17.

13. Liu, Yicen, Lu, Y., Qiao, W. and Chen, X. (2018). A dynamic composition mechanism of security service chaining oriented to SDN/NFV-Enabled networks. *IEEE Access*, 6, 53918–53929.

14. İnternet: Software-Defined Networking (SDN) Definition - Open Networking Foundation. URL: https://www.opennetworking.org/sdn-definition/, Son Erişim Tarihi: 31.01.2019

15. Kreutz, D., Ramos, F. M. V., Verissimo, P. E., Rothenberg, C. E., Azodolmolky, S. and Uhlig, S. (2015). Software-defined networking: A comprehensive survey. *Proceedings of the IEEE*, 103(1), 14–76.

16. İnternet: SDN Architecture Overview (HPE). URL: https://www.opennetworking.org/images/stories/downloads/sdn-resources/technical-reports/SDN-architecture-overview-1.0.pdf, Son Erişim Tarihi: 06.03.2020.

17. Ahmad, I., Namal, S., Ylianttila, M. and Gurtov, A. (2015). Security in Software Defined Networks: A Survey. *IEEE Communications Surveys & Tutorials*, 17(4), 2317–2346.

18. Jarraya, Y., Madi, T. and Debbabi, M. (2014). A survey and a layered taxonomy of software-defined networking. *IEEE Communications Surveys and Tutorials*, 16(4), 1955–1980.

19. Kim, H. and Feamster, N. (2013). Improving network management with software defined networking. *IEEE Communications Magazine*, 51(2), 114–119.

20. Hu, F., Hao, Q. and Bao, K. (2014). A survey on software-defined network and OpenFlow: From concept to implementation. *IEEE Communications Surveys and Tutorials*, 16(4), 2181–2206.

21. İnternet: ONOS - A new carrier-grade SDN network operating system designed for high availability, performance, scale-out. URL: https://onosproject.org/, Son Erişim Tarihi: 18.01.2019.

22. Sezer, S., Scott-Hayward, S., Chouhan, P., Fraser, B., Lake, D., Finnegan, J., Viljoen, N., Miller, M. and Rao, N. (2013). Are we ready for SDN? Implementation challenges for software-defined networks. *IEEE Communications Magazine*, 51(7), 36–43.

23. Abdelaziz, A., Ang, T. F., Sookhak, M., Khan, S., Vasilakos, A., Liew, C. S. and Akhunzada, A. (2016). Survey on network virtualization using openflow: Taxonomy, opportunities, and open issues. *KSII Transactions on Internet and Information Systems*, 10(10), 4902–4932.

24. İnternet: OpFlex: An Open Policy Protocol White Paper - Cisco. URL: https://www.cisco.com/c/en/us/solutions/collateral/data-center-virtualization/application-centric-infrastructure/white-paper-c11-731302.html, Son Erişim Tarihi: 24.05.2019.

25. İnternet: Northbound Interfaces Working Group - ONF. URL: https://www.opennetworking.org/images/stories/downloads/working-groups/charter-nbi.pdf, Son Erişim Tarihi: 24.05.2019.

26. Chiosi, M., Clarke, D., Willis, P., Reid, A., Feger, J., Bugenhagen, M., Khan, W., Fargano, M., Cui, C., Deng, H., Telekom, D. and Michel, U. (2012). *Network Functions Virtualisation: An introduction, benefits, enablers, challenges and call for action*. SDN and OpenFlow World Congress, Darmstadt, Germany.

27. Deng, J., Hu, H., Li, H., Pan, Z., Wang, K. C., Ahn, G. J., Bi, J. and Park, Y. (2016). *VNGuard: An NFV/SDN combination framework for provisioning and managing virtual firewalls*. IEEE Conference on Network Function Virtualization and Software Defined Network, Californa, USA.

28. Han, B., Gopalakrishnan, V., Ji, L. and Lee, S. (2015). Network function virtualization: Challenges and opportunities for innovations. *IEEE Communications Magazine*, 53(2), 90–97.

