ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE ENGINEERING AND TECHNOLOGY

GRADE OF SERVICE (GoS) BASED CONTROLLER FRAMEWORK FOR SOFTWARE DEFINED HETEROGENEOUS NETWORKS (SDHetN)

M.Sc. THESIS

Müge EREL

Department of Computer Engineering

Computer Engineering Programme

MAY 2015

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE
ENGINEERING AND TECHNOLOGY

GRADE OF SERVICE (GoS) BASED CONTROLLER FRAMEWORK FOR SOFTWARE DEFINED HETEROGENEOUS NETWORKS (SDHetN)

M.Sc. THESIS

Müge EREL
(504131534)

Department of Computer Engineering

Computer Engineering Programme

Thesis Advisor: Asst. Prof. Dr. Berk CANBERK

MAY 2015

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

YAZILIM TANIMLI HETEROJEN AĞLAR İÇİN SERVİS DERECESİ TABANLI KONTROLÖR YAPISI

YÜKSEK LİSANS TEZİ

Müge EREL
(504131534)

Bilgisayar Mühendisliği Anabilim Dalı

Bilgisayar Mühendisliği Programı

Tez Danışmanı: Asst. Prof. Dr. Berk CANBERK

MAYIS 2015

Müge EREL, a M.Sc. student of ITU Graduate School of Science Engineering and Technology 504131534 successfully defended the thesis entitled **"GRADE OF SERVICE (GoS) BASED CONTROLLER FRAMEWORK FOR SOFTWARE DEFINED HETEROGENEOUS NETWORKS (SDHetN)"**, which he/she prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor :	**Asst. Prof. Dr. Berk CANBERK** Istanbul Technical University	....
Jury Members :	**Prof. Dr. Özgür Barış Akan** Koç University	....
	Assoc. Prof. Dr. Güneş Zeynep Karabulut Kurt Istanbul Technical University	....
		....

Date of Submission : **4 May 2015**
Date of Defense : **27 May 2015**

To my spouse and parents

FOREWORD

This thesis is written as completion to the master of Computer Engineering, at Istanbul Technical University. The studies during master period, were based on Software Defined Network (SDN) paradigm. SDN is one of the popular next generation technology that makes easy to manage whole network from centralized view. Because of promising better performance for network management and being suitable to solve problem with written algorithms, I decided to study on this technology.

At the beginning of master, I had a lack of knowledge about academic writing. How conference or journal papers should emphasize proposed approaches? How contributions can be expressed in one sentence briefly and clearly? What are the main properties that reviewers pay attention in submitted papers? Now, I have good experience about them with my published publications, my membership of Technical Program Committee (TPC) in conferences and my reviews in journals. I would like to thank my supervisor, Asst. Prof. Dr. Berk CANBERK from Computer Engineering Department of Istanbul Technical University, to give me this chance and endeavor by concerning each detail about my academic career. He was always leading me, coming up with full energy and having another plan if something was going wrong.

I would thank to my parents for all their devotion in my whole education life. They were always patient when I had to study during nights. Thanks them for being next to me for each difficult time. Moreover, I would also thank to my father and mother in law. Although we have recognized each other for a few years, they take me for their own girl since the first time. Thanks them for their sincere behaviors and all support for each time. Furthermore, I would thank to my colleagues for their nice conversation in our breaks. These small breaks during day, was increasing our energy and motivation to our studies. Thanks them to be near me for all time.

Finally, I would thank to owner of my heart, Yusuf. With his calm and smiling face, I could begin with full motivation to my day. I am grateful for his inexhaustible patience and struggle to get me laugh at any time in my life. I couldn't do these all things without him, thanks to make life livable for me.

May 2015

Müge EREL
Research and Teaching Assistant

TABLE OF CONTENTS

Page

ABBREVIATIONS

AP	**:** Access Point
CON	**:** Conventional
FACA	**:** Flow Admission Control Algorithm
FAVS	**:** Flow Authority Virtual Switch
FAVSF	**:** Flow Authority Virtual Switch Farm
FMT	**:** Flow Management Table
GoS	**:** Grade of Service
HetNet	**:** Heterogeneous Network
OF	**:** OpenFlow
QoS	**:** Quality of Service
RAT	**:** Radio Access Technology
SDHetN	**:** Software Defined Heterogeneous Network
TCA	**:** Topology Control Algorithm

LIST OF TABLES

LIST OF FIGURES

Page

LIST OF SYMBOLS

i	:	The index of i^{th} flow
j	:	The index of j^{th} OpenFlow(OF) switch in the Data Plane
t	:	The index of t^{th} Flow Authority Virtual Switch(FAVS) in Flow Authority Virtual Switch Farm(FAVSF)
m	:	Total number of OF switches in the Data Plane
m'	:	Total number of FAVS in FAVSF
n_j	:	The number of traffic flows that are subscribed to j^{th} OF switch
n_t	:	The number of traffic flows that are subscribed to t^{th} FAVS
N	:	Total number of traffic flows in the Data Plane
C_j	:	The number of physical resources of j^{th} OF switch
C_t	:	The number of physical resources of t^{th} FAVS
C	:	Total number of C_j in the topology
ρ_{ij}	:	The intensity of individual i^{th} flow of which destination is j^{th} OF switch (Erlang)
ρ_{it}	:	The intensity of individual i^{th} flow of which destination is t^{th} FAVS (Erlang)
ρ_j	:	Total intensity of flows towards j^{th} OF switch (Erlang)
ρ_t	:	Total intensity of flows towards t^{th} FAVS (Erlang)
GoS_j	:	The Grade of Service of j^{th} OF switch
GoS_t	:	The Grade of Service of t^{th} FAVS
λ_j	:	Total arrival rate of flows that comes to j^{th} OF switch (flow/sec)
μ_j	:	The service rate of j^{th} OF switch (flow/sec)
μ_s	:	The service rate of proposed system Controller (flow/sec)
s_j	:	The state of j^{th} OF switch as busy (1) or not (0)
$E[N]$	:	The expected number of flows handled in the system
F_{SDHetN}	:	The Fairness Index of proposed SDHetN system
F_{CON}	:	The Fairness Index of Conventional system
ψ	:	The Resource Efficiency
γ	:	The Acceleration Factor of Controller
$\mu_s^{'}$	:	The Accelerated Service Rate of Controller with helps of γ (flow/sec)

GRADE OF SERVICE (GoS) BASED CONTROLLER FRAMEWORK FOR SOFTWARE DEFINED HETEROGENEOUS NETWORKS (SDHetN)

SUMMARY

Recently, mobile data traffic has rapid proliferation due to increased user demands in wireless networks. The conventional macrocells are stressed because of huge increase in user requests. Therefore, they are not adequate to serve indoor users and keep their service quality, i.e. Grade of Service (GoS), in acceptable levels.

In order to meet indoor mobile users with higher service quality, i.e. lower GoS, heterogeneous networks (HetNets) are deployed as smallcells, vary according to coverage area such as macrocell, picocell and femtocell etc. However, conventional HetNets cannot handle increased user flow requests by using same number of physical resources, due to inefficient usage of these resources.

The physical resources are clustered and distributed to smallcells statically in conventional HetNets by the operators. Therefore, this static clustering leads an ineffective resource usage. In other words, the number of flows that can be served with acceptable GoS, is restricted into low levels. For example, using two smallcells, each having ten physical resources, can handle more traffic flows than using four smallcells, each having five physical resources, while considering same GoS level. In other words, physical resource efficiency has huge decrease if resources are statically clustered and distributed to small cells. This is also proved by modeling two systems using markov chain and mathematically analyzed according to queuing theory as given in Appendix.

Current solutions that try to increase resource efficiency in the literature, are based on switching technologies between radio access technologies (RATs) such as offloading, and cognitive radio technologies etc. Nevertheless, these solutions cannot reach high number of flows that can be served in acceptable level due to limitation of statically assigned physical resources. Therefore, the restriction caused by static clustering of physical resources and distribution to smallcells, should be removed with adaptive clustering approach.

In this paper, we solve this ineffective static resource assignment by proposing a novel queuing-theoretic Software Defined HetNet (SDHetN) model which orchestrates topology using adaptive and scalable flow management heuristics. The proposed SDHetN takes its flexible and scalable characteristics thanks to two algorithms; the Topology Control Algorithm (TCA) and the Flow Admission Control Algorithm (FACA). Specifically, the proposed TCA clusters several OpenFlow (OF) switches using the flows' GoS in order to optimize physical resource assignment. The proposed FACA fairly distributes each Flow Authority Virtual Switch (FAVS) that are created in TCA by grouping several switches virtually. We also propose a thread-based parallelization in TCA and FACA increasing the response time and service rate of the SDHetN Controller.

YAZILIM TANIMLI HETEROJEN AĞLAR İÇİN SERVİS DERECESİ TABANLI KONTROLÖR YAPISI

ÖZET

Son zamanlarda, mobil cihaz kullanımı oldukça artmaktadır. Bu nedenle, mobil kullanıcıların kablosuz hücresel ağlardaki istekleri büyük bir artış göstermektedir. Bu artış, hareketli veri trafiğinin büyük ölçüde artmasına sebep olmaktadır. Cisco görsel ağ indeks raporunda (Cisco Visual Networking Indeks Report) belirtildiğine göre, 2013 yılında ölçülen veri trafiği ayda 1.5 exabyte iken, bu değerin 2018 yılında 15.9 exabyte seviyesine ulaşacağı tahmin edilmektedir. Başka bir deyişle, 2013 yılına göre veri trafiğinde 11 kat artış yaşanması öngörülmektedir. Bu hızla artan trafik yoğunluğu nedeniyle, hareketli veri trafiğine hizmet veren geleneksel kablosuz haberleşme ağının makro hücreleri (macrocell), kullanıcılara hizmet veremez hale gelmektedir. Böylece, makro hücreleri özellikle ev içi kullanıcılara yeterli sinyal gücüyle ulaşamamakta ve bu kullanıcıların servis kaliteleri (Grade of Service- GoS) kabul edilemeyecek seviyelere düşmektedir.

Ev içi kullanıcılara daha iyi hizmet kalitesi ile servis sağlayabilmek amacıyla heterojen ağ (HetNet) teknolojisi ortaya çıkmıştır. Heterojen ağlarda farklı özellikleri bulunan küçük hücreler (smallcells) bulunmaktadır. Bunlar kapsama alanlarının büyüklüğüne göre, makro hücre (macrocell), piko hücre (picocell), ve femto hücre (femtocell) olarak isimlendirilmiştir. Farklı kapsama alanına sahip bu hücreler ile kablosuz ağların oluşturulması, ev içi kullanıcıların daha yüksek sinyal gücüne erişebilmesini sağlamıştır. Böylece bu kullanıcılar kablosuz ağa dahil olup iletişime geçebilmektedirler. Ancak heterojen ağlarda fiziksel kaynaklar verimsiz kullanılmaktadır. Başka bir deyişle, var olan fiziksel kaynaklar ile daha fazla kullanıcıya daha yüksek kalitede hizmet verilebilecek iken, heterojen ağ yapısından kaynaklı olarak daha az kullanıcıya hizmet sunulmakta ve bu kullanıcıların GoS seviyeleri artan kullanıcı istekleri karşısında kabul edilemeyecek seviyelere yükselmektedir (GoS yükseldikçe sunulan servis kalitesi düşer). Bu durumun temel nedeni Heterojen ağlardaki statik kaynak atamasıdır.

Heterojen ağlarda fiziksel kaynaklar gruplandırılıp statik bir şekilde küçük hücrelere dağıtılmaktadır. Ancak, bu statik gruplama nedeniyle kablosuz ağın hizmet verebileceği trafik akışı sayısı düşük seviyelerde kalmaktadır. Bu da yeterli fiziksel kaynağa sahip bir ağda , kaynakların verimsiz kullanımının sonucudur. Bu durumu sayısal bir örnekle açıklamak gerekirse; toplamda 20 fiziksel kaynağa sahip bir ağda, her birinde 10'ar kaynak bulunduran iki küçük hücre ile kurulmuş topolojinin hizmet vereceği trafik akışı sayısı, her birinde beşer kaynak bulunduruan dört küçük hücre ile kurulmuş topolojinin hizmet vereceği trafik akışı sayısından oldukça fazladır. Başka bir yönden bakmak gerekirse, aynı akış sayısına sahip bu iki topolojide, belirtilen ilk topoloji kullanıcılarına daha yüksek seviyede servis kalitesi (daha düşük GoS) sağlamaktadır. Özetle, fiziksel kaynakların fazla sayıda küçük hücrelere gruplanıp dağıtılması, bu kaynakların verimliliğini oldukça düşürmektedir. Fiziksel kaynak

sayısı değiştirilmeden, kabul edilebilir servis kalitesinde daha fazla kullanıcıya ya da daha düşük GoS (daha yüksek servis kalitesi) ile aynı sayıda kullanıcıya hizmet verilmesi sağlanabilmelidir. Bu da iki topoloji arasında kaynak verimliliğini ortaya net bir biçimde koymaktadır. Bu iki topoloji arasındaki fark, markov zinciri (markov chain) ile modellenmiş ve kuyruklama teorisi kullanılarak (queuing theory) matematiksel ifadelerle, ekte belirtildiği gibi kanıtlanmıştır.

Literatürde kaynak verimliliğini arttırmak için sunulan günümüzdeki çözümler, radyo erişim teknolojileri (radio access technologies- RATs) arasında geçiş tabanlıdır. Bu çözümler bilişsel radyo teknolojisi (cognitive-radio) ve bir hücreden başka bir hücreye geçiş (offload) teknolojisi olarak örneklendirilebilir. Bilişsel radyo teknolojisinde, iki tip kullanıcı vardır. Bunlardan biri frekansın sahibi olan birincil kullanıcı (primary user) ve o anda kullanımda olmayan frekansları ücret ödemeden ve birincil kullanıcının servis kalitesinin de etkilemeyecek şekilde kullanabilen ikincil kullanıcıdır (secondary user). Frekans sahibi olan birincil kullanıcının servis kalitesini düşürmeyecek şekilde ve bu frekansı o an için kullanan başka bir ikincil kullanıcı yok ise, ikincil kullanıcı bu frekans bandına zıplayıp, bu frekans üzerinden iletişimini gerçekleştirebilmektedir. Böylelikle ikincil kullanıcı kullanılmayan frekansları kullanarak hem frekans verimliliğini arttırır hemde kendi servis kalitelerini yüksek seviyelerde tutma imkanı bulur. Ancak bilişsel radyoda ikincil kullanıcı zıplayacağı frekans durumunu doğru şekilde belirleyebileceği sezme algoritmalarına ihtiyaç duyar. Zıplanılacak frekans sahibinin o an için frekans bandını kullanıyor olup olmadığının yanlış tahmin edilmesi durumu hem birincil kullanıcı hemde ikincil kullanıcı açısından servis kalitesi düşüklüğüyle sonuçlanacaktır. Bu nedenle sezme algoritmalarının profesyonel şekilde tasarlanması gerekmektedir. Bu durumda son kullanıcı olan ikincil kullanıcıları daha karmaşık bir yapı kullanmak zorunda bırakır. Farklı frekanslarda çalışan iki ayrı hücre arasındaki geçiş (offload) teknolojisinde ise, servis kalitesi kötü olan hücreden kullanıcı kendi kararıyla servis kalitesini yükseltebileceğini düşündüğü farklı bir frekansta çalışan daha az yoğun diğer hücreye geçiş yapabilmektedir. Böylece kullanıcı az yoğunluğa sahip küçük hücrede iletişim kalitesini kabul edilebilir seviyede tutabilmektedir.

