T.C.
İSTANBUL ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

DOKTORA TEZİ

YAZILIM TABANLI SAĞLAYICIDAN BAĞIMSIZ ERİŞİM NOKTASI KONTROLÜ GELİŞTİRİLMESİ

Mehmet Ali ERTÜRK

Bilgisayar Mühendisliği Anabilim Dalı

Bilgisayar Mühendisliği Programı

DANIŞMAN
Yrd. Doç. Dr. Muhammed Ali AYDIN

II. DANIŞMAN
Doç. Dr. Luca VOLLERO

Aralık, 2016

İSTANBUL

Bu çalışma 12.12.2016 tarihinde aşağıdaki jüri tarafından Bilgisayar Mühendisliği Anabilim Dalı, Bilgisayar Mühendisliği Programında Doktora tezi olarak kabul edilmiştir.

Tez Jürisi

Yrd. Doç. Dr. Muhammed Ali AYDIN(Danışman)
İstanbul Üniversitesi
Mühendislik Fakültesi

Prof. Dr. Ahmet SERTBAŞ
İstanbul Üniversitesi
Mühendislik Fakültesi

Prof. Dr. A.Halim ZAİM
İstanbul Ticaret Üniversitesi
Mühendislik Fakültesi

Prof. Dr. Selim AKYOKUŞ
Doğuş Üniversitesi
Mühendislik Fakültesi

Doç. Dr. Fırat KAÇAR
İstanbul Üniversitesi
Mühendislik Fakültesi

20.04.2016 tarihli resmi gazetede yayımlanan Lisansüstü Eğitim ve Öğretim Yönetmeliğinin 9/2 ve 22/2 maddeleri gereğince; Bu Lisansüstü teze, İstanbul Üniversitesi'nin abonesi olduğu intihal yazılım programı kullanılarak Fen Bilimleri Enstitüsü'nün belirlemiş olduğu ölçütlere uygun rapor alınmıştır.

Bu çalışma İstanbul Üniversitesi Bilimsel Araştırma Projeleri Yürütücü Sekreterliğinin 35205 numaralı projesi ile desteklenmiştir.

ÖNSÖZ

Tez çalışmalarım boyunca göstermiş olduğu her türlü yardım ve destekten dolayı çok değerli hocam Muhammed Ali AYDIN'a en içten dileklerimle teşekkür ederim. İtalya ziyaretlerimde ve sonrasında sağladığı katılarından dolayı ikinci danışmanım Luca VOLLERO'ya ve tez izleme komitemde olan ve fikirlerini esirgemeyen Prof.Dr. A. Halim ZAİM'e ve Prof.Dr. Ahmet SERBAŞ'a teşekkürlerimi sunarım.

Tez çalışmalarıma destek sağlayan İstanbul Üniversitesi Bilimsel Araştırma Projeleri Birimi'ne teşekkürü borç bilirim.

Dualarını hiçbir zaman eksik etmeyen anneme ve babama, desteğiyle her zaman yanımda olan eşime sonuz teşekkür ederim.

Aralık 2016 Mehmet Ali ERTÜRK

İÇİNDEKİLER

Sayfa No

ÖNSÖZ ..... **i**
İÇİNDEKİLER ..... **ii**
ŞEKİL LİSTESİ ..... **iv**
TABLO LİSTESİ ..... **viii**
SİMGE VE KISALTMA LİSTESİ ..... **ix**
ÖZET ..... **xii**
SUMMARY ..... **xiv**
1. GİRİŞ **1**
2. GENEL KISIMLAR **4**
2.1. IEEE 802.11 KABLOSUZ AĞLAR 4
2.1.1. DCF - Dağıtık Koordinasyon Fonksiyonu 5
2.1.2. HCF - Hibrit Koordinasyon Fonksiyonu (EDCA) 7
2.1.3. İstemcilerin AP ile Bağlantısı 8
2.1.4. MAC Çerçeve Formatları (Frame Format) 8
2.2. İLGİLİ ÇALIŞMALAR 10
2.2.1. Ağ Yönetim Protokolleri 10
2.2.2. SDN Tabanlı Yaklaşımlar 12
2.2.3. CAPWAP ile İlgili Çalışmalar 17
2.2.4. Yük Dengeleme ile İlgili Çalışmalar 18
3. MALZEME VE YÖNTEM **23**
3.1. CAPWAP PROTOKOLÜ 23
3.2. ERİŞİM NOKTASI YÖNETİCİSİ MİMARİSİ 33
3.2.1. AC Uygulaması 35
3.2.2. WTP Uygulaması 37
3.3. ÖNERİLEN IEEE 802.11 YÜK DENGELEME MODELİ 45
3.3.1. QoS ve Yük Dengeleme Problemi 46
3.3.2. Ağ Doygunluğunda (Saturation) Throughput Modellenmesi 47
3.3.3. Optimum EDCA Yapılandırması 49

3.4. ÖNERİLEN IEEE 802.11 YÜK DENGELEME ALGORİTMALARI 50
3.4.1. Önerilen Brute-Force Algoritması 51
3.4.2. Önerilen Genetik Algoritma (GA) 53
3.4.3. Önerilen Branch & Bound (B&B) Algoritması 57
4. BULGULAR 63
4.1. BRUTE-FORCE ALGORİTMASI 63
4.2. GENETİK ALGORİTMA 68
4.3. BRANCH & BOUND ALGORİTMASI 72
4.4. TESTBED ORTAMININ DOĞRULANMASI 80
4.5. B&B ALGORİTMASININ TESTBED İLE GERÇEKLENMESİ 88
5. TARTIŞMA VE SONUÇ 92
KAYNAKLAR 95
ÖZGEÇMİŞ 100

ŞEKİL LİSTESİ

Sayfa No

Şekil 2.1: MAC mimari. 5

Şekil 2.2: DCF akış diyagramı. 6

Şekil 2.3: MAC çerçevesi. 10

Şekil 3.1: CAPWAP protokol mesaj değişimi. 26

Şekil 3.2: CAPWAP durum makinesi. 28

Şekil 3.3: CAPWAP kontrol paketi açık metin. 30

Şekil 3.4: CAPWAP kontrol paketi şifrelenmiş. 31

Şekil 3.5: CAPWAP veri paketi açık metin. 31

Şekil 3.6: CAPWAP veri paketi şifrelenmiş. 31

Şekil 3.7: CAPWAP başlığı. 32

Şekil 3.8: CAPWAP kontrol mesaj başlığı. 32

Şekil 3.9: Kurumsal ağ topolojisi. 35

Şekil 3.10: Erişin Noktası Yöneticisi mimari yapısı 36

Şekil 3.11: WTP uygulama mimarisi. 37

Şekil 3.12: Keşif istek ve cevap sınıf yapısı. 39

Şekil 3.13: Ağa dahil olma istek ve cevap sınıf yapısı. 40

Şekil 3.14: Yapılandırma durum sınıf yapısı. 41

Şekil 3.15: Yapılandırma durum güncelleme sınıf yapısı. 42

Şekil 3.16: Durum değişikliği sınıf yapısı. 42

Şekil 3.17: Yapılandırma sıfırlama sınıf yapısı. 43

Şekil 3.18: İstemci yapılandırma sınıf yapısı. 43

Şekil 3.19: Wireshark ile CAPWAP analizi. 44

Şekil 3.20: AP'ler arası paket iletim güç seviyesi arama ağacı. 51

Şekil 3.21: Brute-Force algoritması akış şeması. 53

Şekil 3.22: AP güç seviyesi çaprazlama örneği. 55

Şekil 3.23: Genetik Algoritma akış şeması. 57

Şekil 3.24: BB algoritması akış şeması. 60

Şekil 4.1: Brute-Force simülasyonu AP-STA dağılım alanı. 64

Şekil 4.2: Brute-Force simülasyon güç veri iletim hızları. 65

Şekil 4.3: Senaryo 1 için ortalama veri transfer hızı. 66

Şekil 4.4: Senaryo 2 için ortalama veri transfer hızı. 66

Şekil 4.5: Senaryo 3 için ortalama veri transfer hızı. 66

Şekil 4.6: Senaryo 1 için min. ağırlıklı veri transfer hızı. 67

Şekil 4.7: Senaryo 2 için min. ağırlıklı veri transfer hızı. 67

Şekil 4.8: Senaryo 3 için min. ağırlıklı veri transfer hızı. 67

Şekil 4.9: GA Senaryo 1 kullanıcı dağılımı. 69

Şekil 4.10: GA Senaryo 2 kullanıcı dağılımı. 69

Şekil 4.11: GA Senaryo 1 toplam throughput. 70

Şekil 4.12: GA Senaryo 2 toplam throughput. 70

Şekil 4.13: GA Senaryo 1 min. ağırlıklı throughput. 71

Şekil 4.14: GA Senaryo 2 min. ağırlıklı throughput. 71

Şekil 4.15: GA Senaryo 1 çalışma zamanı. 71

Şekil 4.16: GA Senaryo 2 çalışma zamanı. 72

Şekil 4.17: BB AP dizilim senaryosu doğrusal. 73

Şekil 4.18: BB AP dizilim senaryosu ızgara. 73

Şekil 4.19: BB AP dizilim senaryosu yedigen. 74

Şekil 4.20: Sinyal kayıp seviyeleri. 74

Şekil 4.21: BB Senaryo 1 min. ağırlıklı throughput. 75

Şekil 4.22: BB Senaryo 2 min. ağırlıklı throughput. 75

Şekil 4.23: BB Senaryo 3 min. ağırlıklı throughput (7 AP). 76
Şekil 4.24: BB Senaryo 1 toplam throughput. 76
Şekil 4.25: BB Senaryo 2 toplam throughput. 76
Şekil 4.26: BB Senaryo 3 toplam throughput (7 AP). 77
Şekil 4.27: BB Senaryo 1 ve 2 algoritma çalışma zamanı. 77
Şekil 4.28: BB Senaryo 3 algoritma çalışma zamanı (7 AP). 77
Şekil 4.29: BB ve GA karşılaştırması min. ağırlık throughput 79
Şekil 4.30: BB ve GA karşılaştırması toplam throughput 79
Şekil 4.31: BB ve GA karşılaştırması çalışma zamanı 79
Şekil 4.32: TestBed ortamı hazırlık çalışmaları. 80
Şekil 4.33: TestBed ortamı Carambola2. 81
Şekil 4.34: TestBed ortamı WrtNode. 82
Şekil 4.35: TestBed ortamı WrtNode2R. 82
Şekil 4.36: TestBed ortamı Raspberry Pi 3. 83
Şekil 4.37: TestBed/EDCA simülasyon karşılaştırma ortamı. 84
Şekil 4.38: TestBed ortamı, 1 AP - 1 STA, 1 Mbps throughput doğrulama. 85
Şekil 4.39: TestBed ortamı, 1 AP - 1 STA, 2 Mbps throughput doğrulama. 86
Şekil 4.40: TestBed ortamı, 1 AP - 1 STA, 5.5 Mbps throughput doğrulama. 86
Şekil 4.41: TestBed ortamı, 1 AP - 1 STA, 11 Mbps throughput doğrulama. 86
Şekil 4.42: TestBed ortamı, 1 AP - 2 STA,1 Mbps throughput doğrulama. 87
Şekil 4.43: TestBed ortamı, 1 AP - 2 STA, 2 Mbps throughput doğrulama. 87
Şekil 4.44: TestBed ortamı, 1 AP - 2 STA, 5.5 Mbps throughput doğrulama. 87
Şekil 4.45: TestBed ortamı, 1 AP - 2 STA, 11 Mbps throughput doğrulama. 87
Şekil 4.46: BB algoritmasının TestBed ortamında gerçekleştirme senaryosu. 88
Şekil 4.47: TestBed ortamında min. ağırlık throughput. 89
Şekil 4.48: TestBed ortamında toplam throughput. 89
Şekil 4.49: TestBed ortamı standart RSSI ile AP-STA bağlantısı. 90

Şekil 4.50: TestBed ortamı BB ile AP-STA bağlantısı. 90

TABLO LİSTESİ

Sayfa No

Tablo 2.1: Erişim kategorileri ve kullanıcı önceliği. 7

Tablo 2.2: MAC kontrol çerçevesi tür ve alt türleri. 9

Tablo 3.1: CAPWAP kontrol mesajları. 33

Tablo 3.2: Keşif istek ve cevap mesaj elemanları. 38

Tablo 3.3: Katılma istek ve cevap mesaj elemanları. 39

Tablo 3.4: Yapılandırma durum istek ve cevap mesaj elemanları 41

Tablo 3.5: IEEE 802.11 mesaj listesi. 44

Tablo 3.6: EDCA parametreleri. 47

Tablo 3.7: Brute-Force algoritması. 52

Tablo 3.8: Genetik algoritma. 56

Tablo 3.9: Branch & Bound algoritması. 61

Tablo 4.1: Model ve simülasyon parametreleri. 63

Tablo 4.2: İstemcin senaryolara göre dağılımı. 65

SİMGE VE KISALTMA LİSTESİ

Simgeler **Açıklama**

n_{AP} : Toplam AP sayısı
n_{STA} : Toplam STA sayısı
X : Bir alandaki aktif istemciler
Y : Bir alandaki aktif AP'ler
A : Bir istemcinin AP ile olan bağlantısı
V : Tüm bağlantılar
E : CB ile değiştirilebilen AP istemci bağlantısı
a_k : k nolu AP
s_i : i nolu istemci
w : Ağırlık
σ : Dilim zamanı
τ : Paket iletim olaslığı
L : Paket boyutu
r : İstemci veri iletim hızı
t_{ap} : AP througput

Kısaltmalar **Açıklama**

AC : Access Controller
AP : Access Point
API : Application Programming Interface
BB : Branch & Bound
BE : Best Efford
BK : Background
BSS : Basic Service Set
BT : Backoff Timer
CAPWAP : Control And Provisioning of Wireless Access Points

CB : Cell Breathing
CMIP : Common Management Information Protocol
CW : Contention Windows
DCF : Distributed Coordination Function
DIFS : DCF Interframe Space
DS : Distributed System
DTLS : Datagram Transport Layer Security
EDCA : Enhanced Distributed Channel Access
EIFS : Extended Interframe Space
ESS : Extended Service Set
FCS : Frame Check Sequence
GA : Genetic Algorithm
HCF : Hybrid Coordination Function
IP : Internet Protocol
JSON : JavaScript Object Notation
LTE : Long-Term Evolution
MAC : Medium Access Controller
Mbps : Megabits per Second
NAT : Network Address Translation
PCF : Point Coordination Function
PL : Path Loss
PHY : Phisical
RFS : Request for Comments
RSSI : Received Signal Strength Indicator
QoS : Quality of Service
RMON : Remote Monitoring
SDN : Software Defined Networking
SIFS : Short Interframe Space
SNMP : Simple Network Management Protocol
SNR : Signal to Noise Ratio
SSH : Secure Shell
STA : Station
UDP : User Datagram Protocol
VI : Video

VLAN : Virtual Local Area Network
VO : Voice
VPN : Virtual Private Network
WiFi : Wireless Fidelity
WLAN : Wireless Local Area Network
WTP : Wireless Termination Point
XML : Extensible Markup Language

ÖZET

DOKTORA TEZİ

YAZILIM TABANLI SAĞLAYICIDAN BAĞIMSIZ ERİŞİM NOKTASI KONTROLÜ GELİŞTİRİLMESİ

Mehmet Ali ERTÜRK

İstanbul Üniversitesi

Fen Bilimleri Enstitüsü

Bilgisayar Mühendisliği Anabilim Dalı

Danışman : Yrd. Doç. Dr. Muhammed Ali AYDIN

II. Danışman : Doç. Dr. Luca VOLLERO

Kurulum kolaylığı ve ucuz maliyetleri sebebiyle IEEE 802.11 Kablosuz Ağlar her alanda hızlı bir şekilde yayılmış ve kullanılmaya başlanmıştır. Yaygın olarak kullanılmasıyla birlikte yönetimsel sorunlarda beraberinde gelmektedir. Kullanıcı ve kablosuz erişim noktasının az olduğu ağ altyapılarını yönetmek kolay olmakla birlikte, büyük ölçekli kurumlarda geleneksel metotlar ile bu altyapıları yönetmek zaman ve iş gücü kaybına neden olmaktadır. Aynı zamanda, yapılandırma işlemlerinin elle kontrol edilmesi kullanıcı hatalarına da kapı aralamaktadır. Büyük ölçekli ağ altyapılarına ihtiyaç duyulan ortamlarda her bir kablosuz erişim noktasının merkezi bir noktada yapılandırılması, izlenmesi ve yönetilebilmesi için Erişim Noktası Yöneticisine ihtiyaç duyulmaktadır.

Erişim Noktası Yöneticisi yapılandırma, izleme ve yönetim olmak üzere üç ana fonksiyonu yürütmektedir. Bu faaliyetleri gerçekleştirmek için günümüzde farklı çözümler mevcuttur. Birçok Erişim Noktası üreticisi firma, kendi ekosistemlerinde çalışmak üzere hazırladıkları çözümleri geliştirerek sunmaktadırlar. Bu çözümler firma tarafından geliştirilen özel firmware ile dağıtıldıkları için dışardan müdahalelere kapalıdırlar.

Tez çalışmamızda donanım üreticilerinin sunmuş olduğu kapalı ve kendi donanımları ile çalışan çözümlere alternatif olacak, yazılım tabanlı ve herhangi bir sağlayıcıdan bağımsız standart protokoller ile çalışan bir Erişim Noktası Yöneticisi geliştirilmiştir. Geliştirilen sistem, kablosuz cihazları yönetmek için tasarlanmış olan CAPWAP protokolünü kullanmakta olup, bu protokolün desteklendiği diğer çözümlerle uyumlu çalışabilmektedir.

Çalışmamızda, bir TestBed ortamı kurularak farklı donanım üreticilerinin farklı yonga seti ve mimarilerinde, geliştirdiğimiz uygulamalar çalıştırılarak test süreçleri tamamlanmıştır. Tasarlanan sistemin mimarisi kablosuz ağ altyapıları için geliştirilen algoritmaların test edilmesine olanak sunan bir yapı sunmaktadır.

IEEE 802.11 standartları gereği, kullanıcılar en güçlü sinyali aldıkları erişim noktasına bağlanırlar bunun sonucunda, ağ üzerinde dengesiz yük dağılımına yol açarak, ağ performansını olumsuz etkilemektedir. Literatür incelendiğinde, mevcut yük dengeleme algoritmaları, kullanıcıları, sayıca en az istemcisi olan veya trafik yükü bakımından en az olan erişim noktasına yönlendirerek çözmeye çalışmaktadır. Bu çalışmalar, trafik önceliğini göz ardı ederek QoS ve yük dengeleme yapılandırmasını göz ardı etmektedirler.

Tez çalışmamızda, QoS ve yük dengeleme problemlerini analiz ve çözüm üretmek amacıyla bir çatı çerçeve hazırlanmış ve bu çatı ile farklı yük dengeleme algoritmaları üzerinde çalışmalar yapılmıştır. Modellenen algoritmalar kapalı bir alanda, geliştirdiğimiz merkezi Erişim Noktası Yöneticisi üzerinden TestBed ortamında gerçek cihazlar ile test edilerek sonuçlar doğrulanmıştır. Elde edilen sonuçlar değerlendirildiğinde, önerilen model, QoS ile yük dengeleme algoritmalarını analiz ve test etmek açısından etkili bir yöntem olduğu gözlemlenmiştir. Ayrıca geliştirilen Erişim Noktası Yöneticisi gerçek bir ortamda yük dengeleme algoritmaları ile test edilerek sistemin uçtan uca testleri yapılmıştır. Algoritmaların gerçek kablosuz ağ altyapılarında test edilmesi ve uygulanması açıcından geliştirilen Erişim Noktası Yöneticisi önemli bir araç olabileceği öngörülmektedir.

Aralık 2016, 117 sayfa.

Anahtar kelimeler: Erişim noktası yöneticisi, IEEE 802.11, kablosuz ağlar, CAPWAP, QoS ve yük dengeleme

SUMMARY

Ph.D. THESIS

DEVELOPMENT OF SOFTWARE BASED VENDOR INDEPENDENT ACCESS POINT CONTROLLER

Mehmet Ali ERTÜRK

İstanbul University

Institute of Graduate Studies in Science and Engineering

Department of Computer Engineering

Supervisor : Asst. Prof. Dr. Muhammed Ali AYDIN

Co-Supervisor : Asst. Prof. Dr. Luca VOLLERO

Wireless hotspots have become popular, due to their ability to provide inexpensive and easy-to-deploy network infrastructures. While it is easy to maintain these networks in small infrastructures, it becomes an issue in large scale network infrastructures using tradational startegies. Also, traditional methods are workfoce and error-prone. It is necessary to use a centrialized Acess Controller Systems to handle configuration, monitoring and management functions in large scale wireless infrastructures.

An Access Controller (AC) carries out three main functions, monitoring, configuration and management. There are different solutions are avilable to ensure these functionalities. However, most of the AC producers solves these problems with their own solutions within their hardware ecosystems. These solutions are distributed with vendor depended firmwares which are closed for any external modifications.

In our study, we will provide an alternative software based Access Controller solution built on top of the standards and independed of the any hardware vendor. Developed solution is based on CAPWAP protocol, which is a standard to manage wireless devcies. Our system is capable of any CAPWAP supported device.

In our study, we built a TestBed enviroment with different hardwares from different vendors and architectures to test our system. Also, designed architecture has abilities to suports management algorithms for dynamic configuration strategies to be applied on wireless networks.

According to the IEEE 802.11 standarts Access Point selection in WiFi hotspots is driven by stations and it is based on the measured strongest RSSI which leads network to be imbalanced and decreases performance of the network. Existing Load Balancing (LB) solutions aim at solving this problem by enforcing the connection of stations to the AP having either the smallest number of associated stations or the lowest traffic load. However, LB solutions do not account for traffic priorities or, when they consider them, they do not deal with the joint configuration of QoS (Quality of Service) and LB parameters.

In this study we present a framework for modeling, analyzing and designing QoS-aware LB solutions. The proposed framework is validated through TestBed environment with real devices in a typical indoor LB scenario. The results show that the model is effective in capturing network performance and in designing LB solutions that account for traffic priorities and the configuration of QoS parameters. Also, developed AC is important tool to provide a real enviroment to test and apply algorithms in real wireless infrastructures

December 2016, 117 pages.

Keywords: Access controller, IEEE 802.11, wireless networks, CAPWAP, QoS and load balancing

1. GİRİŞ

AP (Access Point – Erişim Noktası) bir ağ ortamında birden fazla kablosuz istemcinin aynı ağa dâhil olarak bir birine bağlanmasını sağlayan cihazlardır. Birçok akıllı telefon ve tablet gibi cihazların hızlı bir şekilde artmasıyla kablosuz ağlara gereken ihtiyaçta artmaktadır. Ev gibi küçük ortamlar için bir tane WTP (Wireless Termination Point) yeterli iken, okul, alışveriş merkezleri ve hastane gibi çok kullanıcılı ortamlarda çok sayıda WTP kullanılmaktadır. Bu tür büyük ağ altyapısına ihtiyaç duyulan ortamlarda her bir WTP'nin ayarlarının gerçekleştirilmesi, izlenme ve yönetiminin el yordamıyla yapılması zordur. Ortaya çıkacak yönetimsel problemlerin giderilebilmesi için WTP'lerin kontrolünü sağlayan bir sisteme AC (Erişim Noktası Yöneticisi – Access Controller) ihtiyaç duyulmaktadır.

Erişim Noktası Yöneticisinin, yapılandırma, izleme ve yöntemi olmak üzere üç ana fonksiyonu bulunmaktadır. Bu faaliyetleri gerçekleştirme için WTP ile AC arasındaki iletişimi sağlayan farklı çözüm ortaya konulmuştur. Birçok AC üretici firma kendi çözümlerini geliştirerek ilgili WTP cihazlarına yüklemektedir. Bu çözümler firmanın geliştirmiş olduğu firmware ile birlikte sunulmaktadır ve dışarıdan müdahalelere kapalıdır. Buna alternatif olarak birçok açık kaynak kodlu firmware çözümler de mevcuttur.

En yaygın olarak kullanılan çözümler başında; OpenWRT[1], DebWrt[2], DD-Wrt[3], RouterTech[4] ve HyperWRT[5] firmware türevleri gelmektedir. Bu firmwarelerin birçoğu Linksys firmasının WRT54G serisi için ürettiği firmware üzerine geliştirilmiştir. OpenWRT bunların arasında en yaygın olarak kullanılan sürümdür ve yönlendirme işlevinin yanında, mesh ağ yapısı, wireless repeater, wireless bridge, firewall, NAT, Port

[1] OpenWRT, Gömülü Sistemler için Linux İşletim Sistemi, http://openwrt.org, [Ziyaret tarihi: 10 Ekim 2016]
[2] DebWrt, Gömülü Sistemler için Debian İşletim Sistemi, http://debwrt.net, [Ziyaret tarihi: 10 Ekim 2016]
[3] DD-WRT, Gömülü Sistemler için İşletim Sistemi, http://dd-wrt.com, [Ziyaret tarihi: 10 Ekim 2016]
[4] RouterTech, Gömülü Sistemler için İşletim Sistemi, http://routertech.org, [Ziyaret tarihi: 10 Ekim 2016]
[5] HyperWRT, Gömülü Sistemler için İşletim Sistemi, https://sourceforge.net/projects/hyperwrt, [Ziyaret tarihi: 10 Ekim 2016]

Forwarding, UPnP, IPS, Network Scheduler, Traffic Shaping, IP tunneling, Dynamic DNS, DHCP özelliklerini bünyesinde barındırmaktadır. Bu nedenle geliştirilecek bir AC cihaz veya yazılımının OpenWRT'yi desteklemesi sistem kurulumunda daha fazla üreticinin geliştirdiği cihazı kullanabilecek bir yapı sunar.

Ağ altyapılarının yönetimi SNMP, RMON, CMIP, CAPWAP gibi protokoller ile SDN, SHH, JSON ve XML tabanlı yaklaşımlar olmak üzere birçok yöntem mevcuttur [1], [2], [3], [4], [5], [6]. Bu yöntemler incelendiğinde CAPWAP haricindeki protokoller, genel ağ yönetim standart ve teknikleri olup IEEE 802.11 kablosuz ağlar için özel olarak tasarlanmamışlardır.

CAPWAP (Control And Provisioning of Wireless Access Points - RFC5415): Ağ üzerindeki WTP'leri yönetmek amacıyla geliştirilen bir protokoldür. L2 (Layer 2) bağımsız olarak kablosuz AP'lerin kontrol ve provizyonunu sağlamak amacıyla geliştirilmiştir [7]. AC ve WTP'ler arasındaki iletişim IP - Internet Protokolü ile sağlanmaktadır. Protokol Local ve Split Mac olmak üzere iki çalışma modunu desteklemektedir. Split Mac modunda, bütün L2 veri ve yönetim çerçeveleri CAPWAP protkolü ile paketlenerek AC ve WTP arasında iletişim sağlanır. Local Mac modunda, veri çerçeveleri local olarak köprülenebilir veya 802.3 çerçeveleri olarak tünellenebilir. L2 kablosuz yöntemi çerçeveleri WTP tarafından işlenerek AC'ye yönlendirilmektedir.

Mevcut standartlar ve protokoller incelendiğinde, IEEE 802.11 kablosuz ağlar için özelleştirilmiş ortak bir dile ihtiyaç duyulması üzerine, CAPWAP geliştirilmiştir. CAPWAP IEEE 802.11 WiFi ağları için özel olarak tasarlanan tek protokolüdür. Bu protokol sayesinden WTP'ler üzerinde yönetimsel ve izleme faaliyetleri yürütülebilir ve hatta WTP'ler için gerek duyulan yapılandırma fonksiyonelliği sunulabilir. Böylelikle marka ve donanım bağımsız, yazılım tabanlı bir Erişim Noktası yöneticisi geliştirilebilir ve yönetsel faaliyetler gerçekleştirilebilir.

Yapılan çalışmalar, WTP'lerin yönetimsel faaliyetlerini, markalara özgü yöntemler ile gerçekleştirdiği, marka bağımsız standart protokoller ile çalışan bir Erişim Noktası Yöneticisine (Access Controller - AC) ihtiyaç olduğu görülmektedir. Bu tez çalışmasında, CAPWAP protokolü kullanılarak marka ve sağlayıcıdan bağımsız yazılım tabanlı bir Erişim Noktası Yöneticisi geliştirilmiştir. Geliştirilen AC ile TestBed ortamı kurularak,

farklı üreticilerin geliştirdiği farklı yonga seti ve mimariler üzerinde uygulama test edilerek gerçek ortam testleri yapılmıştır. Hazırlamış olduğumuz AC ve TestBed ortamı 802.11 kablosuz ağlarında geliştirilecek olan algoritmaların test edilmesine imkan sunan bir mimariye sahiptir.

Ayrıca IEEE 802.11 standartları gereği istemciler AP seçimi yaparken, en güçlü sinyali aldıkları AP'yi seçerek bağlantı kurmaktadırlar. Bu da ağdaki yükün adil olarak dağılmamasına ve performans problemlerine yol açmaktadır. Günümüze kadar gelen çalışmalar incelendiğinde 802.11 kablosuz ağlar üzerinde yük dengeleme algoritmaları üzerinde çalışmalar yapıldığını fakat bu çalışmalar ya bir AP ile veya QoS'in kapsam dışı bırakılarak çözüldüğü varsayılmaktadır. Bu çalışmamızda, QoS ve yük dengeleme problemlerini analiz ve çözüm üretmek amacıyla bir çatı çerçeve hazırlanmış ve bu çatı ile farklı yük dengeleme algoritmaları üzerinde çalışmalar yapılmıştır. Modellenen algoritmalar, kapalı bir alanda geliştirdiğimiz merkezi Erişim Noktası Yöneticisi üzerinden TestBed ortamında gerçek cihazlar ile test edilerek sonuçlar doğrulanmıştır. Elde edilen sonuçlar değerlendirildiğinde, ortaya atılan model, QoS ile yük dengeleme algoritmalarını analiz ve test etmek açısından etkili bir yöntem olduğu gözükmektedir. Ayrıca geliştirilen Erişim Noktası Yöneticisi gerçek bir ortamda yük dengeleme algoritmaları ile test edilerek sistemin uçtan uca testleri yapılmıştır. Algoritmaların gerçek kablosuz ağ altyapılarında test edilmesi ve uygulanması açıcından geliştirilen Erişim Noktası Yöneticisi önemli bir araç olacağı öngörülmektedir.

Bu çalışmanın 2. Genel Kısımlar Bölümü, kablosuz ağlar ve MAC katmanı hakkında genel bilgiler ile kablosuz ağların merkezi yönetimi ve yük dengeleme algoritmaları üzerine bir literatür özeti içermektedir. Bölüm 3. Malzeme ve Yöntem; AC mimarisini, CAPWAP protokolünün uygulanması ve yük dengeleme algoritmaları hakkında yapılan çalışmaları içermektedir. Elde edilen çıktılar 4. Bölümde Bulgular başlığı altında belirtilmiştir. Sonuçların değerlendirilmesi ve gelecekte yapılacak çalışmalara 5. Bölüm Taraşıma ve Sonuç başlığı altında yer verilmiştir.

2. GENEL KISIMLAR

Kolay kurulum ve düşük maliyetleri sebebiyle kablosuz WiFi ağlar günümüzde oldukça yaygın bir biçimde kullanılmaktadır. İlk olarak 1997 yılında IEEE 802.11 tarafından standartları belirlenen ve halen geliştirilmeye devam eden bir kablosuz iletişim teknolojisidir. 1999 yılında yayınlanan 802.11b, 802.11 ailesi içerisinde en yaygın olanıdır ve sırasıyla 802.11e, 802.11g, 802.11n, 802.11ac izler. IEEE 802.11, MAC (Medium Access Control) ve Fiziksel Katman (PHY) spesifikasyonunun tanımlarını içerir [8]. Bu çalışmada, bir erişim noktası yöneticisinin yönetsel faaliyetlerinin gerçekleştirebilmesi için gerekli olan MAC katmanı standartları üzerinde çalışmalar yapılmıştır. Fiziksel katman bu çalışmanın kapsamı dahilinde tutulmamıştır.

2.1. IEEE 802.11 KABLOSUZ AĞLAR

Kablosuz ağları diğer geleneksel ağlardan ayıran birçok karakteristik özellik bulunmaktadır. En belirgin özellik, istemcilerin kablolu ağlarda olduğu gibi sabit noktada bulunmaması ve mobil olmasıdır. Ayrıca veri iletim ortamı kabloludan farklı olması birçok güçlüğü de beraberinde getirmiştir. Veri iletimi kablolu ağlara göre daha az güvenilirdir, dinamik topolojiler mevcuttur ve sinyal girişimi bu zorlukların başında yer almaktadır.

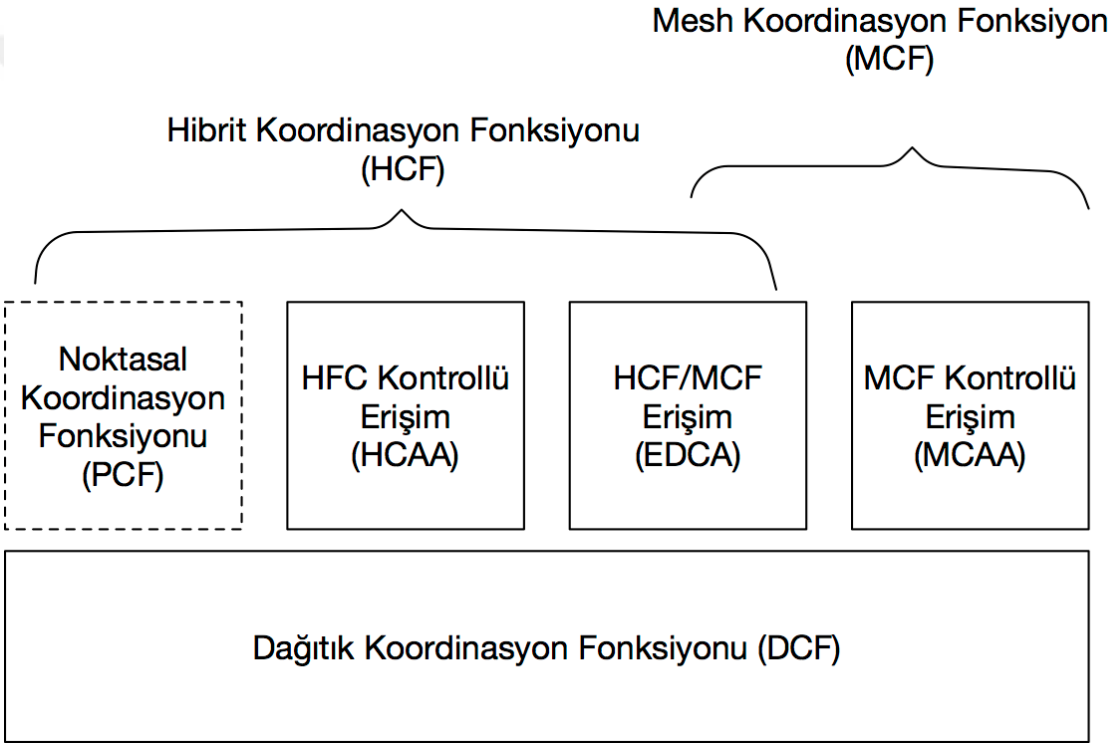
Kablosuz iletişim, birden fazla istemcinin ortak bir kanal paylaşmasının yönetimini gerektirir. Ortak bir kanal paylaşımını yönetmek için izlenecek stratejileri merkezi ve dağıtık olmak üzere iki ana kategoride gruplamamız mümkündür.

- Merkezi iletim kontrol stratejisi; tek bir birimin (genellikle AP) ağ köprüsü veya çıkış kapısı olarak veri transferini zamanlaması ile kanal erişimini yönetir.
- Dağıtık iletim kontrol stratejisi; bu durumda istemciler dağıtık bir algoritma kullanarak eş zamanlı olarak kanala erişir ve oluşabilecek hataları minimize eder.

Kablosuz yerel ağları için geliştirilmiş bir referans standardı olan IEEE 802.11 hem merkezi hem de dağıtık kontrol stratejilerini içerir. Noktasal Koordinasyon Fonksiyonu (The Point Coordination Function – PCF), merkezi bir zamanlama mekanizması olup, bir

istemci veya AP, diğer istemcilerin ileteceği çerçevelerin (frame) zaman planlamasını gerçekleştirir. Diğer yandan Dağıtık Koordinasyon Fonksiyonu (The Distributed Coordination Function - DCF) IEEE 802.11 standardı olup günümüzde birçok dizüstü bilgisayar, mobil telefon ve kablosuz ağ kartı üreticileri tarafından kullanılmaktadır. Ayrıca DCF diğer dağıtık mekanizmaları ile uyumludur, örneğin; İyileştirilmiş Dağıtık Kanal Erişimi (Enhanced Distributed Channel Access - EDCA).

MAC Katmanı mimarisi Şekil 2.1'de özetlenmiştir. Temelinde DCF fonksiyonu bulunan farklı ihtiyaçlar için genişletilen bir kanal erişim mekanizması sunmaktadır [8].

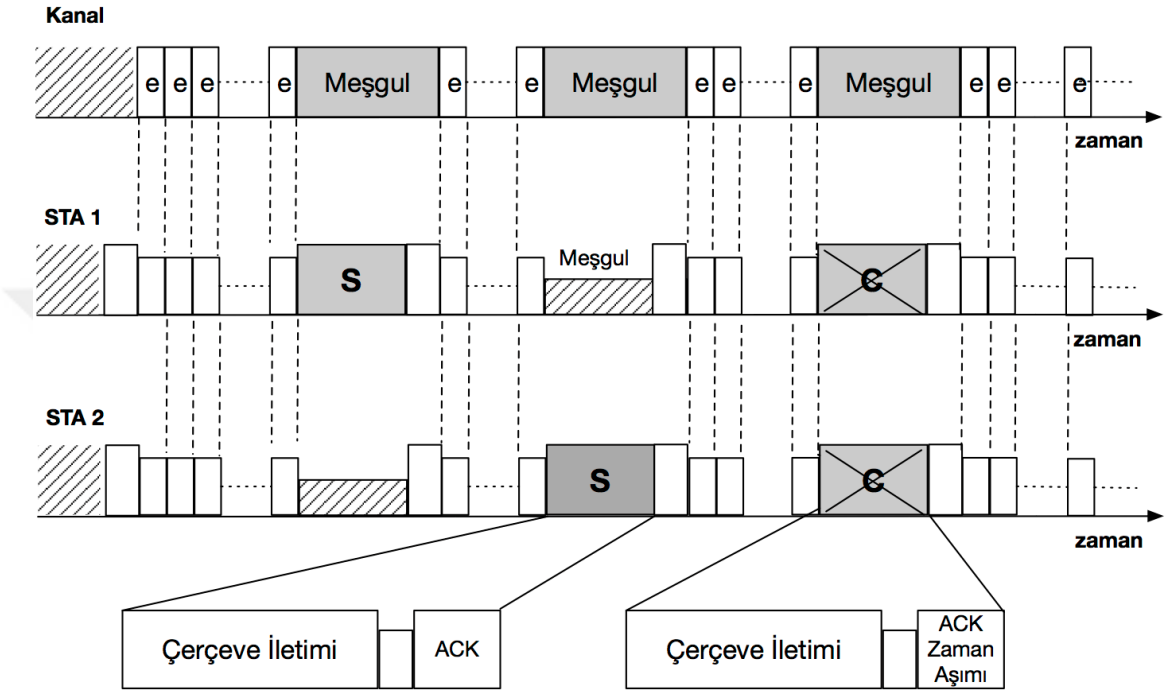


Şekil 2.1: MAC mimari.

