Enabling large scale cloud services by software defined wide area network

by

Haifa SASSI

THESIS PRESENTED TO ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
IN PARTIAL FULFILLMENT FOR A MASTER'DEGREE
WITH THESIS IN ELECTRICAL ENGINEERING
M.A.Sc

MONTREAL, JULY 11, 2017

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

BOARD OF EXAMINERS (THESIS M.SC.A. OR THESIS PH.D.)

THIS THESIS HAS BEEN EVALUATED

BY THE FOLLOWING BOARD OF EXAMINERS

Mr Mohamed Cheriet, Thesis Supervisor
Department of Automated Production Engineering, École de technologie supérieure

Mr Kim Khoa Nguyen, Thesis Co-supervisor
Department of Electrical Engineering, École de technologie supérieure

Mr Georges Kaddoum, President of the Board of Examiners
Department of Electrical Engineering, École de technologie supérieure

Mr Michel Kadoch, Member of the jury
Department of Electrical Engineering, École de technologie supérieure

THIS THESIS WAS PRESENTED AND DEFENDED

IN THE PRESENCE OF A BOARD OF EXAMINERS AND PUBLIC

5 JULY 2017

AT ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

ACKNOWLEDGMENT

I would like to give a special thanks to my supervisor Mr Mohamed Cheriet**,** professor in the department of Automated Production Engineering and director of Synchromedia laboratory, for his guidance, his availability and his scientific and moral support during this master.

Also, I would like to thank my co-supervisor**,** Mr Kim Khoa Nguyen, professor in the department of Electrical Engineering for his valuable suggestions and corrections and his patience which gave me the effective support for my project.

I would like to thank Mrs Saida Khazri and all Synchromedia Laboratory members for their support, their help and for the great research environment.

Finally, I would like to dedicate this master to my parents and my fiancé: my dear and loving mother; the source of my inspiration and my success, my dear father; the man who taught me that the respect is the best kind of knowledge and my dear fiancé; the most supportive person in my life.

APPROVISIONNEMENT À GRANDE ECHELLE DES SERVICES BASÉS SUR LE NUAGE INFORMATIQUE À TRAVERS SDN DANS LES RÉSEAUX ÉTENDUS

Haifa SASSI

RÉSUMÉ

Interconnecter les centres des données (DC) d'une manière efficace et utiliser la totalité des ressources disponibles dans le réseau étendu (WAN) présentent les challenges les plus difficiles pour les fournisseurs de services (SP). Dans ce mémoire, nous présentons une nouvelle approche pour optimiser l'ingénierie de trafic dans le réseau WAN qui interconnecte les centres des données (Inter-DC WAN) en utilisant le nouveau concept de réseaux programmables (SDN). Nous formulons un modèle mathématique pour l'optimisation de l'allocation de la bande passante aux différents flux appartenant aux différentes classes de services (CoS) en se basant sur la priorité des services et l'état actuel du réseau. L'objectif de notre formulation mathématique est de maximiser l'utilisation des ressources dans le réseau et de minimiser la consommation d'énergie. En outre, le modèle proposé prend en considération la communication entre les domaines avec différentes spécifications et respecte les spécifications techniques sous-jacentes telles que MPLS.

Afin de construire notre modèle, nous considérons quatre formulations mathématiques pour la consommation d'énergie des nœuds et liens existants dans la topologie : le modèle au repos, le modèle entièrement proportionnel, le modèle agnostique et le modèle en escalier et nous adoptons le modèle MPLS pour le réseau WAN qui interconnecte les centres des données. Nous proposons un algorithme déterministe qui se base sur les solveurs de programmation linéaire pour résoudre notre problème d'optimisation et nous comparons la performance de notre algorithme avec celle de deux modèles existants : SWAN, le modèle proposé par Microsoft, qui s'intéresse à la maximisation de l'utilisation des ressources et un autre modèle qui vise à minimiser la consommation d'énergie pendant la procédure d'allocation de la bande passante aux différentes flux. Les expérimentations effectuées dans l'environnement de simulation montrent que la solution proposée est capable d'une part d'exploiter la capacité physique existante dans le réseau afin de répondre à la demande de l'utilisateur en termes de bande passante et d'autre part de minimiser l'énergie consommée pour router le trafic. Comme le problème d'optimisation proposé est NP-difficile, nous proposons également une heuristique de type glouton pour améliorer le temps d'exécution de la solution proposée.

Mots clés: Inter-DC WAN, réseau programmables (SDN), ingénierie de trafic centralisée, consommation d'énergie, maximisation de l'utilisation des ressources, MPLS.

ENABLING LARGE SCALE CLOUD SERVICES BY SOFTWARE DEFINED WIDE AREA NETWORK

Haifa SASSI

ABSTRACT

Interconnecting data centers (DCs) efficiently and using the fully available capacity of existing resources in Wide Area Network (WAN) seems to be one of the most challenging issues for service providers (SPs). In this master memory, we investigate a new approach to optimize traffic engineering in WAN which interconnects DCs (Inter-DC WAN) using Software Defined Networking (SDN). We propose a model to optimize bandwidth allocation to flows belonging at different Classes of Services (CoS) according to their priority and the current network state. The proposed model aims to maximize the throughput in the network and to minimize the overall energy consumption. The proposed model takes into account inter-domain communication and respects underlying technology specifications such as Multi-Protocol Label Switching (MPLS).

To build our model, we consider four mathematical expressions for energy consumption of the topology nodes and links namely: the idle, the fully proportional, the agnostic and the step increasing models, and we adopt the MPLS model for Inter-DC WAN. We propose a deterministic algorithm to solve the optimization problem using Linear Programming (LP) solvers and we compare its performances with two existing models: Microsoft solutions' SWAN which focuses on throughput maximization, and a base line model which aims to minimize energy consumption while allocating bandwidth to different flows. Experiments in the simulation environment show that the proposed solution can optimally exploit available physical capacity in the network to afford users demand in terms of bandwidth and uses the minimum energy to carry traffic. The proposed optimization model is NP-hard, so we propose a greedy heuristic to improve the runtime of the proposed solution.

Keywords: Inter-DC WAN, Software Defined Networking (SDN), centralized traffic engineering, energy consumption, throughput maximization, MPLS.

TABLE OF CONTENTS

Page

LIST OF TABLES

Page

LIST OF FIGURES

Page

LIST OF ABBREVIATIONS

ACL	Access Control List table
ARIMA	Auto-Regressive Integrated Moving Average
ATM	Asynchronous Transfer Mode
BFS	Breadth-first search
BGP	Border Gateway Protocol
BW	Bandwidth
B4	Google's solution for Inter-DC WAN
CMU	Carnegie Mellon University
CoS	Class of Services
CPU	Central Processing Unit
CSPF	Constrained Shortest Path First
DC	Data Center
DFS	Depth- First Search
DiffServ	Differentiated Services
DOV	Delay-Optimized Video traffic routing
DSPP	Dynamic Service Placement Problem
DSCP	Differentiated Service Code Point
D-ViNE	Deterministic Virtual Network Embedding
ECMP	Equal Cost Multipath Routing
FG	Flow Group
GR	Guaranteed Rate
G-WAN	Google Wide Area Network
IETF	Internet Engineering Task Force
IGP	Interior Gateway Protocol
InterServ	Integrated Services
InP	Infrastructure Provider
IP	Internet Protocol
IPsec	Internet Protocol Security
ISIS	Intermediate System- Intermediate System protocol

LAN	Local Area Network
LP	Linear Programming
LPM	Longest Prefix Match table
L3 VPN	Layer 3 Virtual Private Network
MCF	Multi-Commodity Flows
MCTEQ	Multi-Class QoS Guarantee in Inter-Data Center Communications
MCTE	Variant of MCTEQ without delay constraint
MIP	Mixed Integer Programming
MLBF	Mean Load Balancing Factor
MMF	Maximum-Minimum Fairness approach
MPC	Model Predictive Control
MPLS TE	Multi-Protocol Label Switching Traffic Engineering
NA	Nash Equilibrium
NCA	Network Control Applications
NCS	Network Control Servers
NSF	National Science Foundation
NUM	Network Utility Maximization
OFA	OpenFlow Agent
OFC	OpenFlow Controller
OTN	Optical Transport Network
OS	Operating System
QoS	Quality of Service
RAP	Routing Application Proxy
RIB	Routing Information Base
R-ViNE	Random Virtual Network Embedding
SDN	Software Defined Networking
SD-WAN	Software Defined Wide Area Networking
SP	Service Provider
SLA	Service Level Agreement
SWAN	Software-Driven Wide Area Network

SWP	Social Welfare Problem
TCP	Transmission Control Protocol
TED	Traffic Engineering Database
TE op	Traffic Engineering Operation
TE	Traffic Engineering
TG	Tunnel Group
UCB	University of California, Berkeley
VC	Virtual Circuit
ViNEYard	Virtual Network Embedding Algorithms With Coordinated Node and Link Mapping
VN	Virtual Network
VPLS	Virtual Private LAN Service
VPN	Virtual Private Network
WAN	Wide Area Network
WDM	Wavelength Division Multiplexing
WiNE	Generalized Virtual Network Embedding algorithm
W-MPC	Extended Model Predictive Control

CHAPTER 1

INTRODUCTION

1.1 Context

The Wide Area Network (WAN) that interconnects data centers (DCs) is a critical infrastructure, it is very sensitive to congestion and failures, and its resources are often under-utilized. In fact, WAN links are typically used to 30%-40% average utilization (Jain et al., 2013). Moreover, bandwidth demand continues to increase rapidly and WAN traffic is growing at an even faster rate; as a consequence, it is required to improve DCs interconnection to overtake existing problems related to network utilization and QoS guarantee.

Nowadays, researchers are focusing on two main objectives: first, enhance the efficiency and the flexibility of WAN that interconnects DCs, second, minimize the total cost of this network. In fact, the infrastructure of Inter-DC WAN is an expensive resource with amortized annual cost of 100s of millions of dollars when it provides 100s of Gbps to Tbps of capacity over long distances (Hong et al., 2013) .For these reasons, interconnecting DCs is considered as a large investment for Service Providers (SPs) and this investment must allow Inter-DC WAN to meet the following requirements:

- Respecting Service Level Agreements (SLAs) contracted between customers and SPs;
- Increasing the average of use of available resources (links and nodes) in order to allow WAN to carry more traffic with the same capacity;
- Reducing the total cost for the customer (resource costs) and for the SP (reconfiguration costs, maintenance costs and operation costs).

To this end, major challenges must be addressed. In this master thesis, we will focus on three main challenges which are: bandwidth, cost and operations.

Bandwidth: due to the rapid growth and the diversity of applications in WAN such as video conference, IPTV, video streaming, etc., the aggregate size of inputs and outputs is huge. Therefore, Inter-DC WAN must provide reliability and high capacity connections to ensure efficiency and scalability (Jain et al., 2013).

Cost: for a SP, it's primordial to minimize the total cost which includes the allocation costs, reconfiguration costs, maintenance costs, operation costs, etc. That's why; the industry is always looking for new techniques to enhance high-speed networking and to interconnect DCs at the lowest possible cost. An important portion of costs also include energy consumption which is increasing exponentially (Zhang et al., 2010). In order to provide a green and energy aware solutions, it is essential to focus on the minimization of these factors when designing Inter-DC WAN.

Operations: establishing a connection between two DCs should be fast. Managing this connection should not require manual operational tasks because they are expensive, complex, slow, and so they can produce many errors. Therefore, it's very important to minimize the manual operations by automating frequent tasks in order to design a Zero Touch Network: ZTN (Sassi et Subedi, 2017).

Prior solutions dealing with challenges related to bandwidth utilization, cost minimization, and the automation while ensuring efficiency, reliability and scalability in Inter-DC WAN can be classified into three main categories:

Traditional technologies: they are used in legacy WAN that interconnects DCs such as: Virtual Private Networks (VPNs) (eTutorials.org, 2017) , Multi-Protocol Label Switching Traffic Engineering (MPLS-TE) (Parra, Rubio et Castellanos, 2012), VPN over MPLS (Cittadini, Battista et Patrignani, 2013; Dumka et al., 2015; Sun et Wu, 2012) and Virtual Private LAN Service (VPLS) (Battista, Rimondini et Sadolfo, 2012; Lu et Xu, 2010) . These techniques lack of flexibility and efficiency, and they are highly complex in terms of configuration and maintenance.

Virtual WAN: it interconnects DCs using the concept of network virtualization (Chowdhury, Rahman et Boutaba, 2012; Yunos et al., 2013). This approach is proposed to enhance

flexibility in Inter-DC WAN and to maximize resources utilization compared with traditional solutions. However, virtual WAN does not optimize resource allocation based on the current and the global state of the network.

Software Defined Wide Area Network (SD-WAN): is based on two main concepts which are network virtualization and Software Defined Networking (SDN) applied to WAN (Hong et al., 2013; Jain et al., 2013; Liu, Niu et Li, 2016; Wang et al., 2014) . The main contribution of this approach is the centralized and global view of the current state of the network provided by SDN which can be used to optimize resource allocation in order to ensure efficiency, flexibility and scalability in Inter-DC WAN. Many companies, like Google and Microsoft, are adopting SD-WAN because of its ability to minimize the total cost of Inter-DC WAN design while improving its utilization (Hong et al., 2013; Jain et al., 2013). However, adopting this technique to interconnect DCs is a new approach, that's why, existing solutions based on SD-WAN still has limits which should be taken into account in future work.

1.2 Problem statement

Adopting SDN technology in Inter-DC WAN is a new trend in networking field. SD-WAN provides a globally central view of the current state of the network which can be used to make optimal decisions for resource allocation, unlike legacy WAN which is based on local decisions made by each network node. In fact, using traditional approaches, Inter-DC WAN can be stuck in suboptimal state related to resource allocation as shown in Figure 1.1 which can affect the required performance of Inter-DC WAN.

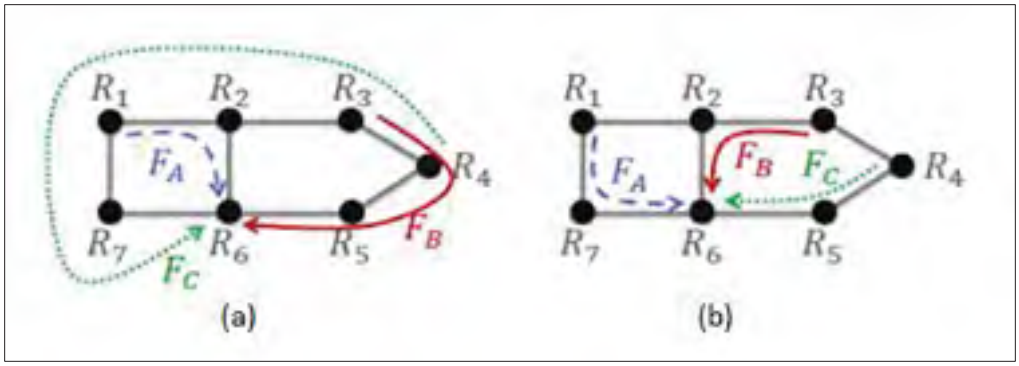


Figure 1.1 Local path selection using MPLS TE (a) vs global optimal path (b)
Taken from Hong et al. (2013)

This thesis lies within this context, and is dedicated to the optimization of traffic engineering in WAN in order to minimize the total cost (e.g., energy consumption) using the SD-WAN approach. Our problem is to overlay optimally a virtual network on the top of a physical Inter-DC WAN as shown in Figure 1.2 while minimizing the total cost and satisfying application requirements in terms of bandwidth demand and QoS. Existing solutions such as B4 (Jain et al., 2013) presented by Google, and SWAN (Hong et al., 2013) of Microsoft, both focus on optimizing resource allocation in WAN while respecting priorities of different flows CoS in order to maximize resource utilization and throughput in the network. These solutions do not take into account network specifications and inter-domain communication. On the other hand, these solutions are not energy aware: they do not consider energy parameters in their problem formulation.

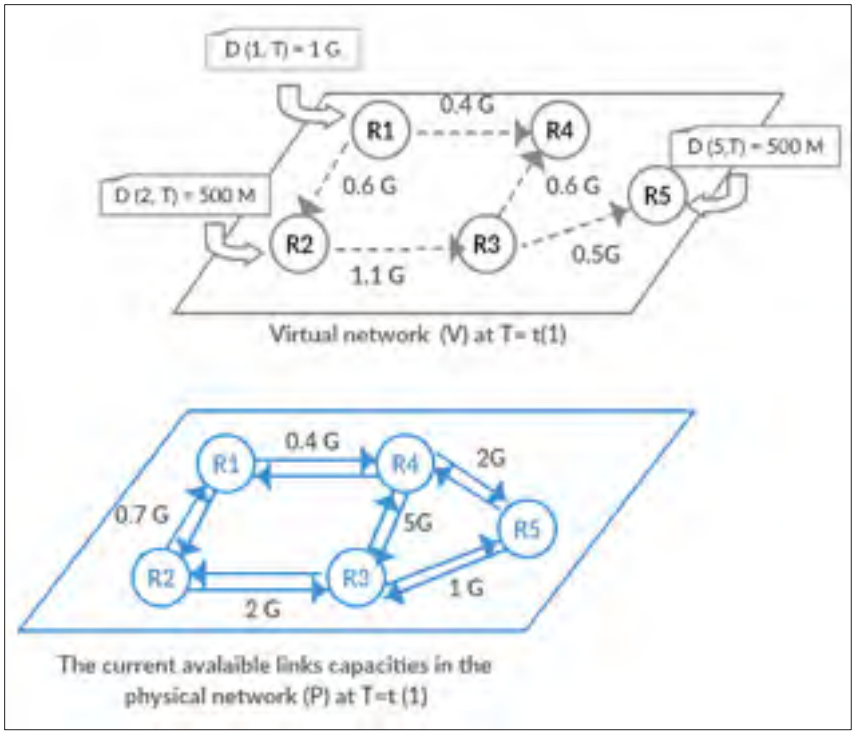


Figure 1.2 Graphical presentation of the problem statement

Our main goal is to maximize network utilization and to minimize the total energy consumption while taking into account network specifications and inter-domain communication. Evolved from the SWAN (Hong et al., 2013), we will use the global view of the current state of the network provided by SDN controllers to optimize the virtual network on top of the physical WAN. Figure 1.3 presents the architecture of SDN network which contains three main layers: infrastructure, control and application layer. Our solution will be deployed as a SDN application in order to be generic and to ensure the portability and the adaptability with different WAN infrastructure belonging to different SPs.

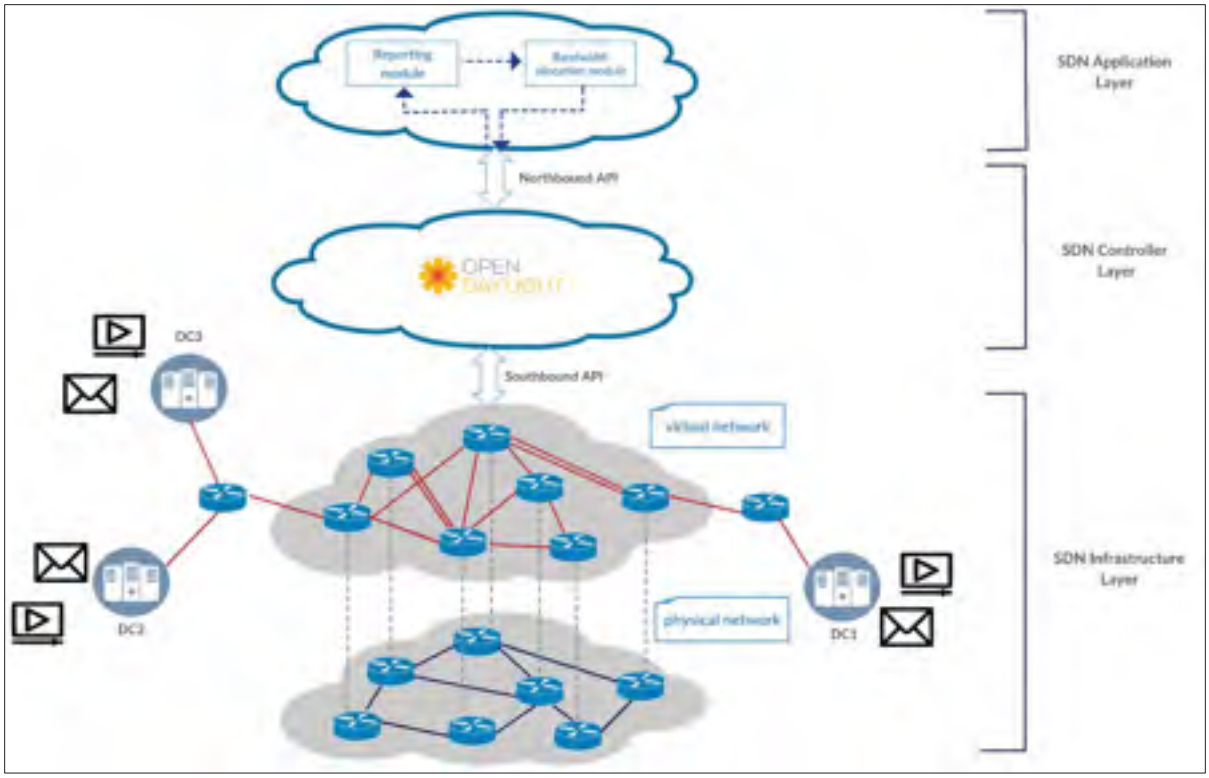


Figure 1.3 Deploying the proposed solution in SDN application layer

To achieve our goals, a number of research questions need to be addressed throughout this thesis. Therefore, a research objective will be defined, which will then be divided into specific objectives. Our research methodology will be proposed targeting these objectives.

1.3 Research questions

In order to build a virtual network on top of a physical Inter-DC WAN, so that the utilization of available physical resources will be maximized and the total cost in terms of energy consumption will be minimized, it is primordial to deal with the following research questions:

RQ1: How is a virtual network mapped into the physical network while respecting existing physical capacities of different components (links and nodes) in real time?

The first challenge in this thesis is the mapping problem: mapping virtual network into the physical network while taking into account existing physical capacities and maximizing throughput in the network. To optimally embed virtual links onto the physical network, we should have a global view of the current state of the network which is provided by SD-WAN approach. Traditional and virtual solutions for Inter-DC WAN use the local state of the network provided by routers which can lead to a suboptimal mapping and a waste of resources. In order to provide efficient traffic engineering, we propose in this thesis to investigate how to use the centralized and the global view of the state of the network provided by SD-WAN to optimize the mapping between virtual and physical network in order to maximize throughput in Inter-DC WAN in real time.

RQ2: How can the network be energy efficient when bandwidth is allocated to applications belonging to different CoS?

Existing works such as B4 (Jain et al., 2013) and SWAN (Hong et al., 2013) do not take into account energy parameters in their proposed models. This work aims to provide an optimal and green traffic engineering in Inter-DC WAN. To meet this goal, the minimization of the total energy consumption must be taken into account in the proposed optimization model. So, it is primordial to model energy consumption of different components (links and nodes) based on the current load in the network. In this thesis we consider the energy efficiency problem from the high level which contains logical nodes and links, when applying our proposed model to a specific network (e.g: DWDM) other components should be taken into account to compute the total energy consumption.

RQ3: How to add new constraints related to network specifications and inter-domain communication to the optimization model?