Hücreler arası geçiş hücre yoğunluğuna göre belirlenebildiği gibi son kullanıcının harcayacağı enerjiye göre de belirlenebilir. Yapılan tahminlere göre farklı frekanslarda çalışan iki hücre arası geçiş (Makro hücre ve WiFi v.b.), son kullanıcının hem erişim noktasına (Access Point) olan yakınlığı hem de kullandığı frekansı açısından daha az enerji harcamasını sağlayabilir. Bu da iki hücre arası geçişin önemini arttıran bir parametredir. Ancak bu çözümler kullanıcı kararı bazlı olup, topolojiye genel bakamadıkları için fiziksel kaynak verimliliğini bir dereceden sonra arttıramamaktadırlar. Yine yukarıda belirtilen fiziksel kaynak verimsizliği statik gruplama ve dağıtımdan kaynaklı olarak hizmet verilebilecek akış sayısını kısıtlamaktadır. Bu nedenle, fiziksel kaynakları verimli kullanabilmek ve hizmet verilebilcek akış sayısını arttırabilmek için, önerilen yeni dinamik kaynak gruplaması ve dağıtımı sayesinde fiziksel kaynakların yarattığı bu kısıt ortadan kaldırılmalıdır.

Bu tezde, statik kaynak atamasının verimsizliğini ortadan kaldıracak yeni bir yapı tasarlanmıştır. Yazılım Tanımlı Heterojen Ağ (Software Defined Heterogeneous Networks- SDHetN) olarak tanımladığımız bu model, heterojen ağ topolojisini adaptif ve esnek akış yönetimi tabanlı algoritmalar ile merkezi olarak yönetmektedir. Tez boyunca bu modelin veri katmanı (Data Plane) ve kontrol katmanı (Control Plane) kuyruk teorisi (queuing theory) ile modellenip, matematiksel olarak analiz edilmiştir.

Veri katmanında akış yönlendirme işlemini gerçekleştiren ve OpenFlow (OF) protokolü ile kontrol katmanıyla haberleşebilen fiziksel anahtarlar bulunmaktadır. Fiziksel anahtarların $M/M/C_j/K$ markov modeli şematize edilmiş ve trafik akışını bloklama olasılığının (blocking probability) matematiksel ifadesi çıkarılmıştır. Bu değer aynı zamanda trafik akışının servis kalitesini (GoS) belirlemektedir.

Kontrol katmanı ise m adet farklı $M/M/1/1$ sistemi ile modellenmiş ve Jackson teoremine göre sistemdeki trafik akış sayısı (expected number of user) değeri çözümlenmiştir. Bu katman temel iki bileşenden oluşmaktadır. Bunlar Akış Yöneten Sanal Anahtar Havuzu (Flow Authority Virtual Switch Farm- FAVSF) ve kontrol algoritmalarıdır. İlk kısım veri katmanındaki fiziksel anahtarların sanal temsilcilerini barındıran sanal anahtarların oluşturduğu havuzdur. İkinci kısım ise, SDHetN modeline esneklik özelliklerini katan algoritmalardan oluşmaktadır. Bu algoritmalar Topoloji Kontrol Algoritması (Topology Control Algorithm - TCA) ve Flow Kayıtlama Kontrol Algoritmasıdır (Flow Admission Control Algorithm - FACA). Özellikle, önerilen TCA, akışların servis kaliteleri (GoS) göz önünde bulundurularak birçok OF fiziksel anahtarlarını gruplar. Bu algoritmada fiziksel kaynak verimliliği, akışların GoS değerlerine ve oluşacak grup sayısına göre maksimize edilir. Önerilen FACA ise, trafik akışlarını TCA ile oluşturulan sanal anahtarlara (FAVS) dengeli bir biçimde dağıtır. Böylece ilk algoritmada yapılan gruplama sayesinde statik kaynak atamasının yarattığı kısıtlama ortadan kaldırılmış olur ve ikinci algoritma ile akışlar topolojideki sanal anahtarlara dengeli bir biçimde dağıtılarak akışların servis kalitesi (GoS) oldukça iyileşir. Bir diğer önerilen yapı ise, kontrolörde bu algoritmaların çalışmasının akışın performansı üzerinde negatif bir etki yaratmasını engellemek için, algoritmalar her bir sanal anahtar başına paralel olarak çalıştırılır. Böylece SDHetN kontrolörünün cevap süresi (response time) ve servis oranı (service rate) iyileşmektedir. SDHetN modelinin başarım analizi 48 farklı senaryo ile yürütülmüş ve önerilen SDHetN yapısının geleneksel yapıya göre daha esnek ve daha adil akış yönetimine sahip olduğu birçok performans çıktı grafiklerinde görülmektedir.

1. INTRODUCTION

In this thesis, a novel SDHetN Controller framework is examined by defining each significant components. First of all, SDHetN paradigm is introduced in this chapter by studying main challenge in HetNets, motivation and contribution of thesis. In chapter 2, the proposed SDHetN framework is defined by looking network architecture, Data and Control Plane models. In chapter 3, we offered general view on system models by comparing Conventional (CON) and SDHetN frameworks in terms of fairness expressions. The evaluation performance of the proposed scheme is also offered in that section. Finally, the thesis is summarized and concluded in chapter 4. This thesis also have Appendix section that includes theoretical proof of main challenge.

1.1 Background Information about HetNets

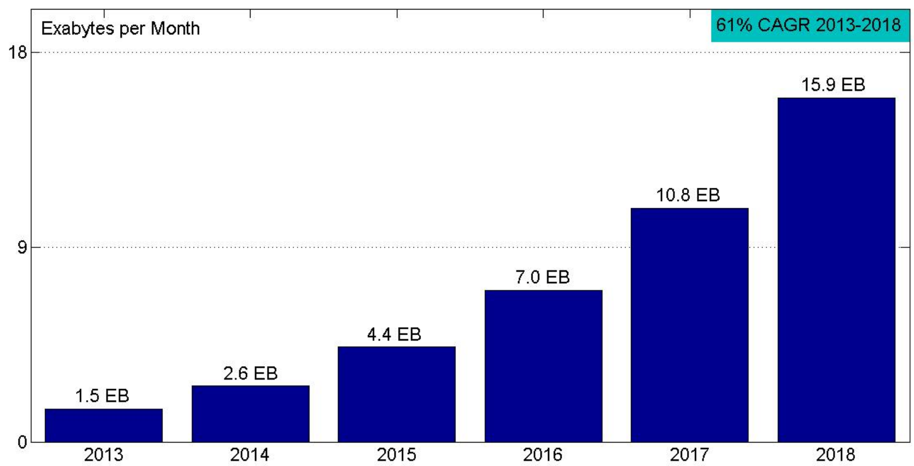


Figure 1.1: Growth of Mobile Data Traffic [1].

Today's rapid proliferation of mobile data traffic as well as the high increase in user service requests stress the conventional cellular wireless technology [3]. According to Cisco Visual Networking Index (VNI) report published in 2014, by 2018 the increase of global mobile data traffic is predicted to grow nearly an 11 fold increase over 2013 as seen in Figure 1.1. The amount of mobile data traffic was 1.5 exabytes per month, whereas it is expected to reach 15.9 exabytes per month by Cisco. With this explosion on mobile data traffic, various GoS demands of mobile users have also emerged, thereby stressing the macrocells to handle more users with heterogeneous GoS levels.

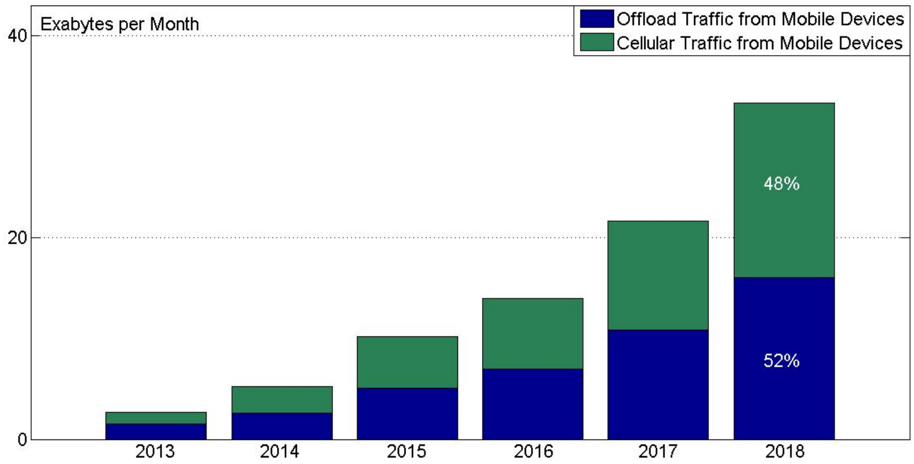


Figure 1.2: Offloaded Mobile Data Traffic [1].

The stressed conventional macro cells of cellular wireless technology is not enough to serve more users and keep their service quality in an acceptable level. Moreover, macro cells could not reach indoor mobile users with better service quality thereby increasing the number of dissatisfied users in indoor applications. To reach indoor mobile users with higher quality, different and novel management solutions such as small cells are to be deployed. The small cell deployment solution, which was proposed to handle the increased mobile traffic demands, bring an important mobile data offloading challenge with it. This can be explained as follows. One can say that the occurrence of mobile data traffic comes true mostly from indoor users [1]. Here, because of receiving higher power from smallcells and WiFi rather than macrocells inside homes, users prefers to offload their traffic to these radio access technologies (RATs) in order to have better GoS. According to same Cisco VNI report, the amount of transition from cellular technology to fixed broadband technologies such as small cells and WiFi is shown as seen in Figure 1.2. The explosion on mobile data traffic is also seen by this graph as indicated in Figure 1.1. Here, the offloaded mobile data is expected to increase from 45 % (1.2 exabytes per month) to 52 % (17.3 exabytes per months) by 2018. Therefore, the transitions between different RATs becomes more significant in next generation wireless heterogeneous networks.

The solutions based on switching among different RATs such as offloading techniques, cognitive radio and dynamic spectrum access technologies could not handle the increasing flow intensities after a certain scalability level, due to limitation in the statically assigned physical resources. Thus, the resource constraints occur and this degrades the overall resource efficiency of the system. In following subsection, the resource constraint will be defined and illustrated with a specific example.

1.2 The Main Challenge: Physical Resource Constraints in HetNets Deployments

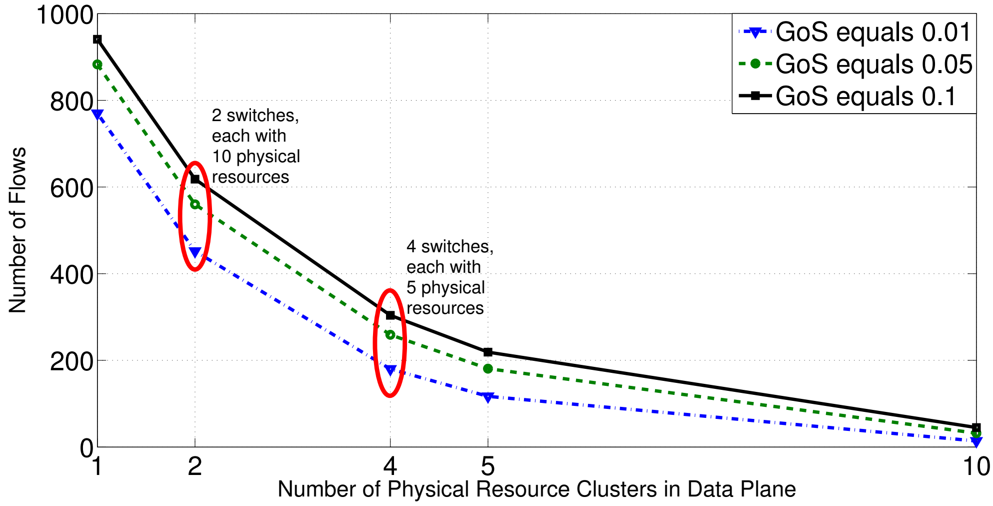


Figure 1.3: Clustering Effect into the Number of Flows under certain GoS values on typical HetNet Scenario.

In generic wireless architectures, the number of flows that a base station can handle is constant. They are statically clustered. This static clustering limits the serving flows by conceding GoSs of each flow. In Figure 1.3, the clustering effect on the number of flows is shown with respect to the blocking probabilities. a.k.a. GoS. Here, GoS is modeled by $M/M/C/K$ markov model [4] where C is the number of physical resources and $[K-C]$ is the queue length. The x-axis of Figure1.3 indicates different topologies with same number of physical resources. In each topology, there are different number of physical clusters, i.e. Access Points (APs) of HetNets. However, as seen in the figure, this clustering of physical resources decreases the number of flows that can be handled. For instance, while GoS parameter is 0.01 and there are 20 physical resources in total, approximately 180 flows can be served with 4 clusters (4 APs), each having 5 physical resources. By using 2 clusters (2 APs) instead of 4, 452 flows can be handled with the same GoS value [1]. Consequently, the resource efficiency decreases because of static clustering of physical resources. In other words, the resource efficiency is limited by physical resource constraints. In the following subsection, the recent technologies that try to increase resource efficiency, are examined in detail.

1.3 Literature Survey: Current Solutions in RATs

There exists many studies that try to overcome aforementioned challenge of HetNets by different approaches.

The offloading between different RATs offers suitable solution that increases resource efficiency and quality of service (QoS) of offloaded flow. [9] presents various offloading strategies to offload users from stressed macrocell to smallcells. [10] emphasizes that inefficiently resource utilization can be removed with their proposed model named as Network Virtualization Substrate (NVS) for effective virtualization of wireless resources in HetNets. In [11], the importance of Network Function Virtualization (NFV) is also clarified as offering rapid service, greater flexibility, improved operational efficiencies etc.

Another technique to increase resource efficiency is cognitive radio technologies. In [12], the authors study on spectrum sharing and power allocation in heterogeneous cognitive radio networks with energy efficiency perspective. The energy-efficient resource allocation is examined and formulated in heterogeneous cognitive radio networks with femtocells as a Stackelberg game. [13] also studies resource allocation optimization for one cell that subscribes multiuser for the case of the primary user existence in cognitive radio network. The authors in [14] propose location based solution to inefficient utilization of spectrum problem. The novel architecture is designed, which is called the Cognitive Capacity Harvesting network (CCH) that is an aggregation of relay stations which enhance the allocation for secondary users with cognitive capability.

[1]This decrease on the number of flows is also proved by applying queuing theory as seen in Appendix.

On the other hand, [15] tries to have scalable network by indicating an alternative resource allocation mechanism which guarantees a fully efficient allocation when users are price taking. [16] explains the significance of cooperative communication in resource constrained wireless networks. They give wide survey about optimal power allocation for various network topologies and propose cooperation scheme. In [17], a random access protocol that provides fair access to spectrum for different radio systems is proposed by modeling system with queueing theory. This is also extended to spectral agile radio in order to provide general model about dynamic spectrum access. Apart from these, there are studies that solve emphasized challenge by using flow based approaches. [18] solves trade-off between resource efficiency and QoS by proposing opportunistic link overbooking technique that is quality guaranteed in flow-level. In [19], the authors manages trade-off between resource efficiency and user fairness. They propose adaptive resource allocation method that consists of subcarrier assignment and power allocation algorithms. Besides flow-based solutions, self-organized networks also offers solution to resource efficiency problem of HetNets. Due to having self-management methods such as self-optimization and self-configuration, E3 project of [20] enhances wireless network efficiency by using cognitive, self-organized resource reconfiguration.

1.4 Motivation

The aforementioned technologies try to increase resource efficiency in the topology. However, they are not enough to meet increasing user demands after a certain scalability level. To overcome this challenge,

- Physical resource constraint should be isolated in order to handle more flows.
- Flows should be fairly distributed to physical resource clusters in order to
  - increase scalability in the topology.
  - enhance GoS per flow.

In the light of these motivations, a novel Controller framework that manages flows adaptively is proposed in Software Defined Heterogeneous Network (SDHetN).

1.5 Contribution: SDHetN paradigm

SDHetN redefines network management by separating Data Plane and centralized Controller in Control Plane [5]. Incoming flows are forwarded by OF switches[2] which have only forwarding capability. If a flow could not match any flow table entries in switch, it is defined as newcomer flow and so as to decide new forwarding rule, the first packet sends to Controller. In order to assign this newcomer flow in topology, Controller gives appropriate decision and updates flow tables in switches by considering fair flow distribution.