2.1.1. DCF - Dağıtık Koordinasyon Fonksiyonu

DCF, istemcilerin kablosuz kanalı adil kullanması için geliştirilmiş bir Ortam Erişim Kontrolü (Medium Access Control – MAC) protokolüdür. DCF algoritmasının ana parametrelerini minimum ve maksimum Contention Windows (CWmin, CWmax), tekrar limiti (Rety Limit – R) oluşturmaktadır. Bu parametreler kablosuz cihaz üreticileri tarafından üretilen cihazlara göre değişiklik gösterebilmektedir. İstemciler arasında adil bir kanal atama işlemi, bu parametrelerin tüm istemcilerde eşit şartlarda yapılandırılmış olmasına bağlıdır. Standartlar tarafından tanımlanan diğer bir parametre ise boş zaman

dilimi (empty slot time duration - *mu*) ve çerçeveler arası zaman dilimleridir; DCF Çerçeveler Arası Boşluk (Interframe Space - DIFS), Kısa Çerçeveler Arası Boşluk (Short Interframe Space - SIFS) ve Genişletilmiş Çerçeveler Arası Boşluk (Extended Interframe Space - EIFS)



Şekil 2.2: DCF akış diyagramı.

Doygunluk (Saturation), bir ağda tüm istemcilerin, paket transfer kuyruklarında tamamen dolu olması durumudur. Bu durum bir ağı analitik olarak modellemek için kullanılan bir yöntemdir. DCF'in çalışma Şekil 2.2'de görüldüğü üzere şu şekilde gerçekleşmektedir:

- Yeni bir paket iletilir iletilmez, başka bir paket iletim için hazırdır ve istemci şu işlemleri gerçekleştirir: (i) *CW* parametresi *CW=CWmin* olacak şekilde atanır, (ii) istemci *{0,1, ..., CW-1}* kümesinden düzgün rastgele bir tamsayı seçer, Geri Sayım Sayacı (Backoff Timer - *BT*) olarak belirler ve (iii) *DIFS* kadar kanalın boş olmasını bekler,
- Kanal *DIFS* süresi kanal boş ise, istemci her *mu* zamanında, *BT* sayacını bir azaltır. Eğer kanal meşgul ise (başka bir istemci transfer yapıyorsa), istemci *BT* sayacını durdurur ve *DIFS* kadar boş zaman oluştuğunda sayaç tekrar başlatılır,
- *BT=0* olduğunda istemci paketi iletir.

- Bir paket iletim işlemi başarılı ya da başarısız olarak sonuçlanabilir (çakışma yada gürültüden dolayı). Eğer iletim başarılı olduysa, istemci DCF algoritmasını kuyruktaki diğer paket için tekrar başlatır.
- Eğer iletim başarısız ise, istemci iletimi tekrar başlatır. İstemci *CW* parametresini, şu koşulda artırır $CW = min\{2CW, CW_{max}\}$ ve $\{0,1, ..., CW-1\}$ olacak şekilde yeni bir BT oluşturur. Algoritma ilk pakette olduğu gibi çalışmaya devam eder.
- Eğer iletim sırasında paketler düşerse ve *DCF* yeniden başlatılırsa, her paket iletimi, *R* limiti kadar tekrar denenir.

2.1.2. HCF - Hibrit Koordinasyon Fonksiyonu (EDCA)

IEEE 802.11e ile birlikte var olan kablosuz ağlara QoS (Quality of Service) standart getirilmiştir. EDCA mekanizması ile istemciler için sekiz farklı kullanıcı önceliği ile dağıtık kanal erişimi sunulmaktadır. EDCA dört erişim kategorisi istemcilerin paket iletimini sağlamaktadır. Tablo 2.1'de belirtilmiş öncelikler gösterilmiştir [8].

Tablo 2.1: Erişim kategorileri ve kullanıcı önceliği.

Öncelik	Kullanıcı Önceliği	Erişim Kategorisi	Açıklama
Düşük	1	AC_BK	Arka plan
	2	AC_BK	Arka plan
	0	AC_BE	En iyi erişim
	3	AC_BE	En iyi erişim
	4	AC_VI	Video
	5	AC_VI	Video
	6	AC_VO	Ses
Yüksek	7	AC_VO	Ses

IEEE 802.11 kablosuz ortam, Yönetenim (Management), Kontrol (Control) ve Veri (Data) çerçeveleri ile yönetilmektedir. Yönetim çerçevelerinin alt türlerinden olan Beacon çerçeveleri belirli aralıklarla AP tarafından yayınlanır ve kablosuz ağ hakkında bilgi içerir.

EDCA, yapılandırılabilir DCF parametrelerini sunmaktadır. Buna göre CW_{min} ve CW_{max} değerleri Beacon çerçeveleri ile yayınlanır ve istemcilere paket iletim önceliği sağlanabilmektedir. Bu yapılandırılabilirlik sayesinde QoS ile yük dengeleme algoritmalarını gerçek bir ortamda, standartlara müdahale etmeden test edebilmek mümkündür.

2.1.3. İstemcilerin AP ile Bağlantısı

Bağlantı (Association) prosedürü bir istemcinin Basit Servis Kümesine (Basit Service Set – BSS) katılmasından ibarettir [8]. Bu işlem istemci tarafından yönetilir ve üç fazda gerçekleşir:

- Birinci fazda; istemci ortamı tarar ve uygun AP'leri Beacon çerçeveleri ile dinler (pasif tarama) veya Probe istekleri (Probe Request) göndererek AP'leri tarar (aktif tarama).
- İkinci aşamada; istemci bulduğu tüm AP'ler arasından en uygun olanı ile bağlantı (association) kurmaya karar verir.
- Son aşamada; istemci bağlantı istekleri (Association Request) ile seçtiği AP'ye bağlanmaya çalışır. İstemci AP seçerken, AP'ler tarafından yayımlanan bilgiler ile bu işleme karar verir. İstemci AP'ye bağlantı isteği gönderdiği an kimlik denetleme (authentication) işlemi başlar.

Başarılı bir kimlik denetlemenin ardından istemci BSS'nin bir parçası olur ve kendisine izin verilen bütün istemciler ile aynı BSS ve AP ile iletişim kurabilir. IEEE 802.11 standartları gereği istemci sadece bir AP ile bağlantı kurabilir ve bu AP'i seçerken, en yüksek RSSI (Received Signal Strength Indicator) değerine göre karar verir. Bu da bir ağ içerisinde dengesiz yük dağılımına sebep olmaktadır. Bu tez çalışmasında bu soruna çözüm olacak yöntemler Bölüm 3.3' de detaylı olarak anlatılmıştır.

2.1.4. MAC Çerçeve Formatları (Frame Format)

MAC katmanında kablosuz ortamda iletişim MAC çerçeveleri ile sağlanmaktadır. Her bir çerçevede olması gereken temel bileşenler:

- Çerçeve kontrolü, süre, adres ve opsiyonel dizi kontrol bilgisi, opsiyonel QoS kontrol bilgisi (QoS veri çerçevelerinde) ve opsiyonel HT kontrol (+HTC çerçevelerinde) alanlarını içeren bir MAC başlığı;
- Çerçevenin türüne ve alt-türüne göre değişken uzunlukta çerçeve gövdesi;
- IEEE 32-bit CRC içeren bir FCS (Frame Check Sequence) çerçeve denetim dizisi içermelidir.

Genel bir MAC çerçevesinin yapısı Şekil 2.3'de gözükmektedir, çerçevenin ilk iki baytlık kısmı çerçeve kontrol alanı olarak belirlenmiştir. Çerçeve kontrol alanı, protokol

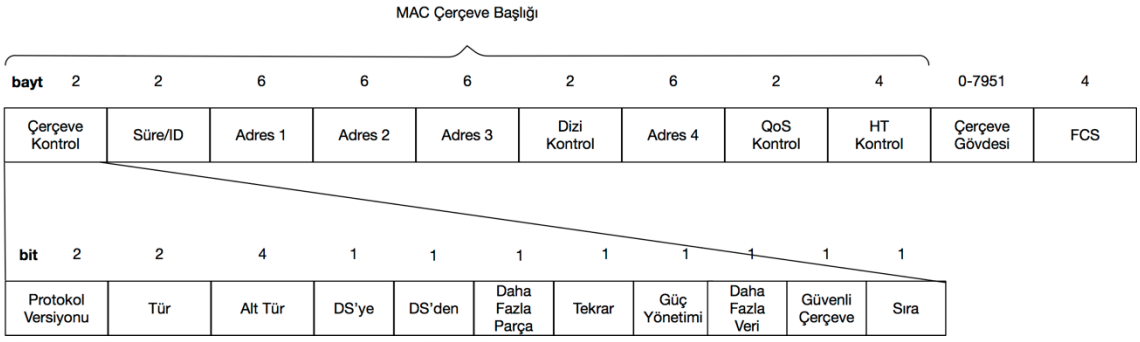
versiyonu, tür, alt tür, DS'ye, DS'den, daha fazla parça, tekrar, güç yönetimi, güvenli çerçeve ve sıra alt alanlarından oluşmaktadır. Alt ve alt tür çerçevenin türünü belirlemektedir ve Yönetenim (Management), Kontrol (Control) ve Veri (Data) olmak üzere üç ana çerçeve türü mevcuttur. Bu çerçeve türleri Tablo 2.2'de listelenmiştir.

Tablo 2.2: MAC kontrol çerçevesi tür ve alt türleri.

Tür Değeri	Tür Açıklaması	Alt Tür Değeri	Alt Tür Açıklaması
00	Yönetim	0111	Rezerve
00	Yönetim	1000	Beacon
00	Yönetim	1001	ATIM
00	Yönetim	1010	Bağlantı Kesme -
00	Yönetim	1011	Kimlik Doğrulama
00	Yönetim	1100	Kimlik Doğrulama İptal
00	Yönetim	1101	Eylem
00	Yönetim	1110	Eylem ACK Yok
00	Yönetim	1111	Rezerve
01	Kontrol	0000–0110	10 & Rezerve
01	Kontrol	0111	Kontrol Sarıcı
01	Kontrol	1000	Blok ACK İsteği (BlockAckReq)
01	Kontrol	1001	Blok AC (BlockAck)
01	Kontrol	1010	PS-Poll
01	Kontrol	1011	RTS
01	Kontrol	1100	CTS
01	Kontrol	1101	ACK
01	Kontrol	1110	CF-End
01	Kontrol	1111	CF-End + CF-Ack
10	Veri	0000	Veri
10	Veri	0001	Veri + CF-Ack
10	Veri	0010	Veri + CF-Poll
10	Veri	0011	Veri + CF-Ack + CF-Poll
10	Veri	0100	Null (veri yok)
10	Veri	0101	CF-Ack (veri yok)
10	Veri	0110	CF-Poll (veri yok)
10	Veri	0111	CF-Ack + CF-Poll (veri yok)
10	Veri	1000	QoS Veri
10	Veri	1001	QoS Veri + CF-Ack
10	Veri	1010	QoS Veri + CF-Poll
10	Veri	1011	QoS Veri + CF-Ack + CF-Poll
10	Veri	1100	QoS Null (veri yok)
10	Veri	1101	Rezerve
10	Veri	1110	QoS CF-Poll (veri yok)
10	Veri	1111	QoS CF-Ack + CF-Poll (veri yok)
11	Rezerve	0000–1111	Rezerve

MAC çerçevesinin tür bilgisine göre çerçeve türleri anlamlandırılmaktadır. Kontrol çerçeveleri bir paketin iletilip iletilmediği ile ilgilenirken, yönetim çerçeveleri AP yapılandırmasına göre istemcilerin AP ile nasıl bağlantı kuracağına ve bağlantı

kurulurken kullanılacak olan yapılandırmayı belirler. Bölüm 3.2'de çalışma şekline detaylı olarak yer verilecek olan erişim noktası yöneticisi, AP yönetimi ve istemcilerin QoS yapılandırmasını yönetim çerçevelerini üzerinden takip etmektedir.



Şekil 2.3: MAC çerçevesi.

2.2. İLGİLİ ÇALIŞMALAR

Bir ağ altyapısında bulunan cihanları yönetmek için birçok yöntem ve standart geliştirilmiş olup SNMP, RMON, CMIP, CAPWAP gibi protokoller ile SDN, SHH, JSON ve XML tabanlı yaklaşımlar başlıcalarıdır.

2.2.1. Ağ Yönetim Protokolleri

SNMP (Simple Network Management Protocol): TCP/IP protokolünün bir parçasıdır ve ağda bulunan cihazların çeşitli bilgilerini içerisinde tutar [1]. Yönetici tarafında çalışan SNMP yazılımı (NMS - Network Management System), yönetilen cihaz tarafında çalışan yazılım (agent) ve her cihazın yerelinde bulunan cihazdaki agent tarafından erişim sağlanan ve cihazla ilgili bilgileri bulunduran bir veri tabanı vardır. SNMP mesaj gönderme ve alma mantığıyla çalışır. Ağdaki cihaza bir mesaj gönderir. Cihaz üzerinde çalışan agent bu mesaja gerekli cevabı verir. Mesajlaşmaları sırasında güvenli olmayan bir aktarım gerçekleştirir. SNMPv3'te güvenlik açıkları giderilmeye çalışılsa da kişisel ağlarda kullanımı pek yaygınlaşmamıştır [9].

RMON (Remote Monitoring): SNMP'de ağ izleme gibi eksiklikleri gidermek için geliştirilmiş bir protokoldür [2]. RMON ile ağ üzerindeki bilgileri istatistiksel olarak toplanarak ağ yöneticilerine sunulmaktadır. SNMP'de bulunan cihaz bazlı izlemelere karşılık RMON akış tabanlı yerel ağ izlemesi sağlamaktadır. RMON'un ilk versiyonu layer1 ve layer2 üzerindeki istatistiksel bilgiler toplarken RMON2 diğer katmanları desteklemektedir [10].

CMIP (Common Management Information Protocol): OSI tarafından geliştirilmiş bir yönetim protokolüdür [3], [4]. TCP/IP'de bulunan SNMP protokolüne alternatif olarak tasarlanmış bir protokoldür. Karmaşık yapısı ve çok fazla kaynak isteyen yazılım (agent) uygulamalarından dolayı TCP/IP cihazlarında desteklenmemektedir. Günümüzde bazı Telekom cihazlarının yönetiminde kullanılmaktadır.

SSH, JSON ve XML Tabanlı Yaklaşımlar: SSH (Secure Shell) Ağ üzerindeki bir cihaza uzaktan güvenli bağlantı kurmayı sağlaya bir protokoldür [5]. Güvenli bir kanal üzerinden iletişim gerçekleşir ve cihaz üzerinde yönetimsel betikler çalıştırılabilir. Uzak noktada bulunan ağ cihazının üzerinde çalışan özelleştirilmiş bir yazılıma, ssh kanalı ile json/xml yapılandırmaları gönderme/alma işlemleri yapılarak cihazların yöntemi sağlanabilmektedir.

SDN (Software Defined Networking): Yazılım Tanımlı Ağ iletişimi, ağ üzerindeki kontrol ve yönlendirme işlemlerini ağ altyapısından soyutlayarak, programlanabilen bir hizmet olarak sunan bir mimaridir[6]. Bu mimarinin temel yapı taşları, direk programlanabilir, çevik, merkeziyetçi ve standartlardan bağımsız olarak tanımlanabilir[7]. SDN herhangi bir standart protokol tanımlamamaktadır. Ağ üzerindeki trafiğin merkezi bir yerden kontrol edilmesini sağlayan bir mimaridir. Bir ağ üzerinde çalışabilmesi için OpenFlow Switch yazılımlarına ihtiyaç vardır. Bu sayede trafik yönlendirmesi yapılabilmektedir. Genel bir mimari sunmakla birlikte kablosuz ağlara özel yönetsel faaliyetleri içeren bir tanım bulunmamaktadır.

CAPWAP (Control And Provisioning of Wireless Access Points - RFC5415): Ağ üzerindeki WTP'leri yönetmek amacıyla geliştirilen bir protokoldür. L2 (Layer 2) bağımsız olarak kablosuz AP'lerin kontrol ve provizyonunu sağlamak amacıyla geliştirilmiştir [7]. AC ve WTP'ler arasındaki iletişim IP - Internet Protokolü ile sağlanmaktadır. Protokol Local ve Split Mac olmak üzere iki çalışma modunu desteklemektedir. Split Mac modunda, bütün L2 veri ve yönetim çerçeveleri CAPWAP protokolü ile paketlenerek AC ve WTP arasında iletişim sağlanır. Local Mac modunda,

[6] SDN, Software Defined Networking, https://www.opennetworking.org/sdn-resources/sdn-definition, [Ziyaret tarihi: 10 Ekim 2016]
[7] OpenFlow, OpenFlow Switch Errate, https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-spec-v1.0.2.pdf , [Ziyaret tarihi: 10 Ekim 2016]

veri çerçeveleri local olarak köpülenebilir veya 802.3 çerçeveleri olarak tünellenebilir. L2 kablosuz yöntemi çerçeveleri WTP tarafından işlenerek AC'ye yönlendirilmektedir.

RFC5416 - (CAPWAP - Protocol Binding for IEEE 802.11) IEEE 802.11 kablosuz cihazların CAPWAP protokolü ile nasıl kullanılacağını belirten bir ilişkilendirme (binding) standardıdır [11].CAPWAP kontrol mesaj alanları, yeni kontrol mesajları ve mesaj elemanlarının nasıl kullanılacağı tanımlanmıştır.

Bu çalışmaların içerisinde CAPWAP protokolü IETF tarafından geliştirilmiş bir standart olup IEEE 802.11 kablosuz ağları hedef alan spesifik bir protokoldür [7]. CAPWAP dışında SDN (Software Defined Network) yaklaşımları da son zamanlarda hız kazanmıştır [12].

2.2.2. SDN Tabanlı Yaklaşımlar

Yan ve diğ. [13] tarafından yapılan çalışmada, SDN'nin eksik yönlerinden biri olan kablosuz ağlar için çözüm üretmeyi hedeflemiş bir çalışmadır. OpenFlow spesifikasyonu tarafından türetilen SDN kavramını [14] genişleterek, kablosuz port, kanal ve etkinlik desteği ekleyerek ÆtherFlow adı altında bir çatı geliştirilmiştir. OpenWRT sistemleri SDN destekli olarak dağıtılmadığından, CPqD SoftSwitch (ofsoftswitch) ile bir eklenti olarak bu yetkinlik kazandırılmıştır. ofsoftswitch eklentileri hostapd[8] ile haritalanarak kablosuz fiziksel ve mantıksal portlar oluşturulmuştur. Bu haritalama ile ÆtherFlow gelen mesajlar hostapd iletilmekte ve aynı şekilde hostapd alınan veriler ÆtherFlow'a ofsoftstwitch ile yönlendirilmektedir. Sadece AP'lerin kanal, iletim gücü, versiyon, ssid, bssid ve güvenlik ile ilgili yapılandırılması yapılabilmektedir. Uygulama hostapd yetenekleri ile sınırlı olup, hostapd bağımlılığı vardır. Gerçek ortam testleri sadece handoff mekanizması ile test edilmiş olup, bir ağ performans optimizasyonu yapılmamıştır. Ayrıca, WTP'lerin yönetimini içeren faaliyetlere yer verilmemiştir.

Dely ve diğ. [15] OpenFlow aracılığıyla IEEE 802.11 MAC katmanı çerçevelerini bir yönetici konsoluna gönderen ve AP yöntemini bulut katmanına çekmeyi hedefleyen bir çalışmadır. Sadece ACK ve tekrar iletim çerçeveleri AP tarafından gönderilmektedir, diğer tüm çerçeveler tünelleme yöntemiyle buluta aktarmaktadır. Transfer edilen çerçeveler merkezi bir noktada işlenebilmektedir. Çalışma OpenWRT destekli Atheros

[8] hostapd, AP sunucusu, http://w1.fi/hostapd/, [Ziyaret tarihi: 10 Ekim 2016]

yonga seti ile test edilmiştir. WTP'ler üzerindeki WLAN arayüzleri izleme (monitor) moduna alınarak, ham çerçevelerin alınması ve iletilmesi sağlanmaktadır. WTP'ler tarafından iletilecek paketlerin hangi fiziksel hızda ileteceğinin belirlenebilmesi için kablosuz ağ cihazının sürücüsü (driver) genişletilmiştir. Başka bir deyişle uygulama sadece bir yonga seti üreticisi için çalışmaktadır ve çalışmanın uygulanabilirliği markaya bağımlı kalmıştır.

Nakauchi ve diğ. [16] yapmış olduğu çalışmada, AP'lerin sanallaştırılarak dinamik baz istasyonları oluşturma işlemi hedeflenmiştir. Böylelikle gerekli şartlar altında ve ihtiyaçlar doğrultusunda sanal AP oluşturulabilecektir. VoIP, SIP çağrı başlangıç zamanını azaltmaya ve handover paket kayıplarını azaltmaya yönelik test çalışmaları yapılmıştır. WTP'lerin nasıl yönetildiğine dair detaylı bir bilgi içermemektedir.

Patro ve diğ. [17] çalışmasında openflow API'leri ile evlerde bulunan kablosuz ağları bulut platformu üzerinden yönetilmesini sağlayacak bir mimari sunmuştur. Sunulan servis tabanlı mimari COAP (Coordination framework for Open APs) ile isimlendirilmiş olup, büyük apartmanlar içinde uygulanabileceği bilgisine yer verilmiştir. Yapılan çalışmada komşu AP'lerin aynı servisi kullanması durumunda bir bölgedeki kanal çakışma ve sinyal girişiminin planlanabileceği gösterilmiştir. Üç ana modülden oluşmaktadır, APConfigManager; kontrol ünitesinden aldığı mesajları AP üzerinde gerçekleştirmektedir. BasicStatsReporter; kablosuz istatistikleri kontrol ünitesine göndermektedir. DiagnosticStatsReporter; tanı amaçlı detaylı kablosuz istatistikleri kontrol ünitesine iletir. Geliştirilen uygulamanın AP tarafı OpenWRT destekli click çatısı ile hazırlanmıştır. click çatısının son güncel versiyonu 24 Eylül 2011 tarihinde yayınlanmıştır. APConfigManager; AP tarafında yapılan yapılandırma işlemini luci aracı ile yapmaktadır. luci temel olarak, işletim sisteminde bulunan yapılandırma dosyalarını güncelleyerek değişikliği devreye almaktadır. EDCA parametrelerinin yapılandırması, beacon ve yönetimi çerçevelerinin oluşturulması gibi işlevleri yerine getirememektedir. Bu noktalardaki eksiklerden dolayı tam anlamıyla bir AP yönetimi yapılamamaktadır.

Soetens ve diğ. [18] tarafından yapılan araştırmada ev ortamları için hibrit ağ yönetim modeli ortaya konulmuştur. Önerilen yöntem SDN tabanlı olup OpenFlow anahtarlarının (switch) üzerine inşa edilmiştir. Heterojen bir ağ içerisindeki bağlantı anahtarlamalarının normal şartlar altında nasıl davrandığı araştırılmıştır. Linux işletim sistemi ile yapılan

gerçek ortam testlerinde, bir UDP trafiğinin WiFi'den Ethernet bağlantısına geçişi incelenmiştir. Çalışmanın temelini heterojen bağlantı noktalarından oluşan bir ağ trafiğinin performansını iyileştirmeye yönelik olup, WTP yönetimi ve yapılandırma çalışma kapsamına alınmamıştır.

Syrivelis ve diğ. [19] yayınlamış olduğu makalede SDN yaklaşımı ile kesintisiz hareket, trafik filtreleme, QoS şema desteği ve VPN olmadan özel ağ kurulumu hizmetlerini verebilecek bir mimari tasarımı yapılmıştır. QoS desteği paket başlıklarına eklenen ek bilgi ile yönetilmektedir, istemci bazlı bilgiler AP üzerinde tutularak trafik akışlar bu bilgiye göre oluşturulmaktadır. Literatürde bulunan QoS algoritmaları kapsam dışı tutulmuştur. Uygulama click platformu üzerinde geliştirilmiş olup, bu platformun en güncel versiyonu 24 Eylül 2011 tarihinde yayınlamıştır. WTP tarafında kanal yapılandırma, paket iletim güç seviyesinin ayarlanması... vb. yönetim fonskiyonlarına çalışmada yer verilmemiştir.

Stiti ve diğ. [20] sunduğu çalışmada roaming sorunlarına çözüm bulmak amacıyla SDN tabanlı bir mimari geliştirmişlerdir. Sanallaştırma yöntemi ile WiFi AP'ler tamamen izole edilerek bir birinden bağımsız olarak aynı WiFi yonga setinin üzerinde çalıştırılabileceği aktarılmıştır. Sanallaştırmanın güvenlik, esneklik ve ağ kapsama alanını genişletmesi yönüyle artıları vurgulanmıştır. Çalışmanın testlerin yapıldığı hakkında bilgi olmakla birlikte sonuçlar sayısal verilere dayanmamaktadır ve sadece kavramsal bir model sunulmuştur. WTP tarafında yapılması gereken işlemler ve yönetim yöntemleri hakkında bir bilgi içermemektedir.

Yamasaki ve diğ. [21] yapmış oldukları çalışma, kampüs kablosuz ağlarının yönetimini sağlamak amacıyla esnek OpenFlow tabanlı bir yaklaşım ile çözüm sunmaya çalışılmıştır. Kampüs ağlarında bulunan çok sayıdaki VLAN olması ve her bir düğümün ayrı ayrı yapılandırması gerektiği düşünüldüğünde, ihtiyaç duyulan iş gücünü azalmak için kolay yapılandırmanın yapılabileceği bir mimari sunulmuştur. Çalışma NEC'in OpenFlow kontrol ünitesi ile Linux üzerinde FreeRADIUS sistemleri test edilmiştir. Testler kullanıcı kimlik denetim ve doğrulaması ile icmp protokolünün cevap süreleri üzerinden test edilmiştir. AP yönetimini içeren paket iletim güç seviyesinin belirlenmesi, EDCA yapılandırılmasının yapılıp yapılamaması gibi süreçler hakkında herhangi bir bilgi içermemektedir.

Villain ve diğ. [22] tarafından gerçekleştirilen çalışma, kimlik denetim ve doğrulama yöntemlerini SDN yaklaşımıyla bulut mimarisini aktarmak üzerine bir çözüm sunmuştur. Çalışmanın temel hedefinde bulut kontrolü ile misafir erişiminin kontrol edilebileceği bir yapı kurulması üzerine yoğunlaşmıştır. Test sonuçları sayısal veriler üzerinden verilmemiş olup, VPN ile karşılaştırılmalı bir liste üzerinden sunulmuştur. WTP yönetimi üzerine herhangi bir yapıdan bahsedilmemektedir.

Sun ve diğ. [23] yapmış oldukları çalışmada kognitif radyo (cognitive radio) yaklaşımını kullanarak, heterojen geniş bant teknolojilerini kolay yönetebilmek için SDN tabanlı bir mimari sunmuşlardır. LTE femtocell ve WiFi teknolojilerinin bir arada bulunduğu bir ağda spektrum izleme ve yönetim faaliyetlerini gerçekleştirebilmek çalışmanın temel hedefi olmuştur. Kontrol ünitesi KVM ve QEMU ile sanallaştırılarak bulut altyapısı oluşturulmuştur. Kontrol ünitesi ile kablosuz cihazlar arasında iletişim sağlamak amacıyla geliştirilmiş bir katman vardır. Kontrol ve veri olmak üzere iki kanal ile kablosuz ortam simule edilmektedir. Çalışmada spektrum kullanımı ve dinamik spektrum geçişleri üzerine bir tasarım ortaya konulup herhangi bir sayısal veri paylaşılmamıştır, ayrıca ağ üzerinde kablosuz cihazların yönetsel faaliyetlerine yer verilmemiştir.

Zubow ve diğ.[24] tarafından yapılan çalışmada trafik ihtiyaçlarının ve spektrum bilgisinin merkezi bir yerde toplanarak, spektrum atamalarının hakkında karar veren bir yapı sunulmuştur. Çalışmada spektrum atama üzerine bir algoritma geliştirilmiştir. Buna göre istemciden bağımsız adil hücre (cell) kullanımı, istemci bazlı spektrum kullanımı, aktif istemci bazlı spektrum kullanımı, hücre içerisindeki aktif akış bazlı spektrum ataması modeli geliştirilerek testleri yapılmıştır. Çalışmada spektrum kullanımına ağırlık verilmiş olup IEEE 802.11'e özel yönetim faaliyetlerini içermemektedir.

Raschellà ve diğ.[25] yayınlamış oldukları makale, SDN tabanlı bir çalışma olup FF (Fittingness Factor) ile AP seçimi sağlayan merkezi bir algoritma sunar. Kontrol ünitesi üzerinde çalışan algoritma, yeni bir akış oluştuğu zaman, FF algoritması ile belirtilen eşik değere göre akış ilgili AP'e atanmaktadır. Çalışmalar simülasyon ile desteklenmiş olup gerçek ortamda test edilmemiştir. Ayrıca AP bazlı yapılandırma üzerine herhangi bir çalışma yapılmamıştır.

Wu ve diğ.[26] tarafından yapılan çalışmada, OpenFlow ağ altyapısı olarak belirlenmiş ve 802.11 kablosuz ağlar üzerinden felaket senaryolarında iletişim sağlamak amacıyla FC-WiFi adı altında bir prototip sunulmuştur. Merkezi bir kontrol ünitesi tarafından ağ üzerindeki trafik akışının sürekliliği sağlanmak istenilmiştir. Yapılan testler handover mekanizması üzerine olup, kanal yapılandırması, paket iletim güç seviyesinin belirlenmesi veya QoS yapılandırılması gibi senaryolardan bahsedilmemektedir.

Rangisetti ve diğ.[27] yapmış oldukları çalışmada, AP üzerindeki yük bilgisini merkezi bir yapıya taşıyarak handoff algoritmaları ile yük dengeleme üzerine bir çalışma yapılmıştır. Çalışmada programlanabilir ODIN mimarisi kullanılarak Atheros destekli yonga seti için eklentiler yapılmıştır. Handoff algoritmaları ile yük dengeleme üzerine testler yapılmıştır fakat AP yapılandırılmasına dair herhangi bir bilgi içermemektedir.

Vestin ve diğ.[28] sunulan makalede QoS yönetiminin SDN yardımıyla bulut (CloudMAC) altyapısında gerçekleştirilmesini anlatan bir çalışma yapılmıştır. Öncelikli olarak, trafik önceliği ve kuyruk yönetim stratejileri için gerekli alt yapı çalışmaları yapılarak OpenFlow altyapısı genişletilmiştir. Ağ üzerindeki cihazlara kurulmak üzere OpenDaylight uygulaması geliştirilmiştir. Böylelikle, CloudMAC üzerinde yapılan değişikliklerle trafik önceliği cihazlara aktarılabilmektedir. Son olarak da, CloudMAC üzerinde trafik önceliği yönlendirecek stratejiler geliştirilmiştir. Testler Atheros yonga seti OpenWRT destekli AP'ler ile Linux tabanlı kontrol üniteleri ile yapılmıştır. Yapılan test çalışmalarında trafik önceliğini ön planda tutmaktadır, çalışma mac80211 (hwsim) sürücüsünün değiştirilmiş versiyonunu içermektedir. Bu da sürücü ve donanım üreticisi bağımlılığına yol açmaktadır.

Chu ve diğ.[29] tarafından yapılan çalışmanın odak noktası düşük maliyetlerde son kullanıcının hizmet kalitesini QoE'nin ölçülebilmesi için bir mimari sunulmuştur. Makalede klasik mimarilerde ağ üzerindeki cihazlardan bilgi toplamak daha maliyetli olduğuna ve SDN tabanlı çözümlerle, merkezi bir nokta üzerinde trafik akışı üzerinden QoE ölçülebilirliği tartışılmıştır. Yapılan testlerde gerçek ortam yerine, sanallaştırma yöntemi ile son kullanıcılar simule edilmiştir. Çalışmanın sonunda elde edilen çıktılar bir referans noktasıyla karşılaştırılmamış yanı sıra WTP yönetimi üzerinde de detaylı bilgi içermemektedir.

2.2.3. CAPWAP ile İlgili Çalışmalar

Bernaschi ve diğ.[30] tarafından yapılan çalışmada çok sayıdaki AP kurumlarının yönetim sorununu çözmek amacıyla IETF tarafından sunulan CAPWAP protokolünün bir açık kaynak kod implemantasyonu yapılmıştır. QoS, yük dengeleme ve kanal frekans planlamasının CAPWAP mimarisi ile gerçekleştirilmesi üzerine yapılan bir araştırmadır. Yine [31] tarafından yayınlanan diğer bir çalışmada AP'lerin yönetim sorunu çözmek amacıyla sunulan CAPWAP protokolünün üzerinde çeşitli algoritmalar kullanılarak yönetimsel faaliyetler test edilmiştir. CAPWAP protokolü hakkında bilgi, uygulamanın mimarisi ve CAPWAP ile QoS uygulanabilirliği üzerinde durulmuştur. Yapılan çalışmalarda QoS yapılandırmalarının CAPWAP ile nasıl yapılacağı hakkında bilgi verilmiştir. Testlerin sonuçları ağ üzerinde oluşan gecikme sürelerinin ölçümü üzerinden gerçekleştirilmiştir. Makalede sunulan yük dengeleme algoritması; ağdaki istemci sayısını maksimize etme ve bir WTP'ye bağlı maksimum istemci sayısını minimize etmek üzerine kurulmuştur. Bu tür bir yük dengeleme algoritması her zaman ağ üzerinde dengeli yük dağılımı yapıldığı anlamına gelmez, farklı kullanıcıların istekleri doğrultusunda farklılık gösterebilir. Ayrıca, testlerde sinyal kayıp (path loss) modeli kullanılmamasının yanı sıra, istemcilerin fiziksel veri iletimi, paket boyutları göz ardı edilmiştir. Testlerde kullanılan WTP'ler Madwifi sürücüsüne bağımlıdırlar.

Bernaschi ve diğ.[32] tarafından yapılan çalışmada CAPWAP protokolünün çalışma prensibi ve durum makinesi ile protokol çalışma prensibi incelenmiştir. Ayrıca yazarlar tarafından geliştirilmiş olan açık kaynak kodlu OpenCapwap projesinin 0.92 versiyonuyla birlikte gelen özel veri paketleri gönderilmesini sağlayan bir eklentide geliştirilmiştir. Komut satırı ile WTP listesi, güncel yapılandırma, yapılandırma ekleme silme ve WTP cihazını yeniden başlatma gibi özellikler barındırmaktadır. Ayrıca çalışmada openCapwap kullanılarak frekans planlaması üzerine bir çalışma yapılmıştır. WTP tarafında Atheros yona setlerine sahip Madwifi sürücüsü kullanılmış olup çalışmalar belirli bir marka üzerinden test edilmiştir.

Levanti ve diğ.[33], [34] tarafından hazırlamış olan makalede spesifik bir CAPWAP uygulaması geliştirilerek, geniş kablosuz ağlarda otomatik yönetim ve frekans optimizasyonu üzerine çalışmalar yapılmıştır. CAPWAP protokolü hakkında genel bilgilerle frekans çakışması ve planlaması yöntemleri üzerinde durulmuştur. Yapılan çalışmalar ns2 ile simüle edilmiş olup TestBed ortamının hakkında detay içermemektedir.

Clancy ve diğ. [35] kablosuz ağlarda hızlı handover problemine çözüm getirmek için tasarlanan, IEEE 802.11r, CAPWAP, HOKEY protokollerini karşılaştırmaktadır. IEEE 802.11i protokolünü ise güvenlik açısında ele almıştır. Her bir protokolün handover süreleri karşılaştırmalı olarak değerlendirilmiştir. Makaleye göre, CAPWAP protokolü, kurumsal WLAN altyapılarını hızlı kurmak ve yönetmek için idealdir. HOKEY genel amaçlı bir handover mekanizmasını sunarken IEEE 802.11r ise merkezi yönetimine ihtiyaç duyulmayan performans ve hız ihtiyaçlarının ön planda olduğu handover senaryoları için idealdir. Çalışma protokol karşılaştırması yapmış olup WTP'lerin yönetimsel faaliyetleri hakkında detaylı bilgi bulunmamaktadır.

Sarikaya ve diğ.[36] yapmış oldukları çalışmada CAPWAP protokolü hakkında genel bir bilgi vermekle birlikte, protokol üzerine handover yetkisi kazandırmak amacıyla CAPWAPHP'ü (CAPWAP handover protocol) geliştirmişlerdir. Geliştirilen yöntem OPNET simülasyonu üzerinde gerçekleştirilmiş olup, gecikme zamanları ölçülerek değerlendirilmiştir. WTP'lerin yönetim fakiyetlerinin bulunmaması ve gerçek ortam testlerinin yapılmaması çalışmanın eksik yönleridir.

2.2.4. Yük Dengeleme ile İlgili Çalışmalar

IEEE 802.11 standartlarına göre, istemciler AP seçerken ve AP ile bağlantı kurarken, AP üzerindeki yük göz ardı edilerek, RSSI seviyesi en yüksek olan AP ile bağlantı kurulur [8]. Bunun doğal sonucu olarak kullanıcılar bir ağ üzerinde eşit olmayan bir şekilde dağılır ve kullanıcı sayısı ile birlikte ağ üzerinde alınan hizmet kalitesi de düşmektedir. Hem hücresel hem de WiFi ağlarında yük dengeleme için birçok çalışma yapılmış olup bu alanda yapılan önemli çalışmalar şu şekildedir.

Bahl ve diğ. [37] WLAN'da, tüm ağın çıktısını (throughput) maksimum seviye çıkartmak için bir Cell Breathing (CB) algoritması sunmuştur. CB kullanıcıların AP'lere bağlanmasında kullanılabilecek merkezi, basit ve uygulanabilir bir yöntemdir. CB algoritması istemcilerin, AP'ler tarafından gönderilen RSSI seviyelerini kontrol ve analiz ettiğini varsayımına dayanır. CB algoritmasının çalışması için standartlarda herhangi bir değişikliğe gerek kalmamaktadır. CB algoritması AP'ler tarafından gönderilen beacon çerçevelerindeki güç bilgisinin değiştirerek kullanıcının AP ile bağlantı kurma stratejisini kontrol eder.

Bejerano ve diğ. [38], Bahl ve diğ. [37] tarafından yapılan çalışmayı genişleterek, CB ile AP'ler arası adil kullanım şeması sunar. Makalede sunulan algoritmalar, kullanıcıların AP'ye bağlanma stratejisi, ağ çıktısı (throughput) veya trafik ihtiyaçlarına göre geniş bir alanda yük dengeleme tanımlarını desteklemektedir. Makalede, maksimum-minimum, bant genişliği, zaman ve ağırlık tabanlı adil kullanım için çözümler sunsa da iki konuyu göz ardı etmektedir. (i) ölçülen RSSI değerinin fiziksel veri hızına etkisi ve (ii) kullanıcıların AP'lere bağlanma etkileşimi, fiziksel veri hızı ve QoS yapılandırması.

Song ve diğ. [39] çalışmasında Kablosuz Ağ (Wireless LAN) ile Hücresel (Cellular) arasında kaynak paylaşımı yapan bir çözüm sunmaktadır. WLAN üzerindeki yükü belirtilen ilkeler (policy) doğrultusunda hücresel ağlara dikey el değiştirme (vertical handover) ile gerçeklemektedir. Bu çözümde ortaya çıkan en büyük sorun sadece WLAN'dan oluşan bir ağ yapısına uygulanamamasıdır.

Yanmaz ve diğ. [40] çalışmalarında sunulan yöntem dinamik yük dengelemesinin kanal ayırma (channel allocation) yöntemi ile yapılmaktadır. Yazarlar bu çalışmada ISM bantlarındaki çakışmaları inceleyerek iCar sisteminin performansını artırmaktadırlar. Yapılan çalışmalar kanal ayırma ve çağrı tıkama olasılıklarını (call blocking probabilities) temel almaktadır. WLAN ağlarda AP ile istasyon sabit bir kanal üzerinden iletişime başlar ve çakışmadan kullanılabilecek kanal sayısı üç ile sınırlı olduğu için, kanal ayırma, kanal ödünç alma (channel borrowing) gibi yöntemler WLAN'larda uygulamamaktadır.

Kim ve diğ. [41] herhangi bir merkezi sunucu kullanmadan global performans iyileştirmesi yapılan çalışmada yazar, kullanıcı terminal bağlantısı homojen olmayan trafik yapıları için incelmektedir. Geliştirilen sistem, WiMAX2, 3GPP-LTE downlink kullanıcı bağlantısında (downlink user association) hücreler arası çakışmanın statik ve kabul edilebilir bir seviyede varsayılmaktadır. Bu model LTE ve WiMAX gibi hücresel teknolojileri için tasarlanmış olup, WLAN için kullanılabilirliği hakkında detaylı bir çalışma yoktur.