29. İnternet: ETSI - Standards for NFV - Network Functions Virtualisation | NFV Solutions. URL: https://www.etsi.org/technologies/nfv, Son Erişim Tarihi: 15.10.2019.

30. İnternet: OpenFlow-enabled SDN and Network Functions Virtualization. URL: https://www.opennetworking.org/wp-content/uploads/2013/05/sb-sdn-nvf-solution.pdf, Son Erişim Tarihi: 15.10.2019.

31. Matias, J., Garay, J., Toledo, N., Unzilla, J. and Jacob, E. (2015). Toward an SDN-enabled NFV architecture. *IEEE Communications Magazine*, 53(4), 187–193.

32. Ma, W., Medina, C. and Pan, D. (2015). *Traffic-aware placement of NFV middleboxes*. IEEE Global Communications Conference, San Diego, USA.

33. Medhat, A. M., Taleb, T., Elmangoush, A., Carella, G. A., Covaci, S. and Magedanz, T. (2017). Service Function Chaining in Next Generation Networks: State of the Art and Research Challenges. *IEEE Communications Magazine*, 55(2), 216–223.

34. Tajiki, M. M., Salsano, S., Chiaraviglio, L., Shojafar, M, and Akbari, B. (2018). Joint Energy Efficient and QoS-aware Path Allocation and VNF Placement for Service Function Chaining. *IEEE Transactions on Network and Service Management*, 16(1), 374–388.

35. Hu, H. and Ahn, G.-J. (2015). *Virtualizing and Utilizing Network Security Functions for Securing Software Defined Infrastructure*. Looking Beyond the Internet Workshops. Washington D. C., USA.

36. He, T., Toosi, A. N. and Buyya, R. (2019). Performance evaluation of live virtual machine migration in SDN-enabled cloud data centers. *Journal of Parallel and Distributed Computing*, 131, 55–68.

37. Masoudi, R. and Ghaffari, A. (2016). Software defined networks : A survey. *Journal of Network and Computer Applications*, 67, 1–25.

38. Farhady, H., Lee, H. and Nakao, A. (2015). Software-Defined Networking : A survey. *Computer Networks*, 81, 79–95.

39. Heller, B., Seetharaman, S., Mahadevan, P., Yiakoumis, Y., Sharma, P., Banerjee, S. and McKeown, N. (2010). *ElasticTree: Saving Energy in Data Center Networks*. USENIX Symposium on Networkes Systems Design and Implementation, California, USA.

40. Ballard, J. R., Rae, I. and Akella, A. (2010). *Extensible and scalable network monitoring using opensafe*. Internet Network Management Conference on Research on Enterprise Networking, California, USA.

41. Aydeger, A., Saputro, N. and Akkaya, K. (2019). A moving target defense and network forensics framework for ISP networks using SDN and NFV. *Future Generation Computer Systems*, 94, 496–509.

42. Jain, S., Kumar, A., Mandal, S., Ong, J., Poutievski, L., Singh, A., Venkata, S., Wanderer, J., Zhou, J., Zhu, M., Zolla, J., Hölzle, U., Stuart, S. and Vahdat, A. (2013). Experience with a globally-deployed software defined WAN. *ACM SIGCOMM Computer Communication Review*, 43(4), 3–14.

43. Lorenz, C., Hock, D., Scherer, J., Durner, R., Kellerer, W., Gebert, S., Gray, N., Zinner, T. and Tran-Gia, P. (2017). An SDN/NFV-Enabled Enterprise Network Architecture Offering Fine-Grained Security Policy Enforcement. *IEEE Communications Magazine*, *55*(3), 217–223.

44. Bera, S., Misra, S. and Vasilakos, A. V. (2017). Software-Defined Networking for Internet of Things: A Survey. *IEEE Internet of Things Journal*, 4(6), 1994–2008.

45. Bizanis, N. and Kuipers, F. A. (2016). SDN and Virtualization Solutions for the Internet of Things: A Survey. *IEEE Access*, 4, 5591–5606.

