UNIVERSITÄT DES SAARLANDES

ERICSSON

Traffic Engineering and Energy-Efficient Routing in IP-Based Mobile Networks

Master Thesis in CuK
by

Dinesh Kumar Lakshmanan

Supervised by

Prof. Dr. Thorsten Herfet

reviewed by

Prof. Dr. Dietrich Klakow

advised and reviewed by

Mr. Steffen Bretzke

Saarbrücken, August 9, 2012

Statement in Lieu of an Oath

I hereby confirm that I have written this thesis on my own and that I have not used any other media or materials than the ones referred to in this thesis.

I hereby confirm the congruence of the contents of the printed data and the electronic version of the thesis.

Declaration of Consent

I agree to make both versions of my thesis (with a passing grade) accessible to the public by having them added to the library of the Computer Science Department.

Saarbrücken, ...

(Datum / Date) (Unterschrift / Signature)

Abstract

Current IP mobile backbone networks exhibit poor power efficiency, running network devices at full capacity all the time regardless of the traffic demand and distribution over the network. Network operators usually build networks with redundant and over-provisioned links resulting in low link utilization during most of the time. While these redundant links and bandwidth greatly increase the network reliability, they also greatly reduce the network's energy efficiency as all the network devices are powered ON at full capacity but highly under-utilized most of the time. Most research on router power management treat routers as isolated devices. An alternate approach is to facilitate power management at network level by routing traffic through different paths to adjust the workload on individual routers or links. In this thesis work focuses on intra-domain traffic engineering mechanism, EE-TE formulated based on MCF model and it is solved using the powerful optimization engine CPLEX. The EE-TE model maximizes the total power saving of routers in a network by finding energy efficient routing paths for each OD pair and also shows how to split the traffic on a chosen path depending on the actual traffic demand measured in the network. EE-TE maximizes the number of links that can be put into sleep under given performance constraints such as link utilization and traffic split ratio. Using random network topologies and traffic demand profiles, the evaluation shows that EE-TE can reduce line-cards' power consumption by 20 % to 40 % under constraints that the maximum link utilization is below 50 % for IP-based mobile backbone networks.

Acknowledgements

This report is a result of a Master's thesis project at Ericsson Eurolab R & D in Aachen. This thesis would not have been possible without the guidance and the help of several individuals who in one way or another contributed and extended their valuable assistance in the preparation and completion of this study.

This project would not been successful without these persons :

I would like to thank Prof. Thorsten Herfet for giving me the opportunity to pursue this thesis at Ericsson and also for accepting to supervise my thesis work. I am grateful to do thesis under him for his excellent knowledge and skills in a broad area of telecommunications, rapid responses to e-mails, helpful suggestions, and genuine kindness.

I would like to specially thank Prof. Dietrich Klakow for accepting to be my reviewer for his support and help when I needed it.

I would like to thank heart-fully my supervisor Mr. Steffen Bretzke at Ericsson for giving me the opportunity to pursue the thesis in his project. It's a great honor to work under him for his extensive support, motivation and helpful advising throughout the thesis work.

I would like to thank Mr. Joerg Aelken and Mr. Frank Sell for providing advice and guidance during this thesis work, their valued views and suggestions on traffic engineering and routing studies in IP-mobile networks were very supportive.

Other people that I want to mention includes Sascha Smets, Norbert Niebert, Stephan Kruska and Robert Farac, three master thesis students at Ericsson that I worked with them during my thesis.

Finally, I am very thankful to my Family and Friends for their immense love and support.

Thank you all for your valuable support and guidance.

Contents

List of Figures

List of Tables

Abbreviations

GHG	**G**reen **H**ouse **G**ases
ICT	**I**nformation and **C**ommunication **T**echnology
ISP	**I**nternet **S**ervice **P**rovider
GNT	**G**reen **N**etwork **T**echnology
IP	**I**nternet **P**rotocol
OD	**O**rigin-**D**estination
TE	**T**raffic **E**ngineering
EE-TE	**E**nergy **E**fficient-**T**raffic **E**ngineering
MCF	**M**ulti-**C**ommodity **F**low
LP	**L**inear **P**rogramming
MIP	**M**ixed **I**nteger **P**rogramming
IGP	**I**nterior **G**ateway **P**rotocol
OSPF	**O**pen **S**hortest **P**ath **F**irst
IS-IS	**I**ntermediate **S**ystem-**I**ntermediate **S**ystem
AS	**A**utonomous **S**ystem
MPLS	**M**ulti **P**rotocol **L**abel **S**witching
MT	**M**ulti **T**opology
LAN	**L**ocal **A**rea **N**etwork
EATe	**E**nergy **A**aware **T**raffic **e**ngineering
EAR	**E**nergy **A**ware **R**outing
SPT	**S**hortest **P**ath **T**ree
CCU	**C**entral **C**ontrol **U**nit
RAN	**R**adio **A**ccess **N**etwork
CN	**C**ore **N**etwork
3GPP	**T**hird **G**eneration **P**artnership **P**roject

RNC	**R**adio **N**etwork **C**ontroller
UMTS	**U**niversal **M**obile **T**elecommunication **S**ystem
MBH	**M**obile **B**ack **H**aul
LRAN	**L**ow **R**adio **A**ccess **N**etwork
HRAN	**H**igh **R**adio **A**ccess **N**etwork
RBS	**R**adio **B**ase **S**tation
ATM	**A**synchronous **T**ransfer **M**ode
TDM	**T**ime **D**ivision **M**ultiplexing
IMS	**I**nternet **P**rotocol **M**ultimedia **S**ubsystem
CS	**C**ircuit **S**witched
PS	**P**acket **S**witched
EPC	**E**volved **P**acket **C**ore
SAE	**S**ystem **A**rchitecture **E**volution
LTE	**L**ong **T**erm **E**volution
CSP	**C**ommunication **S**ervice **P**rovider
CDMA	**C**oded **D**ivision **M**ultiple **A**ccess
SGW	**S**erving **G**ate **W**ay
PGW	**P**acket **G**ate **W**ay
SGSN	**S**erving **G**ateway **S**upport **N**ode
GPRS	**G**eneral **P**acket **R**adio **S**ervice
GGSN	**G**PRS **G**ateway **S**upport **N**ode
NOC	**N**etwork **O**peration **C**enter
LSA	**L**ink **S**tate **A**dvertisements
SE	**S**mart **E**dge Router
LIC	**L**ine **I**nterface **C**ards
OPL	**O**ptimzation **P**rogramming **L**anguage

Dedicated to my Beloved Parents

Chapter 1

Introduction

Power consumption has become a key issue in the last few years, due to rising energy costs and serious environmental impacts of Green House Gases (GHG) emissions. Pollution and energy saving are keywords that are becoming more and more of interest to people and to governments, and the research community is also becoming more sensible towards these topics. Focusing on ICT, a number of studies estimate a power consumption related to ICT varying from 2 % to 10 % of the worldewide power consumption [1]. This trend is expected to increase notably in the near future.

In the last few years, Telcos, ISPs, and the public organizations around the world reported statistics of network energy requirements and the related carbon footprint, showing an alarming and growing trend. The Global e-Sustainability Initiative (GeSI) [2] estimated an overall network energy requirement of about 21.4 TWh in 2010 for European Telcos, and forsees a figure of 35.8 TWh in 2015 if no Green Network Technologies (GNTs) will be adopted.

Research on energy management has traditionally focused on battery-operated devices, and more recently, stand-alone servers and server clusters in data centers. The underlying network infrastructure, namely routers, switches and other network devices, still lacks effective energy management solutions. Epps et al. [3] from Cisco report that a high-end router CRS-1 with maximum configuration can consume as much as one MegaWatt. Cisco [4] forcasted a Global mobile data traffic to grew 2.6 fold in 2010 and 26-fold by 2015. The Global mobile data traffic forecast by region is shown in Figure 1.1. The report points out that, driven by exponential growth of Mobile data traffic, router system requirements are outpacing silicon and cooling technologies.

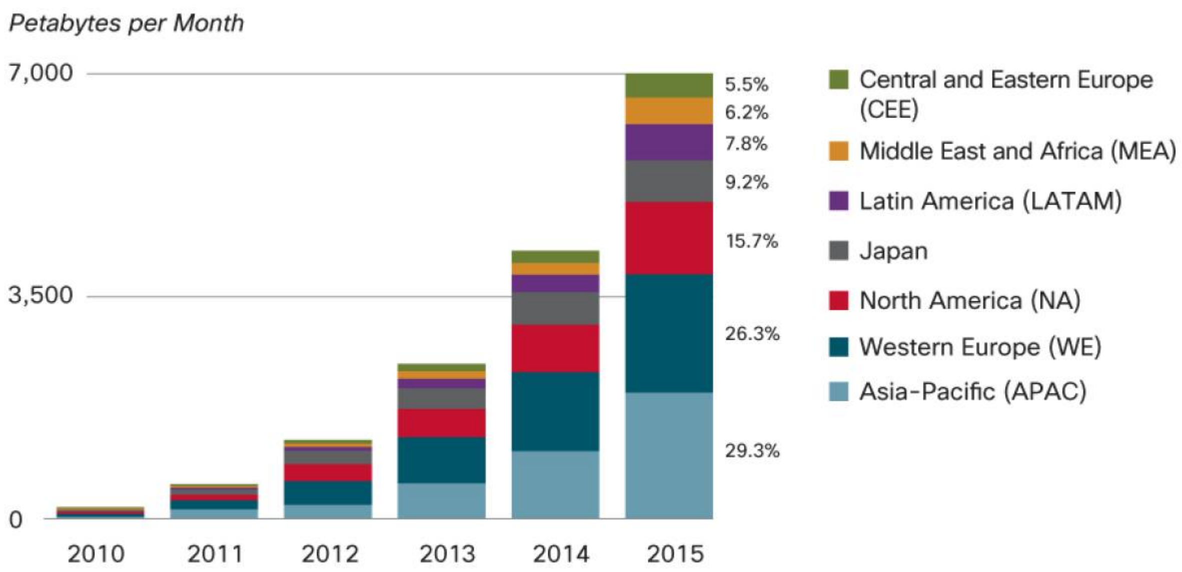

FIGURE 1.1: Global Mobile Data Traffic [4]

To this extent, networking devices like IP backbone routers consume a large majority of energy [5], and the energy consumption of a network can almost double when including air conditioning and cooling costs. It is therefore not surprising that researchers, manufactures and network providers are aiming to reduce the power consumption of ICT systems from different angles. In short, router power consumption has become an increasing concern for Telcos, ISPs and data centers.

A variety of approaches have been proposed to minimize power consumption, such as dynamic voltage scaling [6] where the voltage used in a component is increased or decreased depending upon circumstances. Among all different approaches, sleep mode is one of the most effective methods for power saving. It enables electronic devices to operate in low power mode to save power. This significantly reduces power consumption compared to leaving the devices on full power but with nothing to do. Such autonomic and localized sleep mode cannot be easily implemented in routers, because they often exchange routing information (i.e. non-data packets) through routing protocols, and thus they cannot stand idle even though there is no data traffic. In addition, if it was not required to handle any data packets, a router entering sleep mode will make itself inactive and unresponsive to network traffic. Without a network-wide coordination, this may cause the topology to virtually disconnect, which could lead to connection blocking. In [7], the authors discussed the issue of un-coordinated and coordinated sleeping in routers of a network. However, that coordination was based on the past history of traffic in the network, and the setting was static so that it cannot adapt to dynamic changes in network traffic.

1.1 Motivation

Existing research on router power management treats routers as isolated devices and focuses on reducing power consumption at hardware component level. Earlier Gupta et al. [7] suggested to consider routers in the network context and create more power saving opportunities by adjusting the amount of traffic going through routers, but they did not propose specific solutions.

There are link-level solutions which put line-cards to sleep when there is no traffic on the link [8], however the power saving from opportunistic sleeping is limited by the inter-arrival time of packets. Complementary to component-level and link level solutions are network-level solutions. Today's IP-based mobile backbone networks are designed and operated to carry the most traffic in the most reliable way without considerations of energy efficiency. A network usually builds many redundant links and aggressively over-provisions link bandwidth to accommodate potential link failures and traffic bursts. While these redundant links and bandwidth greatly increase the network reliability, they also greatly reduce the network's energy efficiency as all network devices are powered ON at full capacity 24X7 but highly under-utilized most of the time. Rule of thumb states that today's backbone links are used by 40% or lower [9] in their capacity. The high path redundancy and low link utilization provide unique opportunities for energy based traffic engineering. Intuitively, when there are multiple paths between the same origin-destination (OD) pair, and the traffic volume on each path is low, one can move the traffic to a fewer number of paths so that the other paths do not carry any traffic for an extended period of time. Routers that have idle links can put the links to sleep for energy conservation. This approach can be combined to achieve network energy efficiency.

Network-level solutions require network-wide coordination of routers. The challenges are two-fold, namely how to manipulate the routing paths to make as many idle links as possible to maximize the power conservation, and how to achieve power conservation without significantly affecting network performance and reliability. Since energy based traffic engineering uses fewer number of links in a network at any moment, it is important to make sure that links are not overloaded.

The scope of the thesis work focuses on Energy-Efficient Traffic Engineering (EE-TE) algorithm, which is based on load concentration traffic engineering mechanism [10] that reduces network power consumption while still maintaining network performance at desired levels. EE-TE is formulated as a Mixed Integer Programming (MIP) problem based on Multi-Commodity Flow (MCF) model with a total power saving as the objective to be maximized. Performance requirements such as maximum link utilization is considered as one of the important constraints in the problem. While this problem formulation bears similarity to that of traditional traffic engineering (i.e. load balancing mechanism) research, the main contribution of this work is the solution results. Traditional traffic engineering and energy based traffic engineering (i.e. load concentration mechanism) have two opposite optimization goals: the former tries to spread traffic evenly to all the links, while the latter tries to concentrate traffic to a subset of the links in a network. The main goal of the thesis work is to implement the EE-TE algorithm and solve using random network topologies and traffic demand profiles with few assumptions demonstrating that it is both promising and feasible for IP-based mobile backbone networks. The Energy-Efficient Traffic Engineering (EE-TE) algorithm is evaluated in terms of power saving, link utilization and traffic split ratio mechanism. The solution of the EE-TE model obtains the maximum power saving for the network as well as the link utilization for each link. In addition, the solution also determines which paths to use for each OD pair and how to split traffic among these paths. Results show that EE-TE can achieve between 20% to 40% power saving on line-cards under the constraints that the maximum link utilization is below 50% depending on the network topology and actual traffic demand predicted in the network. In the evaluations, it is assumed that each line-card is connected to a single link; therefore a line-card can be put to sleep when there is no traffic on the link. The main focus of this thesis work is to investigate traffic engineering mechanisms, and also show that in evaluation how it can be realized in network-level for IP-based mobile backbone networks.

1.2 Organization of Thesis

The rest of the thesis work is organized as follows. chapter 2 gives an background information on widely used terminologies such as Intra-domain traffic engineering, Multi-Commodity Flow model followed with the previous work related to the traffic engineering mechanisms in optimizing the network. chapter 3 gives an overview of state-of-the-art IP-based mobile reference network and Traffic Engineering framework. The architecture of a typical mobile network and an outline of traffic engineering framework guidelines will be discussed. chapter 4 gives an overview of Energy-Efficient optimization strategy followed with Router's power model, assumptions and design constraints. chapter 5 introduces the energy-efficient traffic engineering (EE-TE) algorithm formulation by defining each constraints considered in the model. A structure of a EE-TE model implementation and it's pseudo code will also be covered here. chapter 6 evaluates the EE-TE model based on power saving scenarios for various constraints to see the network performance with the help of random network topologies and traffic demand profiles being considered in this scope. Finally, conclusion and future work are presented in chapter 7.

Chapter 2

Background

In this section, lets have a closer look in understanding the principle terminologies which are considered in this thesis work and also related work investigated based on traffic engineering and routings.

2.1 Traffic Engineering

Traffic engineering (TE) is a method that jointly optimizes network efficiency and performance to the current network conditions. Broadly speaking traffic engineering covers all network related concepts, such as network traffic measurement, analysis, modeling, characterization, simulation and control. Basically the only network related concept, that is not part of TE is network engineering, manipulates network resources and manages the network on a long time scale. The forecasting of network usage and investment in the right parts and recourses in the network is essential in cost-efficient network business. Carefully designed network planning also helps utilizing TE methods and it improves the network efficiency because network optimization becomes an easier, sometimes trivial task. In this thesis, the focus is on Intra-domain Traffic Engineering for IP-based mobile backbone networks.

2.2 Intra-domain Traffic Engineering (TE)

Traffic engineering depends on having a set of performance objectives that guide the selection of paths as well as effective mechanisms for the routers to select paths that satisfy the objectives.

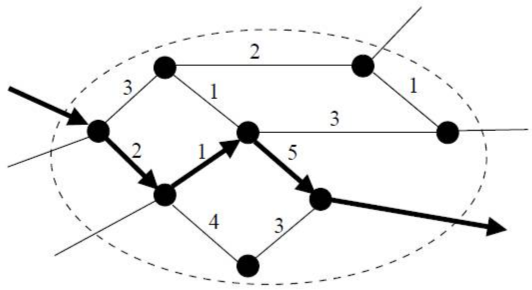

FIGURE 2.1: Shortest Path Routing within an AS

Most existing IP networks run Interior Gateway Protocol (IGPs) such as OSPF(Open Shortest Path First) [11] or IS-IS (Intermediate System - Intermediate System) [12] that select paths based on static link weights configured by network operators. Routers use these protocols to exchange link weights and constructs a complete view of the topology inside the Autonomous System (AS) as shown in Figure 2.1. Then, each router computes shortest paths (as the sum of weights) and creates a table that controls the forwarding of IP packet to the next hop in its route. In this thesis work, focused on techniques for selecting the paths based on k-shortest paths [13] rather than the underlying mechanisms for packet forwarding. Traditionally, IP forwarding depends on the destination address in the IP header of each packet. In the recent times, routers running Multi-Protocol Label Switching (MPLS) can forward packets based on the label in the MPLS header. In either case, it is concerned only with how the path is chosen rather than how the packets are forwarded.

2.3 Multi-Commodity Flow Problem

Network flows is a problem domain that lies at the cusp between several fields of inquiry including applied mathematics, computer science, engineering, telecommunications, management and operations research. There are different kinds of network flows approached depending on the problem. The most fundamental network flow problem is minimum cost flow problem models, where it deals with a single commodity over a network. Multi-commodity flow problems arise when several commodities use the same underlying network. The commodities may either be differentiated by their physical characteristics or simply by their origin-destination (OD) pairs. Different commodities have different origins and destinations, and commodities have separate mass balance constraints at each node. However, the sharing of the common arc capacities binds the different commodities together. In fact the essential issue addressed by the multi-commodity flow problem is the allocation of the capacity of each arc to the individual commodities in a way that minimizes overall flow costs.

In a communication network, nodes represent origin and destination stations for messages and arcs represents transmission lines or channel/link capacities. Messages between different pairs of nodes define distinct commodities; the supply and demand for each commodity is the number of messages to be sent between the origin and destination nodes of that commodity.

2.4 Related Work

There have been a considerable amount of research in traffic engineering (TE) recently. Traffic engineering aims at making more efficient use of network resources in order to provide better and more reliable services to the customers. Current traffic engineering and routing in IP networks are performed in both the management plane and the control plane. In the management plane, a network operators configure weights of the links in a network, which indirectly control the selection of routing paths. These weights can be set inversely proportional to the link capacities as recommended by Cisco [14], or they may be optimized for traffic engineering objective functions in a network with given traffic demands. Computing optimal link weights is NP-hard, therefore heuristics have been developed [15, 16]. However, these heuristics are still computationally intensive: for current backbone networks they may require hours of computation time. When the link weights are computed in the management plane, routing is performed in the control plane based on link-state routing protocols such as OSPF and IS-IS. The forwarding paths between two routers is the shortest path considering the sum of the link weights along the path. When a topology changes, as when a link fails, the forwarding paths are recomputed in the routers individually using Dijkstra's algorithm [17], which is efficient and can be done in milliseconds. However, this approach does not provide efficient traffic engineering since the link weights are not re-optimized for the new topology.