Tonguz ve diğ. [42] matematiksel bir çatı (framework) sunan çalışmada yazar hücresel ağlarda sunulan çözümlerin benzerliğinden yola çıkmaktadır. Çalışmadaki ana amaç simülasyon ortamı kullanmadan, algoritmaların performansını ölçen bir model sunmaktadır. Ortaya atılan teoriye göre model, el değiştirme (handover) performans

problemleri, kanal kullanımı (channel utilization), farklı erişim şemaları için farklı servislerin analizi, düzgün (uniform) olmayan trafik dağılımının incelenmesi, çağrı engelleme (call blocking) olasılığı hesaplanması... vb. konularda kullanılabilmektedir. Yazar ayrıca bu modelin sadece hücresel ağlarda değil aynı zaman WLAN'lar ve AP içinde kullanılabileceğini iler sürmektedir, fakat WLAN ile kullanılabileceğine dair herhangi bir çalışma sunmamıştır.

Bejerano ve diğ. [43] bu çalışmada yazarlar kullanıcılara adil servis sağlamak için çözüm ortaya sunmaktadır. Ortaya atılan fikir ile kullanıcı ile AP arasındaki bağlantının akıllı bir şekilde kurulmasını sağlamaktır. Bu çalışmada ortaya atılan yöntemde kullanıcı ile AP bağlantısı kontrol edilebileceği varsayımı yapılmıştır. Bu durumda kullanıcıların eklentiler kurmalarını gerektirmektedir (ayrıca standartların dışına çıktığı için) ki buda halka açık alanlar için uygulanabilir çözüm olmaktan çıkmaktadır.

Gong ve diğ. [44]sezgisel (heuristic) algoritmalar ile WiFi ağlarında yük dengeleme çözümü sunmaktadır. Geliştirilen algoritma AP'lerde beacon frame ile araştırma paketlerine (probing packets) ek veriler ekleyerek yapmaktadır. Çalışma standartların dışına çıktığı için AP ve kullanıcılara eklenti kurmayı gerektiren bir durumdur. Bu tür çalışmaların halka açık alanlarda uygulanabilirliği olmadığı için kullanım alanı kısıtlı kalmaktadır.

Park ve diğ. [45] sunduğu çözüm ile hücresel ağlarda video uygulamalarını hedef almaktadır. Çalışmada sunulan optimizasyon video uygulamalarında çerçeve kayıplarını (frame drop) minimize etmektir ve bu yöntem ile sistemin yükü dengelenmektedir. Çerçeve kayıplarının az olduğu hücrelere kullanıcılar atanmaktadır. Çalışma hücresel ağlarda yapılmış olup 802.11 kablosuz ağları için herhangi bir bilgi içermemektedir.

Muñoz ve diğ. [46] çalışmasında bulanık mantık algoritmaları ile hücresel ağlardaki (LTE) yük dengeleme problemlerini çözmeye çalışmaktadır. Bahsedilen yöntem hücresel ağlar üzerinden çalışılmış olup bu yöntemin 802.11 ağları için uygulanabilirliği hakkında yeterli bilgiye yer verilmemiştir.

Ağ altyapı yönetimi için ortaya atılan yöntem ve stratejiler incelendiğinde, standart dışı yöntemler veya belirli bir markaya ait donanım (yonga seti) üzerinde çalışmalar yapıldığı görülmektedir. Yapılan çalışmalar ya kavramsal olarak kalmış veya gerçek ortamlarda

testleri yapılmamıştır. Benzer bir durum yük dengeleme algoritmaları için geçerli olup, çalışmaların birçoğu hücresel ağlar için tasarlanmış olup 802.11 kablosuz ağlarda az sayıda çalışma vardır. IEEE 802.11 kablosuz ağlar için yapılan çalışmalarda ise, standartlar üzerindeki kısıtlar birçok araştırmacı standart dışı yöntemlerle yük dengeleme çözümleri üretmeye itmiştir [43], [44] bunun sonucunda sunulan çözüm halka açık ağlarda kullanılabilirliğini yitirmektedir.

Ağ yönetim protokolleri ve diğer yaklaşımlar incelendiğinde, yapılan çalışmaların ana iki konuda eksik olduğu gözlemlenmektedir;

1. IEEE 802.11 Kablosuz Ağlara özel standart bir yöntem ile ağ alt yapıları merkezi bir noktadan yönetilmemektedir.
2. Yapılan çalışmalar simülasyon ortamlarında test edilmiş veya sadece bir donanıma özel ortamlarda test edilmiştir. Sağlayıcıya, donanıma ve mimariye bağımsızlık sunamamaktadır.

Ayrıca, yük dengeleme algoritmaları incelendiğinde yapılan çalışmaların beş noktada eksik olduğu gözlemlenmiştir;

1. Sunulan yük dengeleme algoritmaları paket boyutu veya veri iletim hızını göz ardı etmektedir.
2. QoS ile kullanıcıların hizmet önceliği göz ardı edilmiştir.
3. Sadece bir AP'a bağlı kullanıcılar arası adil trafik kullanımı üzerine çözümler sunulmuştur ve bir den fazla AP olduğu ortamlar göz ardı edilmiştir.
4. IEEE 802.11 standartlarının dışına çıkılarak son kullanıcılara eklentiler yüklenerek çözüm aranmıştır. Halka açık alanda uygulanabilirliği bulunmamaktadır.
5. Geliştirilen yük dengeleme algoritmaları gerçek simülasyonlar ile sınırlı testler yapılmıştır.

Standart protokoller vasıtasıyla bir kablosuz ağ altyapısı yönetim faaliyetlerini gerçekleştiren, yazılım tabanlı ve marka bağımsız çözümlere ihtiyaç duyulmaktadır. Bu tez çalışması kapsamında sunulan çözüm CAPWAP destekli olup, merkezi bir noktadan WTP'lerin yönetim ve yapılandırmasına imkan sunmaktadır. Böylelikle IEEE 802.11e kablosuz ağlar için tasarmış olduğumuz QoS (Quality of Service) tabanlı yük dengeleme

algoritmaları, AC vasıtasıyla TestBed ortamında gerçek cihazlar üzerinde test edilerek sonuçlar değerlendirilmiştir. Bu sayede yukarda bahsedilen sorunlara çözüm sunulmaya çalışılmıştır.

3. MALZEME VE YÖNTEM

IEEE 802.11 kablosuz ağlar klasik hücresel ağlara benzer prensipler barındırmaktadır. Her bir hücre; Temel Servis Kümesi (Basic Service Set - BSS) olarak adlandırılır ve bir baz istasyonu(AP) tarafından yönetilmektedir. Her bir hizmet kümesinin kendisine ait kimliği vardır (Service Set Identifier - SSID) ve bu kimlik beacon mesajları ile anons edilir. Eğer birden fazla AP aynı SSID ile MAC katmanına bağlı ise bu sistemlere Dağıtık Sistemler (Distributed System - DS) denilir ve Genişletilmiş Servis Kümesini (ESS) oluştururlar. IEEE 802.11 spesifikasyonları, istemciler için; Kimlik Doğrulama, Kimlik Doğrulama İptali, Gizlilik ve MSDU iletim servislerini içerirken, dağıtım hizmetleri için; AP ile Bağlantı Kurma, Bağlantı Bitirme, Tekrar Bağlantı Kurma ve ESS Dağıtımı... vb. hizmetleri içermektedir. Bu mimaride ağ alt yapısı birinden bağımsız AP'ler tarafından temsil edilir ve her bir AP ayrı ayrı yapılandırılmaktadır. Ayrı bir merkezi operasyon ve yönetim sistemi bulunmamaktadır. AP'lerin yoğun olarak kullanıldığı ağ altyapılarında her bir AP'nin ayrı ayrı yapılandırılması ek iş gücü ve zaman gerektirmektedir. Tek bir AP'de yapılacak eksik yapılandırma ağ performansını ve güvenliği etkileyeceği için hatalara da kapı aralamaktadır. AP'lerin merkezi bir noktadan yönetilme ihtiyacı IETF görülmüş ve CAPWAP protokolü geliştirilerek yayınlanmıştır [7], [11].

CAPWAP protokolü, büyük ölçekli ağlarda kablosuz ağların kolay kurulumuna, ağ genelinin izlenmesine ve ağın merkezi bir noktadan yapılandırmasına imkan sunan merkezi bir yönetim mimarisi tanımlar. Protokol tarafından Yerel MAC (Local MAC) ve Ayrı MAC (Split MAC) olmak üzere iki farklı ağ mimarisi belirlenmiştir. Yerel MAC mimarisinde, tüm MAC fonksiyonları WTP tarafından yürütülmektedir, AC ise WTP'nin yapılandırmasını ve erişim politikalarını kontrol etmektedir. Ayrı MAC mimarisinde, MAC katmanın bazı fonksiyonları AC tarafından yürütülmektedir.

3.1. CAPWAP PROTOKOLÜ

CAPWAP, Kablosuz Erişin Noktalarının Kontrol ve Provizyonu (Control And Provisioning of Wireless Access Points) protokolü, ağ altyapısında ki WTP'lerin yönetimi sağlamak amacıyla geliştirilmiş protokoldür [7]. IEEE 802.11 kablosuz ağlarda

karşılaşılan yönetim sorununu çözmek için geliştirilse de, WTP radyo teknolojisinden bağımsız olarak, diğer kablosuz teknolojiler ile kullanılabilir. RFC5416, kablosuz protokol bağlantılarının nasıl yapılacağını belirtmektedir [11].

CAPWAP protokolünün hedefleri şu şekildedir;

- Kablosuz ağlardaki kimlik doğrulama, yönetim, yapılandırma politikaların merkeziyetçi bir yapıya dönüştürülmesi, maliyetleri düşüreceği gibi ağdaki verimliliği artıracaktır.
- WTP üzerinden yüksek seviyedeki işlemlerin kaldırılması, WTP tarafındaki işlemci gücünü daha etkili kullanmayı sağlayacaktır.
- Protokol; spesifik bir kablosuz teknoloji için geliştirilmemiştir bu sayede gelecekte, birçok kablosuz cihazla kullanım olanağı olacaktır.

CAPWAP protokolü AC ve WTP arasındaki iletişimi ve yönetimin nasıl yapılacağını tanımlar. İki farklı UDP kanalı üzerinden kontrol ve veri paketleri taşınmaktadır. DTLS (Datagram Transport Layer Security) protokol güvenliğini sağlamaktadır [47]. WTP'lerin yönetim, kontrol ve izleme işlemleri CAPWAP kontrol mesajları ayrı bir UDP portu üzerinden gerçekleştirilmektedir. WTP tarafından AC'ye gönderilen kablosuz çerçeveler, CAPWAP veri mesajları ile paketlenerek iletilmektedir.

CAPWAP protokolü bir Keşif (Discovery) fazıyla başlar. WTP tarafından gönderilen bir keşif istek mesajı *DiscoveryRequest* , AC tarafından *DiscoveryResponse* cevap mesajıyla karşılanır. WTP, AC seçerek güvenli bir bağlantı kurar. Güvenli bağlantı kurulduktan sonra yapılandırma değişimi gerçekleşir. Bu işlemlerden sonra WTP operasyonel işlemler için hazırdır.

Protokol, WTP'ye bağlı istemcileri yönetmek için gerekli komutları AC üzerinden iletilmesini sağlayan bir mekanizma sunar. Ayrıca AC, WTP üzerinden istatistiksel bilgilerde toplayabilirler. Ayrıca, protokol tarafından iletişim kanalının aktif kalması için WTP - AC arasında *keep-alive* mesajları gönderilir, eğer AC erişim sağlanmazsa, WTP yeni bir AC keşfine başlar.

CAPWAP, herhangi bir WTP radyo teknolojisine bağımlı değildir. Protokolde kullanılan elemanlar, kablosuz teknolojilerin ihtiyaçlarını standart bir biçimde tanımlamasını içerir.

CAPWAP protokolünün herhangi bir kablosuz teknoloji ile implementasyonunun yapılabilmesi için, ilgili teknoloji için bağlantı (binding) ihtiyaçlarını yerine getirmelidir. Bu bağlantıların herhangi bir teknolojiyle kullanılabilmesi:

- WTP tarafından taşınan isteklerde (Event Request) teknolojiye özel İstatistik (Statistics) mesaj elemanları tanımlanmalı.
- WTP tarafından istemci yapılandırmalarında kullanılan İstemci Yapılandırma İstek (Station Configuration Request) mesajları.
- WTP Radyo Bilgisi içeren mesaj elemanı

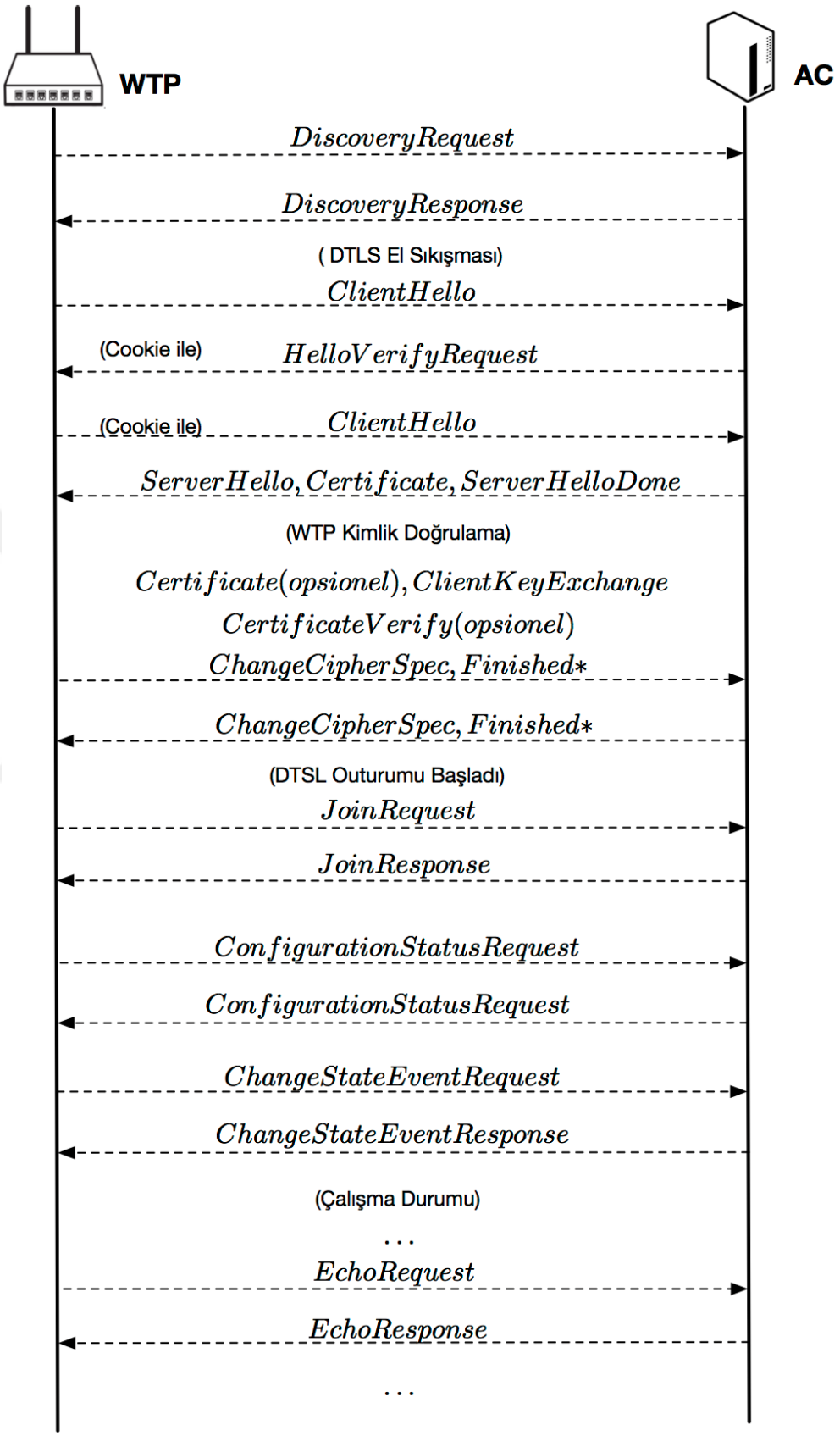
gereksinimlerinin (minimum) sağlanması gerekmektedir.

CAPWAP protokolü ile bir oturum başlaması süreci Şekil 3.1'de gösterilen akış şemasına göre gerçekleşmektedir. WTP keşif istekleriyle başlayan işlemler adım adım izlenir, DTLS oturumu kurulduktan sonra WTP, AC'nin takip etmiş olduğu WTP'lerin arasında yerini alır. Bunun arkasından yapılandırma bilgileri karşılıklı gönderilir ve WTP çalışma durumuna geçer.

Şekil 3.2'de görünen CAPWAP durum makinesi, hem AC hemde WTP tarafından kullanılmaktadır. WTP veya AC özel geçişler ayrıca belirtilmiştir. Her bir durum için sadece kesin mesajlar gönderilmekte ve alınmaktadır. Protokol hangi mesajların hangi durumlar için olduğunun tanımını içermektedir.

WTP sadece bir AC ile iletişim kurduğu için, durum makinesinin sadece bir nesnesi bulunmaktadır. AC için durum farklıdır, birden fazla WTP takip edildiği için AC, dinleyen (Listener), keşif (Discovery) ve servis (Service) olmak üzere üç ayrı alt işlem parçacığı(thread) mekanizması ile bu süreci yönetmektedir.

Listener: DTLS oturum isteklerini yöneten alt işlemdir. *DTLSListen* komutu alınmasının ardından, DTLS oturum başlatılması durumuna geçer, oturum doğrulandığında Kimlik Doğrulama (*Authorize*) durumuna geçilir. WTP oturumlarını yönetmek üzere bir servis (*Service*) alt işlemi oluşturulur.



Şekil 3.1: CAPWAP protokol mesaj değişimi.

Discovery: AC üzerindeki keşif alt işlemi, keşif mesajlarını dinler ve bu mesajlara cevap verir. WTP bazlı herhangi bir durum takibi yapılmaz.

Service: Her bir WTP için bir oturumun yönetildiği işlemdir. *Listener* tarafından oluşturulur ve WTP ile iletişim bittiğinde işlem biter, WTP bağlı bütün kaynaklar serbest bırakılır.

Şekil 3.2'de verilen CAPWAP protokolü durum makinası ve durum geçişleri aşağıdaki şekilde özetlenmiştir.

Başlangıç -> Idle: Cihaz açılış işlemini tamamlayıp agent uygulama çalıştığı zaman bu duruma geçer.

Idle -> Keşif: WTP ilk keşif isteğini gönderir ve keşif aralığını belirleyen zamanlayıcı başlatır. AC keşif istediğini aldığı zaman cevap mesajı oluşturur.

Keşif -> Keşif: WTP tarafından sayaç sıfırlandığında bu duruma geçer. Eğer birden fazla AC ile yapılandırma mevcut ise WTP diğer AC'lar için keşif mesajları göndermeye başlar.

Keşif -> Idle: AC tarafından WTP'ye keşif cevabı gönderildiği durumdur.

Keşif -> Sulking: Keşif işlemi başarısız gerçekleştiği durumdur.

Sulking -> Idle: WTP keşif işlemini belirlenen zamanlayıcı sonrasında tekrar başlatır.

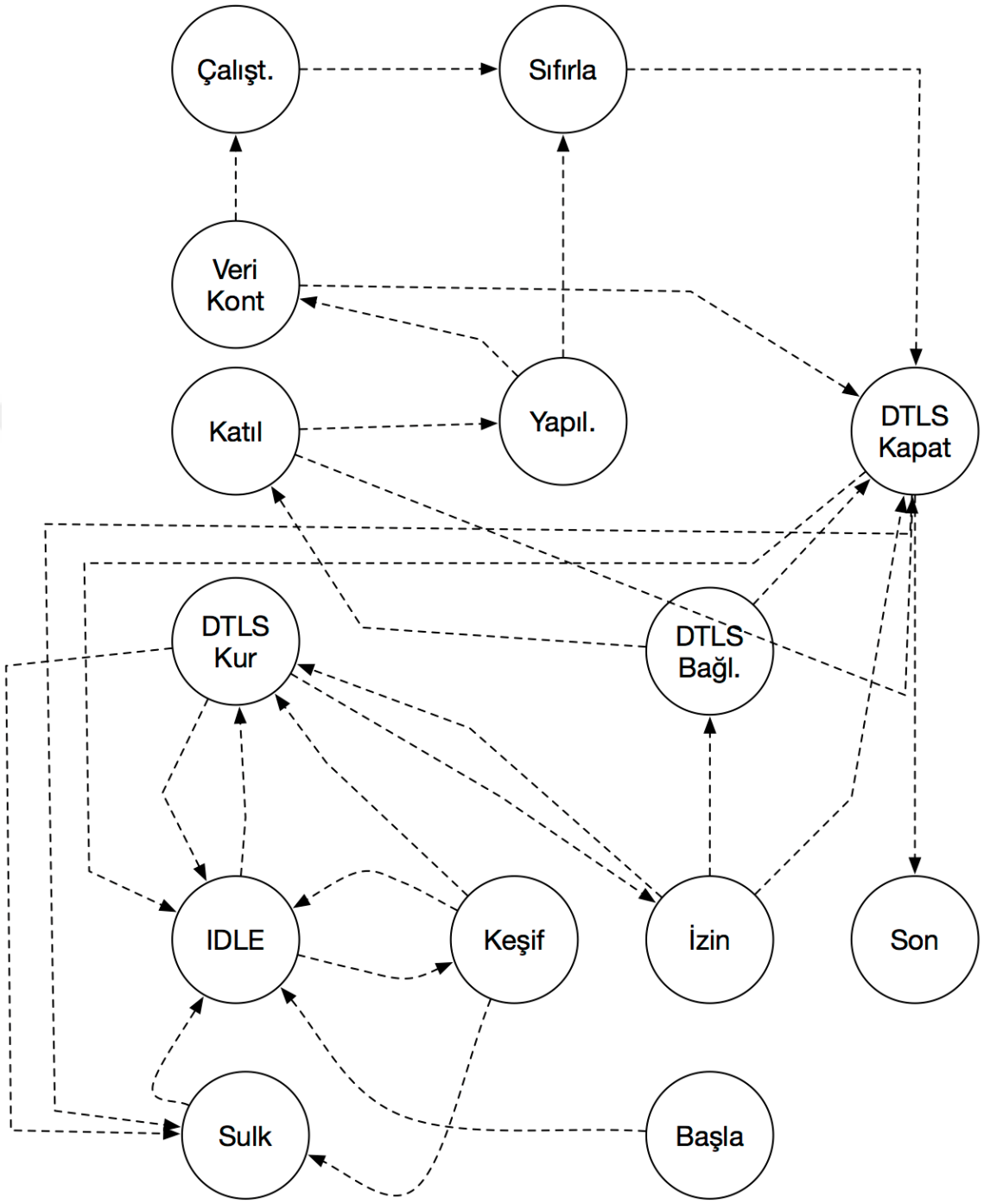
Sulking -> Sulking: DoS saldırılarını önlemek için WTP sessiz bir duruma geçer.

Idle -> DTLS Kurulum: WTP güvenli DTLS bağlantısını başlatılır.

DTLS Kurulum -> Idle: DTLS bağlantısının başarısız olduğu durumdur.

DTLS Kurulum -> Sulking: Tekrarlanan başarısız DTLS bağlantı denemesinin sonucunda oluşan durumdur.

DTLS Kurulum -> DTLS Kurulum: Başarısız DTSL bağlantısının ardından AC'nin tekrar DTLS dinleme moduna geçtiği durumdur.



Şekil 3.2: CAPWAP durum makinesi.

DTLS Kurulum -> Yetkilendirme: DTLS bağlantısının kurulduğunda oluşan durum.

Yetkilendirme -> DTLS Kurulum: DTLS bağlantısı sonrasında AC tarafından yeni bir DTLS bağlantısı için hazırlık yaptığı durumdur.

Yetkilendirme -> DTLS Bağlantısı: DTLS bağlantısı başarılı olduktan sonra, DTLS komutlarının gönderildiği durumdur.

Yetkilendirme -> DTLS Kapatma: WTP'nin DTLS bağlantısındaki Yetkilendirmenin başarısız olduğu durumdur.

DTLS Bağlantısı -> DTLS Kapatma: DTLS oturumunun başarısız olduğu durumdur.

DTLS Bağlantısı -> Katılma: DTLS oturumu başarılı bir şekilde kurulduktan sonra WTP, Katılma (Join) mesajlarını gönderdiği durumdur.

Katılma -> DTLS Kapatma: Katılma mesajlarının başarısız olduğu durumdur.

Join -> Görüntü Verisi: Firmware güncelleme işlemlerinin yapıldığı durumdur.

Join -> Yapılandırma: WTP ve AC tarafından yapılandırma verilerinin paylaşıldığı durumdur.

Yapılandırma -> Sıfırlama: Bir hata oluşması durumunda bağlantının sıfırlandığı durumdur.

Yapılandırma -> DTLS Kapatma: DTLS hatasından dolayı yapılandırma durumun sıfırlanmasıdır.

Görüntü Verisi -> Görüntü Verisi: Firmware güncellemesinin indirildiği durumdur.

Görüntü Verisi -> Reset: Firmware güncelleme verisinin indirme işleminin bittiği veya zaman aşımına uğradığı durumdur.

Görüntü Verisi -> DTLS Kapatma: DTLS hatasından dolayı indirme işleminin sıfırlanmasıdır.

Yapılandırma -> Veri Kontrol: AC ve WTP tarafından yapılandırmanın doğrulandığı durumdur.

Veri Kontrol -> DTLS Kapatma: Yapılandırmanın doğrulamasının yapılamadığı durumdur.

Veri Kontrol -> Çalışma: Veri ve kontrol kanallarının kurulduğu ve WTP ve AC'nin normal çalışmaya başladığı durumdur.

Çalışma -> DTLS Kapatma: DTLS kaynaklı bir hatadan dolayı çalışmanın durduğu durumdur.

Çalışma -> Çalışma: Operasyonun normal bir akışta devam ettiği durumdur.

Çalışma -> Sıfırlama: Normal bir operasyon sonucu veya beklenmedik bir hata sonunda çalışmanın sıfırlanması durumudur.

Sıfırlama -> DTLS Kapatma: DTLS oturumunun kapatılması durumudur.

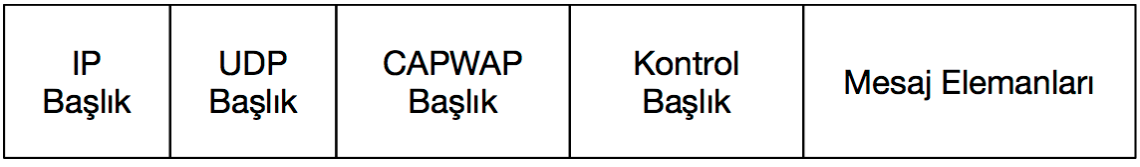
DTLS Kapatma -> Idle: DTLS oturumunun kapatıldığı durumdur.

DTLS Kapatma -> Suking: Tekrar eden başarısız DTLS oturumunun sonucunda oluşan durumdur.

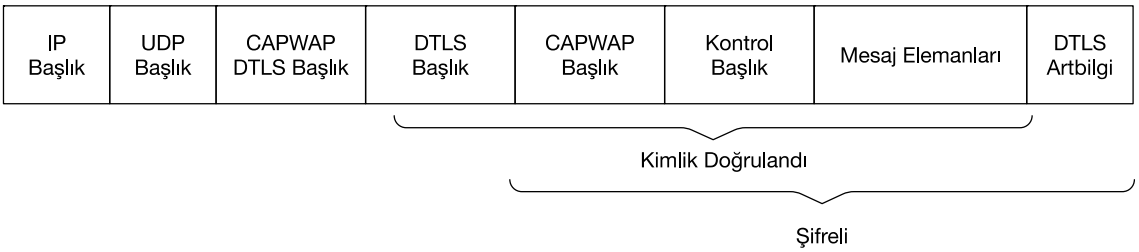
DTLS Kapatma -> Sonlandırma: DTLS oturumunun tamamen kapandığı durumdur.

Bir CAPWAP protokol paketi bir veya daha fazla mesajdan oluşabilmektedir. CAPWAP mesajları Kontrol türünde veya Veri türünde olmaları mümkün. Kontrol türündeki paketler sinyal taşırken, Veri türündeki paketler kullanıcı veri yüklerini taşır.

Kontrol protokolü her zaman şifrelenmemiş açık metin olarak iki mesaj türünü barındırır. *DiscoveryRequest* ve *DiscoveryResponse* mesajları hiç bir zaman şifrelenmezler. Bu iki mesaj için genel format Şekil 3.3'de gözükmektedir. Diğer mesaj türleri için mesajlar DTLS ile şifrelenmesi gerekmektedir, bu paketler için format Şekil 3.4'de verilmiştir.

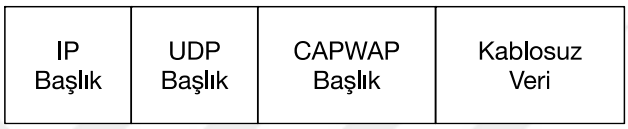


Şekil 3.3: CAPWAP kontrol paketi açık metin.

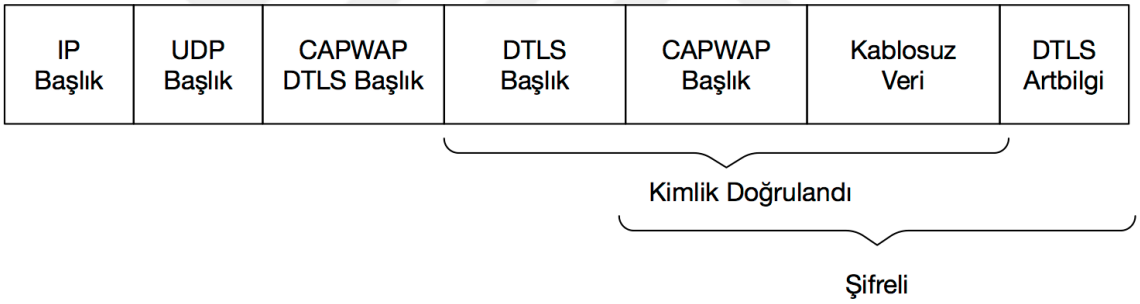


Şekil 3.4: CAPWAP kontrol paketi şifrelenmiş.

Protokol veri paketlerinin de opsiyonel olarak güvenliğini sağlayabilir. AC politikaları ile bu işlem sağlanabilir. Şekil 3.6'de açık metin, Şekil 3.6'de ise şifrelenmiş veri paketlerinin formatı gözükmektedir.



Şekil 3.5: CAPWAP veri paketi açık metin.



Şekil 3.6: CAPWAP veri paketi şifrelenmiş.

UDP Başlığı: Tüm CAPWAP çerçeveleri UDP (veya UDP-Lite IPv6 kullanılıyorsa) ile paketlenmelidir.

CAPWAP DTLS Başlığı: Tüm DTLS ile şifrelenmiş CAPWAP protokol paketleri, CAPWAP DTLS ön başlık bilgisini içermelidir.

DTLS Başlığı: DTLS başlığı, CAPWAP tarafından taşınan verilere, kimlik doğrulama ve şifreleme yetisi sağlar.

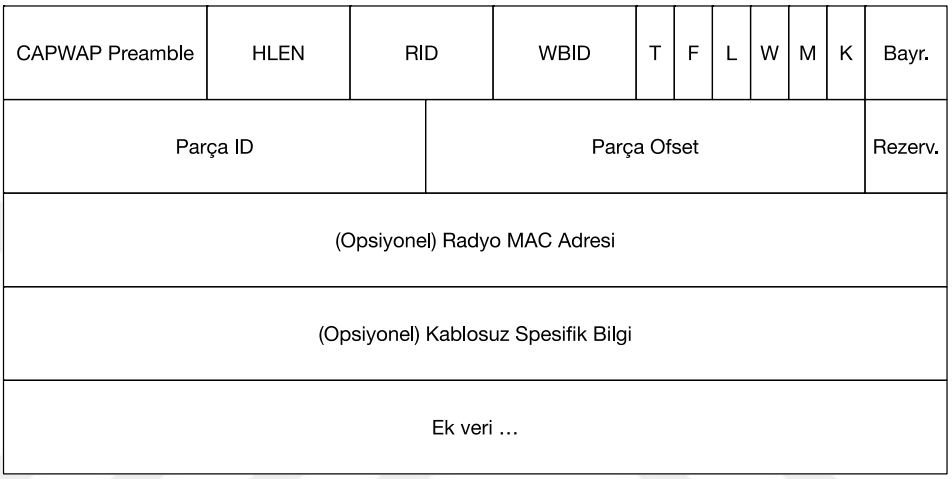
CAPWAP Başlığı: Tüm CAPWAP protokol paketlerinin ortak taşıdığı başlıktır.

Kablosuz Veri: CAPWAP veri paketidir. Veriler için herhangi bir format belirtmez.

Kontrol Başlığı: Kontrol protokolüne ait sinyalleri içeren paket başlığıdır.

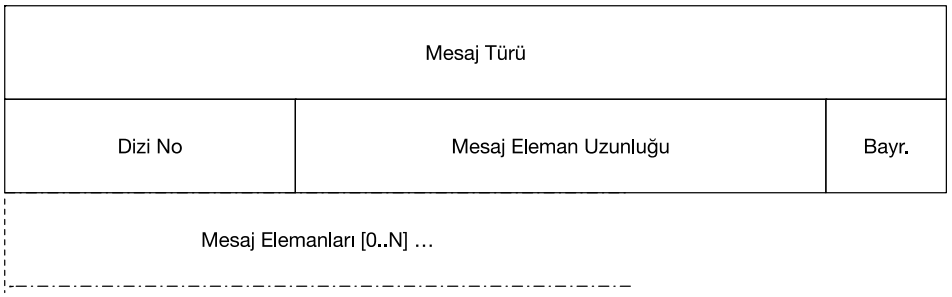
Mesaj Elemanları: Kontrol başlığını takip eden mesaj elemanlarıdır, Tür/Uzunluk/Veri formatındadırlar.

Bütün CAPWAP protokol mesajları ortak bir başlık içermek zorundadır. Bu başlık protokol türüne göre (kontrol veya veri), içerdiği alanlardaki veri değerleri değişiklik gösterir. Bu başlık bilgisinin formatı Şekil 3.7'de gösterilmiştir.



Şekil 3.7: CAPWAP başlığı.

Bütün CAPWAP kontrol mesajları, Şekil 3.8'de gösterilen kontrol başlık formatı ile gönderilmektedir.



Şekil 3.8: CAPWAP kontrol mesaj başlığı.

Mesaj Türü: CAPWAP kontrol mesajın türünü belirtmektedir. Protokol gereği mesaj türleri istek ve cevap olmak üzere çiftler halinde tanımlanması gerekmektedir. Tüm istek mesajları tek, cevap mesajları da çift sayı olacak şekilde numaralandırılmıştır. Geçerli CAPWAP kontrol mesajları Tablo 3.1'de listelenmiştir.

Tablo 3.1: CAPWAP kontrol mesajları.

Kontrol Mesajı	Mesaj Değeri
DiscoveryRequest	1
DiscoveryResponse	2
JoinRequest	3
JoinResponse	4
ConfigurationStatusRequest	5
ConfigurationStatusResponse	6
ConfigurationUpdateRequest	7
ConfigurationUpdateResponse	8
WTPEventRequest	9
WTPEventResponse	10
ChangeStateEventRequest	11
ChangeStateEventResponse	12
EchoRequest	13
EchoResponse	14
ImageDataRequest	15
ImageDataResponse	16
ResetRequest	17
ResetResponse	18
PrimaryDiscoveryRequest	19
PrimaryDiscoveryResponse	20
DataTransferRequest	21
DataTransferResponse	22
ClearConfigurationRequest	23
ClearConfigurationResponse	24
StationConfigurationRequest	25
StationConfigurationResponse	26

3.2. ERİŞİM NOKTASI YÖNETİCİSİ MİMARİSİ

Kurumsal bir ağ topolojisi incelendiğinde Şekil 3.9 olduğu gibi son kullanıcılar (istemciler), AP/WTP, switch/hub ağ elemanları ve suculardan oluşan bir sistemden meydana geldiği görülür. Ağ altyapısının büyüklüğüyle orantılı olarak WTP sayısı değişecektir. Büyük ölçekli bir kablosuz ağ altyapısını, geleneksel yöntemlerle yönetmek hem iş gücü hem de ağda meydana gelecek hataların tespiti ve giderilmesi açısında oldukça zordur ve zaman kayıplarına yol açmaktadır. Geliştirmiş olduğumuz yazılım tabanlı Erişim Noktası Yöneticisi (AC) ile bu ve benzeri altyapıları tek bir merkezi noktadan izleme, yapılandırma ve yönetimsel faaliyetleri gerçekleştirmek mümkün olacaktır.

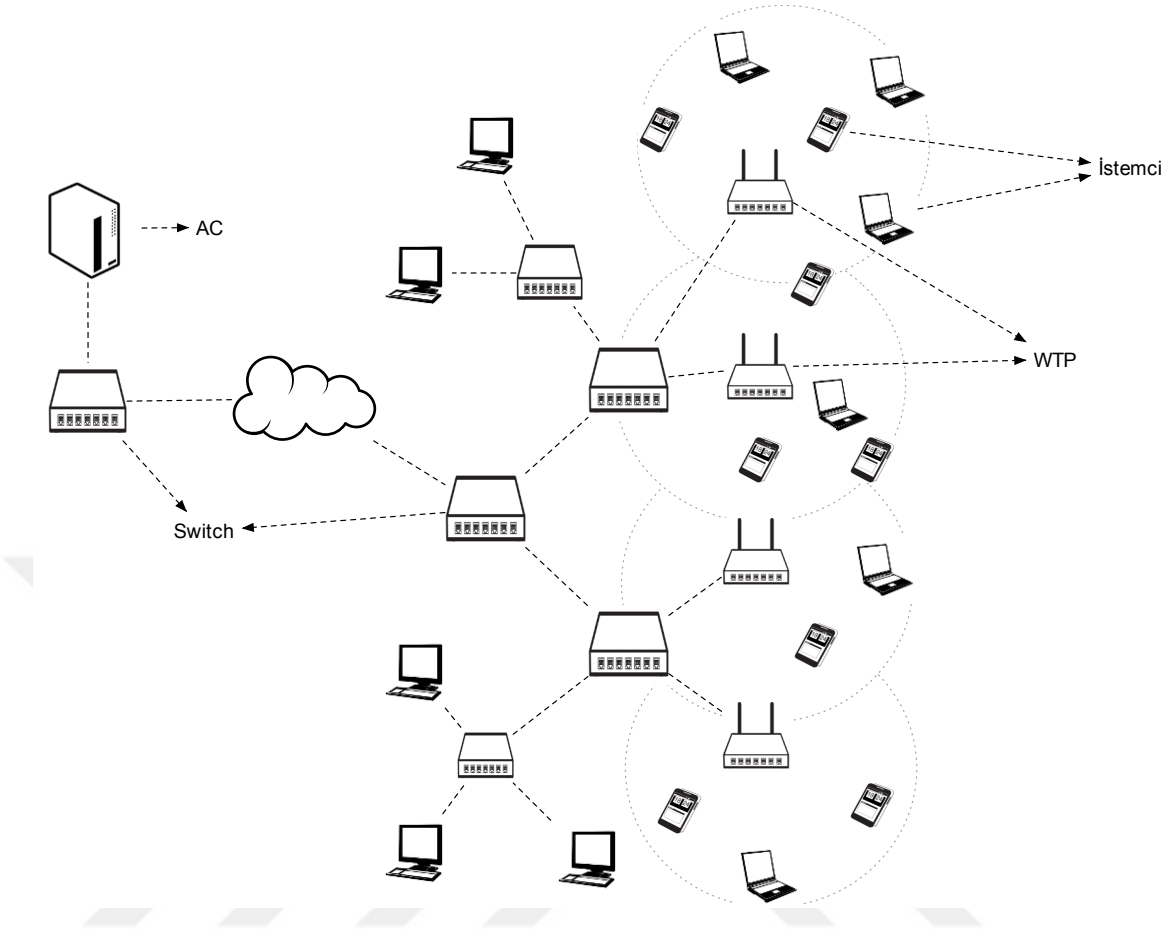
Erişim Noktası Yöneticisi, AC ve WTP olmak üzere iki ana uygulamadan oluşmaktadır. Geliştirilen uygulamalar CAPWAP protokolünün bir implementasyonunu içermektedir [7]. İzleme, yapılandırma ve yönetim faaliyetleri AC tarafından WTP'lere gönderilen CAPWAP kontrol mesajları üzerinden yürütülmektedir. WTP'ler üzerinde çalışan bir uygulama bu mesajları dinleyerek, AC'den gelen komutlara göre gerekli işlemleri gerçekleştirir.