Taking into account constraints related to the nature of WAN is very important to improve outputs of the optimization model proposed in this thesis. In fact, considering network

specifications, in our case MPLS, and the communication inter domains with different policies such as: maximum/minimum capacity that could be allocated to single flow, etc. will simulate the real behavior of Inter-DC WAN and make our approach very realistic: it can be deployed easily in a real environment unlike existing works which consider just two main layers: the virtual layer and the physical layer.

1.4 Objectives

The main objective of our work is to optimally define the virtual network on the top of the existing physical network in order to satisfy applications requirements (Bandwidth and QoS) and to minimize the total cost related to energy consumption and resource utilization. This objective can be divided into five specific objectives based on previous research questions.

SO1: Model the energy consumption of links and nodes and add these parameters to the proposed optimization model in order to minimize energy consumption when allocating bandwidth to applications.

To answer our third research question related to the minimization of the total energy consumption of the proposed solution it is primordial to model energy consumption of links and nodes in the network based on the allocated bandwidth in each of these components. So, we must start by studying existing energy consumption models, then, proposing our mathematical formulation. Then, we must add these parameters to the proposed optimization model in order to be able to generate an energy aware solution.

SO2: Add constraints related to network specifications (MPLS) and inter-domain communication to the proposed optimization model.

Existing solutions do not consider network specifications or communication inter-domain; in fact, physical WAN can be a combination of different MPLS domains with different specifications. So it is very important to consider the diversity of domains and their different specifications in the proposed optimization model. Through this specific objective, we will

focus on adding constraints related to network specifications (MPLS in our case) and communication inter-domain to the proposed problem formulation.

SO3: Build an optimization model and implement a solution to optimize resources allocation and to maximize throughput in the network while minimizing the total cost in terms of energy consumption and respecting underlying specifications and inter-domain communication.

To respond to our research questions related to throughput maximization through the network virtualization, mapping and energy efficiency, we build an optimization model for throughput maximization which allocates bandwidth to flows based on their CoS while respecting existing physical capacity in the physical network (link and node capacities), minimizing energy consumption and respecting network specifications. As a first step, this model will be inspired from Microsoft solutions' SWAN (Hong et al., 2013). The proposed model uses linear programming (LP) for bandwidth allocation: it uses the throughput maximization approach to allocate bandwidth to flows belonging to different CoS with different priorities. It uses MMF approach to allocate bandwidth to flows having the same CoS. This model generates a virtual topology which should be defined on top of the physical topology to meet bandwidth demand in each node in the network and to guarantee the required QoS. Then, we propose a deterministic algorithm to solve the proposed optimization model using LP solvers.

SO4: Propose a greedy heuristic to improve the runtime of the deterministic algorithm when physical topology is dense.

Finally, in order to improve the performance of the proposed algorithm, which uses LP to solve the proposed optimization model, we propose a greedy heuristic which is able to produce a solution meeting our requirements in terms of the minimization of energy consumption and the maximization of throughput in the network within a shorter runtime.

1.5 Hypothesis

By optimizing the design of the virtual WAN that should be over-layered on the top of the physical WAN, we would satisfy applications demand in terms of resource requirements and QoS while minimizing energy consumption, maximizing resource utilization and optimizing resource allocation.

1.6 Research originality

Our main contributions in this work are: (i) energy consumption modeling for different components (links and nodes) in the network (ii) an optimization model which takes into account different energy profiles of network elements (nodes and links) and cover different layers (iii) a deterministic algorithm proposed to solve the optimization problem using LP solvers and (iv) a greedy heuristic proposed to improve the runtime of the proposed solution.

1.7 Plan

This thesis will be divided into four main chapters:

- In the second chapter, we will present our background and our motivation to deal with the problem statement presented in chapter 1 by introducing traditional solutions used in Inter-DC WAN and SDN technology:
  - The first section in this chapter will expose traditional technologies used in legacy WAN to interconnect DCs, their advantages and their limitations;
  - In the second section of this chapter, we will present Software Defined Networking (SDN) technology: its architecture, its protocols and its contributions;
  - Finally, we will summarize benefits and drawbacks of different solutions presented in this chapter for Inter-DC WAN.
- The third chapter will be dedicated to present the related works:

- The first section will present contributions and limitations of two solutions proposed to interconnect DCs and to optimize resource allocation using network virtualization;
- Then, we will expose four solutions proposed for Inter-DC WAN using the concept of SDN applied to WAN (SD-WAN) and we will discuss their contributions and their limitations;
- The final section will be reserved to present benefits and drawbacks of different solutions presented in this chapter for Inter-DC WAN in order to highlight our main contributions through this thesis.

➢ The fourth chapter will present our methodology to achieve our objectives listed previously in section 1.4.

- In the first section of this chapter, we will present an optimization model which aims to maximize throughput in the network based on bandwidth demand in different nodes;
- In the second section, we will present four models for the energy consumption of different components in the network (links and nodes): the idle, the fully proportional, the agnostic and the step-increasing models;
- Then, we will introduce MPLS network modeling which are constraints that should be added to the proposed optimization model in order to generate a solution which takes into account underlying specifications;
- Then, we will present the final energy aware optimization model with different constraints related to network specifications and inter-domain communication;
- The final section of this chapter will be reserved to introduce the deterministic algorithm proposed to solve the optimization problem using LP solvers and the greedy heuristic proposed to improve the runtime of the deterministic algorithm when the physical topology is dense.

➢ The last chapter, which deals with the results and validation, will present:

- First, the validation protocol proposed to validate our work and our proposed solution;

- Then, the implementation steps of the deterministic algorithm and a comparative study between the performance of the proposed algorithm and two existing solutions which are: Microsoft's solution SWAN and a base line solution;
- Finally, the implementation of the greedy heuristic and a comparative study between its performance and the performance of the deterministic algorithm and Microsoft's solution: SWAN.

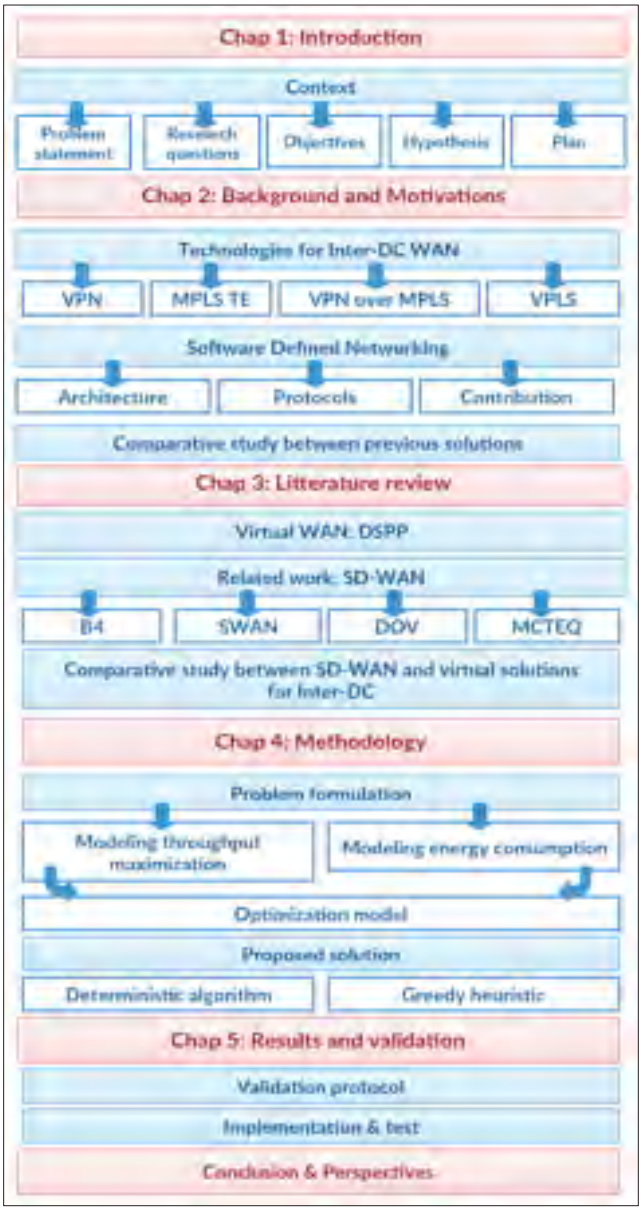


Figure 1.4 Graphical presentation of the plan of the memory

CHAPTER 2

BACKGROUND AND MOTIVATIONS

2.1 Introduction

In this chapter, we present existing approaches used nowadays to interconnect DCs. We start by traditional technologies such as VPN, MPLS TE, VPN over MPLS and VPLS used in legacy WAN, and we will highlight their features and limitations. Then, we introduce the new technology "Software Defined Networking: SDN": its architecture, its protocols and its different contributions. Finally, we introduce few motivations which encourage research to use SDN concept in WAN.

2.2 Traditional solutions for Inter-DC WAN

2.2.1 Layer 3 Virtual Private Network: L3 VPN

2.2.1.1 Overview

Layer 3 Virtual Private Networks (L3 VPNs) are used by SPs in order to provide VPN services to their customers by transmitting sensitive information across the public Internet using their IP backbone. A L3 VPN (Jerkins, 2015) is a set of sites that share common routing information and whose connectivity is controlled by a set of policies and rules. Before introducing MPLS architecture, private networks were deployed by using two basic techniques which are: the overlay model and the peer-to-peer model.

- The overlay model (Detti, Caricato et Bianchi, 2010) uses the virtual circuits of a Frame Relay or ATM service, which means that sites can be interconnected through IP layer above a Layer 2 connectivity service. It is a traditional approach to interconnect distant DCs based on setting up secure end-to-end connections which emulates virtual circuits

over public networks. In this model, the service is negotiated and rendered only in the two end points of the created tunnel;

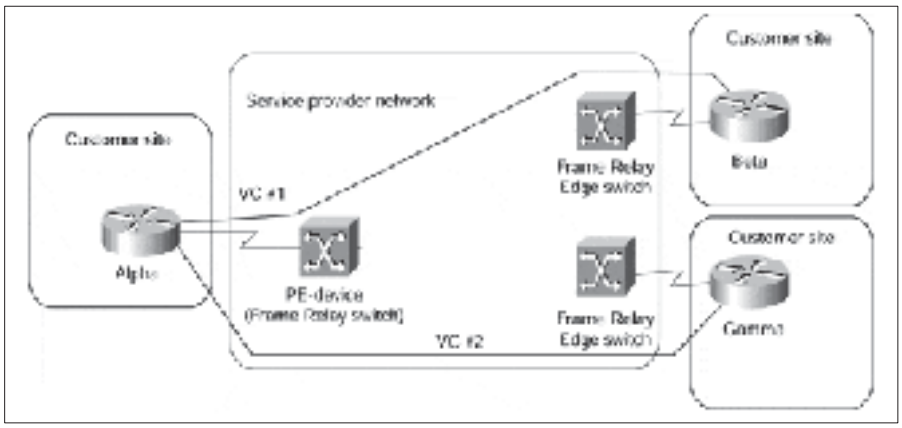


Figure 2.1 Example of VPN using overlay model
Taken from eTutorials.org. (2017)

- In the peer-to-peer model, some limitations of the overlay model were taken into consideration by replacing the use of multiple virtual circuits with a direct exchange of routing information in aggregated tunnels as shown in Figure 2.2.

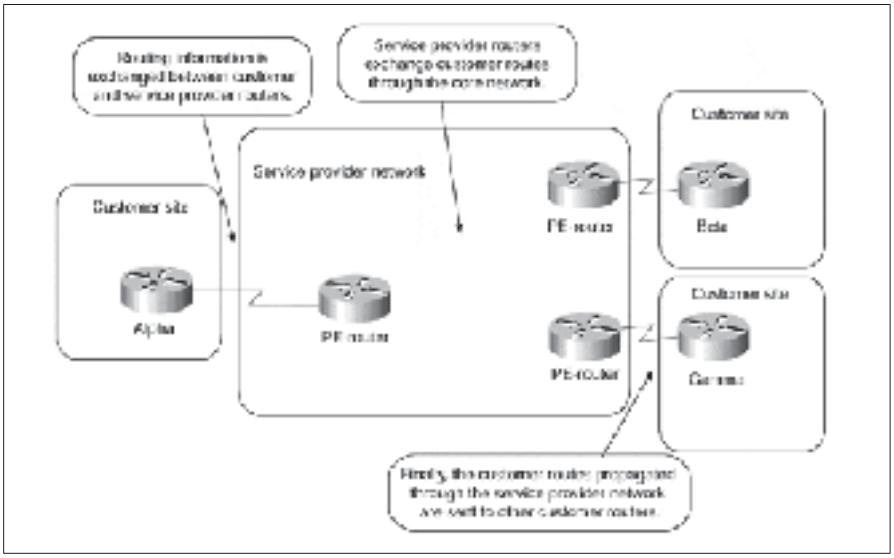


Figure 2.2 Example of VPN peer-to-peer model
Taken from eTutorials.org. (2017)

2.2.1.2 Advantages

The L3 VPN overlay model has the following advantages:

- Possibility to use duplicated addresses;
- Isolation offered by this model between the control plan and security plan.

The L3 VPN peer to peer model also has many advantages compared to the overlay model. The most important ones are:

- Routing and installation procedures are very simple and easy, due to the elimination of multiple virtual circuits concepts used in the overlay model;
- Elimination of the problem related to the size of circuits;
- High scalability compared to the overlay model.

2.2.1.3 Limitations

The main disadvantages of the overlay model are:

- Difficulty and complexity in the optimization of the virtual circuits' size between sites;
- Meshed circuits are required to interconnect the different equipment in the network. In fact, this model has a serious scaling problem when the number of egress routers is large due to the need to establish a tunnel between every two sites in WAN;
- Provisioning is a problem with this model because the time required to configure VPN tunnels increases with the size of the network. That's why, fully meshed networks are very hard to configure and to manage;
- A simple link failure can result in dozens of VCs going down, forcing the IP routing protocols into a major re-convergence.

On the other hand, main drawbacks of the peer-to-peer model are the following:

- Requirements for an IGP protocol to allow the SP to manage all the VPN tunnels;
- This model does not allow using duplicated addresses between customers unlike the overlay model.

2.2.2 Multi-Protocol Label Switching Traffic Engineering: MPLS-TE

2.2.2.1 Overview

Multi-protocol Label switching (MPLS) is an IETF standard which merges layer 2 and layer 3 protocol by using labels switching in the core network, thus reduces the workout of looking the routing table overhead (Malis, 2012; Winter, 2011). Nowadays, many WANs are operated using the MPLS-TE. To effectively use network capacity, MPLS-TE spreads traffic across a number of tunnels between ingress-egress router pairs. Ingress routers split traffic, typically equally across the tunnels to the same egress using the Equal Cost Multipath Routing (ECMP) (Hong et al., 2013) which is a load balancing algorithm used in the session level to choose a path between equal cost paths. To estimate the traffic demand for each tunnel and to find network paths for it, MPLS TE uses the Constrained Shortest Path First (CSPF) algorithm, which identifies the shortest path that can be used to carry the traffic in the network with the respect of some constraints related to service priorities and link capacity (Jain et al., 2013).

MPLS offers a single QoS paradigm; but more importantly, it makes it simple for services to request QoS and have that request mapped through to traffic engineering (Awduche, 1999). MPLS TE is based on the differentiation between services using service priorities and the Differentiated Service Code Point (DSCP) which is a field in the IP header which can be used to classify services.

MPLS TE forwards packets based on labels placed in the header, this method can reduce the complexity of the routing procedure: the source router is responsible for the complex task which is adding the label to the traffic, and routers in the core network simply read the label and forward the packet based on the rules created for that label.

2.2.2.2 Advantages

MPLS TE unifies advantages of two technologies used before to ensure the scalability and the QoS which are the Integrated Services (InterServ) (Yildirim et Girici, 2014) and the Differentiated Services (DiffServ) (Wang et Guo, 2012). So the most important advantages of MPLS TE are the following (Malis, 2012):

- MPLS TE provides an efficient way to forward traffic across the network and to avoid overutilization and underutilization of links and nodes in the WAN;
- MPLS TE takes into account the configured and the available bandwidth of different links in the topology;
- MPLS TE computes the shortest path for services with the highest priority;
- MPLS TE supports the rerouting of traffic around a failed link;
- MPLS TE optimizes the routing procedure by introducing the concept of labels switching;
- MPLS increases network scalability, simplifies network service integration, offers integrated recovery, and simplifies network management (Awduche, 1999).

2.2.2.3 Limitations

MPLS TE is widely used for DCs interconnection, there is some advanced MPLS TE implementation used today by many online SPs such as Google and Microsoft. However, even enhanced versions of MPLS TE have major limitations that can affect the performance of this technology (Strelkovskaya, Solovskaya et Paskalenko, 2015):

- With MPLS TE, services send their traffic without having a global idea about the current state of the network;
- The allocation model used by MPLS TE is inefficient because it is based on local decisions. In fact; no router in a MPLS-TE network has a global view of the current state of the network; as a result, the network can be stuck in a local optimal solution which is globally suboptimal;
- With MPLS TE there is no possibility to coordinate changes in the network. For example, it is impossible to move some flows, in a real time, in order to free links to be used by other flows. MPLS TE does not support such flexibility in WAN.

2.2.3 VPN over MPLS

2.2.3.1 Overview

MPLS VPN is a VPN built on top of an MPLS network (Cittadini, Battista et Patrignani, 2013). It is usually used by SPs to ensure connectivity between companies or DCs. The terms "MPLS IP VPN," "MPLS VPN," "MPLS-based VPN" and “VPN over MPLS” can be used synonymously.

VPN over MPLS is built according to a peer to peer VPN model and uses a connection less architecture to create a highly scalable solution and to ensure the required QoS (Dumka et al., 2015). In this technology, VPN and MPLS work together to create a private virtual network which is extremely efficient at labeling and delivering traffic compared with using only VPN or MPLS TE separately. VPN over MPLS has the ability to ensure the required QoS demanded by enterprises, comparable to other technologies previously presented in this section.

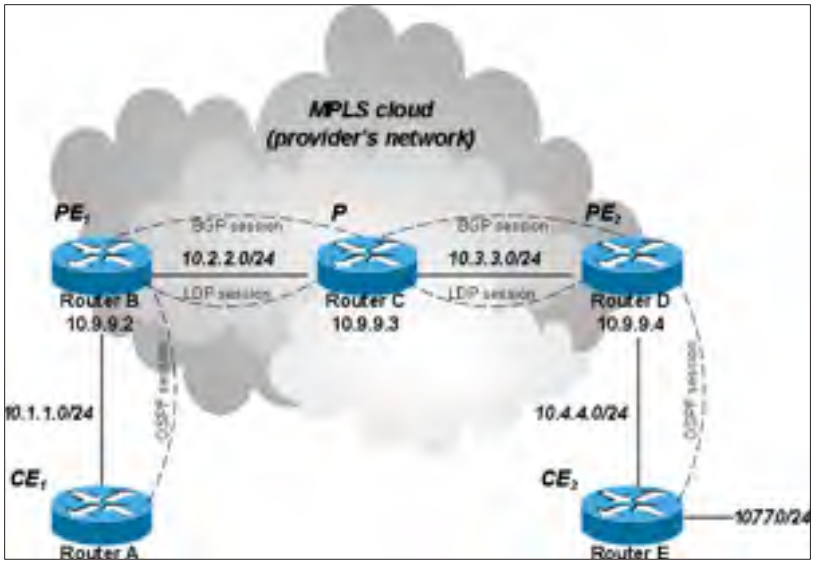


Figure 2.3 VPN over MPLS example
Taken from MikroTik (2010)

2.2.3.1 Advantages

Obviously, VPN over MPLS combines advantages of VPN and MPLS TE. In addition, this technology has other improvements such as (Yunos et al., 2013):

- VPN over MPLS performs in connection less state which means that no prior action is required to establish the communication between two hosts. This criterion makes the procedure of communication establishment very efficient and flexible;
- VPN over MPLS is structured according to the peer-to-peer model to ensure high scalability in the network. In fact, according to this model, a customer site is required to communicate only with one provider to eliminate drawbacks of point to point approach or virtual circuits;
- Another advantage of VPN over MPLS is that no intelligence is required in VPN devices because all of the VPN functions are executed in the core network. That's why there is no need to tunneling protocols like IPsec. So, using VPN over MPLS allows the customer to use less expensive devices or to continue to use same devices;
- And finally, provisioning is easier with VPN over MPLS because this procedure needs to be done only on backbone network devices.

2.2.3.2 Limitations

There are some known limitations of this technology; the most important ones are the following:

- VPN over MPLS does not provide a guaranteed QoS in terms of Service Level Agreement (SLAs) because QoS cannot be provided by MPLS. On the other hand, trying to guarantee the required QoS and adding traffic engineering require keeping state in the network which can generate a scalability problem;
- MPLS VPNs are only private at the routing level: no authentication, confidentiality, or integrity is provided by the architecture (Cittadini, Battista et Patrignani, 2013). In fact, SPs is able to inspect all traffic carried in WAN and the isolation in this case depends on the provider;
- In VPN over MPLS, the provider and the customer share the maintenance of the network which makes this mission very complicated in case of failures or connectivity problems.

2.2.4 VPLS

2.2.4.1 Overview

VPLS is an emerging layer 2 MPLS based services standard that provides multi-point-to-multi-point connectivity to enterprise users. It connects all of its enterprise sites that are located in a single metro area or spanned over multiple metro areas using the scalable-shared IP/MPLS infrastructure to provide a transparent LAN connectivity. Enterprises can create one or more VPLS domain in their carrier's network (Biradar, Alawieh et Mouftah, 2006; Lu et Xu, 2010). Virtual Private LAN Services (VPLS) supports distributed Ethernet LANs, it uses MPLS as the backbone network to route packets. A 100 Mbps interface to VPLS can support an SLA at speeds of 1 Mbps to 100 Mbps of traffic.

2.2.4.2 Advantages

- VPLS has been proven to be an effective and inexpensive way to provide a multi-point Layer 2 Ethernet network business services, as well as network edge infrastructure to residential voice, video and data (Triple Play) services (Hachimi et Bennani, 2007);
- Using VPLS, multiple Ethernet domains are considered as a single Ethernet LAN which allows sharing the same broadcast domain between different domains;
- Using VPLS to interconnect DCs minimize latency in the network;
- In addition of IP, VPLS supports legacy protocols.

2.2.4.3 Limitations

VPLS has the following limitations:

- VPLS has a good performance when the number of sites is limited, in fact, VPLS has a scalability problem;
- The process of provisioning a new VPLS circuit is complicated because the service provider has to check device configurations one by one. Therefore, it is difficult to provide the service on demand (Hu, Yang et Liu, 2016);
- Finally, VPLS is a complex technology because to manage a VPLS network, you need to manage a large number of MAC addresses since there is no hierarchy like IP addresses.

2.3 Software Defined Networking

2.3.1 Overview

The traditional WAN technologies used to interconnect DCs have some issues when the traffic increases; it is characterized by a poor efficiency and poor sharing which can degrade the performance of the current network devices. Besides, these devices can't provide enough flexibility to deal with different packet types and to dynamically execute some path changes

to avoid the underutilization or the sub-utilization of Inter-DC's resources (Hong et al., 2013). As a consequence, WAN needs to be able to adapt some dynamic changes in resource allocation based on the current state of the network in order to increase the average of utilization of links and nodes in inter-DC WAN , to reduce the delay and to ameliorate WAN performance when the traffic is important.

A potential solution proposed to this problem is to extract the data handling rules from the hardware and implement it as software modules. This method enables to have more centralized control over the network traffic which allows improving the performance of the network in terms of efficiency and resource optimization. Such an idea is presented in an innovative technology, called Software-Defined Networking (SDN), its concept was originally proposed by Nicira Networks based on their earlier development at UCB, Stanford, CMU, Princeton (Hu, Hao et Bao, 2014).