Current routing protocols in the network calculates the shortest path to a destination in some metric without knowing anything about the traffic demand or link load. Manual configuration by the network operator is therefore necessary to balance load between available alternate paths to avoid congestion. One way of simplifying the task of the network operator and improve use of the available network resources is to make the routing protocol sensitive to actual traffic demand.

Gupta et al. identified the power saving problem in the internet and proposed sleeping as the approach to conserve energy [7]. Specifically, they suggest two options - uncoordinated sleeping which works at link level and coordinated sleeping which operates at network level. This approach works effectively in LANs due to its specific traffic patterns; however, it might not be applicable to backbone networks where different traffic patterns have to be considered for IP-based mobile backbone networks. In [8], Nedevschi et al. propose the buffer-and-burst approach which shapes traffic into small bursts to create greater opportunities for network components to sleep. The same work also brings up the idea of rate-adaptation, which adjusts operating rates of links according to the traffic condition. This work is also focused on links in network-level solutions.

Chabarck et al. explore power-awareness in the design of networks and routing protocols in wire-line networks [18]. The authors reveal the significant power saving potential in operational networks by including power-awareness, but they do not come up with specific energy efficient routing design.

Vasic et al. propose EATe [19] to reduce power consumption through traffic engineering. EATe considers sleeping of links and routers as well as link rate adaptation. EATe achieves its routing decisions in a distributed fashion via router coordination and thus requires routers to be able to send announcement and feedback to each other. In contrast, the EE-TE model is mostly compatible with current operation practice.

Antonio et al. propose an energy saving routing solution called Energy-Aware Routing (EAR) algorithm [20], compatible with classical link-state routing protocols, i.e. OSPF, that could be easily implemented in a network. According to the OSPF protocol each router computes its own Shortest Path Tree (SPT) using Dijkstra algorithm. The EAR algorithm is based on the definition of two sets of routers, the "exporters routers" and the "importers routers". The routers belonging to first set calculate the SPTs that are used to fix the packet routing paths; the router of the second set take as a reference the SPTs of the exporter routers to modify their own SPTs and to determine the links that have to be switched off. However this solution needs some changes in routing design and also it calculates only single shortest path where other paths are eliminated which might be a better optimal path in terms of energy saving. In contrast, in the EE-TE model maximum number of K-shortest path are considered for each OD pair as this will not eliminate even a single energy efficient path.

Internet traffic engineering is a widely studied topic. Fortz and Thorup first propose the idea of IGP link weight optimization for the purpose of traffic engineering [21, 22]. However, frequent changes to link weights would cause problems such as network-wide routing convergence and traffic shift.

Heller et al. propose ElasticTree [23], which optimizes the energy consumption of Data Center Networks by turning off unnecessary links and switches during off-peak hours. ElasticTree also models the problem based on the MCF model, but is focused on Fat-Tree or similar tree-based topologies. ElasticTree takes link utilization and redundancy into consideration when calculating the minimum power network subset, and is implemented using OpenFlow [24].

MATE [25] and TeXCP [26] perform traffic engineering by splitting traffic among multiple MPLS paths. MPLS-based traffic engineering can achieve optimal routing, but does not scale well as the size of the network grows. In contrast, the EE-TE model is suitable for performing traffic engineering through hybrid IP/MPLS routing. This helps to achieve optimal routing with a small number of MPLS tunnels, and thus alleviates the scalability problem.

Optimization techniques have been applied to many different problems in communication networks. Earlier, the work by Applegate and Cohen uses an optimization framework to investigate how changing traffic demand can affect network utilization [27]. In this thesis work also, the approach in investigating energy efficiency is similar to that considers both network topology and traffic demand.

There have been research initiatives in centralized control recently [28–30], which advocate that the control of an AS should be performed in a centralized fashion with *direct control*: Instead of manupulating link weights, which influence indirectly the forwarding decisions in individual routers, a centralized server controls the routers forwarding decisions directly. A centralized control scheme simplifies the decision process, and reduces the functions required in the routers. An architectural framework of Central Control Unit is elaborated in chapter 3. However, in this thesis work focuses more on traffic engineering and routing.

2.5 Approach to EE-TE Model

The approach of EE-TE has been summarized in this section. The energy-efficient traffic engineering (EE-TE) model is approached based on multi-commodity flow problem and applies intra-domain traffic engineering mechanism. Several constraints are designed in this EE-TE model such as link utilization, thresholdlevel, traffic split ratios and capacity constraints. Instead of tuning link weights, the Central Control Unit (CCU) computes the EE-TE model based on the input data collected from all routers in the network. To compute EE-TE model, it is solved efficiently using CPLEX and the problem will be of Mixed Integer Programming (MIP) type, where parts of a variables will be declared as binary (integer) variables. Solving an MIP problem will obtain the total power saving as the objective to be maximized. Also, it calculates link utilization, energy efficient routing path per link with traffic split ratios depending on the path chosen.

Chapter 3

Mobile Network Architecture and TE Framework

3.1 Reference Mobile Network: State-of-the-art

Before discussing the details of how the packet core infrastructure provides mobile solutions, it is important to first understand a few things about the mobile network architecture and the evolution towards 3G and 4G networks. Though the focus of this thesis work is on the mobile packet backbone network, the end-to-end mobile architecture is relevant and essential in terms of understanding the core. To this end, this section will introduce a generalized wireless architecture to establish some context followed with packet core network evolution which has similar attributes and also the wide use of IP mobile backbone networks.

3.2 Generalized View of the Mobile Network Infrastructure

As illustrated in Figure 3.1, any mobile network infrastructure can be generalized into two main parts: the Radio Access Network (RAN) and the Core Network (CN).

3.2.1 Radio Access Network

The radio access network includes all base stations in a mobile network, such as 2G/3G/4G base stations as well as future non-3GPP base stations (WiMAX/Wi-Fi - pico/femto cells). It connects the cell sites (base stations and antennas) to the core network via

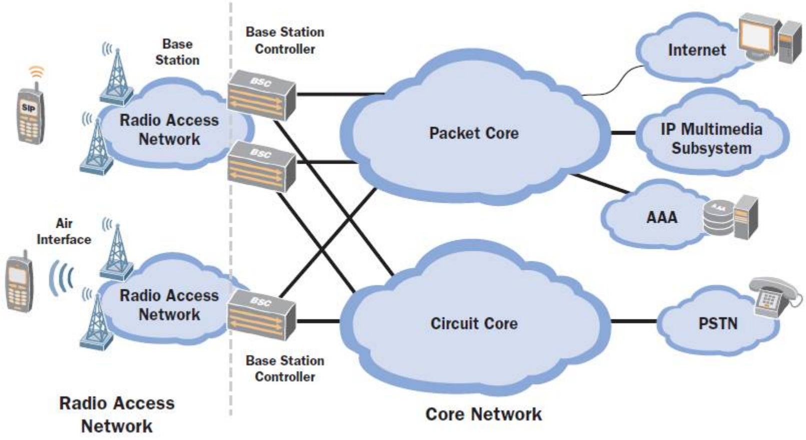

FIGURE 3.1: Mobile Network Infrastructure - General View

backbone networks. The RAN also consists of a Base Station Transceiver and Base Station Controllers (also known as Radio Network Controllers, or RNCs, according to the terminology of certain networks such as UMTS).

The access network also known as Mobile Backhaul Network (MBH) connects the cell sites to the core network and aggregates the traffic from the cell sites in several stages. The access network can be further divided into two segments, a Low RAN (LRAN) and a High RAN (HRAN) as shown in Figure 3.2

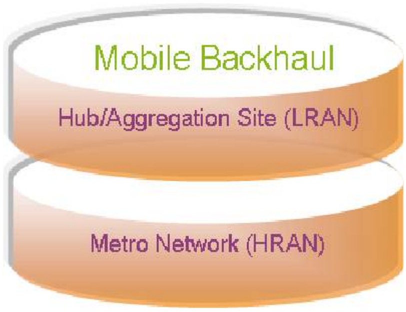

FIGURE 3.2: Mobile Backhaul Networks

3.2.2 Low Radio Access Network (LRAN)

As illustrated in Figure 3.3 the LRAN provides the last mile of connectivity for the cell sites and typically aggregates traffic from 10 to 100 Radio Base Stations (RBS) sites and feeds it into the HRAN. Typical LRAN is structured in a tree topology. The aggregation takes place at Layer2, e.g. Ethernet, ATM, or TDM. LRAN can use multiple transport technologies, depending on operator strategy and availability at the site. Microwave usually provides the lowest total cost of ownership when no other infrastructure is available at the cell site. In higher density areas electrical or optical transport can be used.

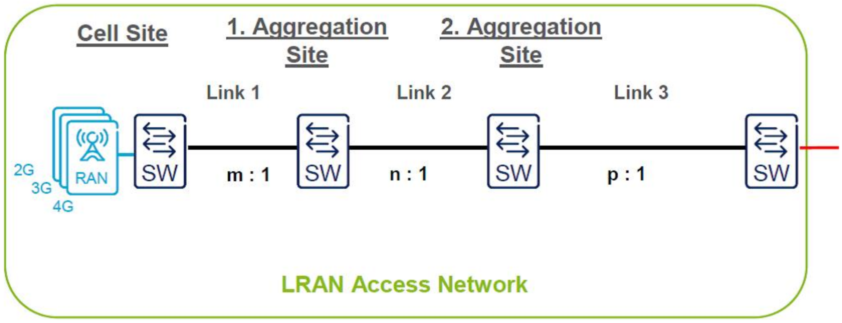

FIGURE 3.3: Schematic view of a LRAN structure

3.2.3 High Radio Access Network (HRAN)

As illustrated in Figure 3.4 the HRAN typically aggregates traffic from several LRAN networks using an existing electrical or optical fiber network. The HRAN can be organized in a ring structure, e.g. build on an optical METRO network providing Layer2 Ethernet connectivity between the LRAN and the core network, or an Multi-Protocol Label Switching (MPLS) based routed network, providing Layer3 (L3) IP transport between LRAN and the core network.

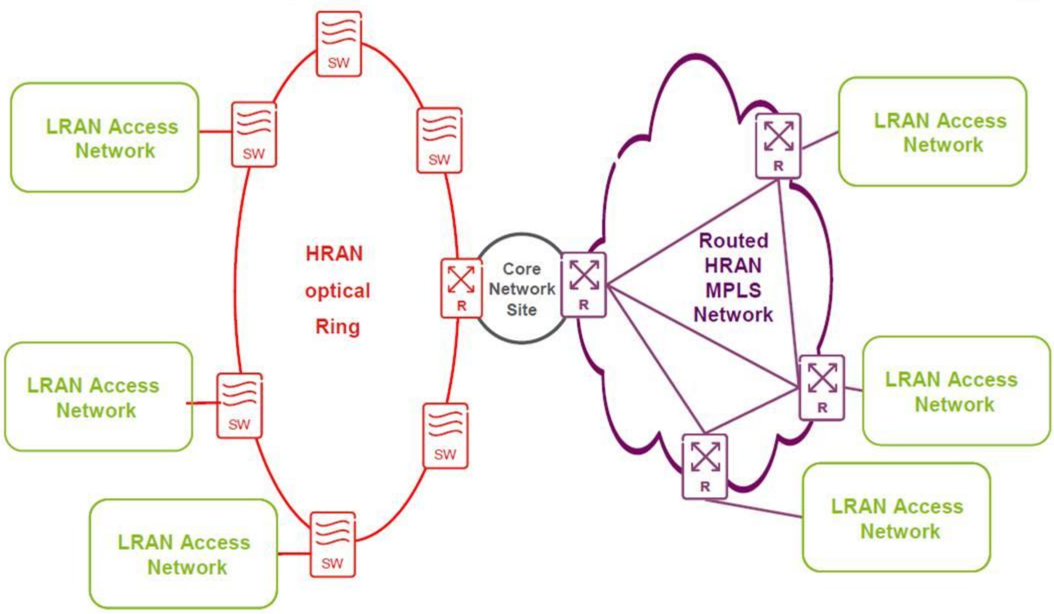

FIGURE 3.4: Schematic view of a HRAN structure

3.2.4 Core Network

The Core Network can be divided up into an IP Multimedia Subsystem (IMS), a Circuit Switched (CS) domain, and a Packet Switched (PS) domain. IMS is a collection of network elements that provide IP-based multimedia-related services like text, audio, and video. The data related to these services is further transmitted through the PS domain. In short, the Core Network includes the CS, PS, and IMS domains. A CS-type connection is a traditional telecommunication-style connection with dedicated resources allocated for the duration of the connection. In contrast, in a PS-type connection the information is typically transported in packets and each packet is routed in a distinct and autonomous fashion. The following sections discuss specific details about packet core evolution and IP mobile backbone networks as this work is focused only on packet switch elements.

3.2.5 Evolved Packet Core

Evoled Packet Core is designed to support all Communication Service Providers (CSP) services, including data, high quality voice and multimedia. Current packet switched mobile networks have been designed to accommodate user initiated communications. EPC is also known as System Architecture Evolution (SAE). Evolved Packet Core plays an essential role in unifying mobile networks, offering interoperability between LTE and existing wireless access technologies. With these capabilities, it offers smooth transition to 3GPP R8 architecture and technology, independent of the access technologies currently deployed in Communication Service Providers (CSP) networks.

Evolved Packet Core is based on 3GPP R8 architecture [31] which specifies as a flat all-IP network architecture that is ready to carry all CSP services, from mobile broadband data to high quality voice and multimedia. EPC links the wireless access networks to the service and content networks. In this position, it is the natural point to enforce charging and traffic treatment policies to allow the CSP to stay in control of how network resources are used.

Evolved Packet Core is required to support the LTE radio network, subscriber mobility and to offer interoperability between LTE and other CSP access networks. Handovers between LTE and other networks must be supported and service continuity must be ensured. 3GPP R8 specifies optimized interworking between LTE and other 3GPP accesses and CDMA to minimize handover times.

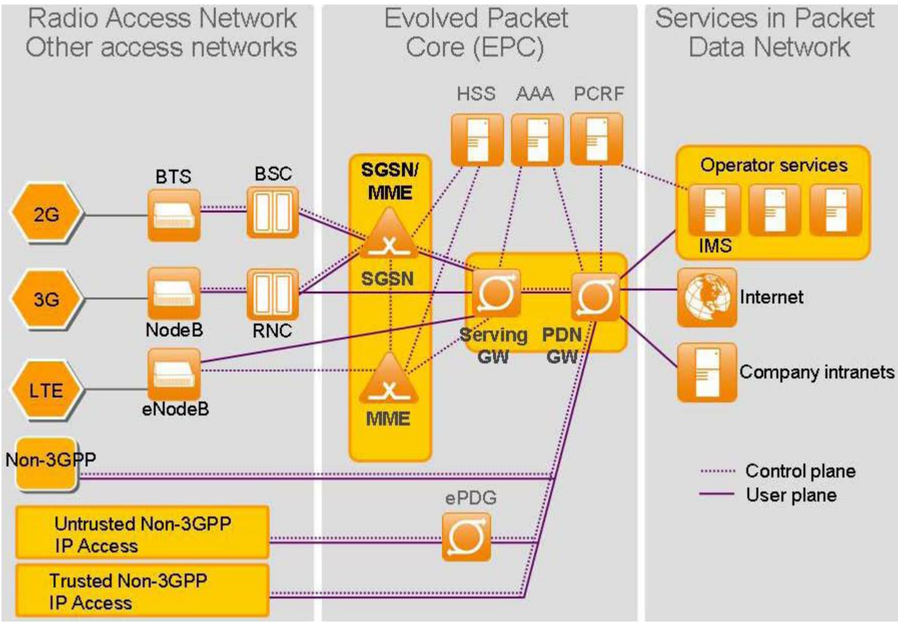

FIGURE 3.5: Overview of Evolved Packet Core [32]

3.2.6 EPC Architecture

The most significant design of EPC architecture is an all-IP network where all services are provided over IP based connections. However, interoperability with current circuit switched networks is provided to ensure, for example, voice call continuity. Interoperability with existing packet switched networks is also specified, allowing subscribers to move easily between different access networks. As illustrated in Figure 3.5 shows the EPC architecture. Simplifications are introduced by implementing the radio network functionality in a single node, the evolved Node-B. Traffic flows are separated in the core network user plane and control plane, allowing a more flexible network architecture. The user plane data is carried from the eNode-B directly to the S/P-GW. To handle control plane traffic, EPC introduces a Mobility Management Entity (MME) that takes the role of SGSN as a dedicated control plane element, taking care of such things as session and mobility management. The Serving Gateway (S-GW) and Packet Data Network Gateway (P-GW) together take the role of the current GGSN. These functions can be implemented in one or two separate network elements. S-GW acts as a user plane anchor for mobility between the 2G/3G access system and LTE access system. P-GW acts as mobility anchor for all accesses and as a gateway towards the Internet, company intranets and CSP services. It acts as a centralized control point for policy enforcement, packet filtering and charging.

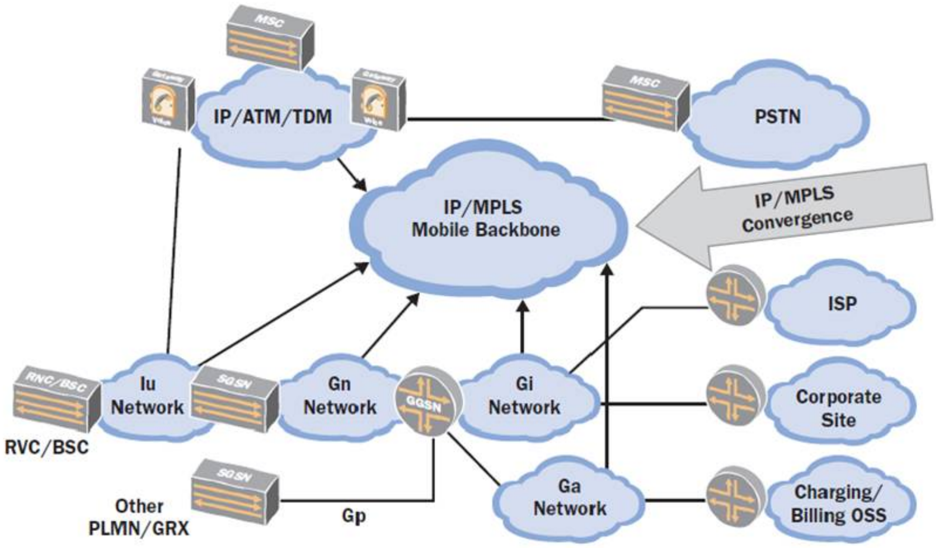

Figure 3.6: General Mobile IP/MPLS Backbone

3.2.7 Mobile IP Backbone Network

As illustrated in Figure 3.6, gives an overview of typical Mobile IP backbone network. The packet-switched IP backbone was first introduced with 2.5G GPRS networks, and since then most packet-based data services have been built on the underlying assumption of IP. Even today, many of the applications are best-effort in nature (such as text, multimedia messages, and ringtone downloads) and are being served by typical best-effort IP backbones. The mobile backbone network solution provides the IP connectivity between all nodes in the core network taking into account the important issues of performance, scalability, security and manageability. This comprehensive IP infrastructure connects the core network, service network, radio access networks and external networks, such as the internet and corporate sites.

Mobile IP backbone networks is based on Multi Topology (MT) routing is one of the most promising technologies providing robust traffic engineering also in failure scenarios. MT for IS-IS and OSPF have Internet standards defined in [33] and [34]. In MT, routers maintain several independent logical topologies allowing different kinds of traffic to be routed independently through the network. MT techniques can be used in online and offline TE, centralized and distributed TE and IP and MPLS. The drawback of this method is massive consumption of computation resources and memory in router [35].