A Software Defined Network (SDN) is an architecture, decouples the Data Plane, which contains dummy devices to forward the traffic, and the Control Plane, which controls how the traffic will flow through the network. The motivation in SDN paradigm is unpredictable distributed network configurations, which all network devices individually configured [21] and [22]. For small-scale networks, the configurations are easy to handle, however when the scale gets larger, the network configurations get more complicated [23]. Besides the operational cost of the large-scale networks, the capital costs are the other reason to trend to SDN. Since the network logic is moved to the centralized Controller, hardware may get cheaper.

1.5.1 Components of SDHetN

In conventional networks, Infrastructure Layer (Data Plane) and Control Layer (Control Plane) are combined and Application Layer does not exist. With SDHetN, these three layers are separated and this separation allows the network operators to control the whole network behavior from a single control software. Therefore, this new SDHetN architecture moves the network intelligence to an application called Controller as seen in Figure 1.4.

- **The Infrastructure Layer:** This layer holds the network devices responsible for forwarding the traffic.
- **The Control Layer:** It contains the software logic which communicates with the Data Plane with an OF like software interface.

[2] The terms "OF switch" and "switch" will be used interchangeably throughout the thesis.

- **The Application Layer:** This layer consists of the end-user applications for network operators to configure the control software.

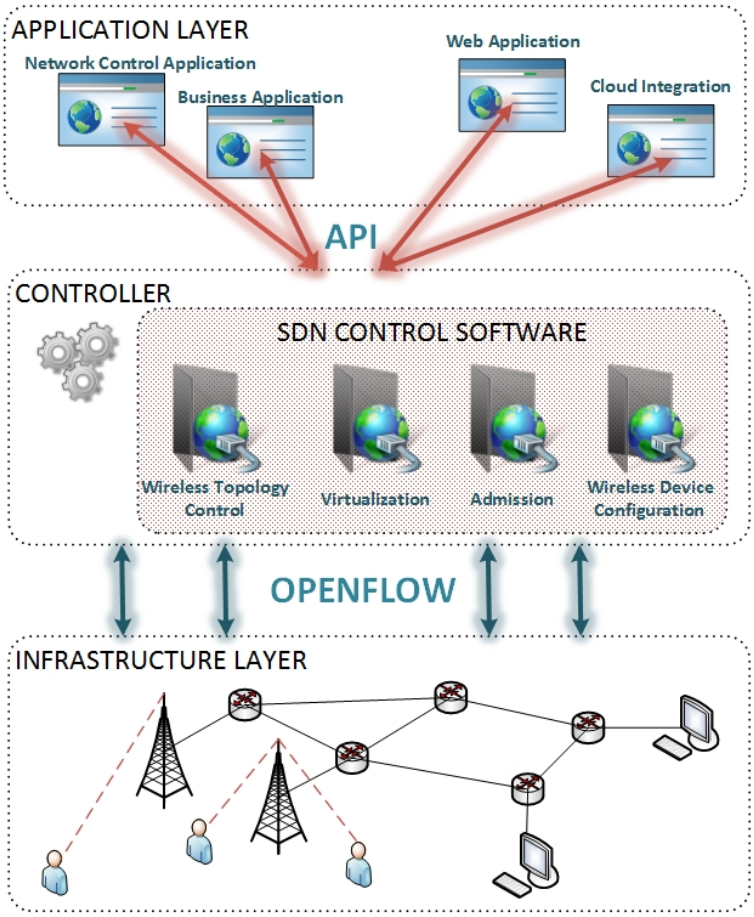


Figure 1.4: SDHetN Architecture [2].

SDHetN based architectures has the promise to redefine network management by decoupling the Control Plane and Data Plane and SDHetN has also emerged simplifying the configurations with abstraction of these two planes [3]. As indicated in [24] and [25], SDHetN simplifies the network management by dynamically managing the traffic isolation. Moreover, SDHetN offer opportunities to enable network operations on a logically centralized global network view [26]. Akyıldız et. al. indicates with the SDHetN approach network resource utilization is improved, network orchestration is simplified, and operating costs are reduced in [27]. These are also emphasized in [28] and [25] as SDHetN simplifies the deployment by decreasing total cost of managing enterprise and carrier networks. In [29], Kim et. al. indicate the continual network state, changes in real-world deployments and it is significant to emphasize that operators require SDHetN-based solutions to adapt their network to frequent alteration. Moreover, SDHetN utilizes the abstraction of defined two planes by describing the network traffic in flows on the contrary of traditional routers and switches, in which the fast packet forwarding (data path) and the high level routing decisions (control path) occur on the same device [2].

Briefly, the main benefits of SDHetN can be listed as follows:

- Centralized Controllers have global network view, so the operators can program the whole network from a single point.
- Network operators can dynamically change the whole network traffic flow to meet their demands.
- SDN provides vendor-neutral control because Controller software development does not depend on devices types.

The Data Plane and Control Plane communicate each other via OF protocol. The importance and benefits of OF protocol in SDHetN is given in the following subsection.

1.5.2 OpenFlow (OF) protocol

The OF protocol is a significant enabler of SDHetN. It provides interface between Controller in Control Plane and physical switches in Data Plane. The secure channels are defined for communication between the Controller and physical OF switches.

When a flow comes, the header fields of packets are matched with flow tables entries of OF switch by a predefined looking up mechanism. If a matched row can be found, the action defined in that entry is added to packet's action set. If there is no match in that flow table, this flow is defined as a new coming flow [31] and first packet of flow is sent to Controller to take suitable forwarding rule. Due to serve a new comer flow, Controller updates flow tables placed in physical OF switches [32] via OF protocol over secure channels.

The OF switch has flow table pipeline that many tables are ordered one by one. The flow forwarding procedure is run in each table as described previous paragraph. After required actions are added to action set of that flow, it is forwarded to next table in order to search another required actions for it. At that time, another flow can be processed in previous flow tables. In other words, the pipeline processing makes SDHetN flow forwarding easier for dummy physical OF switches.

There are studies that try to enhance SDHetN performance by special changing in Data Plane. For example, in [33] the authors propose three extensions to OF protocol and flow table. The acceleration factor of this approach is studied in the network. In [34], Dely et. al. state that by integrating the OF to their cloud architecture, they achieved a new level of flexibility and configuration to save more energy. In [33], for the sake of low-cost implementation and deployment, the authors proposes three extensions to OF protocol and Flow Table and also they believe deployments will be accelerated in production networks.

However, SDHetN has a significant challenge because of having a single central Controller to assign all newcomer flows. This characteristic causes not only flow load bottleneck but also latency on traffic flow. Therefore, this brings communication overhead on quality of flow [6]. In order to remove this negative effect, the acceleration on traffic flow is needed. This is implemented in both communication between Data Plane and Control Plane, and also in Controller service rate (μ_s) with the help of OF flow table extensions and Accelerating Controller service rate:

- **OF flow table extensions:** In order to provide communication between Data Plane and Control Plane, the OF protocol is used. [2] states that the OF protocol defines flow based traffic in network thanks to the advantage of virtualization of these two planes. In [7], the authors enhance flexibility and configurability of system to save more energy by deploying the OF to their cloud architecture. In order to overcome SDHetN communication overhead with Data Plane based enhancement, the authors propose three extensions to OF protocol and flow table in [8]. They claim that these approach will accelerate the production in networks. Therefore, some extensions to OF flow table structure is adapted to our SDHetN framework.

- **Accelerating Controller:** In order to accelerate Controller, time spent for running the flow admission control should be decreased. To do this, a thread-based parallelization approach for Flow Admission Control Algorithm (FACA) is studied as being Control Plane based enhancement. Therefore, we propose the Topology Control Algorithm (TCA) that adaptively clusters physical resources of Data Plane, i.e. switches (dummy APs), so as to help the the FACA to operate efficiently. Thanks to a new virtual Data Plane that is created by TCA, FACA for each cluster can be run in parallel. Hence, the SDHetN overhead can be minimized by multi-threaded Controller and service rate (μ_s) can be increased.

Consequently, the following contributions are proposed by a novel GoS based and adaptive flow management framework in SDHetN:

- The Data and Control Planes of SDHetN are modeled with $M/M/C_j/K$ and $m-dimensional$ markov systems in queuing theory.
- Adaptive TCA of Controller creates Flow Authority Virtual Switch Farm (FAVSF) by optimizing resource efficiency (ψ) with respect to the number of FAVS (m') and the Grade of Service of t^{th} FAVS (GoS_t).
- Each FAVS in FAVSF provides data to FACA of Controller as being interface between Data Plane and Control Plane.
- A FACA distributes flows fairly to physical resources via global view in order to increase scalability of system under acceptable GoS guarantees.
- The FACAs are run in parallel for each FAVS in order to increase service rate (μ_s) of Controller with the help of $SpeedUp(\gamma)$ calculation.

2. PROPOSED CONTROLLER FRAMEWORK

In this chapter, the proposed SDHetN controller framework is introduced by giving each main components. After looking network architecture that defines whole topology view of SDHetN, Data and Control Plane models are visualized with markov based systems. In the Controller, two significant algorithms are defined. Topology Control Algorithm (TCA) that optimizes resource efficiency and Flow Admission Control Algorithm (FACA) that distributes newcomer flows in a fairer way on created virtual topology. These each subpart are studied in detail as seen in following subsections.

2.1 Network Architecture

Network architecture is crucial for each proposed studies due to define place of each component in the topology. Therefore, the global view on topology where the proposed framework will be run, is given by examining network architecture. It has two components named as Data and Control Planes as seen in Figure 2.1.

In the Data Plane , there are m OF switches, i.e. Access Points (APs), in order to handle mobile users. These switches are dummy devices that are orchestrated by the Control Plane. In our heterogeneous topology, each switch has different service rate $\mu_j(flow/sec)$ and number of physical resources (C_j). Moreover, for each j^{th} smallcell area there are n_j flows and each of flow has different intensity represented as $\rho_{ij}(Erlang)$. The queuing theory based study of Data Plane is given in Section 2.2.

The Control Plane is responsible from the orchestration of the topology. The Controller has two main components named as Control Algorithms and Flow Authority Virtual Switch Farm (FAVSF). As seen in Figure2.1, there are m' FAVSs which are virtual representers of m OF switches. Each FAVS is responsible from its physical OF switch cluster. Moreover, the communication between Data and Control Planes is provided by using the OF protocol. Each FAVS is periodically gets in contact with own physical

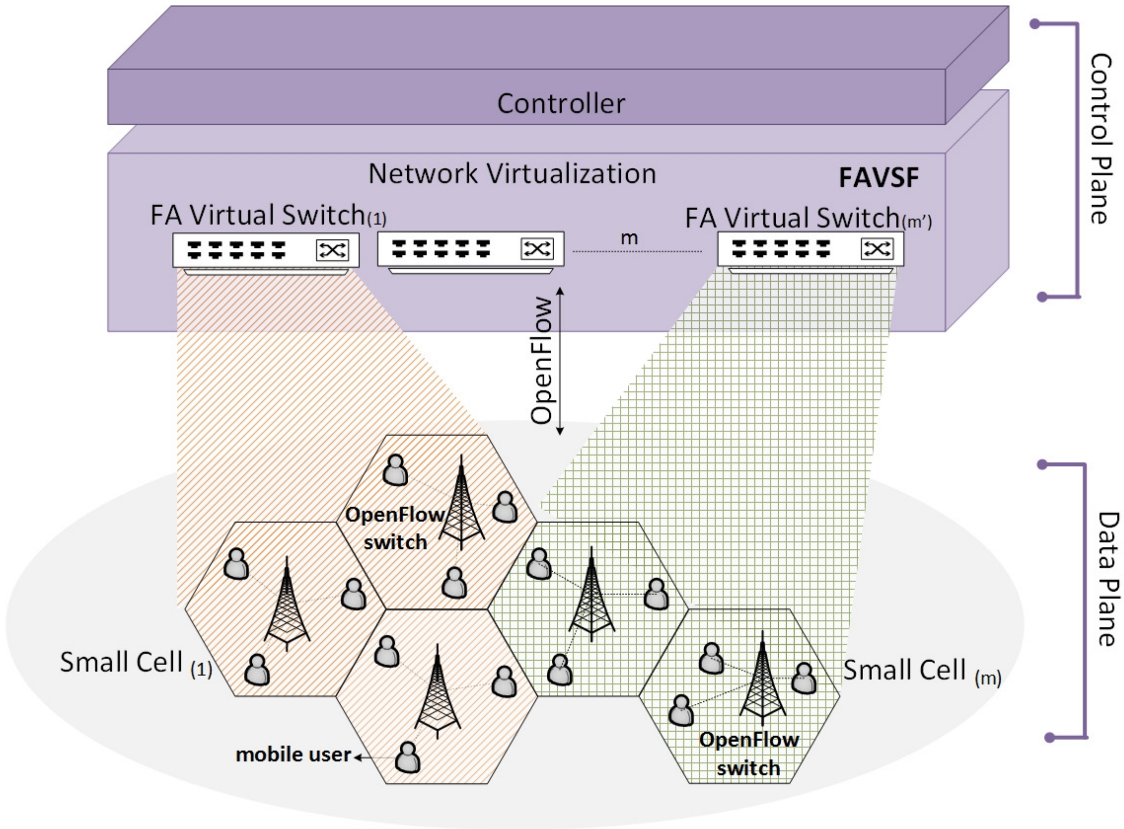


Figure 2.1: Proposed SDHetN Network Architecture.

switches so as to collect required data from Data Plane and send them to Control Plane. These data can be exemplified as the intensity of flow $\rho_{ij}(Erlang)$, flow id, destination switch id etc. The Controller orchestrates FAVSs by operating Control Algorithms which are an adaptive TCA to create FAVSF and a FACA to manage fair flow distribution according to each switch physical resource. The queuing theory based study of Control Plane is given in section 2.3.

2.2 Data Plane Model

In order to define GoS per each flow in Data Plane, first of all the OF switch should be modeled markov based system. Afterwards, the physical component of flow forwarding mechanism which is named as Flow Management Table (FMT), is examined with all main fields in OF switch memory.

2.2.1 Markov model

In the Data Plane Model, we define a flow as:

$$flow_i :< \text{ID (i), Destination switch (j), Intensity } (\rho_{ij}) > \tag{2.1}$$

where i is index of the flow, $\forall i \in (0, 1, ..., N)$; j is index of the OF switch where flow is forwarded, $\forall j \in (0, 1, ..., m)$. $\rho_{ij}(Erlang)$ is the traffic request of i^{th} flow. The total arrival rate of flows coming to j^{th} OF switch is defined as $\lambda_j(flows/sec)$.

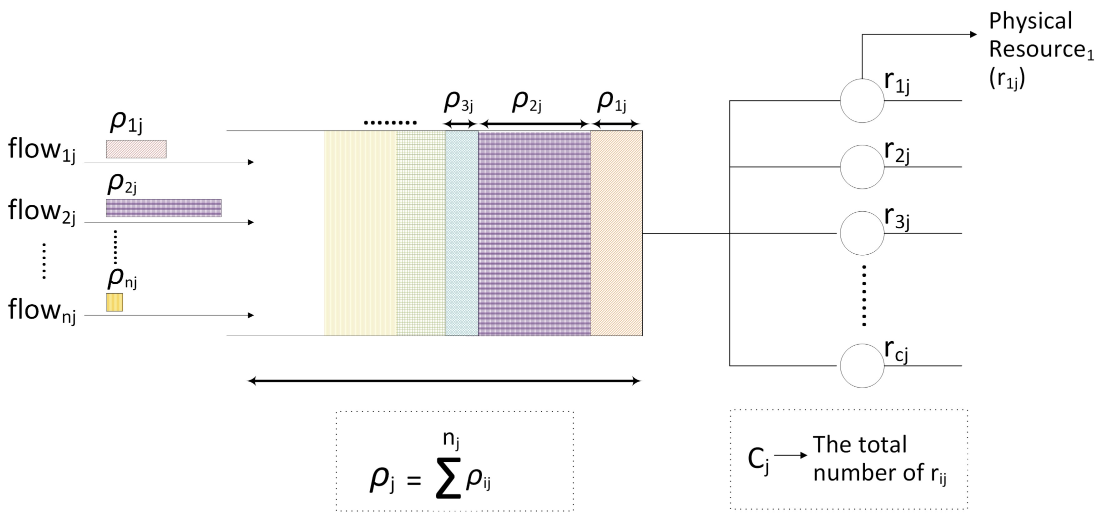


Figure 2.2: The model of a Proposed SDHetN based j^{th} OF switch.