Software-defined networking (SDN) is a new approach to design, build, and manage networks, this approach is based on the separation between the control plane and the data plane to better optimize each part. SDN centralizes control of the network by implementing it as software in a separate entity called the controller (Kreutz et al., 2015). This controller is responsible for inserting rules in routers and switches in data plane based on decisions taken by the control plane in order to simplify data delivery in the infrastructure layer. This method helps to make changes globally and to improve the efficiency of the network and to simplify its management (Nunes et al., 2014).

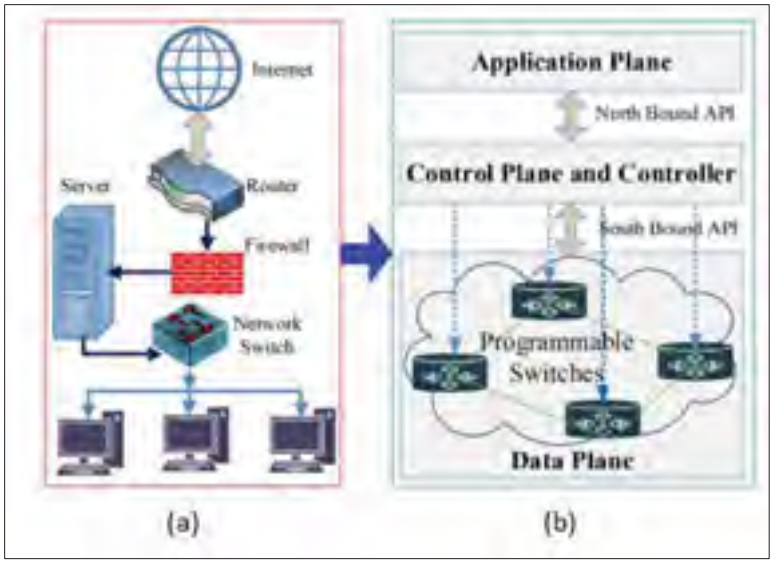


Figure 2.4 Traditional network architecture (a)
versus SDN architecture (b)
Taken from Rawat et Reddy. (2017)

Figure 2.5 shows SDN framework which contains three main layers: data plane, control plane and application plane.

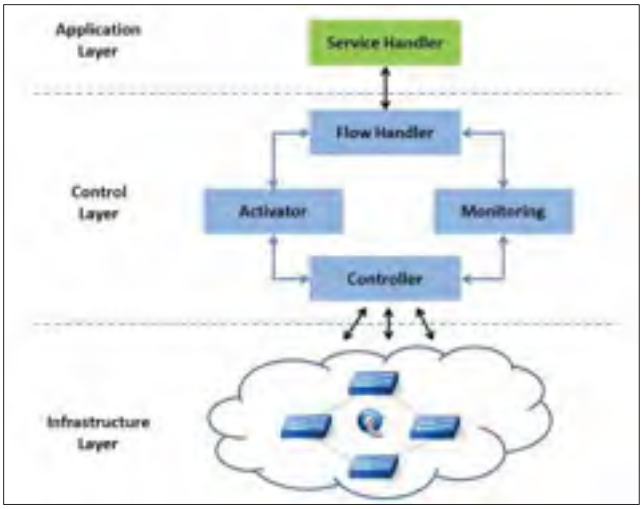


Figure 2.5 SDN framework
Taken from Gouveia et al. (2014)

Data plane: comprises network elements such as switches and routers which expose their availability to the centralized controller via southbound interfaces.
Control plane: contains the centralized controller which is responsible for routing, traffic engineering and mobility.
Application plane: contains SDN applications, which communicate their network requirements to the controller via northbound interfaces.

2.3.2 SDN protocol: OpenFlow

There are many standard protocols used by SDN to ensure the communication between the control plane and the data plane, the most popular one is the "OpenFlow" which allows SDN implementation in both hardware and software devices (Lara, Kolasani et Ramamurthy, 2014). OpenFlow is flow-oriented protocol; it is used for communication between the infrastructure layer and the control layer. In fact, the controller communicates with switches and manages them through OpenFlow. An OpenFlow switch can contain multiple flow tables, a group table and an OpenFlow channel; therefore, OpenFlow switches in the data plane are connected to each other via the OpenFlow ports. To understand the different functionalities of OpenFlow, it is primordial to explain first the following key words:

Flow tables: contains a set of flow entries, the flow table is used to match or to specify an action to incoming packets and to direct them to its destination based on its header. A flow table entry is identified by its match fields and priority: the match fields and priority taken together identify a unique flow entry in the flow table (TS-007, 2012).
Group tables: contains a set of group entries, a flow table may direct a flow to a group Table, which may execute a variety of actions that affect one or more flows.
OpenFlow channel: is the interface used for communication between an OpenFlow switch and an SDN controller. First, the controller uses this interface to configure and to manage switches. Second, it uses it to receive events from switches.
OpenFlow ports: there are physical network interfaces used to pass packets between OpenFlow processing and the rest of the network.

2.3.3 Contributions of SDN

SDN concept is a strong candidate to replace existing architectures because it has many advantages that can overtake limitations of traditional technologies used in WAN to interconnect DCs. So in this section, we will expose the most important advantages of SDN which encourage SPs to think about using to this new approach:

Intelligence and speed: SDN is able to optimize the distribution of the workload in the network thanks to the centralized control and the global knowledge of the current state of different links and nodes in the topology. This optimization can lead to a high-speed transmission in the network and enhance the efficiency of resource allocation.

Easy network management: There is a remote and global control over the network and there is also the possibility to change the network characteristics based on the workload patterns. This is the main advantage of the centralized architecture proposed by SDN. Additionally, the centralized architecture proposed by SDN is a motivation for researchers in traffic characterization and prediction field. In fact, they can develop advanced algorithms to predict failures or link overutilization and based on this prediction module, the controller can update in a real time some rules in different switches in the data plane.

Multi-tenancy: SDN ensures that all tenants have good cross site performance isolation, unlike existing architectures which do not support control network ability between different intra-tenants and inter-tenants.

These advantages listed above present a really good motivation for researchers who are looking for a solution to control and manage their networks in an optimal and efficient way, to minimize the total cost of resource allocation and to achieve the required level of scale. So, using the new concept of SDN in inter-DC WAN will remedy to some limitations of traditional or even virtual technologies used nowadays to interconnect DCs.

2.4 Comparative study between traditional solutions for Inter-DC

In the beginning of this report we mentioned three challenges (bandwidth, cost and operations) that should be taken into account while over layering virtual network on top of the physical Inter-DC WAN. We will present, in the following table, how each solution presented in this chapter deals with these three challenges.

Table 2.1 Existing solutions and Inter-DC WAN challenges

Solution	Bandwidth allocation	Cost	Operations
Layer 3 VPN	Dedicated Distributed (routers) Static	Bandwidth utilization	Manual
MPLS TE	Shared Distributed (routers) Static	Bandwidth utilization Shortest path	Manual
VPN over MPLS	Shared Distributed (routers) Static	Bandwidth utilization Shortest path	Manual
VPLS	Shared Distributed (routers) Static	Bandwidth utilization Shortest path	Manual
SDN	Shared Centralized Dynamic	Bandwidth Energy consumption Delay, load balancing, etc.	Dynamic

2.5 Conclusion

In this chapter, we introduced first traditional technologies used in legacy WAN to interconnect DCs such as: VPN, MPLS TE, VPLS and VPN over MPLS. We studied their advantages and we highlighted their limitations related to resource utilization in order to highlight our contribution. Then, we presented the new concept SDN: its architecture, its protocols and finally its advantages in order to expose our motivation to use SDN approach to optimize traffic engineering in Inter-DC WAN. Finally, we compared the proposed technology for Inter-DC WAN based on three criteria presented previously in the introduction chapter: bandwidth, cost and operations.

CHAPTER 3

RELATED WORKS

3.1 Introduction

In this chapter, we introduce some advanced solutions proposed to optimize the interconnection between DCs which are known as “virtual WAN solutions.” We present how network virtualization can optimize resource allocation and maximize throughput in Inter-DC WAN. Then, we expose four related works which have same objectives related to resource maximization and the minimization of the total cost using SD-WAN approach. And finally, we introduce a comparative study which highlights advantages and limitations of each existing solution used to interconnect DCs presented in chapter 3 in order to emphasize our contribution through this thesis.

3.2 Virtual WAN

3.2.1 DSPP

3.2.1.1 Overview

As we know, large-scale online SPs have made their choice to geographically distributed their cloud infrastructure to host and deliver services in order to be able to ensure performance requirements specified in SLA while satisfying QoS expected by users. The most difficult challenge faced by SPs is to locate different applications while minimizing the hosting cost and ensuring key performance requirements. Existing solutions for the service placement has usually ignored: cost fluctuations, reconfiguration costs, availability of resources and impact of the geographical location. In fact, existing solutions are mostly static or time invariant. The problem of service placement is very similar to the problem of application deployment in WAN proposed in this thesis, both problems focus on optimizing

resource allocation in Inter-DC WAN with the minimum cost while satisfying application requirements and respecting existing physical capacities.

.

The suggested solution in this work is to propose as a first step a control framework based on Model Predictive Control (MPC) approach to provide an online adaptive control mechanism which focuses on reducing service provider costs, resource allocation and reconfiguration costs (Zhang et al., 2012). And as a second step, this solution will be extended to a game-theory model to consider the competition among multiple SPs while taking into account the capacity constraint of each data center (Zhang et al., 2012). This work uses game theory as a tool to optimize resource allocation in case of having multiple SPs competing for distributed resources.

This work proposes three cases: first, it takes into consideration resources required by a single SP and it analyzes how a single SP can respond to its demand in terms of resource allocation and minimize the total cost. Second, they simulate the competition between multiple SPs using game theory. And finally, they take into consideration the contribution of an infrastructure provider to ensure the welfare between different SPs. The main objective is to minimize the total cost dynamically over time with the respect of two main constraints which are SP's demand and resource price fluctuations. Qi Zhang et al focus on helping SPs to dynamically control the number of active servers placed in each data center and to minimize the total resource cost while satisfying SLA and QoS requirements.

3.2.1.2 Proposed solution for single SP

Architecture

To understand the problem of dynamic resource allocation, it is primordial to present the architecture used for distributed service placement, this architecture contains:

Request routers: redirect requests from customers to appropriate servers.

The monitoring module: collects statistics related to the network state.

Analysis and prediction module: is responsible to model the dynamicity of the demand and price fluctuations. On the other hand, it computes and forecasts their future values.

Resource controller: solves DSPP and makes online control decisions at run time. Moreover, it informs the request routers about the number of servers allocated in each data center for each SP.

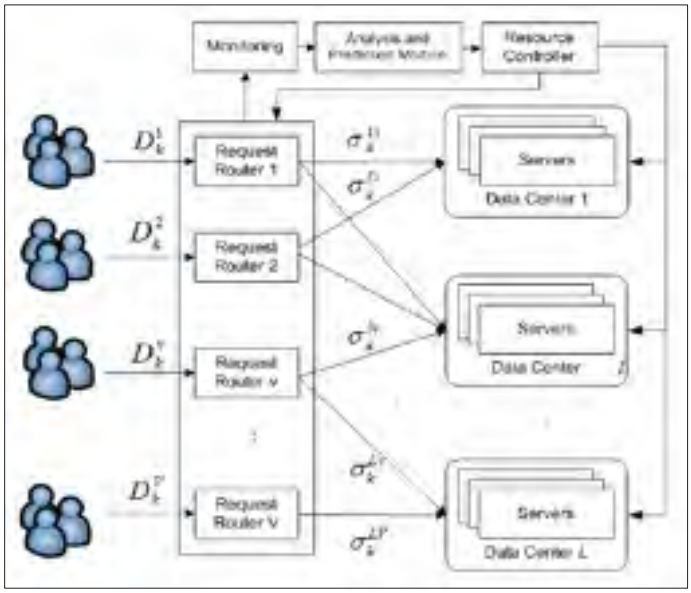


Figure 3.1 Dynamic Service placement architecture
Taken from Zhang et al. (2012)

MPC and W-MPC algorithm

The main objective of SP is to minimize its total cost which contains resource allocation costs and reconfiguration costs. In fact, SP must minimize the total cost under two main constraints: demand constraints and SLA constraints. So, DSPP is a linear quadratic problem that can be solved using the standard methods like: comparison method, elimination method and substitution method. MPC algorithm is used to solve DSSP dynamically based on

resource availability and resource costs fluctuations in different DCs. This algorithm can be summarized in three main steps:

1. At time k, the resource controller predicts the future demand for multiple periods using the Auto-Regressive Integrated Moving Average (ARIMA) (Ariyo, Adewumi et Ayo, 2014) model which predicts a value in a time series using a linear combination of past values in past time series;
2. The controller solves the optimization problem for the prediction horizon starting with the initial state;
3. The solution of the optimization problem will contain a set of values; the controller will only execute the first one in the sequence and other values in the sequence will be used to predict the resource allocation while minimizing the reconfiguration cost.

To conclude, MPC helps to find $u(l, v, k)$ which is the number of servers that should be added between the two states k and $k + 1$ in order to serve the demand v in the location l. But there is a serious problem in this stage: MPC is not enough to guarantee the required SLA in case of underestimated future demand. So to avoid this issue, they propose to increase the frequency at which MPC algorithm runs in order to reduce the duration in which servers are under-provisioned.

After proposing a solution for a single SP, Qi Zhang et al propose to extend it with multiple SPs which can share resources in the existing infrastructure. It is obvious that the goal of each SP is to minimize its operational costs while respecting previous constraints which are SLA performance requirements and demand. In this case, there is an additional constraint that should be added to the previous optimization model which is the current data center capacity and the current resource availability. For each SP_i, the $DSPP_i$ problem is used to minimize its total cost while respecting an additional constraint related to total capacities of different DCs. In this case, competition between SPs is simulated using "Game theory": N player dynamic non-cooperative game, in fact each SP_i makes its decision independently. So the objective here is to find the NE (Nash Equilibrium) which is the stable outcome of the current competition.

To solve this problem, Qi Zhang et al propose to work with W-MPC Nash Equilibrium because it responses to their requirements, they proved that to find the NE that minimizes the cost function, they must find the NE that minimizes the Social Welfare Problem (SWP) function. It means that they try to find NE where costs incurred by all service providers are minimized. But, if SPs decide the placement of their own resources based on the price, some of them will hog all cheap resources and hence the others will get only expensive resources. That's why, Qi Zhang et al extend their work to introduce the contribution of infrastructure providers (InP) to increase the total revenue and to minimize costs for all SPs while ensuring fairness.

So InP's objective is to minimize its utility function which contains a convex penalty that captures the penalty of demand rejection. Utility function of InP can be determined by the revenue from selling resources to different SPs and service quality expressed by the number of rejected resources request. To solve the problem presented previously, InP uses the following algorithm which can be summarized in five main steps:

1. At time k, InP announces the future resource prices for a window $1 \leq t \leq W$, and each $SP_i \in N$ submits its demand;
2. At each step, each SP_i solves the problem its $DSPP_i$;
3. At each step, InP solves an optimization problem that minimize its total cost with the respect of price fluctuations;
4. Finally, the algorithm updates the value of the Lagrangian parameter in each iteration;
5. This process repeats until the solution converges to a local optimal solution.

3.2.1.3 Advantages and limitations

This work uses Game Theory to optimize hosting costs dynamically over time based on both demand and price fluctuations. Game Theory in this case is used to control the network's state when the centralized control defined in SDN concept is absent. In case of uncoordinated SPs which can act in a selfish way, the resulting NE can be significantly suboptimal. So, Qi

Zhang et al propose to take into consideration the contribution of InP in order to maximize the social welfare.Obviously, this framework has achieved its main goal which is the ability to adjust resource allocation based on the dynamicity of resource prices. But the proposed algorithm is very complicated and can generate a huge overhead in the network, especially when the number of players (SPs) that are competing for resources is important, in this case the convergence of the proposed algorithm can take time. In addition, algorithms proposed in case of single SPs or multiple SPs do not take into consideration the difference between price models used in different DCs while solving the placement problem, which can have an important impact on the stable outcome of this competition. Finally, this work does not take into account the minimization of the total energy consumption while allocating resources to different SPs, it just consider the load balancing as a main objective.

3.2.2 ViNEYard

3.2.2.1 Overview

Mosharaf et al propose a new heuristic to deal with the problem of virtual network (VN) embedding which is known as an NP-hard problem. The proposed solution “ViNEYard: Virtual Network Embedding Algorithms With Coordinated Node and Link Mapping” (Chowdhury, Rahman et Boutaba, 2012) allows multiple SPs to use and share the same resources efficiently.

The VN embedding problem focus on the mapping of virtual network (virtual links and virtual nodes) in the existing physical infrastructure while respecting constraints related to physical resources availability and existing capacities. Moreover, this work deals with the VN embedding problem with online requests for the build of VNs, however, the VN embedding problem is known to be NP-hard even in the offline case (Chowdhury, Rahman et Boutaba, 2012). Existing works, usually, separate the mapping of virtual nodes and virtual links but even when all nodes are already mapped into the physical infrastructure, the mapping of virtual links still an NP-hard problem. On the other hand, making the decision of

nodes mapping without taking into account constraints related to link mapping can generate a poor solution which can affect the required performance. As a consequence, this work proposes a solution which takes into consideration the correlation between nodes and links mapping in order to produce an optimal solution for VN embedding with online and offline VN requests.

The main objective of this work is to propose an efficient alternative to solve the VN embedding problem while ensuring an optimal share of existing resources between different SPs and maximizing the acceptance ration of VN requests in order to increase the revenue of the InP.

3.2.2.2 Network modeling

In order to model the VN embedding problem**,** Mosharaf et al consider the network as weighted undirected graph with a set of substrate nodes and a substrate of links. Each node is characterized by its capacity in terms of the CPU and its location in the topology. On the other hand, each link has its capacity in terms of bandwidth. So the substrate network is modeled as shown in Figure 3.2 (a).

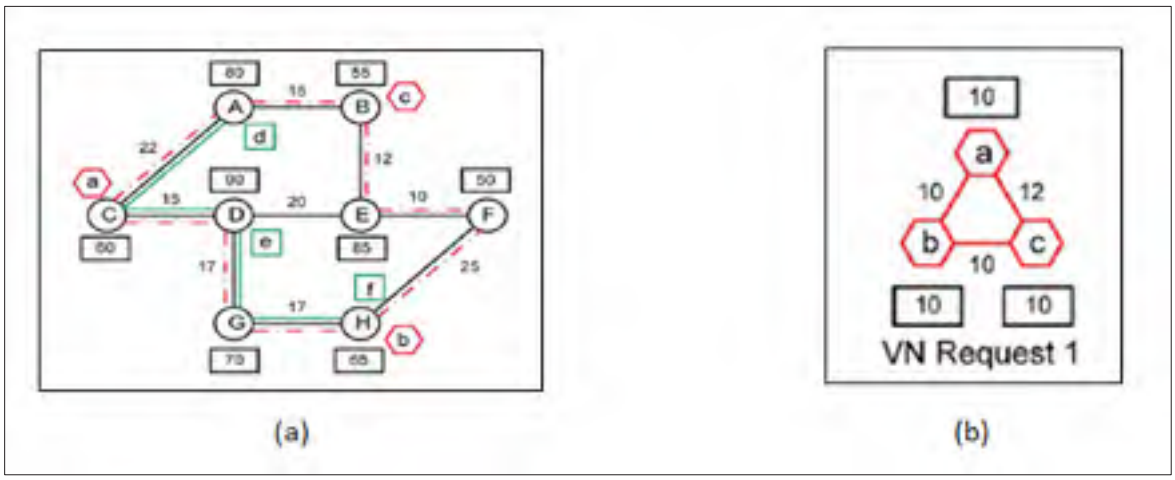


Figure 3.2 Modeling the substrate network (a), Example of VN request with node and link constraints (b)
Taken from Chowdhury, Rahman et Boutaba. (2012)

On the other hand, VN request is also modeled as a weighted undirected graph (Figure 3.2 (b)). Each VN request has its own requirements which are the following: preferred location, how far a virtual node can be embedded from its preferred location and required capacity in terms of BW and CPU (Chowdhury, Rahman et Boutaba, 2012). If there is a request for VN, the first step is to check if this request can be accepted based on its requirements in terms of BW and CPU and the existing capacities in the substrate network. In case of accepting the request, it is primordial, to select an assignment for this new VN that should be embedded. The procedure of assignment can be divided into two main components: node assignment and link assignment. The main objective of this work is to map multiple VN request while respecting virtual nodes and links requirements, increasing revenue for InP, decreasing costs for SPs and ensuring load balancing in the substrate network.

3.2.2.3 Problem formulation

The main contribution of this work is the correlation between nodes mapping and links mapping. In order to emphasize this correlation, Mosharaf et al propose to extend the substrate network to an augmented substrate graph while respecting location requirements of virtual nodes (Figure 3.3).

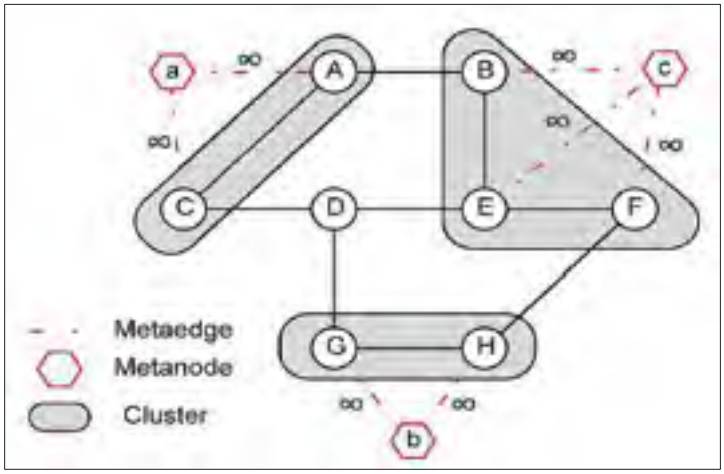


Figure 3.3 An augmented graph based on VN request
Taken from Chowdhury, Rahman et Boutaba (2012)

The objective function presented in this work to optimize VN embedding focus on the minimization of the total cost while ensuring load balancing under the following constraints: capacity constraints, Flow-related constraints, Meta and Binary constraints and finally domain constraints.

3.2.2.4 Solutions

The problem formulation for VN embedding is a Mixed Integer Programming (MIP) Formulation which leads to a NP-hard problem. That's why; this work proposes to relax the integer constraint related to the domain in order to have a linear program. After having a LP formulation, they use two approaches to compute an optimal solution:

- Deterministic technique (D-ViNE): in this approach they use deterministic rounding technique by introducing the function φ which is at the beginning set to zero for all unused nodes in the substrate network. If a virtual node is mapped into a substrate node the value of φ is set to 1 to avoid the use of this node in the next mapping procedure;
- Random technique (R-ViNE) which uses randomized rounding technique; once the values p_z are calculated as in D-ViNE, R-ViNE normalizes those values to restrict them within the 0 to 1 range where p_z presents the total flows routed through an edge z (Chowdhury, Rahman et Boutaba, 2012).

To improve their proposed solution, Mosharaf et al consider a realistic scenario when the VN requests can arrive at the same time. In this case, there is a need to use a queue by InP in order to optimize resource allocation. As a consequence, they propose a new approach based on the store of incoming VN requests for a waiting period until processing them. WiNE, which is the extended ViNEYard algorithm based on the look ahead capabilities, uses a window based batch processing of VN requests in order to queue incoming demands.