WTP'ler sadece bir AC ile iletişim halinde bulurken, AC birden fazla WTP ile iletişim halinde olup her bir WTP'nin oturumunu takip etmesi gerekmektedir. Protokol standartlarında tavsiye edilen yöntem üzerine, geliştirilen AC uygulaması üç farklı alt işlem parçacığı içermektedir. AC, CAPWAP tarafından tanımlanan durum makinasını (Şekil 3.2) bu alt işlemler parçaları ile takip eder.

Keşif İş Parçacığı: WTP'ler tarafından gönderilen keşif isteklerini (Discovery Request) takip ederek, isteklere cevap veren işlemlerin takip edildiği iş parçasıdır.

Oturum (Hizmet) İş Parçacığı: DTLS oturumu başlatıldıktan sonra, her bir WTP'yi ayrı ayrı takip ve yapılandırmasını yapan işlem parçacığı.

Dinleyen İş Parçacığı: DTLS oturumlarının yönetildiği iş parçaççığıdır.

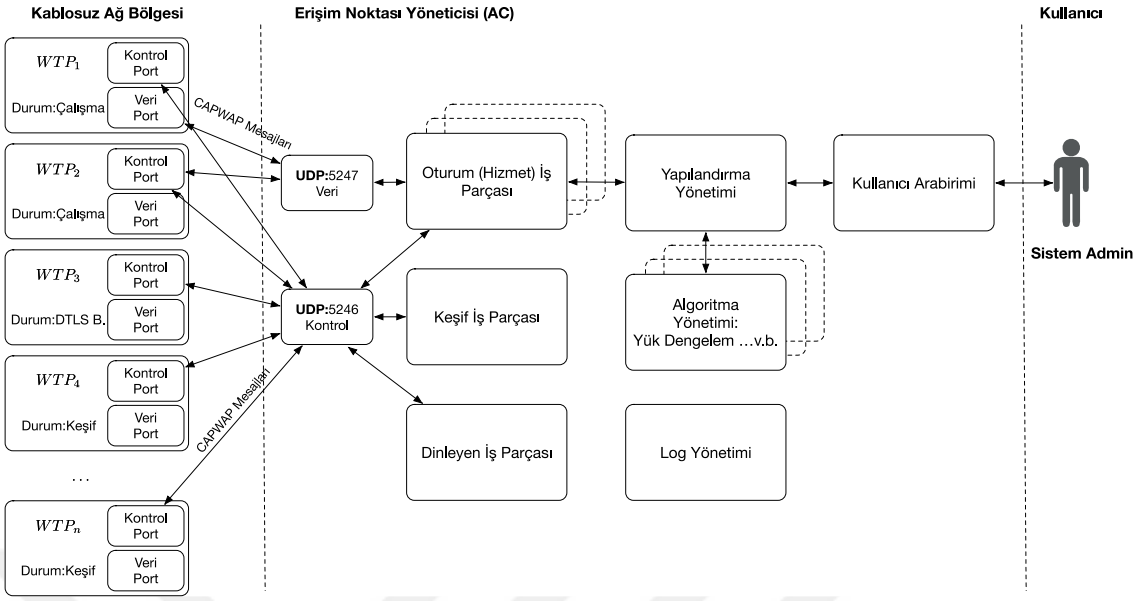


Şekil 3.9: Kurumsal ağ topolojisi.

3.2.1. AC Uygulaması

Erişim Noktası Yöneticisinin bir parçası olan AC uygulamasının WTP'ler ile olan iletişimi mimarisi Şekil 3.10'de gösterilmiştir. WTP'ler kontrol ve veri kanalları ile AC ile iletişim kurmaktadırlar. CAPWAP kontrol protokolü 5246/udp, veri protokolü ise 5247/udp portları üzerinden iletişim kanalları kurulur. AC uygulaması C++14[9] programlama dili ile farklı platformlara port edilebilecek şekilde geliştirilmiştir. AC uygulamasının testleri Apple macOS platformu ile Ubuntu Linux üzerinde test edilmiş olup, Windows platformunda da derlenip çalıştırılabilir.

[9] C++14, Genel bakış, https://isocpp.org/wiki/faq/cpp14, [Ziyaret tarihi: 10 Ekim 2016]



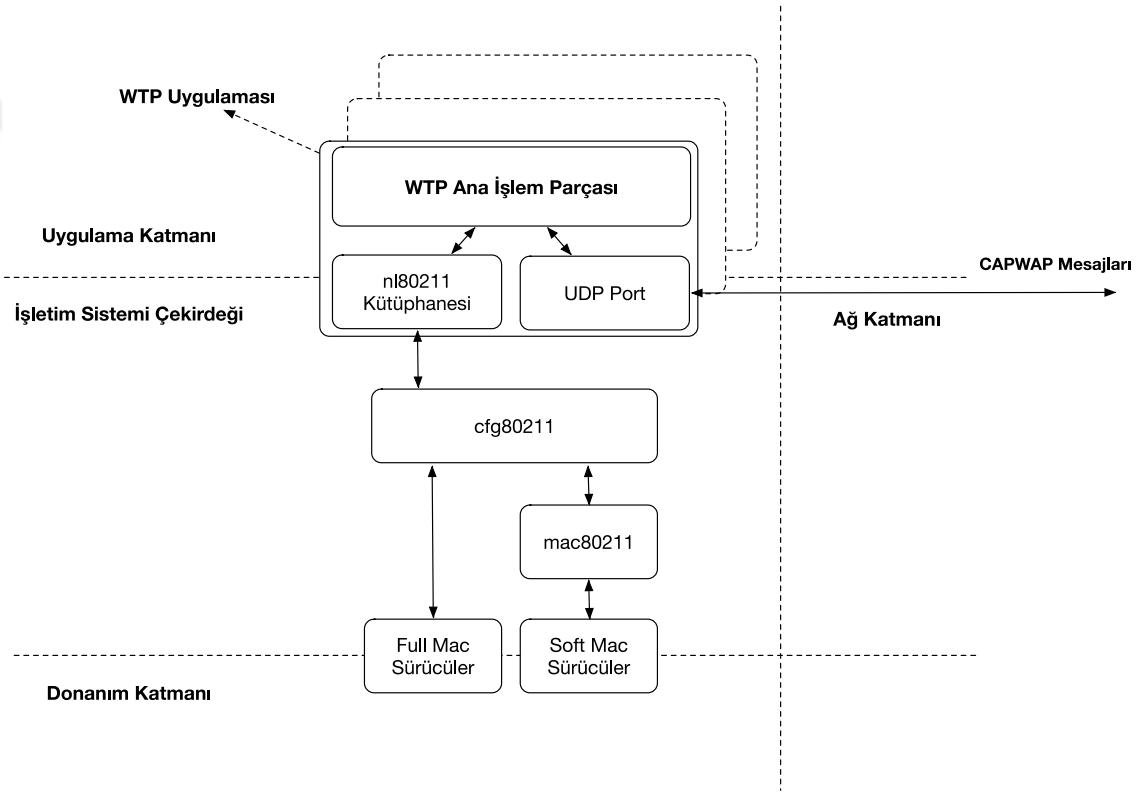
Şekil 3.10: Erişin Noktası Yöneticisi mimari yapısı

Şekil 3.10'da görüldüğü üzere, AC uygulaması, Oturum, Keşif, Dinleme (DTLS), Yapılandırma, Algoritma, Log ve Kullanıcı arabirimlerinin oluşturduğu modüllerden oluşmaktadır. WTP'ler ile olan bağlantı ve kimlik doğrulama işlemlerini Keşif ve Dinleme (DTLS) modülleri tarafından gerçekleştirilmektedir. Kimlik doğrulama yapıldıktan sonra, her bir WTP ayrı ayrı Oturum modülü ile takip edilmektedir ve Yapılandırma modülünden almış olduğu parametreler doğrultusunda WTP'lerin üzerindeki parametreleri güncellemektedir. Sistem yöneticisi tarafından Kullanıcı Arabirimine aktarılan Yeni bir WLAN ekleme/silme, QoS parametreleri güncelleme, Kullanıcı ekleme/silme... vb. komutlar, Yapılandırma modülünden, Oturum modülüne aktarılır ve WTP'ler üzerindeki uygulama sayesinde cihaz üzerinde icra edilir. Loglama modülü ise, WTP-AC arasında gerçekleşen her türlü işlemi metin türünde kayıt altına alarak saklar. Ayrıca yapılandırma işlemleri, istatistiksel veriler ve Log kayıtları kullanıcı arabirimi tarafından takip edilebilmektedir.

Algoritma Yönetim modülü ise, ağ üzerindeki uygulanacak Yük Dengeleme, QoS ile adil hizmet kalitesi gibi çalışmaları otomatik olarak gerçekleştirmesini sağlayan bir yapıya sahiptir. Yük Dengeleme algoritmaları çalıştırıldığında, ağ üzerindeki WTP'lerin ve kullanıcıların bilgilerinden yola çıkılarak, gerekli güç ve EDCA yapılandırması yapılarak sistem üzerinde uygulanmaktadır. Ağ üzerindeki toplam çıktı ise istatistiklerden takip edilebilmektedir.

3.2.2. WTP Uygulaması

WTP uygulaması ana bir işlem parçacığı barındırmaktadır ve AC üzerinden gelen protokol mesajlarını analiz ederek, gerekli yapılandırmaları nl80211 kütüphanesi ile işlem sistemi çekirdeğine iletir. Şekil 3.11'de görüldüğü üzere işletim sistemi ile olan entegrasyon nl80211 kütüphanesi ile yürütülmektedir. AC tarafından iletilen CAPWAP mesajları işlenmesinin yanı sıra, mevcut yapılandırma bilgisinde işletim sisteminden sorgulanarak mevcut parametreler CAPWAP paketlerine dönüştürülerek ilgili AC'a iletilmektedir.



Şekil 3.11: WTP uygulama mimarisi.

Netlink[10], kullanıcı arabirimi ile işletim sistemi çekirdeği arasında bilgi paylaşmasını sağlayan bir sistemdir. Standart soket tabanlı ara yüzlerden oluşur ve işletim sistemi çekirdek API'lerini kullanmaya olanak tanır. nl80211[11] bir netlink kütüphanesi olup, 802.11 kablosuz ara yüzlerine erişim imkanı sunmaktadır. nl80211 kütüphane sayesinde, işletim sistemi çekirdeği üzerinden kablosuz arabirimlere ulaşım sağlanıp

[10] netlink, işletim sistemi çekirdeği ile kullanıcı arasındaki arabirim, http://man7.org/linux/man-pages/man7/netlink.7.html, [Ziyaret tarihi: 10 Ekim 2016]

[11] nl80211, netlink kablosuz arabirim kütüphanesi, https://wireless.wiki.kernel.org/en/developers/documentation/nl80211, [Ziyaret tarihi: 10 Ekim 2016]

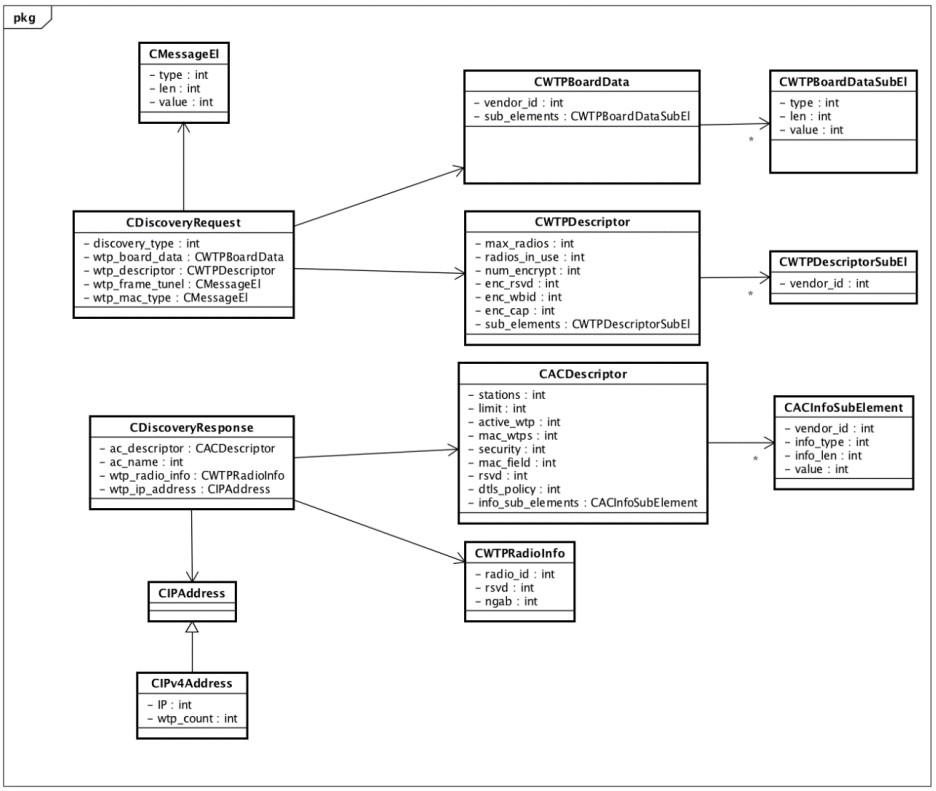
sorgulanabilmektedir ve gerekli birçok yapılandırma bu arayüz ile yapılabilmektedir. nl80211'in bize sunduğu en önemli özellik, standart bir arabirim olmasının yanı sıra OpenWRT destekli işletim sistemlerinde çalışmasıdır. Bu sayede OpenWRT çalıştığı her platformda, herhangi bir donanım veya sürücü bağımlılığı olmadan WTP'ler yönetilebilmektedir. Uygulama geliştirilirken OpenWRT'nin son güncel stabil sürümü olan Chaos Clamer v15.05.1 kullanılmıştır[12].

WTP uygulaması çalıştığı zaman ilk olarak, işletim sistemine nl80211 arayüzünden mevcut yapılandırmaları sorgulayarak, bir veri yapısında tutmaktadır. Daha sonra ise, ağ üzerindeki aktif AC'leri bulmak için keşif istekleri *DiscoveryRequest* göndermeye başlar. WTP'ler belirtilen aralıklarda bu mesajları bir cevap *DiscoveryResponse* mesajı alıncaya kadar göndermeye devam ederler. Belirtilen maksimum mesaj gönderme isteğine ulaşıldığında WTP'ler *Sulking* durumuna geçerek beklerler. AC bir keşif isteği aldığında *DiscoveryResponse* gönderir ve WTP AC ile oturuma dahil olmak için bir sonraki duruma geçerler. WTP ve AC tarafından gönderilen istek ve cevapların olduğu sınıf diyagramları Şekil 3.12'de gösterilmiştir, ayrıca Tablo 3.2'de keşif istek ve cevaplarda bulunan mesaj elemanları gösterilmiştir.

Tablo 3.2: Keşif istek ve cevap mesaj elemanları.

Mesaj Elemanı	İstek/Cevap
WTPBoardData	İstek
WTPDescriptor	İstek
WTPFrameTunnelMode	İstek
WTPMACType	İstek
WTPRadioInformation	İstek/Cevap
ACDescriptor	Cevap
ACName	Cevap

[12] OpenWRT, Chaos Calmer sürümü, https://downloads.openwrt.org/chaos_calmer/15.05.1/, [Ziyaret tarihi: 10 Ekim 2016]



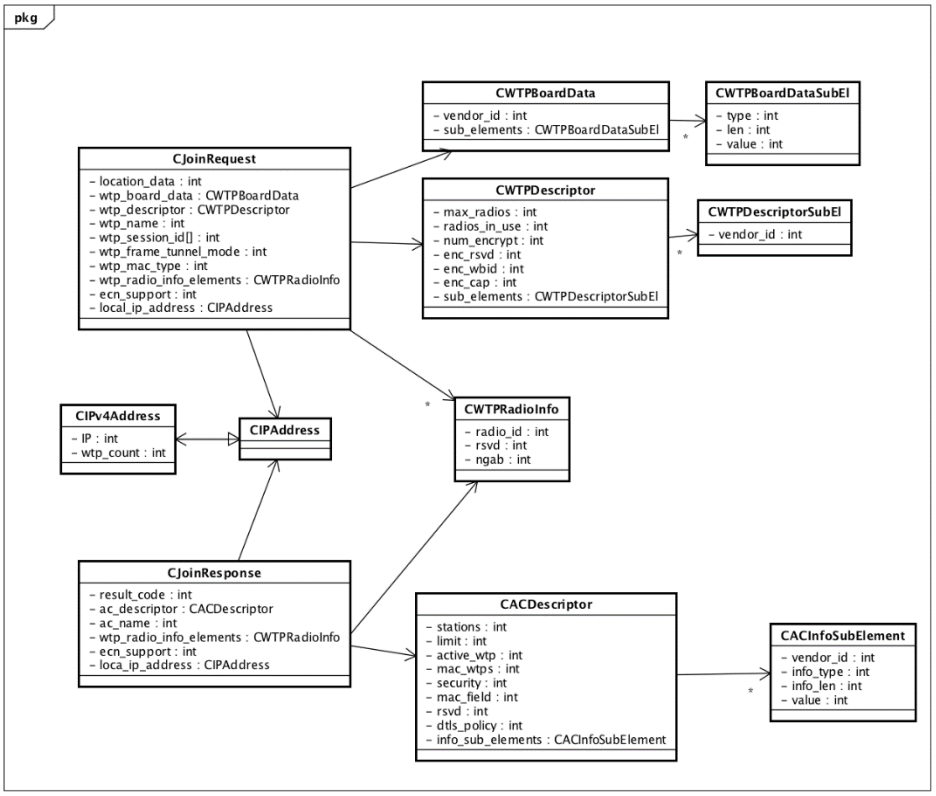
Şekil 3.12: Keşif istek ve cevap sınıf yapısı.

Katılma isteği *JoinRequest* WTP tarafında, AC'den hizmet almak istediğini belirten mesajdır. Güvenli bir kanaldan katılma isteğini aldığında AC, aynı kanal üzerinden *JoinResponse* mesaj gönderir.

Tablo 3.3: Katılma istek ve cevap mesaj elemanları.

Mesaj Elemanı	İstek/Cevap
LocationData	İstek
WTPBoardData	İstek
WTPDescriptor	İstek
WTPName	İstek
SessionID	İstek
WTPFrameTunnelMode	İstek
WTPMACType	İstek
WTPRadioInformation	İstek/Cevap
ECNSupport	İstek
CAPWAPLocalIPv4/CAPWAPLocalIPv6	İstek
ResultCode	İstek
ACDescriptor	Cevap
ACName	Cevap
ECNSupport	Cevap
CAPWAPControlIPv4Address/CAPWAPControlIPv6	Cevap
CAPWAPLocalIPv4Address/CAPWAPLocalIPv6	Cevap

Katılma istek ve cevapları sınıf yapısı Şekil 3.13'de gösterilmiştir. Tablo 3.3'de bu mesaj türü içinde gönderilen mesaj elemanlarını içermektedir.



Şekil 3.13: Ağa dahil olma istek ve cevap sınıf yapısı.

Kanal yönetimi için, *EchoRequest* ve *EchoResponse* mesajları gönderilmektedir. WTP çalışma durumuna geçtiği vakit, *EchoRequest* mesajlarını belirtilen aralıklarda göndermeye başlar. Bu mesajlara AC, *EchoResponse* mesajları ile cevap gönderir. Eğer AC, bir zamanlayıcı dahilinde *EchoRequest* istekleri almazsa, WTP iletişim kaybetmiş kabul eder.

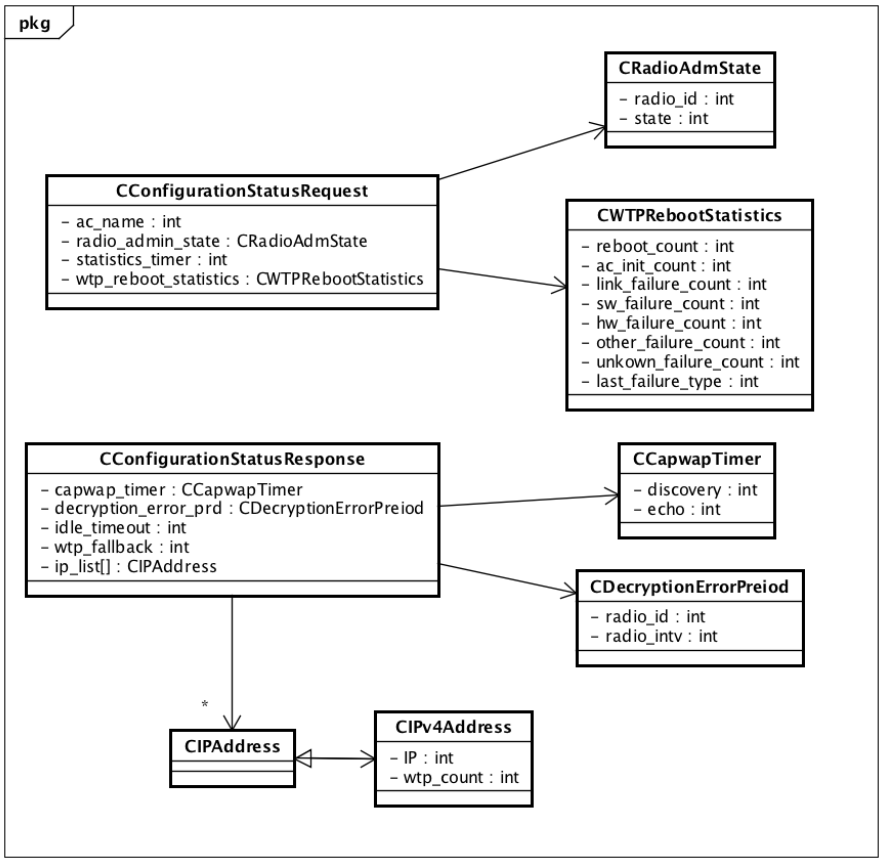
Protokol gereği WTP iki şekilde yapılandırmaları takip edebilir. Birinci yöntem; WTP yapılandırma tutmaz, AC'den gelen yapılandırmaları kabul eder. İkinci yöntem ise yapılandırmaları kaydeder. Geliştirdiğimiz uygulamalarda WTP yapılandırmaları kayıt altına almaktadır.

WTP mevcut yapılandırmasını AC'ye *ConfigurationStatusRequest* isteği ile iletir. Mesaj, ilgili cihazın yapılandırmasını içerir ve WTP yapılandırma durumunda ise bu mesaj gönderir. Şekil 3.14'de sınıf yapısı, Tablo 3.4 ise mesaj elemanları görülmektedir.

Tablo 3.4: Yapılandırma durum istek ve cevap mesaj elemanları

Mesaj Elemanı	İstek/Cevap
ACName	İstek
RadioAdministrativeState	İstek
StatisticsTimer	İstek
WTPRebootStatistics	İstek
CAPWAPTimers	Cevap
DecryptionErrorReportPeriod	Cevap
IdleTimeout	Cevap
WTPFallback	Cevap

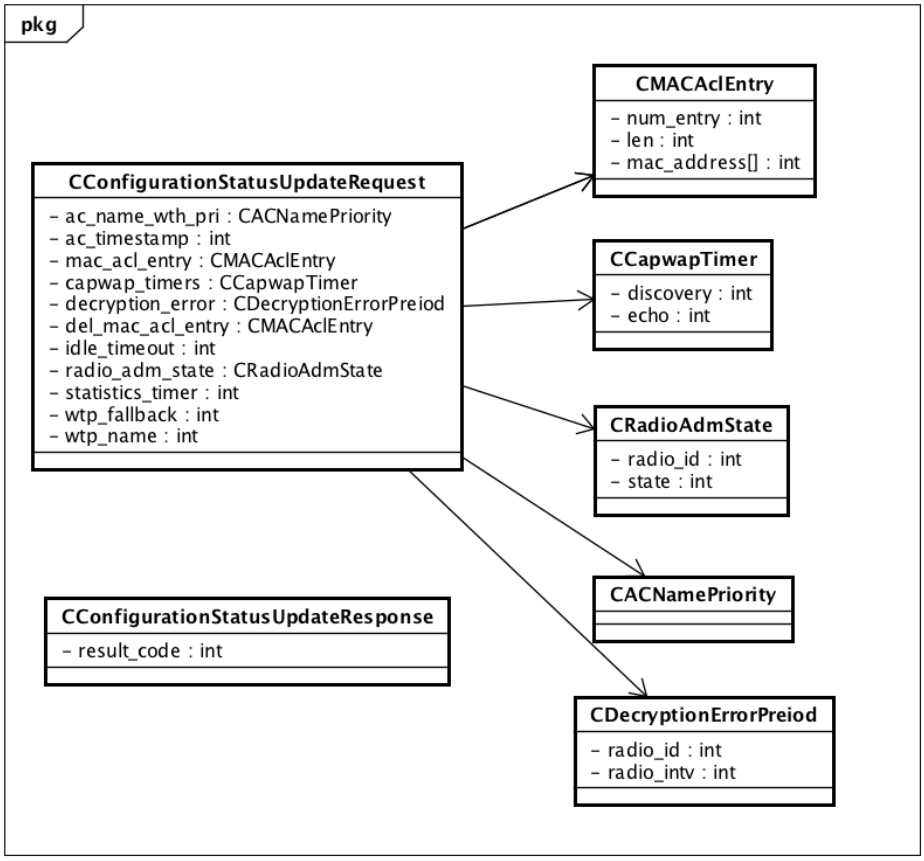
WTP çalışma zamanında yapılandırmasının güncellenmesi gerektiği durumda ise AC, WTP'ye *ConfigurationUpdateRequest* mesajı gönderir. Bu mesaj WTP tarafından alındıktan sonra gerekli yapılandırmaları yaparak AC'ye *ConfigurationUpdateResponse* cevabı gönderir. Şekil 3.15'de sınıf yapısı gösterilmiştir.



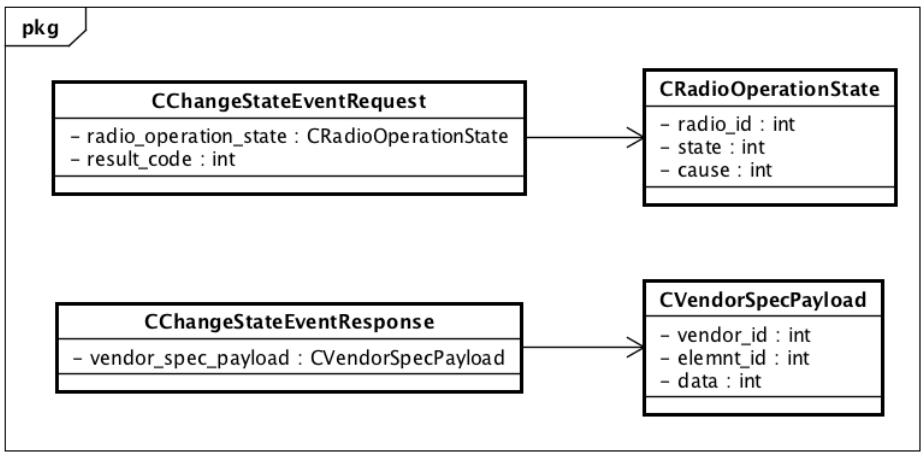
Şekil 3.14: Yapılandırma durum sınıf yapısı.

ChangeStateEventRequest WTP tarafından AC, yapmış olduğu yapılandırmayı doğrulamak için gönderilir veya çalışma zamanında olumsuz bir durum ortaya çıkarsa

bunu AC'ye iletmek için kullanılır. AC tarafından *ChangeStateEventResponse* ile cevaplanır. Sınıf yapısı Şekil 3.16'da gösterilmiştir.



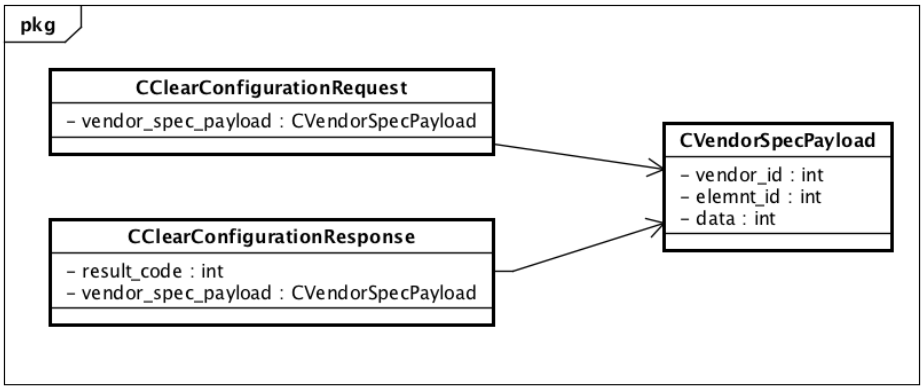
Şekil 3.15: Yapılandırma durum güncelleme sınıf yapısı.



Şekil 3.16: Durum değişikliği sınıf yapısı.

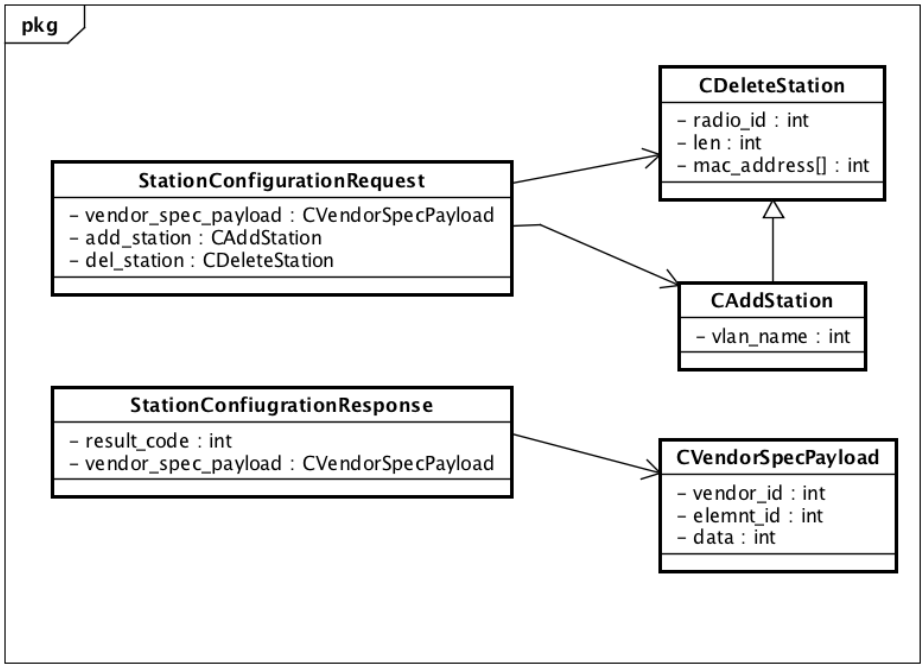
WTP üzerinde yapılandırma sıfırlanmak istenildiğinde AC, *ClearConfigurationRequest* isteğini WTP'ye iletilir. Bu istek WTP'nin çalışma zamanında gönderilir. WTP

yapılandırmasını sıfırladığında karşılık olarak *ClearConfigurationResponse* mesajı gönderir. Bu mesaj türüne ait sınıf yapısı Şekil 3.17'de gösterilmiştir.

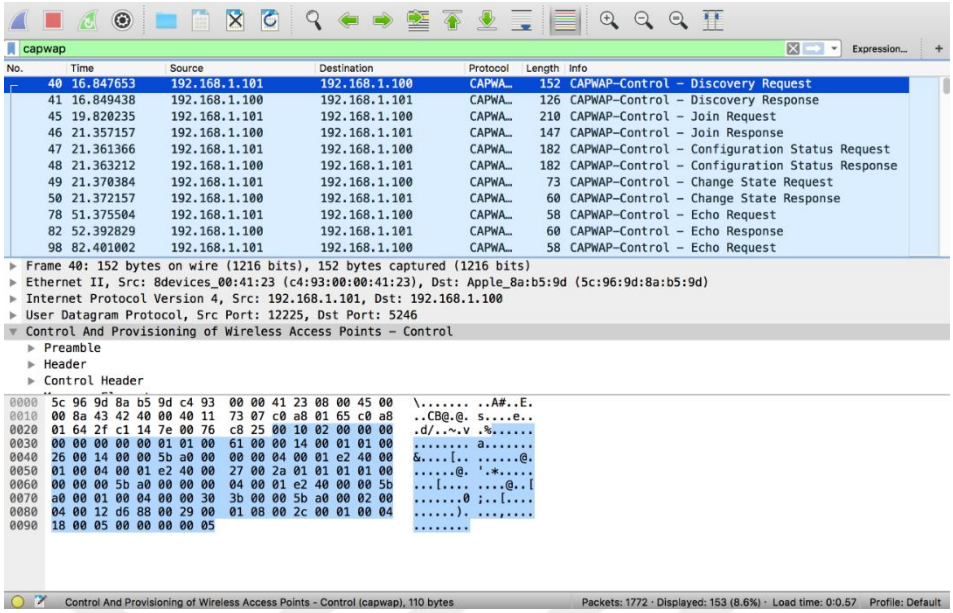


Şekil 3.17: Yapılandırma sıfırlama sınıf yapısı.

AC bir WTP'ye ait istemci yapılandırma işlemlerini *StationConfigurationRequest* mesajı ile iletir. Buna göre AC, bir istemciyi WTP'ye dahil edebilir veya var olan WTP ile olan aktif bağlantısını kesebilir. İşlemlerin WTP tarafından gerçeklemesinin ardından WTP *StationConfigurationResponse* mesajı iletir. İstemci yapılandırma sınıf yapısı Şekil 3.18'de gösterilmiştir.



Şekil 3.18: İstemci yapılandırma sınıf yapısı.



Şekil 3.19: Wireshark ile CAPWAP analizi.

Tablo 3.5: IEEE 802.11 mesaj listesi.

IEEE 802.11 Mesajları
IEEE 802.11 Yeni WLAN Ekle
IEEE 802.11 Anten Bilgisi
IEEE 802.11 WTP BBID
IEEE 802.11 WLAN Sil
IEEE 802.11 Kanal Kontrol
IEEE 802.11 Bilgi
IEEE 802.11 MAC Operasyon
IEEE 802.11 MIC Hata Ölçümleri
IEEE 802.11 Multi-Domain Yetenekler
IEEE 802.11 OFDM Kontrol
IEEE 802.11 Hız Kontrol
IEEE 802.11 RSNA Hata Rapor
IEEE 802.11 İstemci
IEEE 802.11 İstemci Qos Profili
IEEE 802.11 İstemci Oturum Anahtarı
IEEE 802.11 İstatistik
IEEE 802.11 Desteklenen Hızlar
IEEE 802.11 Paket İletim Gücü
IEEE 802.11 Paket İletim Güç Seviyesi
IEEE 802.11 İstemci QoS Güncelle
IEEE 802.11 WLAN Güncelle
IEEE 802.11 WTP QoS
IEEE 802.11 WTP Radyo Yapılandırması
IEEE 802.11 WTP Radyo Hata Alarm
IEEE 802.11 WTP Radyo Bilgisi

CAPWAP protokolü [11] 'de tanımlı olan ve 802.11 ile ilgili mesaj türleri ve yapılandırma işlemlerine (Yeni WLAN oluşturma, QoS güncelleme... vb.) ait mesaj listesi Tablo 3.5'de listelenmiştir.

Geliştirilen Erişim Noktası Yöneticisi, gerçek ortamda testleri yapılarak, ağ protokol analiz aracı olan wireshark[13] ile doğrulamaları yapılmıştır. Şekil 3.19'da görüldüğü üzere AC ile WTP arasındaki iletişim sağlıklı bir şekilde yapılmaktadır. CAPWAP mesajları doğru oluşturulup iletildiğinden, wireshark analiz aracı CAPWAP protokolünü otomatik tanır ve formatlı bir şekilde gösterir. Ayrı mesaj türlerini ve mesajların taşıdığı elemanları gösterebilmektedir.

3.3. ÖNERİLEN IEEE 802.11 YÜK DENGELEME MODELİ

Mevcut IEEE 802.11 standartlarının yük dengeleme ile ilgili herhangi bir bilgi yöntem sunmaması, istemcilerin AP seçimi yaparken yük gözetmeksizin en yüksek RSSI seviyesine göre belirlemesi, ağ üzerinde performans problemlerine yol açmaktadır. [8]. Bunun sonucunda ağda yük problemi ortaya çıkmaktadır. Bir AP yük altındayken, yakınında bulunan başka bir AP daha az bir yük ile çalışabilmektedir. Bu sorunu çözmek ve kullanıcılara adil bir hizmet sunmak için, seçilen bir AP'e bağlı olan kullanıcıların kendi aralarında ve diğer AP'ler arasındaki ilişkinin modellenmesi ve bu model doğrultusunda yük dengeleme problemine çözüm aranması gerekmektedir.

- frekans planlaması ile her bir AP'nin diğer AP'ler ile frekanslarının çakışmasını engellemek,
- kullanıcıların AP'ler ile bağlanma stratejilerini belirleyerek ağ performansını artırmak,
- QoS mekanizması ile ağ üzerinde bazı servislere öncelikler vererek, uygulamaların performansını arttırmaktadır.

Kullanıcılara sunulan hizmet kalitesini, QoS ve EDCA parametreleri artırılmaya çalışılacak olup, kullanıcıların AP'ler arasındaki dağılımını ise, CellBreathing yöntemi ile optimum paket iletim seviyesini seçerek çözmeye çalışacağız. Bu tez çalışması frekans çakışması olmadığını veya çözülmüş olduğunu kabul etmekteyiz.

[13] wireshark, ağ protokol analiz aracı, https://www.wireshark.org, [Ziyaret tarihi: 10 Ekim 2016]

3.3.1. QoS ve Yük Dengeleme Problemi

Bir kablosuz ağda toplam n_{AP} kadar AP ve n_{STA} kadar istemciden oluştuğunu varsayarsak. Bir alandaki tüm aktif istemciler $X = \{s_1, s_2, \dots s_{n_{STA}}\}$ ve $Y = \{a_1, a_2, \dots a_{n_{AP}}\}$ bir alandaki aktif AP'ler olmak üzere, $A = X \rightarrow Y$ bir istemcinin bir AP ile olan bağlantısını ifade eder.

Tüm bağlantıları $V = Y^X$ olarak ele alırsak, birden fazla AP'in kapsama alanına giren ve AP'in güç iletim seviyesini göre AP-STA bağlantısının kontrol edilebilen V kümesine ait bağlantıları da F olarak ifade edebiliriz. CB yöntemi ile bir istemcinin başka bir AP ile olan bağlantısının değiştirilebildiği stratejileri E ile ifade ediyoruz. Burdan yola çıkarak E aşağıdaki şekilde tanımlarız. Benzeri [37], [38] çalışmalarda olduğu gibi AP'ler arasındaki sinyal girişiminin olmadığını veya çözüldüğünü varsayarsak her $A \in E$ için EDCA ile QoS önceliklerini kontrol edebildiğimizi (istemcinin IEEE 802.11e desteklediğini) varsayıyoruz.

Ağırlık maks-min adil hizmet kullanımı için, istemci öncelikleri s_i ile $w_i \, \forall \, i \in \{1,2,..n_{STA}\}$ ile ifade edilebilmektedir. w_i değerinin büyüklüğü istemcinin s_i önceliğinin belirtmektedir.