On the other hand, there are OF switches, i.e. wireless APs, in Data Plane and each of them is based on an $M/M/C_j/K$ (where $\forall K > C_j$) markov model as seen in Figure 2.2. The flows that have different traffic intensity (ρ_{ij}), are collected into queue of an switch. Then, this j^{th} switch serves flows through total C_j physical resources. Therefore, the GoS of j^{th} switch (GoS_j) is defined with probability of being P_K^{th} state of $M/M/C_j/K$ markov model as follows:

$$GoS_j = \frac{\left(\sum_{i=0}^{n_j} \rho_{ij} \right)^K}{C_j^{K-C_j} \, . \, C_j! \, . \, \left(\sum_{n=0}^{C_j-1} \frac{(\sum_{i=0}^{n_j} \rho_{ij})^n}{n!} + \sum_{n=C_j}^{K} \frac{(\sum_{i=0}^{n_j} \rho_{ij})^n}{C_j^{n-C_j} . C_j!} \right)} \tag{2.2}$$

where $\forall i, j, n, n_j, C_j, K \in N$. n_j shows the number of traffic flows that are subscribed to j^{th} switch and $K - C_j$ shows the length of queue on j^{th} switch.

Moreover, the individual GoS of each flow is equal to GoS_j. Due to be subscriber of this j^{th} switch, GoS_j represents the GoSs of these flows. Therefore, we calculate GoSs of each flow by evaluating the GoS_j formula of j^{th} switch as in eq.2.2.

2.2.2 Flow management table (FMT)

In the Data Plane, each OF switch has FMT that is filled and updated by SDHetN Controller. These fields are described as follows:

- **OpenFlow Switch Id:** 8 bits unsigned integer represents the j^{th} index which is identity number of switch.
- **Flow Id:** 8 bits unsigned integer defines the i^{th} index which is identity number of flow.
- **In/Out Ports:** This 32 bits unsigned integer field keeps the in/out port numbers in order to use in forwarding.
- **Utility:** This field contains sub-fields for Control Algorithms to utilize while orchestrating topology. Subfields are grouped under one field in order to access the related part easily.
  - C_j: This sub-field stores 16 bits unsigned integer which indicates the total number of physical resources for j^{th} switch.
  - $\mu_j(flow/sec)$: This sub-field holds 16 bits unsigned integer which represents the service rate of j^{th} switch.
  - $\rho_{ij}(Erlang)$: This sub-field keeps 16 bits unsigned float which shows the intensity of i^{th} flow in j^{th} switch.
- **Action:** This field resembles the OF action field, but in our system it is more related to FACA in Controller. If it is needed to exemplified action field, "Forward i^{th} flow over j^{th} switch".

2.3 Control Plane Model

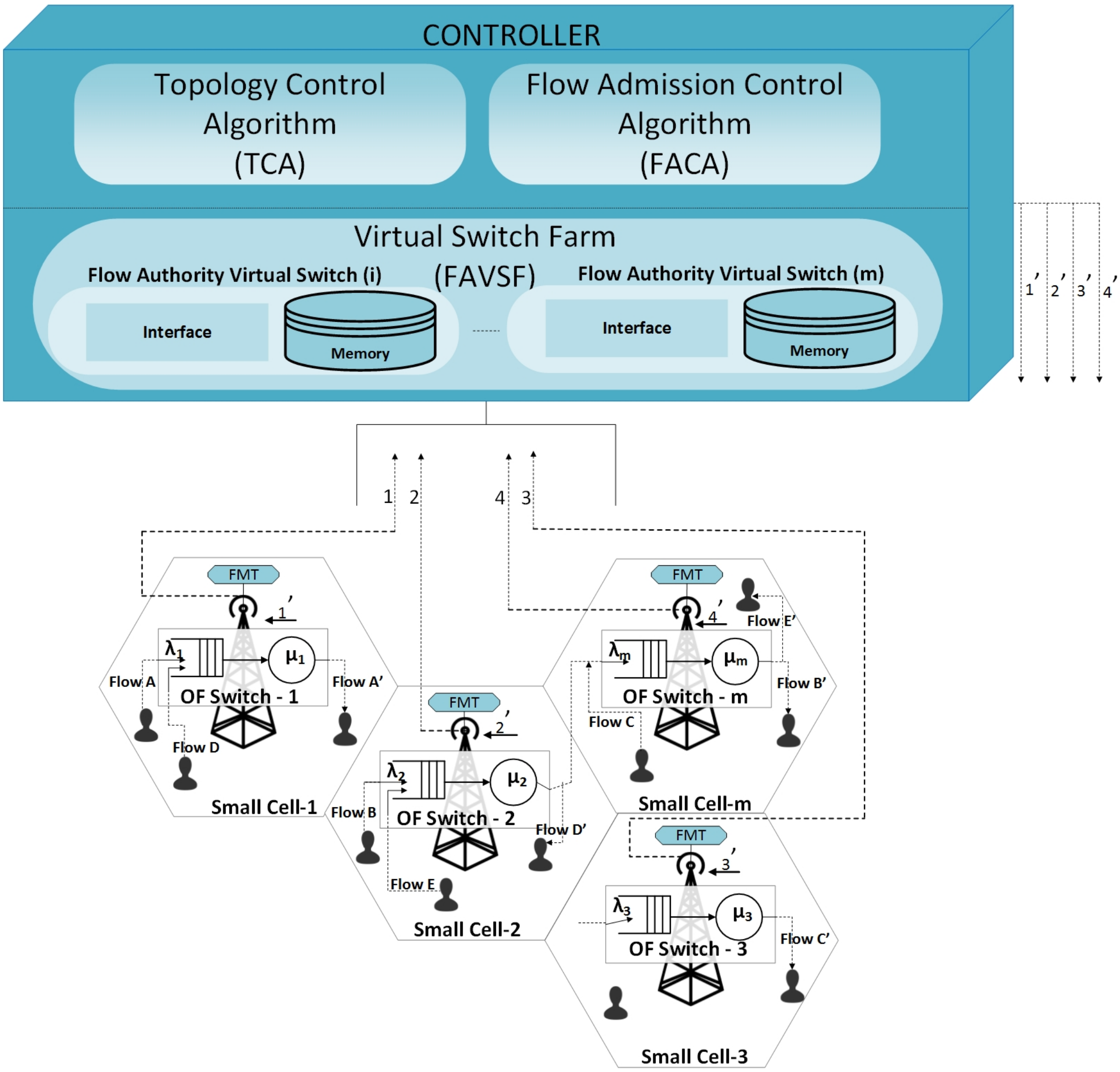


Figure 2.3: Proposed SDHetN based System Model.

In the Control Plane, there is an SDHetN Controller which distributes the flows in a fairer way to switches as seen in Figure 2.3. The Control Plane is modeled as $m-dimensional$ markov chain. In the subsections, we explain markov model of Controller for SDHetN and Fairness(F) expression to indicate difference between Conventional(CON) and SDHetN frameworks.

2.3.1 Markov model

The Control Plane is modeled by m independent $M/M/1/1$ systems. In the Figure 2.4, for $m = 3$ the markov chain is visualized.

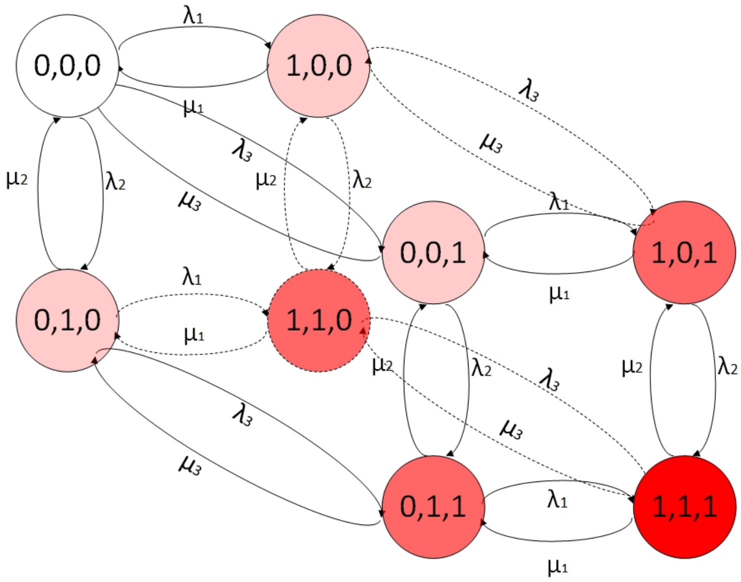


Figure 2.4: The Markov Chain of SDHetN based Proposed Control Plane Model for the topology that has three OF switch in Data Plane ($m = 3$).

Each $M/M/1/1$ system represents each switch with different $\rho_j(Erlang)$. For j^{th} switch, the total arrival rate of flows is defined as $\lambda_j(flow/sec)$ and the service rate switch is named as $\mu_j(flow/sec)$. By considering these different $\rho_j = \frac{\lambda_j}{\mu_j}$ for each switch, the markov chain of Control Plane is studied.

In order to calculate expected number of flows ($E[N]$) in the whole topology, the general steady-state probability of whole system is calculated. Due to have m independent $M/M/1/1$ system, we can get help from Jackson's theorem [4]. According to this, the steady-state probabilities of independent each $M/M/1/1$ system are examined according to different ρ_j. Then, these systems are combined as mentioned in eq.**A.5** of Appendix. Therefore, the general formula of whole system is expressed as follows:

$$P(s_1, s_2, s_3, ..., s_m) = P(s_1)P(s_2)P(s_3)...P(s_m) \qquad \forall s_j \in 0,1 \tag{2.3}$$

In eq.2.3, the probability of that all switches (1 to m) are busy, is shown as:

$$P(1,1,1,...,1) = \prod_{j=1}^{m} P(s_j = 1) \qquad \textbf{(2.4)}$$

This is represented with $(1,1,1)$ state in Figure 2.4. At this point, the $P(s_j = 1)$ is same as P_K^{th} state of j^{th} switch as indicated in section2.2. As mentioned in that section, this probability is also shown by GoS_j of j^{th} switch. Therefore, the $P(1,1,1,...,1)$ is manipulated by using GoS_j of switch as following equations:

$$P(1,1,1,...,1) = \prod_{j=1}^{m} GoS_j \qquad \textbf{(2.5)}$$

$$P(1,1,1,...,1) = \prod_{j=1}^{m} \frac{(\sum_{i=0}^{n_j} \rho_{ij})^K}{C_j^{K-C_j} C_j! (\sum_{n=0}^{C_j-1} \frac{(\sum_{i=0}^{n_j} \rho_{ij})^n}{n!} + \sum_{n=C_j}^{K} \frac{(\sum_{i=0}^{n_j} \rho_{ij})^n}{C_j^{n-C_j} C_j!})} \qquad \textbf{(2.6)}$$

We assume that K, i.e. length of queue, is constant for each switch. Therefore, the expression of expected number of flows $(E[N])$ is obtained by using eq.**A.2** as follows:

$$E[N] = P(1,1,1,...,1) \qquad \textbf{(2.7)}$$

Therefore, by combining eq.2.6 and eq.2.7, the final formula for $E[N]$ is obtain as seen in following equation:

$$E[N] = \frac{(\ \prod_{j=1}^{m} \sum_{i=0}^{n_j} \rho_{ij}\ )^K}{\prod_{j=1}^{m} C_j^{K-C_j}\ .\ C_j!\ .\ (\ \sum_{n=0}^{C_j-1} \frac{(\sum_{i=0}^{n_j} \rho_{ij})^n}{n!} + \sum_{n=C_j}^{K} \frac{(\sum_{i=0}^{n_j} \rho_{ij})^n}{C_j^{n-C_j} C_j!}\ )} \qquad \textbf{(2.8)}$$

2.3.2 Modules

The Controller has two components: Flow Authority Virtual Switch Farm (FAVSF) and Control Algorithms. All flows are forwarded according to FMTs that are dynamically updated by FAVSF according to Control Algorithms outcomes. On the other hand, FAVSF is created adaptively by TCA that is one of the Control Algorithms. The details are explained in following subsections:

2.3.2.1 Flow authority virtual switch farm (FAVSF)

As seen in Figure 2.3, the FAVSF includes FAVSs that are virtual representers of switches. This occurs via virtualization capability of SDHetN. Thanks to this characteristic, complexity of Data Plane is isolated to Controller. Namely, each control algorithm can run in less complexed topology which is defined as FAVSF in SDHetN. Therefore, not only control algorithms orchestrate topology easier but also the time spent for control algorithm significantly reduces. In order to make this characteristic clear, the virtualization of Data Plane is tried to visualized in Figure2.5.

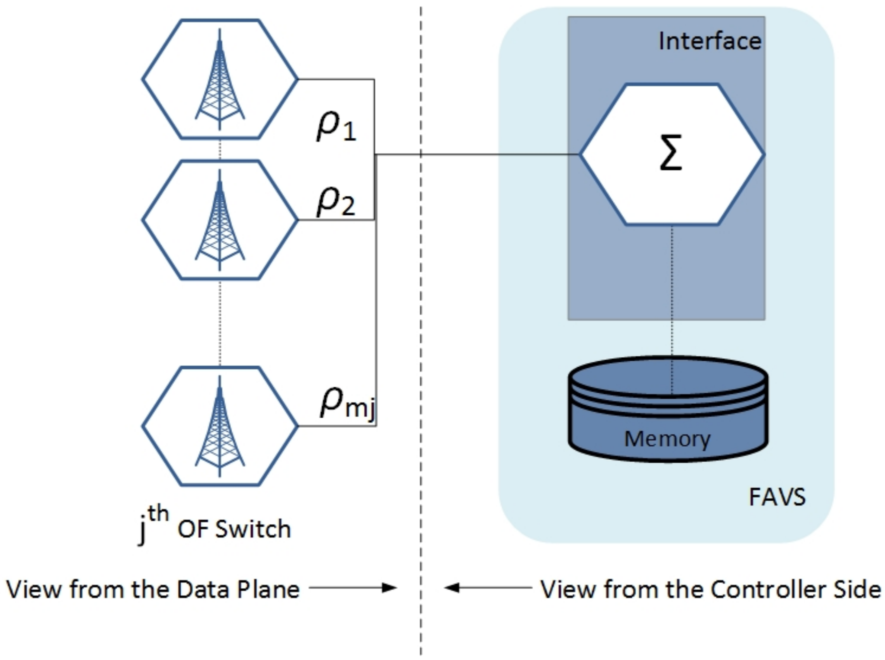


Figure 2.5: Virtualization of Data Plane to Controller side.

As seen in this figure, each FAVS collects required data from own switches and stores these data for usage of Control Algorithms. We virtualize more than one switches and makes Controller seen as one. In other words, according to Controller, there are m' switches which have $\rho_t(Erlang)$ as traffic intensity in the topology. In reality ,they are member in one cluster of switches, which have more than one switches. There are two components of virtualized structure: $Interface$ and $Memory$ as seen from Figure 2.3 and Figure 2.5:

- **Interface:** The Interface of FAVS collects the ρ_{ij} from each flow, calculates the $\rho_j = \sum_{i=1}^{n_j} \rho_{ij}$ for each physical switch that is member of this cluster and total $\rho_t =$

$\sum_{j=1}^{m_t} \rho_j$ which will be seen by Controller. Thanks to this virtualization mechanism, Controller shows one switch (each FAVS in FAVSF), whereas in reality it has many switches in Data Plane. Apart from these, it connects establishment by j^{th} switch in Data Plane and stores required data in own Memory.