3.2.2.5 Advantages and limitations

ViNEYard algorithms are efficient in terms of embedding VN while optimizing resource utilization, increasing revenue for InP, decreasing costs for SPs and ensuring load balancing in the substrate network. The proposed algorithms: D-ViNE, R-ViNE and WiNE have a higher performance compared with existing works in both types of topologies: meshed topologies and hub and spoke topologies. But ViNEYard algorithms have the following limitations:

- ViNEYard algorithms are based on the resolution of two LP programs in order to optimize the embedding of VN in the substrate network. Solving two LP programs can generate a long execution delay due to the important number of variables and constraints which can affect the required performance;
- Using the window concept to store incoming VN requests is not an efficient decision in some worst-case examples because allowing an online algorithm to see the next VNs would not yield any advantage in the worst case since any sequence of n VN requests can be replaced by a new nk size sequence of VN requests, where each VN is replicated times (Chowdhury, Rahman et Boutaba, 2012);
- This work does not take into account the minimization of the energy consumption while solving the VN embedding problem. Moreover, it does not take into account networking specifications while embedding nodes and links in the substrate network.

3.3 SD-WAN

There are two approaches to build an SD-WAN network: Google's approach (Jain et al., 2013) which is to design the network from scratch as a SD-WAN network with OpenFlow switches and a centralized controller and the second approach is Microsoft's approach (Hong et al., 2013) which is to build SD-WAN on the top of the existing network. The second approach is more complicated because, in this case, there is a need to add network elements (NE) (brokers or aggregators) in the existing topology in order to interconnect the

infrastructure plane with the SDN control plane as shown in Figure 3.4. In our work, we adopt the second approach which aims to optimize resource allocation and to ensure efficiency, flexibility and scalability in Inter-DC WAN using existing resources (Gouveia et al., 2014) .

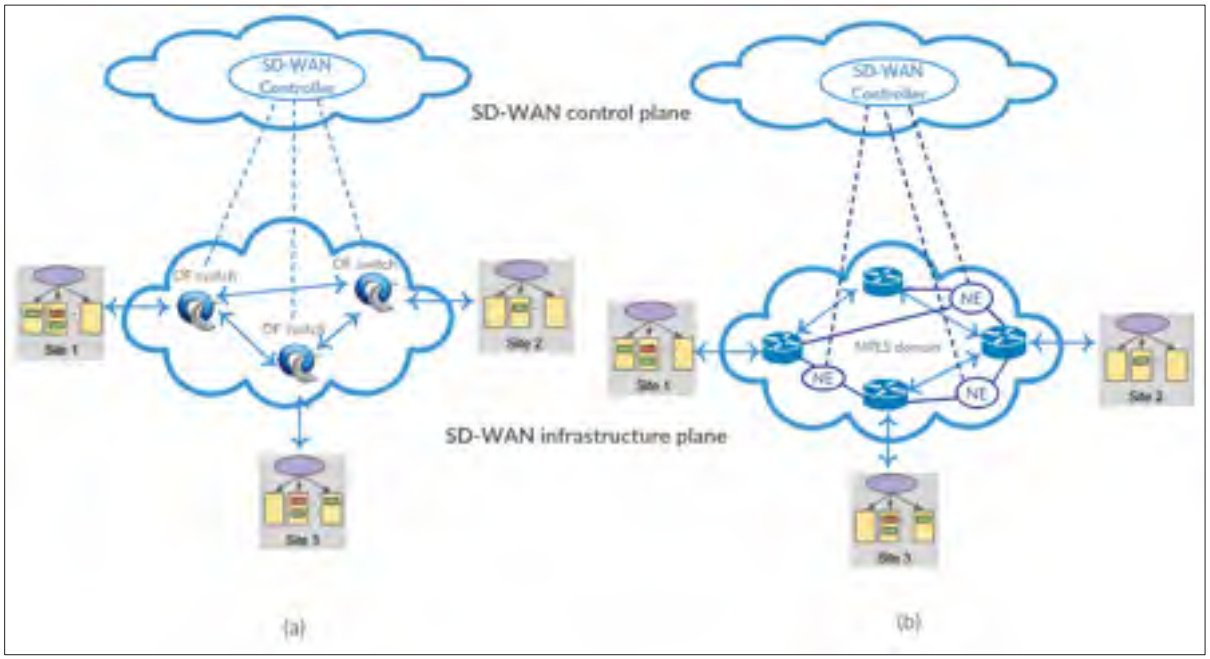


Figure 3.4 Google's approach (a) vs Microsoft's approach (b) to build an SD-WAN network

Now, we will present four works which are closely related to our project and share with us the same objectives. These solutions focus on the optimization of traffic engineering in inter-DC WAN using benefits of t SD-WAN, basically, using the centralized architecture proposed by this technology. We will study each proposed solution by discussing its advantages and its possible limitations in order to emphasize our contribution through this thesis.

3.3.1 Google's solution B4

3.3.1.1 Overview

B4 (Jain et al., 2013) presents the private WAN that interconnects Google's DCs across the planet using SDN technology. In fact, SD-WAN, with its unique specifications, has the potential to response to the following needs of Google's inter-DC WAN:

Huge bandwidth requirements: Google's inter-DC WAN carries applications that are characterized by massive bandwidth requirements and need to perform in large scale network. As a consequence, Google's inter-DC WAN must be flexible, efficient and elastic and must provide a high level of bandwidth average.

Moderate number of sites: Google's inter-DC WAN contains a few dozen sites across the world.

Full control of the network: actually, Google controls applications, servers and LANs in different parts of its WAN.

Cost sensitivity: in fact, Google wants to minimize the cost of their WAN using the centralized architecture provided by SDN technology.

B4 uses the centralized management and control functionalities provided by SDN architecture to build, with the minimum cost, a private inter-DC WAN that responses to Google's applications requirements in terms of bandwidth and QoS. B4 proposes new customized traffic engineering protocols based on SDN concept in order to enhance scalability of Google's Inter-DC WAN to meet applications requirements.

3.3.1.2 Architecture

B4's architecture contains three main layers: the global layer, the site controller layer and the switch hardware layer:

The global layer: contains logically centralized applications used for the centralized control and management of Inter-DC WAN.

The site controller layer: hosts the centralized site controllers NCSs which contain an OpenFlow controller (OFC) and some network applications used for control (NCAs).

The switch hardware layer: contains switches which just forward traffic based on rules provided by the centralized controller. In fact, in B4, Google uses their own customized switches to overtake limitations of existing hardware related to the high cost and the high complexity.

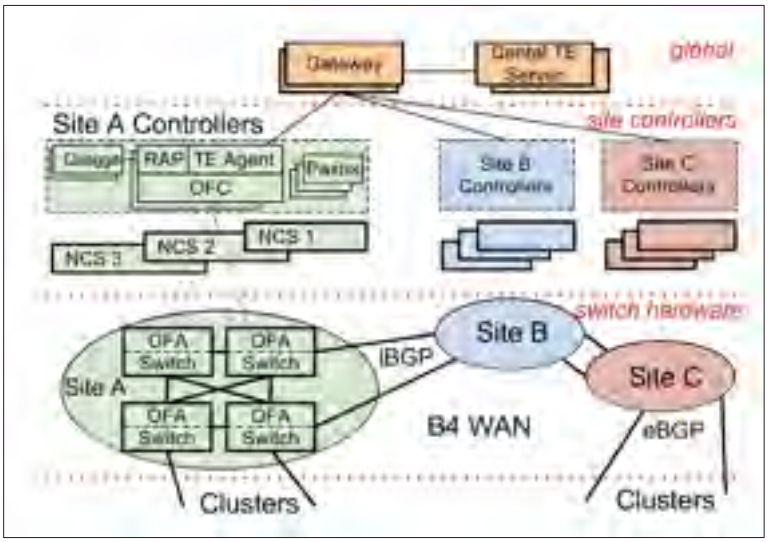


Figure 3.5 B4 architecture
Taken from Jain et al. (2013)

3.3.1.3 Traffic engineering in B4

B4 proposes a new TE optimization algorithm to share bandwidth between different applications using traffic splitting. So the main goals of this new optimization algorithm are: first to ensure the MMF while allocating bandwidth to applications belonging to different CoS and second to push links to use near to 100% of their available capacities.

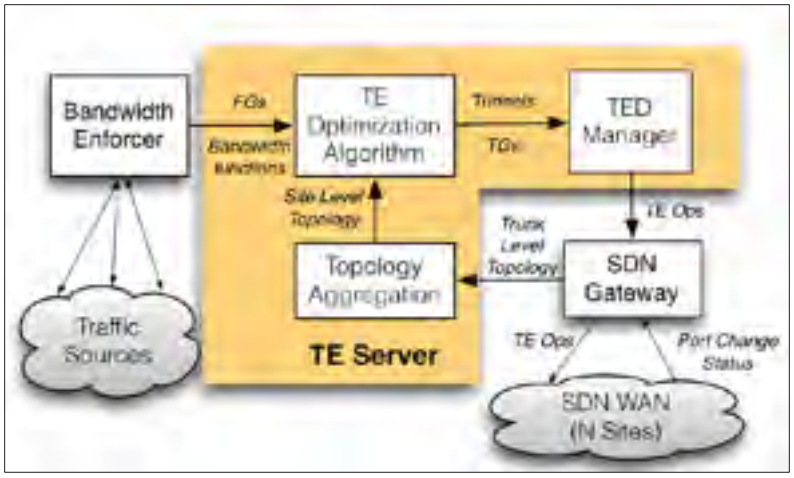


Figure 3.6 Traffic engineering architecture
Taken from Jain et al. (2013)

As shown in Figure 3.6, SDN Gateway collects topology events and current state of the network from SDN WAN and sends these events to the topology aggregator block in the TE server. Then, TE aggregator aggregates and sends received trunks to the TE optimization algorithm bloc. Based on the outputs of bandwidth function bloc, TE optimization bloc creates tunnels and group tunnels (TG) and sends it to TED manager to be used in order to create “TE op”.

Bandwidth function bloc is associated with each application to assign to it the required bandwidth based on the captured priority and the weight of the application. The output of this function will be used by the traffic engineering optimization algorithm to map flows into tunnels and tunnels into physical links while respecting existing physical capacities. In B4 deployment, in order to optimize resources allocation, bandwidth function computes bandwidth for a Flow Group (FG) and not for a single application or flow. This mapping problem is NP-hard, as a consequence, Sushant Jain et al propose a new algorithm which can achieve a similar fairness and a faster performance than the LP. The proposed algorithm performs with two main components:

Tunnel Group Generation: It is responsible for capacity allocation to FGs according to the output of their bandwidth function which is the required bandwidth for each FGs. Using this

approach can guarantee that all competing FGs will receive an equal fair share and can satisfy their demands. This bloc starts by iterating to find the bottleneck edge that minimizes fair share with its capacity, the main goal here is to increase fair share between FGs on their preferred tunnels. The algorithm stops when each FG is allocated its full demand or when the algorithm is unable to select a preferred tunnel for a FG or a preferred tunnel does not exist.

Tunnel Group Quantization: this component is responsible for adjusting splits based on the existing hardware capacities; this adjustment can be modeled as an optimization problem that can be solved using integer linear programming. A greedy approach is used in this case to determinate the optimal split quantization due to the complexity of the problem.

3.3.1.4 Advantages and limitations

The most important advantage of B4 is the improvement of link utilization; in fact, B4 makes the link use near to 100% of its available capacity thanks to the centralized TE algorithm. But B4 has some limitations which can be summarized in the following points:

First, the important latency that can be introduced in the network in case of topology changes when it is necessary to add or to remove some edge in the network or in case of failure.

Second, B4 has a problem of scalability in the path between OFC and OFA that must be fixed.

Third, B4 considers the WAN as a single domain and it does not take into account network specifications and underlying network characteristics.

And finally, B4 does not take into account the minimization of the energy consumption while allocating BW to different flows.

3.3.2 Microsoft's solution: SWAN

3.3.2.1 Overview

In the same context of B4, Microsoft proposes its own solution called SWAN (Hong et al., 2013) to interconnect their distributed data centers. This work presents a new model to maximize network utilization and to support flexible bandwidth sharing in Inter-DC WAN. It uses the centralized architecture of SDN to optimize resource allocation and bandwidth sharing and to maximize network utilization. SWAN supports a limited number of priority classes and allocates bandwidth based on service priorities while preferring the shortest path to the highest priority. SWAN classifies flows into three classes based on its delay sensitivities: interactive, elastic and background services.

SWAN focus on building new techniques and algorithms to maintain efficiency and high utilization in Inter-DC WAN using the centralized architecture of SDN. The main goal of SWAN is to find solutions to increase the efficiency in inter-DC WAN and to boost the average of link utilization to allow the network to use all its available capacity.

3.3.2.2 Architecture

As shown in the following figure, SWAN's architecture contains the following components:

Service hosts: the host OS estimates its demands for the next 10 seconds and sends it to the broker to demand an allocation.
Service broker: collects and aggregates demand received from different hosts and updates the centralized controller every 5 minutes. Then, the broker receives bandwidth allocation details from the controller every 10 seconds and sends it to hosts.
Network agent: is responsible for topology and traffic tracking. In fact, in case of topology change or a failure event this agent sends immediately these updates to the controller. Otherwise, it collects and reports information about the current state of the network every 5 minutes.

Controller: based on different information received from brokers and network agents, the controller computes the bandwidth allocations for services and sends forwarding rules to switches every 5 minutes.

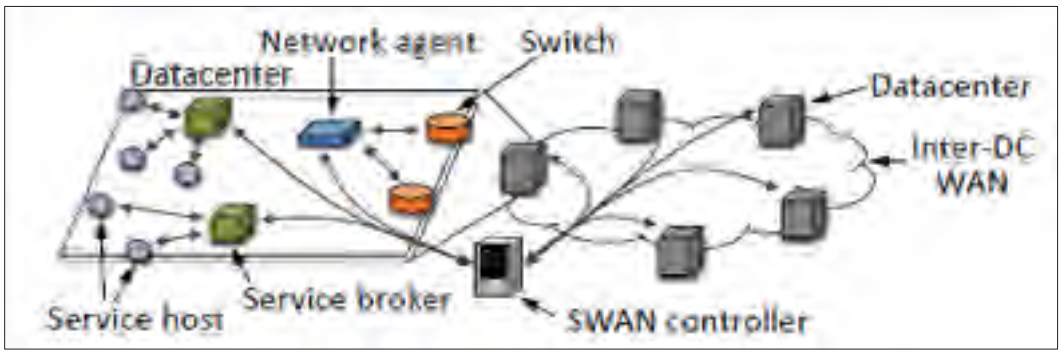


Figure 3.7 SWAN architecture
Taken from Hong et al. (2013)

3.3.2.3 Centralized bandwidth allocation

The controller is responsible for computing the allocated rate for services with the respect of the following two constraints: first, the controller must maximize the network utilization with the respect of services priorities, second, it must allocate rates using MMF for services in the same priority class. Running this problem in an unconstrained multi-commodity manner will lead to allocations that generate an important number of rules in switches which can violate its existing capacities. So it is necessary to take into account in the mathematical model available usable paths to simplify data plane updates. So to solve such a complicated problem LP seems to be the best nominee according to Microsoft. Figure 3.8 illustrates the inputs and the outputs of the bandwidth allocation model adopted in SWAN controllers.

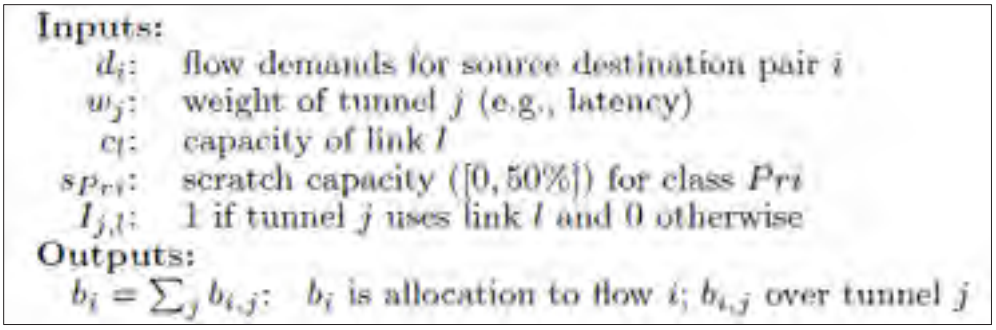


Figure 3.8 Inputs and outputs of SWAN bandwidth allocation model
Taken from Hong et al. (2013)

Figure 3.9 illustrates the algorithm used by SWAN for bandwidth sharing between different applications belonging to different CoS. SWAN uses MCF to model the bandwidth allocation problem. In fact, MCF maximizes the throughput and prefers in the same time the shortest path for the highest priority using LP. In the same class, SWAN allocates bandwidth using an approximated MMF by invoking MCF in T steps and in each step flows are allocated rates in a fix range that respects the scratch capacity. In this case, flows are frozen when it is allocated its full demand.

Figure 3.9 SWAN bandwidth allocation algorithm
Taken from Hong et al. (2013)

But, after executing this algorithm, there is another problem which is the possible complexity of the solution produced by the LP; this solution may be not feasible because of rules count limits on switches. In fact, the procedure of tunnels selection for traffic forwarding is an NP-complete problem. The proposed solution for this issue is to pick tunnels with the smallest latency between each data center pair and to use only these chosen tunnels as inputs of the LP problem, in this case the outputs are certainly implementable in the network.

3.3.2.3 Advantages and limitations

Definitely, SWAN enables a high efficiency and flexibility in inter-DC WAN by centralizing the bandwidth allocation. In fact, SWAN pushes the network to use near to 100% of its available bandwidth which is impossible with some traditional model like MPLS TE.

Moreover, SWAN allows the network to use is its available capacity during updates; in fact the network with SWAN continues to carry traffic and to maintain at the same time high network utilization during updates forwarding due to the concept of scratch capacity. But there are some limitations that can affect SWAN performance:

- Moreover, SWAN does not consider the communication inter-domain in their model and does not take into account under layering characteristics of the network in their mathematical formulation;
- And finally, SWAN does not take into account the minimization of energy consumption in their problem formulation.

3.3.3 SDN-based multi-class QoS-guaranteed inter-data center traffic management

3.3.3.1 Overview

To face the important growth of traffic and data exchanged in inter-DC WAN, service providers must think about improving the efficiency of their backbone network instead of over-provisioning it. To achieve this goal, there are many challenges that must be taken into account while thinking about new solutions for end-to-end bandwidth sharing which are:

- Limitation of existing hardware capacities in terms of the number of forwarding rules;
- Traffic classification: as SWAN, this work proposes to classify traffic into three main classes with different priorities: interactive, elastic and background;
- The proposed solution must be flexible and must respond to changes in the network and link/switch failures.

MCTEQ (Wang et al., 2014) distinguishes itself from the other proposed works such as B4 and SWAN by:

- MCTEQ allocates bandwidth to different traffic classes in a joint manner unlike SWAN which allocates bandwidth sequentially based on the traffic priority. To prioritize the high priority traffic in grabbing bandwidth, MCTEQ associates a large weight to their utility function;
- MCTEQ minimizes explicitly end-to-end delay for applications belong to interactive class by adding some delay constraints in the problem formulation;
- MCTEQ takes into account the constraint of delay which makes the problem NP-hard, this work proposes an efficient approximation to transform the problem into branch-and-bound using LP.

This work focus on allocating bandwidth to different traffic classes while ensuring the scalability, maximizing network utilization and responding to application requirements in terms of QoS, delay and fairness. The main objective of this work is to propose a new model for bandwidth sharing that takes into account traffic classification, traffic priorities and application requirements (QoS and delay) and allocates bandwidth in a joint manner unlike while ensuring fairness.

3.3.3.2 QoS for interactive application

MCTEQ takes into consideration explicitly the constraint of delay for interactive traffic. The end-to-end delay on a given path is characterized by the sum of the propagation delays for all the links and the queuing delays for all the nodes on the path (Wang et al., 2014). In fact, propagation delay is simple to compute, however, queuing delay is more complicated because it depends on various factors such as bandwidth allocated in each link and scheduling policies used in each link. That's why there is a need to bound the queuing delay for interactive flows.

This work assumes that the service discipline used on all devices in the network is guaranteed rate (GR) schedulers. Based on this assumption, the end-to-end delay of each packet can be bounded. To guarantee QoS requirement for interactive flows in terms of

delay, it is about ensuring that the end-to-end delay respects a tolerable delay bound. This constraint must be respected when a fraction of an interactive flow i use a specific path p. That's why it is primordial to use a Boolean parameter to indicate paths usage by different flows in the network.

3.3.3.3 MCTEQ formulation

The main goal of MCTEQ is to maximize the sum of utilities function of flows belonging to different classes in inter-DC WAN under the following constraints: required bandwidth, link capacity constraints and end-to-end delay constraints and requirements for interactive services. MCTEQ is NP-hard because it is non-convex due to the delay constraint which contains the bilinear term; it means that the proposed formulation presents an unfeasible problem. That's why an approximation is needed in this case.

3.3.3.4 MCTEQ solution

MCTEQ is non-convex problem which belongs to the NP-hard category due to the Log function used as a utility function in the problem formulation to allocate bandwidth in a joint manner. To solve this problem, this work suggests transforming the proposed problem into a Mixed Integer Problem (MIP) using branch-and-bound approach using LP. The presence of the Log function in the mathematical model imposes to use convex optimization solvers which can lead to a long runtime compared with LP solvers. As a consequence, this work proposes to linearize the Log function and to add three extra constraints related to delay and few continuous variables in the formulation to make it possible to solve the problem using LP instead of using convex optimization solvers. Using this approach guarantee the minimization of the computing time.

3.3.3.5 Advantages and limitations

The main contributions of MCTEQ can be summarized in the following points:

- The idea of using the joint bandwidth allocation approach to improve network utilization. This method helps to increase network throughput which responds to the need of SP;
- Considering explicitly end-to-end delay constraints in the problem formulation guarantees delay requirements for interactive flows.

However, limitations of this work are the following: it is true that this work maximizes network utilization and guarantees end-to-end delay but it allocates less bandwidth for interactive flows compared to SWAN. In fact, interactive flows have the highest priority compared to elastic and background traffics which are insensitive to delay. So it is useless to allocate more bandwidth for elastic and background traffic instead of interactive traffic to maximize network utilization. On the other hand, MCTEQ does not consider optimizing the energy consumption of nodes and links existing in the physical topology while allocating bandwidth to different incoming flows. Moreover, as SWAN and B4, MCTEQ ignores adding parameters to their problem formulation related to communication inter-domains and network specifications.

3.3.4 Delay-Optimized video traffic routing

3.3.4.1 Overview

Unlike prior works presented in this section, this proposed solution focus on optimizing flow latency based on the application's delay sensitivity. DOV (Liu, Niu et Li, 2016) distinguished itself from previous solutions such as B4 and SWAN in four main aspects:

- It focuses on reducing delays for video flows instead of focusing on throughput maximization;

- In addition, this work does not propose a fixed number of classes for services and applications;
- And finally, the solution proposed by this work is implemented in SDN application layer, unlike B4 and SWAN that are implemented in the network layer.

This work focus on selecting optimal paths for video flows in order to respond to their latency requirements and minimize their transmission delay while maximizing network throughput and resource utilization. The objective of DOV is to propose an algorithm which can generate optimal paths for video streaming flows by solving an optimization problem using LP.