3.3 Traffic Engineering Framework

Realizing the need for EE-TE model in IP backbone networks requires coordination among all routers in the network with the help of Central Control Unit [8]. An outline of such coordination and its aspects can be seen in this section. The basic principle in EE-TE design is to use existing protocols and mechanisms as much as possible for the benefits of compatibility and deployablility. In this scope, it is assumed that networks run both OSPF (or any link state routing protocols) and MPLS.

3.3.1 Central Control Unit

As in conventional traffic engineering, EE-TE relies on a logically centralized controller in the Network Operation Center (NOC) to make decisions on traffic engineering. The physical implementation of such a controller can have hardware redundancy for better reliability. Figure 3.7 shows the Central Control Unit collects input information (i.e. network topology and traffic demand) from routers, solves the EE-TE problem to get new routing configurations (i.e. which links are up and how much traffic on each link), and disseminates the results to routers. Each router will then turn ON/OFF some line-cards or ports according to the EE-TE solution. As traffic demand changes over time and sometimes unpredictably, the process described above needs to be done periodically. The frequency of such routing adjustment depends on how often the traffic demand changes and by how much.

3.3.2 Gathering Input Information for EE-TE

The CCU collects network topology and traffic demand from OSPF's Link State Advertisements (LSAs) defined in RFC2328 [11]. In OSPF, each router floods its LSAs whenever its link state changes. Thus the CCU can readily collect all the link state information and compile the up-to-date network topology.

Both the network topology and traffic demand information are collected by the CCU passively. The CCU does not poll any specific router, nor has any explicit point-to-point conversation with any individual router. All information is announced via LSAs. This design choice is compatible with existing mechanisms, simplifies operations, and also inherits the delivery reliability provided by LSA flooding.

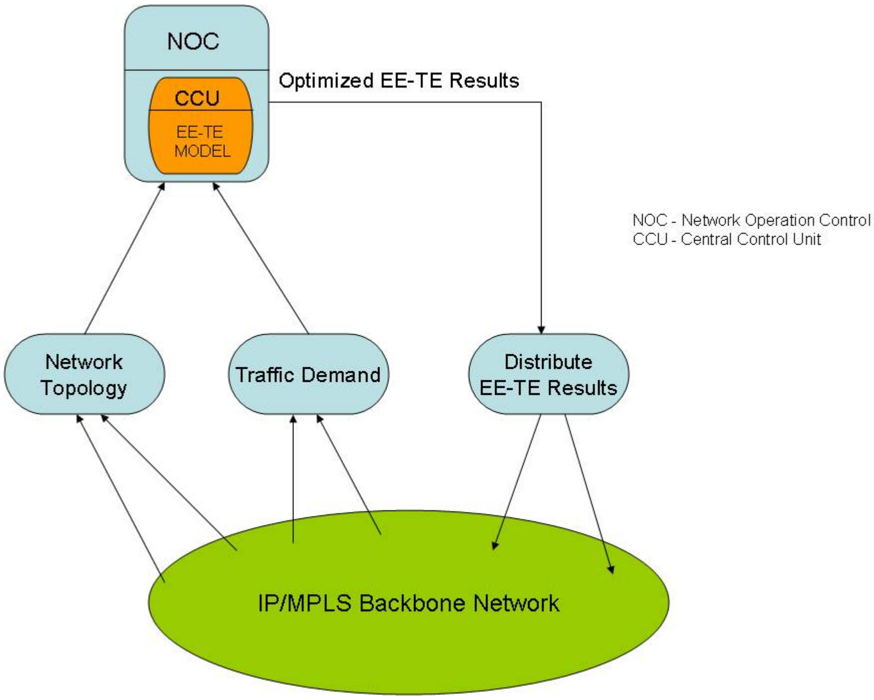

FIGURE 3.7: Traffic Engineering Framework with CCU

3.3.3 Distributing EE-TE Results

With the network topology and traffic demand, the CCU solves the EE-TE problem to get which links to be turned ON or OFF, and distributes this information to routers via the Traffic Engineering Metric (TE-Metric) attribute, another extension to OSPF defined in RFC 3630 [36]. Both TE-LSA and TE-Metric messages are flooded in the network as regular OSPF LSAs; therefore they can reach all routers as long as the network is connected. Two routers can exchange messages to confirm that they are ready. Such messages can be MPLS signaling messages, OSPF Hello messages, or simple messages designed specifically for this purpose.

Chapter 4

Overview of Energy Efficient Optimization Strategy

Today's mobile IP backbone network usually have redundant and over-provisioned links, resulting in low-link utilization during most of the time. The reason is that network operators prioritize maximum reliability which is ensured by redundancy in the network. Suppose two links are available between two routers say A and B, then traffic is evenly distributed over both paths. At times of low traffic demand between Origin-Destination (OD) pairs, the link usage is not justifying the energy consumption for keeping up two links. Consider Figure 4.1 shown below, two routers A and B are connected by several links. The basic idea is to take one of the multiple links out of service by determining the traffic volume on each link flow and finally switching off the least-utilized links between routers A and B. This example illustrates the need for new traffic engineering mechanism following today's router configuration model and assumptions considered for the traffic engineering model in the following sections.

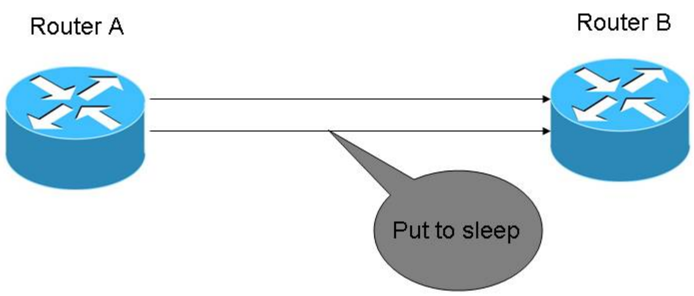

Figure 4.1: Router and Link connection

4.1 Need for New Traffic Engineering (TE) Mechanism

The traffic engineering (TE) focuses on facilitating power management at network level by routing traffic through different paths to adjust the workload on individual routers or links in a network. Combining high path redundancy and low-link utilization, provide a unique opportunity for energy efficient traffic engineering as illustrated in the theoretical example shown in Figure 4.2. Traditional traffic engineering which is nothing but load balancing approach, spreads the traffic evenly in a network as shown in Figure 4.2(a), trying to minimize the chance of congestion induced by traffic bursts. However, in load concentration traffic engineering in Figure 4.2(b), one can free some links by moving their traffic onto other links, so that the links without traffic or with low traffic can go to sleep for an extended period of time. This should result in more power saving in the network than pure opportunistic links sleeping because, the sleep mode is much less likely to be interrupted by traffic. The solution to this approach is based on load concentration mechanism inspired from energy based traffic engineering [10] which is discussed more in detail in chapter 5. In order to evaluate this EE-TE model, few assumptions in the IP-mobile backbone network will be considered such as configuring the router's LIC's and its power characteristics, network topologies and traffic demand profiles which are defined as input to the model.

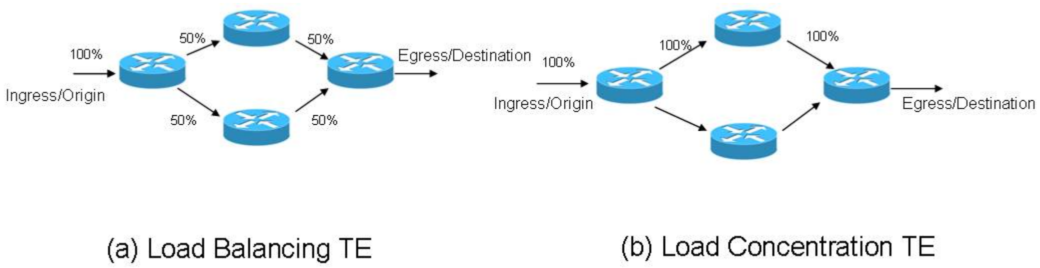

Figure 4.2: Load Balancing vs Load Concentration

4.1.1 Router Configuration and Power Model

As this traffic engineering focuses on energy efficient routing in IP-backbone networks, there is a need to analyze the hardware configurations of a router with respect to energy consumption. Since the main objective is to save power by turning off links or interchangeably, putting line-cards (or their ports) to sleep. So, here planned to analyze the today's existing typical hardware to know it's hardware configuration and it's power characteristics. Table 4.1 and 4.2 show typical configuration of SmartEdge SE 400 and SE 800 routers with low to high interface rates. It is possible to predict, line-cards contribute a significant portion to the total power consumption of a router.

Table 4.3 and 4.4 shows the power budget of a SmartEdge SE 400 and SE 800 routers. All the line-cards together consume 418.56/645.12 Watts for SE 400/SE 800 router's respectively.

CARDTYPE	WATTS
1-port 10 Gigabit Ethernet(PPA2)	130.56
1-port 10 Gigabit Ethernet/OC-192C DDR	96
10-port Gigabit Ethernet DDR(PPA2)	96
10-port Gigabit Ethernet DDR(PPA2)	96
XCRP4	105.6
XCRP4	105.6
FAN and ALARM UNIT	76.8

Table 4.1: SE 400 Typical Router Configuration [37]

CARDTYPE	WATTS
1-port 10 Gigabit Ethernet(PPA2)	130.56
1-port 10 Gigabit Ethernet/OC-192C DDR	96
10-port Gigabit Ethernet DDR(PPA2)	96
10-port Gigabit Ethernet DDR(PPA2)	96
1-port 10 Gigabit Ethernet/OC-192C DDR	96
10-port Gigabit Ethernet DDR(PPA2)	96
1-port 10 Gigabit Ethernet(PPA2)	130.56
XCRP4	105.6
XCRP4	105.6
FAN and ALARM UNIT	76.8

Table 4.2: SE 800 Typical Router Configuration [38]

SLOT No.	COMPONENTS	WATTS
1,2,3,4	Line-cards	418.56
5,6	Route Processors	211.2
	Chassis	76.8
	Total Inuse Power	706.56

Table 4.3: SE 400 Power Budget [39]

SLOT No.	COMPONENTS	WATTS
1,2,3,5,9,12	Line-cards	645.12
7,8	Route Processors	211.2
	Chassis	76.8
	Total Inuse Power	933.12

Table 4.4: SE 800 Power Budget [40]

4.1.2 Assumptions and Design Constraints

Energy Efficient Traffic Engineering (EE-TE), like any other power saving mechanisms, needs support from the underlying hardware. The following assumptions are made based on today's typical router architectures and hardware in designing the energy-efficient Traffic Engineering (TE) model. However, most of them can be relaxed to take advantage of better hardware support in the future without impacting the basic energy efficient problem formulation and solution.

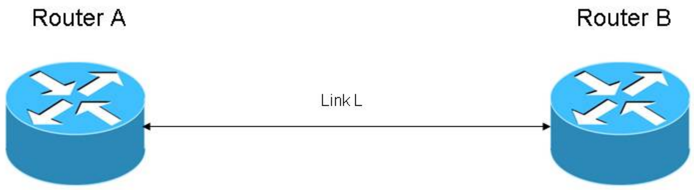

FIGURE 4.3: Bi-directional Links

A link is defined as a two-way bi-directional connection between network elements as shown in Figure 4.3 in which a link L connects from A to B and B to A. In this scope, assumed network elements as routers consisting of "Line Interface Cards(LIC's)" and other components. Each LIC has a number of ports. A link is a connection of one port in a LIC in router A with a port in a LIC in router B shown in Figure 4.4. The multiple ports of one line-card may connect to the same remote router, making it a bundled link, or connect to different remote routers. When a link is put to sleep, the port that connects to the link can go to sleep. If all ports in a LIC are powered off, then the entire LIC may be put to sleep mode, resulting in more power saving due to the line-card's base power consumption. Ideally, if all LIC's in one router are put to SLEEP mode, then the entire router could be put to sleep. But this requires that the router is still capable of receiving wake-up signals through signaling protocols.

This scope addresses only point-to-point connections, (i.e.) direct links between two routers in the network. In this scope, a central control entity will be assumed to take care of controlling all the routers in the network as discussed in traffic engineering framework. This Central Control Unit basically collects the information needed to take the decision to turn off links. This means, it sends a signal to all routers in the network. The Central Control Unit will analyze the status (network topology and current traffic volume) of the entire network to decide which link will be taken out of service. Based on the input information, CCU will solve EE-TE model and calculate the utilization for each link in the network and then decide which links to put into sleep mode. After a link has been removed, the network will need time to re-configure itself. During this time, no further changes shall be made in the network. After this time has elapsed, the

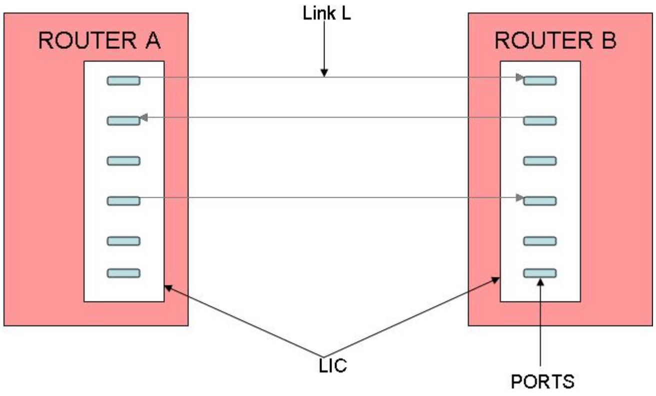

FIGURE 4.4: Router-Port-LIC-Link Connection

Central Control Unit may check for further optimizations again. This ensures that no links are taken out in parallel, which could lead to negative side-effects such as part of the network being unreachable.

Chapter 5

Energy Efficient Traffic Engineering Model

Energy Efficient Traffic Engineering (EE-TE) is based on load concentration mechanism as discussed in previous chapter 4. The purpose of this model is to find energy efficient routing paths so that it achieves optimal power saving in the network, given the input network topology and the traffic matrix to find an optimized routing solution. The network topology will be an IP-based mobile backbone network consisting of routers as network elements and a link is used to connect all routers in a network. The traffic matrix basically contains the amount of traffic volume to be carried on each link between each OD pair in the network based on assumptions. The main scope of this thesis work is to formulate and implement the EE-TE model based on equal-cost K shortest paths with constraints such as link utilization, traffic split ratio and finally solve using CPLEX [42]. By developing this energy-efficient traffic engineering (EE-TE) model, it can achieve maximum power saving from turning off links or line-card's completely as well as satisfying performance constraints which applies to energy optimized routing including link utilization.

5.1 EE-TE Model Problem Formulation

The general EE-TE model problem contains the input information such as network topology and traffic demand for each OD pair. Based on the input data, this model can be formulated to calculate the link utilization to find the energy efficient routing path by applying a threshold level (bound) on the link utilization calculated for each link. Using calculated results, it possibly helps to analyze which links are highly utilized and least utilized, so that finally re-route the traffic on the chosen energy efficient routing path obtained from the model.

The energy efficient traffic engineering (EE-TE) problem can be formulated based on MCF model, shown in subsection 5.1.2. Then the model can be implemented using OPL [41] mathematical modeling language. Once the model is formulated, the problem can be solved using MIP solvers such as CPLEX [42].

The network can be modeled as a directed graph G = (V,E), where V is the set of nodes (i.e. routers) and E is the set of links. A port can be put to sleep if there is no traffic or low traffic on the link, and a line-card can be put to sleep if all its ports are asleep. Initially this model focuses on the case of one port per LIC. So the power saving from turning off one port is P_L. The objective is to find a routing that maximizes the total power saving of a Port or entire LIC in the network.

5.1.1 EE-TE Mathematical Model

The idea of energy-efficient traffic engineering (EE-TE) problem can be approached mathematically based on the MCF model. To formulate the model, we have to define and declare the variables, parameters and constraints which are required to formulate the model mathematically. A list of parameters and variables and their notation is shown in Table 5.1 and 5.2 for the proposed mathematical model.

PARAMETERS	DESCRIPTION
NodeNum(N)	Set of Node ID's
LinkNum(L_N)	Set of Link ID's (each link indexed)
PathNum(K)	Maximum Number of Shortest Paths for each OD pair
ThresholdLevel(T_L)	Bound on Maximum Link Utilization(MLU)
Traffic Demand($D_{s,t}$)	Traffic demand between each OD pair
Capacity(C_L)	Capacity of link L
Paths($R_i^{s,t}$(L))	1 - if the i^{th} path contains link L 0 - otherwise
Pair(B_L)	Bi-directional links are paired between each OD pair
LinkPower(P_L)	Power consumption of Link L

Table 5.1: EE-TE Parameter Notations

The parameters shown in the table are the input to the optimization problem. All these information will be provided in a data model which is required to solve this optimization problem. All these parameters are assumed based on the data gathered from today's typical router architecture SmartEdge (SE 400 and SE 800) routers which is already discussed in subsection 4.1.1.

VARIABLES	DESCRIPTION
TrafficFlowPerLink ($F_L{}^{s,t}$)	Traffic Demand between OD pair routed through link L
Traffic Split Ratio ($\alpha_i{}^{s,t}$)	Ratio of Traffic Demand between OD pair routed through the i^{th} Paths [$R_i{}^{s,t}(L)$]
Link Utilization (U_L)	Utilization of link L
PowerStateofLink (X_L)	1 - if link L is sleeping 0 - otherwise
Power Saving	To calculate the total power saving of the given network
MaxUtil (U_T)	Max. Utilization of a link will be estimated

TABLE 5.2: EE-TE Variable Notations

5.1.2 Problem Formulation

$$\text{Maximize} \quad \sum_{L \in L_N} P_L X_L \tag{1}$$

$$s.t. \quad F_L{}^{s,t} = \sum_{0 \le i \le K} R_i{}^{s,t}(L)\, D_{s,t}\, \alpha_i{}^{s,t}$$

$$\text{s,t} \in N,\ L \in L_N,\ s \neq t \tag{2}$$

$$\alpha_i{}^{s,t} = 0 \tag{3}$$

$$\sum_{0 \le i \le K} \alpha_i{}^{s,t} = 1$$

$$s, t \in N, s \neq t \tag{4}$$

$$U_L = 1/C_L \sum_{s,t \in N, s \neq t} F_L{}^{s,t} \quad , L \in L_N \tag{5}$$

$$X_L = X_{B_L} \quad , L \in L_N \tag{6}$$

$$X_L + U_L \le 1 \quad , L \in L_N \tag{7}$$

$$U_L \le U_T \quad , L \in L_N \tag{8}$$

$$\Big(\sum_{L \in L_N} X_L P_L\Big) / \Big(\sum_{L \in L_N} P_L\Big) \quad , L \in L_N \tag{9}$$

The mathematical formulation shown is based on MCF model. Now we will see how each constraint is formulated in detail for the energy-efficient traffic engineering (EE-TE) problem in the following sections. Before understanding the constraints used in the mathematical formulation, we have to justify how this approach brings efficient for the energy-efficient traffic engineering (EE-TE) problem.

Generally speaking, in existing LP models [17, 43], they search solutions in all possible paths for each OD pair, making the search space extremely large. In this EE-TE model, since it considers binary (integer) variables X_L, that denote the power state of link L make the model a MIP problem. Generally, MIP problems are NP-hard, thus its computation time for networks with medium and large sizes is of a concern. So, the models considered in the papers [17, 43] though optimizes the required objective, it does not optimize in terms of power saving for a network.

5.1.3 Practical Constraints

In order to take into account the above conditions, a problem is proposed based on energy based traffic engineering [10] in such a way that it takes those practical constraints into account when comparing with the existing LP models. So, this mathematical model proposed not only meet the required objective in optimizing the network, but also satisfies the practical constraints. Let's have a closer look at the parameters such as Paths [$R_i^{s,t}(L)$], MaxUtilization [U_T] and so on and also see how it influences the constraints effectively.