46. Ji, X., Yu, H., Fan, G. and Fu, W. (2017). SDGR: *An SDN-Based Geographic Routing Protocol for VANET*. IEEE International Conference on Internet of Things, Paris, France.

47. Alex, A., Ahmad, A., Mijumbi, R. and Hines, A. (2020). 5G network slicing using SDN and NFV : A survey of taxonomy , Architectures and Future Challenges. *Computer Networks*, 167.

48. Ordonez-lucena, J., Ameigeiras, P., Lopez, D., Ramos-munoz, J. J., Lorca, J. and Folgueira, J. (2017). Network Slicing for 5G with SDN / NFV : Concepts, Architectures, and Challenges. I*EEE Communications Magazine,* 55(5), 80–87.

49. Krishnaswamy, D., Kothari, R. and Gabale, V. (2016). *Latency and policy aware hierarchical partitioning for NFV systems*. IEEE Conference on Network Function Virtualization and Software Defined Network, California, USA.

50. İnternet: What is an Intrusion Detection System? URL: http://techgenix.com/intrusion_detection_systems_ids_part_i__network_intrusions_attack_symptoms_ids_tasks_and_ids_architecture/, Son Erişim Tarihi: 24.05.2019.

51. Xiong, Z. (2014). *An SDN-based IPS Development Framework in Cloud Networking Environment*. Doctoral Disseratation, Arizona State University, USA.

52. Van Adrichem, N. L. M., Doerr, C. and Kuipers, F. A. (2014). *OpenNetMon: Network monitoring in OpenFlow software-defined networks*. IEEE/IFIP Network Operations and Management Symposium: Management in a Software Defined World, Krakow, Polonya.

53. Tang, T. A., Mhamdi, L., McLernon, D., Zaidi, S. A. R. and Ghogho, M. (2018). *Deep Recurrent Neural Network for Intrusion Detection in SDN-based Networks*. IEEE Conference on Network Softwarization and Workshops, Montreal, Canada.

54. Ceron, J. M., Margi, C. B. and Granville, L. Z. (2016). *MARS: An SDN-based malware analysis solutio*n. IEEE Symposium on Computers and Communications, 525–530, Messina, Italy.

55. Carl, G., Kesidis, G., Brooks, R. R. and Rai, S. (2006). Denial-of-service attack-detection techniques. *IEEE Internet Computing*, 10(1), 82–89.

56. Braga, R., Mota, E. and Passito, A. (2010). *Lightweight DDoS flooding attack detection using NOX/OpenFlow*. Conference on Local Computer Networks, Colorado, USA.

57. İnternet: The Role of DPI in an SDN World. URL: http://www.javierrodriguez.com.es/blog/wp-content/uploads/2013/05/Heavy_Reading-Qosmos_DPI-SDN-WP_Dec-2012.pdf, Son Erişim Tarihi: 24.05.2019.

58. Hu, H., Han, W., Ahn, G.-J. and Zhao, Z. (2014). *FlowGuard: Building Robust Firewalls for Software-Defined Network*s. Workshop on Hot Topics in Software Defined Networking, Chicago, USA.

59. Suh, M., Park, S. H., Lee, B. and Yang, S. (2014). *Building firewall over the software-defined network controller*. International Conference on Advanced Communication Technology, PyeongChang, Korea.

60. Dolberg, L., Festor, O. and Engel, T. (2014). *Network Security through Software Defined Networking : a Survey*. ACM Conference on Principles, Systems and Applications of IP Telecommunications, Chicago, USA.

61. Mckeown, N., Anderson, T., Peterson, L., Rexford, J., Shenker, S., Louis, S. (2008). OpenFlow: enabling innovation in campus networks, *ACM SIGCOMM Computer Communication Review*, 38(2), 69–74.

62. Hu, H, Ahn, G., Han, W. and Zhao, Z. (2014). *Towards a Reliable SDN Firewall*. Open Networking Summit, Santa Clara, Calif, USA.

63. Zhang, L., Shou, G., Hu, Y. and Guo, Z. (2013). *Deployment of Intrusion Prevention System based on Software Defined Networking*. International Conference on Communication Technology, , Bandung, Indonesia.