- **Memory:** Memory is designed as cache memory that holds necessary data calculated and provided by both Interface and also control algorithms (TCA and FACA). They reads required data from this place and updates them. These data can be j, $\rho_{j,t}$ and $GoS_{j,t}$. Instead of using a single memory for each FAVSs, we prefer using cache memories in each FAVS in order to support parallel operation in Controller, leading increase on service rate.

2.3.2.2 Control algorithms

The Control Algorithms of proposed framework are placed in the Controller. These modules are defined as TCA and FACA as seen in Figure 2.3. The TCA is responsible from run TCA in order to maximize resource efficiency in topology without changing any alteration on Data Plane. It clusters the physical resources of switches according to ρ_j until dropping to acceptable GoS_j level for each cluster. The reason of stopping clustering while reaching an acceptable GoS_j level, is having trade off by μ_s which is service rate of SDHetN Controller with the help of parallel operation of each FACA in Controller. On the other hand, the FACA is responsible to run FACA for each FAVS in order to serve more flows without exceeding individual GoSs. It separates flows to switches in a fair way by considering ρ_j of each switch so as to provide acceptable GoS per flows in the topology. Because of each FACA is run per cluster that are created by TCA, they can operated independently. Thanks to this characteristic of system, the parallel running of each FACA per FAVS becomes possible and μ_s of Controller can be increased. Therefore, the communication overhead because of centralized approach of SDHetN is removed with increase on μ_s, namely decrease on time spent for Control Algorithms operating.

The sequence diagram that shows communication between FAVSF, TCA and FACA in Controller can be seen in Figure2.6. The TCA and FACA are periodically communicate with the FAVSF in Control Plane. The TCA is represented with iterative loop1 and it is operated until reaching the best GoS threshold value which is 0.01 [35]

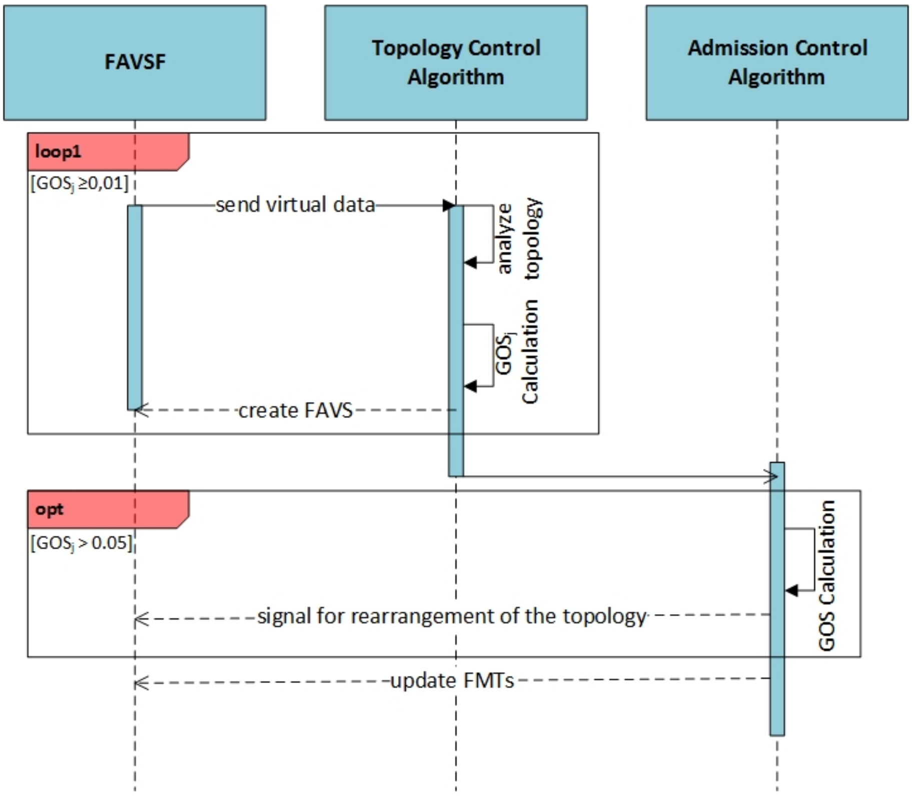


Figure 2.6: Sequence Diagram of the proposed Control Algorithms.

as explained in subsection 3. The option fragment illustrates the creating signal from FACA to TCA because of reaching acceptable GoS threshold which is 0.05 [35]. With the help of this signal, TCA rebuilds the FAVSF. Otherwise, the FACA updates the created FAVSF and manages flow distribution rate in a fairer way by considering each GoS_j.

Topology Control Algorithm (TCA): The TCA analyzes the Data Plane and with TCA creates suitable FAVSF by clustering switch physical resources in order to increase resource efficiency as indicated in advance. We define resource efficiency (ψ) as objective function of optimization formula given in eq.2.9 where theoretically ρ_{max} is the maximum total ρ_{ij} that can be handled by all physical resources in one cluster. The maximum resource efficiency (ψ) is obtained by creating FAVSF with only one FAVS, i.e. one clusters with all physical resources. However, because of SDHetN trade-off between latency and flow load bottleneck, we propose thread-based operating of FACA for each cluster that is created by TCA in Controller. Therefore, the number of FAVS (m') in FAVSF has significant role in terms of acceleration of Controller service rate (μ_s). To both maximize resource efficiency and speed up the Controller while operating FACA per FAVS, we propose the following optimization formula:

$$max \quad \psi = \frac{\sum_{j=1}^{m} \rho_j}{\rho_{max}} \times 100$$

$$\text{s.t.} \qquad m' < m \tag{2.9a}$$

$$\frac{(\sum_{i=0}^{n_t} \rho_{it})^K}{C_t^{K-C_t} \, . \, C_t! \, . \, (\sum_{n=0}^{C_t-1} \frac{(\sum_{i=0}^{n_t} \rho_{it})^n}{n!} + \sum_{n=C_t}^{K} \frac{(\sum_{i=0}^{n_t} \rho_{it})^n}{C_t^{n-C_t} C_t!})} \leqslant 0.01 \tag{2.9b}$$

where t is the index of FAVS, n_t shows the number of flows that are subscribed to t^{th} FAVS and $C_t = \sum_{j=1}^{m_t} C_j$. In order to provide higher quality for each flow, TCA clusters switches by considering this 1% GoS threshold value. Therefore, FACA can operate flow management on a suitable area thanks to this optimization formula.

As indicated in advance, the number of FAVSs in FAVSF is important to operate FACA independently in parallel. Therefore, the Controller service rate (μ_s) speeds up thanks to created several FAVSs. Because of this, TCA stops the clustering switches; i.e. physical resources, while reaching the best acceptable GoS value (1%). According to Amdahl's Law [36], SpeedUp(γ) which is defined as acceleration factor of Controller service rate (μ_s) is calculated as follows:

$$\gamma = \frac{1}{(1-p) + \frac{p}{m'}} \qquad \forall p, m' \in (1, 2, ..., m) \tag{2.10}$$

where p indicates the percentage of algorithm piece that can be operated in parallel independently, and m' indicated the number of threads which is also equal to the number of FAVS created by TCA. Subsequently, with the help of SpeedUp, we can calculate new service rate of Controller ($\mu_s^{'}$) as:

$$\mu_s^{'} = \mu_s \times \gamma \tag{2.11}$$

TCA optimizes resource efficiency(ψ) by optimizing speeding up(γ) of Controller and GoS value of each switch using the above equations on their pseudocode implementations given in Algorithm 1 and Algorithm 2.

In TCA (Algorithm 1), required data collected from Data Plane are operated and FAVSF is created with suitable number of several FAVSs according to optimization formula. The required GoS calculations, which takes $\rho_t = \sum_{j=1}^{m_t} \rho_j$, $C_t = \sum_{j=1}^{m_t} C_j$ and K (C_t + $queue\ length$) as an input and calculate GoS_t, are operated with Algorithm 2

Algorithm 1 TCA

Require: m,ρ_t,C_t
Ensure: m', FAVSF

Describe FAVS, $Array[m]$ and $Continue$ flag
for $x \leftarrow 1$ to m **do**
    $Continue \leftarrow false$
    $tempm' \leftarrow ceil(m/x)$
    Create an array FAVS $Array'[tempm']$
    Sort $Array[m]$ by descending order of ρ_t values using Quick Sort
    for $k \leftarrow 0$ to m **do**
        $ind \leftarrow$ index of $Array'[tempm']$ element with minimum ρ_t value
        $Array'[ind] \leftarrow Array[k]$
        Update ρ_t and C_t fields of $Array'[ind]$
        $Array'[ind].GoS_t \leftarrow CalculateGoS(Array'[ind].\rho_t, Array'[ind].C_t)$
    end for
    for $i \leftarrow 0$ to $tempm'$ **do**
        if $Array'[i].GoS_t \geq 0.01$ **then**
            $Continue \leftarrow true$
            break
        end if
    end for
    Update m' according to $tempm'$
end for

Algorithm 2 GoS Calculation Algorithm ($CalculateGoS$)

Require: ρ_t, C_t, K
Ensure: GoS_t

$sum \leftarrow 1$, $sum2 \leftarrow 0$
for $i \leftarrow 1$ to C_t **do**
    $sum \leftarrow sum + \frac{(\rho_t)^i}{i!}$
end for
for $i \leftarrow C_t$ to K **do**
    $sum2 \leftarrow sum2 + \frac{(\rho_t)^t}{(C_t)^{(i-C_t)}C!}$
end for
return $\frac{(\rho_t)^K}{(C_t)^{(K-C_t)}C_t!(sum+sum2)}$

in TCA. The inputs of TCA are ρ_t and C_t for each FAVS and the number of FAVS, which is initially equal to the number of switch(m). A data structure named *FAVS* is described in order to store required values of each FAVSs which have initially same value with switches. In this *FAVS* data structure, there exists five fields which are the identifier (id), traffic intensity (ρ_t), the number of physical resources (C_t)and calculated GoS (GoS_t). Firstly, $Array[m]$ which is the array of FAVS, is created and initialized as seen in line (1). A *Continue* flag is created and initialized also in line (1) to check whether the clustering is terminated or not. With the outer for loop located between lines (2) and (20), switch included in a FAVS is increased one by one and with the help of inner loop located between lines (7) and (12) a temporary FAVSF is created considering GoS_j of each FAVS. The FAVSF is changed with each iteration of the outer for loop until the condition given in eq.2.9b holds for each FAVS as controlled with a for loop between lines (13) and (18). Whenever all FAVSs in temporary FAVSF holds the conditions, the algorithm terminates and FAVSF is created with m' output.

Flow Admission Control Algorithm (FACA): The FACA is responsible from operating FACA for each FAVS in FAVSF that is built by TCA. Because of FACA is operated for each FAVS, this algorithm distributes intensity of flows in a fairer way by considering this FAVS physical switches intensities (ρ_j) and physical resources. Therefore, each FACA can operated independently and parallel in the Controller.

Algorithm 3 FACA

Require: t,$threshold$,$Array$,$Array'$,
Ensure: update FMTs of switches that t^{th} FAVS is responsible from
for all j^{th} switches that are member of t^{th} FAVS **do**
    $Array[j].GoS_j \leftarrow CalculateGoS(Array[j].\rho_j, Array[j].C_j)$
    while $Array[j].GoS_j \geq threshold$ **do**
        find minimum GoS'_j in t^{th} FAVS
        $min \leftarrow j'$
        if $min = j$ **then**
            return Create a Signal to TCA in order to manipulate FAVSF
        end if
        $maxflow \leftarrow i$
        assign maxflow to min switch, update FMTs
        $Array[j].GoS_j \leftarrow CalculateGoS(Array[j].\rho_j, Array[j].C_j)$
        $Array[min].GoS_min \leftarrow CalculateGoS(Array[min].\rho_min, Array[min].C_min)$
    end while
end for

The FACA is operated as in Algorithm 3. The inputs of FACA are t that is the index of FAVS, $threshold$ that is the acceptable GoS value which vary as (0.01, 0.05, 0.1), $Array$ which keeps switches and $Array'$ which keeps FAVSs that is created by TCA from TCA. Initially, it creates two integers named as *min* that keeps j index of switch which has minimum GoS_j value in t^{th} FAVS and $maxflow$ that keeps i index of flow which has maximum intensity ρ_{ij} in j^{th} switch. With outer loop between line (1) and line (14), all switches that are member of t^{th} FAVS are scanned. For each of them, in line (2) GoS_j is updated. Between line (3) and line (13), until GoS_j drops below threshold, i.e. dissatisfied switch becomes satisfied, flows are distributed the other switches that are satisfied. To do this, the j' switch that has minimum GoS_j in t^{th} FAVS is found and assigned to *min* as seen in lines (4) and (5). Then, the flow which has highest ρ_{ij} is found in this dissatisfied j^{th} switch in line (9). This flow is assigned to min switch by updating Destination switch id of flow in FMTs with line (10). The GoS of j^{th} switch and *min* switch is updated because of trading flow between each other in lines (11) and (12). However, if there is no way to enhance GoS_j under current circumstances, with if block between lines (6) and (8) FACA creates a signal to TCA in order to manipulate virtual topology and rebuilds FAVSF with the helps of optimization formula eq.2.9.

3. THE COMPARISON OF CON AND SDHetN FRAMEWORKS

In this chapter, CON and SDHetN frameworks are compared in terms of fairness, resource efficiency, GoS per flow, number of served flows. First of all, the Jain's fairness index is defined for both frameworks. Afterwards, they are evaluated in 48 different scenarios that details are given in following sections.

3.1 Fairness Expression

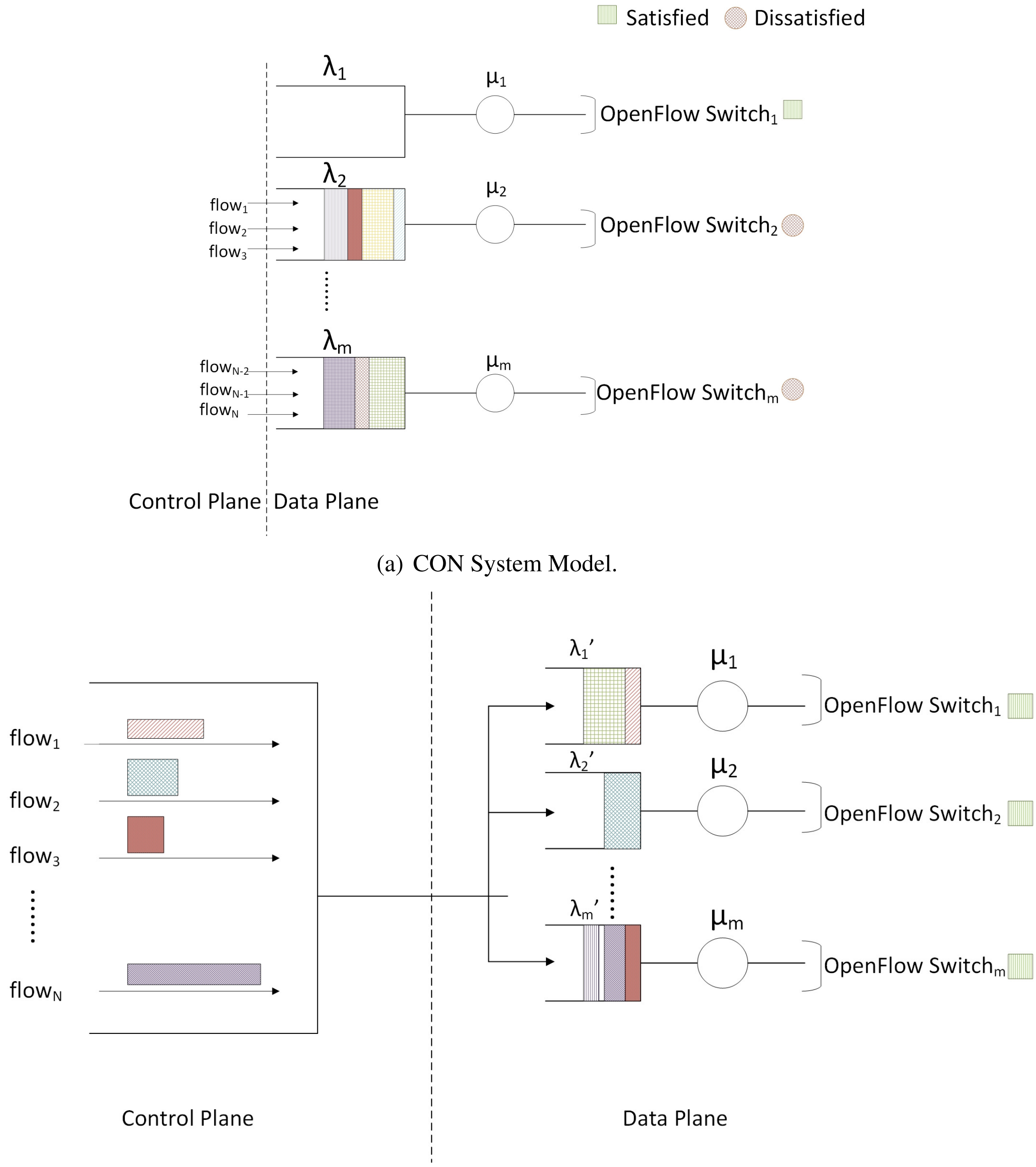
The system model for SDHetN is shown in Figure 2.3. It has been examined as Data and Control Plane Models in previous chapters. Both CON and SDHetN systems have same Data Plane Model. However, SDHetN works with FMTs in each switch differ from the conventional in Data Plane. On the other hand, Control Plane Model is defined only for SDHetN due to have centralized Controller. In order to clarify differences, we examine CON and SDHetN system model as indicated in Figures 3.1(a) and 3.1(b).