5.1.4 Parameter Paths [$R_i^{s,t}(L)$]

First let's have a look at the introduction of parameter Paths [$R_i^{s,t}(L)$]. The parameter Paths is derived based on the K-shortest path algorithm [13]. The main advantage of using K-shortest path is, it will avoid searching the solution in all possible paths during runtime. K-shortest path help to find a maximum number of shortest paths for each OD pair based on the directed graph provided as input to the model, therefore each OD pair has at most K-shortest path. Equation (2) and (4) are equivalent to the flow conservation constraints under this change. It reduces the overall computation time as well as adds the path length as another constraint. Referring to the existing LP models, it will consider all possible paths for each OD pair, making the search space extremely large. To reduce this search space and computation time, for each OD pair, we have to pre-compute its set of shortest paths and only search for solutions within this set. The idea is to make use of parameter Paths [$R_i^{s,t}(L)$] in the input model, since all shortest paths are pre-computed between each OD pair based on the parameter PathNum[K] defined for the network. The parameter Paths [$R_i^{s,t}(L)$] is declared as boolean.

Paths [$R_i^{s,t}(L)$]

1 - if the i^{th} path contains link L
0 - otherwise
i - index value for each OD pair based on PathNum [K] set
s,t - Source and Destination - (OD)
L - link

The reason for declaring this parameter "Paths [$R_i^{s,t}(L)$]" as boolean is as follows. Firstly, a shortest path is pre-computed for each OD pair based on the value K. This is done to reduce the computation time. Secondly, since each link is indexed in the network, the intention is to find which link occurs in the i^{th} path obtained from the pre-computed K-shortest path. If the i^{th} path contains link L, then it is set to 1, otherwise it is set to 0 which don't fall under this pre-computed shortest path list.

Finally, it is possible to formulate by pre-computing this parameter "Paths" for the links which occur for each OD pair based on the K-shortest path method. Since the K-shortest path are pre-computed with network topology as input, they do not change with the traffic matrix and the computation does not add run-time overhead. Note that, when K is set to be large enough, it can actually consider all possible paths for each OD pair, which will give optimal solution for the problem. However, the computation time increases with the value of K; therefore there is a tradeoff between the precision of the heuristic and the computation time. Searching solutions only within the K-shortest paths also avoids very long paths.

5.1.5 ThresholdLevel [T_L]

Another important parameter is the use of the threshold level [T_L], which adds a bound on the maximum link utilization in a network. The basic idea behind setting this parameter is to identify which links are highly utilized and low-utilized based on the calculated constraints link utilization [U_L]. A threshold level can be set to apply a bound on the links formed in the network. This helps to decide and re-route the traffic on a path in which links are above the threshold level and finally it is possible to free some of the links considered least-utilized by putting those links to sleep to optimize the power saving.

Similarly, other parameters defined in this EE-TE model also play equally an important role in this problem to optimize the energy efficiency of the network. Now lets have a closer look at the mathematical model defined in the following sections.

5.1.6 Objective

$$Maximize \sum_{L \in L_N} P_L X_L \qquad (1)$$

Equation (1) states the objective for the proposed problem. The idea is to maximize the total power saving of a link in a network where $\mathrm{P_L}$ is the power consumption of a link L and $\mathrm{X_L}$ is the binary integer variable that denotes the power states of the link L. By adding the total links in the network, equation (1) will results in how much power can be saved from the overall power consumption of a network.

5.1.7 Constraints

In this section, lets look into the constraints in detail and how the formulation is done in order to meet the required objective of this problem.

5.1.8 Subject to TrafficFlowPerLink

Subject to TrafficFlowPerLink [$F_L{}^{s,t}$]

Before finding the traffic flow per link on a path, initially we need to know the flow of links for each OD pair based on the input network topology model. So at first, all possible combinations of traffic flow per link will be loaded for each OD pair based on the input parameter NodeNum (N) and LinkNum ($\mathrm{L_N}$). Equation (2) shows that each link will be mapped to each OD pairs in the network. This helps to find the traffic flow per link on a path [$\mathrm{F_L}^{\mathrm{s,t}}$].

5.1.9 Subject to NullPaths

$$\textit{Subject to NullPaths (i,s,t) \&\&} \sum_{L \in L_N} R_i{}^{s,t}(L) = 0$$

NullPaths is basically used to obtain the traffic flow per link information which are not included in the i^{th} path of parameter Paths [$R_i{}^{s,t}(L)$]. The basic idea of this step is to find the non-existing paths (i.e. other than pre-computed shortest paths). This helps to reduce the work, since in the previous step in equation (2), a traffic flow per link will be considered in general between each OD pair irrespective of the network topology given. It will check for each OD pair, if the corresponding i^{th} path exists in the parameter "Paths" which has a pre-computed shortest path for each pair. If it exists, it will not be treated as NullPaths since it is considered to be one of the shortest paths already defined. In the other case, if it doesn't exists, that particular i^{th} path will be considered as NullPaths. Similarly, all possible NullPaths for a given network topology can be found based on this approach.

Equation (3) states that α is 0 when there is no existence of shortest paths from the parameter "Paths" ((i.e.) pre-computed shortest paths for each OD pair that contains link L). This will help to reduce the burden in computing the traffic split ratios for all paths in the network. The concept of traffic split ratios can be seen in the following sections.

5.1.10 Subject to Ratios

$$\textit{Subject to Ratios [s,t]}$$

Initially, the ratio of traffic split between each OD pair will be assigned and normalized to 1. This is done because, before computing the traffic split ratio for a particular traffic flow per link on a path [$F_L{}^{s,t}$], it is assigned that traffic flow for each link between each OD pair should be same throughout the network. However, the traffic split is purely depends on the actual traffic demand between OD pair and splits accordingly such that it is normalized. By this approach, it will help to solve the traffic split ratio mechanism used in this model.

5.1.11 Traffic Split Ratio [alpha (α)]

Once the ratios are assigned to each OD pair, now the traffic split ratio can be computed for a chosen path. To understand how this mechanism works consider the example shown in Figure 5.1

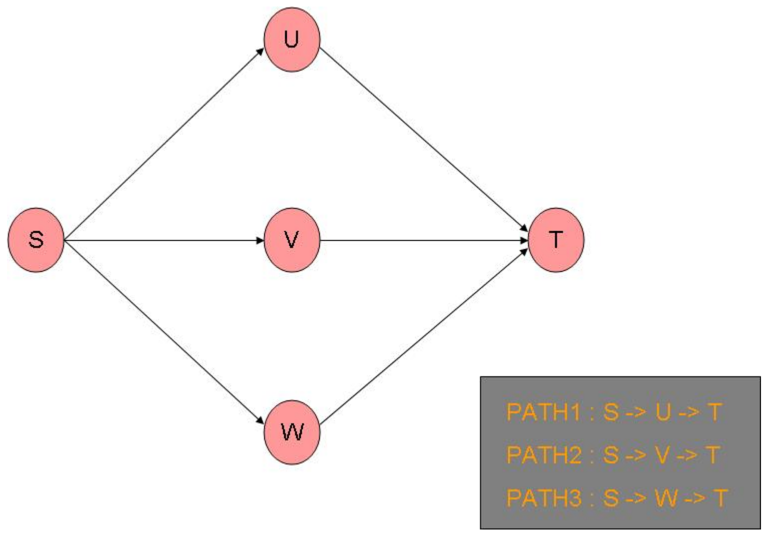

FIGURE 5.1: Traffic Split Ratio Mechanism

Suppose a traffic demand D from s to t, and a set of paths from s to t as shown in Figure 5.1, then basically using equation (4) alpha (α) tells us how to split the traffic among all the paths from s to t so that the sum of the alphas (α) should be 1. Since $\alpha_i^{s,t}$ is the ratio of traffic demand from s to t that is routed through the i^{th} shortest path, they should be summed to 1 so that all traffic from s to t are satisfied. For example in Figure 5.1, three paths are chosen between source s and destination t such as PATH1, PATH2 and PATH3 and the traffic are defined for these paths. So to split the traffic here on all three considered paths such that its ratio is equal to 1 and at the same time it splits depending on actual traffic volume and link utilization obtained on the chosen links in order not to violate the network performance constraints. If 0.5 of the traffic go to the first shortest path (PATH1), 0.3 of the traffic go to the second shortest path (PATH2) and 0.2 of the traffic go to the third shortest path (PATH3) so that it achieves in splitting the traffic and satisfies the constraint. In fact this is one of the major advantages of this model. Another advantage of performing traffic split ratios is that it can avoid changing link weights. As it might lead to network re-convergence too often.

5.1.12 Traffic Flow Per Link [$F_L^{s,t}$]

Once the traffic split ratio alpha (α) is known for the chosen paths (sequence of links) which is obtained from parameter "Paths", it is possible to compute the traffic flow per link on a path $F_L^{s,t}$ using the input traffic demand $D_{s,t}$ given. As for equation (1), the product of parameter Paths [$R_i^{s,t}(L)$] and alpha (α) traffic split ratio gives us the traffic volume from s to t on a certain path with traffic split ratios considered on that path based on the traffic demand. The idea is to check if the chosen link L from s to t is exists in the pre-computed shortest paths which can be verified in parameter "Paths". This path is chosen based on the parameter Path [$R_i^{s,t}(L)$] as already explained in subsection 5.1.4. ($R_i^{s,t}(L)$ * $D_{s,t}$ * $\alpha_i^{s,t}$) equals to ($D_{s,t}$ * $\alpha_i^{s,t}$) if link L is on that path, and 0 otherwise. The summation of equation (1) will give all links to get $F_L^{s,t}$ (i.e. traffic flow per link for a path is obtained).

5.1.13 Subject to Utilization

Subject to Utilization [U_L]

Initially based on the number of links, utilization of each link will be assigned based on the link index. By default it is set to 0. To compute the utilization of each link in the given network topology, it's link capacity C_L and it's corresponding traffic flow per link on a path $F_L^{s,t}$ should be known. Using equation (5), it will obtain the utilization of each link in the network. The variable link utilization have threshold level (T_L) factor, through which maximum link utilization is set in the model and finally it is possible to estimate which all links are under the threshold level (i.e. low utilized links) and above the threshold level (i.e. highly utilized links). By estimating the link utilization based on the threshold level, it is easy to find out which links can be put to sleep. This helps in optimizing the objective of this model.

5.1.14 Subject to Bi-direction [B_L]

As explained in subsection 4.1.2, a link is defined as as a two-way communication between routers and also each link is indexed. Hence two links will be modeled between two routers, one will be a forwarding link and the other one will be a reverse link such that both the links are paired using the bi-direction constraints. The parameter Pair

$[B_L]$ will be an input data which can be obtained with the help of network topology directed graph. Basically, this parameter Pair $[B_L]$ will have details of which link is paired with the other link. Likewise, for the whole network, all links will be paired to their corresponding bi-directional links. Equation(6) ensures that links are put to sleep in pairs (i.e. there is no in-bound traffic nor out-bound traffic for the bi-directional links formed between the routers).

5.1.15 Subject to TurnOFF [L]

In order to compute exactly which links should be turned OFF, this constraints helps in achieving the objective. It also ensures certain preventive measures to considerable amount to avoid link failures, traffic bursts etc. before the link can be put to sleep. Equation (7) states that a link can be put to sleep only if there is no traffic on it, and when it is ON, it does not carry traffic more than its capacity. Since, X_L is a binary integer variable, it is set to 0 for active links, 1 for sleeping links. The objective of the problem defined here is to find how much total power can be saved from the network. The Power state of sleepiing links $[X_L]$ variable is set to bounds either 0 (active) or 1 (Sleep). For each link defined in the network, it will be verified with the help of calculated corresponding link utilization to satisfy this constraint.

5.1.16 Subject to MaxUtilization [L]

Subject to MaxUtilization is basically an additional constraint, which helps in determining the MaxUtilization of a link for overall network based on the calculated utilization of each link in a network using Equation (5). Equation (8) not only determines MaxUtilization of a link in a network, but also satisfies that each link utilization in a network should not cross beyond its MaxUtilization of a link in a network.

5.1.17 Subject to Power Saving

This constraint Power Saving will satisfy the main objective of this optimization model. Solving equation (9) gives how much power is saved in a network for the links. To calculate the Power Saving, equation (9) is equated to sum of total power of sleeping links over the sum of total power of all links in a network.

Solving the EE-TE model, obtains the total power saving for a network as well as the utilization for each link. In addition, the solution also gives the energy-efficient routing path to use for each OD pair and how to split traffic among these paths. The computation flow of EE-TE model is shown in pseudo code format.

PSEUDO CODE : Energy-Efficient Traffic Engineering Algorithm

REQUIRE :

NodeNum (N)
LinkNum (L_N)
PathNum (K)
ThresholdLevel (T_L)
Traffic Demand ($D_{s,t}$)
Capacity C_L)
Paths ($R_i^{s,t}(L)$)
Pair (B_L)
LinkPower (P_L)

TO FIND :

TrafficFlowPerLink ($F_L^{s,t}$)	$\forall\ s,t \in N,\ L \in L_N,\ s \neq t$
TrafficSplitRatio ($\alpha_i^{s,t}$)	$\forall\ s,t \in N,\ L \in L_N,\ s \neq t,\ 0 \leq i < K$
LinkUtilization (U_L)	$\forall\ L \in L_N$
PowerStateofLink (X_L)	$\forall\ L \in L_N$

LOOP :

/* To find $F_L^{s,t}$: $\alpha_i^{s,t}$ is chosen based on existence of links in parameter Paths and PathNum(K) on each iteration for each link */

IF(link L exists in $R_i^{s,t}(L)$) **THEN**
SET $R_i^{s,t}(L) = 1$
SET $\sum\limits_{0 \leq i \leq K} \alpha_i^{s,t} = 1$
UPDATE $F_L^{s,t}$
ELSE

SET $R_i^{s,t}(L) = 0$
SET $\alpha_i^{s,t} = 0$
UPDATE $F_L^{s,t}$
ENDIF

/* To find U_L and X_L : $F_L^{s,t}$ and C_L are known */

IF$(U_L \leq T_L \ \&\& \ U_L \leq U_T)$ **THEN**
UPDATE U_L
ENDIF
IF$((X_L + U_L \leq U_T) \ \&\& \ (X_L == X_{B_L}))$ **THEN**
UPDATE X_L
ENDIF

UNTIL$((U_T > T_L)$ OR (alternative termination criterion))

RETURN THE OPTIMAL OBJECTIVE SOLUTION WITH VARIABLES

5.2 EE-TE Model Implementation

EE-TE model is formulated using Optimization Programming Language (OPL) [41] and implemented with the help of CPLEX LP format where the problem will be of MIP type. The OPL is formulated in two sections such as,

1. OPL model section
2. OPL data section

5.2.1 OPL Model Section

OPL model section is further classified into,

- Parameters section

- Variables section

- Objective section

- Constraints section

First section is the parameter section contains input parameters which are required to solve the EE-TE model. In this section the input parameters are declared and initialized. Second section is the variable section contains variables which need to be found during optimization of EE-TE model. In this section, the output variables are declared and initialized. Third section is the most important section which is the objective of the problem will be defined. In EE-TE model, the objective is defined to maximize the total power saving of a network. Fourth section is also a significant portion of the problem, where different constraints are defined based on MCF, which helps in optimizing the problem for the defined objective.

5.2.2 OPL Data Section

The OPL data section will provide the data for the defined input parameters in the model which helps in solving the problem. In EE-TE model, the following data will be set in the data section such as NodeNum (N), LinkNum ($\mathrm{L_N}$), PathNum (K), ThresholdLevel ($\mathrm{T_L}$), Traffic Demand ($\mathrm{D_{s,t}}$), Capacity ($\mathrm{C_L}$), Paths ($\mathrm{R_i}^{\mathrm{s,t}}$(L)), Pair ($\mathrm{B_L}$) and LinkPower ($\mathrm{P_L}$).

5.2.3 CPLEX LP

Once the formulation is done using OPL model, to compute the problem in CPLEX, it has to be implemented to a CPLEX readable format which is CPLEX LP format [44]. The CPLEX LP format is intended for coding LP/MIP problem data. It is a row-oriented format that assumes the formulation of LP/MIP problem. The EE-TE model is then computed using IBM ILOG CPLEX INTERATIVE OPTIMIZER [45]. CPLEX can understand the problem type based on the formulation. In this case, the EE-TE model is of MIP type, since the power state of link ($\mathrm{X_L}$) is considered to be a binary integer variable. The advantage of using CPLEX in general, is considered to be the most powerful optimization solvers. The CPLEX interactive optimizers helps in producing better results using command line interface (CLI). The EE-TE model formulation in OPL format and its implementation example in CPLEX LP format is shown in Appendix A and Appendix B sections.

Chapter 6

Evaluation and Results

In this chapter, EE-TE model is evaluated and shown that it is able to achieve considerable power savings using random network topologies and traffic demands with minor impact of network performance.

6.1 Experiment Setup

To understand how the evaluation is performed, an experimental setup needs to be clearly defined before directly providing the results. Lets take a closer look at the random network topologies and traffic demands configuration for EE-TE model being considered in this evaluation.

6.1.1 Network Topology

One of the main input to the EE-TE model is the network topology. Two different random network topologies are considered which helps in evaluating the EE-TE model.

NETWORK	TYPE	NODES	LINKS
TOPOLOGY I	FULLY-MESHED	5	20
TOPOLOGY II	SEMI-MESHED	8	22

TABLE 6.1: Random Network Topologies

The network topology dimensioned in this evaluation are based on reference to IP-based mobile backbone networks. The Table 6.1 shows the details about the random network topologies considered for the evaluation. It is assumed that each node in the topology corresponds to a router. TOPOLOGY I network is a random network which is a fully-meshed network considered for evaluating the EE-TE model. It consists of 5 nodes and 20 links which is shown in Figure 6.1. TOPOLOGY II network is also a random network which is a semi-meshed network as shown in Figure 6.2. The idea of using two different kinds of random networks is to see how the model behaves based on the network.