3.3.3.2 Problem formulation

The objective function of this work focuses on the minimization of the delay while respecting network capacity and maximizing resource utilization. DOV considers that minimizing delay for video flows while optimizing resource allocation for different applications will lead automatically to maximize network throughput and resource utilization. The model used by DOV has mainly two constraints which are: session rate target must be equal to flow rates distributed to different trees and link capacities must be respected while allocating bandwidth to different flows.

This work proposes another alternative which aims to minimize tree depth instead of delay. Actually, this variation ensures that session with high delay-sensitivity and high priority will use trees with the minimum depth which will lead to trees with the lowest latency. The first proposed formulation needs some measurements about trees latency to achieve the goal of delay minimization unlike the second formulation which minimizes for each session the sum of all distinct tree depth values. To summarize both formulations have the same objective which is obviously routing delay sensitive flows with minimum delay. In fact, the proposed formulations are non-convex optimization problems because of the identity function that appears in the objective function. That's why there is a necessity to propose a heuristic to solve this optimization problem.

3.3.3.3 Proposed heuristic and implementation

To overtake the non-convexity of the two previous proposed problem formulations, this work proposes to use the Log-det heuristic which is usually used to solve matrix rank minimization. They propose to replace the identity function that appears in the problem formulation by a Log function and to minimize the linearization of this function like the previous work MCTEQ.

This work proposes to implement their solution in the application layer of SDN unlike the existing solutions such as B4 and SWAN. This decision leads to benefit from the advantages offered by SDN in terms of intelligent traffic engineering and global view of the network. In addition, implementing the whole system in the application layer makes its deployment very easy in inter-DC network of any SP without being forced to adjust the system based on their infrastructure characteristics. The proposed system is composed of two main components: the controller, which is responsible for computing the routing decision, and the forwarding device. Based on the received information about session link latency and link capacities from different forwarding devices, the controller runs the proposed algorithm to compute routes and rates assigned to different routes for each session based on its delay sensitivity.

3.3.3.4 Advantages and limitations

Main contributions of this work are the following:

- The consideration of delay as an objective instead of throughput maximization which prioritizes the high delay-sensitive applications and maximizes the network utilization at the same time;
- This solution is deplorable in SDN application layer which will make it easy to adopt it as a cloud provider without having any problem related to infrastructure restrictions;
- This solution improves the video delivery and the quality transfer in inter-DC networks, even when video flows are transmitted with other competing traffic.

But the most significant drawback which can be extracted from the observations is the fact that incoming sessions between two successive optimizations could be victims of suboptimal routing decisions that can affect their requirements in terms of QoS and delay. On the other hand, as prior works, DOV does not take into account the minimization of the total energy consumption while allocating bandwidth to different incoming flows. Moreover, the model used in this work ignores adding parameters related to communication inter-domain and network specifications.

3.4 Comparative study between existing solutions for DCs interconnection

In this section, we will summarize the methodology followed by each related work in order to highlight their contributions and their limitations. The following table will be used to emphasize our contributions based on different limitations of these related works.

As shown in Table 3.1, existing works are not energy-aware: they ignore the consideration of energy consumption parameters in their optimization model. On the other hand, they do not consider network specifications (MPLS, Ethernet, OTN, WDM, etc). And finally, all exposed solutions, presented in this chapter, consider that WAN contains one domain. This assumption is not realistic and can affect the performance of the proposed solution.

Table 3.1 Comparative study between related works

Solution	**Methodology**	**Contributions**	**Limitations**
DSPP	DSPP adopts virtual WAN in Inter-DC WAN DSPP minimizes the total cost under two main constraints: demand and SL DSPP is extended using game theory to consider the competition among multiple SPs.	Adjusting resource allocation based on the dynamicity of resource prices.	The proposed algorithm is very complicated and can generate a huge overhead in the network *DSPP does not take into account the minimization of the total energy consumption*
ViNEYard	ViNEYard proposes an efficient alternative to solve the VN embedding problem while ensuring an optimal share of existing resources between SPs. ViNEYard maximizes the acceptance ration of VN requests in order to increase the revenue of the InP.	ViNEYard algorithms optimize resource utilization, increase revenue for InP, decrease costs for SPs and ensure load balancing in the substrate network.	ViNEYard algorithms are based on the resolution of two LP programs which generates a long execution delay. *ViNEYard does not take into account the minimization of the energy consumption and network specifications while solving the VN embedding problem.*
B4	B4 adopts SD-WAN for DCs interconnection. B4 uses customized devices. B4 deploys a centralized traffic engineering server for bandwidth allocation.	Maximizing network utilization. Links can reach 100% utilization of their available capacity.	*B4 does not take into account network specification.* *(MPLS, OTN, WDM, etc).* *B4 does not consider the minimization of energy consumption.*
SWAN	SWAN uses LP to optimize traffic engineering. SWAN proposes a mechanism to update rules in the forwarding tables of switch SWAN proposes an approach to update the data plane without causing congestion.	Maximizing network utilization. Ensuring the required QoS.	SWAN does not take into account network specifications and energy consumption in the problem formulation. *SWAN does not take into account interconnection inter-domain*.

Solution	Methodology	Contributions	Limitations
DOV	Considering the minimization of delay as an objective function. Proposing solutions for some extreme case related to hungry flows	Prioritize flows with high delay sensitivity. This solution is deployed in the application layer of SDN which makes it easy to adopt.	Incoming sessions CAN be stuck in suboptimal routing decisions which can affect the required QoS. *DOV does not take into account network specifications, energy consumption and communication inter-domain in the problem formulation*.
MCTEQ	MCTEQ takes into consideration explicitly the delay constraints. MCTEQ shares bandwidth between different traffic classes in a joint manner.	Prioritize applications with high delay sensitivity. Maximize network utilization	*MCTEQ does not take into account network specification, interconnection inter-domain and energy consumption parameters in their model*

3.5 Conclusion

In this chapter, we presented six related works: four solutions based on SD-WAN which are very close to our work and two solutions based on network virtualization. We detailed the methodology used in each work and their different advantages. On the other hand, we studied their possible limitations in order to emphasize our contribution. Then, we exposed a comparative study which summarizes methodologies, contributions and limitations of these related works presented in this chapter.

CHAPTER 4

METHODOLOGY

4.1 Introduction

In this chapter, we present our methodology to achieve our objectives defined in chapter 1 (Section 1.4). At first, we introduce our throughput maximization model. Secondly, we integrate different models presented in the literature to calculate the energy consumption of different network components (nodes and links) and we propose our own formulation. Then, we present two mathematical models: a baseline model aimed at minimizing the total energy consumption while allocating bandwidth to different flows under constraints related to network specifications and inter-domain communication. The second model focus on both throughput maximization and energy consumption minimization while respecting constraints used in the previous model. Finally, we will introduce a deterministic algorithm proposed to solve the optimization problem using LP solvers. And we will present a greedy heuristic algorithm to improve the runtime of the proposed solution.

4.2 Throughput maximization modeling

Our objective is to build a model to optimize bandwidth allocation. The proposed model aims at maximizing throughput in the network in order to allow the existing infrastructure to carry more traffic with the same component capacities. Our goal is to define the virtual network V dynamically on top of the physical network P while ensuring throughput maximization and satisfying both application requirements and underlying constraints in terms of bandwidth demand and QoS. We allocate bandwidth to flows based on the demand $d_i, i \in N, N = \{nodes\}$ in each node and the remaining physical capacities in links and nodes in the physical MPLS topology at $T = t$.

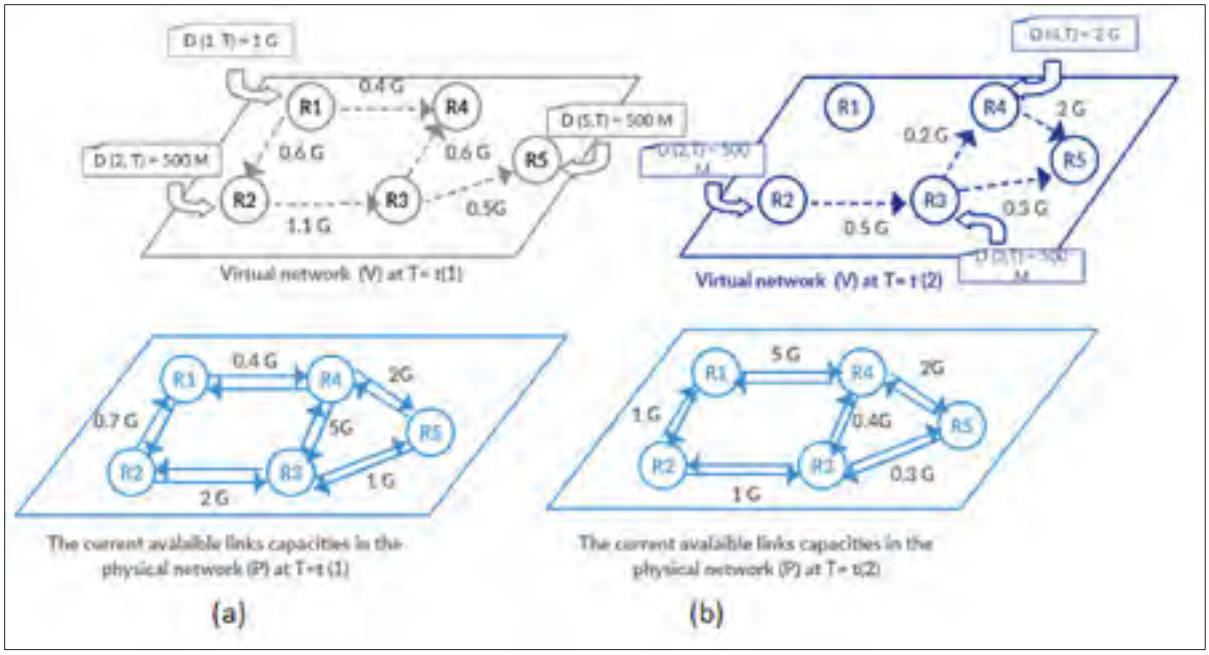


Figure 4.1 Building dynamically the virtual network on top of the physical network based on the required demand

Figure 4.1 (a) and (b) shows an example of an optimal mapping between the virtual network and the physical network while respecting constraints related to existing physical capacities and the current demand in each node and maximizing the throughput in the network. In fact, in Figure 4.1 (a), we have 3 demands in $R1, R2$ and $R5$, and there are multiple possible paths which can accommodate this traffic. For example, there is a demand of $1Gbps$ in the node $R1$ to be sent to $R4$ using the path which maximizes throughput (the direct path $R1 - R4$). However, this path has only $0.4\ Gbps$ capacity. So, the demand will be carried using two paths: $R1 - R4$ and $R1 - R2 - R3 - R4$.

The throughput maximization problem can be modeled using MCF (Multi Commodity Flow) approach which maximizes the overall throughput while preferring shorter paths for high priority services or flows. In fact, several important variants of the maximum flow problems involve multiple source-sink pairs(s_1, t_1)... (s_k, t_k) rather than just one source and one sink. Assuming that the traffic that the sources want to send to the sinks is of the same type, the problem is to find multiple feasible flows $f_1(.,.)$. . . $f_k(.,.)$, where $f_i(.,.)$ is a feasible flow

from the source s_i to the sink t_i, and such that the capacity constraints are satisfied (Cheng et al., 2012) (Trevisan, 2011):

$$\sum_{i=1}^{k} f_i(u,v) \leq c(u,v) \;\; \forall\, (u,v) \in E \tag{4.1}$$

Flows which satisfy the previous constraint are called Multi Commodity Flows. We propose to use this formulation to model throughput maximization in the network. We suggest the following objective function which aims to maximize throughput in the network while penalizing links with high latency w_j, $\quad j \in L\,, L = \{links\}$

$$max\ \sum_i b_i - \varepsilon\ \sum_{i,j} b_{i,j} w_j \qquad i,j\ \in N\,, N = \{nodes\} \tag{4.2}$$

Where:

$$b_i = \sum_j b_{i,j}\,, \qquad i,j\ \in N\,, N = \{nodes\} \tag{4.3}$$

Table 4.1 Symbols in the load balancing model

Symbols	**Description**
b_i	Bandwidth allocated to flow (Mbps).
$b_{i,j}$	Bandwidth allocated to flow i over the tunnel j (Mbps).
w_j	Latency of different links in the topology.
ε	A small constant.

4.3 Energy consumption modeling

In order to build an energy-aware solution, we model the energy consumption of links and nodes (router or switch) in the network. We consider four energy consumption models: fully proportional model, idle energy model, agnostic model and step increasing models (Addis et al., 2014; Bianzino et al., 2010; Ghazisaeedi, Huang et Yan, 2015; Zemmouri, Vakilinia et Cheriet, 2016).

4.3.1 The fully proportional model for energy consumption

In the fully proportional energy model (Addis et al., 2014; Bianzino et al., 2010) the energy consumption of a link or a node is a linear function $E = E^a x + E^0$ where E^a is the slope of the linear function, E^0 is the initial energy consumption of the link/node when its load is null, $a \in \{l\,(link), n(node)\}$, and x is the element usage (link/node usage in the network). In our case, the element usage is the amount of bandwidth allocated in this component. So, the total energy consumption will be modeled as follows:

$$\frac{1}{2}\left(\sum_{i,j \in L} \frac{b_{i,j}+b_{j,i}}{c^l_{i,j}}\, E^l_{i,j} + x_{i,j} E^o_{i,j}\right) + \left(\sum_{n \in N} \frac{v_n E^d_n}{c^d_n} + x_n E^0_n\right),\ N = \{nodes\},\ L = \{links\} \quad (4.5)$$

Where:

$$v^i_n = \sum_{i,n \in L} b_{i,n} + \sum_{n,i \in L} b_{n,i} \quad (4.6)$$

$x_{i,j \in N}$ indicates if the link from the node $i \in N$ to the node $j \in N$ is used to carry traffic or no. Similarly, $x_{n \in N}$ indicates if the node $n \in N$ is used to carry traffic in the network. In this model we assume that if a node/link is on and the bandwidth allocated in this component is null; it consumes a constant amount of energy which is equal to E_0.

(4.5) presents the energy consumption of links and nodes based on the current load in each component. To compute the total energy consumed by a link between the node i and the node j, we must consider the bandwidth allocated from the node i to node j which is $b_{i,j}$ and the bandwidth allocated from the node j to the node i which is $b_{j,i}$because we assume that the network is modeled as a directed graph. On the other hand, in order to model the total energy consumption of a node, it is primordial to consider the total amount of bandwidth allocated in a single node v^i_n. To meet that end, we must consider the bandwidth allocated to incoming traffic and bandwidth allocated to outgoing traffic.

Table 4.2 Symbols in the idle energy consumption model

Symbols	Description
v_n^i	Total bandwidth allocated to flow i in the device n (Mbps).
$c_{i,j}^l$	Capacity of the link between i and j (Mbps).
c_n^d	Capacity of the device n in terms of bandwidth (Mbps).
$E_{i,j}^o$	Initial energy consumption (energy consumption of a link when its load is null) of the link between i and j (Watts).
E_n^0	Initial energy consumption (the energy consumption of a device when its load is null of the device (Watts).
$E_{i,j}^l$	Slope of the energy function corresponding to the link between i and j (Watts).
E_n^d	Slope of the energy function corresponding to the device n (Watts).

4.3.2 The idle model for energy consumption

Unlike the fully proportional model in which the energy consumption of a component is a linear function $E = E^a x + E_0$. The idle model (Ghazisaeedi, Huang et Yan, 2015) (Bianzino et al., 2010) considers that the device does not consume energy when there is no load. So in the fully proportional model we have $E_0 = 0$. In this case we model the energy consumption of the network as follows:

$$\frac{1}{2}\left(\sum_{i,j \in L} \frac{b_{i,j}+b_{j,i}}{c_{i,j}^l} E_{i,j}^l\right) + \sum_{n \in N} \frac{v_n}{c_n^d} E_n^d,\ L = \{links\}, N = \{nodes\} \quad (4.7)$$

Where:

$$v_n^i = \sum_{i,n \in L} b_{i,n} + \sum_{n,i \in L} b_{n,i} \quad (4.8)$$

4.3.3 The agnostic model for energy consumption

The energy agnostic model (Bianzino et al., 2010) (Ghazisaeedi, Huang et Yan, 2015) considers that the energy consumption is constant for all nodes and links:

$$E = M \ \forall i \in L = \{links\}, \forall j \in N = \{nodes\} \quad (4.9)$$

In this model the energy consumption is independent of the amount of bandwidth used in each component. In our work, we assume that the energy consumption varies based on the

variation of the load in the network in order to achieve our objectives related to this thesis. So, we will not use this model in our problem formulation.

Figure 4.2 shows a conceptual presentation of the three energy consumption models: idle, fully proportional and agnostic on a network node which is similar to the models (4.5), (4.7) and (4.9).

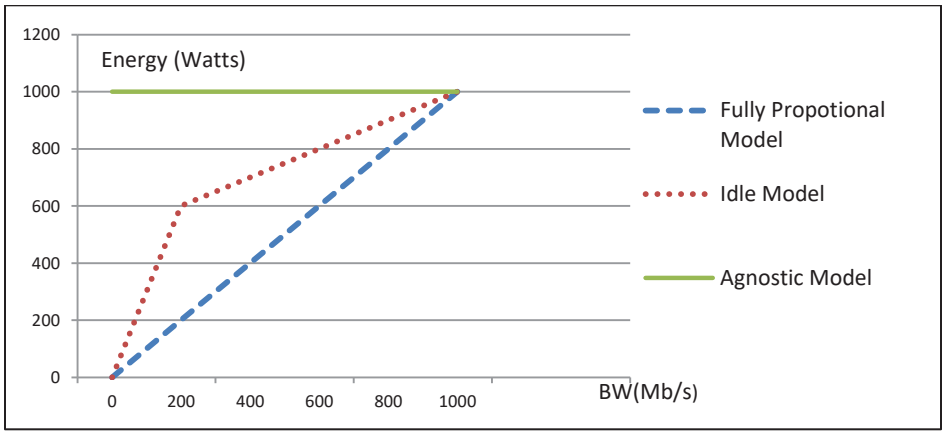


Figure 4.2 Graphical presentation of energy consumption models: idle, fully proportional and agnostic model

4.3.4 Discrete step increasing model for energy consumption

The discrete step-increasing model (Zemmouri, Vakilinia et Cheriet, 2016) for energy consumption takes into account traffic load in components (node/link). Based on this model, the energy consumption of links is presented as follows:

$$E_l(b_{i,j}) = \begin{cases} E_0^l \ if \quad 0 < b_{i,j} < c_0^l \qquad \forall i,j \in N \\ E_1^l \ if \quad c_0^l < b_{i,j} < c_1^l \\ E_2^l \ if \quad c_1^l < b_{i,j} < c_2^l \\ \vdots \\ E_{m-1}^l \ if \quad c_{m-2}^l < b_{i,j} < c_{m-1}^l \\ E_m^l \ if \quad c_{m-1}^l < b_{i,j} < c_m^l \end{cases} \tag{4.10}$$

Each link $l_{i,j}$ consumes a fixed amount of energy E_m^l , if the allocated bandwidth in this link $b_{i,j} \in [c_{m-1}^l, c_m^l]$, where c_k^l are constant $\forall k \in \{1, .., m\}$ and E_m^l are energy levels for a link. Same for nodes, the energy consumption is modeled as follows:

$$E_n(n_i) = \begin{cases} E_0^n \; if \quad 0 < v_n^i < c_0^n \qquad \forall i, n \in N \\ E_1^n \; if \quad c_0^n < v_n^i < c_1^n \\ E_2^n \; if \quad c_1^n < v_n^i < c_2^n \\ \qquad . \\ \qquad . \\ E_{m-1}^n \; if \quad c_{m-2}^n < v_n^i < c_{m-1}^n \\ E_m^n \quad if \quad c_{m-1}^n < v_n^i < c_m^n \end{cases} \tag{4.11}$$

Where:

$$v_n^i = \sum_{i,n \in L} b_{i,n} + \sum_{n,i \in L} b_{n,i} \tag{4.12}$$

Each node n_i consumes a fixed amount of energy E_m^n as shown in the conceptual presentation in Figure 4.3 if the incoming and the outgoing bandwidth in this node $v_n^i \in [c_{m-1}^n, c_m^n]$, where c_k^n are constant $\forall k \in \{1, .., m\}$ and E_m^n are energy levels for a node.

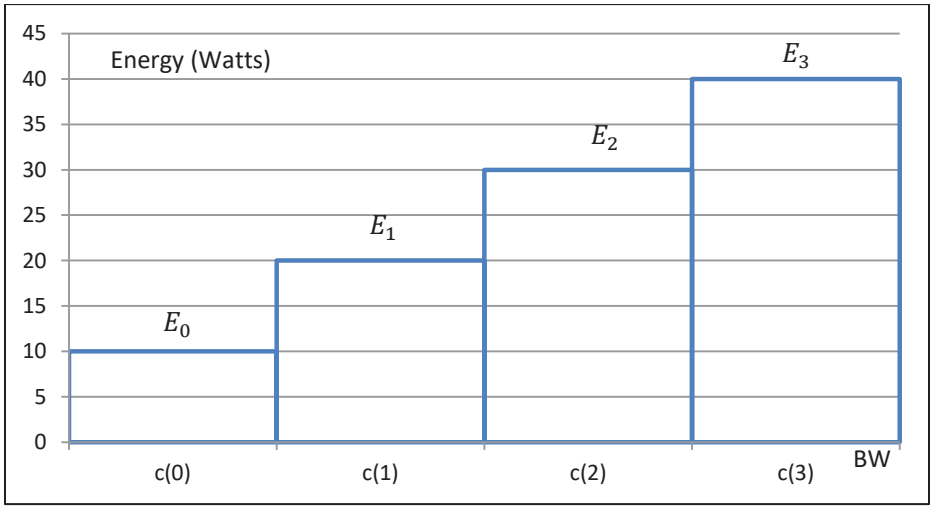


Figure 4.3 Discrete step increasing model for energy consumption of A node in the network

In our work, as a first step, we will adopt the idle energy model to mathematically formulate the energy consumption of links and nodes in the network because it is the most common model used in the literature. Then, we will use the step increasing model for energy consumption and we will compare the performance of these two alternatives in our proposed model.

4.4 MPLS network modeling

In order to model an MPLS network as shown in Figure 4.4, we use graph theory, in which the network is modeled as a directed graph G. $l_{i,j\in N}$, $N = \{nodes\}$ is the link between the two nodes $i \in N$ and $j \in N$ and . $l_{j,i\in N}$, $N = \{nodes\}$ is the link between the two nodes $j \in N$ and $i \in N$. The existing physical capacity between two nodes i and j is $c_{l\in L} = c_{i,j\in N} + c_{j,i\in N}$ where l is the link between the two nodes $i \in N$ and $\in N$. $L_{i\in N}^{in}$ is the number of the available labels for incoming traffic in the node $i \in N$ and $L_{i\in N}^{out}$ is the number of the available labels for outgoing traffic in the node $i \in N$.