64. Kwon, J., Seo, D., Kwon, M., Lee, H., Perrig, A. and Kim, H. (2015). An incrementally deployable anti-spoofing mechanism for software-defined networks. *Computer Communications*, 64, 1–20.

65. Yao, G., Bi, J., Feng, T., Xiao, P. and Zhou, D. (2014). *Performing software defined route-based IP spoofing filtering with SEFA*. International Conference on Computer Communications and Networks, Toronto, Canada.

66. Cox, J. H., Clark, R. J. and Owen, H. L. (2016). *Leveraging SDN for ARP security*. IEEE South East Conference, Nortfolk, USA.

67. Alharbi, T., Durando, D., Pakzad, F. and Portmann, M. (2016). *Securing ARP in Software Defined Networks*. IEEE Conference on Local Computer Networks, Dubai, UAE.

68. Ali, S. T., Sivaraman, V., Radford, A. and Jha, S. (2015). A Survey of Securing Networks Using Software Defined Networking. *IEEE Transactions on Reliability*, 64(3), 1086–1097.

69. Shin, S., Porras, P., Yegneswaran, V., Fong, M., Gu, G., Tyson, M., Texas, A., Station, C. and Park, M. (2013). *Fresco: Modular composable security services for software-defined networks,* Annual Network & Distributed System Security Symposium, San Diego, USA.

70. Jafarian, J. H., Al-Shaer, E. and Duan, Q. (2015). An Effective Address Mutation Approach for Disrupting Reconnaissance Attacks. *IEEE Transactions on Information Forensics and Security*, 10(12), 2562–2577.

71. Jafarian, J.H, Al-Shaer, E. and Duan, Q. (2012). *OpenFlow Random Host Mutation: Transparent Moving Target Defense using Software Defined Networking* ACM Workshop on Hot Topics in Software Defined Networks, Helsinki Finland.

72. Achleitner, S., Porta, T. La, McDaniel, P., Sugrim, S., Krishnamurthy, S. V. and Chadha, R. (2016). *Cyber Deception: Virtual Networks to Defend Insider Reconnaissance*. ACM CCS International Workshop on Managing Insider Security Threats, Vienna Austria.

73. Chiang, C. Y. J., Gottlieb, Y. M., Sugrim, S. J., Chadha, R., Serban, C., Poylisher, A., Marvel, L. M. and Santos, J. (2016). *ACyDS: An adaptive cyber deception system*. IEEE Military Communications Conference, Baltimore Maryland, USA.

74. Robertson, S., Alexander, S., Micallef, J., Pucci, J., Tanis, J. and Macera, A. (2015). *CINDAM: Customized information networks for deception and attack mitigation*. IEEE International Conference on Self-Adaptive and Self-Organizing Systems Workshops, Cambridge, MA, USA.

75. Wang, J., Wen, R., Li, J., Yan, F., Zhao, B. and Yu, F. (2018). Detecting and Mitigating Target Link-Flooding Attacks Using SDN. *IEEE Transactions on Dependable and Secure Computing*, 16(6), 944–956.

76. Kang, M. S., Lee, S. B. and Gligor, V. D. (2013). *The crossfire attack*. IEEE Symposium on Security and Privacy, San Francisco, USA.

77. Aydeger, A., Saputro, N., Akkaya, K. and Rahman, M. (2016). *Mitigating Crossfire Attacks Using SDN-Based Moving Target Defense*. Conference on Local Computer Networks, Dubai, UAE.

78. Allybokus, Z., Perrot, N., Leguay, J., Maggi, L. and Gourdin, E. (2018). Virtual Function Placement for Service Chaining with Partial Orders and Anti-Affinity Rules. *Networks*, 71(2), 97–106.

79. Ma, W., Jonathan, B., Pan, Z., Pan, D. and Pissinou, N. (2017). SDN-Based Traffic Aware Placement of NFV Middleboxes Wenrui. *IEEE Transactions nn Network and Service Management*, 14(3), 528–542.