The proposed framework aims to distribute flows in a fairer way to switches. In order to define fairness of flow distribution, the Jain's fairness index [37] is used as seen in following equation:

$$F = \frac{(\sum_{i=1}^{N} \rho_j)^2}{N \sum_{i=1}^{N} (\rho_j)^2} \tag{3.1}$$

where F is the fairness index of system according to ρ_j which is total intensity of flows and N that is total number of flows in the topology. If F is closer to 1, that means the system is quite fair with respect to flow distribution.

In the Figures 3.1(a) and 3.1(b), the difference on fairness index (F) is visualized by comparing CON and SDHetN frameworks. In these figures, switches are marked as satisfied or dissatisfied according to their flow numbers (n_j). We defined switch satisfaction, i.e. flow satisfaction, as being satisfied if $GoS_j \leq threshold$, otherwise being dissatisfied. According to [35], the different GoS thresholds vary as 1%, 5%, and 10%. In a such system, GoS_j less than 1% is defined as perfect service quality per



(a) CON System Model.

(b) SDHetN System Model.

Figure 3.1: The Satisfaction of CON and SDHetN System Model.

flow, approximately 5% as acceptable service quality for each flow. However, if GoS_j is higher than 10%, this flow could not be handled in the system.

When the CON and SDHetN systems are compared by examining figures, we propose the F_{SDHetN} is higher than F_{CON}. As we can see from the figures, some dissatisfied switches that is congested by flows and cannot provide acceptable GoS to own subscribers, become satisfied thanks to SDHetN fair flow distribution to switches approach. Thanks to fair flow distribution, we can also serve more flows in the same Data Plane. This claim will be simulated in the following performance evaluation section.

3.2 Performance Evaluation

Table 3.1: Performance Evaluation Scenario Details

m	C_j	highest ρ_j in Case A (Erlang)	highest ρ_j in Case B (Erlang)
1	60	24.96	42.00
2	30	18.61	31.32
3	20	13.77	23.17
4	15	10.843	18.24
5	12	8.91	15.00
6	10	7.56	12.73
10	6	4.70	7.91
12	5	3.95	6.65
15	4	3.19	5.37
20	3	2.41	4.06
30	2	1.62	2.73
60	1	0.81	1.37

The proposed SDHetN system is evaluated using the specific parameters. The topology is 1000x1000 grid. We assume that two cases: Case A and Case B. Case A has 24.96 Erlangs in total, whereas Case B has 42 Erlangs in total. In both cases, there are 60 switches. Queue length (K-C) for each switch is 5.

In the performance evaluation, there are totally 48 different scenarios as seen in Table 3.1. By changing m from 1 to 60, there occurs 12 different scenarios. In addition to this, these 12 scenarios are run for Case A and Case B. This 24 scenarios are also run for SDHetN and CON frameworks. The performance results of our proposed

SDHetN system is compared with CON system that could not do any novel adaptive flow management framework in whole topology. Moreover, each row in Table 3.1 belongs to one switch in Data Plane. For example, in third row, the Data Plane has 3 switches, each of them has 20 physical resources and the most dense switch has 13.77 Erlangs traffic intensity in Case A and 23.17 Erlangs in Case B. The remained 2 switches has different traffic intensities of which summation is $(24.96 - 13.77 = 11.19)$ Erlangs for Case A and $(42 - 23.17 = 18.83)$ Erlangs for Case B.

3.2.1 Evaluation for TCA

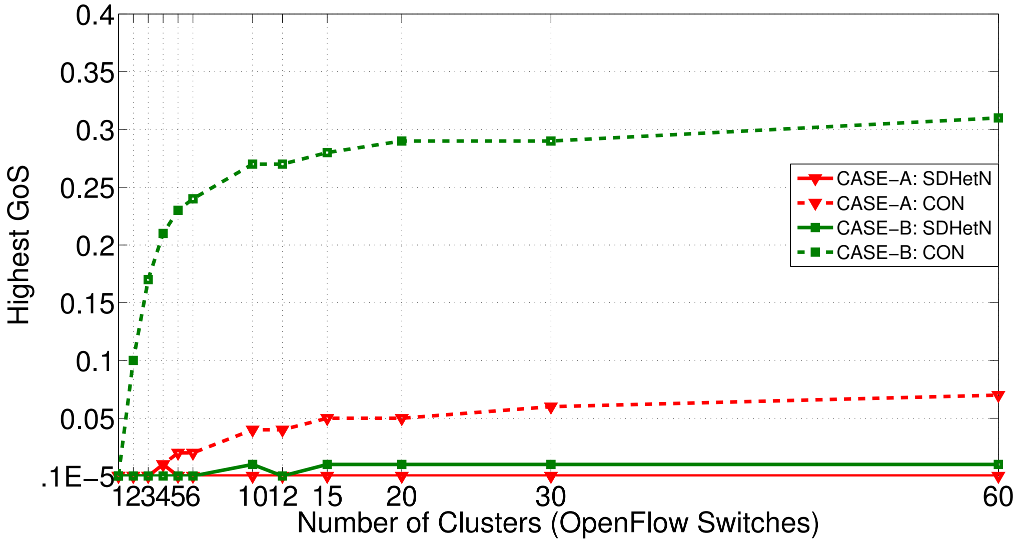
Table 3.2: Number of Created FAVS in Topology Control Algorithm for Two Cases of the Topology

Number of OF Switches in Data Plane	1	2	3	4	5	6	10	12	15	20	30	60
Number of FAVS in Case A	1	2	3	4	2	3	5	6	5	10	15	30
Number of FAVS in Case B	1	1	1	2	1	3	5	3	3	5	5	5

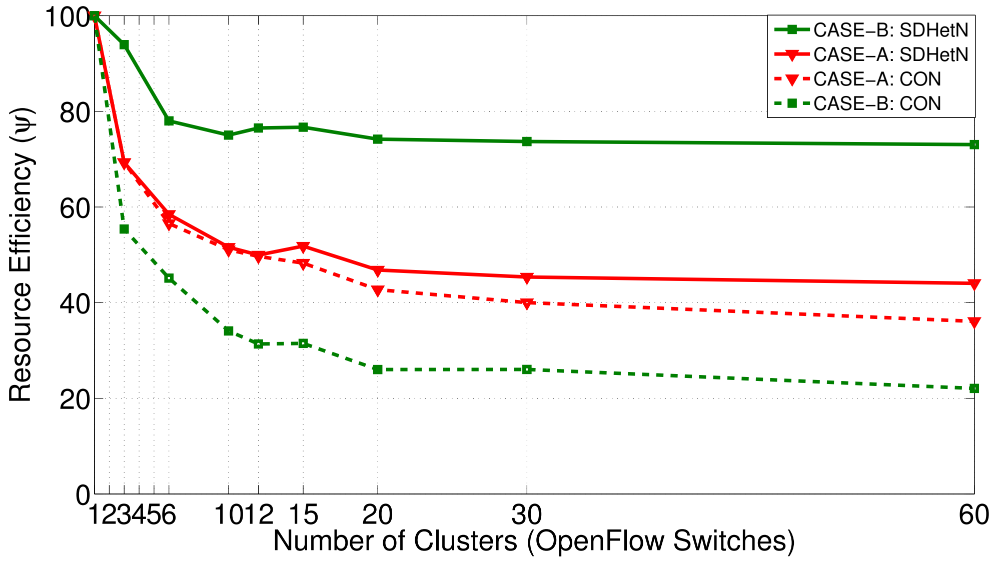
The TCA is evaluated for each scenario that is given in Table 3.1. The number of created FAVS by TCA is seen in Table 3.2. This table represents the first side of optimization formula eq.2.9a in TCA. Because of being dense than Case A, the more physical resource are clustered in few FAVS in Case B in order to offer maximum 0.01 GoS value. Therefore, the number of FAVS is less than Case A in Case B.

The highest GoS_j of topology which is based on second part of optimization formula eq.2.9b of TCA is examined. Figure3.2(a) represents the highest GoS_j, i.e. maximum GoS_j, that changes according to different scenarios in Table 3.1. As we can see in Case A and Case B, CON framework while m increases, the GoS_j of flows also increases because of resource efficiency(ψ) decrease as proved in Appendix. However, in proposed SDHetN system, with the help of TCA, m is manipulated with m' as indicated in Table 3.2. Thanks to this created FAVSF, even highest GoS_j is kept under acceptable level. Therefore, physical resource constraint is isolated to Controller and each flow in the topology can be served with acceptable GoS_j value.

Subsequently, the Resource Efficiency (ψ) is examined as seen in Figure3.2(b). With the help of two side of optimization formula in TCA, physical resource efficiency is much more increased in SDHetN than being in CON. As we can see from figure, both



(a) The highest GoS_j .



(b) The Resource Efficiency(ψ) .

Figure 3.2: The highest GoS_j and Resource Efficiency(ψ) comparison of CON and SDHetN system while the number of physical resource clusters increasing in terms of Case A and Case B topology.

Case A & Case B proposed SDHetN offers larger resource efficiency than CON offers. The increase on resource efficiency is observed as approximately %10 for Case A and %50 for Case B.

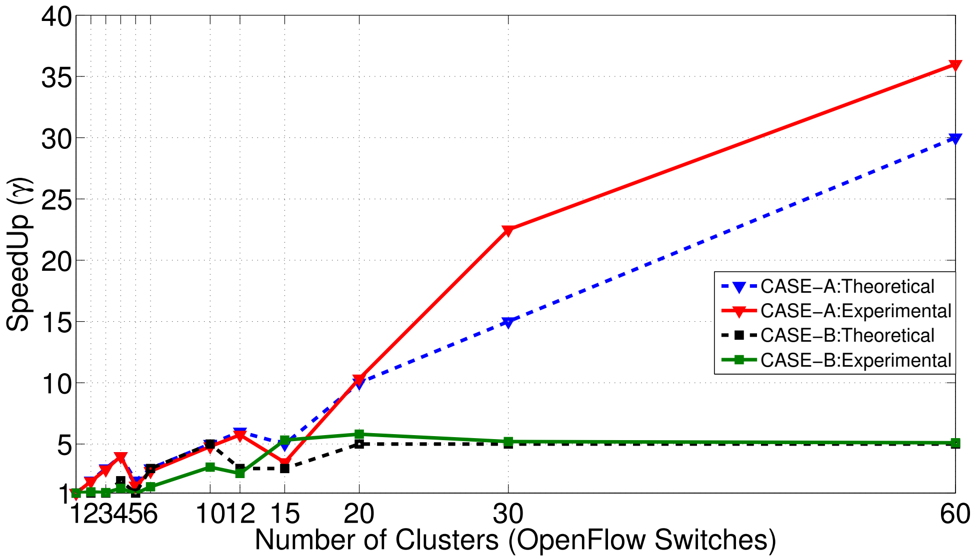
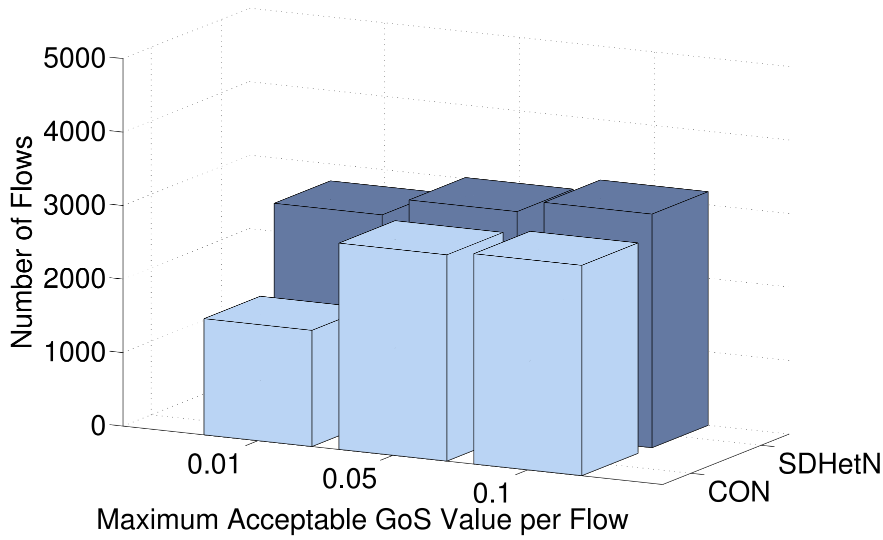


Figure 3.3: The SpeedUp(γ) of SDHetN Controller with help of TCA by evaluating both theoretically and experimentally in CaseA & CaseB.

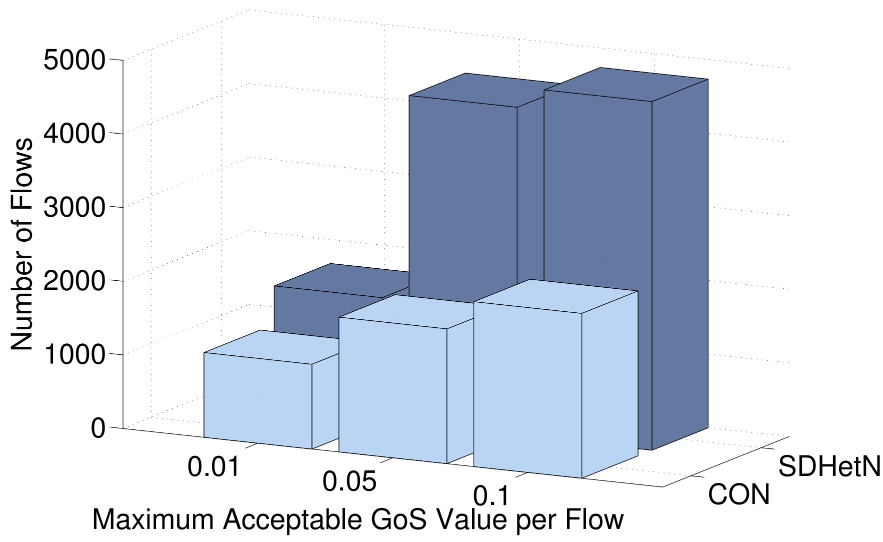
As indicated in advance, FACA is operated in parallel for each FAVS created by TCA. Therefore, this approach is expected to speed up Controller service rate (μ_s). With the help of Table 3.2, Speed Up(γ) is calculated theoretically based on eq.2.10 by comparing thread-based and sequential operations. At the same time, the run time is obtained both thread-based and sequential operating FACA for each FAVS from simulation and plot Figure 3.3. From this figure, we see overlapping of theoretical and experimental (simulation) results for SpeedUp(γ) parameter. Moreover, because of having more FAVS in FAVSF in Case A, SpeedUp of Case A can be higher than Case B.