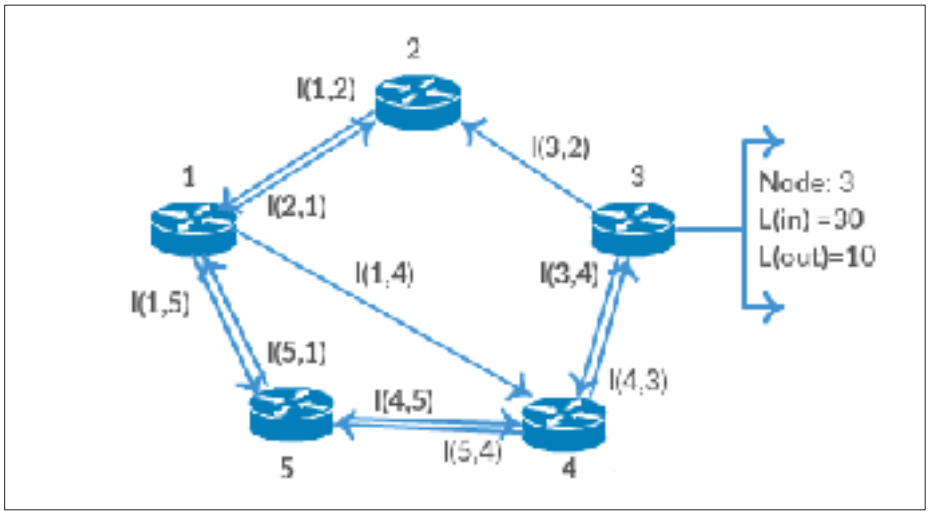


Figure 4.4 MPLS network modeling

We assume that Inter-DC WAN contains many MPLS domains with different specifications and we assume that ECMP multipath is not activated for MPLS-TE (which is commonly

used in reality). Our model takes into account underlying characteristics of MPLS domains, namely: number of available labels, switching capacities of nodes and maximum and minimum bandwidth available for allocation in links in each domain. These specifications can be expressed through the following constraints:

$$b_{low}^{domain^k} \leq b_i \quad \forall\, i \in D \;\; \forall\, k \in K \tag{4.13}$$

$$b_i \leq \min(d_i, b_{high}^{domain^k}) \;\forall\, i \in D \;\; \forall\, k \in K \tag{4.14}$$

$$\sum_{j \in D} \varphi(b_{i,j}) \leq L_{max}^{out}(i), \;\; \forall i \in D \tag{4.15}$$

$$\sum_{j \in D} \varphi(b_{j,i}) \leq L_{max}^{in}(i), \;\; \forall i \in D \tag{4.16}$$

$$v_j^i = \sum_{i \in D} b_{i,j} + \sum_{i \in D} b_{j,i} \;\; \leq sc_j \tag{4.17}$$

In our mathematical formulation, we use $\varphi(b_{i,j})$ which is a binary decision variable where:

$$\varphi(b_{i,j}) = \begin{Bmatrix} 0 \; if \; b_{i,j} = 0, \forall\, i,j \in D \\ 1 \; otherwise \end{Bmatrix} \tag{4.18}$$

Constraints (4.13) and (4.14) state that the bandwidth allocated to flow i in the domain k must respect capacity policies (maximum and minimum bandwidth that can be allocated to a single flow) related to this domain. Constraint (4.15) states that outgoing tunnels from node i in the domain k respect the maximum number of existing labels dedicated for outgoing traffic in this node. Similarly, the constraint (4.16) ensures that the number of incoming tunnels does not exceed the maximum number of labels dedicated for incoming traffic in the node i in the domain k. Constraint (4.17) guarantees that the incoming and the outgoing bandwidth in a single node in the topology do not exceed its switching capacity. The switching capacity depends on the number of ports in each node and the capacity of each port.

Table 4.3 Symbols in MPLS specifications modeling

Symbol	Description
$K = \{1,2, \dots\}$	Set of domains in the existing network.
$b_{low}^{domain^k}$	Minimum bandwidth that can be allocated to a flow i in the domain k.
$b_{high}^{domain^k}$	Maximum bandwidth that can be allocated to flow i in the domain k.
$d_i = \sum_{j \in D} d_{i,j}$	Demand in terms of bandwidth in each node i.
$d_{i,j}$	Demand in terms of bandwidth from the node i to the node j.
$L_{\max}^{in}(i)$	Maximum number of labels available for ongoing traffic in a node i.
$L_{\max}^{out}(i)$	Maximum number of labels available for outgoing traffic in a node i.
$sc_j = \sum_{m \in M^j} c_m^j \ n_m^j$	Switching capacity of the node $j \in D$.
$M^j = \{1,2, \dots\}$	Set of ports in the node $j \in D$.
c_m^j	Capacity of the port m in the node j.
n_m^j	Number of port m in the node j.

4.5 Optimization model

4.5.1 Objective function

As we said previously, in this work we propose two models: the first is an energy aware model which takes into account the minimization of the total energy consumption and the second model ensures throughput maximization and energy consumption minimization. Therefore, the network is a directed graph G. The main objective of these proposed models is to find $b_{i,j \in N}, N = \{nodes\}$ which maximizes throughput and minimizes the total energy consumption of different links and nodes in the network. This problem is comparable to the virtual network embedding problems (Nonde, El-Gorashi et Elmirghani, 2015; Su et al., 2014) ; however, it takes into account different energy consumption models, in particular, discrete ones, and the underlying specifications like MPLS.

4.5.1.1 Base line objective function

Our goal through the following objective function is to minimize the energy consumption of the virtual network which should be built on the top of the existing physical network; in this formulation we use the fully proportional model for energy consumption:

$$Min\ \ \frac{1}{2}\left(\sum_{i,j\in D} x_{i,j}\left(\frac{(b_{i,j}+b_{j,i})}{c_{i,j}^{l}}\ E_{i,j}^{l}+\ E_{i,j}^{o}\right)\right)+\ \sum_{j\in D} x_j\left(\frac{v_j^i E_j^d}{c_j^d}+E_j^0\right) \tag{4.19}$$

In the previous objective function, we used the following decision variables: $\varphi(b_{i,j})$ is a binary variable where $\varphi(b_{i,j})=0$ if $b_{i,j}=0$ and $\varphi(b_{i,j})=1$ otherwise, $\forall\, i,j\ \in N$. Values of $x_{i,j}$ and x_i can be computed based on values of $\varphi(b_{i,j})$ as follows:

$$x_{i,j\in D}=\begin{Bmatrix}0\ if\ \varphi(b_{i,j})=0\ and\ \varphi(b_{j,i})=0,\ \ \forall\, i,j\ \in D\\ 1\ otherwise\end{Bmatrix} \tag{4.20}$$

$$x_{j\in D}=\begin{Bmatrix}0\ if\ \ \sum_{i\in D}\varphi(b_{i,j})+\ \sum_{i\in D}\varphi(b_{j,i})=0,\forall\, i,j\ \in D\\ 1\ otherwise\end{Bmatrix} \tag{4.21}$$

$x_{i,j}$ indicates if a link is used by the flow i or not, on the other hand, the variable x_j indicates if a node is used to route the flow i or not.

4.5.1.2 Throughput maximization and energy aware objective function

In this objective function, we take into account both throughput maximization and the minimization of the energy consumption; in fact, our goal is to maximize resource utilization to push links and nodes in WAN to use 100% of their available capacity which is very important in the context of DCs interconnection where resources are very expensive. In order to optimize resource utilization and allocation while minimizing the total cost, which is in our case: total energy consumption and bandwidth utilization, we propose the following objective function:

$$Min\ (E(b^*_{i,j})), \forall i,j\ \in L\ where\ b^*_{i,j} = argmax\ (B(b_{i,j})) \quad (4.22)$$

Where:

$$E(b_{i,j}) = \frac{1}{2}\left(\sum_{i,j\in D} x_{i,j}(\frac{(b_{i,j}+b_{j,i})}{c^l_{i,j}}\ E^l_{i,j} + E^o_{i,j})\right) + \sum_{j\in D} x_j\left(\frac{v^i_j E^d_j}{c^d_j} + E^0_j\right) \quad (4.23)$$

$$B(b_{i,j}) = \sum_{i\in L} b_i - \varepsilon \sum_{i,j\in D} b_{i,j}\ w_j \quad (4.24)$$

$$v^i_j = \sum_{i\in D} b_{i,j} + \sum_{i\in D} b_{j,i} \quad (4.25)$$

The main objective function (4.22) maximizes throughput of the virtual network that should be over layered on top of the existing physical network (4.24). Then, $Min\ (E(b^*_{i,j}))$ checks if this solution is energy aware by using the fully proportional model (4.23) to compute the total energy consumption. To meet that end, we need to use (4.25) to quantify the total amount of incoming and outgoing traffic in each node in order to be able to compute the total energy consumed by nodes and links in Inter-DC WAN.

4.5.2 Constraints

To improve our optimization model, we propose to add the following constraints which are related to: networking policies, networking specifications and communication inter-domain. Unlike prior works which consider only two generic layers, virtual and physical, our model takes into account specific characteristics of underlying layers.

$$b^{domain^k}_{low} \leq b_i \quad \forall\ i\ \in D\ \forall\ k \in\ K \quad (4.26)$$

$$b_i\ \leq \min(d_i, b^{domain^k}_{high})\ \forall\ i\ \in D\ \forall\ k \in\ K \quad (4.27)$$

$$\forall\ i\ \in D\quad b_i = f_i \quad (4.28)$$

$$\forall\, l\ \sum_{i,j} b_{i,j}\ I_{j,l}\ \leq \min\left(c_l^{remain}, (1 - S_{pri})c_l\right) \tag{4.29}$$

$$\sum_{j \in D} \varphi\left(b_{i,j}\right) \leq L_{max}^{out}(i)\,,\ \forall i\ \in D \tag{4.30}$$

$$\sum_{j \in D} \varphi\left(b_{j,i}\right) \leq L_{max}^{in}(i),\ \ \forall i \in D \tag{4.31}$$

$$v_j^i = \sum_{i\,\in D} b_{i,j} + \sum_{i \in D} b_{j,i}\ \ \leq sc_j \tag{4.32}$$

$$\sum_{j \in D} b_{i,j} - \sum_{j \in D} b_{j,i} = \left\{ \begin{array}{c} d_{i,j}\ if\ i = s(source) and\ j = t(sink) \\ -d_{i,j}\ if\ i = t(sink)\ and\ j = s(source) \\ o\ otherwise \end{array} \right\} \tag{4.33}$$

Table 4.4 Symbols in our proposed model

Symbol	**Description**
c_l^{remain}	Remaining link capacity in the network.
c_l	Existing physical capacity of links in the network.
$S_{pri} \in [0, 50\%]$	Scratch capacity reserved to carry tunnels updates in order to avoid congestion while updating forwarding tables of different devices in the network. If there is not update to carry, this capacity will be used to carry background traffic.
$x_{i,j}$	indicates if a link is used by the flow i or not.
x_j	indicates if a node is used to route the flow i or not.
$b_{i,j}\ I_{j,l}$	Path used to carry the flow i through different links l ∈ L.
f_i	Flow i.

Constraint (4.28) guarantees that the bandwidth allocated to the flow i will respond to demand in the node i. Constraint (4.29) checks whether the bandwidth allocated to the flow i must respect the existing capacity in the physical network. On the other hand, we want to send flows from a specified node "s" in the network, called the source, to another specified node "t", called the sink while ensuring the load balancing and minimizing the total energy consumption. Constraint (4.33) ensures that the bandwidth allocated from the node i (source) to the node j (sink) must be equal to the required demand from i to j.

The optimization model is a Mixed Integer Linear Programming problem (MILP) and it can be reduced to the problem of virtual network embedding which is an NP-hard problem (Cai et al., 2010) (Chowdhury, Rahman et Boutaba, 2012). As an NP-hard problem, the execution delay of the proposed problem increases exponentially with the size of the existing physical topology used as input to the proposed model. To solve this model; first, we propose an algorithm using LP solvers. Then, we propose a heuristic, which is based on greedy approach, to avoid long computing time in case of dense topologies.

4.6 Deterministic solution

To find the optimal solution which fulfills to our requirements related to the maximization of the throughput and the minimization of the total energy consumption while respecting constraints listed in the previous section we propose the deterministic algorithm (Algorithm 4.1). The output of the proposed algorithm is a capacity matrix of the virtual network that should be over layered on the top of the existing physical network in order to respond to the demand in each node. By optimizing the definition of this virtual network we optimize the design of an energy-aware virtual Inter-DC WAN while taking into account networking policies and communication Inter-domain.

Algorithm 4.1 allocates bandwidth to different flows belonging to different CoS (interactive, elastic, background) while minimizing the total energy consumption. In line 2 (main procedure), it checks if the incoming flows belong to the same CoS. If the flows have different CoS, throughput maximization function will be executed in line 3 (main procedure) to maximize bandwidth allocation for the highest priority flows. Otherwise, an approximate MMF function in line 5 (main procedure) calls T times the MCF function and in each call flow bandwidth are assigned with a value between B_low and B_high. Both throughput maximization and MMF functions are based on the resolution of the proposed optimization problem which is done in MCF function. In this function, we use an LP solver to solve our optimization problem.

To find a solution with the minimum energy consumption, Algorithm 4.1 calls the throughput maximization function or the approximate MMF function M times (line 1 in the main procedure). M is defined in Figure 4.5 .So, Algorithm 4.1 efficiently solves the proposed problem using benefits of LP solvers and produces M solutions to respond to the same demand under the same constraints; it selects the solution with the minimum total energy consumption.

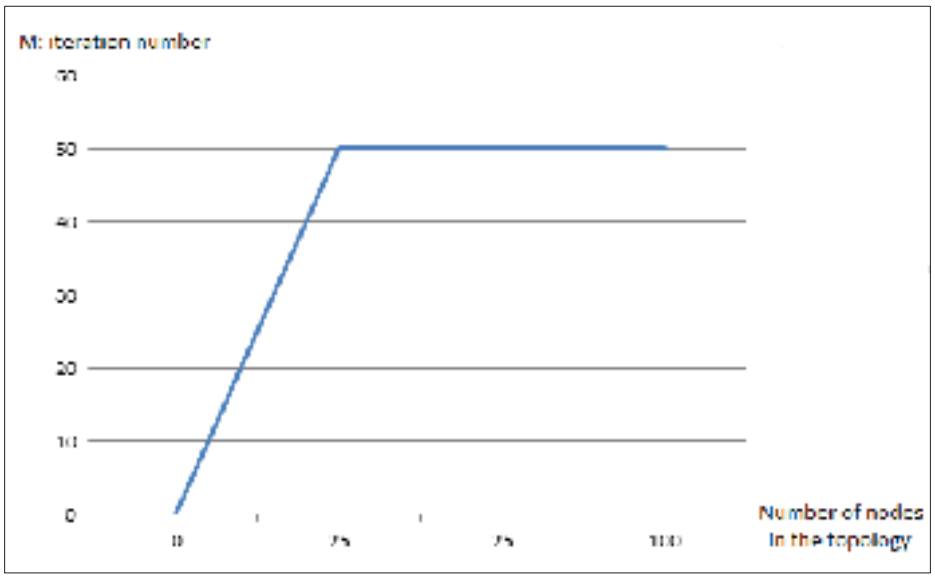


Figure 4.5 Iteration number M based on the number of nodes

After running Algorithm 4.1, M matrix are generated to present the virtual network that can be over-layered on top of the existing physical network and respond to the current demand in each node while ensuring throughput maximization. Then, the total energy consumption is computed for each solution to determine the solution with minimum energy to carry the required traffic while respecting the constraints. The value of M is defined as follows: $M = 2 * N$, N is the number of nodes in the topology, if $N > 50$, $M = 50$. In fact, the value of M must be reviewed in order to avoid having a long running time delay which will may affect the performance of the proposed solution.

Algorithm 4.1 Deterministic algorithm

Algorithm

Inputs:

N // *number of nodes in the topology*
Phy_capacity_matrix // *matrix capacity for physical topology*
M //*iteration number*
D // *demand in each node*

Outputs:

$b_i = \sum_{j \in N} b_{i,j}$ // *Bandwidth allocation to flow i over tunnel j*

Main procedure:

```
1: While (iteration_number < M)
2:    if (flows in different CoS)
3:       B = Throughput_Max(Phy_capacity_matrix, N, D, B_low,    B_high)
4:    else
5:       B = Max_Min_fairness (Phy_capacity_matrix, N, D)
6:    end
7:    Compute_ energy_consumption(B, N)
8:    Iiteration_number = iteration_number +1
9: end
10:  Select_solution_min_energy ();
11:  B= B_energy_min
12: end
```

Function: Throughput_Max(Phy_capacity_matrix, N, D, , B_low, B_high)

1: B= MCF(Phy_capacity_matrix, N, D, B_low, B_high)
2: return B
3: end

Function: Max_Min_fairness (Phy_capacity_matrix, N, D)

1: $T = \log(D/U)/\log(\alpha)$
2: For k=1:T
3: B_low= $U\alpha^{k-1}$
4: B_high= $U\alpha^{k}$
5: B = MCF (Phy_capacity_matrix, N, D, , B_low, B_high)
6: return B
7: end
8: end

Function: MCF (Phy_capacity_matrix ,N, D, B_low, B_high)

1: Solve (P)
2: P = { $\max \sum_i b_i - \varepsilon \sum_{i,j} b_{i,j} * w_j$
3: $B\_low \leq b_i \leq \min(d_i, B\_high) \quad \forall i \in D \ \forall k \in K$
4: $\forall i \in D \quad b_i = f_i$
5: $\forall l \ \sum_{i,j} b_{i,j} * I_{j,l} \leq \min\left(Phy\_capacity\_matrix, (1 - S_{pri})c_l\right)$
6 : $\sum_{j \in D} \varphi(b_{i,j}) \leq L_{max}^{out}(i), \ \forall i \in D$
7: $\sum_{j \in D} \varphi(b_{j,i}) \leq L_{max}^{in}(i), \ \forall i \in D$
8: $v_j^i = \sum_{I \in D} b_{I,j} + \sum_{i \in D} b_{j,I} \leq sc_j$
9: : $\sum_{j \in D} b_{i,j} - \sum_{j \in D} b_{j,i} = \left\{ \begin{matrix} d_{i,j} \text{ if } i = s \text{ and } j = t \\ -d_{i,j} \text{ if } i = t \text{ and } j = s \\ o \text{ otherwise} \end{matrix} \right\}$ }
10: end

4.7 Greedy heuristic

In order to improve the proposed solution, we propose a heuristic to avoid the use of MCF function. In fact, MCF uses LP solvers which can generate a huge execution delay if the number of nodes and links in the existing physical topology is dense and the real time demand is huge. We suggest replacing MCF function by another alternative based on a greedy approach (Kodaganallur et Sen, 2010; Ouerfelli et Bouziri, 2011).

In order to improve the runtime of the proposed solution, we propose a heuristic to avoid the use of MCF function when the input topology is dense, resulting in a large number of variables. We replace MCF function by a greedy approach. In the greedy heuristic presented as the algorithm 4.2 (Ghosal et Ghosh, 2014; Ouerfelli et Bouziri, 2011), first, we sort increasingly nodes based on their energy consumption. Algorithm 4.2 starts with the node which consumes the smallest amount of energy; $d_i, i \in N^s, N^s = \{sorted\ nodes\}$ the demand in the latter node is satsified by selecting paths which can accommodate the required traffic with the minimum cost (energy consumption) while respecting the underlying specifications and constraints related to inter-domain communication. The procedure of path selection is based on a greedy approach too. In fact, the proposed heuristic selects the first available path which consumes the minimum energy, able to accommodate the incoming traffic and satisfies constraints presented in our optimization model using BFS algorithm (Bücker et Sohr, 2014; Cui et al., 2012; Ueno et al., 2016). BFS algorithm is able to build a path from the required source to the required destination by visiting first the nearest nodes in the existing network.

Algorithm 4.2 is able to produce a solution meeting our requirements in terms of the minimization of energy consumption and the maximization of throughput in the network with a short runtime. The complexity of the proposed greedy heuristic is $o(N * L)$, where N is the number of nodes in the existing topology and L is the number of links.

Algorithm 4.2 Greedy heuristic

Greedy heuristic

Inputs:

N = {nodes}

L = {links}

c_l^{remain} // remaining link capacity

$D = \{d_i, i \in N\}$

Outputs:

$B = \{b_{i,j}, \ i,j \in N\}$

Greedy function:

1: List N^S = nodes(N)
2: For n in N^S
3: [Path^j, $b_{n,j}$] = path (n, N, L, d_i , c_l^{remain})
4: end

Function: node (N)

1: List node = compute_cost_node (N)
2: List N^S = sort_node (node)
3: return N^S
4: end

Function: path (n, N, L, d_i , c_l^{remain})

1: List *Path* = BFS (n, L,N , c_l^{remain})
2: Boolean $Stop \leftarrow False$
3: Boolean $Check \leftarrow True$
4: While(($Stop == False$) && (j in $Path$) &&($Check == True$))
5: if ($b_{n,j} < d_i$)
6: $d_i \leftarrow d_i - b_{n,j}$
7: $Path^n = [Path^n \ j]$
8: *Check* = check ($Path^n$)
9: $c_l^{remain} \leftarrow c_l^{remain} - j$
10: $B \leftarrow B + b_{n,j}$
11: $Stop \leftarrow False$
12: $j \leftarrow j + 1$
13: end
14: if ($b_{n,j} \leq d_i$)
15: $Stop \leftarrow True$
16: $Path^n = [Path^n \ j]$
17: *Check* = check ($Path^n$)
18: $c_l^{remain} \leftarrow c_l^{remain} - j$
19: $B \leftarrow B + b_{n,j}$
20: end
21: end
22: return $Path^n$, $b_{n,j}$
23: end

4.8 Conclusion

In this chapter, we presented our mathematical model to optimize resource allocation and resource utilization in Inter-DC WAN in order to maximize throughput in the network and to minimize the total cost in terms of energy consumption. We propose an algorithm which uses LP solvers in order to guarantee the required performance in terms of execution delay and optimization of resource allocation. On the other hand, we propose a heuristic to be used in case of dense topologies in order to avoid solving long equations using LP. In the final chapter, we will implement our proposed solution (our algorithm and our heuristic) and we will evaluate the performance of our work using some specific parameters. Moreover, we will compare the performance of our solution with the performance of other existing and related works which share with us the same main objectives.

CHAPTER 5

RESULTS AND VALIDATION

5.1 Introduction

In this chapter, we present the implementation of the proposed algorithm and heuristic. Then, we evaluate our proposed solution based on two criteria which are total energy consumption and load balancing. In addition, we compare the performance of our proposed solution with two other works which are: Microsoft's solution SWAN and a baseline energy-aware model which aims to minimize the total energy consumption without taking into account throughput maximization. The baseline model has been presented in the section 4.5.1.1. Finally, based on results obtained during the validation process, we highlight advantages of our solution.

5.2 Validation protocol

In reality, our solution will be deployed as a module in the SDN's application layer in order to ensure its portability and its generality. To validate the proposed deterministic algorithm and the heuristic, we implemented our solution and other algorithms under study (SWAN and baseline energy-aware model) using Matlab. To test the performance of our algorithm, we used different topologies as input: SWAN (Hong et al., 2013), NSF(Galán, 2008) and a variety of randomly generated topologies. The output of our solution is a matrix which presents the virtual network that should be over layered on top of the existing physical topology, satisfying users demand in different nodes in WAN and respecting the required QoS. A SDN controller will use this matrix to create rules and insert it into switches in the infrastructure layer.