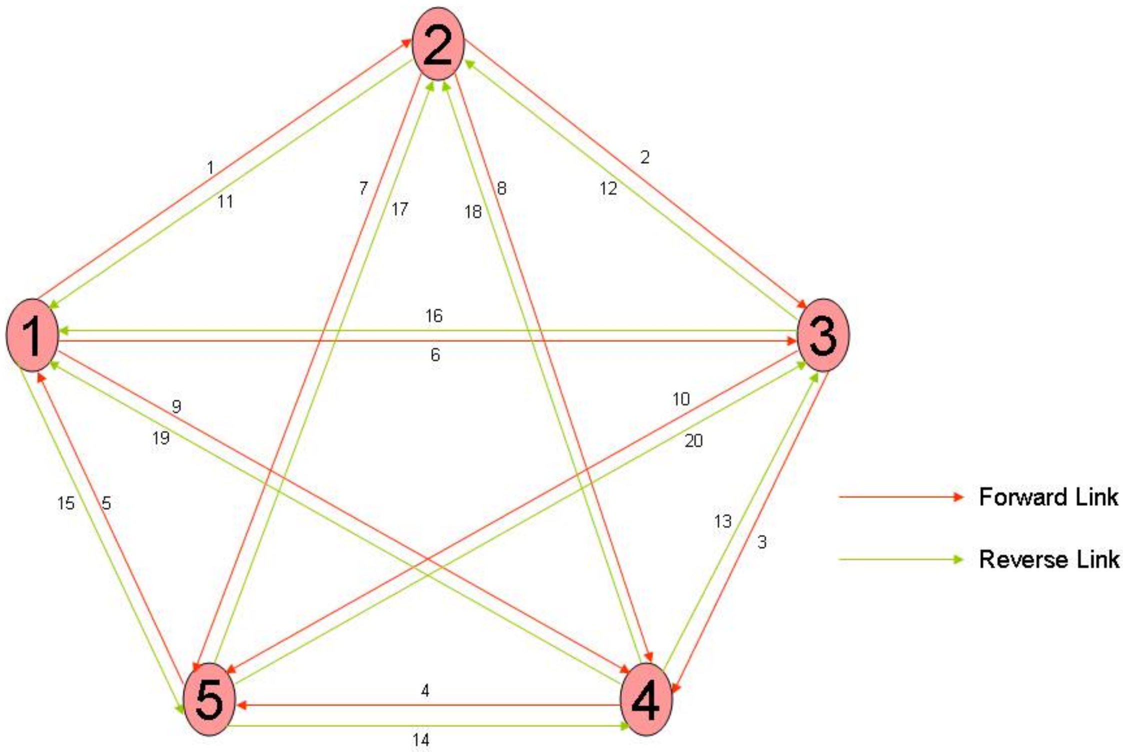

FIGURE 6.1: RANDOM NETWORK TOPOLOGY I - FULLY-MESHED IP MOBILE BACKBONE NETWORK

6.1.2 Traffic Demand

Traffic demand is also considered to be one of the input to the EE-TE model. Different traffic demand profiles are considered in this evaluation and assumed to carry different traffic patterns such as voice, video, http etc. and it is measured in bits/sec. The traffic demand profiles are generated randomly based on power saving scenarios for evaluating different constraints used in the EE-TE model. Since capacities are configured in this EE-TE model, it is assigned to links using the method described in [26] and based on that traffic demand profiles are generated randomly for evaluating different constraints

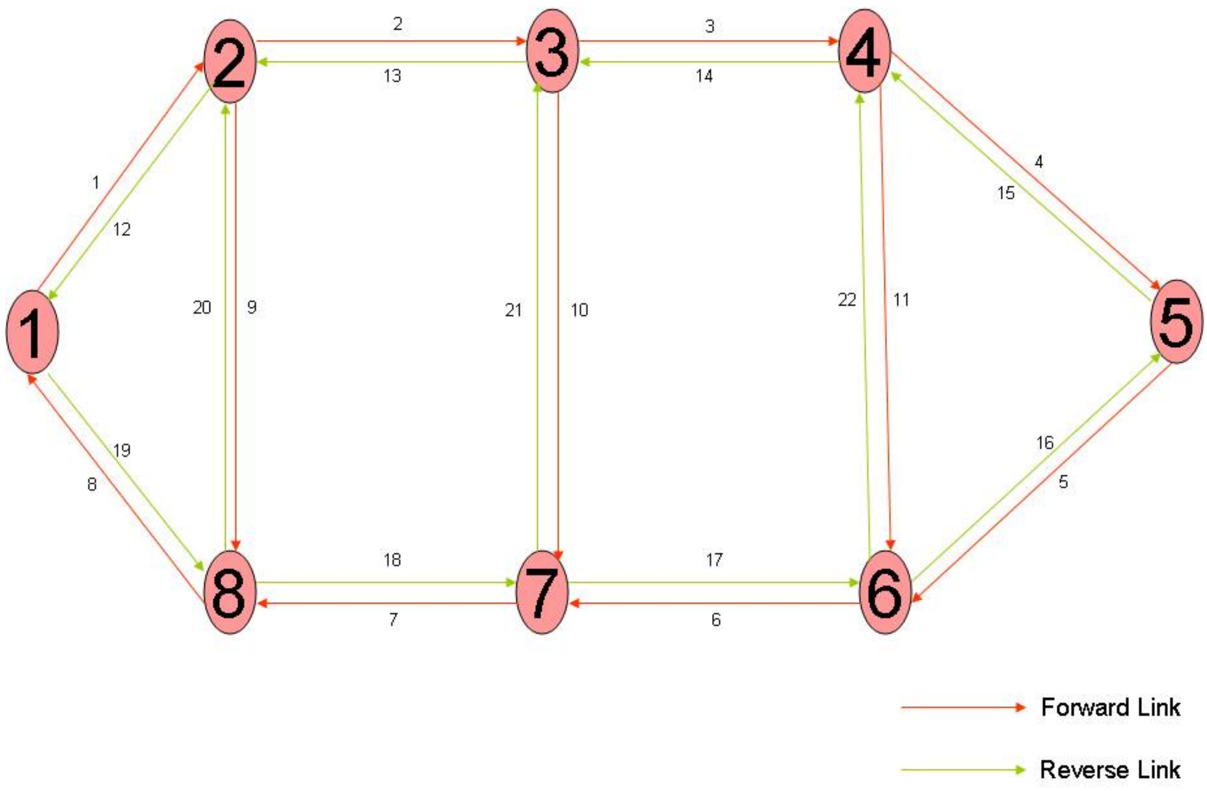

Figure 6.2: RANDOM NETWORK TOPOLOGY II - SEMI-MESHED IP MOBILE BACKBONE NETWORK

performance. Similarly, random traffic demand profiles are assumed in the configuration to evaluate the model.

6.1.3 EE-TE Network Configuration

In order to evaluate the EE-TE model, the input parameters defined have to be configured. As noted in subsection 4.1.1, the line-cards contribute with significant portion of a router's total power budget. The power consumption of line-cards used in the evaluation is specified in Table 6.2.

LINE-CARD	SPEED(Mbps)	Power(Watts)
10-port Gigabit Ethernet DDR(PPA2)	1000	96
1-port 10 Gigabit Ethernet(PPA2)	9953	131

Table 6.2: Power Consumption of Smart Edge 800 Router Line-Card's [40]

Based on this line-card configuration, the link capacities and link power for all the links in a network are configured. This assumption is made because the information of physical connections among line-cards is not available in the data set.

Once the capacity of links is defined, then different traffic demand profiles are configured to perform different constraints performance used in the model. Usually in a real network, the traffic varies according to the day-night pattern observed. Similarly, a mix of traffic volume is assumed in the evaluation for each OD pair based on the capacity configured in the model. The following traffic demand profiles shown in Figure 6.3 and Figure 6.4 are used in the evaluation which are configured according to the network topology configuration.

```
param TrafficDemand :=

[*,1] 2 240   3 150   4 390   5 120

[*,2] 1 132   3 210   4 109   5 310

[*,3] 1 130   2 1200 4 1020 5 102

[*,4] 1 1300 2 318   3 163   5 221

[*,5] 1 173   2 1400 3 1010 4 331

;
```

FIGURE 6.3: Random Traffic Demand - Profile I - TOPOLOGY I

```
param TrafficDemand :=
[*,1] 2 80 3 56 4 88 5 11 6 60 7 49 8 90
[*,2] 1 30 3 39 4 24 5 13 6 50 7 64 8 69
[*,3] 1 12 2 40 4 65 5 94 6 56 7 48 8 72
[*,4] 1 12 2 29 3 90 5 78 6 42 7 28 8 67
[*,5] 1 13 2 98 3 80 4 30 6 45 7 82 8 32
[*,6] 1 45 2 19 3 63 4 20 5 87 7 38 8 60
[*,7] 1 22 2 79 3 29 4 92 5 63 6 16 8 83
[*,8] 1 65 2 24 3 95 4 44 5 12 6 79 7 12
;
```

FIGURE 6.4: Random Traffic Demand - Profile II - TOPOLOGY II

Given the configuration such as network topology, traffic demand profiles, link capacities and link power, it is possible to pre-compute the parameter Paths using K-shortest path algorithm using Matlab [46] for each OD pair and solve the EE-TE model using CPLEX [42]. The K-shortest Path is computed by considering equal cost for each link in the network. From the solution of the model, obtain the total power saving for a network as well as the utilization for each link. In addition, the solution also gives which paths to use for each OD pair and how to split traffic among these paths based on the actual traffic demand considered in the traffic demand profiles.

In the evaluation, it is assumed that each line-card is connected to a single link; therefore a line-card can be put to sleep when there is no traffic on the link. This is achieved with the help of bi-directional Pair constraints used in the EE-TE model. This makes sure that the links are put to sleep in pairs.

6.2 Power Saving Scenarios

To explore the power saving potential under EE-TE, different constraints are evaluated for the proposed configurations. The power saving ratio is computed as the total power of sleeping line-cards over the total power of all line-cards in the network. Since line-card consumes more power in router's total power; therefore it is meaningful to measure the power saving ratio of line-cards. The following evaluations are performed for two random network topologies with different traffic demand profiles being considered.

6.2.1 Link Utilization

Intuitively, EE-TE would affect the utilization of links as fewer links are used to carry traffic. In this subsection, we evaluate the impact of EE-TE on link utilization and see how the power saving ratio performs. Specifically, it shows how the maximum link utilization of the network is performed for different configurations. The maximum link utilization ($\mathrm{U_T}$) is limited based on the parameter Threshold Level ($\mathrm{T_L}$) set in the problem. This is because the solver only focuses on putting the links to sleep as long as the maximum link utilization ($\mathrm{U_T}$) is greater than $\mathrm{T_L}$.

6.2.2 TOPOLOGY I and TRAFFIC DEMAND - PROFILE I

In this evaluation, the TOPOLOGY I and TRAFFIC DEMAND-PROFILE I configuration is chosen. The topology I is a random network which is fully-meshed and traffic demand profile I has the actual traffic demand for each OD pair which will be below 2000 Mbps. Figure 6.5 shows the power savings ratio obtained under different maximum link utilization ($\mathrm{U_T}$) set by ThresholdLevel ($\mathrm{T_L}$). When maximum link utilization ($\mathrm{U_T}$)

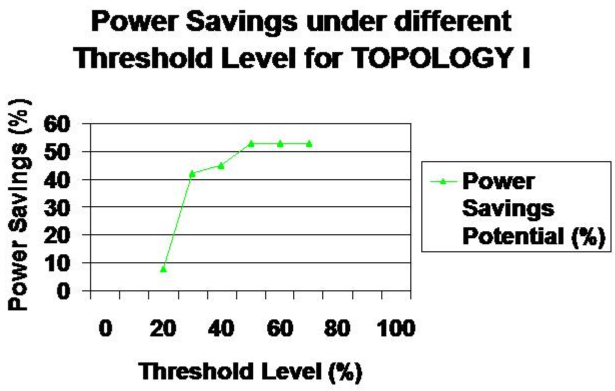

Figure 6.5: Power Savings Potential for TOPOLOGY I under different Threshold Level

is increased, the EE-TE model is able to achieve better power savings. The maximum power saving achieved is 53%, when ThresholdLevel (T_L) is set to 50% which is the maximum link utilization (U_T) maintained for this configuration. Table 6.3 shows the corresponding Total Power Saved and the number of sleeping links (X_L) obtained under different ThresholdLevel specified in this configuration. Another observation to note here is that, the number of sleeping links ($X_L = 1$) also increases when ThresholdLevel (T_L) increases.

U_T	Total Power Saved(Watts)	Power Savings(%)	No. of Sleeping Links($X_L = 1$)
$T_L = 20\%$	192	8%	2
$T_L = 30\%$	960	42%	10
$T_L = 40\%$	1030	45%	10
$T_L = 50\%$	1222	53%	12
$T_L = 60\%$	1222	53%	12
$T_L = 70\%$	1222	53%	12

Table 6.3: Maximum Link Utilization at different ThresholdLevel - TOPOLOGY I

6.2.3 TOPOLOGY II and TRAFFIC DEMAND - PROFILE II

In this evaluation, TOPOLOGY II and TRAFFIC DEMAND-PROFILE II is chosen. Topology II is a random network which is semi-meshed type and traffic demand profile II has the actual traffic demand for each OD pair set below 100 Mbps. Figure 6.6 shows the power savings ratio obtained under different maximum link utilization (U_T) set by ThresholdLevel (T_L). When maximum link utilization (U_T) is low (i.e. $T_L \leq 30\%$) there is no power saving achieved. This is because, the link utilization is not in the range set by the ThresholdLevel (T_L). In this case, it exceeds the ThresholdLevel (T_L)

and the power saving results in zero. However, as the ThresholdLevel (T_L) increases, there is a potential power saving. The maximum power saving achieved is 38%, when ThresholdLevel (T_L) is set to 80%, which will be the maximum link utilization (U_T) maintained in this configuration. However, when ThresholdLevel T_L = 80%, if sudden traffic bursts occurs, the network cannot handle since the number of sleeping links are more as ThresholdLevel increases.

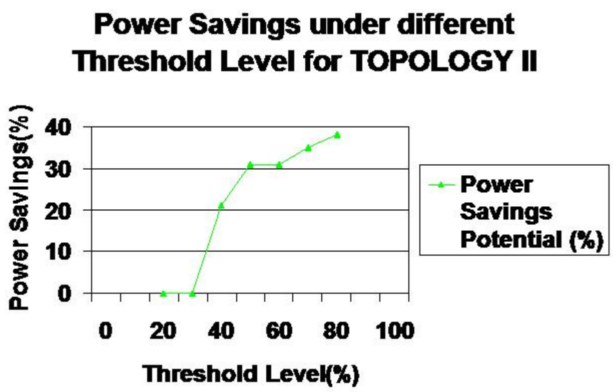

FIGURE 6.6: Power Savings Potential for TOPOLOGY II under different Threshold Level

The following general observations have to be noted for link utilization. First, the maximum link utilization (U_T) must not exceed the ThresholdLevel (T_L). Otherwise, it will result in infeasible solution as shown in the observations from Table 6.4. This purely depends on the actual traffic demand. Second, the ThresholdLevel when T_L = 50% is considered to be the better optimal settings, where it obtains an optimal power saving with reasonable amount of sleeping links (X_L). This is because, there is an issue related to network performance to observe if more number of links goes to sleep. Another issue is when a link fails or a burst of traffic arrives, the network needs to find alternative paths to accomodate the traffic if the traffic being affected is of very high volume. This problem exists in any IP network.

U_T	Total Power Saved(Watts)	Power Savings(%)	No. of Sleeping Links($X_L = 1$)
T_L = 20%	infeasible	-	-
T_L = 30%	infeasible	-	-
T_L = 40%	524	21%	4
T_L = 50%	786	31%	6
T_L = 60%	786	31%	6
T_L = 70%	908	35%	8
T_L = 80%	978	38%	8

TABLE 6.4: Maximum Link Utilization at different ThresholdLevel - TOPOLOGY II

6.2.4 PathNum (K)

PathNum (K) is a parameter used to set the maximum number of shortest paths for each OD pair in the given network. This parameter PathNum (K) plays an important role depending on the value K set in the model. Using this parameter, it is possible to include more energy-efficient paths for OD pairs and helps in optimizing the power savings of a network. Here also, the network performance of the two random network topologies is investigated based on the value K set for different configurations with ThresholdLevel ($T_L = 50\%$)

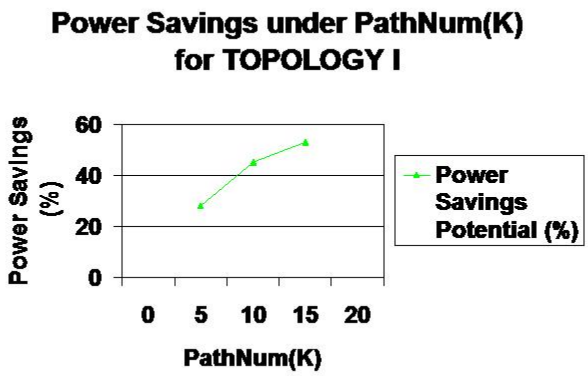

FIGURE 6.7: Power Savings Potential for TOPOLOGY I under PathNum(K)

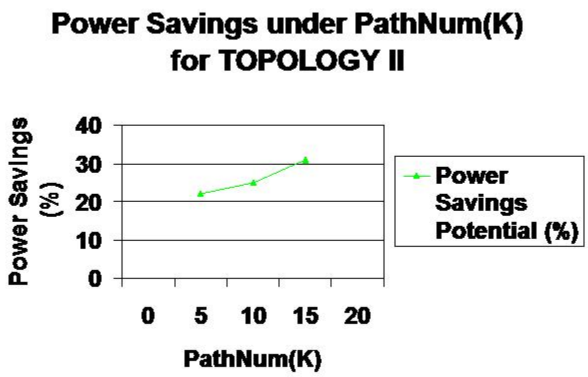

FIGURE 6.8: Power Savings Potential for TOPOLOGY II under PathNum(K)

In this evaluation, the TOPOLOGY I and TOPOLOGY II random network topologies will be used. Figure 6.7 shows that the power saving potential grows as the value of PathNum (K) increases. However, increasing K also increases the computation time shown in Table 6.5. Increasing PathNum (K) only improves the power saving potential by a negligible amount. Therefore, EE-TE is able to achieve near optimal power savings as long as K is reasonably large. Similarly, Figure 6.8 and Table 6.6 shows the corresponding observations for TOPOLOGY II network topology.

PathNum(K)	**Computation Time**	**Power Savings(%)**
5	0.06 sec	28%
10	0.09 sec	45%
15	0.16 sec	53%

Table 6.5: Maximum No. of Shortest Paths - TOPOLOGY I

PathNum(K)	**Computation Time**	**Power Savings(%)**
5	0.09 sec	22%
10	0.20 sec	25%
15	0.23 sec	31%

Table 6.6: Maximum No. of Shortest Paths - TOPOLOGY II

6.2.5 Optimized Routing Path

In EE-TE model, the optimized routing path is obtained using TrafficFlowPerLink ($F_{\mathrm{L}}{}^{s,t}$) and TrafficSplitRatio ($a_{\mathrm{i}}{}^{\mathrm{s,t}}$) depending on the actual traffic demand. In this evaluation, there are two observations noticed in order to obtain optimized routing path such as optimized single routing path for each OD pair and also optimized multi routing path for each OD pair based on the actual traffic demand.

Consider TOPOLOGY II and TRAFFIC DEMAND PROFILE II configuration. The following variables will be computed such as,

- TrafficFlowPerLink($F_{\mathrm{L}}{}^{s,t}$)

- TrafficSplitRatio($a_{\mathrm{i}}{}^{\mathrm{s,t}}$)

TrafficFlowPerLink ($F_{\mathrm{L}}{}^{s,t}$) is one of the output variables in the EE-TE model, which holds the optimized efficient routing path for the given network. For each OD pair a routing path is obtained based on the links and its corresponding traffic demand is distributed over links. In this case, the TrafficSplitRatio ($\alpha_{\mathrm{i}}{}^{\mathrm{s,t}}$) is set to value 1 for all routing path since the actual traffic demand measured is below 100 Mbps, which is considered to be very low traffic for the EE-TE network configuration mentioned. So thats the reason, there is no need for traffic split ratio being performed where it is applicable for the scenario, when there is a multiple path is obtained.

Consider TOPOLOGY I and TRAFFIC DEMAND - PROFILE I configuration to see the evaluation for traffic split ratio performance in EE-TE model. In this scenario, the actual traffic demand measured is high for the EE-TE network configuration. The following OD pair (1,4), (2,3), (2,5), (4,3) and (3,5) have performed traffic split due to the traffic volume measured is of high. Take a closer look at the OD pair (1,4) in Figure 6.9 from the results to see how traffic split ratio is performed.