3.2.2 Evaluation for FACA

After TCA creates FAVSF as indicated in Table 3.2, it prepares area for FACA operations based on Figure 3.2(a) at the same time. FACA is operated until it gives a rearrangement signal to TCA while new flows are coming to system. It is clear that if m is high, the N can be also high. Because of each FACA is operated for each



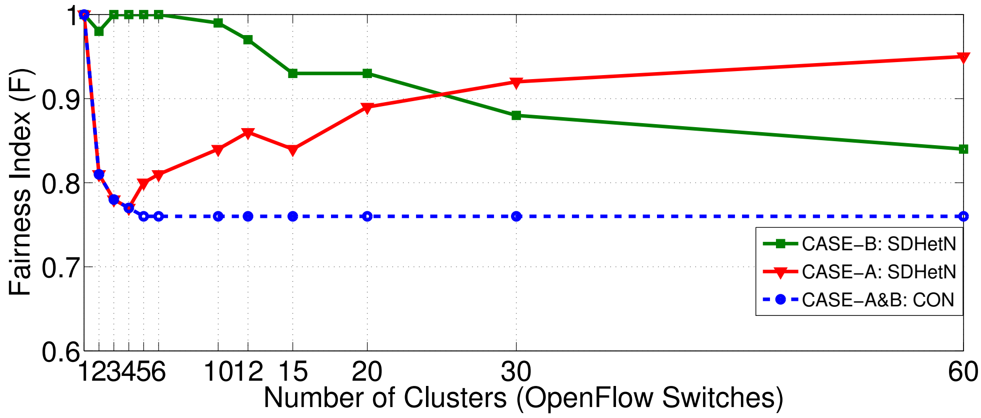
(a) Case A.



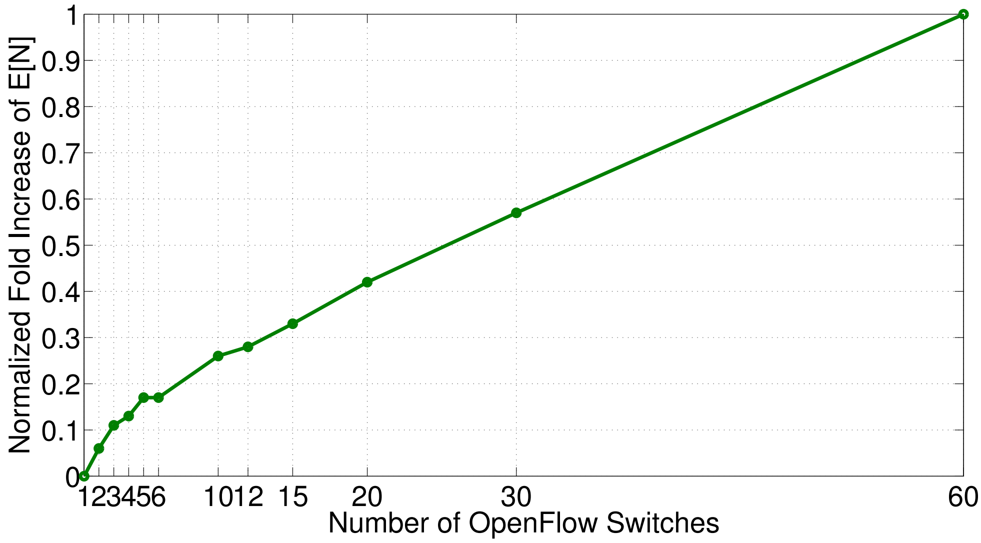
(b) Case B.

Figure 3.4: The comparison of CON and SDHetN in the number of flows that can be handled under different maximum acceptable GoS threshold values.

FAVS, the C_j of each FAVS affects the number of flows that can be served without conceding their GoS_j values. According to results of FACA for 15 switches existence in Data Plane seen in Figure3.4, in both Case A & Case B SDHetN systems can serve much more flow than CON. Moreover, if it is required to compare Case A and Case B, SDHetN can serve much more flow in Case B because of having less FAVS in FAVSF, i.e. having more physical resources in each FAVS, than FAVSs of Case A. The increase on number of flows is nearly %16 for Case A and it offers %60 enhancement as the topology becomes dense while GoS threshold is 0.05.



(a) The comparison of F_{SDHetN} and F_{CON}.



(b) The Expected Number of flow in SDHetN.

Figure 3.5: The Fairness Comparison for SDHetN & CON and Expected Number of Flow Results for SDHetN.

On the other hand, the aim of FACA is to distribute flows to switches in a fairer way. We also show this claim with Figure 3.5(a). As seen from it, proposed SDHetN is fairer than CON in both cases by considering F_{SDHetN} and F_{CON} which are based on eq.3.1. The SDHetN is found roughly %15 fairer than CON according to Jain's fairness index at the end of all novel adaptive flow management framework for 20 switch existence.

Moreover, the Expected Number of flow ($E[N]$) for Case B topology is evaluated for both SDHetN and CON by considering eq.2.8. We show that the rise rate of E[N] in SDHetN based on expected number of flow in CON as seen in Figure 3.5(b). It is clear that in each number of physical resource cluster, SDHetN can serve much more flow than CON according to eq.2.8. For example, in 20 physical clusters scenario, SDHetN can increase expected number of flow by %40 fold based on CON.

3.2.3 Adaptive TCA and FACA

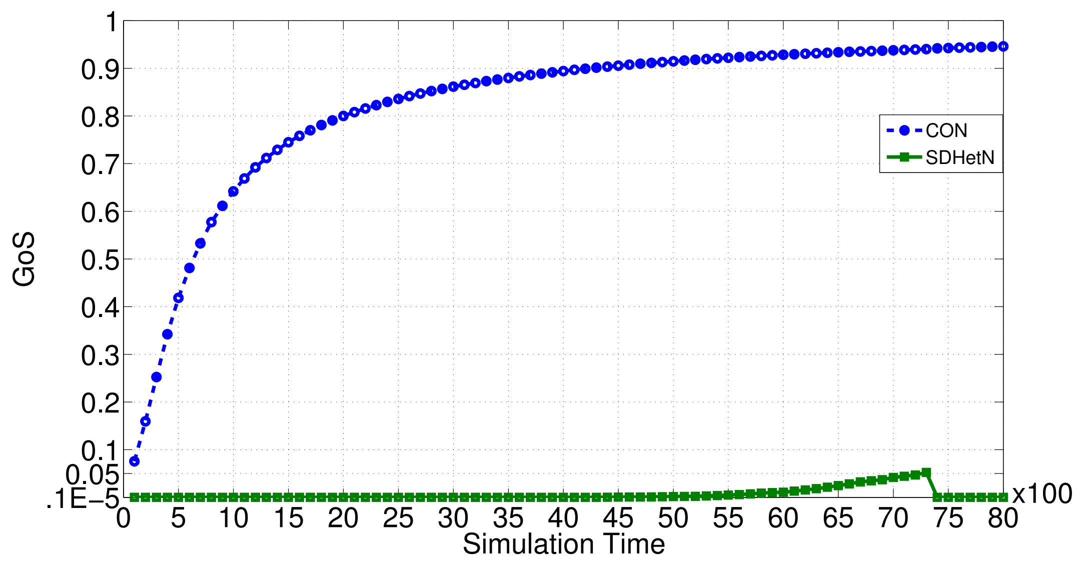


Figure 3.6: The Change on highest GoS_j of topology according to simulation time in terms of SDHetN and CON Systems.

The common working principle of SDHetN system is operating TCA and FACA adaptively as seen in Figure 2.6 while new flows are coming to topology. After a FAVSF is created, FACA operates until it gives rearrangement signal to TCA. Then, TCA that receives rearrangement signal, rebuilds FAVSF and life cycle continues in this manner. Therefore, as seen in Figure3.6, while simulation time passes, the highest GoS_j is kept under an acceptable level in SDHetN, whereas in CON handling this flow in high quality is not possible. Moreover, according to behavior of SDHetN, if highest GoS_j reaches to 0.05 threshold value in operation of FACAs, the rearrangement signal is created and TCA is called once again. Then, the FAVSF is manipulated and highest GoS_j becomes low levels again in the new FAVSF.

4. CONCLUSION

In this thesis, a novel GoS based adaptive flow management framework in SDHetN is proposed. It includes two main structures as TCA and FACA. TCA removes physical resource constraint by creating FAVSF. This proposed system offers virtualization of Data Plane and isolates the resource constraint to Controller. FACA distributes flows to physical resources in a fairer way thanks to global view. Therefore, resource efficiency and scalability of system are increased without any change on physical topology.

Moreover, to remove SDHetN challenges defined as occurrence of bottleneck which creates flow latency, FACAs are operated in parallel. Therefore, the thread-based operating approach speeds up service rate (μ_s) of Controller, and performance of each flow would not be effected negatively. In order to provide communication between Data Plane and Control Plane, an OF protocol extension is defined in this thesis as FMT.

Our evaluations show that the highest GoS providing SDHetN is lower than Conventional system thanks to novel adaptive flow management framework in each simulation time. At the end of performance evaluation, the increase on resource efficiency is observed as approximately %10 for low flow loaded topology and %50 for high flow loaded topology. On the other hand, the increase on number of flows is nearly %16 for low loaded topology and it offers %60 enhancement as the topology becomes dense while GoS threshold is 0.05. The SDHetN is also found roughly %15 fairer than Conventional system according to Jain's fairness index at the end of all novel adaptive flow management framework.

REFERENCES

[1] **Cisco** (2014). Visual Networking Index Report: Global Mobile Data Traffic Forecast Update, 2013–2018, **Technical Report**, Cisco.

[2] **Open Networking Foundation** (2012). Software-Defined Networking: The New Norm for Networks, **White paper**, Open Networking Foundation, Palo Alto, CA, USA, `http://www.opennetworking.org/images/stories/downloads/sdn-resources/white-papers/wp-sdn-newnorm.pdf`.

[3] **Arslan, Z.**, **Erel, M.**, **Ozcevik, Y. and Canberk, B.** (2014). SDoff: A Software-Defined Offloading Controller for Heterogeneous Networks, *IEEE Wireless Communications and Networking Conference (WCNC), 2014*, pp.2827–2832.

[4] **Gross, D.**, **Shortle, J.**, **Thompson, J. and Harris, C.** (2008). *Fundamentals of Queueing Theory*, Wiley, 4th edition.

[5] **Lara, A.**, **Kolasani, A. and Ramamurthy, B.** (2014). Network Innovation using OpenFlow: A Survey, *IEEE Communications Surveys Tutorials*, ***16***(1), 493–512.

[6] **Sünnen, D.**, (28 February 2011). Performance Evaluation of OpenFlow Switches, Master's thesis, Department of Information Technology and Electrical Engineering, Swiss Federal Institute of Technology Zurich.

[7] **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 (GC Wkshps), 2012*, pp.186–191.

[8] **Feng, T.**, **Bi, J. and Hu, H.** (2011). OpenRouter: OpenFlow Extension and Implementation Based on a Commercial Router, *19th IEEE International Conference on Network Protocols (ICNP), 2011*, pp.141–142.

[9] **ElSawy, H.**, **Hossain, E. and Kim, D.I.** (2013). HetNets with Cognitive Small Cells: User Offloading and Distributed Channel Access Techniques, *IEEE Communications Magazine*, ***51***(6), 28–36.

[10] **Kokku, R.**, **Mahindra, R.**, **Zhang, H. and Rangarajan, S.** (2012). NVS: A Substrate for Virtualizing Wireless Resources in Cellular Networks, *IEEE/ACM Transactions on Networking*, ***20***(5), 1333–1346.

[11] **(ISG), N.F.V.N.E.I.S.G.** (2013). Network Functions Virtualization (NFV); Use Cases, V1.1.1, **Technical Report**, ETSI GS NFV, `availableathttp://docbox.etsi.org/ISG/NFV/Open/Published/`.

[12] **Xie, R.**, **Yu, F.**, **Ji, H. and Li, Y.** (2012). Energy-Efficient Resource Allocation for Heterogeneous Cognitive Radio Networks with Femtocells, *IEEE Transactions on Wireless Communications*, ***11***(11), 3910–3920.

[13] **Tachwali, Y.**, **Lo, B.**, **Akyildiz, I. and Agusti, R.** (2013). Multiuser Resource Allocation Optimization Using Bandwidth-Power Product in Cognitive Radio Networks, *IEEE Journal on Selected Areas in Communications*, ***31***(3), 451–463.

[14] **Yue, H.**, **Pan, M.**, **Fang, Y. and Glisic, S.** (2013). Spectrum and Energy Efficient Relay Station Placement in Cognitive Radio Networks, *IEEE Journal on Selected Areas in Communications*, ***31***(5), 883–893.

[15] **Johari, R. and Tsitsiklis, J.** (2006). A Scalable Network Resource Allocation Mechanism With Bounded Efficiency Loss, *IEEE Journal on Selected Areas in Communications*, ***24***(5), 992–999.

[16] **Hong, Y.W.**, **Huang, W.J.**, **Chiu, F.H. and Kuo, C.C.** (2007). Cooperative Communications in Resource-Constrained Wireless Networks, *IEEE Signal Processing Magazine*, ***24***(3), 47–57.

[17] **Xing, Y.**, **Chandramouli, R.**, **Mangold, S. and Sai Shankar, N.** (2006). Dynamic Spectrum Access in Open Spectrum Wireless Networks, *IEEE Journal on Selected Areas in Communications*, ***24***(3), 626–637.

[18] **Liu, J.**, **Jiang, X. and Horiguchi, S.** (2010). Opportunistic Link Overbooking For Resource Efficiency Under Per-Flow Service Guarantee, *IEEE Transactions on Communications*, ***58***(6), 1769–1781.

[19] **Rodrigues, E. and Casadevall, F.** (2011). Control of the Trade-Off Between Resource Efficiency and User Fairness in Wireless Networks Using Utility-Based Adaptive Resource Allocation, *IEEE Communications Magazine*, ***49***(9), 90–98.

[20] **Kuehn, E.**, **Loewel, T.**, **Mange, G. and Nolte, K.** (2010). Enhanced Wireless Network Efficiency By Cognitive, Self-Organized Resource Reconfiguration, *Bell Labs Technical Journal*, ***15***(3), 199–206.

[21] **Goransson, P. and Black, C.**, (2014). Software Defined Networks, Software Defined Networks, Morgan Kaufmann, Elsevier, Boston, `http://www.sciencedirect.com/science/article/pii/B9780124166752000140`.

[22] **Azodolmolky, S.** (October 2013). Get hands-on with the platforms and development tools used to build OpenFlow network applications, *Software Defined Networking with OpenFlow*, Packt Publishing.

[23] **ElSawy, H.**, **Hossain, E. and Kim, D.I.** (2013). HetNets with Cognitive Small Cells: User Offloading and Distributed Channel Access Techniques, *IEEE Communications Magazine*, ***51***(6), 28–36.

[24] **Lara, A.**, **Kolasani, A. and Ramamurthy, B.** (2012). Simplifying Network Management using Software Defined Networking and OpenFlow, *IEEE International Conference on Advanced Networks and Telecommuncations Systems (ANTS), 2012*, pp.24–29.