80. Li, X. and Qian, C. (2015). *The virtual network function placement problem*. IEEE International Conference on Computer Communications, Atlanta, USA.

81. Li, X. and Qian, C. (2016). *A survey of network function placement*. IEEE Annual Consumer Communications and Networking Conference, Las Vegas, USA.

82. Bari, M. F., Chowdhury, S. R., Ahmed, R. and Boutaba, R. (2015). *On orchestrating virtual network functions*. International Conference on Network and Service Management, Barcelona, Spain.

83. Addis, B., Belabed, D., Bouet, M. and Secci, S. (2015). *Virtual network functions placement and routing optimization*. IEEE 4th International Conference on Cloud Networking, Niagara Falls, USA.

84. Luizelli, M. C., Bays, L. R., Buriol, L. S., Barcellos, M. P. and Gaspary, L. P. (2015). *Piecing together the NFV provisioning puzzle: Efficient placement and chaining of virtual network functions*. IFIP/IEEE International Symposium on Integrated Network Management, Ottawa, Canada.

85. Rawat, D. B. and Reddy, S. R. (2017). Software Defined Networking Architecture, Security and Energy Efficiency: A Survey. *IEEE Communications Surveys & Tutorials*, 19, 325–346.

86. Bradley, S. P., Hax, A. C. and Magnanti, T. L. (1977). Integer Programming. *Applied Mathematical Programming*, 272–319.

87. İnternet: Home | Internet2. URL: https://www.internet2.edu/, Son Erişim Tarihi: 18.01.2019.

88. İnternet: GÉANT Network. URL: https://geant3.archive.geant.org/Network/pages/home.aspx, Son Erişim Tarihi: 18.01.2019.

89. Ben Jemaa, F., Pujolle, G. and Pariente, M. (2016). *QoS-aware VNF placement optimization in edge-central carrier cloud architecture*. IEEE Global Communications Conference, Washington, DC, USA.

90. Machuca, C. M. (2006). *Expenditures study for network operators*. International Conference on Transparent Optical Networks, Nottingham.

91. Omnes, N., Bouillon, M., Fromentoux, G. and Le Grand, O. (2015). *A programmable and virtualized network & IT infrastructure for the internet of things: How can NFV & SDN help for facing the upcoming challenges*. 18th International Conference on Intelligence in Next Generation Networks, Paris, France.

92. İnternet: JGAP download | SourceForge.net. URL: https://sourceforge.net/projects/jgap/, Son Erişim Tarihi: 31.01.2019.

93. İnternet: Build the future of Open Infrastructure. URL: https://www.openstack.org/, Son Erişim Tarihi: 18.01.2019.

94. İnternet: Home - OpenDaylight. URL: https://www.opendaylight.org/, Son Erişim Tarihi: 31.01.2019.

95. İnternet: GLPK - GNU Project - Free Software Foundation (FSF). URL: https://www.gnu.org/software/glpk/, Son Erişim Tarihi: 18.01.2019.

96. İnternet: JUNG - Java Universal Network/Graph Framework. URL: http://jung.sourceforge.net/, Son Erişim Tarihi: 31.01.2019.

97. İnternet: Gurobi Optimization - The State-of-the-Art Mathematical Programming Solver. URL: http://www.gurobi.com/, Son Erişim Tarihi: 18.01.2019.

98. İnternet: Welcome to Python.org. URL: https://www.python.org/, Son Erişim Tarihi: 31.01.2019.

99. İnternet: The GARR Network. URL: https://www.thegarrnetwork.org/, Son Erişim Tarihi: 31.01.2019.

100. Qu, L., Assi, C. and Shaban, K. (2016). *Network function virtualization scheduling with transmission delay optimization*. IEEE/IFIP Network Operations and Management Symposium, İsatanbul, Turkey.

101. İnternet: The Topology of CERNET. URL: http://www.edu.cn/english/cernet/structure/200603/t20060323_4298.shtml, Son Erişim Tarihi: 13.09.2019.