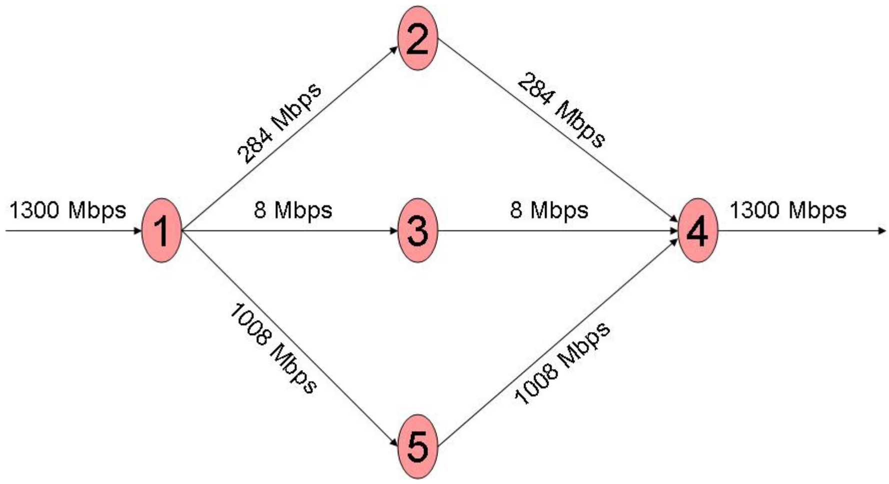

FIGURE 6.9: Traffic Split Ratio for Multi-Path for node (1,4)

As illustrated in Figure 6.9, multi-path is chosen as the optimized routing path. The total traffic volume measured for OD pair (1,4) is 1300 Mbps. A multi-path is chosen based on the link utilization and also due to the single routing path is put to sleep in this network. The three Traffic Split ratio $\alpha(2,1,4)$, $\alpha(3,1,4)$ and $\alpha(4,1,4)$ is chosen and it is split accordingly based on the link utilization such that the summation of traffic split ratio is equal to 1. An example of traffic split ratio implementation of EE-TE model in LP is shown in Appendix B by taking into consideration node (1,4). Similarly, the traffic split ratio is performed for other routing paths as well in this network configuration.

The results of both optimized Single and Multi routing path is shown in Appendix C.

Chapter 7

Conclusion and Future Work

High path redundancy and low link utilization in today's IP mobile backbone networks provide unique opportunities for energy based traffic engineering. By switching traffic onto fewer number of paths, one can free some links from carrying data traffic and put them to sleep for energy conservation. The EE-TE model maximizes the number of links that can be put to sleep under the constraints of link utilization and traffic split ratio mechanism. The approach of EE-TE model has been implemented based on MCF formulation with the help of CPLEX and justified that it is able to achieve near optimal energy savings for IP-based mobile backbone networks. Evaluations based on random network topologies and traffic demand profiles show that EE-TE is able to achieve considerable power savings with minor impacts on the network performance. From the configurations used in the evaluation, it justifies that although a fully-meshed network considerably saves more energy in IP-backbone networks, but the approach is not feasible for real networks due to overall cost of this network will be too high as compared to other topologies. The evaluation also shows that the approach achieves near optimal savings in a semi-meshed network. Also the route selection by EE-TE is relatively stable with the help of traffic split ratio instead of changing link weights so that negative impacts caused by routing adjustments can be limited.

Future work focuses on further investigation and implementation of the model. The investigation includes further optimization on power savings and energy-efficient topologies when introducing multiple ports per line-cards or possibly turning off full nodes, handling link failures and sudden traffic bursts. This investigation could enhance the EE-TE model by taking advantages of recent advances in hardware energy management mechanisms. The next level of implementation includes implementing the EE-TE model in the network simulator and testbed with the CCU framework.

Appendix A

EE-TE Formulation Using OPL

The formulation of EE-TE mathematical model using OPL discussed in chapter 5 is shown here. The following configurations are considered to formulate this model.

S.No	CATEGORY	CONFIGURATION
1	Network Topology	TOPOLOGY I
2	Traffic Demand	TRAFFIC DEMAND - PROFILE I
3	Router	Smart Edge(SE) 800

TABLE A.1: EE-TE Implementation configuration

LINE-CARD	SPEED(Mbps)	Power(Watts)
10-port Gigabit Ethernet DDR(PPA2)	1000	96
1-port 10 Gigabit Ethernet(PPA2)	9953	131

TABLE A.2: Power Consumption of Smart Edge 800 Router Line-Card's [40]

PARAMETERS SECTION

param NodeNum;

param LinkNum;

param PathNum;

param ThresholdLevel default 0.5;

param Capacity {L in 1..LinkNum} default 0;

param Demand {s in 1..NodeNum, t in 1..NodeNum : s <> t} default 0;

param Paths {i in 1..PathNum, s in 1..NodeNum, t in 1..NodeNum , L in 1..LinkNum } default 0, binary;

param Pair {L in 1..LinkNum } default 0, $\leq$ LinkNum;

/**** **Smart Edge 800 Router configuration** ****/

param LinkPower {L in 1..LinkNum} default (
if Capacity[l] < 2000 then 96 */* 10-port Gigabit Ethernet DDR (PPA2) */*
else if Capacity[l] < 10000 then 131 */* 1-port 10 Gigabit Ethernet (PPA2) */*
else 348);

VARIABLES SECTION

var f {s in 1..NodeNum , t in 1..NodeNum , l in 1..LinkNum : s <> t} >= 0;
var a {i in 1..PathNum , s in 1..NodeNum , t in 1..NodeNum : s <> t} >= 0;
var u {L in 1..LinkNum } >= 0, <= ThresholdLevel;
var x {L in 1..LinkNum }, binary;
var Saving >= 0;
var MaxUtil >= 0;

OBJECTIVE FUNCTION

Maximize TotalPowerSaved: sum {L in 1..LinkNum } x[L] * LinkPower[L];

CONSTRAINTS SECTION

subject to **TrafficFlowPerLink** {s in 1..NodeNum , t in 1..NodeNum , L in 1..LinkNum : s <> t}:
f[s, t, L] = sum {i in 1..PathNum } Paths[i, s, t, L] * Demand[s, t] * a[i, s, t];

subject to **NullPaths** {i in 1..PathNum , s in 1..NodeNum , t in 1..NodeNum : s <> t
sum {L in 1..LinkNum } Paths[i, s, t, L] = 0}:
a[i, s, t] = 0;

subject to **Ratios** {s in 1..NodeNum , t in 1..NodeNum : s <> t}:
sum {i in 1..PathNum } a[i, s, t] = 1;

subject to **Utilization** {L in 1..LinkNum }:

u[L] = (sum {s in 1..NodeNum , t in 1..NodeNum : s <> t} f[s, t, L]) / Capacity[L];

subject to **TurnOff** {L in 1..LinkNum}:

x[L] + u[L] <= 1;

subject to **Bi-Direction** {L in 1..LinkNum}:

x[L] = x[Pair[L]];

subject to **Saving**:

PowerSaving = (sum {L in 1..LinkNum} x[L] * LinkPower[L]) / (sum {L in 1..LinkNum} LinkPower[L]);

subject to **MaxUtilization** {L in 1..LinkNum}:

u[L] <= MaxUtil;

OPL DATA SECTION

data;

param NodeNum := 5;

param LinkNum := 20;

param PathNum := 16;

param ThresholdLevel := 0.5;

param Capacity :=

[1] 9953

[2] 1000

[3] 1000

[4] 9953

[5] 9953

[6] 9953

[7] 1000

[8] 9953

[9] 1000

[10] 1000

[11] 9953

[12] 1000

[13] 1000

[14] 9953

[15] 9953

[16] 9953

[17] 1000

[18] 9953

[19] 1000

[20] 1000

;

param Pair :=

[1] 11

[2] 12

[3] 13

[4] 14

[5] 15

[6] 16

[7] 17

[8] 18

[9] 19

[10] 20

[11] 1

[12] 2

[13] 3

[14] 4

[15] 5

[16] 6

[17] 7

[18] 8

[19] 9

[20] 10

;

param Demand :=

[*,1] 2 240 3 150 4 390 5 120

[*,2] 1 132 3 210 4 109 5 310

[*,3] 1 130 2 1200 4 1020 5 102

[*,4] 1 1300 2 318 3 163 5 221

[*,5] 1 173 2 1400 3 1010 4 331

;

*/**** PathNum (K) = 5 ****/*

param Paths :=

[1,1,2,*] 1 1

[1,1,3,*] 6 1

[1,1,4,*] 9 1

[1,1,5,*] 15 1

[1,2,1,*] 11 1

[1,2,3,*] 2 1

[1,2,4,*] 8 1

[1,2,5,*] 7 1

[1,3,1,*] 16 1

[1,3,2,*] 12 1

[1,3,4,*] 3 1

[1,3,5,*] 10 1

[1,4,1,*] 19 1

[1,4,2,*] 18 1

[1,4,3,*] 13 1

[1,4,5,*] 4 1

[1,5,1,*] 5 1

[1,5,2,*] 17 1

[1,5,3,*] 20 1

[1,5,4,*] 14 1

[2,1,2,*] 9 1 18 1

[2,1,3,*] 9 1 13 1

[2,1,4,*] 6 1 3 1

[2,1,5,*] 9 1 4 1

[2,2,1,*] 2 1 16 1

[2,2,3,*] 7 1 20 1

[2,2,4,*] 2 1 3 1

[2,2,5,*] 2 1 10 1

[2,3,1,*] 3 1 19 1

[2,3,2,*] 10 1 17 1

[2,3,4,*] 12 1 8 1

[2,3,5,*] 12 1 7 1

[2,4,1,*] 13 1 16 1

[2,4,2,*] 13 1 12 1

[2,4,3,*] 19 1 6 1

[2,4,5,*] 13 1 10 1

[2,5,1,*] 17 1 11 1

[2,5,2,*] 20 1 12 1

[2,5,3,*] 17 1 2 1

[2,5,4,*] 20 1 3 1

[3,1,2,*] 6 1 12 1

[3,1,3,*] 1 1 2 1

[3,1,4,*] 1 1 8 1

[3,1,5,*] 1 1 7 1

[3,2,1,*] 7 1 5 1

[3,2,3,*] 8 1 13 1

[3,2,4,*] 7 1 14 1

[3,2,5,*] 11 1 15 1

[3,3,1,*] 12 1 11 1

[3,3,2,*] 3 1 18 1

[3,3,4,*] 10 1 14 1

[3,3,5,*] 3 1 4 1

[3,4,1,*] 18 1 11 1

[3,4,2,*] 19 1 1 1

[3,4,3,*] 18 1 2 1

[3,4,5,*] 19 1 15 1

[3,5,1,*] 20 1 16 1

[3,5,2,*] 5 1 1 1

[3,5,3,*] 14 1 13 1

[3,5,4,*] 17 1 8 1

[4,1,2,*] 15 1 17 1

[4,1,3,*] 15 1 20 1

[4,1,4,*] 15 1 14 1

[4,1,5,*] 6 1 10 1

[4,2,1,*] 8 1 19 1

[4,2,3,*] 11 1 6 1

[4,2,4,*] 11 1 9 1

```
[4,2,5,*] 8 1 4 1
[4,3,1,*] 10 1 5 1
[4,3,2,*] 16 1 1 1
[4,3,4,*] 16 1 9 1
[4,3,5,*] 16 1 15 1
[4,4,1,*] 4 1 5 1
[4,4,2,*] 4 1 17 1
[4,4,3,*] 4 1 20 1
[4,4,5,*] 18 1 7 1
[4,5,1,*] 14 1 19 1
[4,5,2,*] 14 1 18 1
[4,5,3,*] 5 1 6 1
[4,5,4,*] 5 1 9 1
[5,1,2,*] 9 1 13 1 12 1
[5,1,3,*] 9 1 18 1 2 1
[5,1,4,*] 1 1 2 1 3 1
[5,1,5,*] 9 1 13 1 10 1
[5,2,1,*] 2 1 3 1 19 1
[5,2,3,*] 7 1 14 1 13 1
[5,2,4,*] 7 1 20 1 3 1
[5,2,5,*] 2 1 3 1 4 1
[5,3,1,*] 12 1 7 1 5 1
[5,3,2,*] 3 1 19 1 1 1
[5,3,4,*] 12 1 7 1 14 1
[5,3,5,*] 3 1 19 1 15 1
[5,4,1,*] 13 1 12 1 11 1
[5,4,2,*] 13 1 10 1 17 1
[5,4,3,*] 19 1 1 1 2 1
[5,4,5,*] 13 1 12 1 7 1
[5,5,1,*] 20 1 3 1 19 1
[5,5,2,*] 20 1 3 1 18 1
[5,5,3,*] 17 1 8 1 13 1
[5,5,4,*] 17 1 2 1 3 1
;

end;
```

Appendix B

EE-TE CPLEX LP Implementation

The CPLEX LP implementation discussed in chapter 5 is shown here. The EE-TE model is formulated in Mixed Integer Programming(MIP) format. This model is implemented based on the formulations shown in Appendix A and computed using IBM ILOG CPLEX INTERACTIVE OPTIMIZER. The coding shown below gives an overview of CPLEX LP implementation.

/**** **EE-TE CPLEX LP Problem** ****/

OBJECTIVE FUNCTION DEFINITION

Maximize

TotalPowerSaved: + 131 x(1) + 96 x(2) + 96 x(3) + 131 x(4) + 131 x(5) + 131 x(6) + 96 x(7) + 131 x(8) + 96 x(9) + 96 x(10) + 131 x(11) + 96 x(12) + 96 x(13) + 131 x(14) + 131 x(15) + 131 x(16) + 96 x(17) + 131 x(18) + 96 x(19) + 96 x(20)

CONSTRAINTS SECTION

*/**** The coding is shown for OD pair (1,4) for all links based on formulations shown in Appendix A ****/*

Subject To

TrafficFlowPerLink(1,4,1): + f(1,4,1) - 1300 a(3,1,4) - 1300 a(5,1,4) - 1300 a(8,1,4) - 1300 a(11,1,4) - 1300 a(13,1,4) = -0
TrafficFlowPerLink(1,4,2): + f(1,4,2) - 1300 a(5,1,4) - 1300 a(12,1,4) - 1300 a(13,1,4) = -0
TrafficFlowPerLink(1,4,3): + f(1,4,3) - 1300 a(2,1,4) - 1300 a(5,1,4) - 1300 a(6,1,4) - 1300 a(11,1,4) - 1300 a(12,1,4) = -0
TrafficFlowPerLink(1,4,4): + f(1,4,4) = -0
TrafficFlowPerLink(1,4,5): + f(1,4,5) = -0
TrafficFlowPerLink(1,4,6): + f(1,4,6) - 1300 a(2,1,4) - 1300 a(7,1,4) - 1300 a(10,1,4) - 1300 a(15,1,4) - 1300 a(16,1,4) = -0
TrafficFlowPerLink(1,4,7): + f(1,4,7) - 1300 a(8,1,4) - 1300 a(11,1,4) - 1300 a(15,1,4) = -0
TrafficFlowPerLink(1,4,8): + f(1,4,8) - 1300 a(3,1,4) - 1300 a(7,1,4) - 1300 a(9,1,4) - 1300 a(14,1,4) - 1300 a(16,1,4) = -0
TrafficFlowPerLink(1,4,9): + f(1,4,9) - 1300 a(1,1,4) = -0
TrafficFlowPerLink(1,4,10): + f(1,4,10) - 1300 a(10,1,4) - 1300 a(13,1,4) - 1300 a(16,1,4) = -0
TrafficFlowPerLink(1,4,11): + f(1,4,11) = -0
TrafficFlowPerLink(1,4,12): + f(1,4,12) - 1300 a(7,1,4) - 1300 a(14,1,4) - 1300 a(15,1,4) = -0
TrafficFlowPerLink(1,4,13): + f(1,4,13) = -0
TrafficFlowPerLink(1,4,14): + f(1,4,14) - 1300 a(4,1,4) - 1300 a(8,1,4) - 1300 a(10,1,4) - 1300 a(13,1,4) - 1300 a(15,1,4) = -0
TrafficFlowPerLink(1,4,15): + f(1,4,15) - 1300 a(4,1,4) - 1300 a(6,1,4) - 1300 a(9,1,4) - 1300 a(12,1,4) - 1300 a(14,1,4) = -0
TrafficFlowPerLink(1,4,16): + f(1,4,16) = -0
TrafficFlowPerLink(1,4,17): + f(1,4,17) - 1300 a(9,1,4) - 1300 a(12,1,4) - 1300 a(16,1,4) = -0
TrafficFlowPerLink(1,4,18): + f(1,4,18) = -0
TrafficFlowPerLink(1,4,19): + f(1,4,19) = -0
TrafficFlowPerLink(1,4,20): + f(1,4,20) - 1300 a(6,1,4) - 1300 a(11,1,4) - 1300 a(14,1,4) = -0

/**** **Subject to Traffic Split Ratios** ****/

/ The traffic is splitted to 16 ratios, since the PathNum (K) is 16 */*

Ratios(1,4): + a(1,1,4) + a(2,1,4) + a(3,1,4) + a(4,1,4) + a(5,1,4) + a(6,1,4) + a(7,1,4) + a(8,1,4) + a(9,1,4) + a(10,1,4) + a(11,1,4) + a(12,1,4) + a(13,1,4) + a(14,1,4) + a(15,1,4) + a(16,1,4) = 1

*/**** **Subject to Link Utilization** ****/*

Utilization(1): - 0.000100472219431327 f(1,2,1)
- 0.000100472219431327 f(1,3,1) - 0.000100472219431327 f(1,4,1)
- 0.000100472219431327 f(1,5,1) - 0.000100472219431327 f(2,1,1)
- 0.000100472219431327 f(2,3,1) - 0.000100472219431327 f(2,4,1)
- 0.000100472219431327 f(2,5,1) - 0.000100472219431327 f(3,1,1)
- 0.000100472219431327 f(3,2,1) - 0.000100472219431327 f(3,4,1)
- 0.000100472219431327 f(3,5,1) - 0.000100472219431327 f(4,1,1)
- 0.000100472219431327 f(4,2,1) - 0.000100472219431327 f(4,3,1)
- 0.000100472219431327 f(4,5,1) - 0.000100472219431327 f(5,1,1)
- 0.000100472219431327 f(5,2,1) - 0.000100472219431327 f(5,3,1)
- 0.000100472219431327 f(5,4,1) + u(1) = -0
Utilization(2): - 0.001 f(1,2,2) - 0.001 f(1,3,2) - 0.001 f(1,4,2)
- 0.001 f(1,5,2) - 0.001 f(2,1,2) - 0.001 f(2,3,2) - 0.001 f(2,4,2)
- 0.001 f(2,5,2) - 0.001 f(3,1,2) - 0.001 f(3,2,2) - 0.001 f(3,4,2)
- 0.001 f(3,5,2) - 0.001 f(4,1,2) - 0.001 f(4,2,2) - 0.001 f(4,3,2)
- 0.001 f(4,5,2) - 0.001 f(5,1,2) - 0.001 f(5,2,2) - 0.001 f(5,3,2)
- 0.001 f(5,4,2) + u(2) = -0
Utilization(3): - 0.001 f(1,2,3) - 0.001 f(1,3,3) - 0.001 f(1,4,3)
- 0.001 f(1,5,3) - 0.001 f(2,1,3) - 0.001 f(2,3,3) - 0.001 f(2,4,3)
- 0.001 f(2,5,3) - 0.001 f(3,1,3) - 0.001 f(3,2,3) - 0.001 f(3,4,3)
- 0.001 f(3,5,3) - 0.001 f(4,1,3) - 0.001 f(4,2,3) - 0.001 f(4,3,3)
- 0.001 f(4,5,3) - 0.001 f(5,1,3) - 0.001 f(5,2,3) - 0.001 f(5,3,3)
- 0.001 f(5,4,3) + u(3) = -0
Utilization(4): - 0.000100472219431327 f(1,2,4)
- 0.000100472219431327 f(1,3,4) - 0.000100472219431327 f(1,4,4)
- 0.000100472219431327 f(1,5,4) - 0.000100472219431327 f(2,1,4)
- 0.000100472219431327 f(2,3,4) - 0.000100472219431327 f(2,4,4)
- 0.000100472219431327 f(2,5,4) - 0.000100472219431327 f(3,1,4)