Yukardaki belirtilen çatıya göre, optimum EDCA-yük dengelemeyi şu şekilde tanımlarız:

- Her AP'nin optimum EDCA ağırlık çözümüne göre yapılandırıldığı durum.
- Minimum ağırlık hızının (weighted rate) farklı bir bağlantı durumunda artmayacağı.

Yukarda belirtilen iki şart EDCA ile yük dengeleme stratejisinde şartlarını açık bir şekilde göstermektedir. Optimum EDCA-yük dengeleme çözümünü hesaplayabilmek için, üç problem göz önüne alınmalıdır:

- Bağlantıları verilen kullanıcıların, EDCA yapılandırması ve veri iletim profilleri, bir model ile trafik çıktısının tahmini.
- Yukarda verilen model ile optimum EDCA yapılandırmasının gerekliliği.
- Verilen bir kümesi ve EDCA yapılandırma bilgisi ile, her bir bağlantının optimum ilişkisinin tanımlanması.

Burada ilk iki maddeye çözüm sunacağız, son madde ise brute-force, branch & bound ve genetik algoritmalarla çözüm sunacağız.

3.3.2. Ağ Doygunluğunda (Saturation) Throughput Modellenmesi

Bir WiFi ağında tüm istasyonların iletmek için sürekli paketlerinin olduğu durumda, ağ doygunluk durumundadır. Bu durum bir ağdaki trafik çıktısını analitik olarak modellemede kullanılmaktadır. Ayrıca, doygunluk durumundaki çıktı (throughput) sınırları belirleyebilmemizde yardımcı olmaktadır. Çalışmalarımızda bu duruma odaklanarak ağda bulunan AP'lerin performansını analiz edebiliriz.

Bir EDCA istasyonu doygunluk durumunda ise, paket gönderim kuyruğu hiçbir zaman boş değildir, buna göre EDCA algoritması Bölüm 2.1.1' de özetlenmiştir. EDCA ile ilgili tüm parametreler Tablo 3.6'da gösterilmiştir.

Tablo 3.6: EDCA parametreleri.

Parametre	Açıklama
AIFS	Çerçeveler Arası Boşluk
CW_{min}	Minimum CW
CW_{max}	Maksimum CW
σ	Slot Zamanı
R	Tekrar İletim Limiti

Diğer çalışmalarda [48], [49] bahsedildiği üzere bahsedildiği üzere $AIFS_i = AIFS = AIFS_{min}$ ve $CW^i = CW^i_{\max} = CW_i \ \forall\, i \in \{1,2,\dots n_{STA}\}$ olacak şekilde ele alınmaktadır. Ayrıca her istemci CW_i değerinin optimizasyon algoritması tarafından sabitlendiğini varsayıyoruz. AP'ler arasındaki sinyal girişimini yok sayarsak AP'lerin kendi içinde performansı bağımsız olarak analiz edilebilir. [48], [49] çalışmalarında kullanılan modelden yola çıkarak;

$$S_{a_k} = \{\, s \in \{s_1, s_2, \dots, s_{n_{STA}}\} \, ve \, A(s) = a_k \,\} \qquad (1)$$

Bir istemci doygunluk durumunda, sabit CW değeri yapılandırılırsa, sabit bir paket iletim olasını olan τ ile paket göndermeye başlar. Bu olasılık, bir kanal slotunun başarılı veya başarısız paket iletim teşebbüslerini ifade etmektedir. Buradan yola çıkarak bir istemcinin paket iletim olasılığı, s_i , CW'nin basit lineer olmayan bir fonksiyonudur.

$$\tau_i = \frac{2}{1 + CW_i} \; \forall i \in \{1,2,\dots,n_{STA}\} \tag{2}$$

Bu bilgiler doğrultusunda, kullanıcı tarafından gönderilen ortalama paket boyutuna L olarak alırsak, kullanıcının ortalama doygunluk anındaki throughput değeri r_i şu şekilde hesaplanabilir;

$$r_i = \frac{P_s^{(i)} L_i}{T_{slot}^k} \tag{3}$$

$P_s^{(i)}$, istemci s_i'nin başarılı paket iletim olasılığını, L_i ortalama paket boyutunu ve T_{slot}^k ise zaman slotunun ortalama değerini belirtmektedir (AP a_k bağlantısında). $P_s^{(i)}$ şu şekilde hesaplanır;

$$P_s^i = \frac{\tau_i}{1 - \tau_i} P_e^{(k)} \tag{4}$$

$P_e^{(k)}$, AP a_k bağlantısında, kanalın dolu olma olasılığını ifade eder ve şu şekilde hesaplanır;

$$P_e^{(k)} = \prod_{j | s_j \in S_{a_k}} (1 - \tau_j) \tag{5}$$

Paket kaybı olmadan T_{slot}^k aşağıdaki varsayımı kabul ediyoruz.

$$m' > m'' \Rightarrow \frac{L_{m'}}{R_{k,m'}^A} \leq \frac{L_{m''}}{R_{k,m''}^A} \; \forall m', m'' \mid s_{m'} \; ve \; s_{m''} \in S_{a_k} \tag{6}$$

AP a_k istemci AP ilişkisindeki, istemci $s_j$tarafından kullanılan fiziksel iletim hızı $R_{k,j}^A$ ile ifade edilmektedir. Zaman slotunun ortalama süresi aşağıdaki formülle hesaplanmaktadır.

$$T_{slot}^{(k)} = P_e^{(k)}\sigma + \mathrm{T}_{\mathrm{busy}}^{(\mathrm{m})}\tau_m + \sum_{j|s_j \in S_{a_k}}^{j>m} T_{busy}^{(j)}\tau_j \prod_{l\,|\,s_l \in S_{a_k}}^{l<j}(1-\tau_l) \quad (7)$$

$$T_{busy}^{(j)} = T_{overhead} + L_j/R_{k,j}^{A} \quad (8)$$

Fiziksel preamble, *AIFS* ve MAC katmanındaki ACK paketlerinden oluşan fazlalıklar T_{overhead} ile ifade edilmektedir. En az slot işgal eden istemcinin indeksini m olarak ele aldığımızda, $m = \min(\{i\ ve\ s_i \in S_{a_k}\}) \Leftrightarrow m = argmin_{j|s_j \in S_{a_k}}\left\{\frac{L_j}{R_{k,j}}\right\}$. 7 nolu denkleme göre, farklı fiziksel veri iletim oranına sahip, farklı paket boyutundaki uygulamalar için doğru hesapladığını göstermektedir. Ortaya konulan modelin farklı bağlantılara sahip istemcilerin, farklı şartlar altındaki sahip oldukları iletişim şartlarına göre paket gönderme olasılıkları, modelin doğru çalışması açısından önem arz etmektedir.

3.3.3. Optimum EDCA Yapılandırması

Optimum EDCA yapılandırmasının yapılabilmesi için CW değerinin her bir istemci için en uygun değerin belirlenerek, kullanıcıların yapılandırılması gerekmektedir. *A* bağlantısında, AP'ler arası sinyal girişiminin olmadığını varsayarsak, problem AP bazında çözülebilir. AP a_k odaklanıldığında ve Formül 3'e göre, optimum maks-min ağırlık çözüme ulaşıldığında, AP a_k bağlı tüm istemciler aynı ağırlıklı oranda veri iletimi yaparlar. Bu durumda;

$$\frac{r_n}{w_n} = \frac{r_m}{w_m} => \frac{\tau_n(1-\tau_m)}{\tau_m(1-\tau_n)} = \frac{w_m'}{w_n'} \quad (9)$$

$\forall n, m$ öyleki s_n ve $s_m \in G_{a_k}, w_m' = w_m/L_n$. Burdan yola çıkarak 9. denklemi yeniden yazarsak;

$$\tau_m = \frac{w_m'\tau_n}{w_n' + \tau_n(w_m' - w_n')} \quad (10)$$

10 nolu denklemde τ_m istemci s_m'nin paket iletim olasılığının, istemcinin CW yapılandırmasına bağlı olduğu görülmektedir $CW_m = int\{2/\tau_m - 1\}$.

Optimum AP yapılandırması, AP'nin r_n ağırlıklı oranlarının maksimize edilmesine yani $\tau_n = 2\ /(1 + CW_n)$ istemcinin CW değerine bağlıdır.

3, 7 ve 10 numaralı denklemlerden r_n , CW_n'nin fonksiyonu olarak yazılabilir. ve bu değerinin maksimum olması optimum EDCA yapılandırmasını vermektedir.

$$G_k(CW_n) = r_n = \frac{P_s^{(n)} L_n}{T_{slot}^{(k)}} \ , a_k = A(s_n) \tag{11}$$

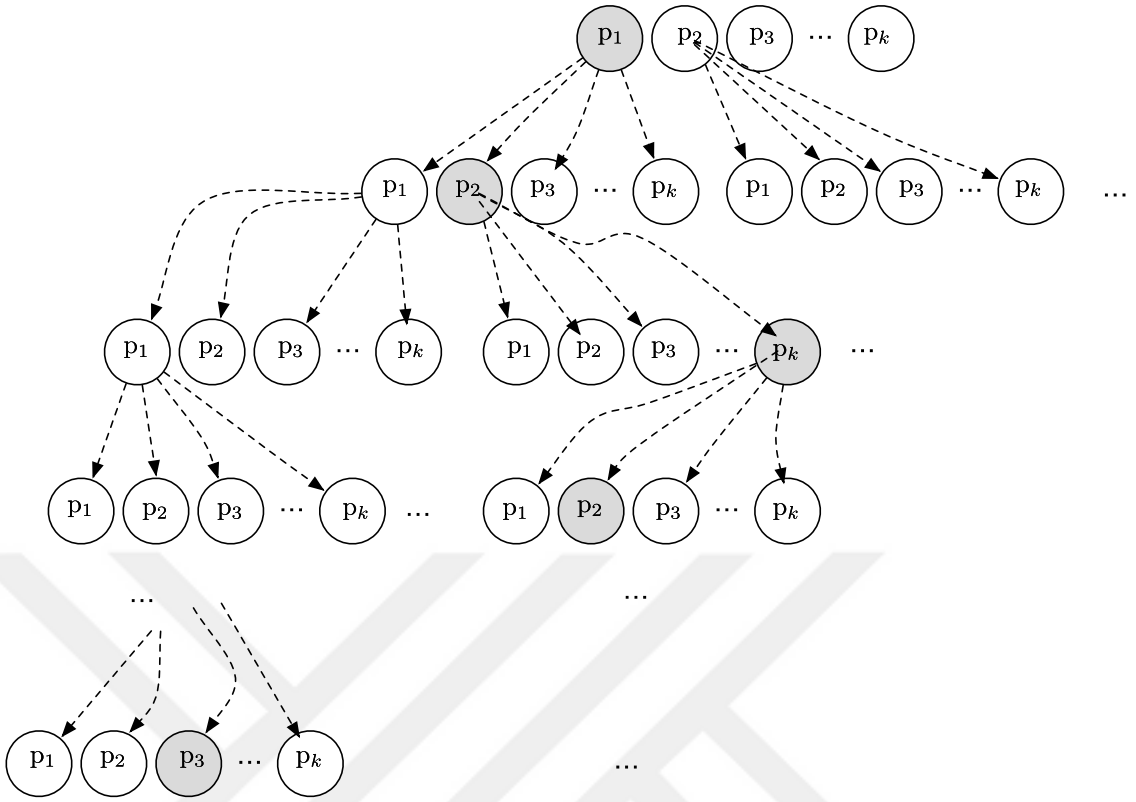
Diğer bir deyişle, optimum EDCA yapılandırması, AP a_k bağlı kullanıcıların maksimum $G_k(CW_n)$ değerine bağlıdır. Bu analiz her bir AP için uyguladığımızda ağ içerisindeki A optimum EDCA yapılandırmasını elde ederiz [50], [54].

3.4. ÖNERİLEN IEEE 802.11 YÜK DENGELEME ALGORİTMALARI

En iyi EDCA-LB (EDCA Yük Dengeleme) çözümü, minimum $r_n\ / w_n$ değerini veren en iyi AP-STA bağlantısına bağlıdır. Bu sebeple en iyi AP-STA bağlantısını bulabilmemiz için Brute-Force, Branch & Bound ve Genetik Algoritma üzerinde çalışılarak çözüm önerileri sunulmuştur.

Bölüm 3.3.1'de sunulan bir hotspot ele alındığında, her bir AP başlangıç noktasında sabit bir güç seviyesiyle (power level) yapılandırıldığını varsayıyoruz. Toplam AP sayısı M olarak ele alır ve paket iletim güç seviyeleri $P = \{P_1, P_2, \ldots, P_K\}$ kümesinde $P_i < P_{i+1}$ şartını sağlayacak şekilde güç seviyelerinin yapılandırıldığını varsayıyoruz. AP, a_i tarafından kullanılan güç seviyesinin $p_i \in P$, olarak ifade edersek, istemci $s_i$ve $a_i$arasındaki mesafe $d_{i,j}$ olarak gösterilebilir. Buna göre kablosuz bir ağ üzerinde optimum güç seviyesini bulabilmek Şekil 3.20'de belirtilen bir ağaç üzerinde arama işlemi yapılması gerekmektedir.

Standart bir ağda istemcilerin (STAs), AP ile olan bağlantıları paket iletim güç seviyesi ile AP-STA arasındaki mesafeye bağlıdır. Verilen bir bağlantının performansı önceki bölümlerde ifade edilen model ile hesaplanarak, optimum güç seviyesi ile AP'ler yapılandırılır. Böylelikle CB yöntemiyle ağdaki iyileştirmeler gerçekleştirilebilir. Optimizasyon fonksiyonu, toplam çıktıyı veya ağ içerisindeki en düşük çıktıyı maksimize ederek hizmet kalitesini artırmak mümkündür.



Şekil 3.20: AP'ler arası paket iletim güç seviyesi arama ağacı.

3.4.1. Önerilen Brute-Force Algoritması

Brute-Force algoritması bir arama ağacındaki olası tüm değerlerin taranması ile çözüm sunan bir yöntemdir. Bu yöntemin uygulanabilirliği AP ve güç seviyesinin sayısına bağlı olarak değişmektedir. Üç adet AP'den oluşan küçük bir ağ düşünüldüğünde ve yapılandırılabilir güç seviyelerinin beşi geçmediği bir hotspotlar da uygulanması mümkündür.

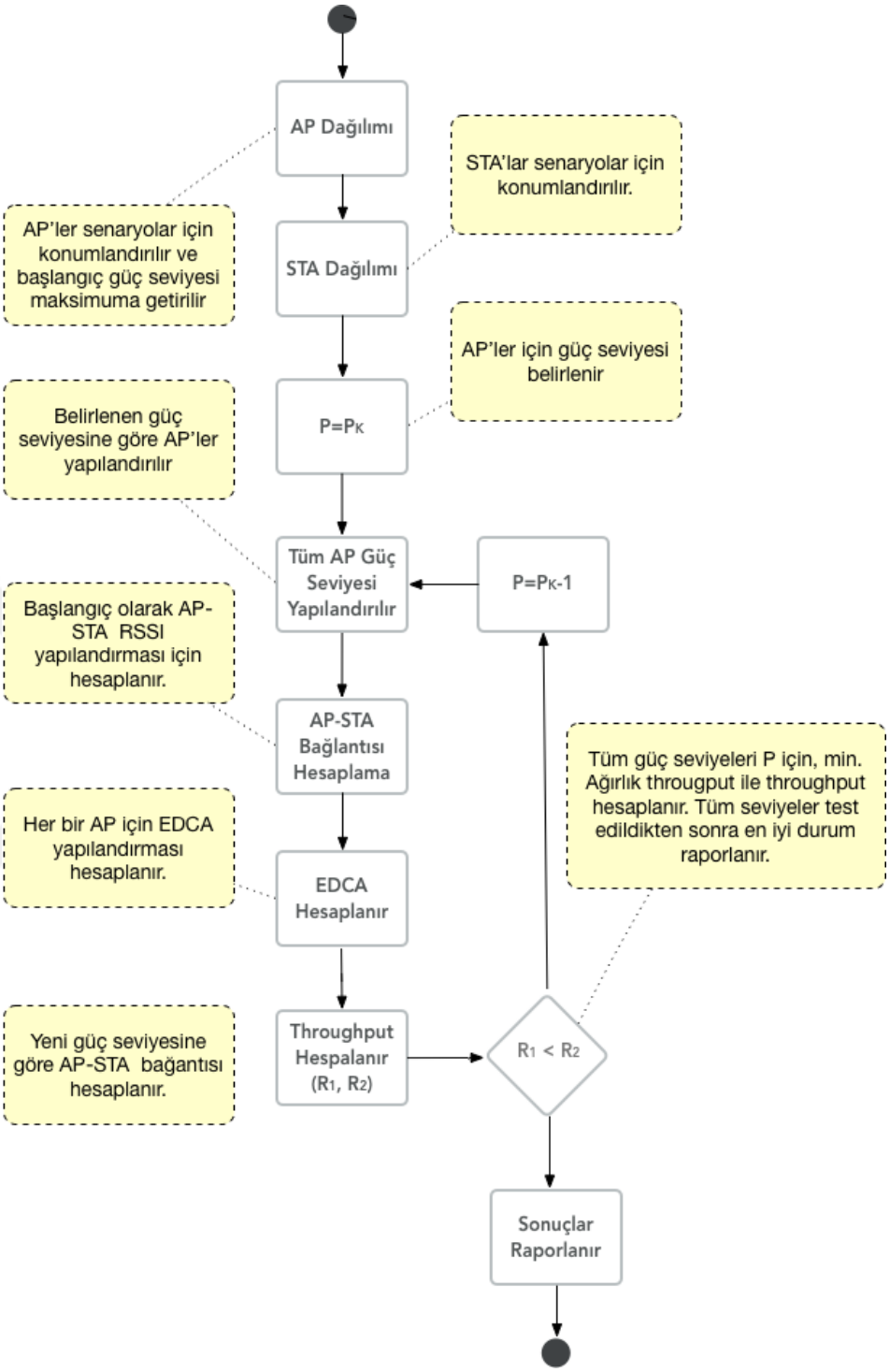
Brute-Force algoritmasının akış şeması Şekil 3.21'de gösterilmiş olup algoritma tüm test senaryoları için şu şekilde çalışmaktadır;

1. Belirtilen senaryo ve yapılandırma doğrultusunda AP'ler alana dağıtılır.
2. Kullanıcılar alana belirtilen kurguda rastgele yerleştirilerek yapılandırmaları (paket boyutu, ağırlık … vb.) yapılır.
3. Güç seviyeleri belirlenerek, AP'ler bu seviyelere göre yapılandırılır.
4. AP-STA bağlantısı hesaplanır.
5. EDCA yapılandırması hesaplanır.

6. Throughput hesaplanarak değerlendirilir. Elde edilen değer değerlendirilerek güç seviyeleri yapılandırılarak süreç 3. adımdan itibaren tekrarlanır veya sonuçlar raporlanır.

Tablo 3.7: Brute-Force algoritması.

```
1:  function BruteForce                 // Algoritma başlangıç noktası
2:    ComputeAssociations(RSSI) // İlk önce standart yöntemler ile bağlantılar hesaplanır
3:    ComputeBruteFoce() // Brute-Force yöntemi ile optimum çözüm bulunmaya çalışılır
4:  end function
5:  function ComputeBruteFoce // Brute-Force ile güç sevilerinin taranır
6:    r_result = []
7:    for each AP
8:       for each power_levels
9:          ComputeAssociations(RSSI)
10:          ComputeMaxMinFairness()
11:          r_out = [r_out STA.R/STA.w]
12:          r_out = min(r_out)
13:          result = r_out : pw
14:    end for
15:  end function
16:  function ComputeAssociations // RSSI seviyesine göre STA-AP bağlantısı hesaplanır
17:    k = 0
18:    for each STA
19:       t = max(AP_tx_pow)
20:       k = find(t)
21:       STA.AP = k
22:    end
23:  end function
24:  function ComputeMaxMinFairness // AP içerisinde EDCA yapılandırması hesaplanır
25:    for each STA
26:       w = [w STA.w]
27:       l = [l STA.l]
28:       r = [r STA.r]
29:    end for
30:    for each allCWs
31:       tau = 2/(CW+1)
32:       tau = w(2:end) * tau(1) ./w(1) + tau(1) * (w(2:end)-w(1))
33:       W = round((2./tau − 1))
34:       t=2./(W +1)
35:       R = ComputeRates()  // CW değeri W için istemcilerin veri transfer hızı hesaplanır
36:       temp = min((R. ∗ l./w))
37:       if temp > R2 then
38:          return R2
39:       end if
40:    end for
41:  end function
```



Şekil 3.21: Brute-Force algoritması akış şeması.

3.4.2. Önerilen Genetik Algoritma (GA)

Sezgisel (Heuristik) yöntemler yerel veya global optimumu garanti etmezler fakat, genelde optimum veya optimuma yakın bir sonuç sunarlar. Genetik algoritma ilk defa

1975'te doğadan ilham alınarak, bilgisayar bilimlerinin birçok alanında kullanılmaya başlanmıştır [50].

- Doğal genetik yöntemlerde bulunan kodlama mantığını kullanması
- Arama uzayını popülasyonlar olarak kullanması
- Olasılığa dayalı olması

Bu çalışmada AP yapılandırmasında en iyi güç iletim (transmission power) seviyesinin tespit edebilmek adına GA kullanıştır. Güç seviyeleri kromozom olarak modellenerek çaprazlama ve mutasyon işlemleri yapılmaktadır. Şekil 3.20'da gösterilmiş olan arama ağacı tüm alt düğümleri birlikte aranması yerine GA ile optimum veya optimuma yakın çözümler için taranmaktadır.

Bir AP'e ait throughput, istemciler tarafından gönderilen başarılı paketlerin toplamı ile ölçülebilir. Bu durumda bir istemci s_j tarafından gözlemlenen toplam throughputr_j Denklem 11'de ifade edilmiştir. Bu durumda AP tarafından ölçülen toplam throughput AP-STA bağlantısındaki toplam maksimum veri iletim hızları olarak ele alınabilir t_{ap}.

$$t_{ap} = \sum_{j=1}^{n} r_j \tag{12}$$

Bu çalışmadaki baz aldığımız optimizasyon (cost function) değeri, hostpot üzerindeki toplam throughput t_{net} maksimize etmeye dayanmaktadır.

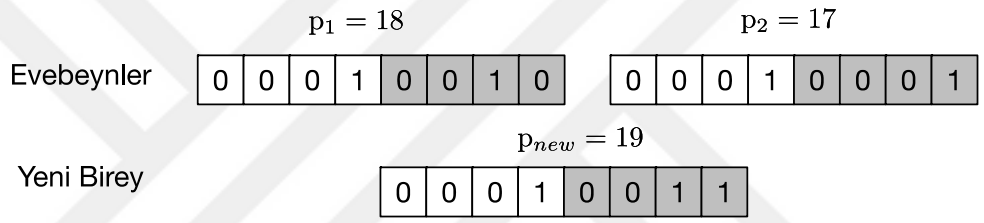
$$t_{net} = \sum_{i=1}^{m} t_{ap_i} \tag{13}$$

Yukarıda bahsedilen problemde AP sayısı arttıkça algoritmanın kompleksliği üstel olarak artmaktadır. GA yöntemi ile muhtemel en iyi çözüm kümesi aranarak sonuç elde edilebilir.

Kromozomlar kodlaması ve modellenmesi GA'nın birinci adımını oluşturmaktadır. Birçok yöntem olmasına rağmen, en yaygın kodlama yöntemi ikili (binary) kodlamadır. İkili kodlama ile sayısal değerler kolay bir şekilde işlem yapılabilmesini sağlamaktadır.

Bu çalışmada bir AP'nin a_i güç seviyesi, bir tam sayı olarak tutulmaktadır ve değer ikili metin olarak kodlanarak çaprazlama ve mutasyon işlemlerinde kullanılmaktadır. Çaprazlama yöntemi ebeveyn kromozomları kullanarak yeni bireyler üretilir. Birçok çaprazlama yöntemi olmakla birlikte, bu çalışmada tek bir noktadan çaprazlama (single point crossover) yöntemi kullanılmıştır. 14. denkleminde gösterildiği üzere c_l yeni bireyin, ebeveynler ise p_i ve p_j olarak gösterilmiştir. Güç iletim seviyeleri bir sınır aralığında olduğu için maskeleme yöntemi kullanılmıştır. Şekil 3.22'de örnek bir çaprazlama ve elde edilen yeni veri gösterilmiştir.

$$c_l = m \,\&\, p_i + m \,\&\, p_j \tag{14}$$



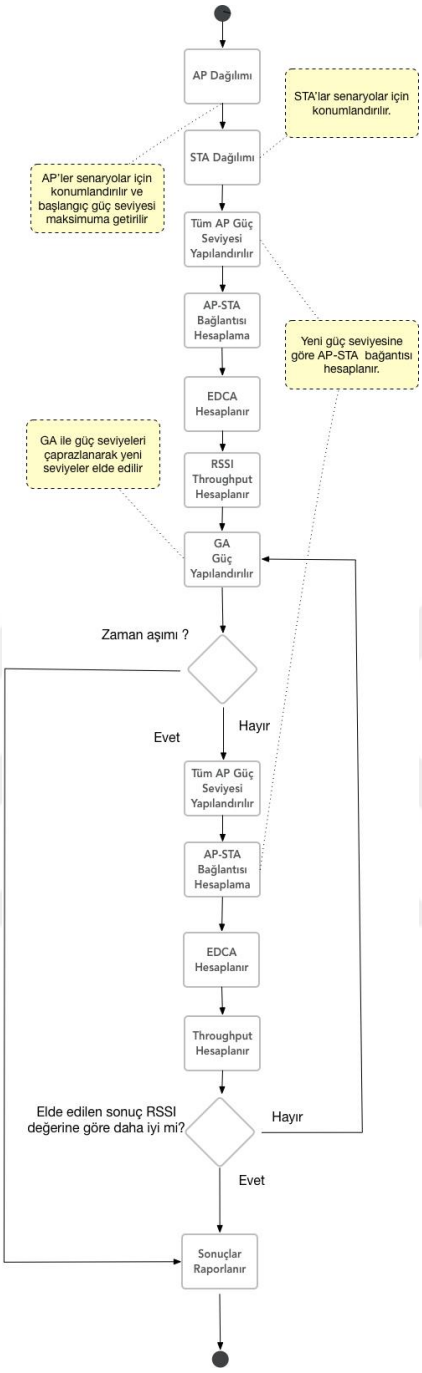
Şekil 3.22: AP güç seviyesi çaprazlama örneği.

Mutasyon ise GA'da kromozom bitlerinin rastgele olarak değiştirilmesi işlemidir. Bu yöntem ile yeni türler elde edilmesi hedeflenir. Bu çalışmada Gaussian dağılımı kullanılarak rastgele sayı ikilileri belirlenerek mutasyon sağlanmıştır [51]. Şekil 3.23'de Genetik Algoritma akış şeması verilmiştir;

1. Belirtilen senaryo ve yapılandırma doğrultusunda AP'ler alana dağıtılır.
2. Kullanıcılar alana belirtilen kurguda rastgele yerleştirilerek yapılandırmaları (paket boyutu, ağırlık… vb.) yapılır.
3. Güç seviyeleri belirlenerek, AP'ler bu seviyelere göre yapılandırılır.
4. AP-STA bağlantısı ve EDCA yapılandırması hesaplanır.
5. RSSI güç seviyesine göre throughput hesaplanır.
6. GA ile başlangıç değerlerine göre yeni güç seviyeleri belirlenir.
7. GA güç seviyesine göre AP-STA bağlantıları ve EDCA yapılandırması yapılır.
8. Yeni throughput hesaplanarak RSSI değeri ile karşılaştırılır. Sonuçlar daha kötüyse güç seviyeleri çaprazlanarak 6. adımdan tekrar başlatılır. Sonuçların daha iyi ise çıktılar raporlanır.

Tablo 3.8: Genetik algoritma.

```
1:  // GA Algoritma Başlatılır
2:  function GA
3:     ComputeAssociations(RSSI) // RSSI değerine göre bağlantılar hesaplanır
4:     ComputeGASolution()       //GA çözümü başlatılır
5:  end function
6:  function ComputeAssociations // RSSI değerine göre AP-STA bağlantısı hesaplanır
7:     k = 0
8:     for each STA
9:        t = max(AP_tx_pow)        // Max RSSI değerine alan istemci AP bağlantısı bulunur
10:       k = find(t)
11:       STA.AP = k
12:    end
13: end function
14: function ComputeMaxMinFairness // Bir AP için EDCA yapılandırması hesaplanır
15:    for each STA
16:       w = [w STA.w]
17:       l = [l STA.l]
18:       r = [r STA.r]
19:    end for
20:    for each allCWs
21:       tau = 2/(CW+1)
22:       tau = w(2:end) * tau(1) ./w(1) + tau(1) * (w(2:end)-w(1))
23:       W = round((2./tau − 1))
24:       t=2./(W +1)
25:       R = ComputeRates()
26:       temp = min((R. ∗1./w))
27:       if temp > R2 then
28:          return R2
29:       end if
30:    end for
31: end function
32: function ComputeGASolution          // GA ile optimum güç seviyesi bulma
33:    AP.P = Pmax                      // Tüm AP'ler güç seviyesi sabitlenir
34:    ComputeMaxMinFairness(RSSI)
35:    R_RSSI = ComputeNetThrougput // Throughput RSSI için hesaplanır
36:    while R_GA < R_RSSI
37:       for each ap = AP
38:          p = CrossPowerLevels(AP.P) // Güç seviyeleri çaprazlanır
39:          AP.P = p
40:       end
41:       ComputeAssociations(GA)      // AP-STA hesaplanır
42:       ComputeMaxMinFairness(GA)
43:       R_GA = ComputeNetThrougput
44:       if time_out
45:          exit()
46:       end if
47:    end while
48: end function
```



Şekil 3.23: Genetik Algoritma akış şeması.

3.4.3. Önerilen Branch & Bound (B&B) Algoritması

Branch & Bound (B&B) optimizasyon problemlerinde kullanılan ve uzay ağacını taranmasına sağlayan bir yöntemdir [52]. BB algoritması, ağ içerisinde, ağırlıklı veri iletim hızı en küçük $t_m = min(\frac{r_i}{w_i})$ olan istemcinin hızını maksimize etmeyi sağlayacak şekilde tarama yapmaktadır.

M adet AP'den oluşan bir ağ içerisindeki en düşük min. ağırlıklı throughput değerini veren t_m maksimize edecek güç seviyeleri $P = [P_1, P_2, \ldots, P_K]$, optimum güç seviyesini belirler;

$$p = \ argmin_{\beta \in P^M} \{t_m(\beta)\} \tag{15}$$

Yukarda bahsedilen problemin gerçek çözümü AP sayısı ile üstel olarak artmaktadır ($O\ (K^M)$). Fakat, B&B algoritmasını kullanarak çözüm kümesini bulabiliriz.

Optimum paket iletim gücünü bulmak için üç adımlık bir algoritma izlenmektedir:

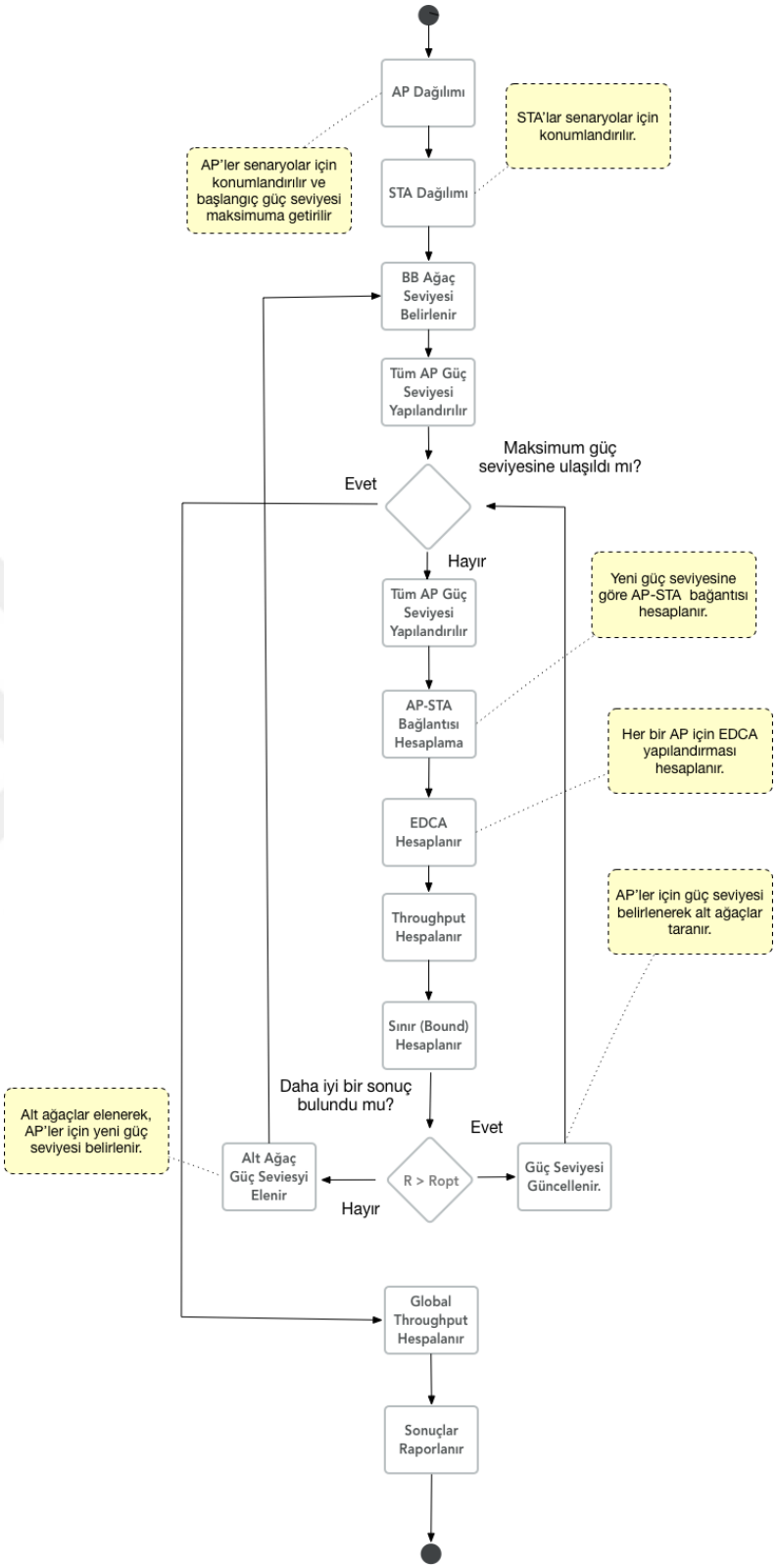
- problem alt problemlere bölünür
- her bir alt problem için sınırlar hesaplanır
- sınır altındaki alt problemler elimine edilerek çözüm kümesi daraltılır.

Öncelikle birince AP a_1 sabitlenerek, alt problemleri oluşturacak olan diğer AP güç seviyeleri belirlenir. Bu şekilde K kadar alt problem elde etmiş oluruz. Her bir alt problem için, maksimum throughput değerini verecek olan paket iletim gücü seviyesi bazında sınır $t_{\max}^{(i)}$ hesaplanır. Bu sınır, birbirinden AP'lerin farklı güç iletim seviyesinde paket iletimi yapmasıyla birlikte, istemcilerin farklı AP'lere bağlanmasına bağlı olarak değişecektir. Diğer bütün AP'ler için gücün maksimum olması, seçili AP'ye bağlı olan kullanıcı sayısının en aza inmesi demektir. Bu durumda seçili AP için minimum ağırlık çıktının maksimize edilmesi demektir. Elde ettiğimiz bu sınır ile alt çözüm kümeleri elimine edilerek nihai çözüm elde edilir.

B & B algoritmasının psedue kod olarak yazılmış hali Tablo 3.9'da gösterilmiştir. Şekil 3.24'de ise algoritmanın akış şeması verilmiştir. Buna göre algoritmanın çalışma şekli özetlenecek olursa;

1. Belirtilen senaryo ve yapılandırma doğrultusunda AP'ler alana dağıtılır.
2. Kullanıcılar alana belirtilen kurguda rastgele yerleştirilerek yapılandırmaları (paket boyutu, ağırlık… vb.) yapılır.
3. Tüm AP'ler için güç seviyeleri maksimum olacak şekilde belirlenir ve AP'ler bu seviyelere göre yapılandırılır.
4. AP-STA bağlantısı hesaplanır.

5. EDCA yapılandırması hesaplanır.
6. Throughput hesaplanarak sonuçlar değerlendirilir.
7. Eğer daha iyi bir sonuç elde edildiyse, BB arama ağacı alt ağaçlardan devam eder.
8. Eğer elde edilen sonuç daha kötüyse alt ağaçlar hesaplanmadan elenir.
9. BB yeni güç seviyesine göre yeni AP-STA bağlantıları hesaplanır.
10. Olası tüm güç seviyeleri dolaşıldığında, genel çözüm elde edilir.



Şekil 3.24: BB algoritması akış şeması.

Tablo 3.9: Branch & Bound algoritması.

```
1:  // BB Algoritmasını Başlatılır
2:  function BrancAndBound
3:     ComputeAssociations(RSSI) // RSSI değerine göre bağlantılar hesaplanır
4:     ComputeBBSolution()          //BB çözümü aranmaya başlanır
5:  end function
6:  function ComputeAssociations // RSSI değerine göre AP-STA bağlantısı hesaplanır
7:     k = 0
8:     for each STA
9:        t = max(AP_tx_pow)        // Max RSSI değerine alan istemci AP bağlantısı bulunur
10:       k = find(t)
11:       STA.AP = k
12:    end
13: end function
14: function ComputeMaxMinFairness // Bir AP için EDCA yapılandırması hesaplanır
15:    for each STA
16:       w = [w STA.w]
17:       l = [l STA.l]
18:       r = [r STA.r]
19:    end for
20:    for each allCWs
21:       tau = 2/(CW+1)
22:       tau = w(2:end) * tau(1) ./w(1) + tau(1) * (w(2:end)-w(1))
23:       W = round((2./tau − 1))
24:       t=2./(W +1)
25:       R = ComputeRates()
26:       temp = min((R. ∗l./w))
27:       if temp > R2 then
28:          return R2
29:       end if
30:    end for
31: end function
32: function ComputeBBSolution          // BB ile güç seviyelerinin tarandığı fonksiyon
33:    if max_pow_level is reached then
34:       rsol = ComputeGlobalSolution(AP, STA)
35:       if rsol > ropt then
36:          ropt = rsol
37:          popt = p
38:       end if
39:       else
40:          rb= ComputeBound(AP, STA, level)
41:          if rb solution is better traverse other power levels then
42:             for each power_levels
43:                ComputeBBSolution(AP, ST As, level)
44:             end for
45:          end if
46:       end if
47: end function
48: function ComputeBound
```

Tablo 3.9 (devam): Branch & Bound algoritması.

```
49:     AP (k).P = p(k)
50:     pow= max(power_levels)
51:     for each k = idx
52:         AP(k).P = pow
53:     end for
54:     ComputeAssociations()
55:     ComputeMaxMinFairness(AP, STA)
56: end function
57: function ComputeGlobalSolution
58:     ComputeMaxMinFairness(AP, STA)
59:     for each k = idx
60:         rout=[rout STA.R/STA.w]
61:     end for
62:     rout = min(rout)
63: end function
```

4. BULGULAR

Yapılan çalışmalar sonucunda, IEEE 802.11 QoS yük dengeleme algoritmaları üzerinde çalışılmış olup, algoritmalar öncelikle simülasyon ortamında test edilmiştir. Elde edilen sonuçlar sunulan yöntemlerin uygulanabilirliği kanıtlandıktan sonra geliştirilen Erişim Noktası Geliştirici ile bir TestBed ortamı oluşturulmuştur. Testbed ortamında elde edilen sonuçlar değerlendirilmiştir.

Analiz işlemini özel geliştirilmiş bir EDCA simülasyon ile frekans çakışması olmayan AP'ler ile test edilmiştir. Simülatör tarafından uygulanan simülasyon parametreleri Tablo 4.1'de özetlenmiştir.

Tüm simülasyon çalışmaları Matlab ortamında geliştirilmiş olup, testler Intel i5 destekli bir Apple Macbook Pro üzerinde sadece bir işlemci kullanılarak yapılmıştır.