102. İnternet: Modeling Topology of Internetworks Home Page. URL: https://www.cc.gatech.edu/projects/gtitm/, Son Erişim Tarihi: 13.09.2019.

103. Qazi, Z. A., Tu, C.-C., Chiang, L., Miao, R., Sekar, V. and Yu, M. (2013). SIMPLE-fying middlebox policy enforcement using SDN. *ACM SIGCOMM Computer Communication Review*, 43(4), 27–38.

104. Yang, K., Zhang, H. and Hong, P. (2016). *Energy-aware service function placement for service function chaining in data centers*. IEEE Global Communications Conference, Washington, DC, USA.

105. Xu, Z., Zhang, X., Yu, S. and Zhang, J. (2018). *Energy-Efficient Virtual Network Function Placement in Telecom Networks*. IEEE International Conference on Communications, Kansas City, USA.

106. Li, D., Shang, Y. and Chen, C. (2014). *Software defined green data center network with exclusive routing*. Proceedings - IEEE International Conference on Computer Communications, Toronto, Canada.

107. Gao, P. X., Curtis, A. R., Wong, B. and Keshav, S. (2012). *It's not easy being green*. ACM SIGCOMM Conference Applications, Technologies, Architectures, and Protocols for Computer Communication, Helsinki, Finland.

108. Khoury, N. El, Ayoubi, S. and Assi, C. (2016). *Energy-Aware Placement and Scheduling of Network Traffic Flows with Deadlines on Virtual Network Functions*. IEEE International Conference on Cloud Networking, Luxembourg.

109. Carpio, F., Dhahri, S. and Jukan, A. (2017). *VNF placement with replication for Loac balancing in NFV networks*. IEEE International Conference on Communications, Paris, France.

110. İnternet: United States Data Center Energy Usage Report | Energy Technologies Area. URL: https://eta-publications.lbl.gov/sites/default/files/lbnl-1005775_v2.pdf, Son Erişim Tarihi: 13.09.2019.

111. Qi, D., Shen, S. and Wang, G. (2019). Towards an efficient VNF placement in network function virtualization. *Computer Communications*, 138, 81–89.

112. Pham, C., Tran, N. H., Ren, S., Saad, W. and Hong, C. S. (2017). Traffic-aware and Energy-efficient vNF Placement for Service Chaining: Joint Sampling and Matching Approach. *IEEE Transactions on Services Computing*, *1374*, 1–14.

113. Bari, M. F., Chowdhury, S. R., Ahmed, R., Boutaba, R. and Duarte, O. C. M. B. (2016). Orchestrating Virtualized Network Functions. *IEEE Transactions on Network and Service Management*, 13(4), 725–739.

114. İnternet: SNDLib. URL: http://sndlib.zib.de/home.action, Son Erişim Tarihi: 13.09.2019.

115. Martins, J., Ahmed, M., Raiciu, C., Olteanu, V., Honda, M., Bifulco, R., Huici, F. and Nsdi, I. (2014). *ClickOS and the art of network function Virtualization.* Symposium on Networked Systems Design and Implementation, Seattle, USA.

116. İnternet: Corporate network security levels. URL: https://www.scnsoft.com/blog/3-levels-corporate-network-security, Son Erişim Tarihi: 18.12.2019.

117. İnternet: CPLEX Optimizer | IBM. URL: https://www.ibm.com/analytics/cplex-optimizer, Son Erişim Tarihi: 18.01.2019.

118. Obraczka, K. and Silva, F. (2002). *Network latency metrics for server proximity*. IEEE Global Telecommunications Conference, Taipei, Taiwan.

119. Kim, S., Han, Y. and Park, S. (2016). *An energy-Aware service function chaining & reconfiguration algorithm in NFV*. IEEE 1st International Workshops on Foundations and Applications of Self-Systems, Augsburg, Germany.

120. Hirwe, A. and Kataoka, K. (2016). LightChain: *A lightweight optimisation of VNF placement for service chaining in NFV*. IEEE Conference and Workshops: Software-Defined Infrastructure for Networks, Clouds, IoT and Services, Seoul, Korea.