- 0.000100472219431327 f(3,2,4) - 0.000100472219431327 f(3,4,4)
- 0.000100472219431327 f(3,5,4) - 0.000100472219431327 f(4,1,4)
- 0.000100472219431327 f(4,2,4) - 0.000100472219431327 f(4,3,4)
- 0.000100472219431327 f(4,5,4) - 0.000100472219431327 f(5,1,4)
- 0.000100472219431327 f(5,2,4) - 0.000100472219431327 f(5,3,4)
- 0.000100472219431327 f(5,4,4) + u(4) = -0
Utilization(5): - 0.000100472219431327 f(1,2,5)
- 0.000100472219431327 f(1,3,5) - 0.000100472219431327 f(1,4,5)
- 0.000100472219431327 f(1,5,5) - 0.000100472219431327 f(2,1,5)
- 0.000100472219431327 f(2,3,5) - 0.000100472219431327 f(2,4,5)
- 0.000100472219431327 f(2,5,5) - 0.000100472219431327 f(3,1,5)
- 0.000100472219431327 f(3,2,5) - 0.000100472219431327 f(3,4,5)
- 0.000100472219431327 f(3,5,5) - 0.000100472219431327 f(4,1,5)
- 0.000100472219431327 f(4,2,5) - 0.000100472219431327 f(4,3,5)
- 0.000100472219431327 f(4,5,5) - 0.000100472219431327 f(5,1,5)
- 0.000100472219431327 f(5,2,5) - 0.000100472219431327 f(5,3,5)
- 0.000100472219431327 f(5,4,5) + u(5) = -0
Utilization(6): - 0.000100472219431327 f(1,2,6)
- 0.000100472219431327 f(1,3,6) - 0.000100472219431327 f(1,4,6)
- 0.000100472219431327 f(1,5,6) - 0.000100472219431327 f(2,1,6)
- 0.000100472219431327 f(2,3,6) - 0.000100472219431327 f(2,4,6)
- 0.000100472219431327 f(2,5,6) - 0.000100472219431327 f(3,1,6)
- 0.000100472219431327 f(3,2,6) - 0.000100472219431327 f(3,4,6)
- 0.000100472219431327 f(3,5,6) - 0.000100472219431327 f(4,1,6)
- 0.000100472219431327 f(4,2,6) - 0.000100472219431327 f(4,3,6)
- 0.000100472219431327 f(4,5,6) - 0.000100472219431327 f(5,1,6)
- 0.000100472219431327 f(5,2,6) - 0.000100472219431327 f(5,3,6)
- 0.000100472219431327 f(5,4,6) + u(6) = -0
Utilization(7): - 0.001 f(1,2,7) - 0.001 f(1,3,7) - 0.001 f(1,4,7)
- 0.001 f(1,5,7) - 0.001 f(2,1,7) - 0.001 f(2,3,7) - 0.001 f(2,4,7)
- 0.001 f(2,5,7) - 0.001 f(3,1,7) - 0.001 f(3,2,7) - 0.001 f(3,4,7)
- 0.001 f(3,5,7) - 0.001 f(4,1,7) - 0.001 f(4,2,7) - 0.001 f(4,3,7)
- 0.001 f(4,5,7) - 0.001 f(5,1,7) - 0.001 f(5,2,7) - 0.001 f(5,3,7)
- 0.001 f(5,4,7) + u(7) = -0
Utilization(8): - 0.000100472219431327 f(1,2,8)
- 0.000100472219431327 f(1,3,8) - 0.000100472219431327 f(1,4,8)
- 0.000100472219431327 f(1,5,8) - 0.000100472219431327 f(2,1,8)
- 0.000100472219431327 f(2,3,8) - 0.000100472219431327 f(2,4,8)
- 0.000100472219431327 f(2,5,8) - 0.000100472219431327 f(3,1,8)

- 0.000100472219431327 f(3,2,8) - 0.000100472219431327 f(3,4,8)
- 0.000100472219431327 f(3,5,8) - 0.000100472219431327 f(4,1,8)
- 0.000100472219431327 f(4,2,8) - 0.000100472219431327 f(4,3,8)
- 0.000100472219431327 f(4,5,8) - 0.000100472219431327 f(5,1,8)
- 0.000100472219431327 f(5,2,8) - 0.000100472219431327 f(5,3,8)
- 0.000100472219431327 f(5,4,8) + u(8) = -0

Utilization(9): - 0.001 f(1,2,9) - 0.001 f(1,3,9) - 0.001 f(1,4,9)
- 0.001 f(1,5,9) - 0.001 f(2,1,9) - 0.001 f(2,3,9) - 0.001 f(2,4,9)
- 0.001 f(2,5,9) - 0.001 f(3,1,9) - 0.001 f(3,2,9) - 0.001 f(3,4,9)
- 0.001 f(3,5,9) - 0.001 f(4,1,9) - 0.001 f(4,2,9) - 0.001 f(4,3,9)
- 0.001 f(4,5,9) - 0.001 f(5,1,9) - 0.001 f(5,2,9) - 0.001 f(5,3,9)
- 0.001 f(5,4,9) + u(9) = -0

Utilization(10): - 0.001 f(1,2,10) - 0.001 f(1,3,10) - 0.001 f(1,4,10)
- 0.001 f(1,5,10) - 0.001 f(2,1,10) - 0.001 f(2,3,10) - 0.001 f(2,4,10)
- 0.001 f(2,5,10) - 0.001 f(3,1,10) - 0.001 f(3,2,10) - 0.001 f(3,4,10)
- 0.001 f(3,5,10) - 0.001 f(4,1,10) - 0.001 f(4,2,10) - 0.001 f(4,3,10)
- 0.001 f(4,5,10) - 0.001 f(5,1,10) - 0.001 f(5,2,10) - 0.001 f(5,3,10)
- 0.001 f(5,4,10) + u(10) = -0

Utilization(11): - 0.000100472219431327 f(1,2,11)
- 0.000100472219431327 f(1,3,11) - 0.000100472219431327 f(1,4,11)
- 0.000100472219431327 f(1,5,11) - 0.000100472219431327 f(2,1,11)
- 0.000100472219431327 f(2,3,11) - 0.000100472219431327 f(2,4,11)
- 0.000100472219431327 f(2,5,11) - 0.000100472219431327 f(3,1,11)
- 0.000100472219431327 f(3,2,11) - 0.000100472219431327 f(3,4,11)
- 0.000100472219431327 f(3,5,11) - 0.000100472219431327 f(4,1,11)
- 0.000100472219431327 f(4,2,11) - 0.000100472219431327 f(4,3,11)
- 0.000100472219431327 f(4,5,11) - 0.000100472219431327 f(5,1,11)
- 0.000100472219431327 f(5,2,11) - 0.000100472219431327 f(5,3,11)
- 0.000100472219431327 f(5,4,11) + u(11) = -0

Utilization(12): - 0.001 f(1,2,12) - 0.001 f(1,3,12) - 0.001 f(1,4,12)
- 0.001 f(1,5,12) - 0.001 f(2,1,12) - 0.001 f(2,3,12) - 0.001 f(2,4,12)
- 0.001 f(2,5,12) - 0.001 f(3,1,12) - 0.001 f(3,2,12) - 0.001 f(3,4,12)
- 0.001 f(3,5,12) - 0.001 f(4,1,12) - 0.001 f(4,2,12) - 0.001 f(4,3,12)
- 0.001 f(4,5,12) - 0.001 f(5,1,12) - 0.001 f(5,2,12) - 0.001 f(5,3,12)
- 0.001 f(5,4,12) + u(12) = -0

Utilization(13): - 0.001 f(1,2,13) - 0.001 f(1,3,13) - 0.001 f(1,4,13)
- 0.001 f(1,5,13) - 0.001 f(2,1,13) - 0.001 f(2,3,13) - 0.001 f(2,4,13)
- 0.001 f(2,5,13) - 0.001 f(3,1,13) - 0.001 f(3,2,13) - 0.001 f(3,4,13)
- 0.001 f(3,5,13) - 0.001 f(4,1,13) - 0.001 f(4,2,13) - 0.001 f(4,3,13)

- 0.001 f(4,5,13) - 0.001 f(5,1,13) - 0.001 f(5,2,13) - 0.001 f(5,3,13)
- 0.001 f(5,4,13) + u(13) = -0

Utilization(14): - 0.000100472219431327 f(1,2,14)
- 0.000100472219431327 f(1,3,14) - 0.000100472219431327 f(1,4,14)
- 0.000100472219431327 f(1,5,14) - 0.000100472219431327 f(2,1,14)
- 0.000100472219431327 f(2,3,14) - 0.000100472219431327 f(2,4,14)
- 0.000100472219431327 f(2,5,14) - 0.000100472219431327 f(3,1,14)
- 0.000100472219431327 f(3,2,14) - 0.000100472219431327 f(3,4,14)
- 0.000100472219431327 f(3,5,14) - 0.000100472219431327 f(4,1,14)
- 0.000100472219431327 f(4,2,14) - 0.000100472219431327 f(4,3,14)
- 0.000100472219431327 f(4,5,14) - 0.000100472219431327 f(5,1,14)
- 0.000100472219431327 f(5,2,14) - 0.000100472219431327 f(5,3,14)
- 0.000100472219431327 f(5,4,14) + u(14) = -0

Utilization(15): - 0.000100472219431327 f(1,2,15)
- 0.000100472219431327 f(1,3,15) - 0.000100472219431327 f(1,4,15)
- 0.000100472219431327 f(1,5,15) - 0.000100472219431327 f(2,1,15)
- 0.000100472219431327 f(2,3,15) - 0.000100472219431327 f(2,4,15)
- 0.000100472219431327 f(2,5,15) - 0.000100472219431327 f(3,1,15)
- 0.000100472219431327 f(3,2,15) - 0.000100472219431327 f(3,4,15)
- 0.000100472219431327 f(3,5,15) - 0.000100472219431327 f(4,1,15)
- 0.000100472219431327 f(4,2,15) - 0.000100472219431327 f(4,3,15)
- 0.000100472219431327 f(4,5,15) - 0.000100472219431327 f(5,1,15)
- 0.000100472219431327 f(5,2,15) - 0.000100472219431327 f(5,3,15)
- 0.000100472219431327 f(5,4,15) + u(15) = -0

Utilization(16): - 0.000100472219431327 f(1,2,16)
- 0.000100472219431327 f(1,3,16) - 0.000100472219431327 f(1,4,16)
- 0.000100472219431327 f(1,5,16) - 0.000100472219431327 f(2,1,16)
- 0.000100472219431327 f(2,3,16) - 0.000100472219431327 f(2,4,16)
- 0.000100472219431327 f(2,5,16) - 0.000100472219431327 f(3,1,16)
- 0.000100472219431327 f(3,2,16) - 0.000100472219431327 f(3,4,16)
- 0.000100472219431327 f(3,5,16) - 0.000100472219431327 f(4,1,16)
- 0.000100472219431327 f(4,2,16) - 0.000100472219431327 f(4,3,16)
- 0.000100472219431327 f(4,5,16) - 0.000100472219431327 f(5,1,16)
- 0.000100472219431327 f(5,2,16) - 0.000100472219431327 f(5,3,16)
- 0.000100472219431327 f(5,4,16) + u(16) = -0

Utilization(17): - 0.001 f(1,2,17) - 0.001 f(1,3,17) - 0.001 f(1,4,17)
- 0.001 f(1,5,17) - 0.001 f(2,1,17) - 0.001 f(2,3,17) - 0.001 f(2,4,17)
- 0.001 f(2,5,17) - 0.001 f(3,1,17) - 0.001 f(3,2,17) - 0.001 f(3,4,17)
- 0.001 f(3,5,17) - 0.001 f(4,1,17) - 0.001 f(4,2,17) - 0.001 f(4,3,17)

- 0.001 f(4,5,17) - 0.001 f(5,1,17) - 0.001 f(5,2,17) - 0.001 f(5,3,17)
- 0.001 f(5,4,17) + u(17) = -0
Utilization(18): - 0.000100472219431327 f(1,2,18)
- 0.000100472219431327 f(1,3,18) - 0.000100472219431327 f(1,4,18)
- 0.000100472219431327 f(1,5,18) - 0.000100472219431327 f(2,1,18)
- 0.000100472219431327 f(2,3,18) - 0.000100472219431327 f(2,4,18)
- 0.000100472219431327 f(2,5,18) - 0.000100472219431327 f(3,1,18)
- 0.000100472219431327 f(3,2,18) - 0.000100472219431327 f(3,4,18)
- 0.000100472219431327 f(3,5,18) - 0.000100472219431327 f(4,1,18)
- 0.000100472219431327 f(4,2,18) - 0.000100472219431327 f(4,3,18)
- 0.000100472219431327 f(4,5,18) - 0.000100472219431327 f(5,1,18)
- 0.000100472219431327 f(5,2,18) - 0.000100472219431327 f(5,3,18)
- 0.000100472219431327 f(5,4,18) + u(18) = -0
Utilization(19): - 0.001 f(1,2,19) - 0.001 f(1,3,19) - 0.001 f(1,4,19)
- 0.001 f(1,5,19) - 0.001 f(2,1,19) - 0.001 f(2,3,19) - 0.001 f(2,4,19)
- 0.001 f(2,5,19) - 0.001 f(3,1,19) - 0.001 f(3,2,19) - 0.001 f(3,4,19)
- 0.001 f(3,5,19) - 0.001 f(4,1,19) - 0.001 f(4,2,19) - 0.001 f(4,3,19)
- 0.001 f(4,5,19) - 0.001 f(5,1,19) - 0.001 f(5,2,19) - 0.001 f(5,3,19)
- 0.001 f(5,4,19) + u(19) = -0
Utilization(20): - 0.001 f(1,2,20) - 0.001 f(1,3,20) - 0.001 f(1,4,20)
- 0.001 f(1,5,20) - 0.001 f(2,1,20) - 0.001 f(2,3,20) - 0.001 f(2,4,20)
- 0.001 f(2,5,20) - 0.001 f(3,1,20) - 0.001 f(3,2,20) - 0.001 f(3,4,20)
- 0.001 f(3,5,20) - 0.001 f(4,1,20) - 0.001 f(4,2,20) - 0.001 f(4,3,20)
- 0.001 f(4,5,20) - 0.001 f(5,1,20) - 0.001 f(5,2,20) - 0.001 f(5,3,20)
- 0.001 f(5,4,20) + u(20) = -0

/**** ***Subject to Turnoff Capacity constraint*** ****/
TurnOff(1): + u(1) + x(1) <= 1
TurnOff(2): + u(2) + x(2) <= 1
TurnOff(3): + u(3) + x(3) <= 1
TurnOff(4): + u(4) + x(4) <= 1
TurnOff(5): + u(5) + x(5) <= 1
TurnOff(6): + u(6) + x(6) <= 1
TurnOff(7): + u(7) + x(7) <= 1
TurnOff(8): + u(8) + x(8) <= 1
TurnOff(9): + u(9) + x(9) <= 1
TurnOff(10): + u(10) + x(10) <= 1
TurnOff(11): + u(11) + x(11) <= 1

TurnOff(12): + u(12) + x(12) <= 1

TurnOff(13): + u(13) + x(13) <= 1

TurnOff(14): + u(14) + x(14) <= 1

TurnOff(15): + u(15) + x(15) <= 1

TurnOff(16): + u(16) + x(16) <= 1

TurnOff(17): + u(17) + x(17) <= 1

TurnOff(18): + u(18) + x(18) <= 1

TurnOff(19): + u(19) + x(19) <= 1

TurnOff(20): + u(20) + x(20) <= 1

/**** ***Subject to Bi-direction*** ****/

Bi-Direction(1): + x(1) - x(11) = -0

Bi-Direction(2): + x(2) - x(12) = -0

Bi-Direction(3): + x(3) - x(13) = -0

Bi-Direction(4): + x(4) - x(14) = -0

Bi-Direction(5): + x(5) - x(15) = -0

Bi-Direction(6): + x(6) - x(16) = -0

Bi-Direction(7): + x(7) - x(17) = -0

Bi-Direction(8): + x(8) - x(18) = -0

Bi-Direction(9): + x(9) - x(19) = -0

Bi-Direction(10): + x(10) - x(20) = -0

Bi-Direction(11): - x(1) + x(11) = -0

Bi-Direction(12): - x(2) + x(12) = -0

Bi-Direction(13): - x(3) + x(13) = -0

Bi-Direction(14): - x(4) + x(14) = -0

Bi-Direction(15): - x(5) + x(15) = -0

Bi-Direction(16): - x(6) + x(16) = -0

Bi-Direction(17): - x(7) + x(17) = -0

Bi-Direction(18): - x(8) + x(18) = -0

Bi-Direction(19): - x(9) + x(19) = -0

Bi-Direction(20): - x(10) + x(20) = -0

/**** ***Subject to Power Saving*** ****/

PowerSaving: - 0.0577092511013216 x(1) - 0.0422907488986784 x(2)

- 0.0422907488986784 x(3) - 0.0577092511013216 x(4)

- 0.0577092511013216 x(5) - 0.0577092511013216 x(6)

- 0.0422907488986784 x(7) - 0.0577092511013216 x(8)

- 0.0422907488986784 x(9) - 0.0422907488986784 x(10)
- 0.0577092511013216 x(11) - 0.0422907488986784 x(12)
- 0.0422907488986784 x(13) - 0.0577092511013216 x(14)
- 0.0577092511013216 x(15) - 0.0577092511013216 x(16)
- 0.0422907488986784 x(17) - 0.0577092511013216 x(18)
- 0.0422907488986784 x(19) - 0.0422907488986784 x(20) + Saving = -0

*/**** **Subject to Maximum Link Utilization** ****/*

MaxUtilization(1): + u(1) - MaxUtil <= -0
MaxUtilization(2): + u(2) - MaxUtil <= -0
MaxUtilization(3): + u(3) - MaxUtil <= -0
MaxUtilization(4): + u(4) - MaxUtil <= -0
MaxUtilization(5): + u(5) - MaxUtil <= -0
MaxUtilization(6): + u(6) - MaxUtil <= -0
MaxUtilization(7): + u(7) - MaxUtil <= -0
MaxUtilization(8): + u(8) - MaxUtil <= -0
MaxUtilization(9): + u(9) - MaxUtil <= -0
MaxUtilization(10): + u(10) - MaxUtil <= -0
MaxUtilization(11): + u(11) - MaxUtil <= -0
MaxUtilization(12): + u(12) - MaxUtil <= -0
MaxUtilization(13): + u(13) - MaxUtil <= -0
MaxUtilization(14): + u(14) - MaxUtil <= -0
MaxUtilization(15): + u(15) - MaxUtil <= -0
MaxUtilization(16): + u(16) - MaxUtil <= -0
MaxUtilization(17): + u(17) - MaxUtil <= -0
MaxUtilization(18): + u(18) - MaxUtil <= -0
MaxUtilization(19): + u(19) - MaxUtil <= -0
MaxUtilization(20): + u(20) - MaxUtil <= -0

Bounds

0 <= u(1) <= 0.5
0 <= u(2) <= 0.5
0 <= u(3) <= 0.5
0 <= u(4) <= 0.5
0 <= u(5) <= 0.5
0 <= u(6) <= 0.5
0 <= u(7) <= 0.5

0 <= u(8) <= 0.5

0 <= u(9) <= 0.5

0 <= u(10) <= 0.5

0 <= u(11) <= 0.5

0 <= u(12) <= 0.5

0 <= u(13) <= 0.5

0 <= u(14) <= 0.5

0 <= u(15) <= 0.5

0 <= u(16) <= 0.5

0 <= u(17) <= 0.5

0 <= u(18) <= 0.5

0 <= u(19) <= 0.5

0 <= u(20) <= 0.5

0 <= x(1) <= 1

0 <= x(2) <= 1

0 <= x(3) <= 1

0 <= x(4) <= 1

0 <= x(5) <= 1

0 <= x(6) <= 1

0 <= x(7) <= 1

0 <= x(8) <= 1

0 <= x(9) <= 1

0 <= x(10) <= 1

0 <= x(11) <= 1

0 <= x(12) <= 1

0 <= x(13) <= 1

0 <= x(14) <= 1

0 <= x(15) <= 1

0 <= x(16) <= 1

0 <= x(17) <= 1

0 <= x(18) <= 1

0 <= x(19) <= 1

0 <= x(20) <= 1

Binary

x(1)

x(2)

x(3)

x(4)
x(5)
x(6)
x(7)
x(8)
x(9)
x(10)
x(11)
x(12)
x(13)
x(14)
x(15)
x(16)
x(17)
x(18)
x(19)
x(20)