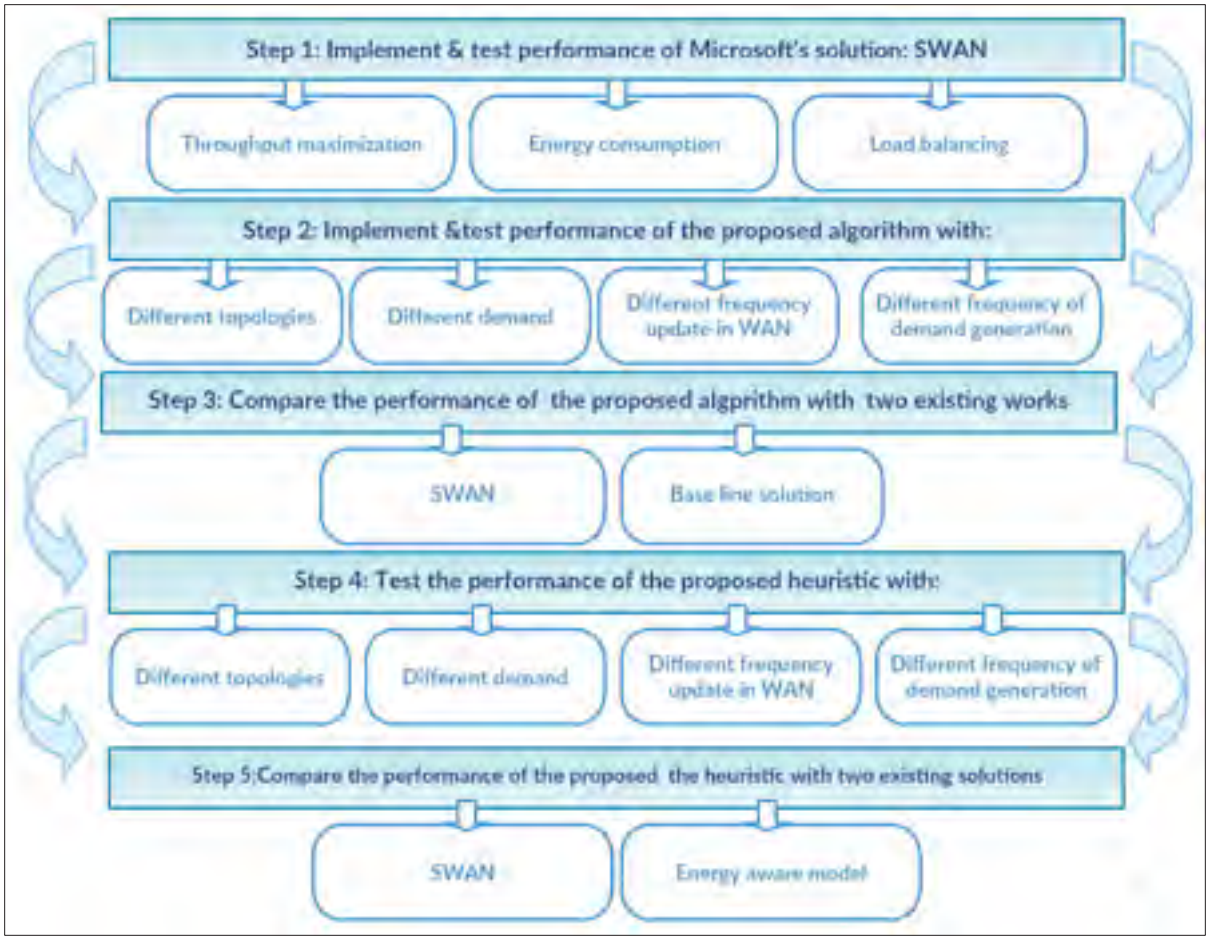


Figure 5.1 Validation protocol

Our validation protocol contains the following steps (Figure 5.1):

- Step 1: re-implement Microsoft's SWAN solution and test its performance in terms of: load balancing and energy consumption in the simulation environment (Matlab) with random data;
- Step 2: implement and test our algorithm with different topologies as input and with each topology, test the performance of the proposed algorithm with different values of demand and different physical capacities of different components (node and links) existing in the topology;

- Step 3: compare the performance in terms of energy consumption and load balancing of our algorithm with the performance of two existing works: SWAN and the baseline energy-aware solution;
- Step 4: implement and test the proposed heuristic using different topologies as inputs and different demands in each node of the physical network;
- Step 5: compare the performance in terms of energy consumption and runtime of the proposed heuristic with the deterministic algorithm and existing works.

Finally, it is important to present some assumptions made for the implementation. In this work, we are not considering turning on/off or putting nodes or links on a sleep mode in order to minimize the total energy consumption. We are focusing on the optimization of the built of a virtual network on top of a physical network based on existing physical capacity of network components and the current demand. We assume that the management of the state of links or nodes which do not appear in the virtual network that should be over layered on top of the existing physical network is considered as another problem. Moreover, in extreme cases like hungry flows which are characterized with a demand that exceeds the existing physical capacity, we assume using MMF approach (Hong et al., 2013) to allocate bandwidth to different flows while ensuring fairness and respecting the existing resource capacity. Finally, to respond to the application's requirements, we adopt the classification of flows into three main classes: interactive, elastic and background. For flows in the same class we use MMF approach otherwise, we use throughput maximization approach.

5.3 Implementation and test

5.3.1 Test with SWAN topology

First, we use SWAN topology, which is used by Microsoft to evaluate their solution (Figure 5.2), as input to our algorithm and we assume that SWAN topology contains two different domains MPLS with different specifications in terms of available labels in different nodes and maximum switching capacity. Moreover, we assume that physical nodes and links which

are not allocated to the virtual topology may be put off or in sleep mode. We compare the performance of our proposed solution with the performance of two other solutions: SWAN and a baseline solution which minimizes the total energy consumption without taking into account any network specifications or inter-domain communication constraints. We evaluate these different solutions based on two main criteria: load balancing and total energy consumption. The demand is generated randomly each $T_d = 1$ minute and the update frequency of the network is set to $T_f = 1$ minute in order to stress our system and to test the response of our algorithm under such an active Inter-DC WAN. For the energy consumption of nodes (routers and switches), we use the numerical model proposed by (Vishwanath et al., 2014) and for energy consumption of links we adopted the numerical model proposed by (Bianzino et al., 2010).

Figure 5.2 SWAN test topology
Taken from Hong et al. (2013)

Figure 5.3 shows a comparative study between the outputs of our algorithm, SWAN and a base line model for 10 iterations. Each $T_d = 1$ minute, we have a new random demand generated in each node in the topology. We execute the three algorithms corresponding to the three models listed previously and we compute the total energy consumption of each solution proposed by each algorithm. Then, we present the total energy consumption of the final solution of each algorithm during 10 iterations. Similarly, in Figure 5.4, we use the same procedure but with a 20 iterations. As shown in Figure 5.3 and Figure 5.4, our solution achieves the best performance in terms of selecting the virtual network with the minimum

total energy consumption while respecting different constraints related to network specifications and communication inter domain.

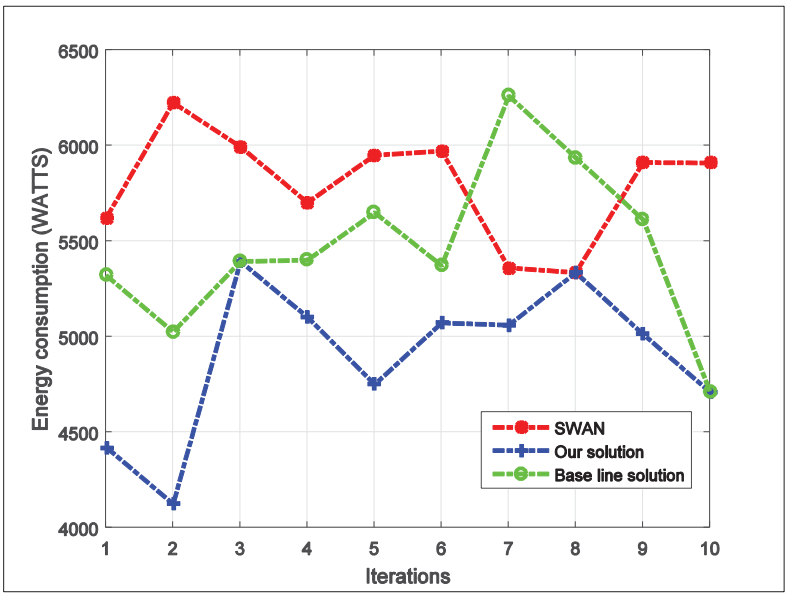


Figure 5.3 Total energy consumption during 10 iterations (SWAN vs our solution vs a base line solution)

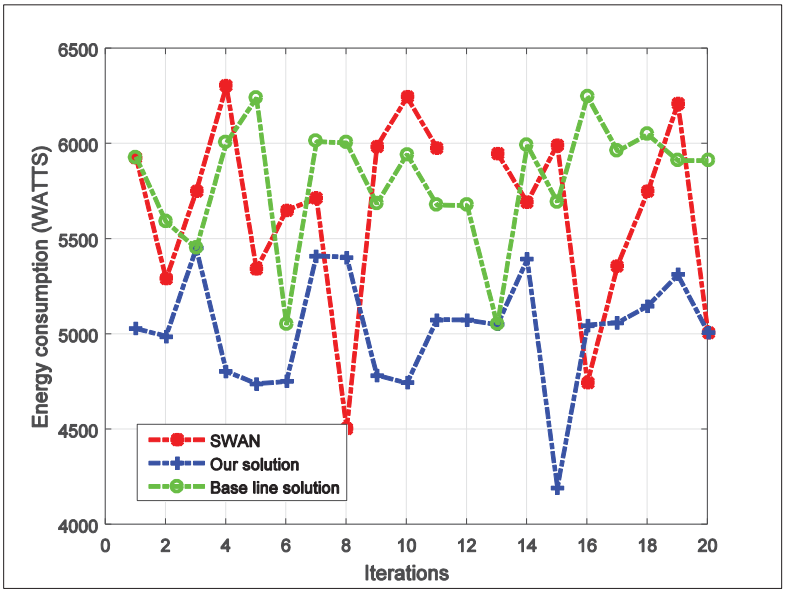


Figure 5.4 Total energy consumption during 20 iterations (SWAN vs our solution vs a base line solution)

We also compute the mean load balancing factor (MLBF) of each solution $s \in \{1,2,3\}$ using the equations (4.28) and (4.29) (1: our solution, 2: SWAN, 3: base line model) each $T_d = 1$ minute for a 10 iterations (Figure 5.5) and for a 20 iterations (Figure 5.6).

$$MLBF^s = \frac{\sum_{i,j \in N} LB_{i,j}}{N^L}, \quad s \in \{1,2,3\}, \; N^L = \#\{Links\} \tag{4.28}$$

$$LB_{i,j} = \frac{Max(b_{i,j}, b_{j,i})}{b_{i,j} + b_{j,i}}, \quad i,j \in N = \{Nodes\} \tag{4.29}$$

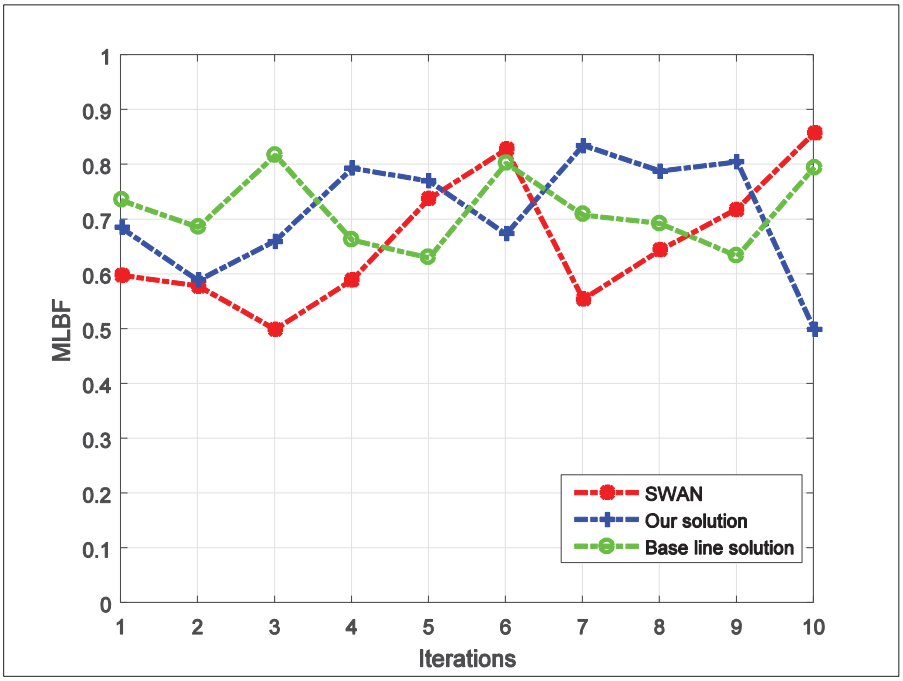


Figure 5.5 Comparing the variation of MLBF during 10 iterations (SWAN vs our solution vs a base line solution)

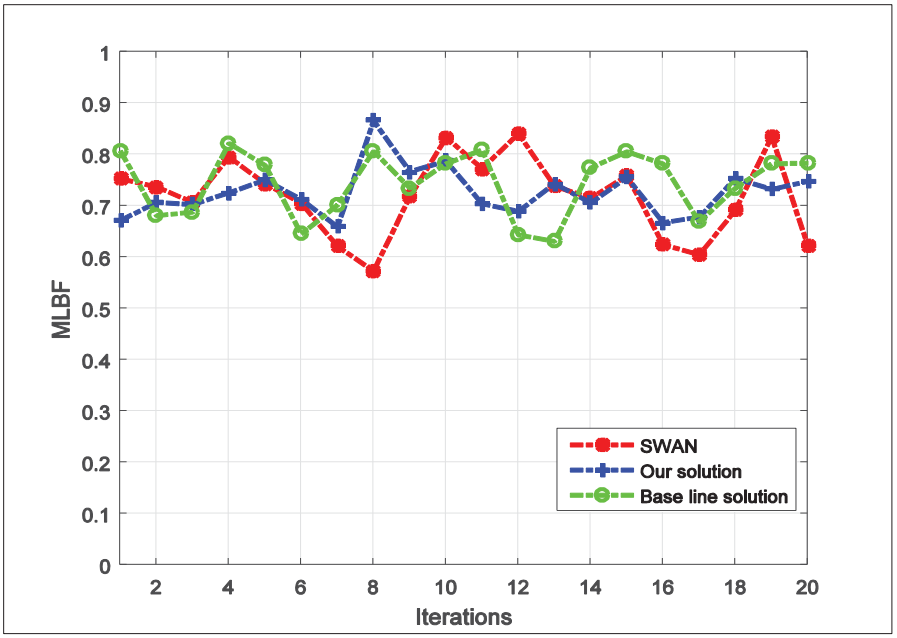


Figure 5.6 Comparing the variation of MLBF during 20 iterations
(SWAN vs our solution vs a base line solution)

As shown in Figure 5.5 and Figure 5.6, the proposed algorithm has a similar performance in terms of ensuring load balancing like SWAN because both solutions allocate bandwidth to different flows using MCF approach. So, to conclude the proposed algorithm minimizes the total energy consumption of the network as shown in Figure 5.7 while maximizing throughput and taking into account technology specifications and communication inter-domain. Moreover, it has a similar performance in terms of the load balancing as the existing solution (SWAN) (Figure 5.8).

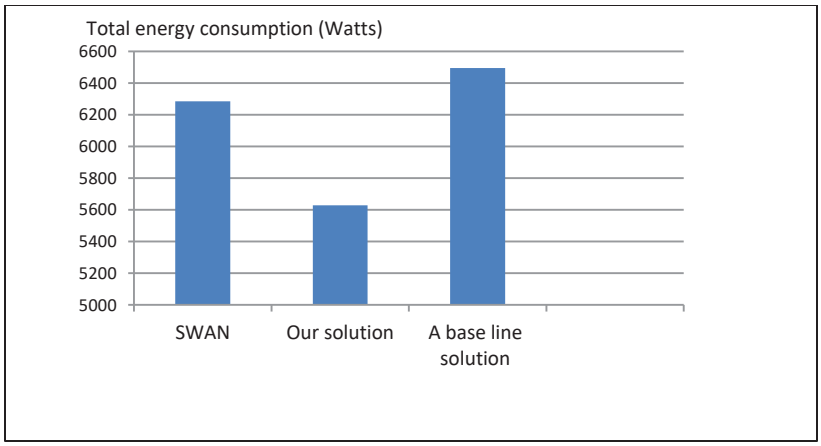


Figure 5.7 Total energy consumption (SWAN vs our solution vs base line model) during 10 iterations

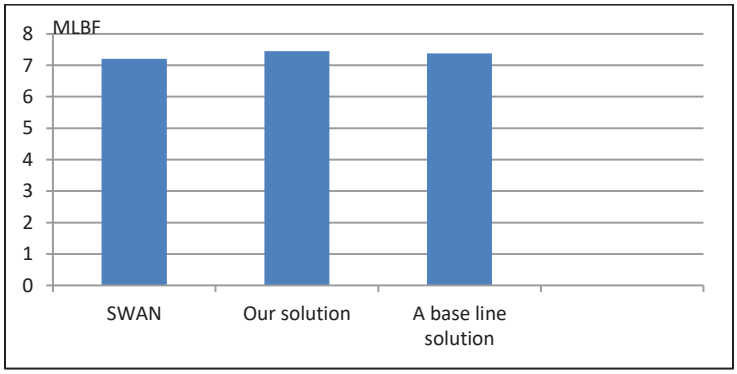


Figure 5.8 MLBF (SWAN vs our solution vs a base line model) during 10 iterations

5.3.2 Test with NSF topology

In the second step, we use a denser topology to test the performance of our algorithm which is NSF topology which contains 14 nodes and 21 links as shown in Figure 5.9.

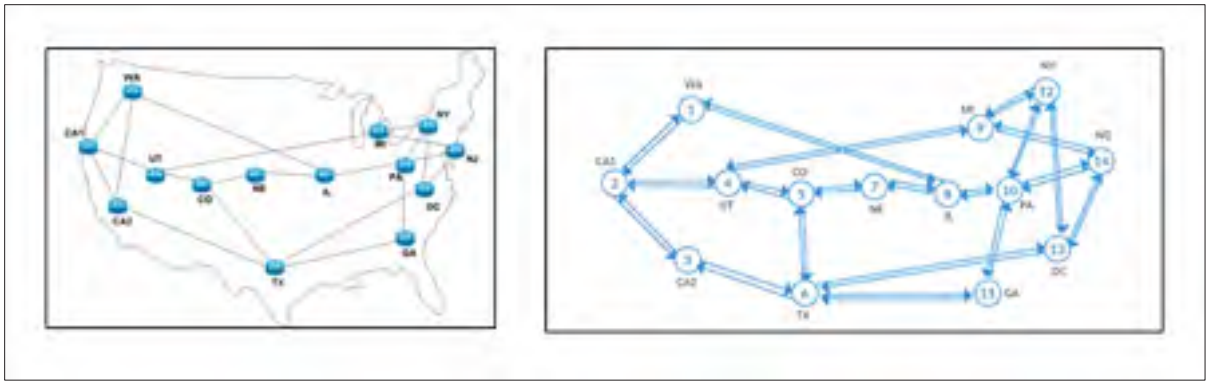


Figure 5.9 NSF topology
Taken from Galán (2008)

With this dense topology, and with a high frequency of network updating and demand generation, the deterministic algorithm provides an energy-aware solution comparable to SWAN as shown in Figure 5.10. The execution delay of the proposed algorithm is similar to SWAN because, at this stage, both algorithm use LP solvers.

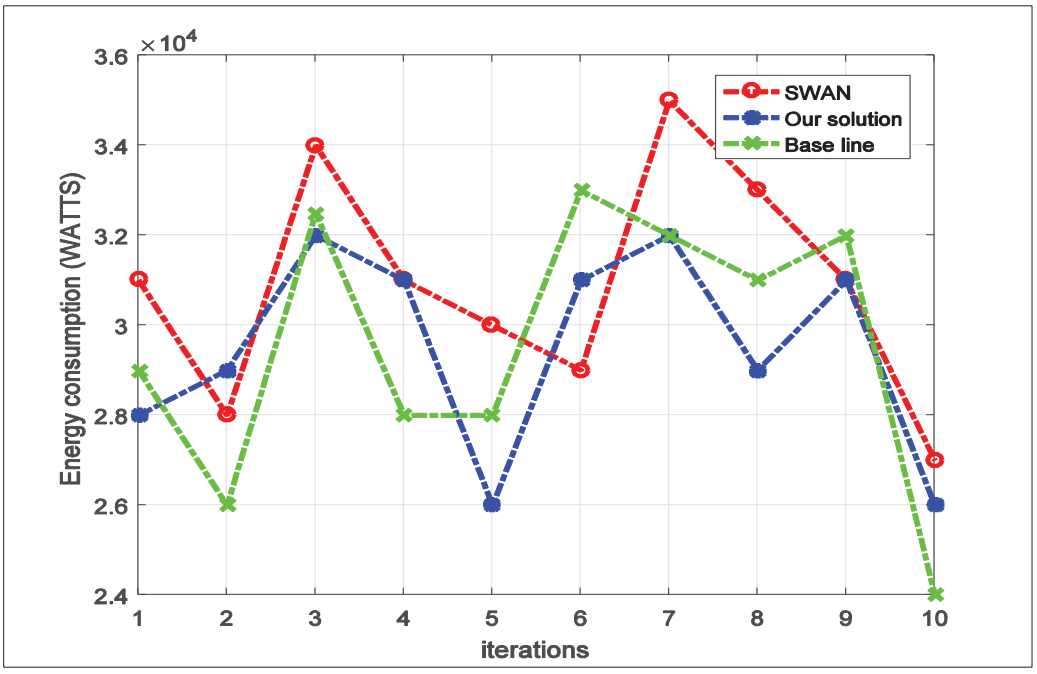


Figure 5.10 Total energy consumption with NSF topology
(SAWN vs our solution vs a base line solution) during
10 iterations

5.3.3 Impact of the energy consumption model in the optimization model

In this section, we compare the performance of the proposed optimization model given input as three energy consumption models presented previously in Chapter 4: the idle model, the step-increasing model and the agnostic model. Testing these three models will help us to study the impact of each model in our problem formulation.

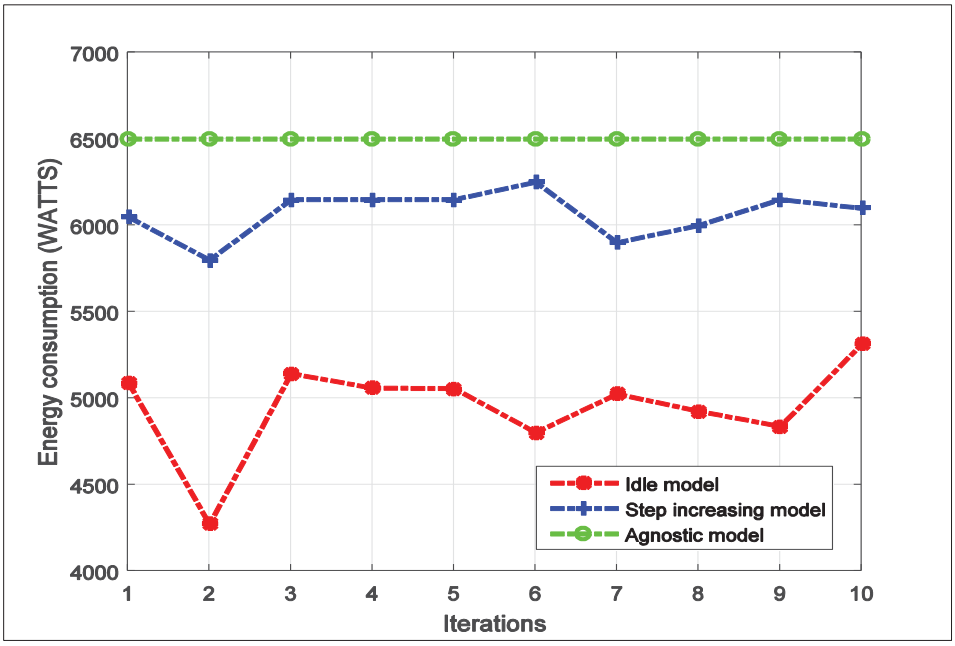


Figure 5.11 Idle vs Step increasing vs agnostic model in the optimization problem during 10 iterations

As shown in Figure 5.11, the agnostic model results in constant energy consumption for all solutions provided by the deterministic algorithm due to the fact that this model ignores the impact of the load on the energy consumption of links and nodes. On the other hand, the discrete step increasing model, which depends on the allocated bandwidth in each component, does not clearly present the variation of the total energy consumption when the network load changes. Finally, the idle model results in a high correlation between network and the total energy consumption.