Tablo 4.1: Model ve simülasyon parametreleri.

Parametre	Açıklama	Değer
AIFS	Çerçeveler Arası Boşluk	500us
SIFS	Kısa Boşluk	10us
T_{ACK}	ACK Çerçevesi için Harcanan Zaman	304us
T_{PLCP}	Fiziksek Katman Başlık Zamanı	192us
σ	Dilim Zamanı	20us
$T_{overhead}$	$SIFS+T_{ACK}+AIFS+T_{PLCP}$	556us

4.1. BRUTE-FORCE ALGORİTMASI

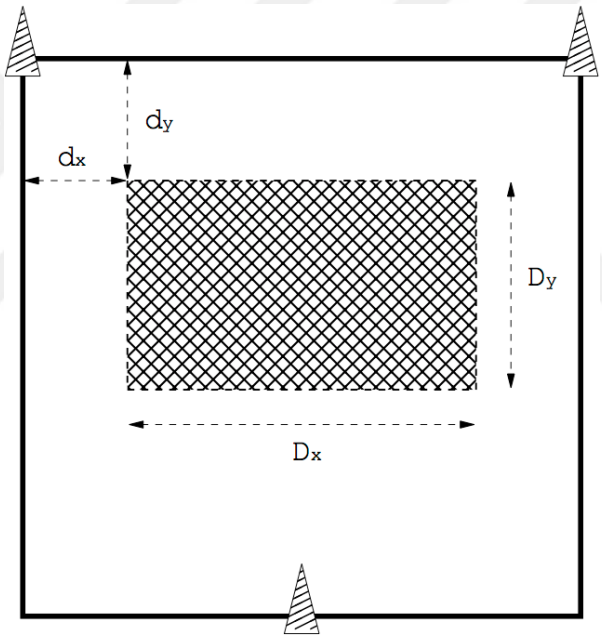
Simülasyon senaryosunda üç AP 100m x 100m'lik bir alanda Şekil 4.1'de olduğu gibi farklı kullanıcıların farklı QoS trafik ihtiyaçlarına göre simülasyonu yapılmıştır. Simülasyon iki farklı bağlantı türü için değerlendirilmiştir:

- RSSI ile EDCA maksimum-minimum ağırlıklı adil trafik kullanımı. Bu stratejide bütün AP'ler sabit eşit güç değerlerinde iletim yaparlar P_{tx} . Bağlantının kurulmasında ve fiziksel veri transfer oranlarında, Şekil 4.2'deki bilgiler baz alınmıştır[14].Her istemci en yakınındaki AP'ye bağlandığına göre fiziksel veri iletim oranı d uzaklık mesafesine bağlı olarak değişmektedir. Mesafeye göre veri

[14] Cisco, Kanallar, Güç Seviyeleri ve Anten Kazanımı, http://www.cisco.com/c/en/us/td/docs/wireless/access_point/1200/vxworks/configuration/guide/ap120scg/bkscgaxa.pdf, [Ziyaret tarihi: 10 Ekim 2016]

iletim hızları; $d \leq 50m\ ise\ 11\ Mbps,\ 50m < d \leq 65m\ ise\ 5.5Mbps,$ $65m < d \leq 80m\ ise\ 2\ Mbps,\ 80m < d \leq 125m\ ise\ 1Mbps$

- Optimum CB ile EDCA (Opt. CB), bu durumda ağ yöneticisinin kullanıcının konumlarını bildiğini varsayıyoruz. Bu durumda AP'lerin iletim gücü minimum ağırlıklı iletim hızını maksimuma çıkartmak için AP'nin yapılandırması seçilir. AP'nin iletim gücü, $\{0.00P_{tx}, 0.25P_{tx}, P_{tx}, 1.5\ P_{tx}, 2.5\ P_{tx}, 5\ P_{tx}\}$ değeri RSSI ile bağlantı kurulurken kullanılan güç değerleridir. Seçilen güç değerlerine göre ve sinyalin yayılmasına göre, AP'lerin kapsama alanları daraltıldı veya genişletildi (16. denklem). Sonuç olarak, Opt. CB ile her bir istemci AP ile daha iyi veri hızı için bağlantısını kurmuştur.

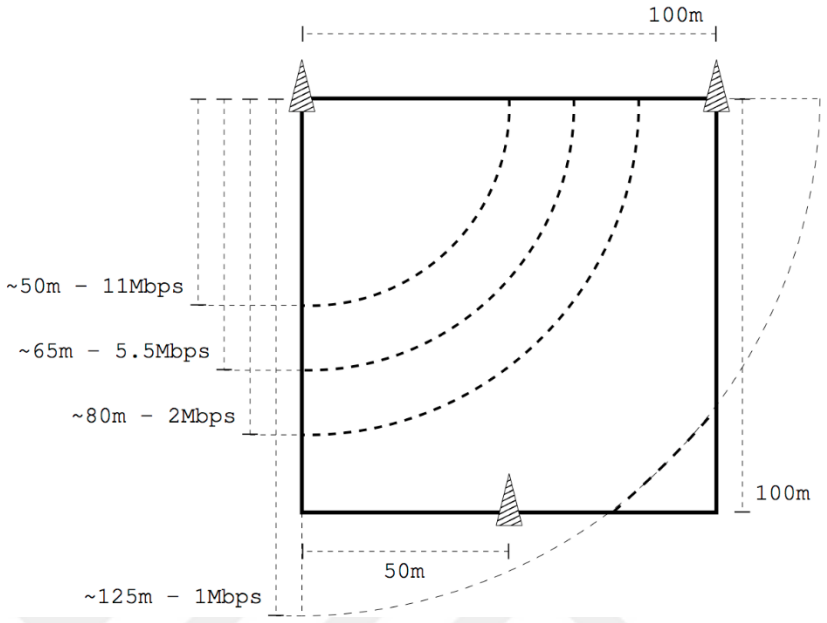


Şekil 4.1: Brute-Force simülasyonu AP-STA dağılım alanı.

$$[50, 65, 80, 125] \rightarrow [50\alpha, 65\alpha, 80\alpha, 125\alpha\ ], \alpha = \sqrt{\bar{P}\ /P_t} \qquad (16)$$

İki farklı bağlantı stratejisi Tablo 3.6'deki değerlere göre Şekil 4.1'de gösterildiği alanda farklı kullanıcı dağılımlarına göre analiz edilmiştir. Her kullanıcı dağılımı 20-100 arasında artan sayılardaki yoğunluklarda test edilmiştir. Her bir istemci rasgele QoS

ağırlık $\{1, 2,3,4\}$ değerleri ile rastgele seçilen paket $\{500, 1000, 1500\}$ byte paket boyutları ile test edilmiştir. Paket boyutları UDP, IP ve MAC başlıklarını içermektedir.



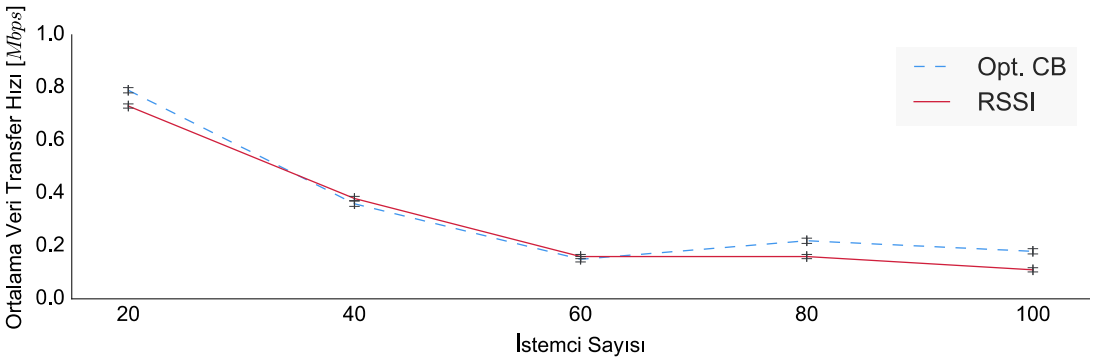
Şekil 4.2: Brute-Force simülasyon güç veri iletim hızları.

Tablo 4.2: İstemcin senaryolara göre dağılımı.

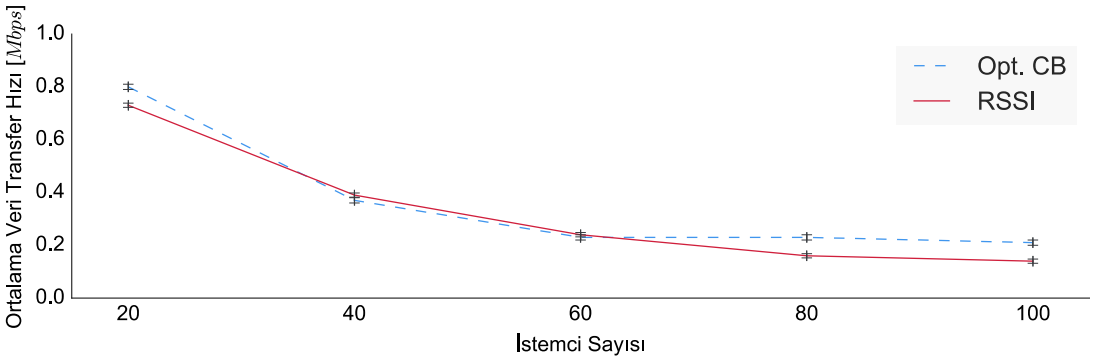
	dx	dy	Dx	Dy
Senaryo 1	0m	0m	100m	100m
Senaryo 2	0m	0m	50m	50m
Senaryo 3	25m	25m	50m	50m

Elde edilen ortalama throughput değerleri Şekil 4.3, Şekil 4.4 ve Şekil 4.5'de, Şekil 4.6, Şekil 4.7 ve Şekil 4.8'de ise minimum ağırlık hızlar gösterilmiştir. Bu çıktılardaki simülasyon sonuçları 50 farklı test denemesinin sonucunda elde edilmiştir.

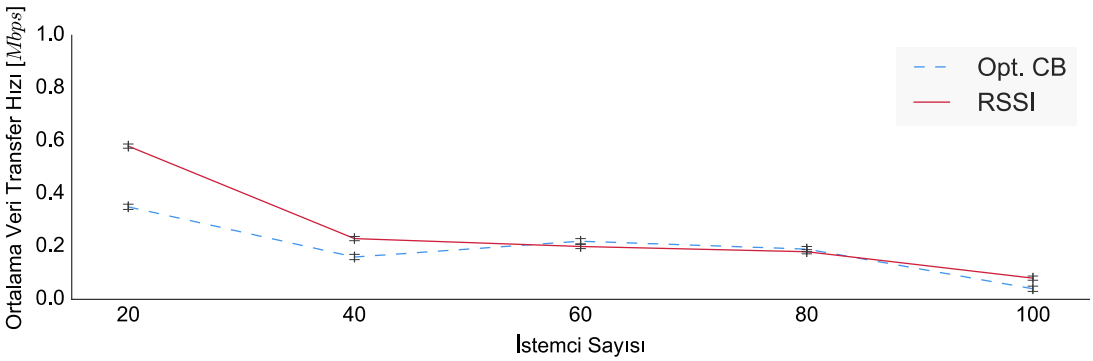
İki farklı bağlantı stratejisi, Tablo 3.6'de verilen üç farklı kullanıcı dağılım senaryosu ile test edilmiştir. CB algoritmasını kullanılarak yapılan LB çalışması Optimum CB (Opt. CB) olarak grafiklerde gösterilmektedir. Opt. CB yöntemini, RSSI ile karşılaştırıldığında hem ortalama değer hem de minimum ağırlıklı throughput için daha iyi sonuçlar verdiğini görmekteyiz.



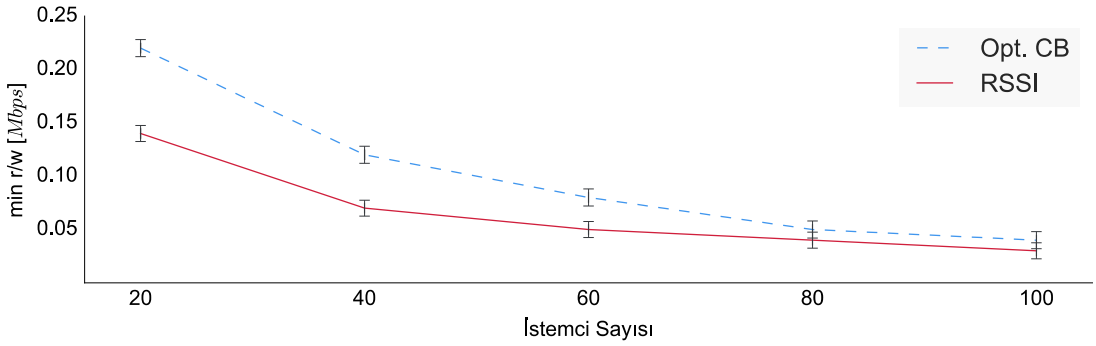
Şekil 4.3: Senaryo 1 için ortalama veri transfer hızı.



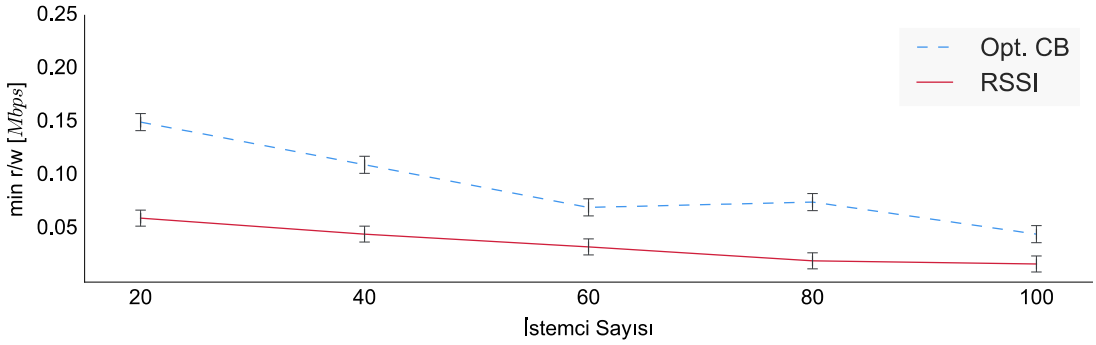
Şekil 4.4: Senaryo 2 için ortalama veri transfer hızı.



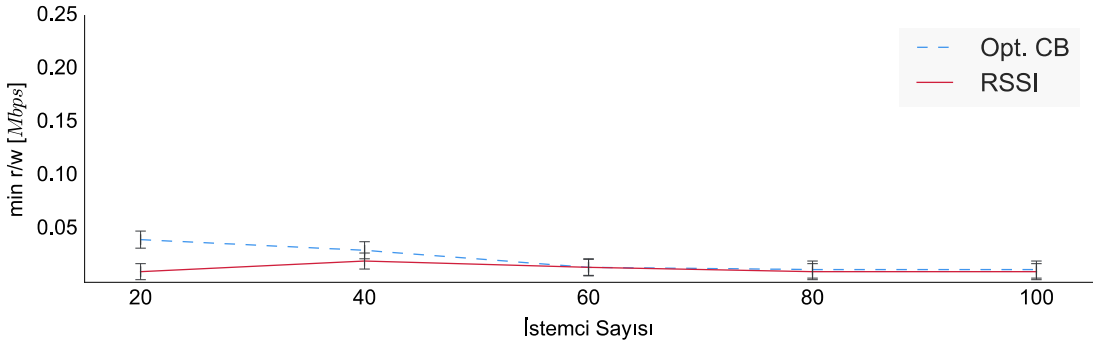
Şekil 4.5: Senaryo 3 için ortalama veri transfer hızı.



Şekil 4.6: Senaryo 1 için min. ağırlıklı veri transfer hızı.



Şekil 4.7: Senaryo 2 için min. ağırlıklı veri transfer hızı.



Şekil 4.8: Senaryo 3 için min. ağırlıklı veri transfer hızı.

Bu sonuçlara göre sunmuş olduğumuz model, QoS ve LB problemlerinde kullanabilecek genel bir çatı oluşturmaktadır. Bu çatı birden fazla AP olduğu ortamda, her bir AP içerisinde ve diğer AP'ler arasında QoS destekli yük dengeleme algoritmalarının tasarlanabileceği üzerine bir önermedir. Önerdiğimiz çatı Brute-Force yöntemi ile üç adet

AP oluşan ve kullanıcı sayısının az olduğu bir ortamda test edilerek doğrulanmış olup, Genetik Algoritma ve BB ile daha yoğun kullanıcılardan oluşan ve AP sayısının daha fazla olduğu durumlarda ayrıca test edilmiştir.

4.2. GENETİK ALGORİTMA

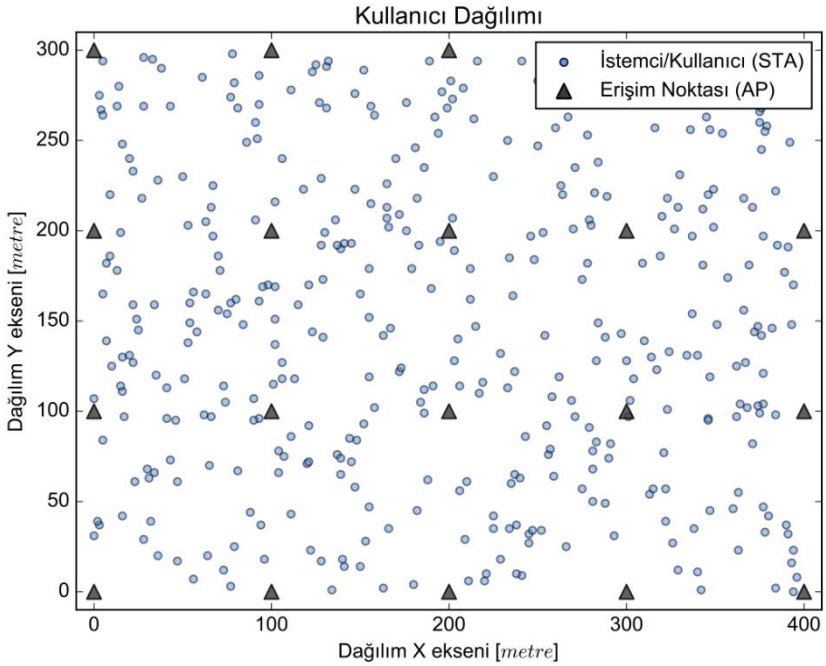
Model iki farklı ağ senaryosu ve iki farklı yapılandırma ve iki farklı kullanıcı dağılımlarından oluşan bir ağ topolojisine göre testler yapılmıştır. Şekil 4.9'da kullanıcılar düzgün bir şekilde tüm alanı dolduracak şekilde senaryo, Şekil 4.10'da ise istemciler toplam alanın orta kısmında yoğunlaşacak oluşturduğu ağ senaryosu bulunmaktadır. Her iki senaryo içinde 50 ile 400 arasında değişen kullanıcı sayıları 400m x 300m alana dağıtılmıştır. 20 adet AP, ızgara şekilde (5x4) alana 100m aralıklarla tüm istemcilere hizmet verecek şekilde konumlandırılmıştır. Her bir istemci farklı ağırlıklarda $\{1, 2, 3, 4\}$ ve rastgele boyutlarda $\{500, 1000, 1500\}$ paketler üreterek, paketleri iletirler.

İki farklı yapılandırma ile tüm AP ve kullanıcı dağılımları için testler gerçekleştirilmiştir;

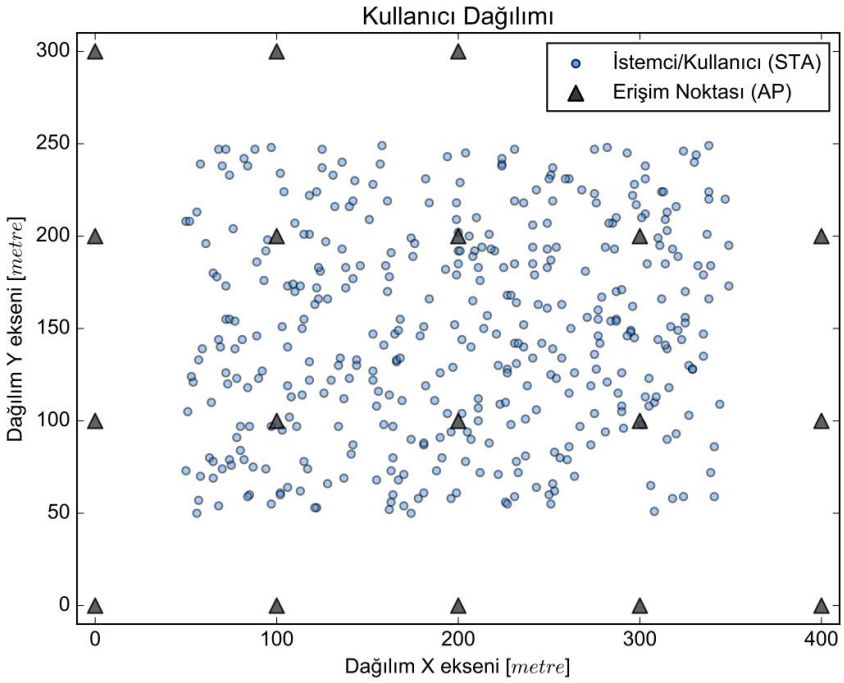
- RSSI, bu stratejide, tüm AP'ler sabit güç iletim seviyesi ile paket iletmektedir. Her bir istemci en yüksek güç göstergesine göre AP'yi seçmektedir. EDCA parametreleri AP'ler içerisinde adil trafik kullanımı için optimize edilmiştir.
- GA, bu stratejide, güç iletim seviyeleri ve EDCA yapılandırmaları optimize edilmiştir. GA ile güç iletim seviyeleri optimize edilerek ağ üzerindeki toplam throughput optimizasyonu sağlanmaktadır.

Testler IEEE 802.11b protokolü ile yapılmış olup, seçilen SNR değerleri $11\ Mbps\ SNR >= 9dB,\ 5.5Mbps\ SNR >= 5dB,\ 2Mbps\ SNR >= 3\ dB$ ve $1\ Mbps\ SNR >= 1dB$ dir. 20dBm'den başlayıp 11dBm'e kadar 10 farklı paket iletim güç seviyesi kullanılmıştır.

Kapalı ortam sinyal kayıp modeli $PL(d) = 40 - 10 * 3.3 * logd$ ile ifade edilmektedir. d AP ve istemci arasındaki mesafeyi verir [38].



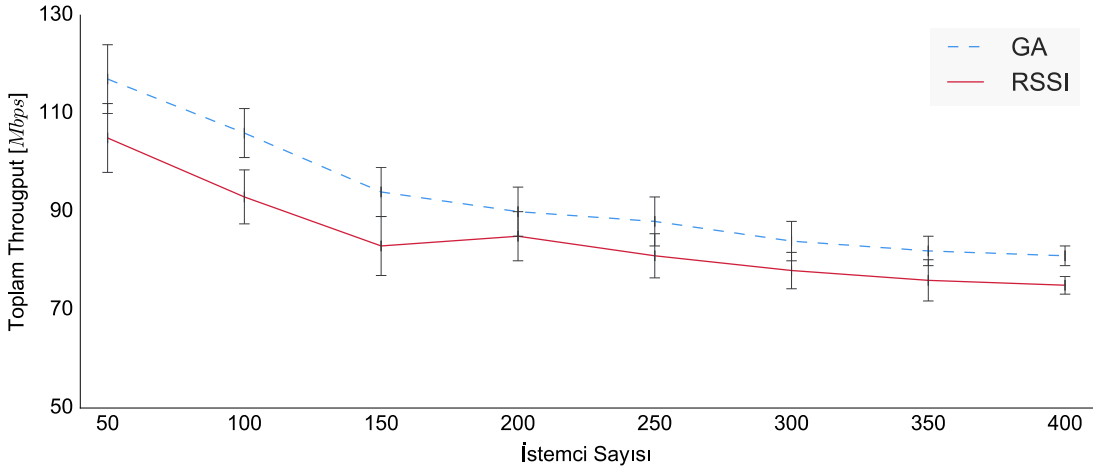
Şekil 4.9: GA Senaryo 1 kullanıcı dağılımı.



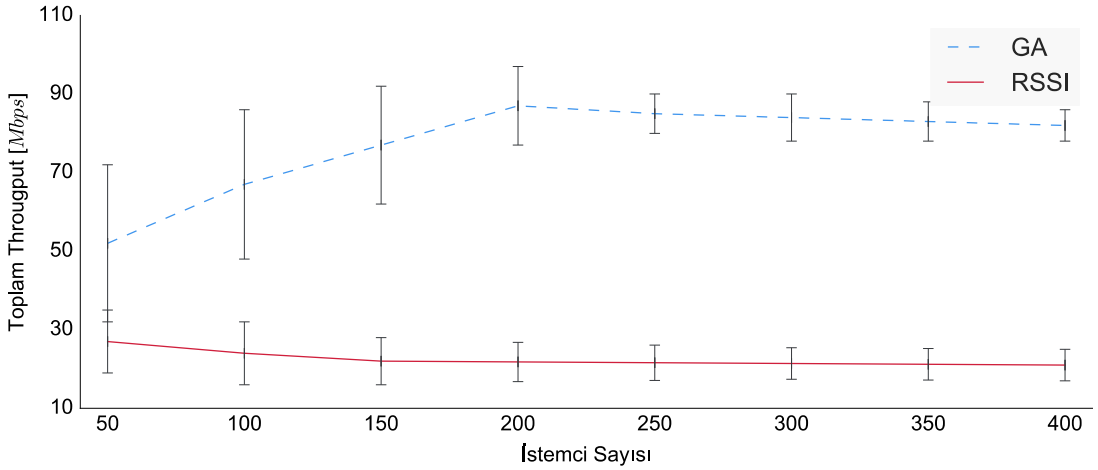
Şekil 4.10: GA Senaryo 2 kullanıcı dağılımı.

Her bir yapılandırma 10 kere, kullanıcıların rastgele dağılımı ile simüle edilmiştir. Şekil 4.11 ve Şekil 4.12'de toplam throughput, Şekil 4.13 ve Şekil 4.14'de ise en düşük ağırlıklı

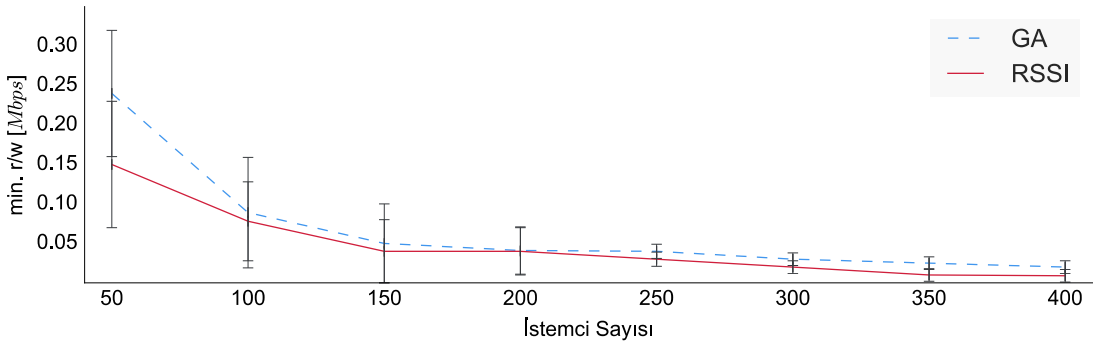
kullanıcıya ait throughput gösterilmiştir. Düz çizgi şeklinde gösterilen kırmızı renkli çıktılar RSSI, kesik çizgiler ile ifade edilen mavi renkli değerler ise Genetik Algoritma optimizasyon sonucunu belirtmektedir. Şekil 4.15 ve Şekil 4.16 ise algoritmanın çalışma zamanı saniye cinsinden belirtilmiştir. Güven aralığımız (Confidential Interval) %95 olup, grafiklerde bar olarak gösterilmiştir.



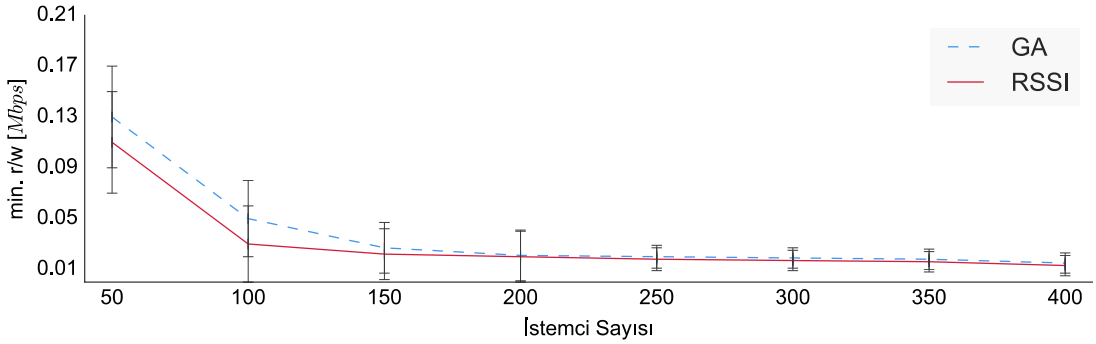
Şekil 4.11: GA Senaryo 1 toplam throughput.



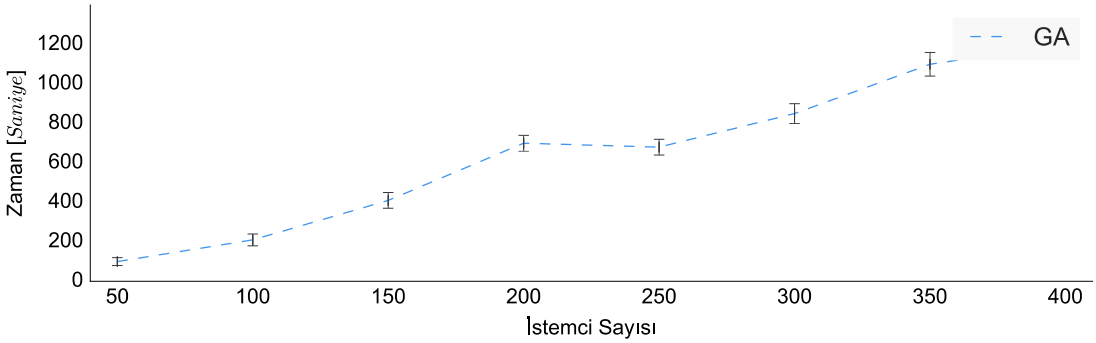
Şekil 4.12: GA Senaryo 2 toplam throughput.



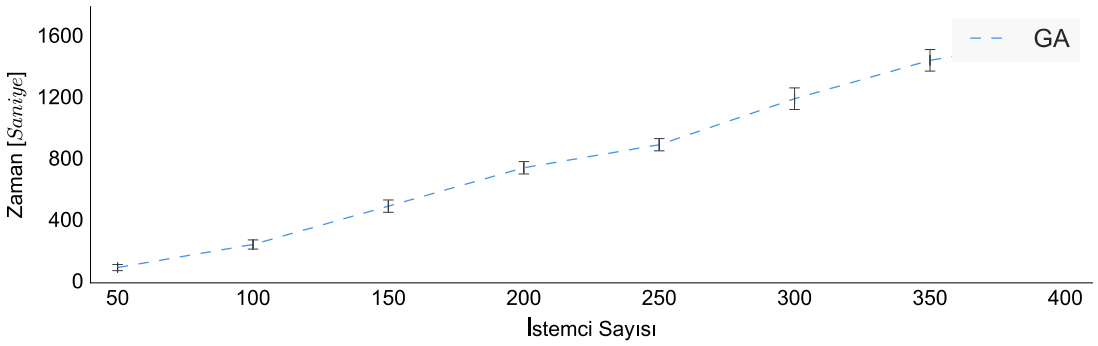
Şekil 4.13: GA Senaryo 1 min. ağırlıklı throughput.



Şekil 4.14: GA Senaryo 2 min. ağırlıklı throughput.



Şekil 4.15: GA Senaryo 1 çalışma zamanı.



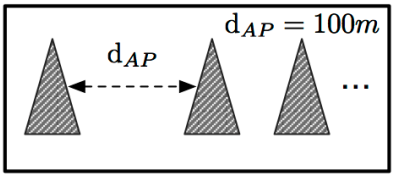
Şekil 4.16: GA Senaryo 2 çalışma zamanı.

Bu çalışmamızda kablosuz ağlarda var olan bir optimizasyon problemini çözmek için sezgisel yöntemler kullanılmıştır. Genetik Algoritma ile AP'lerin iletim güç değerleri planlanarak, kapsama alanları değiştirilmiştir. Bu sayede bir AP'ye bağlı olan kullanıcı sayısı ağ üzerindeki minimum ağırlıklı throughput arttığı gözlemlenmektedir. RSSI ve GA sonuçları karşılaştırıldığında, her iki senaryo ve ağ yapılandırması için GA'nın RSSI'ye göre daha iyi sonuçlar elde ettiğini Şekil 4.11 ve Şekil 4.12 'deki çıktılardan yorumlayabiliriz. GA kullanıcı bağlantısını kontrol etmek için CB yöntemini kullanmaktadır. Senaryo 1 ve 2 incelendiğinde, Senaryo 2'nin Senaryo 1'e göre daha iyi sonuç verdiği gözlemlenmektedir. CB algoritması ile istemci ve AP'nin bağlantısının kolay kontrol edilebildiği ortamlarda, az sayıda kullanıcıya hizmet veren AP'lere kullanıcılar dağıtılarak ağ üzerindeki hizmet verilen kullanıcıların veri iletim hızları artırılmıştır. Bunun doğal sonucunda ise toplam throughput artırılarak hotspot optimizasyonu sağlanmıştır.

Algoritmanın çalışma zamanını incelediğimizde kullanıcı ve AP sayısının artmasına oranla, yükseldiği görülmektedir. Bu yüzden GA, yoğun AP'lerin bulunduğu ve yapılandırılabilir güç seviye sayısının onun üzerinde olduğu ağ topolojilerinde kullanılması daha uygun olacaktır.

4.3. BRANCH & BOUND ALGORİTMASI

BB algoritması test işlemleri için farklı AP yerleşim senaryoları hazırlanarak, testler bu topolojiye göre gerçekleştirilmiştir.

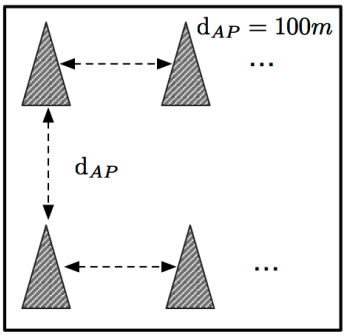


Şekil 4.17: BB AP dizilim senaryosu doğrusal.

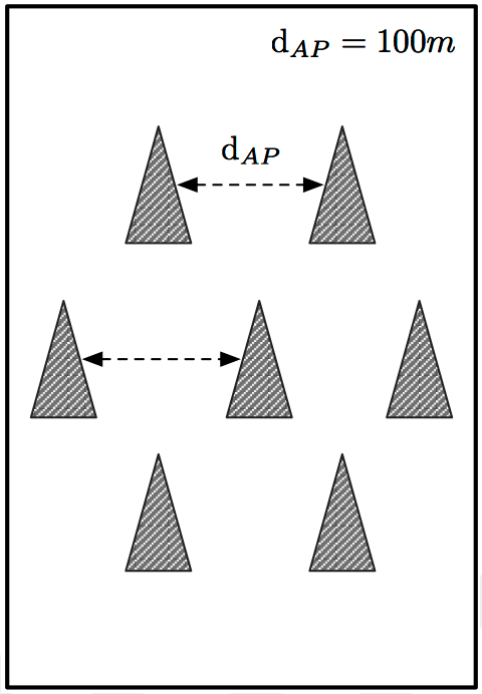
Şekil 4.17, Şekil 4.18 ve Şekil 4.19'da görüldüğü üzere doğrusal, ızgara ve yedigen olmak üzere üç farklı AP dağılımı ve iki farklı yapılandırma ile değişen kullanıcı sayılarına göre farklı senaryolar test edilmiştir. Yapılandırmalar bir birinden bağımsız olarak ayrı ayrı analiz edilmiştir. Simülasyonu yapılan yapılandırmalar;

- RSSI, bu yapılandırmada, tüm AP'ler sabit güç iletim seviyesi ile paket iletmektedir. Sadece EDCA parametreleri AP'ler içerisinde adil trafik kullanımı için optimize edilmiştir.
- BB, bu yapılandırmada, güç iletim seviyeleri ve EDCA yapılandırmaları optimize edilmiştir.

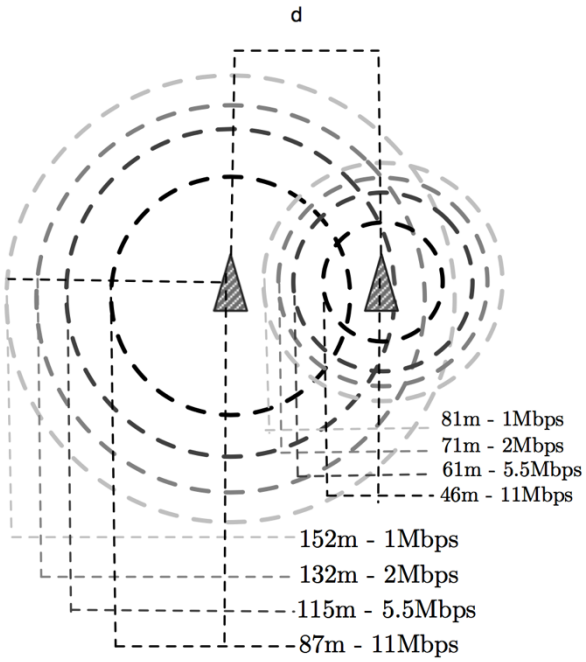
AP dağılımı yapılırken herhangi bir kapsama alanı boşluğu içermemekte ve bütün istemciler bir AP ile bağlantı kurabilmektedir. Testler IEEE 802.11b kablosuz bağlantısında test edilmiştir. GA kullanılan SNR değerleri, bağlantı hızları, kapalı ortam sinyal kayıp modeli $PL(d) = 40 - 10 * 3.3 * logd$, kullanıcı ağırlıkları $\{1, 2, 3, 4\}$ ve rastgele boyutlarda $\{ 500, 1000, 1500 \}$ UDP paketleri kullanılmıştır. Şekil 4.20 örnek bir sinyal seviyesi üzerinden elde edilen çıktılar gösterilmiştir.



Şekil 4.18: BB AP dizilim senaryosu ızgara.



Şekil 4.19: BB AP dizilim senaryosu yedigen.



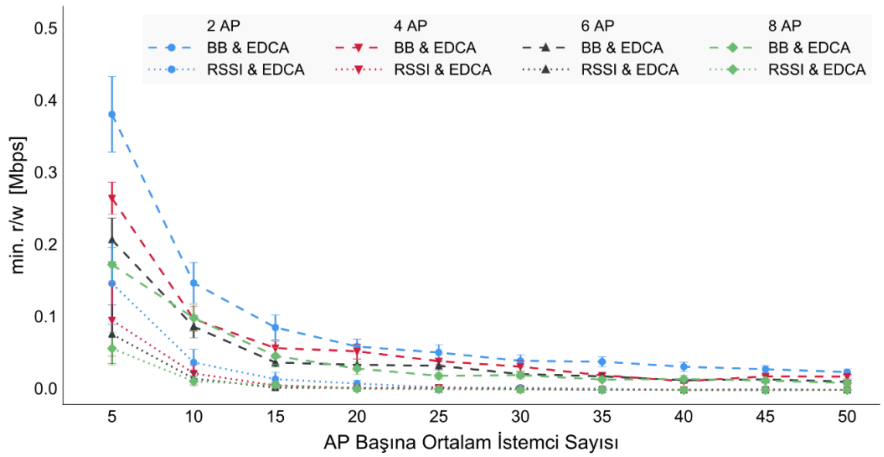
Şekil 4.20: Sinyal kayıp seviyeleri.

Üç senaryo için farklı istemci sayılarında (10-400) farklı sayıdaki kullanıcı dağılımı için 10'er defa tekrarlanmıştır.