[25] **Shukla, V.** (June 2013). OpenFlow and VxLAN, *Introduction to Software Defined Networking*, CreateSpace Independent Publishing Platform.

[26] **Levin, D.**, **Wundsam, A.**, **Heller, B.**, **Handigol, N. and Feldmann, A.** (2012). Logically Centralized?: State Distribution Trade-offs in Software Defined Networks, *Proceedings of the First Workshop on Hot Topics in Software Defined Networks*, HotSDN '12, ACM, New York, NY, USA, pp.1–6, `http://doi.acm.org/10.1145/2342441.2342443`.

[27] **Akyildiz, I.F.**, **Lee, A.**, **Wang, P.**, **Luo, M. and Chou, W.** (2014). A Roadmap for Traffic Engineering in SDN-OpenFlow Networks, *Computer Networks, Elsevier*, ***71***(0), 1 – 30, `http://www.sciencedirect.com/science/article/pii/S1389128614002254`.

[28] **Sezer, S.**, **Scott-Hayward, S.**, **Chouhan, P.**, **Fraser, B.**, **Lake, D.**, **Finnegan, J.**, **Viljoen, N.**, **Miller, M. and Rao, N.** (2013). Are We Ready for SDN? Implementation Challenges for Software-Defined Networks, *IEEE Communications Magazine*, ***51***(7), 36–43.

[29] **Kim, H. and Feamster, N.** (2013). Improving Network Management with Software Defined Networking, *IEEE Communications Magazine*, ***51***(2), 114–119.

[30] **Yeganeh, S.**, **Tootoonchian, A. and Ganjali, Y.** (2013). On Scalability of Software-Defined Networking, *IEEE Communications Magazine*, ***51***(2), 136–141.

[31] **Erel, M.**, **Arslan, Z.**, **Özçevik, Y. and Canberk, B.** (2015). Grade of Service (GoS) based Adaptive Flow Management for Software Defined Heterogeneous Networks (SDHetN), *Computer Networks, Elsevier*, ***76***(0), 317 – 330, `http://www.sciencedirect.com/science/article/pii/S1389128614004010`.

[32] **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., ACM*, ***38***(2), 69–74, `http://doi.acm.org/10.1145/1355734.1355746`.

[33] **Feng, T.**, **Bi, J. and Hu, H.** (2011). OpenRouter: OpenFlow Extension and Implementation based on a Commercial Router, *19th IEEE International Conference on Network Protocols (ICNP), 2011*, pp.141–142.

[34] **Dely, P.**, **Vestin, J.**, **Kassler, A.**, **Bayer, N.**, **Einsiedler, H. and Peylo, C.** (2012). CloudMAC x2014; An OpenFlow based Architecture for 802.11 MAC Layer Processing in the Cloud, *IEEE Globecom Workshops (GC Wkshps), 2012*, pp.186–191.

[35] **ITU-T** (1996). Series E: Telephone Network and ISDN, Network Grade of Service Parameters and Target Values for Circuit-Switched Public Land Mobile Services, **Recommendation E.771(10/96)**, International Telecommunication Union, Geneva.

[36] **Herlihy, M. Shavit, N.** (2008). ***P. Gasbarrini T. Gottehrer****, editor, The Art of Multiprocessor Programming*, Elsevier.

[37] **Jain, R.**, **Chiu, D. and Hawe, W.** (September 1984). Quantitative Measure of Fairness and Discrimination For Resource Allocation, *Shared Computer System, DEC Research Report TR-301*.

APPENDICES

APPENDIX A.1 : Markov Model based Comparison of Two Systems

APPENDIX A.1

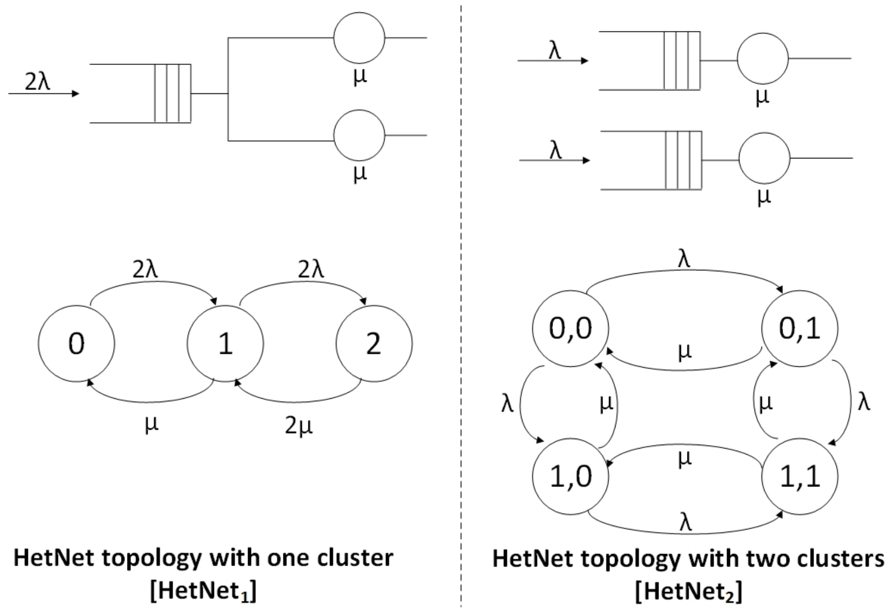


Figure A.1: Comparison of $HetNet_1$ and $HetNet_2$.

As seen in figure, the systems are named according to their number of clusters such as $HetNet_1$ has one cluster with two servers and $HetNet_2$ has two clusters each with one server. Besides, $HetNet_1$ is an M/M/2/2 markov system whereby flows arrive according to Poisson process with rate 2λ and the system has two servers each with μ service rate. Differ from $HetNet_1$, $HetNet_2$ has two separated M/M/1/1 systems that able to be solved by Jackson's theorem [4]. In each M/M/1/1 system of $HetNet_2$, the arrival rate is λ and service rate is μ. The aim is to show significant difference between $HetNet_1$, i.e. using physical resources collectively in one cluster, and $HetNet_2$, i.e. using same number of resources distributed equally to different clusters. In order to support motivation of the thesis, following corollary by examining markov chains of $HetNet_1$ and $HetNet_2$ is proved:

Corollary I: $E[N_{HetNet_2}] < E[N_{HetNet_1}]$

1.1 Examination of $HetNet_2$ Markov Model:

There are two ways to obtain expected number of flows for $HetNet_2$. The first one is considering two dimensional markov chain, otherwise the second one is solving each M/M/1/1 system separately and combining them according to Jackson's theorem.

1.1.1 METHOD I: Two dimensional markov chain

The (i, j) index where $\forall i, j \in \{0,1\}$, indicates each probable state of two dimensional markov chain. $(i, j)^{th}$ state remarks that there are $i + j$ number of flows in the whole system. Moreover, P_{ij} defines the probability of being $(i, j)^{th}$ state, in other words probability of having $i+j$ flows in the system. The general formula of steady-state probabilities of system is calculated by means of [4] as follows:

$$P_{ij} = \frac{(\rho)^{i+j}}{(\rho+1)^2} \qquad \forall i,j \in 0,1 \tag{A.1}$$

where $\rho = \frac{\lambda}{\mu}$. So as to calculate expected number of flows serving in the system, following equations are evaluated:

$$E[N] = \sum_{i=0}^{1} iP_i \tag{A.2}$$

$$E[N_{HetNet_2}] = P_{11} = \frac{\rho^2}{(1+\rho)^2} \tag{A.3}$$

1.1.2 METHOD II: Independent two M/M/1/1 system according to Jackson's theorem

For an M/M/1/1, the general formula of steady-state probabilities are obtained by examining inlines and outlines of (0) and (1) states and the final version is defined as seen in [4]:

$$P_i = \frac{\rho^i}{1+\rho} \qquad \forall i \in 0,1 \tag{A.4}$$

To obtain general steady-state probability formula of $HetNet_2$ by considering two M/M/1/1 systems together, according to Jackson's theorem [4] following equation is obtained:

$$P_{ij} = P_i P_j \tag{A.5}$$

By regulating eq.**A.5**, the final expression of general formula for $HetNet_2$ is evaluated as follows:

$$P_{ij} = \frac{\rho^{i+j}}{(1+\rho)^2} \qquad \forall i,j \in 0,1 \tag{A.6}$$

In order to calculate expected number of flows that are served in the system, eq.**A.2** is used and finalized with same expression in eq.**A.3**.

1.2 Examination of $HetNet_1$ Markov Model:

The (i) index where $\forall i \in \{0,1,2\}$, indicates each probable state of system. $(i)^{th}$ state remarks that there are i flows in the system. Moreover, P_i defines the probability of being in i^{th} state in the markov chain of system. If it is required to extract general formula of steady-state probabilities for $System_1$, firstly the probabilities of each state are expressed in terms of P_0. The final situation of P_0 is obtained as follows by arranging expressions of M/M/c/k systems [4] to M/M/2/2:

$$P_0 = \frac{1}{2\rho^2 + 2\rho + 1} \tag{A.7}$$

Therefore, the general formula of steady-state probabilities is obtained as follows:

$$P_i = \frac{2\rho^i}{2\rho^2 + 2\rho + 1} \qquad \forall i \in 1,2 \tag{A.8}$$

The expected number of flows is calculated by using eq.**A.2** and the final expression is obtained as seen in following equation:

$$E[N_{HetNet_1}] = \frac{2\rho + 4\rho^2}{2\rho^2 + 2\rho + 1} \tag{A.9}$$

1.3 Proof of *Corollary I*:

$$E[N_{HetNet_2}] < E[N_{HetNet_1}] \tag{A.10}$$

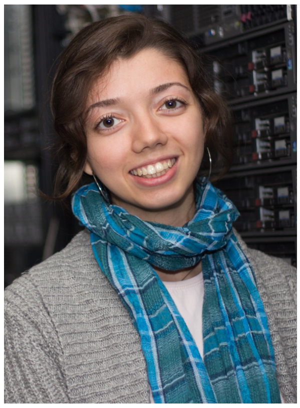
$$\frac{\rho^2}{(\rho+1)^2} < \frac{2\rho + 4\rho^2}{2\rho^2 + 2\rho + 1} \tag{A.11}$$

By simplifying the expression, it is obtain that:

$$\frac{-2\rho^3 - 8\rho^2 - 7\rho - 2}{(\rho+1)^2(2\rho^2 + 2\rho + 1)} < 0 \tag{A.12}$$

As known $\rho > 0$, the numerator of above expression is higher than 0. If we look numerator and denominator of eq.**A.12**, denominator is greater than 0 and numerator is lower than 0. Therefore, the claimed expression in *corollary I* is proved by eq.**A.12**. Namely, if the physical resources are divided and separately assigned to many clusters, the number of flows that can be served with acceptable GoS in whole topology is decreasing.

CURRICULUM VITAE



Name Surname: Müge EREL

Place and Date of Birth: Izmir/TURKEY, 11.11.1991

Adress: BAAL lab-1, Department of Computer Engineering, Istanbul Technical University, Ayazaga 34469, Istanbul-TURKEY

E-Mail: erelmu@itu.edu.tr

M.Sc.: Computer Engineering in Istanbul Technical University, May 2015

B.Sc.: Computer Engineering in Istanbul Technical University, July 2013

Professional Experience and Rewards:

- Second rank of honor in the faculty, in graduation BSc.
- "Best Poster Paper Award" in IEEE Conference on Computer Communications (INFOCOM), HONGKONG, April 2015.
- **Technical Program Committee (TPC) Member:**
  - International Conference on Computing Networking and Communications, Green Computing, Networking and Communications Symposium (ICNC '15 GCNC).
  - IEEE 82nd Vehicular Technology Conference: VTC2015-Fall.
  - The International Workshop on Energy Awareness for Heterogeneous Networks: Recent Hardware and Software-Defined Solutions (EAHN 2014), in conjunction with IEEE DCOSS 2014.
- **Reviews for Journals:**
  - IEEE Transactions in Vehicular Technology (TVT), 2014-present.
  - IEEE Transactions in Wireless Communications (TWC), 2014 -present.
  - The International Journal of Communication Systems (IJCS), 2014-present.
  - The International Journal of Computer and Telecommunications Networking (COMNET), 2015 - present.
- **Reviews for Conferences and Workshops:**
  - IEEE Global Communications Conference Exhibition and Industry Forum (GLOBECOM) Workshops, December 2014.
  - IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom), May 2014.

List of Publications:

- **Journals:**
  ▪ G. Secinti, **M. Erel**, B. Canberk, 'Green Topology and Power Control Model for Mobil LTE Networks', submitted to **IEEE Transaction on Vehicular Technology (TVT)**, April 2015.

  ▪ **M. Erel**, Z. Arslan, Y. Ozcevik, B. Canberk, 'Grade of Service (GoS) based Adaptive Flow Management for Software Defined Heterogeneous Networks (SDHetN)', **Computer Networks- ELSEVIER PUBLICATIONS**,76(0),317-330, January 2015.

  ▪ E. Bozkaya, **M. Erel**, B. Canberk, 'Connectivity Provisioning Using Cognitive Channel Selection in Vehicular Networks', in **Lecture Notes of the Institute of Computer Sciences, Social Informatics and Telecommunications Engineering, Ad Hoc Networks (Springer)** ,vol.140,pp.169-179, November 2014.

- **Book Chapter:**
  ▪ **M. Erel**, Z. Arslan, Y. Ozcevik, B. Canberk, 'Software-Defined Wireless Networking: A New Paradigm for Next Generation Network Management Framework', **Modeling and Simulation of Computer Networks and Systems: Methodologies and Applications, Edited by M.S. Obaidat, F. Zarai and P. Nicopolitidis, ELSEVIER PUBLICATIONS**, 2014.

- **Conference Papers:**
  ▪ Y. Ozcevik, **M. Erel**, B. Canberk, 'Spatio-Temporal Multi-stage OpenFlow Switch Model for Software Defined Cellular Networks',**IEEE 82nd Vehicular Technology Conference (VTC2015-Fall)**, Boston-USA, September 2015.

  ▪ Y. Ozcevik, **M. Erel**, B. Canberk, 'NOC based Banyan OpenFlow Switch for Software Defined Networks', in **IEEE 2015 23. Sinyal Isleme ve Iletisim Uygulamaları Kurultayı (23nd Signal Processing and Communications Applications Conference)**, Malatya-TURKEY, May 2015.

  ▪ G. Secinti, **M. Erel**, B. Canberk, 'XCon: Cross Layer Topology Controller for Energy Efficient LTE Networks', in **IEEE Conference on Computer Communications (INFOCOM)**, HONGKONG, April 2015,***Best Poster Paper Award***.

  ▪ Z. Arslan, **M. Erel**, Y. Özçevik, and B. Canberk, "SDoff: A Software-Defined Offloading Controller for Heterogeneous Networks", **IEEE Wireless Communications and Networking Conference (WCNC)**, Istanbul-TURKEY, April 2014.

  ▪ **M. Erel**, Y. Ozcevik, B. Canberk, 'A Topology Control Mechanism for Cognitive Smallcell Networks Under Heterogeneous Traffic', in **IEEE 14th International Symposium and Workshops on a World of Wireless, Mobile and Multimedia Networks (IEEE WoWMoM)** , Madrid-SPAIN, June 2013.

Publications On The Thesis

- **M. Erel**, Z. Arslan, Y. Ozcevik, B. Canberk, 'Grade of Service (GoS) based Adaptive Flow Management for Software Defined Heterogeneous Networks (SDHetN)', **Computer Networks- ELSEVIER PUBLICATIONS**,76(0),317-330,2015.

- **M. Erel**, Z. Arslan, Y. Ozcevik, B. Canberk, 'Software-Defined Wireless Networking: A New Paradigm for Next Generation Network Management Framework', **Modeling and Simulation of Computer Networks and Systems: Methodologies and Applications, Edited by M.S. Obaidat, F. Zarai and P. Nicopolitidis, ELSEVIER PUBLICATIONS**, 2014.