121. Sahhaf, S., Tavernier, W., Rost, M., Schmid, S., Colle, D., Pickavet, M. and Demeester, P. (2015). Network service chaining with optimized network function embedding supporting service decompositions. *Computer Networks*, 93, 492–505.

122. İnternet: Mininet: An Instant Virtual Network on your Laptop (or other PC) - Mininet. URL: http://mininet.org/, Son Erişim Tarihi: 18.01.2019.

123. İnternet: POX Controller Tutorial | SDN Hub. URL: http://sdnhub.org/tutorials/pox/, Son Erişim Tarihi: 21.06.2019.

124. Dahmouni, H., Girard, A. and Sanso, B. (2012). An analytical model for jitter in IP networks. *Annals of Telecommunications-Annales des Telecommunications*, 67(1-2), 81–90.

125. Amiri, M., Al Osman, H., Shirmohammadi, S. and Abdallah, M. (2015). *An SDN controller for delay and jitter reduction in cloud gaming*. ACM international conference on Multimedia, 1043-1046, Brisbane Australia.

126. Steed, A. and Oliveira, M. F. (2010). *Networked Graphics Building Networked Games and Virtual Environments*, Elsevier.

127. Yang, M., Li, Y., Jin, D., Zeng, L., Wu, X. and Vasilakos, A. V. (2015). Software-Defined and Virtualized Future Mobile and Wireless Networks: A Survey. *Mobile Networks and Applications*, 20(1), 4–18.

ÖZGEÇMİŞ

Kişisel Bilgiler

Soyadı, adı	: DEMİRCİ, Merve Sedef
Uyruğu	: T.C.
Doğum tarihi ve yeri	: 23.06.1991, Erzincan
Medeni hali	: Evli
Telefon	: 0 (312) 582 31 41
e-mail	: sedefgunduz@gazi.edu.tr

Eğitim

Derece	Eğitim Birimi	Mezuniyet Tarihi
Doktora	Gazi Üniversitesi / Bilgisayar Mühendisliği	Devam Ediyor
Yüksek lisans	Gazi Üniversitesi / Bilgisayar Mühendisliği	2014
Lisans	Fatih Üniversitesi / Bilgisayar Mühendisliği	2012

İş Deneyimi

Yıl	Yer	Görev
2012-Halen	Gazi Üniversitesi	Araştırma Görevlisi

Yabancı Dil

İngilizce

Yayınlar

1. Demirci, S., Sagiroglu, Ş. (2020). Energy-Efficient Placement of Virtual Security Functions on SDN Infrastructure. (in preparation for submission to an SCI-E journal)

2. Demirci, S., Sagiroglu, Ş., Demirci, M. (2019). Design and Analysis of Algorithms for Energy-Efficient Virtual Security Function Placement. *IEEE/ACM Transactions on Networking*, (Under review)

3. Demirci, S. Yürekten, Ö., Demirci, M. (2019). Yazılım Tanımlı Ağlar ve Siber Güvenlik, *Siber Güvenlik ve Savunma: Standartlar ve Uygulamalar*. (Birinci Baskı) Ankara: Grafiker, 121-152.

4. Demirci, S., Sagiroglu, Ş. (2019). Optimal placement of virtual network functions in software defined networks: A survey. *Journal of Network and Computer Applications*, 147(1), 1-20.

5. Demirci, S., Demirci, M., Sagiroglu, Ş. (2019). Virtual Security Functions and Their Placement in Software Defined Networks: A Survey. *Gazi University Journal of Science*, 32(3), 833-851.

6. Demirci, S., Sagiroglu, Ş. (2018). *Software-Defined Networking for Improving Security in Smart Grid Systems.* IEEE 7th International Conference on Renewable Energy Research and Applications, Paris, France.

7. Demirci, S., Demirci, M., Sagiroglu, Ş. (2018). *Optimal Placement of Virtual Security Functions to Minimize Energy Consumption.* IEEE International Symposium on Networks, Computers and Communications, Rome, Italy.