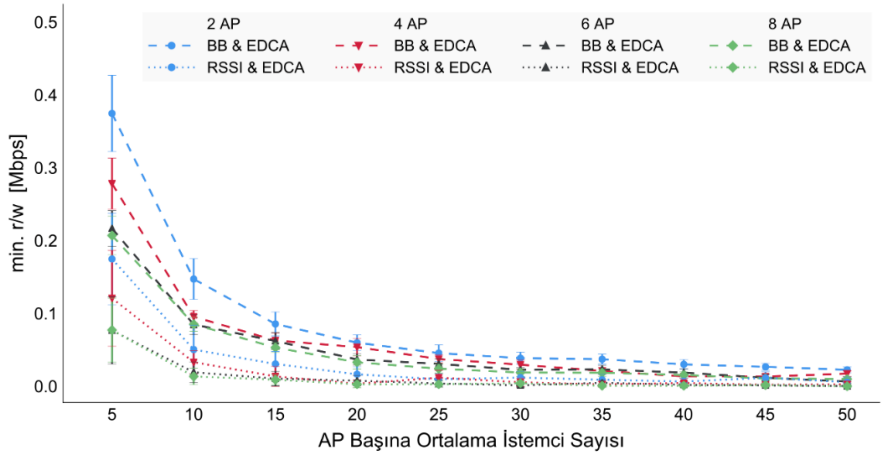
Şekil 4.21, Şekil 4.22 ve Şekil 4.23'de minimum ağırlıklı ortalama, Şekil 4.24, Şekil 4.25 ve Şekil 4.26 'te ise ağ üzerinde toplam throughput değerlerinin farklı AP sayısına göre farklı kullanıcılar için farklı yapılandırmalardaki sonuçları gözükmektedir.

Grafiklerde farklı sayıdaki AP'lerden oluşan test sonuçlarının farklı kullanıcı dağılımına göre sonuçları gösterilmektedir. Kesik çizgi ile belirtilen sonuçlar BB optimizasyonunu, noktalar halinde belirtilen sonuçlar ise standart RSSI yöntemini göstermektedir. Ayrıca güven aralığı %95 olup, hata oranı grafiklerdeki dikey çizgilerle gösterilmiştir.

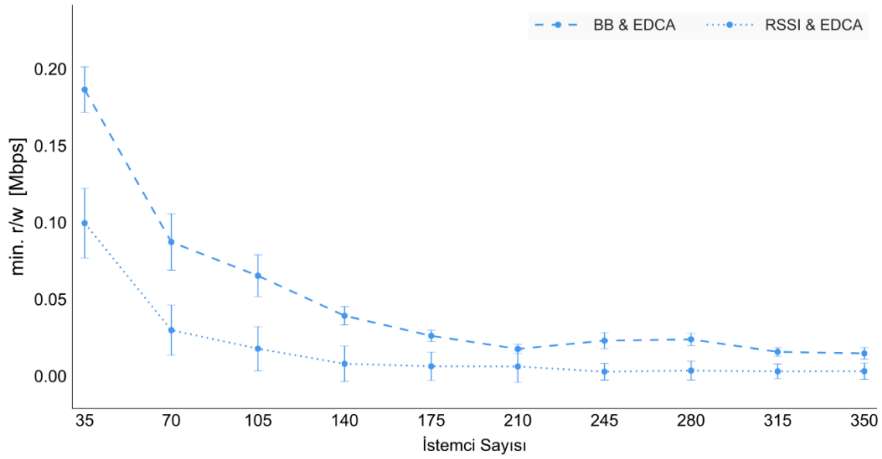
Tüm senaryolar için sonuçlar incelendiğinde BB yönteminin, RSSI'ya göre daha iyi bir grafik verdiğini gözlemleyebiliriz. Her iki yöntemde de beklendiği üzere AP ve kullanıcı sayısın artmasıyla birlikte, ağdaki trafik sıkışıklığı artmakta ve değerler düşmektedir.



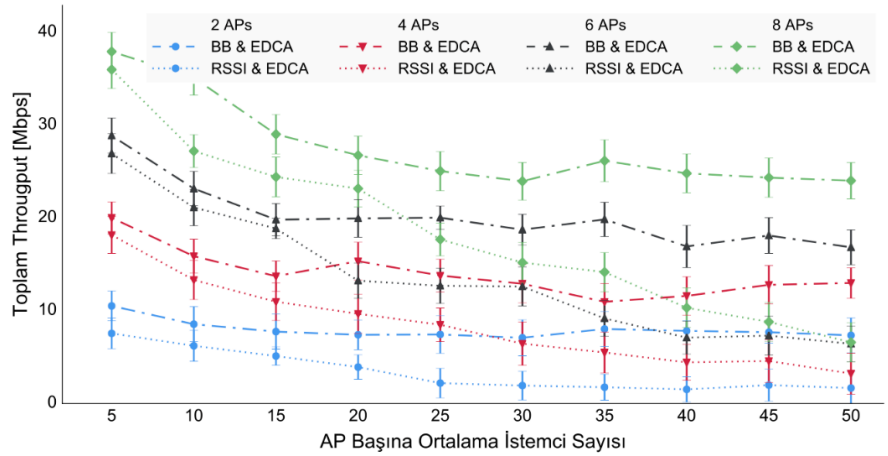
Şekil 4.21: BB Senaryo 1 min. ağırlıklı throughput.



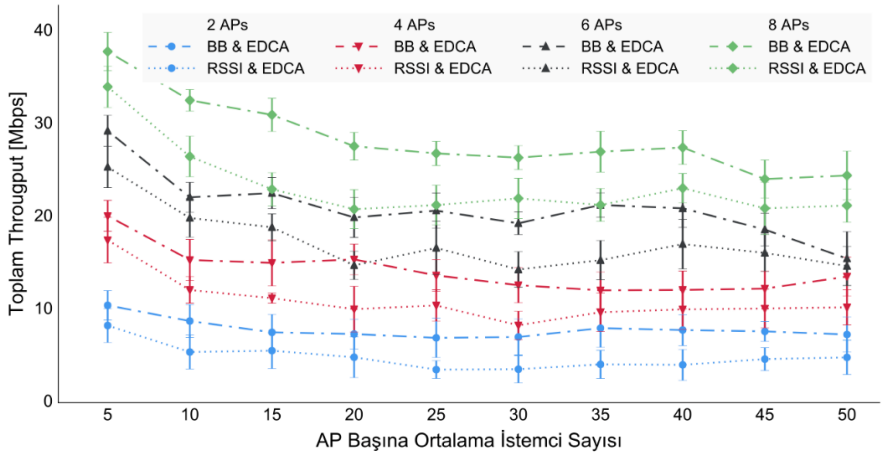
Şekil 4.22: BB Senaryo 2 min. ağırlıklı throughput.



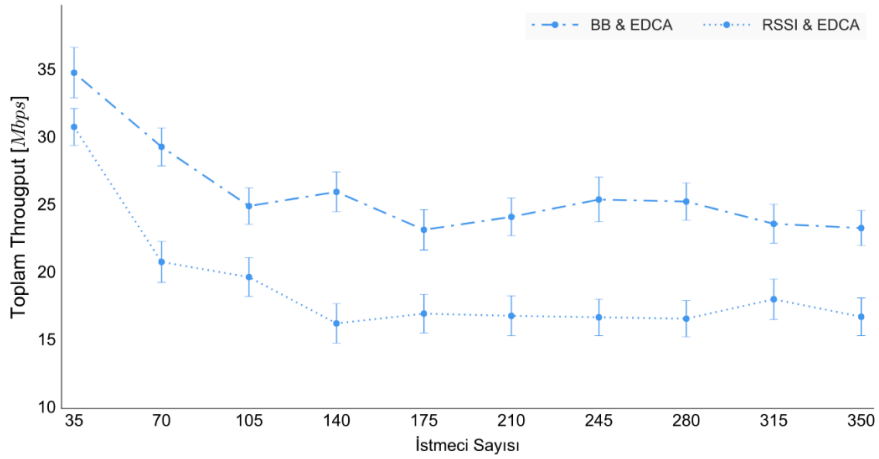
Şekil 4.23: BB Senaryo 3 min. ağırlıklı throughput (7 AP).



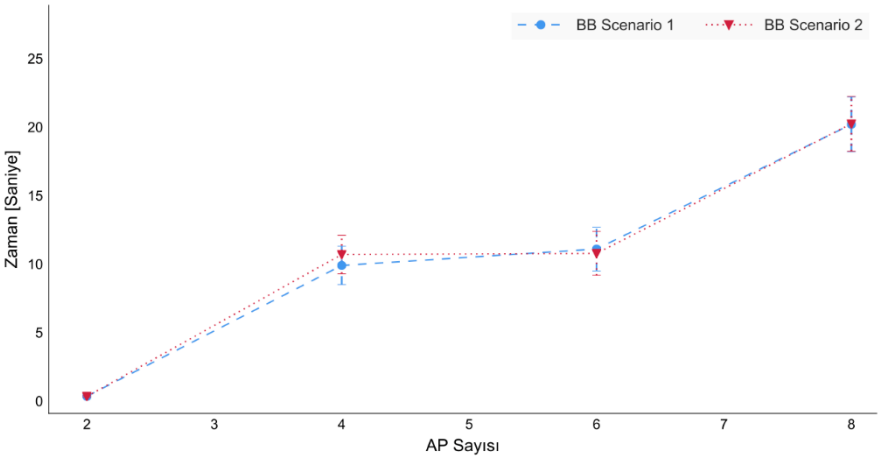
Şekil 4.24: BB Senaryo 1 toplam throughput.



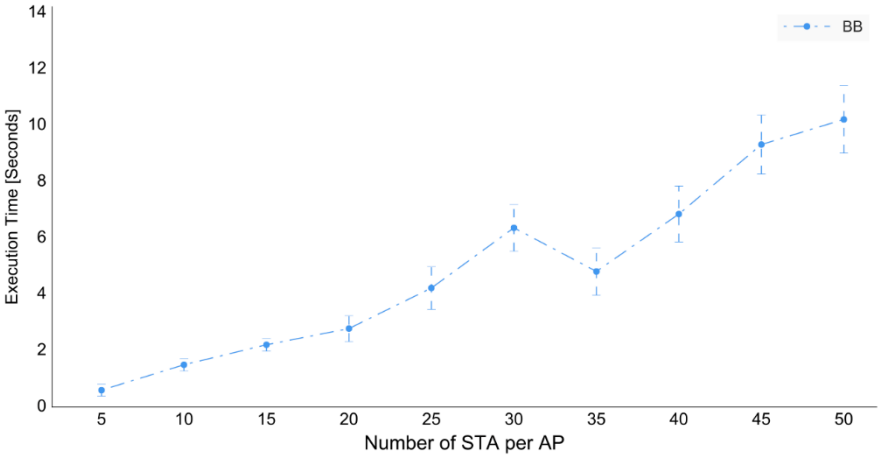
Şekil 4.25: BB Senaryo 2 toplam throughput.



Şekil 4.26: BB Senaryo 3 toplam throughput (7 AP).



Şekil 4.27: BB Senaryo 1 ve 2 algoritma çalışma zamanı.



Şekil 4.28: BB Senaryo 3 algoritma çalışma zamanı (7 AP).

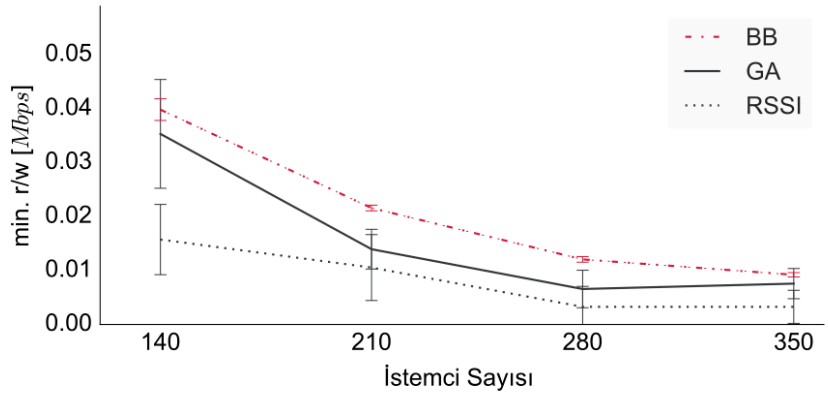
Şekil 4.24, Şekil 4.25, Şekil 4.26'deki toplam throughput incelendiğinde özellikte Şekil 4.26'de 7 AP'den oluşan yoğun kullanıcının olduğu bir topolojilerde, BB algoritmasının ağ üzerindeki 5 Mbps'e yakın bir katkısının olduğu net bir şekilde gözlemlenmektedir.

BB algoritmasının çalışma zamanı Şekil 4.27'de Senaryo 1 ve 2 için, Şekil 4.28'de ise Senaryo 3 gösterilmiştir. Çalışma zamanı 6 – 7 adet AP için 10 saniye seviyelerinde olup kısa bir süre uygulanabilirliği açısında önem arz etmektedir. AP sayısının ve AP başına düşen kullanıcıların artmasıyla algoritma çalışma zamanında arttığı gözlemlenmektedir.

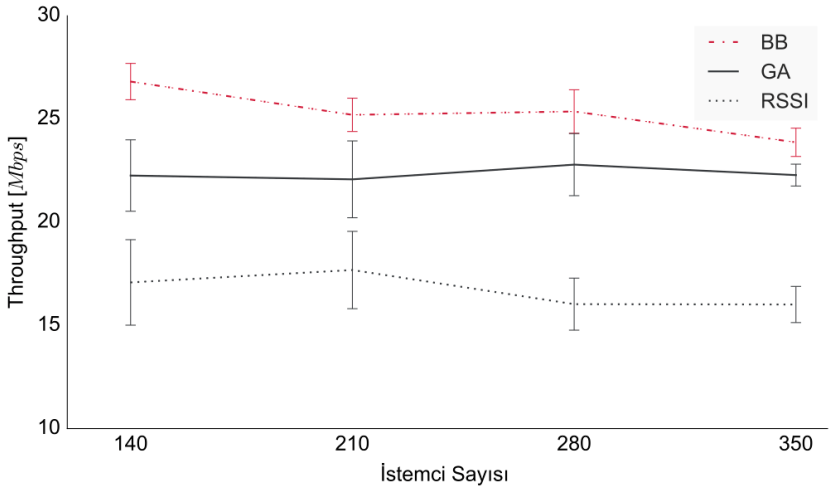
Karmaşık senaryolar ve kullanıcı dağılımlarında, BB algoritması daha çok alt ağaç taraması yaptığı bu süre belirli bir oranda arttığı gözlenmiştir. Bu durumda, CB yöntemi STA-AP bağlantısını kontrol etmek zaman almakta veya optimum çözüm ile standart çözüme çok yakın olabilmektedir. Bu sonucun yansımasını yer görülmektedir. Fakat toplam sonuca bakıldığında BB algoritmasının daha iyi çıktılar verdiği görülmektedir.

BB ve GA algoritmanın çalışmasını karşılaştırmak Şekil 4.19'da verilen yedigen AP dağılım üzerinden bir simülasyon çalışması yapılarak değerlendirilmiştir. Şekil 4.29, Şekil 4.30 ve Şekil 4.31'de min. ağırlıklı throughput, toplam throughput ve çalışma zamanı gösterilmektedir. Her iki algoritmada standart yöntem olan RSSI'den daha iyi sonuç verdiği gözlemlenmektedir. Çalışma zamanı açısından incelendiğinde ise orta ölçekli bir topoloji için BB algoritmasının GA'ya göre daha verilim olduğu ifade edilebilir. GA, ise daha büyük ağ altyapılarında kullanılması uygun olacaktır.

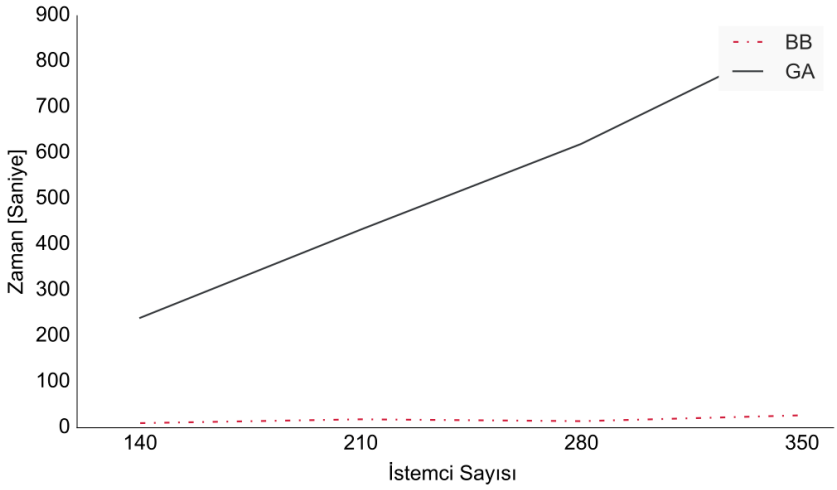
Bu tez çalışmamızda kablosuz ağlarda var olan bir optimizasyon problemini çözmek için çalışmalar yapılmıştır. Branch & Bound algoritması kullanılarak, AP'lerin iletim güç değerleri planlanarak ve kapsama alanları değiştirilmiştir ve bu sayede bir AP'ye bağlı olan kullanıcı sayısı ağ üzerindeki minimum ağırlıklı throughput'u arttırmıştır. BB yöntemi ile elde edilen sonuçları RSSI değerleri ile karşılaştırdığımızda sunduğumuz BB algoritmasının standart yöntemlere oranla daha olumlu sonuçlar ürettiği gözlemlenmiştir.



Şekil 4.29: BB ve GA karşılaştırması min. ağırlık throughput



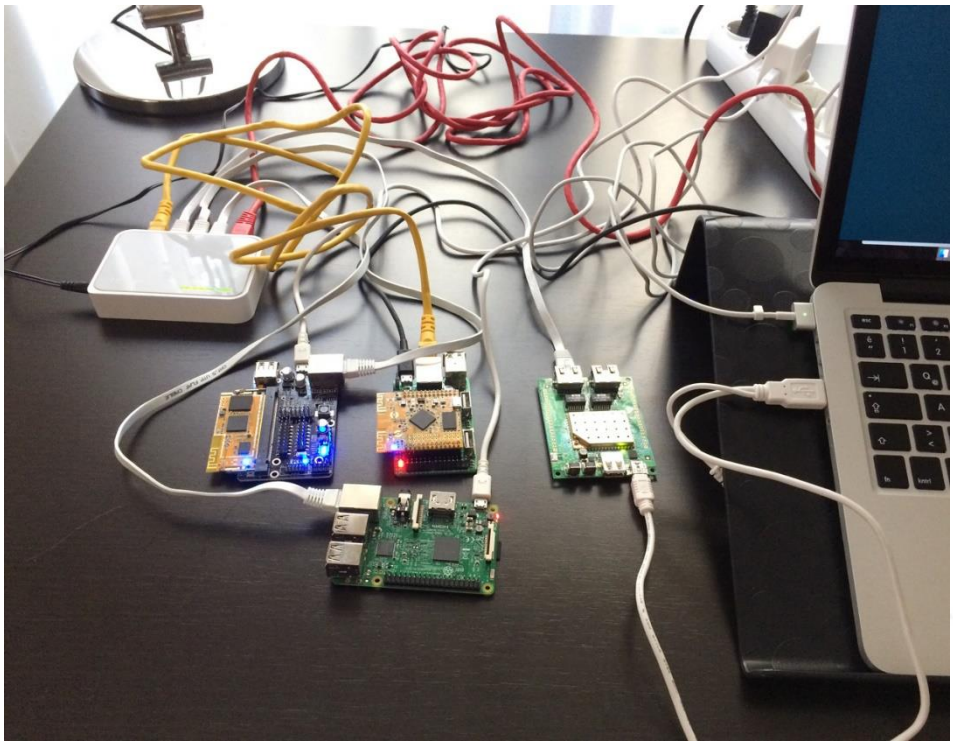
Şekil 4.30: BB ve GA karşılaştırması toplam throughput



Şekil 4.31: BB ve GA karşılaştırması çalışma zamanı

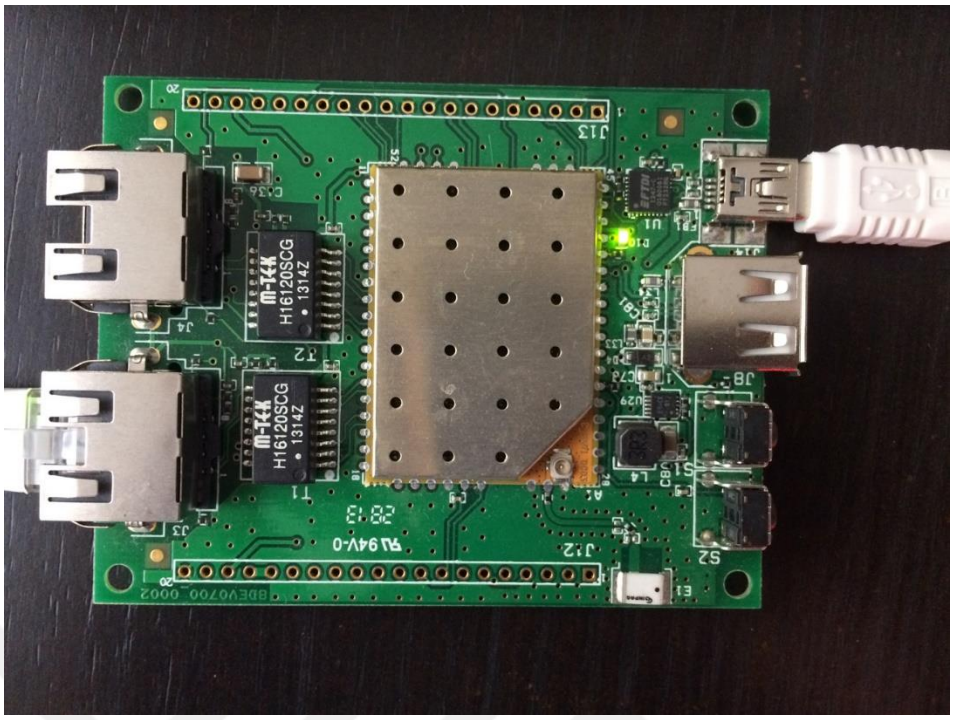
4.4. TESTBED ORTAMININ DOĞRULANMASI

CAPWAP protokolü kullanılarak Kablosuz Erişim Noktası Yöneticisi geliştirilmesi için çalışmalar tamamlanarak, farklı cihaz ve modeller üzerinde test yapmak için TestBed ortamı kurulup, küçük bir kablosuz ağ laboratuvarı hazırlanmıştır. Şekil 4.32'de TestBed ortamı için ön hazırlık çalışmaları gözükmektedir.



Şekil 4.32: TestBed ortamı hazırlık çalışmaları.

Öncelikli olarak gerçek cihazlarla hazırlanan TestBed ortamı simülatörler ile karşılaştırılarak ortam doğrulanmıştır. Ortamın doğrulanması ile birlikte gerçek ortam senaryoları hazırlanarak standart RSSI yöntemi ile BB algoritması tarafından elde dilen sonuçlar değerlendirilmiştir. Elde edilen sonuçlar doğrultusunda, kapalı gerçek bir sınıf ortamında, istemcilerin minimum ağırlıklı throughput ve tüm ağ içindeki toplam throughput optimize edildiği gözlenmiştir. Böylelikle hem bir AP içerisinde hem de tüm hotspot içerisinde kullanıcılara hizmet kalitesi artırılmıştır.



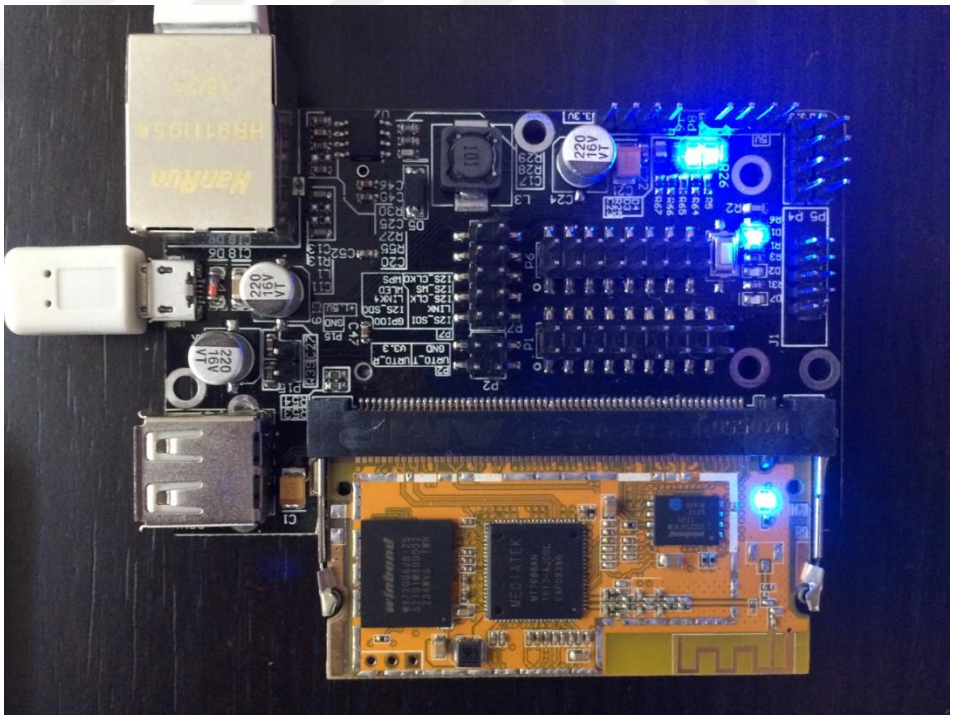
Şekil 4.33: TestBed ortamı Carambola2.

Yazılım Tabanlı ve Sağlayıcı ve Donanımdan bağımsız bir AC geliştirilebilmesi, Linux tabanlı ve kablosuz router geliştirme ortamı olan OpenWRT v15.05.1 kullanmıştır. AC ve WTP için geliştirilen kaynak kod, çapraz olarak x86, mips, mimarilerini üzerinde derlenmiştir. Derlenen uygulamalar, Carambola2[15] (Qualcomm/Atheros AR9331), WrtNode[16], WrtNode2R[17] ve Raspberry Pi 3[18] cihazları üzerinde çalıştırılarak test edilmiştir.

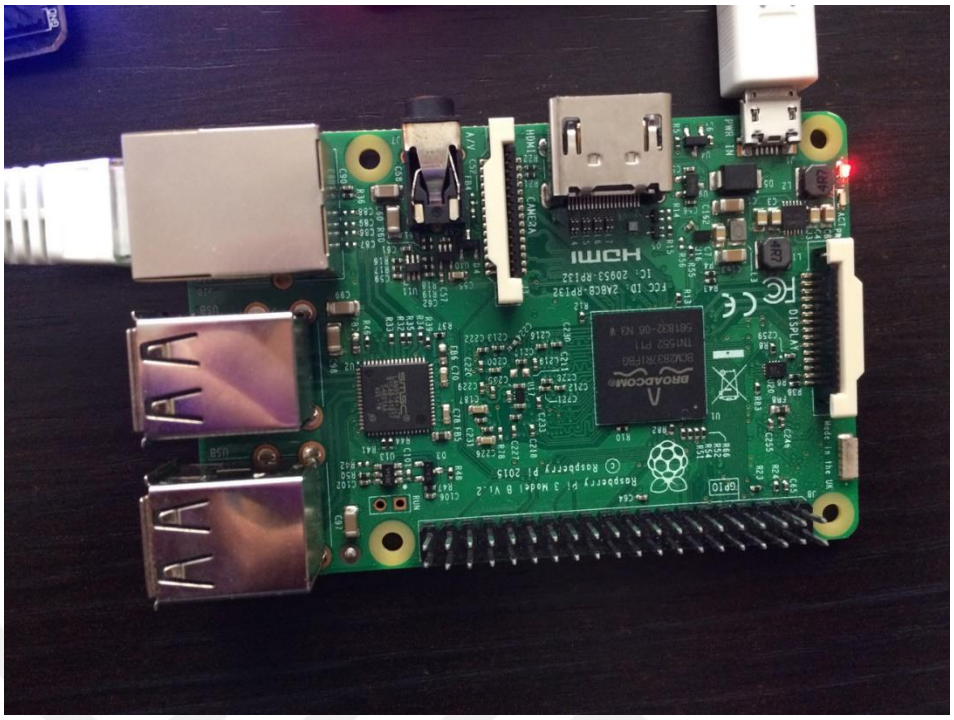
TestBed ortamında kullanılan Carambola2 Şekil 4.33'da, WrtNode Şekil 4.34'de, WrtNode2R Şekil 4.35'de, Raspberry Pi 3 ise Şekil 4.36'de görülmektedir.

[15] Carambola2, WiFi geliştirme kiti, http://www.8devices.com/products/carambola-2, [Ziyaret tarihi: 10 Ekim 2016]
[16] WRTnode - minimal WiFi & Linux gömülü sistemi, http://wrtnode.com/w/, [Ziyaret tarihi: 10 Ekim 2016]
[17] WRTnode2R - minimal WiFi & Linux gömülü sistemi, http://wrtnode.com/w/?p=696, [Ziyaret tarihi: 10 Ekim 2016]
[18] Rasperry Pi model 3, https://www.raspberrypi.org/products/raspberry-pi-3-model-b/, [Ziyaret tarihi: 10 Ekim 2016]

Şekil 4.34: TestBed ortamı WrtNode.



Şekil 4.35: TestBed ortamı WrtNode2R.



Şekil 4.36: TestBed ortamı Raspberry Pi 3.

WTP'ler üzerinde çalışan agent uygulamanın herhangi bir üreticiden ve üreticilerin sunduğu sürücülerden bağımsız çalışması gerekmektedir. Bu sebeplerden dolayı, driver ile etkileşim kurmak yerine Linux çekirdeği ile etkileşmesi gerekmektedir. Bu noktada Linux Netlink kütüphaneleri devreye girmektedir.

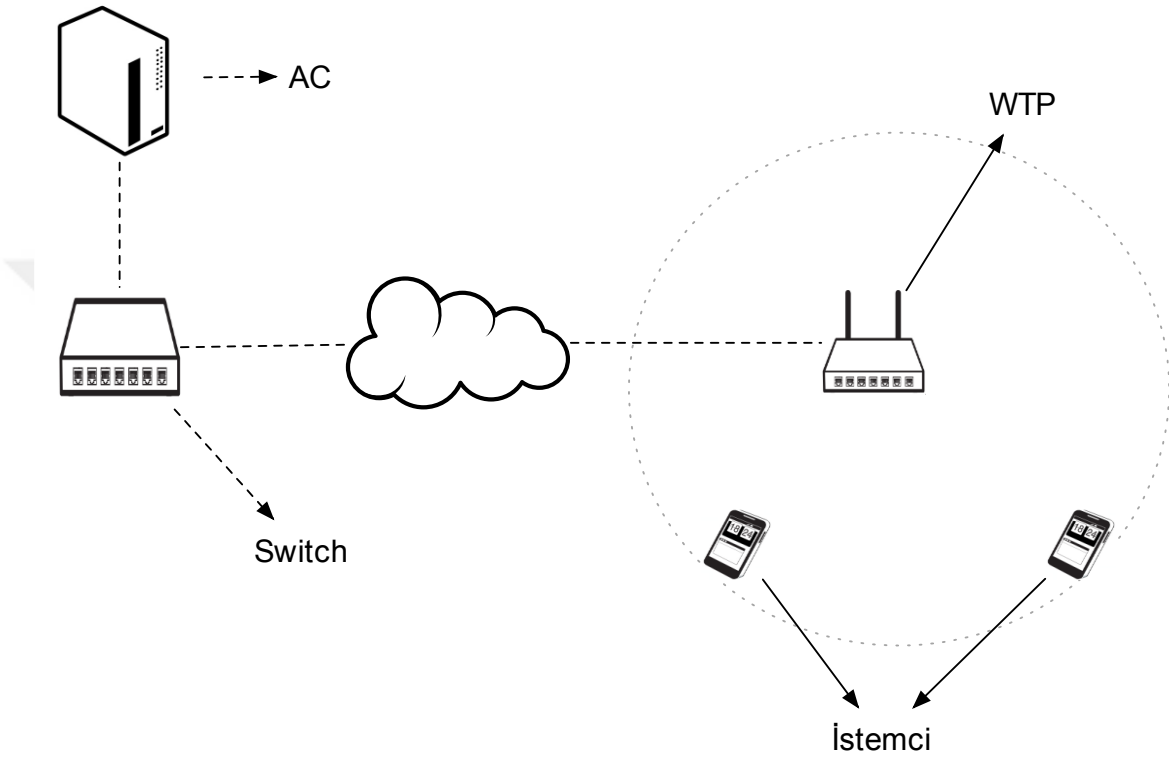
CAPWAP protokol mesajları ile gönderilen yapılandırma ayarları, WTP üzerinde çalışan agentlar tarafından işlenmektedir. Gelen mesaj komutları, netlink mesajlarına dönüştürülerek, nl80211 soketleri ile işletim sistemi çekirdeğine gönderilerek cihazlar yapılandırılmaktadır.

AC tarafından gönderilen EDCA yapılandırmaları, CWmin, CWmax, AIFS, TXOP parametreleri, AP paket iletim gücü kullanılarak istemcilerin AP ile olan bağlantıları kontrol altına alınabilmektedir. Bu sayede önceki bölümde belirtilen yük dengeleme algoritmaları kullanarak TestBed ortamında gerekli testler gerçekleştirilmiştir.

TestBed ortamının sonuçları ile, EDCA simülasyon sonuçları ile karşılaştırmalı olarak değerlendirilmiştir. Şekil 4.37'de gösterildiği gibi bir TestBed ortamı hazırlanmıştır. Buna göre, AC, switch, WTP ve istemcilerden oluşmaktadır. Test yapılırken 1 AP – 1

STA ve 1 AP – 2 STA olmak üzere ayrı ayrı, farklı iletim hızlarında testler gerçekleştirilmiştir. İstemcilerin AP'ye eşit uzaklıkta konumlandırılmışlardır.

TestBed ortamında yapılan doğrulama ve diğer test işlemleri için ağ performans analiz aracı olan iperf[19] kullanılmıştır.



Şekil 4.37: TestBed/EDCA simülasyon karşılaştırma ortamı.

BB algoritması TestBed ortamında test edilmeden önce, gerçek ortam ile EDCA simülasyonu arasındaki farklılıkları ortaya koymak ve gerçek ortam ile EDCA yakınlarını karşılaştırmak için bir doğrulama çalışması yapılmıştır. Testler sonucunda simülasyon ve gerçek ortam değerlerinin birbiriyle uyumlu olduğu gözlemlenmiştir. Elde edilen sonuçlar birbirine yakın olmak ile birlikte, gözlemlenen fark ortam gürültüsü (aynı frekansta yayın yapan diğer kablosuz cihazlardan kaynaklanan frekans girişimi) ve TestBed ortamının tam bir TCP/IP yığının kullanarak çalışan gerçek bir uygulama olmasından kaynaklanmaktadır. Zira, EDCA simülasyonu sadece EDCA MAC katmanını simüle ederek frekans girişimini yok saymaktadır.

[19] iPerf, TC, UDP ve SCTP performans test arcı, https://iperf.fr, [Ziyaret tarihi: 10 Ekim 2016]

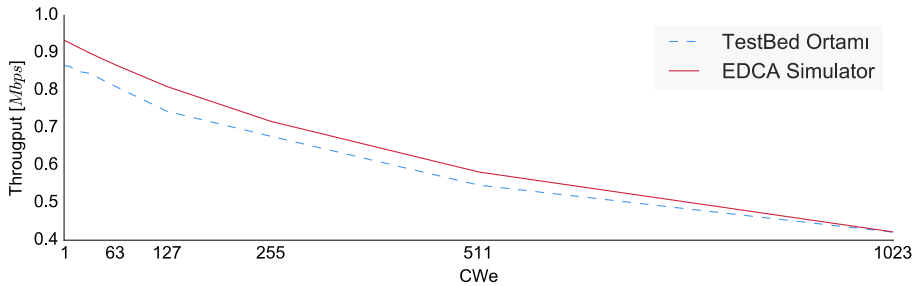
EDCA simülasyonu Tablo 4.1'de belirtilen EDCA parametreleri kullanılmıştır. Hem simülasyon hem de TestBed ortamında 470 byte boyutunda VoIP paketleri kullanılmıştır. Toplam throughput AP tarafından gözlemlenerek kayıt altına alınmıştır.

IEEE 802.11e tarafından belirlenen dört ana kategori üzerinden CW değerleri yapılandırılabilmektedir. CW değerleri $\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$ kümesinden seçilerek, $CWe = 2^{CW} - 1$ olacak şekilde hesaplanarak, $CWe = \{1, 3, 7, 15, 31, 63, 127, 255, 511, 1023\}$ kümesi elde edilmektedir [8]. Simülasyon ve TestBed ortamı bu değerler üzerinden test edilerek doğrulanmıştır.

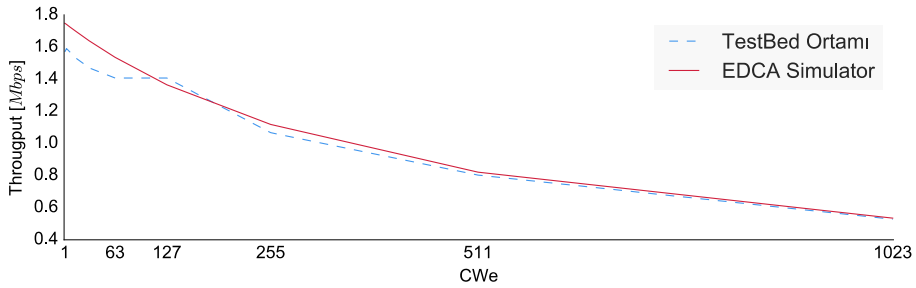
Grafikler incelendiğinde elde edilen sonuçların birbirine yakın olduğu gözlemlenmektedir. Ayrıca, 1 AP ile 1 STA kullanılarak yapılan testler ve elde edilen toplam throughput değerleri

Şekil 4.38, Şekil 4.39, Şekil 4.40 ve Şekil 4.41 de görülebilir. 1 AP ve 2 STA kullanılarak yapılan test sonuçları Şekil 4.42, Şekil 4.43, Şekil 4.44, Şekil 4.45'de ortalama ve toplam throughput olmak üzere gözükmektedir.

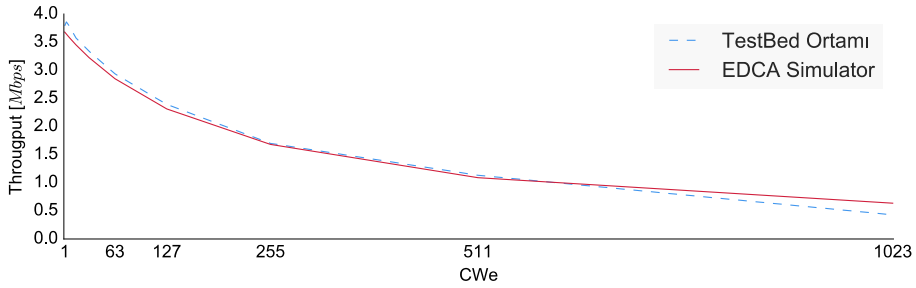
Testler yapılırken IEEE 802.11b protokolü fiziksel veri iletim hızları olan 1Mbps, 2Mbps, 5.5Mbps ve 11Mbps için ayrı ayrı bağımsız testler gerçekleştirilmiştir. 1 AP – 1 STA oluşan test ortamlarında sadece toplam throughput değerlendirilirken, 1 AP – 2 STA oluşan test senaryosunda ortalama değerlerde göz önüne alınmıştır.