End

Appendix C

Optimized Routing Path Results

The results of Optimized Routing Path discussed in subsection 6.2.5 is shown here. Two types of routing path is evaluated such as,

- Optimized Single Routing Path
- Optimized Multi Routing Path

The following configurations are used for evaluation as shown in Table C.1 below,

Routing Path Type	**Network Topology**	**Traffic Demand - Profile**
Single path	TOPOLOGY II	TRAFFIC DEMAND - PROFILE II
Multi path	TOPOLOGY I	TRAFFIC DEMAND - PROFILE I

Table C.1: Optimized Routing Path Configuration

Optimized Single Routing Path

For the given configurations, the results of computed variables TrafficFlowPerLink ($F_L{}^{s,t}$) and TrafficSplitRatio ($\alpha_i{}^{s,t}$) is shown in Table C.2

Table C.2: Optimized Single Routing Path Solution

Variable Name	**Solution Value**
	Continued on next page

Table C.2 – continued from previous page

Variable Name	Solution Value
f(1,2,13)	30.000000
a(3,1,2)	1.000000
f(1,2,18)	30.000000
f(1,2,19)	30.000000
f(1,2,21)	30.000000
f(1,3,18)	12.000000
a(3,1,3)	1.000000
f(1,3,19)	12.000000
f(1,3,21)	12.000000
f(1,4,3)	12.000000
a(3,1,4)	1.000000
f(1,4,18)	12.000000
f(1,4,19)	12.000000
f(1,4,21)	12.000000
f(1,5,4)	13.000000
a(8,1,5)	1.000000
f(1,5,17)	13.000000
f(1,5,18)	13.000000
f(1,5,19)	13.000000
f(1,5,22)	13.000000
f(1,6,17)	45.000000
a(1,1,6)	1.000000
f(1,6,18)	45.000000
f(1,6,19)	45.000000
f(1,7,18)	22.000000
a(1,1,7)	1.000000
f(1,7,19)	22.000000
f(1,8,19)	65.000000

Continued on next page

Table C.2 – continued from previous page

Variable Name	Solution Value
a(1,1,8)	1.000000
f(2,1,2)	80.000000
a(3,2,1)	1.000000
f(2,1,7)	80.000000
f(2,1,8)	80.000000
f(2,1,10)	80.000000
f(2,3,2)	40.000000
a(1,2,3)	1.000000
f(2,4,2)	29.000000
a(1,2,4)	1.000000
f(2,4,3)	29.000000
f(2,5,2)	98.000000
a(1,2,5)	1.000000
f(2,5,3)	98.000000
f(2,5,4)	98.000000
f(2,6,2)	19.000000
a(1,2,6)	1.000000
f(2,6,3)	19.000000
f(2,6,11)	19.000000
f(2,7,2)	79.000000
a(1,2,7)	1.000000
f(2,7,10)	79.000000
f(2,8,2)	24.000000
a(3,2,8)	1.000000
f(2,8,7)	24.000000
f(2,8,10)	24.000000
f(3,1,7)	56.000000
a(2,3,1)	1.000000

Continued on next page

Table C.2 – continued from previous page

Variable Name	**Solution Value**
f(3,1,8)	56.000000
f(3,1,10)	56.000000
f(3,2,13)	39.000000
a(1,3,2)	1.000000
f(3,4,3)	90.000000
a(1,3,4)	1.000000
f(3,5,7)	80.000000
a(2,3,5)	1.000000
f(3,5,8)	80.000000
f(3,5,10)	80.000000
f(3,6,3)	63.000000
a(1,3,6)	1.000000
f(3,6,11)	63.000000
f(3,7,10)	29.000000
a(1,3,7)	1.000000
f(3,8,7)	95.000000
a(2,3,8)	1.000000
f(3,8,10)	95.000000
f(4,1,6)	88.000000
a(2,4,1)	1.000000
f(4,1,7)	88.000000
f(4,1,8)	88.000000
f(4,1,11)	88.000000
f(4,2,13)	24.000000
a(1,4,2)	1.000000
f(4,2,14)	24.000000
f(4,3,14)	65.000000
a(1,4,3)	1.000000

Continued on next page

Table C.2 – continued from previous page

Variable Name	Solution Value
f(4,5,4)	30.000000
a(1,4,5)	1.000000
f(4,6,11)	20.000000
a(1,4,6)	1.000000
f(4,7,6)	92.000000
a(2,4,7)	1.000000
f(4,7,11)	92.000000
f(4,8,6)	44.000000
a(2,4,8)	1.000000
f(4,8,7)	44.000000
f(4,8,11)	44.000000
f(5,1,6)	11.000000
a(3,5,1)	1.000000
f(5,1,7)	11.000000
f(5,1,8)	11.000000
f(5,1,11)	11.000000
f(5,1,15)	11.000000
f(5,2,13)	13.000000
a(1,5,2)	1.000000
f(5,2,14)	13.000000
f(5,2,15)	13.000000
f(5,3,14)	94.000000
a(1,5,3)	1.000000
f(5,3,15)	94.000000
f(5,4,15)	78.000000
a(1,5,4)	1.000000
f(5,6,11)	87.000000
a(2,5,6)	1.000000

Continued on next page

Table C.2 – continued from previous page

Variable Name	Solution Value
f(5,6,15)	87.000000
f(5,7,6)	63.000000
a(3,5,7)	1.000000
f(5,7,11)	63.000000
f(5,7,15)	63.000000
f(5,8,7)	12.000000
a(4,5,8)	1.000000
f(5,8,10)	12.000000
f(5,8,14)	12.000000
f(5,8,15)	12.000000
f(6,1,6)	60.000000
a(1,6,1)	1.000000
f(6,1,7)	60.000000
f(6,1,8)	60.000000
f(6,2,13)	50.000000
a(1,6,2)	1.000000
f(6,2,14)	50.000000
f(6,2,22)	50.000000
f(6,3,14)	56.000000
a(1,6,3)	1.000000
f(6,3,22)	56.000000
f(6,4,22)	42.000000
a(1,6,4)	1.000000
f(6,5,15)	45.000000
a(2,6,5)	1.000000
f(6,5,22)	45.000000
f(6,7,6)	16.000000
a(1,6,7)	1.000000

Continued on next page

Table C.2 – continued from previous page

Variable Name	Solution Value
f(6,8,6)	34.000000
a(1,6,8)	0.430380
f(6,8,7)	79.000000
a(4,6,8)	0.569620
f(6,8,10)	45.000000
f(6,8,14)	45.000000
f(6,8,22)	45.000000
f(7,1,7)	49.000000
a(1,7,1)	1.000000
f(7,1,8)	49.000000
f(7,2,13)	64.000000
a(1,7,2)	1.000000
f(7,2,21)	64.000000
f(7,3,21)	48.000000
a(1,7,3)	1.000000
f(7,4,3)	28.000000
a(1,7,4)	1.000000
f(7,4,21)	28.000000
f(7,5,4)	82.000000
a(3,7,5)	1.000000
f(7,5,17)	82.000000
f(7,5,22)	82.000000
f(7,6,17)	38.000000
a(1,7,6)	1.000000
f(7,8,7)	12.000000
a(1,7,8)	1.000000
f(8,1,8)	90.000000
a(1,8,1)	1.000000

Continued on next page

Table C.2 – concluded from previous page

Variable Name	Solution Value
f(8,2,13)	69.000000
a(3,8,2)	1.000000
f(8,2,18)	69.000000
f(8,2,21)	69.000000
f(8,3,18)	72.000000
a(2,8,3)	1.000000
f(8,3,21)	72.000000
f(8,4,17)	67.000000
a(3,8,4)	1.000000
f(8,4,18)	67.000000
f(8,4,22)	67.000000
f(8,5,4)	32.000000
a(4,8,5)	1.000000
f(8,5,17)	32.000000
f(8,5,18)	32.000000
f(8,5,22)	32.000000
f(8,6,17)	60.000000
a(1,8,6)	1.000000
f(8,6,18)	60.000000
f(8,7,18)	83.000000
a(1,8,7)	1.000000

Optimized Multi Routing Path

For the given configurations, the results of computed variables TrafficFlowPerLink($F_L^{s,t}$) and TrafficSplitRatio($\alpha_i^{s,t}$) is shown in Table C.3

TABLE C.3: Optimized Multi Routing Path Solution

Variable Name	Solution Value
f(1,2,1)	132.000000

Continued on next page

Table C.3 – continued from previous page

Variable Name	**Solution Value**
a(1,1,2)	1.000000
f(1,3,6)	130.000000
a(1,1,3)	1.000000
f(1,4,1)	284.400000
a(3,1,4)	0.218769
f(1,4,3)	8.000000
a(2,1,4)	0.006154
f(1,4,6)	8.000000
f(1,4,8)	284.400000
f(1,4,14)	1007.600000
a(4,1,4)	0.775077
f(1,4,15)	1007.600000
f(1,5,15)	173.000000
a(1,1,5)	1.000000
f(2,1,11)	240.000000
a(1,2,1)	1.000000
f(2,3,2)	200.000000
a(1,2,3)	0.166667
f(2,3,6)	1000.000000
a(4,2,3)	0.833333
f(2,3,11)	1000.000000
f(2,4,8)	318.000000
a(1,2,4)	1.000000
f(2,5,4)	1200.000000
a(4,2,5)	0.857143
f(2,5,7)	200.000000
a(1,2,5)	0.142857
f(2,5,8)	1200.000000

Continued on next page

Table C.3 – continued from previous page

Variable Name	Solution Value
f(3,1,16)	150.000000
a(1,3,1)	1.000000
f(3,2,1)	10.000000
a(4,3,2)	0.047619
f(3,2,12)	200.000000
a(1,3,2)	0.952381
f(3,2,16)	10.000000
f(3,4,3)	163.000000
a(1,3,4)	1.000000
f(3,5,10)	200.000000
a(1,3,5)	0.198020
f(3,5,15)	810.000000
a(4,3,5)	0.801980
f(3,5,16)	810.000000
f(4,1,4)	390.000000
a(4,4,1)	1.000000
f(4,1,5)	390.000000
f(4,2,18)	109.000000
a(1,4,2)	1.000000
f(4,3,4)	69.400000
a(4,4,3)	0.068039
f(4,3,6)	750.600000
a(9,4,3)	0.735882
f(4,3,11)	750.600000
f(4,3,13)	200.000000
a(1,4,3)	0.196078
f(4,3,18)	750.600000
f(4,3,20)	69.400000

Continued on next page

Table C.3 – concluded from previous page

Variable Name	Solution Value
f(4,5,4)	331.000000
a(1,4,5)	1.000000
f(5,1,5)	120.000000
a(1,5,1)	1.000000
f(5,2,1)	310.000000
a(3,5,2)	1.000000
f(5,2,5)	310.000000
f(5,3,5)	102.000000
a(4,5,3)	1.000000
f(5,3,6)	102.000000
f(5,4,14)	221.000000
a(1,5,4)	1.000000

Bibliography

[1] Global Action Plan Report, An inefficient truth, 2007 URL http://www.globalactionplan.org.uk/

[2] Global e-Sustainibility Initiative (GeSI) *SMART 2020: Enabling the low carbon economy in the information age*, URL http://www.theclimategroup.org/assets/resources/publications/Smart2020Report.pdf

[3] G. Epps, D. Tsiang, and T. Boures. System Power Challenges, 2006. URL http://www.slidefinder.net/c/ciscoroutingresearch/seminaraugust29/1562106

[4] Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2010 2015.

[5] R. Tucker, J. Baliga, R. Ayre, K. Hinton, W. Sorin. Energy Consumption in IP Networks. *ECOC Symposium on Green ICT 2008, Belgium.*

[6] T.D. Burd and R.W. Brodersen. Design Issues for Dynamic Voltage Scaling. *Proceedings of International Symposium on Low Power Electronics and Design*, pp. 9-14, 2000.

[7] M. Gupta, S. Singh. Greening of the Internet. *Prceedings of SIGCOMM03*, August 2003.

[8] S. Nedevschi, L. Popa, G. Iannaccone, S. Ratnasamy, and D. Wetherall. Reducing Network Energy Consumption via Sleeping and Rate- Adaptation. *Proceedings of USENIX NSDI*, 2008.

[9] J. Guichard, F. L. Faucheur, and J.-P. Vasseur. Definitive MPLS Network Designs. *Cisco Press*, 2005.

[10] Mingui Zhang, Cheng Yi, Bin Liu, Beichuan Zhang. GreenTE: Power-Aware Traffic Engineering. *proceedings of the 18th IEEE International Conference on Network Protocols (ICNP)*, 2010.

[11] OSPF Version 2 (RFC 2328) - Internet Engineering Task Force, April 1998. URL http://www.ietf.org/rfc/rfc2328.txt

[12] IS-IS (RFC 1195) - Internet Engineering Task Force, December 1990. URL http://www.ietf.org/rfc/rfc1195.txt

[13] Ernesto Q.V. Martins and Marta M.B. Pascoal. A new implementation of Yens ranking loopless paths algorithm, October 2002.

[14] Cisco, Configuring OSPF, 2006.

[15] B. Fortz and M. Thorup. Optimizing OSPF/IS-IS weights in a changing world. *IEEE Journal on Selected Areas in Communications*, vol.20, no.4, pp.756-767, May 2002.

[16] M. Ericsson, M.G.C. Resende and P.M. Pardalos. A genetic algorithm for the weight setting problem in OSPF routing. *Journal of Combinatorial Optimization*, vol.6, no.3, Nov. 2002.

[17] T. H. Cormen, C.E. Leiserson, R.L. Rivest and C. Stein. Introduction to algorithms. *2nd edition, MIT Press, Cambridge Massachusetts*, 2001.

[18] J. Chabarek, J. Sommers, P. Barford, C. Estan, D. Tsiang, and S. Wright. Power Awareness in Network Design and Routing. *Proceedings of IEEE INFOCOM*, 2008.

[19] N. Vasic and D. Kostic. Energy-Aware Traffic Engineering. *Proceedings of the 1st International Conference on Energy-Efficient Computing and Networking*, 2010.

[20] Antonio Cianfrani, Vincenzo Eramo, Marco Listanti, Marco Marazza, Enrico Vittorini. An Energy Saving Routing Algorithm for a Green OSPF Protocol. *Proceedings of IEEE Conference on Computer Communications Workshops*, 2010.

[21] B. Fortz and M. Thorup. Internet Traffic Engineering by Optimizing OSPF Weights. *Proceedings of IEEE INFOCOM*, 2000.

[22] B. Fortz. Optimizing OSPF/IS-IS Weights in a Changing World. *IEEE Journal On Selected Areas In Communications*, vol. 20, no. 4, pp. 756767, May 2002.

[23] B. Heller, S. Seetharaman, P. Mahadevan, Y. Yiakoumis, P. Sharma, S. Banerjee, and N. McKeown. ElasticTree: Saving Energy in Data Center Networks. *Proceedings of USENIX NSDI*, 2010.

[24] OpenFlow. URL http://www.openflow.org/wp/learnmore/.

[25] A. Elwalid, C. Jin, S. Low, and I. Widjaja. MATE: MPLS Adaptive Traffic Engineering. *in Proceedings of IEEE INFOCOM*, 2005.

[26] S. Kandula, D. Katabi, B. Davie, and A. Charny. Walking the Tightrope: Responsive Yet Stable Traffic Engineering. *Proceedings of ACM SIGCOMM*, 2005.

[27] D. Applegate and E. Cohen. Making Intra-Domain Routing Robust to Changing and Uncertain Traffic Demands: Understanding Fundamental Tradeoffs. *Proceedings of ACM SIGCOMM 03*, Karlsruhe, Germany, August 2003.

[28] J. Rexford, A. Greenberg, G. Hjalmtysson, D.A. Maltz, A. Myers, G. Xie, J. Zhan, and H. Zhang. Network-wide decision making: Toward a wafer-thin control plane. *Proceedings of ACM SIGCOMM HotNets Workshop*, November 2004.

[29] A. Greenberg, G. Hjalmtysson, D.A. Maltz, A. Myers, J. Rexford, G. Xie, H. Yan, J. Zhan, and H. Zhang. Clean slate 4D approach to network control and management. *ACM SIGCOMM Computer Communication Review*, Volume 35, Issue 5, October 2005.

[30] Csaszar.A, Enyedi.G, Retvari.G, Hidell.M and Sjodin.P. Converging the evolution of router architectures and IP networks. *IEEE Network Magazine*, Vol. 21, No.4, 2007.

[31] 3GPP Technical Specification 33.401. *System Architecture Evolution (SAE): Security Architecture (Release 8)*. URL http://www.3gpp.org

[32] Enabling mobile broadband growth - Evolved Packet Core. *Technical white paper from Nokia Siemens Network*, 2009.

[33] Psenak P., Mirtorabi S., Roy A., Nguyen L., Pillay-Esnault P. Multi-Topology (MT) Routing in OSPF IETF Request for Comments: 4915, 2007.

[34] Przygienda T., Sagl Z., Shen N., Sheth N., Pillay-Esnault P. M-ISIS: Multi Topology (MT) Routing in Intermediate System to Intermediate Systems (IS-ISs) IETF Request for Comments: 5120, 2008.

[35] Kvalbein A., Lysne O. How can Multi-Topology Routing be used for Intrado-main Traffic Engineering? URL http://www.ietf.org/rfc/rfc3630.txt *SIGCOMM workshop on Internet network management*, 2007.

[36] D. Katz, K. Kompella, and D. Yeung. RFC 3630: Traffic Engineering (TE) Extensions to OSPF Version 2, September 2003. URL http://www.ietf.org/rfc/rfc3630.txt

[37] Smart Edge 400 Router Hardware Guide. *Ericsson Internal.* URL http://calstore.internal.ericsson.com/alexserv?ID=15523&DB=31772-en_lzn7830002_p7d1.alx&FN=26_15330-CRA1191170_1Uen.G.html#CHAPTER5.3

[38] Smart Edge 800 Router Hardware Guide. *Ericsson Internal.* URL http://calstore.internal.ericsson.com/alexserv?ID=15523&DB=31772-en_lzn7830002_p7d1.alx&FN=28_15330-CRA1191170_1Uen.G.html

[39] Smart Edge 400 Power Calculator. *Ericsson Internal Tool.*

[40] Smart Edge 800 Power Calculator. *Ericsson Internal Tool.*

[41] Introduction to OPL. *IBM ILOG CPLEX Optimization Studio V12.4*. URL http://pic.dhe.ibm.com/infocenter/cosinfoc/v12r4/index.jsp

[42] "CPLEX". URL http://www.ilog.com/products/cplex/

[43] D. Cherubini, A. Fanni, A. Frangioni, C. Murgia, M.G. Scutella and P. Zuddas. A Linear Programming Model for Traffic Engineering in 100% Survivable Networks under combined IS-IS/OSPF and MPLS-TE Protocols. *Published in: Journal Computers and Operations Research*, Vol 38, Issue 12, December 2011.

[44] FILE FORMAT REFERENCE MANUAL - CPLEX LP FORMAT, IBM ILOG CPLEX Optimization Studio V12.4. URL http://pic.dhe.ibm.com/infocenter/cosinfoc/v12r4/index.jsp

[45] CPLEX INTERACTIVE OPTIMIZER. URL http://www2.imm.dtu.dk/courses/02713/lpinfo.pdf

[46] Meral Sh. K-Shortest Path- Yen's algorithm, August 2011. URL http://www.mathworks.com/matlabcentral/fileexchange/32513