8. Tepe, N. A., Demirci, S. (2017). *Investigating sentimental relation between stress and heart disease mortality.* IEEE International Conference on Computer Science and Engineering, Antalya, Turkey.

9. Özdemir, S. Ö., Demirci, S. (2017). *An intelligent system for predicting location from text content on social media*. IEEE International Conference on Computer Science and Engineering, Antalya, Turkey.

10. Demirci, S., Sağıroğlu, Ş. (2017). Sosyal Ağ Verilerinin Kullanım Alanları Üzerine Kapsamlı Bir İnceleme. *Gazi Üniversitesi Fen Bilimleri Dergisi Part C: Tasarım ve Teknoloji*, 5(2), 1-21.

11. Arslan, B., Gunduz, S., Sagiroglu, S. (2016). *A review on mobile threats and machine learning based detection approaches*. IEEE International Symposium on Digital Forensic and Security, Little Rock, AR, USA.

12. Sağıroglu, S., Terzi, D.S., Terzi, R., Canbay Y., Gündüz S., Arslan B., Ayaydın A., ve Gökalp A.B (2016). *Büyük Veri Analitiği Güvenliği Ve Mahremiyeti* (Birinci Baskı) Türkiye: Büyük Veri Analitiği Ve Güvenliği (BIDISEC) Araştırma Gurubu.

13. Erkal, Y., Sezgin, M., Gunduz, S. (2015). *A New Cyber Security Alert System for Twitter*. IEEE International Conference on Machine Learning and Applications, Miami, Florida, USA.

14. Gunduz, S., Arslan, B., Demirci, M. (2015). *A Review of Machine Learning Solutions to Denial-of-Services Attacks in Wireless Sensor Networks*, IEEE International Conference on Machine Learning and Applications, Miami, Florida, USA.

15. Bal, M., Ozturk, B., Yayla, E. Gunduz, S., Demirci, M. (2015). *A Review of Techniques and Tools for Gathering Digital Evidence from Mobile Phones.* IEEE 3rd International Symposium on Digital Forensics and Security, Ankara, Turkey.

16. Aydogdu, D., Gunduz, S., Sagiroglu, S. (2015). *Web Uygulama Güvenliği Açıklıkları ve Güvenlik Çözümleri Üzerine Bir Araştırma,* IEEE 3rd International Symposium on Digital Forensics and Security, Ankara, Turkey.

17. Arslan, B., Gunduz, S. Sagiroglu, S. (2015). *Güncel Mobil Tehditler ve Alınması Gereken Önlemler*, IEEE 3rd International Symposium on Digital Forensics and Security, Ankara, Turkey.

18. Gunduz, S. Demirhan, F., Sagiroglu, S. (2014). *Investigating sentimental relation between social media presence and academic success of Turkish universities,* IEEE International Conference on Machine Learning and Applications, Miami, Florida, USA.

19. Gunduz, S., Demirhan, F. (2014). *Analyzing the effects of personality traits on Facebook use for education,* International Conference on Infocomm Technologies in Competitive Strategies, Singapore.

20. Gunduz, S., Yavanoglu, U., Sagiroglu, S. (2013). *Predicting next location of Twitter users for surveillance,* IEEE International Conference on Machine Learning and Applications, Miami, Florida, USA.

21. Yavanoglu, U., Gunduz, S., Sagiroglu, S. (2013). *A Novel Approach on Extracting Forensic Information from Location Based Social Networks,* IEEE International Symposium on Digital Forensics and Security, Elazığ, Turkey.

Patent Başvuruları

1. Gunduz, S., Sagiroglu, S. *Türkçe Metin Tabanlı Sosyal Ağ Verilerini Kullanıcı Tanımlı Kategorilere Ve İçerik Türüne Göre Sınıflandıran Zeki Sistem*

2. Gunduz, S., Sagiroglu. *Sosyal Ağ Verilerini Kullanarak Adli Takibe Yardımcı Konum Tahmini Ve Tespiti Yapan Sistem*

Hobiler

Yüzme, Seyahat etme, Sudoku

GAZİ GELECEKTİR...