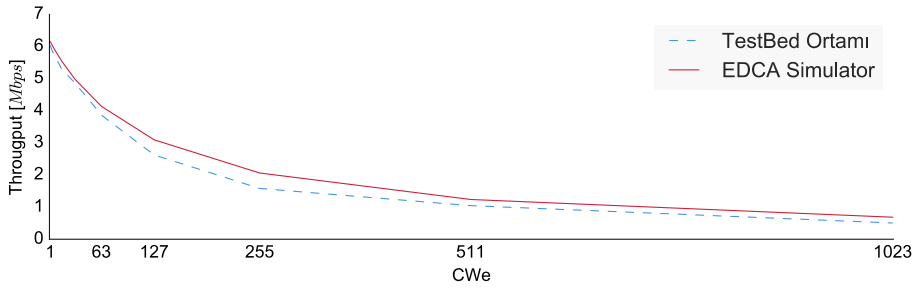
Şekil 4.38: TestBed ortamı, 1 AP - 1 STA, 1 Mbps throughput doğrulama.



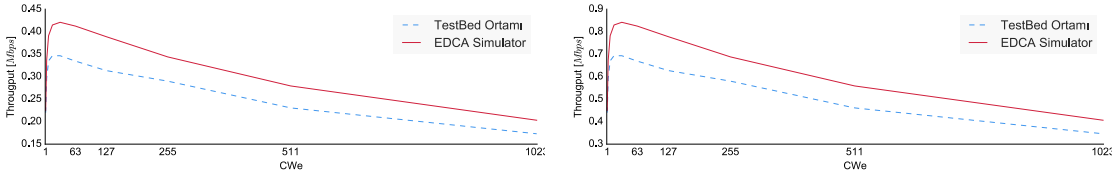
Şekil 4.39: TestBed ortamı, 1 AP - 1 STA, 2 Mbps throughput doğrulama.



Şekil 4.40: TestBed ortamı, 1 AP - 1 STA, 5.5 Mbps throughput doğrulama.

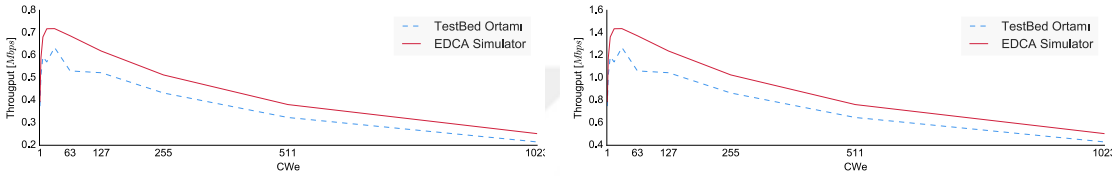


Şekil 4.41: TestBed ortamı, 1 AP - 1 STA, 11 Mbps throughput doğrulama.



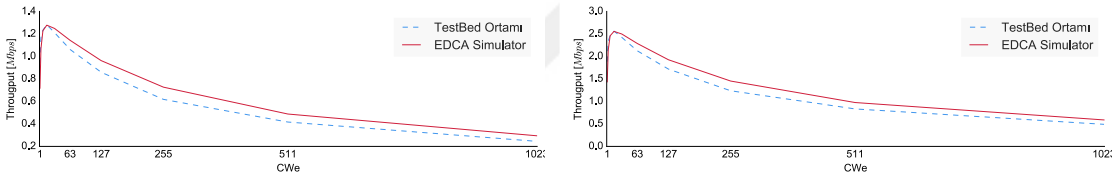
(a) Ortalama throughput. (b) Toplam throughput.

Şekil 4.42: TestBed ortamı, 1 AP - 2 STA,1 Mbps throughput doğrulama.



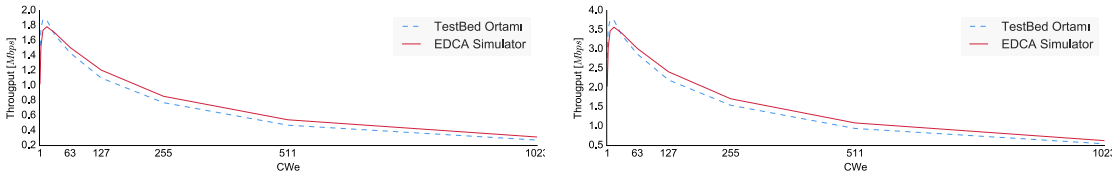
(a) Ortalama throughput. (b) Toplam throughput.

Şekil 4.43: TestBed ortamı, 1 AP - 2 STA, 2 Mbps throughput doğrulama.



(a) Ortalama throughput. (b) Toplam throughput.

Şekil 4.44: TestBed ortamı, 1 AP - 2 STA, 5.5 Mbps throughput doğrulama.



(a) Ortalama throughput. (b) Toplam throughput.

Şekil 4.45: TestBed ortamı, 1 AP - 2 STA, 11 Mbps throughput doğrulama.

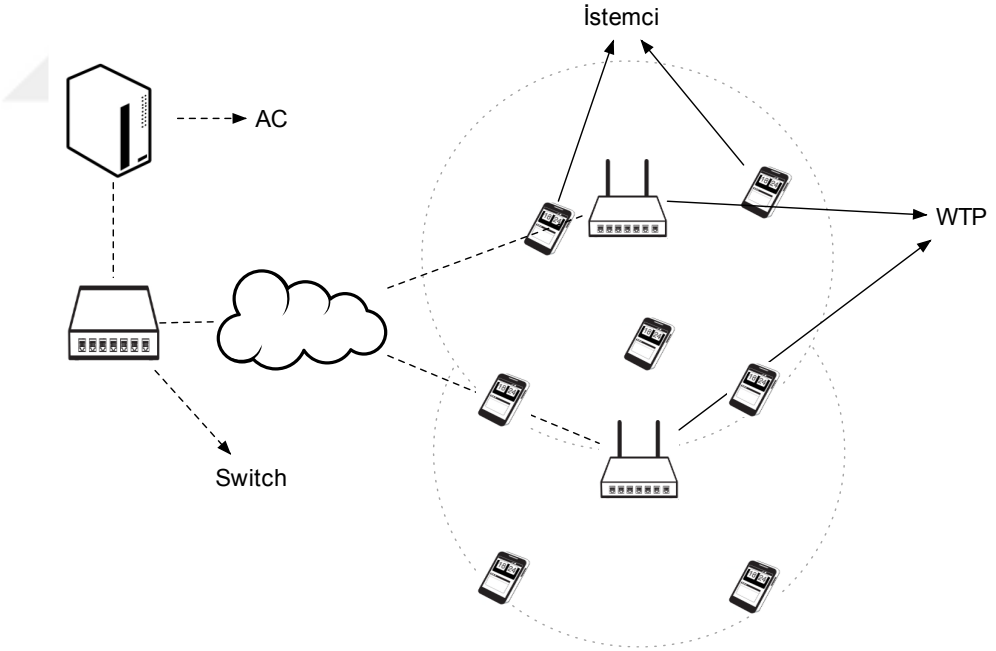
Yukardaki grafiklerden de görüldüğü üzere, hem 1 AP – 1 STA hem de 1 AP – 2 STA için throughput değerleri EDCA simülasyonu ve TestBed ortamı ile oldukça benzer özellikler göstermektedir. Bazı noktalarda ortaya çıkan farklılıklar sinyal girişiminden kaynaklandığı gibi, uçtan uca çalışan gerçek bir TestBed ortamında, işlem sistemleri versiyon çeşitliliği, üretici kaynaklı cihaz farklılıklarından (donanım ve sürücü) kaynaklanabilmektedir.

Bu sonuçlar doğrultusunda hazırlamış olduğumuz TestBed ortamı birçok araştırmacı içinde simülasyonlara alternatif ve gerçek sonuçlar ile çalışma ortamı sunan bir çatı olmuştur. Önceki dönemlerde yaptığımız çalışmalar bu çatıda test edilerek sonuçları da ayrıca değerlendirilmiştir.

4.5. B&B ALGORİTMASININ TESTBED İLE GERÇEKLENMESİ

BB yük dengeleme çalışmasıyla ilgili detaylı bilgi bir önceki raporda belirtilmiş olup, test işlemi sadece EDCA ile yapılarak değerlendirilmiştir. TestBed ortamının hazırlanması ile birlikte, algoritma gerçek cihazlar üzerinde test edilmiştir.

TestBed ortamı Şekil 4.46'de gösterildiği şekilde hazırlanmış olup kullanıcılar w = $\{1, 2, 3, 4\}$ olacak şekilde ağırlığa ve pl = $\{470, 940, 1410\}$ kümesinde bulunan boyutlarında rastgele paket gönderecek şekilde hazırlanmıştır. BB ve RSSI olmak üzere iki farklı yapılandırma kullanılarak testler yapılmıştır. Testler 2 AP ve 5-8 arasında değişen kullanıcı sayıları ile test edilmiş olup, her bir test 10 kez tekrarlanmıştır.



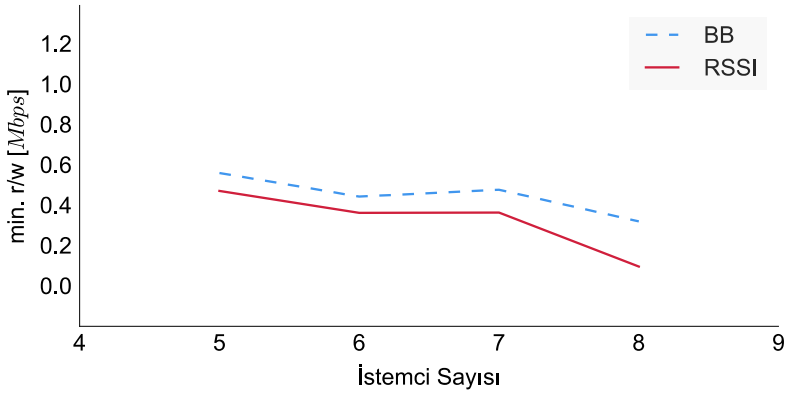
Şekil 4.46: BB algoritmasının TestBed ortamında gerçekleştirme senaryosu.

Test sonuçlarına göre Şekil 4.47'de minimum ağırlıklı throughput ile Şekil 4.48'da toplam throughput değerleri gözükmektedir. Her iki durumda da BB algoritması RSSI'a göre daha iyi sonuç verdiği gözükmektedir. Bu sayede EDCA yapılandırması kullanarak ağda

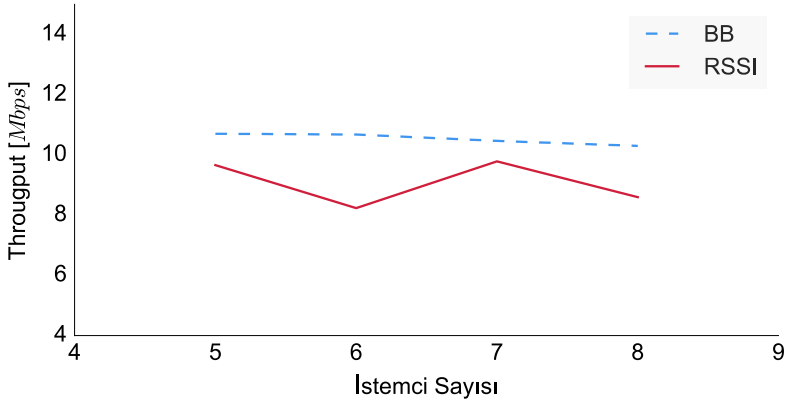
bulunan kullanıcıların hizmet kalitesi artırılarak, daha adil hizmet verildiği gözlemlenmektedir.

Testler gerçekleştirilirken kullanıcılar ortama rastgele dağıtılmaktadır ve aldıkları en yüksek RSSI değerine göre AP ile bağlantı kurmaktadırlar. Bu durumda zaman zaman bir AP'ye bağlanan kullanıcı sayısı bir veya iki olurken, diğer tüm istemciler ikinci AP'ye bağlanabilmektedirler. Veya istemciler iki AP'ye eşit dağılabilmektedir.

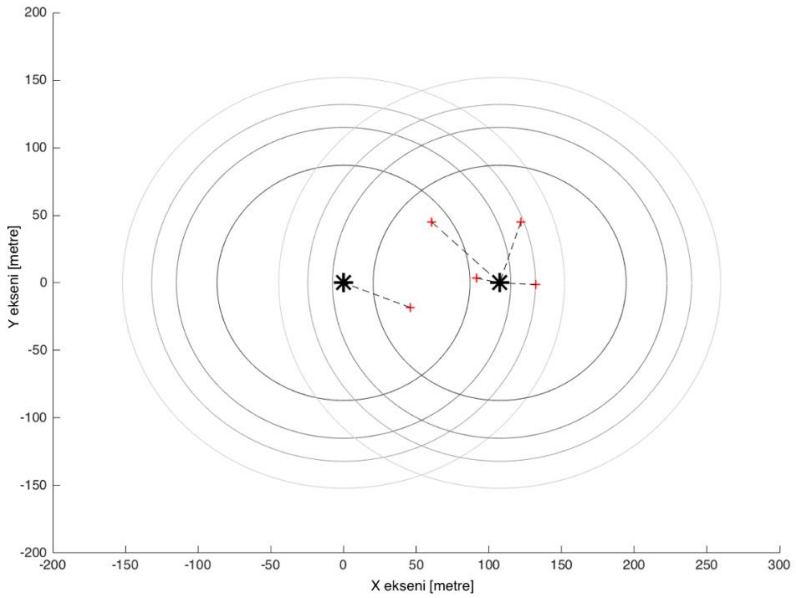
Şekil 4.48'de görüldüğü üzere istemci sayısına göre RSSI yapılandırmasında toplam throughput değeri farklılıklar gösterebilmektedir. BB optimizasyonu ile kullanıcılar arasındaki bağlantılar yeniden optimum seviyede sağlandığından toplam throughput değerinin RSSI yapılandırmasına göre daha iyi sonuç verdiği gözlemlenmektedir.



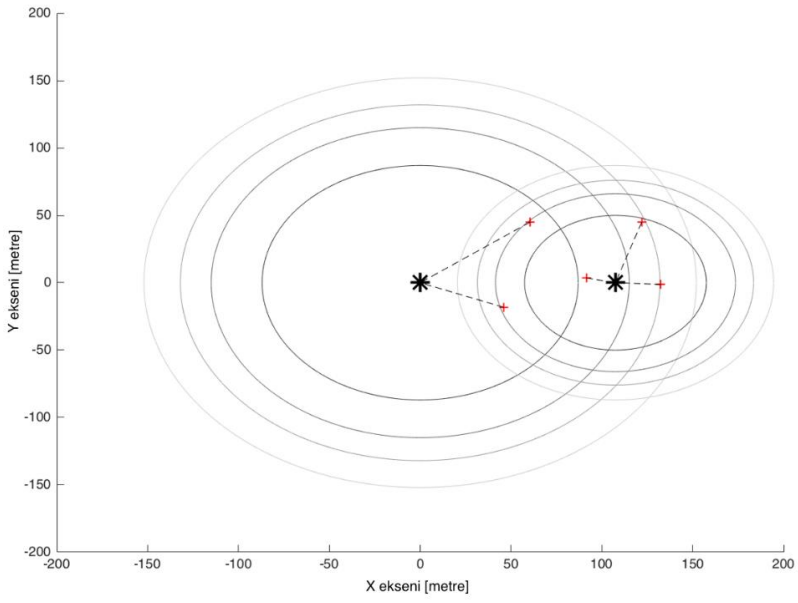
Şekil 4.47: TestBed ortamında min. ağırlık throughput.



Şekil 4.48: TestBed ortamında toplam throughput.



Şekil 4.49: TestBed ortamı standart RSSI ile AP-STA bağlantısı.



Şekil 4.50: TestBed ortamı BB ile AP-STA bağlantısı.

TestBed ortamında BB algoritmasının CB yöntemi ile AP'lerin kapsama alanını nasıl değiştirdiği Şekil 4.49 ve Şekil 4.50'de görülebilir. Şekil 4.49'da gözüken ilk durumda eşit güç seviyesi ile yapılandırılan AP'ler, dengesiz kullanıcı dağılımına sahiptirler. Bu durumda ağ üzerinde adil bir kullanım söz konusu değildir. Bir AP, sadece bir kullanıcıya

hizmet verirken, diğer AP, dört kullanıcıya hizmet vermektedir. Şekil 4.50'de BB ile güç seviyeleri tekrar yapılandırılarak, AP kapsama alanları güncellenmiş ve AP-STA bağlantılarının tekrar kurulması sağlanmıştır. Bu durumda AP'ler arası kullanıcı dağılımı ve hizmet kalitesi arttırılmıştır.

Yük dengeleme üzerine yapmış odluğumuz gerçek ortam çalışmalarının sonucunda, geliştirmiş olduğumuz Erişim Noktası Yöneticisi gerçek bir ortamda, test edilerek doğrulanmıştır. Yapmış olduğumuz tüm çalışmaların özgün değerlerini özetlediğimizde;

- Yazılım tabanlı, marka ve donanım bağımsız, standartlarla uyumlu bir Erişim Noktası Kontrolü yazılım tabanlı olarak geliştirilmiştir.
- Araştırmalarda kullanılmak üzere bir TestBed ortamı oluşturulmuştur.
- QoS ve yük dengeleme algoritmaları çalışmaları için bir çatı çerçeve oluşturulmuştur.
- Tasarlanan QoS ve yük dengeleme algoritmaları, TestBed ortamında gerçek ortam testleri yapılmıştır.

5. TARTIŞMA VE SONUÇ

Kablosuz ağların yaygınlaşması ve geleneksel yöntemler ile bu ağ altyapılarının yönetilmesi oldukça güçtür. Klasik yönetim stratejisini, merkeziyetçi bir yapıya taşıyarak tek bir noktadan kontrol etmek sistem yöneticilerine kolaylık sağlamasının yanı sıra, ağda oluşabilecek hataları minimize etmek için ideal bir yöntemdir. Günümüzde merkeziyetçi bir yapı ile kablosuz ağların yönetildiği sistemler olmakla birlikte, her bir sistem kendi üreticisinin çözümleri ile entegre çalışmakta olup, farklı firma ve donanımları desteklememektedir.

Orta ve büyük ölçekli ağ altyapılarının yönetimsel faaliyetlerini gerçekleştirebilmek için SNMP, RMON, CMIP, CAPWAP gibi protokoller ile SDN, SHH, JSON ve XML tabanlı yaklaşımlar olmak üzere çeşitli yöntemler geliştirilmiştir. Bu yöntemler incelendiğinde CAPWAP haricindeki protokoller, genel ağ yönetim standart ve teknikleri olup IEEE 802.11 kablosuz ağlar için özelleştirilmemiştir. Bu ihtiyaç doğrultusunda, IEEE 802.11 kablosuz ağlar için özelleştirilmiş ortak yönetimsel bir dil olan, CAPWAP geliştirilmiştir. CAPWAP IEEE 802.11 WiFi ağlarının yönetimi için özel olarak tasarlanan standart bir protokolüdür. Bu protokol sayesinden WTP'ler üzerinde yönetimsel ve izleme faaliyetleri yürütülebilmekte ve hatta WTP'ler için gerek duyulan yapılandırma fonksiyonelliği sunulabilmektedir. Böylelikle marka ve donanım bağımsız, yazılım tabanlı bir Erişim Noktası yöneticisi geliştirilebilir ve yönetsel faaliyetler gerçekleştirilebilir.

Mevcut çalışmalar incelendiğinde, WTP'lerin yönetimsel faaliyetlerini, markalara özgü yöntemler ile gerçekleştirdiği, marka bağımsız standart protokoller ile çalışan bir Erişim Noktası Yöneticisine (Access Controller - AC) ihtiyaç olduğu görülmektedir. Bu tez çalışmasında, CAPWAP protokolü kullanılarak marka ve sağlayıcıdan bağımsız yazılım tabanlı bir Erişim Noktası Yöneticisi geliştirilmiştir. Geliştirilen Erişim Noktası Yöneticisi, farklı üreticilerin geliştirmiş olduğu donanım, yonga seti ve mimarilerinde kurularak gerçek ortam testleri yapılmıştır. Uygulama geliştirme çalışmalarının ardından farklı donanımlardan oluşan TestBed ortamı kurularak bu ortam ile çalışmalarda kullandığımız simülasyon ortamı ile karşılaştırılarak gerekli sağlamalar yapılmıştır. Ayrıca, farklı donanım üretici ve farklı mimariler üzerinde test edilerek sağlayıcı bağımsızlığı doğrulanmıştır. Geliştirilen yazılım Erişim Noktası Yöneticisi ve

hazırlanmış olduğumuz TestBed ortamı IEEE 802.11 kablosuz ağlarında geliştirilecek olan algoritmaların test edilmesine imkan sunan bir mimariye sahiptir.

IEEE 802.11 standartları gereği istemciler bir AP ile bağlantı kurarken, AP'nin mevcut yük durumunu gözetmeksizin, en iyi sinyali aldığı AP ile bağlantı kurarlar. Bu da ağ üzerindeki yükün dengesiz dağılmasına ve performans problemlerine yol açmaktadır. Araştırmacılar bu soruna çözüm bulmak için çalışmalar yapmış olup, sunulan yöntemler, kullanıcıların AP içerisindeki trafik kullanımı göz ardı eden, standartlarla uyumlu olmayan veya sadece bir AP'den oluşan ortamlar için tasarlanmış çözümlerden oluşmaktadır. Standart dışı yöntemler son kullanıcıya eklenti kurulması gerektirdiği için halka açık alanda uygulanabilirliğini yitirmektedir. AP'ye bağlı kullanıcıların hizmet önceliğinin göz ardı edilmesi, önerilen çözümlerin, VOIP gibi hizmet kalitesinin ön planda olduğu kablosuz ağlarda etkili olarak kullanılamamaktadır. Ayrıca çalışmaların çoğunluğu simülasyon ortamında gerçekleştirilmiş olup gerçek kullanıcı ve donanımlar ile testleri yapılmamıştır.

Bu tez çalışmasında, literatürdeki eksikleri kapatmak ve kablosuz ağlardaki yük dengeleme problemlerine çözüm bulmak amacıyla QoS destekli dinamik yük dengeleme algoritmaları üzerinde çalışmalar yapılmıştır. Hazırlamış olduğumuz yöntem CB ile AP'lerin paket iletim güç seviyesini yapılandırmaktadır ve bu sayede istemci ile AP arasındaki bağlantılar standart yöntemler ile son kullanıcıya ek bir yük getirmeden kontrol edilebilmektedir. AP'lerin kullanıcılar ile olan bağlantısı ve tüm AP'lere bağlı olan kullanıcıların QoS yapılandırmaları da göz önüne alınarak gerçekleştirilmektedir.

Geliştirilen optimizasyon algoritmaları önce simülasyon ortamında test edilerek, ağ üzerinde hizmet alan kullanıcıların deneyimlediği ağırlıklı throughput ve toplam throughput artırılarak performans artışları gözlemlenmiştir. Daha sonra, TestBed ortamında test edilerek gerçek ortam sonuçları değerlendirilerek ağ optimizasyonu açısında olumlu sonuçlar elde edilmiştir.

Gelecekte yapılacak olan çalışmalarımızda mevcutta kullanılan yük dengeleme algoritmaları, frekans planlama çalışmalarıyla desteklenerek ağ üzerindeki optimizasyon çalışmalarına devam edilecektir. Bunu için merkezi bir frekans planlaması algoritması geliştirilecek ve bu algoritma ile IEEE 802.11 destekli kablosuz cihazlarla donatılmış bir

ağda, WTP'ler arasında frekans çakışmalarını minimize eden bir çalışma yapılması planlanmaktadır. Geliştirilecek olan yöntem öncelikli olarak simülasyon ortamında sonrasında ise hazırlamış olduğumuz TestBed ortamında gerçek testlerinin yapılması öngörülmektedir.

Tez çalışmasında hazırlamış olduğumuz Erişim Noktası Yöneticisi, öncelikli olarak IEEE 802.11 destekli kablosuz ağ kullanımı için geliştirilmiştir. Gelecek çalışmalarımızda Erişim Noktası Yöneticisinin kapsamı genişleterek, farklı kablosuz teknoloji altyapılarını yönetilebilir hale getirilmesi planlanmaktadır. CAPWAP protokolü tasarım amacıyla, herhangi bir kablosuz teknolojiye bağımlılığı bulunmamaktadır. Bu sebeple, farklı teknolojileri içeren heterojen ağlarda da kullanımına olanak sunmaktadır. WiFi, Bluetooth, ZigBee… vb. heterojen teknolojilerden oluşan ağ altyapılarının merkezi bir noktadan yapılmasına ve yönetilmesine imkan sunacak bir sistem geliştirilmesi planlanmaktadır.

Farklı teknolojilerin farklı yönetimsel ihtiyaçları barındırmasından dolayı, yapılacak çalışmalarda her bir teknoloji adım adım incelenerek adaptasyon için gerekli yazılımsal geliştirmeler yapılması hedeflenmektedir. Geliştirme süreçleri tamamlandıktan sonra farklı donanımlar üzerinden testleri yapılmasının ardından, IEEE 802.11 ağlarını da barından kablosuz cihazlardan oluşan bir heterojen TestBed ortamı kurulması planlanmaktadır. Hazırlanacak olan heterojen TestBed ortamı farklı algoritmaların geliştirilebildiği ve gerçek ortam testlerinin yapılabildiği bir mimaride olması planlanmaktadır.

KAYNAKLAR

[1]. Case J., Fedor M., Schoffstall M. L. and Davin J., 1990, Simple Network Management Protocol (SNMP), *The Internet Engineering Task Force*, IETF.

[2]. Waldbusser S., Cole R., Kalbfleisch C. and Romascanu D., 2003, Introduction to the Remote Monitoring (RMON) Family of MIB Modules, *The Internet Engineering Task Force*, IETF.

[3]. Warrier U. S. and Besaw L., 1989, Common Management Information Services and Protocol over TCP/IP (CMOT), *The Internet Engineering Task Force*, IETF.

[4]. Warrier U. S., Besaw L., LaBarre L. and Handspicker B. D., 1990, Common Management Information Services and Protocols for the Internet (CMOT and CMIP), *The Internet Engineering Task Force*, IETF.

[5]. Ylonen T. ve Lonvick C., 2006, The Secure Shell (SSH) Protocol Architecture, *The Internet Engineering Task Force*, IETF.

[6]. McKeown N., Anderson T., Balakrishnan H., Parulkar G., Peterson L., Rexford J., Shenker S. and Turner J., 2008, OpenFlow: Enabling Innovation in Campus Networks, *SIGCOMM Comput. Commun.* Rev., 38, 69-74.

[7]. Calhoun P., Montemurro M. and Stanley D., 2009, Control And Provisioning of Wireless Access Points (CAPWAP) Protocol Specification, *The Internet Engineering Task Force*, IETF.

[8]. IEEE Standard for Information technology--Telecommunications and information exchange between systems Local and metropolitan area networks--Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, 2012, *IEEE Std 802.11-2012 (Revision of IEEE Std 802.11-2007)*, 1-2793.

[9]. Blumenthal U. and Wijnen B., 2002, User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3), *The Internet Engineering Task Force*, IETF.

[10]. Waldbusser, S., 2006, Remote Network Monitoring Management Information Base Version 2, *The Internet Engineering Task Force*, IETF.

[11]. Calhoun P., Montemurro M. and Stanley D., 2009, Control and Provisioning of Wireless Access Points (CAPWAP) Protocol Binding for IEEE 802.11, *The Internet Engineering Task Force*, IETF.

[12]. Kim H. and Feamster N., 2013, Improving network management with software defined networking, *IEEE Communications Magazine*, 51, 114-119.

[13]. Yan M., Casey J., Shome P., Sprintson A. and Sutton A., 2015, ÆtherFlow: Principled Wireless Support in SDN, *IEEE 23rd International Conference on Network Protocols (ICNP)*, San Francisco, CA, 432-437.

[14]. Casey C. J., Sutton A. and Sprintson A., 2014, tinyNBI: Distilling an API from Essential OpenFlow Abstractions, *Proceedings of the Third Workshop on Hot Topics in Software Defined Networking*, New York, NY, USA, 37-42.

[15]. Dely P., Vestin J., Kassler A., Bayer N., Einsiedler H. and Peylo C., 2012 *CloudMAC - An OpenFlow based architecture for 802.11 MAC layer processing in the cloud*, IEEE Globecom Workshops, Anaheim, CA, 186-191.

[16]. Nakauchi K., Lei Z., Shoji Y., Kitatsuji Y., Ito M. and Yokota H., 2014, Bring your own network - Design and implementation of a virtualized WiFi network, *IEEE 11th Consumer Communications and Networking Conference (CCNC)*, Las Vegas, NV, 483-488.

[17]. Patro, A. and Banerjee S., 2015, Outsourcing coordination and management of home wireless access points through an open API, *IEEE Conference on Computer Communications (INFOCOM)*, Kowloon, 1454-1462.

[18]. Soetens N., Famaey J., Verstappen M. and Latré S., 2015, SDN-based management of heterogeneous home networks, *Network and Service Management (CNSM)*, Barcelona, 402-405.

[19]. Syrivelis D., Paschos G. S. and Tassiulas L., VirtueMAN: A software-defined network architecture for WiFi-based metropolitan applications, *IEEE 19th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, Athens, 95-99.

[20]. Stiti O., Braham O. and Pujolle G., 2015, Virtual openflow-based SDN Wi-Fi access point, *Global Information Infrastructure and Networking Symposium (GIIS)*, Guadalajara, 1-3.

[21]. Yamasaki Y., Miyamoto Y., Yamato J., Goto H. and Sone H., 2011, Flexible Access Management System for Campus VLAN Based on OpenFlow, IEEE/IPSJ 11th International Symposium on *Applications and the Internet*, Munich, Bavaria, 347-351.

[22]. Villain B., Ridoux J., Rotrou J. and Pujolle G., 2014, Mutualized OpenFlow architecture for network access management, *IEEE 3rd International Conference on Cloud Networking (CloudNet)*, CloudNet, 413-419.

[23]. Sun G., Liu G. and Wang Y., 2014, SDN architecture for cognitive radio networks, *2014 1st International Workshop on Cognitive Cellular Systems (CCS)*,Germany, 1-5.

[24]. Zubow A., Döring M., Chwalisz M. and Wolisz A., 2015, A SDN approach to spectrum brokerage in infrastructure-based Cognitive Radio networks, *IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, Stockholm, 375-384.

[25]. Raschellà A., Bouhafs F., Seyedebrahimi M., Mackay M. and Shi Q., 2016, A centralized framework for smart access point selection based on the Fittingness Factor, *23rd International Conference on Telecommunications (ICT)*, Thessaloniki, 1-5.

[26]. Wu J., Chen M., Hu C. and Zhang G., 2015, FC-WiFi: An OpenFlow Based WiFi Network with Free Configuration, *Ninth International Conference on Frontier of Computer Science and Technology*, Dalian, 52-58.

[27]. Rangisetti A. K., Baldaniya H. B., P. K. B and Tamma B. R., 2014, Load-aware hand-offs in software defined wireless LANs, *IEEE 10th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, Larnaca, 685-690.

[28]. Vestin J. and Kassler A., 2015, QoS enabled WiFi MAC layer processing as an example of a NFV service, *Proceedings of the 2015 1st IEEE Conference on Network Softwarization (NetSoft)*, London, 1-9.

[29]. Chu Y. H., Lin W. T., Wang Y. C., Hsieh C. T., Yang Y. L. and Cheng K. M., 2014, Software-defined QoE measurement architecture, *The 16th Asia-Pacific Network Operations and Management Symposium*, Hsinchu, 1-4.

[30]. Bernaschi M., Cacace F., Iannello G., Vellucci M. amd Vollero L., 2009, OpenCAPWAP: An open source CAPWAP implementation for the management and configuration of WiFi hot-spots, *Computer Networks*, 53, 217-230.

[31]. Bernaschi M., Cacace F., Davoli A., Guerri D., Latini M. and Vollero L., 2011, A CAPWAP-based solution for frequency planning in large scale networks of WiFi Hot-Spots, *Computer Communications*, 34, 1283-1293.

[32]. Bernaschi M., Cacace F., Iannello G., Vellucci M. and Vollero L., 2008, OpenCAPWAP: an open-source CAPWAP implementation for management and QoS support, *4th International Telecommunication Networking Workshop on QoS in Multiservice IP Networks*, Venice, 72-77.

[33]. Levanti A., Giordano F. and Tinnirello I., 2007, A CAPWAP Architecture for Automatic Frequency Planning in WLAN, *12th IEEE Symposium on Computers and Communications,* Las Vegas, NV, 51-56.

[34]. Levanti A., Giordano F. and Tinnirello I., 2007, A CAPWAP-Compliant Solution for Radio Resource Management in Large-Scale 802.11 WLAN, *IEEE GLOBECOM 2007 - IEEE Global Telecommunications Conference, Washington, DC,* 3645-3650.

[35]. Clancy T. C., 2008, Secure handover in enterprise WLANs: capwap, hokey, and IEEE 802.11R, *IEEE Wireless Communications,* 15, 80-85.

[36]. Sarikaya B., Zheng X. and Sarikaya B., 2006, *CAPWAP Handover Protocol*, *IEEE International Conference on Communications, Istanbul,* 1933-1938.

[37]. Bahl P., Hajiaghayi M. T., Jain K., Mirrokni S. V., Qiu L. and Saberi A., 2007, Cell Breathing in Wireless LANs: Algorithms and Evaluation, *IEEE Transactions on Mobile Computing*, 6, 164-178.

[38]. Bejerano Y., Han S. J. and Bejerano Y., 2009, Cell Breathing Techniques for Load Balancing in Wireless LANs, *IEEE Transactions on Mobile Computing*, 8, 735-749.

[39]. Song W., Zhuang W. and Cheng Y., 2007, Load balancing for cellular WLAN integrated networks, *IEEE Network*, 21, 27-33.

[40]. Yanmaz E. and Tonguz O. K., 2004, Dynamic load balancing and sharing performance of integrated wireless networks, *IEEE Journal on Selected Areas in Communications,* 22, 862-872.

[41]. Kim H., Veciana G. de, Yang X. and Venkatachalam M., 2012, Distributed alpha -Optimal User Association and Cell Load Balancing in Wireless Networks, *IEEE/ACM Transactions on Networking,* 20, 177-190.

[42]. Tonguz O. K., Yanmaz E. and Tonguz O.K., 2008, The Mathematical Theory of Dynamic Load Balancing in Cellular Networks, *IEEE Transactions on Mobile Computing,* 7, 1504-1518.

[43]. Bejerano Y., Han S. J. and Li, L., 2007, Fairness and Load Balancing in Wireless LANs Using Association Control, *IEEE/ACM Transactions on Networking*, 15, 560-573.

[44]. Gong H., Kim J. and Gong H., 2008, Dynamic load balancing through association control of mobile users in WiFi networks, *IEEE Transactions on Consumer Electronics,* 54, 342-348.

[45]. Park, J.S., Han S.J. and Han S.J., 2008, Load Balancing for Video Streaming Services in Hierarchical Wireless Networks, *Computer Networks*, 52, 259-274.

[46]. Muñoz P., Barco, R. and Bandera, I., 2013, Optimization of load balancing using fuzzy Q-Learning for next generation wireless networks, *Expert Systems with Applications*, 40, 984-994.

[47]. Rescorla, E. and Modadugu, N., 2012, Datagram Transport Layer Security Version 1.2, *The Internet Engineering Task Force*, IETF.

[48]. Banchs, A. and Vollero, L., 2006, Throughput analysis and optimal configuration of 802.11e EDCA, *Computer Networks*, 50, 1749-1768.

[49]. Serrano, P., Banchs, A., Patras P., and Azcorra, A., 2010, Optimal Configuration of 802.11e EDCA for Real-Time and Data Traffic, *IEEE Transactions on Vehicular Technology*, 59, 2511-2528.

[50]. Goldberg, D.E.,1898, Genetic Algorithms in Search, Optimization and Machine Learning, Boston, *MA: Addison-Wesley Longman Publishing Co., Inc*.

[51]. Ronco C. C. Da and Benini E., 2012, A Simplex-Crossover-Based Multi-Objective Evolutionary Algorithm, *IAENG Transactions on Engineering Technologies: Special Issue of the World Congress on Engineering and Computer Science*, ISBN: 978-94-007-6818-5, 583-598

[52]. Lawler E. L. and Wood D. E., 1966, Branch-and-Bound Methods: A Survey, *Oper. Res.*, 14, 699-719.

[53]. Ertürk, Mehmet Ali and Vollero, Luca and Aydin, Muhammed Ali and Turna, Özgür Can and Bernaschi, Massimo, 2014, A framework for modeling and implementing QoS-aware load balancing solutions in WiFi hotspots, *11th International Symposium on Wireless Communications Systems (ISWCS)*, Barcelona, 33-38.

[54]. Bianchi, G., Fratta L. and Oliveri M., 1996, *Performance evaluation and enhancement of the CSMA/CA MAC protocol for 802.11 wireless LANs. Seventh IEEE International Symposium on In Personal, Indoor and Mobile Radio Communications*, Taipei, 2, 392–396.

[54]. Erturk, M.A., Vollero, L., Aydin, M.A., (Submitted), Optimal Joint Load Balancing and EDCA Configuration of IEEE 802.11 Wireless Hotspots, *International Journal of Network Management*

[55]. Erturk, M.A., Vollero, L., Aydin, M.A., (Submitted), Vendor Neutral Wireless LAN Controller, *Elsevier Computer Communications*

ÖZGEÇMİŞ

Kişisel Bilgiler	
Adı Soyadı	Mehmet Ali ERTÜRK
Doğum Yeri	İzmir
Doğum Tarihi	02.12.1983
Uyruğu	☑ T.C. ❒ Diğer:
Telefon	0 212 440 00 00
E-Posta Adresi	mehmetali.erturk@istanbul.edu.tr
Web Adresi	http://mehmetalierturk.com

Eğitim Bilgileri	
Lisans	
Üniversite	Doğuş Üniversitesi
Fakülte	Mühendislik Fakültesi
Bölümü	Bilgisayar Mühendisliği
Mezuniyet Yılı	28.06.2007

Yüksek Lisans	
Üniversite	İstanbul Üniversitesi
Enstitü Adı	Fen Bilimleri Enstitüsü
Anabilim Dalı	Bilgisayar Mühendisliği Anabilim Dalı
Programı	Bilgisayar Mühendisliği Programı
Mezuniyet Tarihi	28.12.2010

Doktora	
Üniversite	İstanbul Üniversitesi
Enstitü Adı	Fen Bilimleri Enstitüsü
Anabilim Dalı	Bilgisayar Mühendisliği Anabilim Dalı
Programı	Bilgisayar Mühendisliği Programı
Mezuniyet Tarihi	12.12.2016

Makale ve Bildiriler
[1].Erturk, M.A., Vollero, L. and Aydin, M.A., (Submitted), Vendor Neutral Wireless LAN Controller, *Elsevier Computer Communications*.

[2]. Erturk, M.A., Vollero, L. and Aydin, M.A., (Submitted), Optimal Joint Load Balancing and EDCA Configuration of IEEE 802.11 Wireless Hotspots, *International Journal of Network Management*.

[3]. Erturk M.A., Aydin M.A., Ibali H.I., Gurkas Aydin G.Z. and Zaim A.H., Service Oriented Mobile Wallets with NFC, *International Journal of Applied Mathematics, Electronics and Computers*, 4, 83-86

[4]. Erturk, M.A., Vollero, L., Aydin, M.A., Turna, O.C. and Bernaschi, M., A framework for modeling and implementing QoS-aware Load Balancing solutions in WiFi Hotspots, *The Eleventh International Symposium on Wireless Communication Systems*, ISWCS 2014, Barcelona, Spain, 33-38.

[5]. Erturk M.A., Zaim A.H., Aydın, M.A., Akyokus, S., Integrating NFC with Cloud Computing in Healthcare Systems, *International Conference on Networking and Future Internet*, ICNFI 2012, April 2012, Istanbul, Turkey.

[6]. Erturk M.A., Zaim A.H., Akyokus, S., Semantic Information Retrieval on Peer-to-Peer Networks, *The Second International Conference on 'Networked Digital Technologies*, (NDT 2010), Prague, Czech Republic.

[7]. Turna O.C., Yuksel M.E., Erturk M.A., Bilgiye Erisim Sistemlerinde Veri Arama ve Eslestirme, *Akademik Bilisim*, 2010, Mugla, Turkey

[8]. Erturk M.A., Zaim A.H., Akyokus, S., Document Searching and Matching on Unstructured Turkish Documents, (AAS 2009), Ohrid, Macedonia