5.3.4 Large scale cloud services

As this work is dedicated to optimize traffic engineering for large scale cloud services in WAN, we study the correlation between demand and the energy consumption. Figure 5.12 shows the energy consumption increases with the growth of the demand for the three solutions: SWAN, our proposed solution and the baseline solution. In average, the proposed algorithm provides the solution which consumes the minimum amount of energy while ensuring throughput maximization.

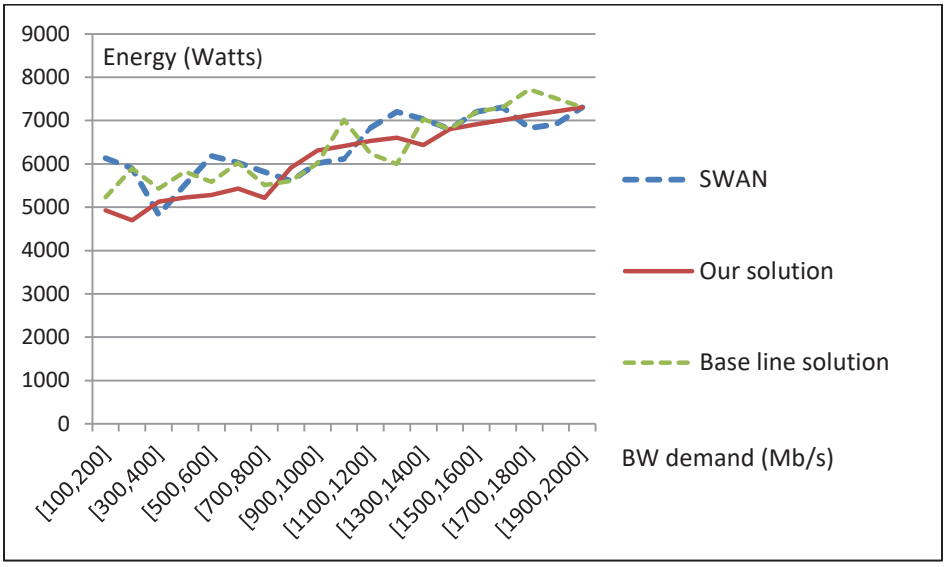


Figure 5.12 Correlation between the growth of the demand and the total energy consumption (SWAN vs our solution vs a base line solution)

5.3.5 Implementation of the greedy heuristic

In this section, we will compare the performance of the greedy heuristic with the performance of the proposed deterministic algorithm and SWAN in terms of total energy consumption and MLBF. Figure 5.13 shows the total energy consumption of the deterministic algorithm, the proposed heuristic and SWAN for 10 minutes and 20 minutes. The proposed heuristic, which is based on a greedy approach, provides a good performance

in terms of total energy consumption. This is because the heuristic focuses on minimizing the total energy consumption of the virtual network while checking constraints related to network specifications and inter-domain communication.

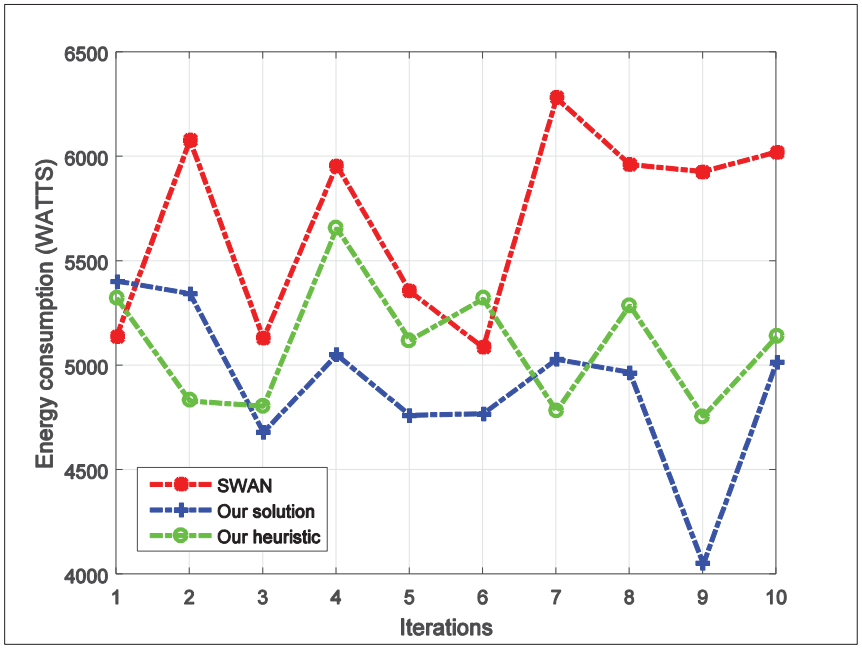


Figure 5.13 Total energy consumption during 10 iterations
(SWAN vs deterministic algorithm vs the heuristic)

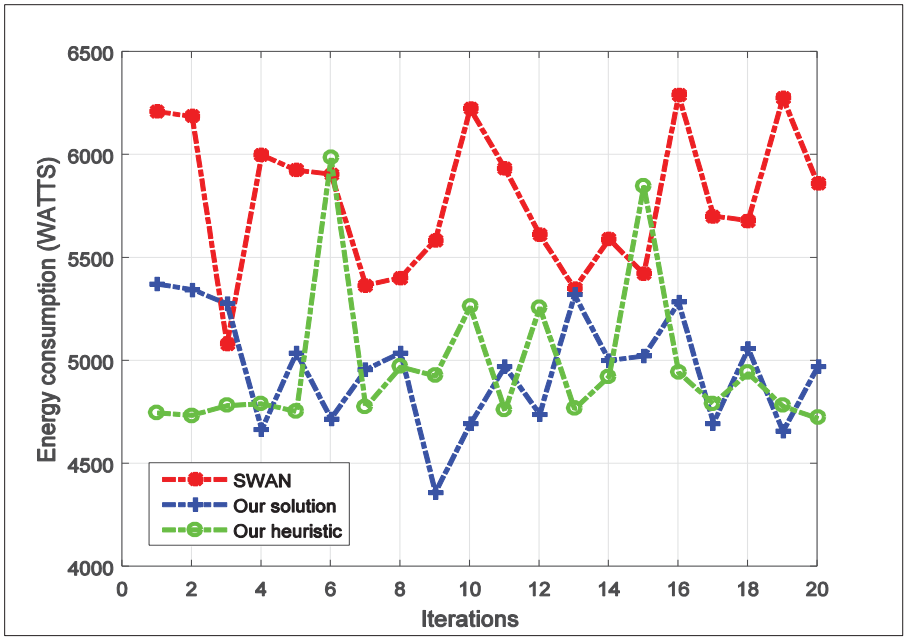


Figure 5.14 Total energy consumption during 20 iterations
(SWAN vs the deterministic algorithm vs the heuristic)

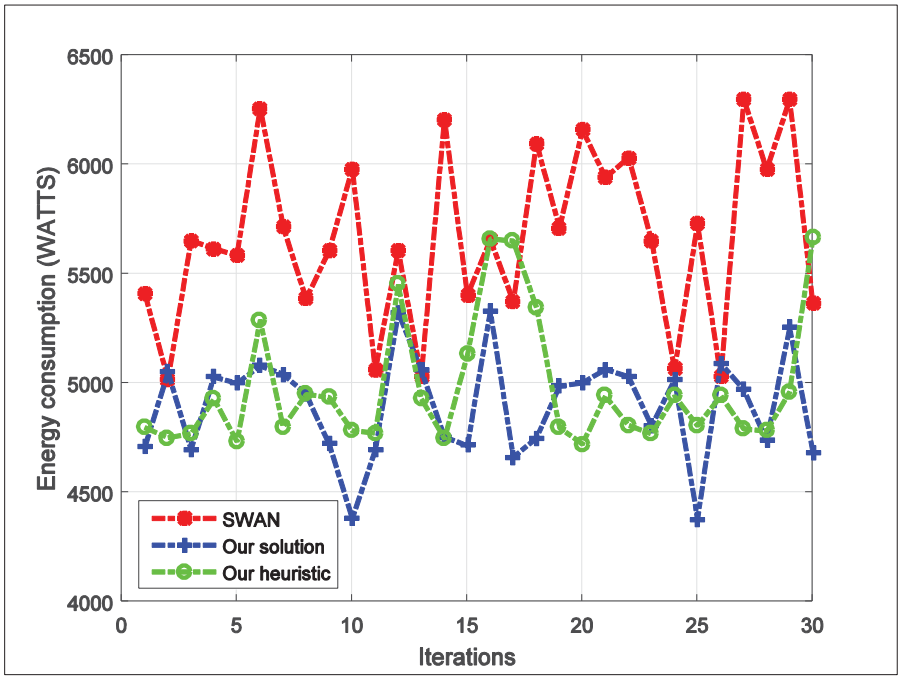


Figure 5.15 Total energy consumption during a 30 iterations
(SWAN vs the deterministic algorithm vs the heuristic)

Figures 5.13, 5.14 and 5.15 show the total energy consumption during a 10-minute, a 20-minute and a 30-minute execution of SWAN, the deterministic algorithm and the greedy heuristic. Through this test, we can realize that the proposed heuristic is able to provide an efficient solution which minimizes the total energy consumption. In fact, between 20% and 30% of results provided by the greedy heuristic coincide with results provided by the LP solver.

The proposed heuristic uses a greedy approach which allocates bandwidth using Breadth-first search (BFS) algorithm (Ueno et al., 2016) based on the energy consumption of different links and nodes in the topology. This approach improves the runtime of the proposed deterministic algorithm as shown in Figure 5.16; in fact, the computing time generated by the LP solvers used in the deterministic algorithm is higher than the runtime generated by the greedy heuristic using SWAN topology as inputs and random demands in terms of bandwidth in each node in the network.

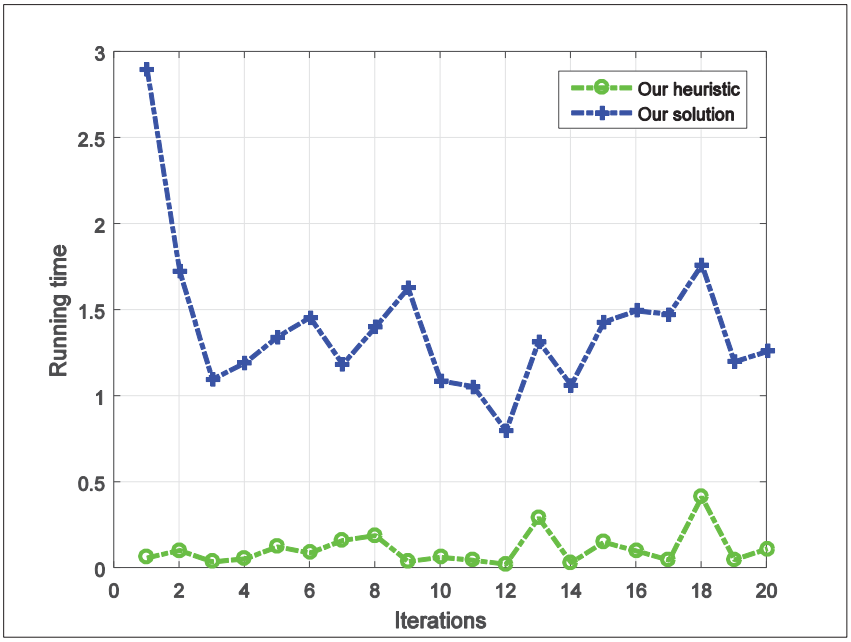


Figure 5.16 Runtime of the deterministic algorithm vs the heuristic during 20 iterations

On the other hand, starting from a topology with 18 nodes, the runtime of LP solvers used in the deterministic algorithm increases exponentially, unlike the runtime of the greedy heuristic (Figure 5.17). So, it is recommended to use the greedy heuristic when the number of nodes in the physical topology overtakes 18 nodes in order to generate a solution with a reasonable computing time.

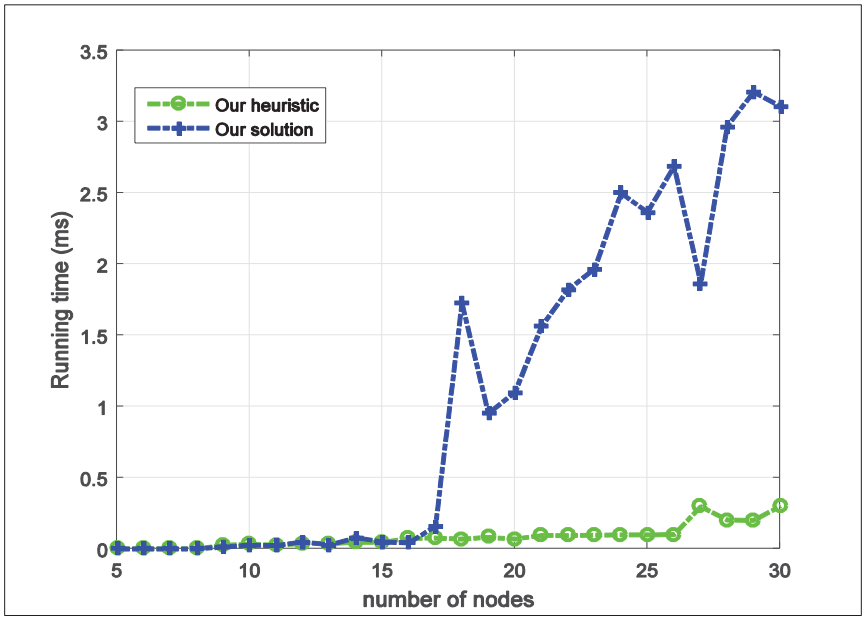


Figure 5.17 The variation of the runtime based on the number of nodes in the topology (the algorithm vs the heuristic)

5.4 Conclusion

In this chapter, we presented the implementation and experimental results to validate the proposed solution (the deterministic algorithm and the greedy heuristic). Also, we presented comparative study between performance achieved by our proposed solution and existing solutions (SWAN and a baseline solution). Results show that the deterministic algorithm is able to maximize network throughput and at the same time minimize the total energy consumption while taking into account characteristics of underlying layers. Finally, we presented the greedy heuristic which improves the runtime of the proposed solution.

CONCLUSION

The main objective of Infrastructure Providers (InPs) is the maximization of resource utilization and the optimization of resource allocation in order to maximize their revenue and to carry more traffic with the same existing physical capacities. On the other hand, SPs objectives are to minimize the total cost while responding to their demand in terms of bandwidth and required QoS based on flows priorities. Moreover, service needs in terms of QoS (delay, jitter, and packet loss) evolve in an exponential way while having the same networking infrastructure. So, it is primordial to find new solutions to let the existing infrastructure carry more traffic with the same existing resources while guaranteeing the required QoS.

Prior work, like B4 and SWAN, focuses on the optimization of resource allocation while taking into account existing physical capacities. Other works, like MCTEQ and DOV, focus on the minimization of delay while maximizing the throughput in the network in order to guarantee the required QoS for interactive services. None of them takes into account characteristics of underlying layers and the total energy consumption of the virtual network. Our contributions in this work are:

- Energy consumption modeling of different components in the network (nodes and links);
- An optimization model which takes into account different energy profiles of network elements (nodes and links) and underlying network specifications (i.e., MPLS);
- A deterministic algorithm proposed to solve the optimization problem using LP solvers;
- A greedy heuristic proposed to improve the runtime of the proposed solution.

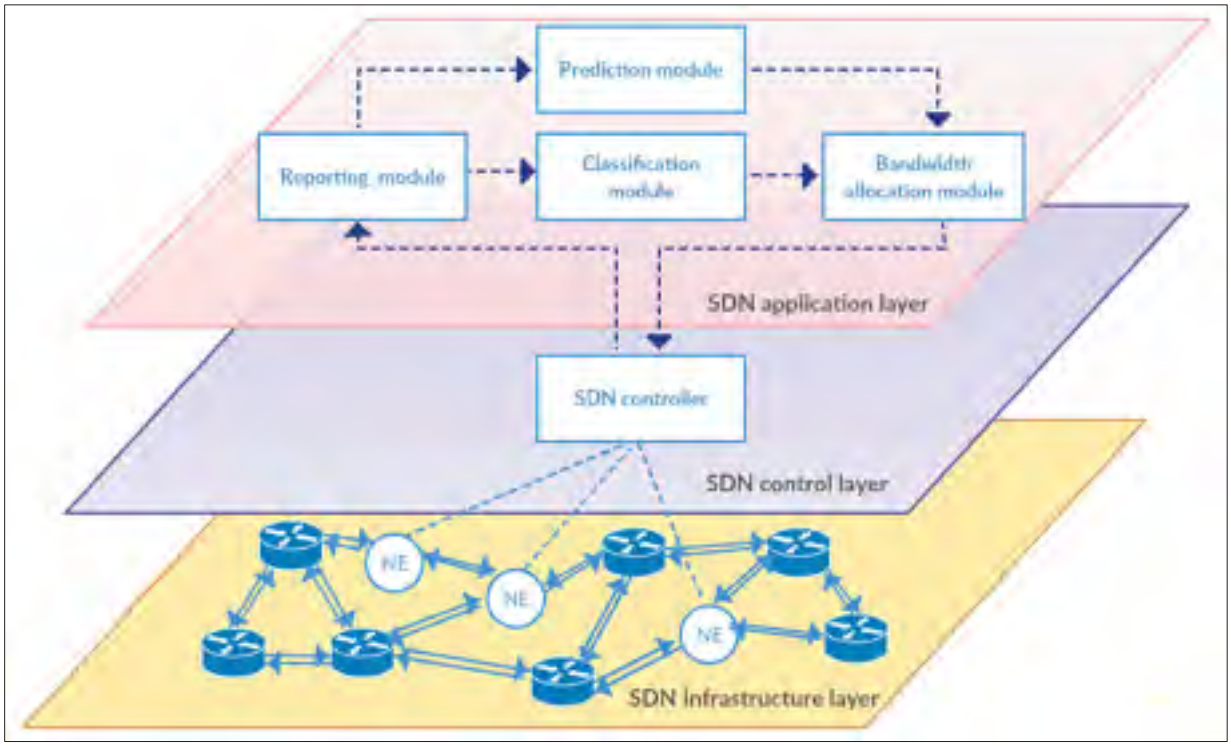
Our objective is to define the optimal virtual network on top of the existing physical network in order to minimize the energy consumption and to maximize throughput in Inter-DC WAN. To this end, we used the MCF model to maximize throughput in the network while allocating bandwidth to flows belonging to different CoS. Then, we integrated four energy consumption models of nodes and links: the idle, the fully proportional, the agnostic and the step-

increasing model. Finally, we built our energy aware optimization model to maximize resources utilization while minimizing the total energy consumption and taking into account network specifications (MPLS) and inter-domain communication.

In this work, we assumed that physical nodes and links which are not allocated to the virtual topology may be put off or in sleep mode. Moreover, in extreme cases like hungry flows that demand higher bandwidth than the existing physical capacity, we assumed using MMF approach to allocate bandwidth to different flows while ensuring fairness and respecting the existing resource capacity. To meet application requirements, we adopted the classification of flows into three main classes: interactive, elastic and background.

We proposed a deterministic algorithm which calls LP solvers. Then, to improve the runtime of the proposed solution, we proposed a greedy heuristic, which is particularly efficient regarding dense topologies. We validated our work by implementing the deterministic algorithm and the greedy heuristic using Matlab and compared their performance with two existing works: SWAN and a baseline solution. Our proposed heuristic can ensure the scalability of the proposed solution, maximize network utilization and minimize the total energy consumption while respecting different constraints related to networking policies and technologies' specifications. The experimental results show that between 80% and 90% of results provided by the proposed algorithm and the greedy heuristic consume the minimum energy to carry the same traffic as SWAN and the baseline solution. This work has been submitted to IEEE transactions on networking and services management (TNSM).

In future research, we propose to extend our work to take into account other technologies, such as optical networks and Ethernet network. On the other hand, we will investigate how to improve the proposed heuristic in order to get more accurate results. Finally, we propose to develop our solution by adding two blocs in the SDN application layer as shown in the following figure: a prediction bloc to predict the remaining physical capacities in the network based on current values and a classification bloc to classify incoming services based on their requirements in terms of QoS (delay, jitter and packet loss).



Our perspectives

BIBLIOGRAPHY

Addis, B., A. Capone, G. Carello, L. G. Gianoli et B. Sansò. 2014. « Energy Management Through Optimized Routing and Device Powering for Greener Communication Networks ». *IEEE/ACM Transactions on Networking,* vol. 22, n^{o} 1, p. 313-325.

Ariyo, A. A., A. O. Adewumi et C. K. Ayo. 2014. « Stock Price Prediction Using the ARIMA Model ». In *2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation*. (26-28 March 2014), p. 106-112.

Awduche, D. O. 1999. « MPLS and traffic engineering in IP networks ». *IEEE Communications Magazine,* vol. 37, n^{o} 12, p. 42-47.

Battista, G. Di, M. Rimondini et G. Sadolfo. 2012. « Monitoring the status of MPLS VPN and VPLS based on BGP signaling information ». In *2012 IEEE Network Operations and Management Symposium*. (16-20 April 2012), p. 237-244.

Bianzino, A. P., C. Chaudet, F. Larroca, D. Rossi et J. L. Rougier. 2010. « Energy-aware routing: A reality check ». In *2010 IEEE Globecom Workshops*. (6-10 Dec. 2010), p. 1422-1427.

Biradar, S., B. Alawieh et H. Mouftah. 2006. « Delivering reliable real-time multicast services over virtual private LAN service ». In *2nd International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities, 2006. TRIDENTCOM 2006.* (0-0 0), p. 6 pp.-552.

Bücker, H. M., et C. Sohr. 2014. « Reformulating a Breadth-First Search Algorithm on an Undirected Graph in the Language of Linear Algebra ». In *2014 International Conference on Mathematics and Computers in Sciences and in Industry*. (13-15 Sept. 2014), p. 33-35.

Cai, Z., F. Liu, N. Xiao, Q. Liu et Z. Wang. 2010. « Virtual Network Embedding for Evolving Networks ». In *2010 IEEE Global Telecommunications Conference GLOBECOM 2010*. (6-10 Dec. 2010), p. 1-5.

Cheng, Y., H. Li, P. J. Wan et X. Wang. 2012. « Wireless Mesh Network Capacity Achievable Over the CSMA/CA MAC ». *IEEE Transactions on Vehicular Technology,* vol. 61, n^{o} 7, p. 3151-3165.

Chowdhury, Mosharaf, Muntasir Raihan Rahman et Raouf Boutaba. 2012. « ViNEYard: virtual network embedding algorithms with coordinated node and link mapping ». *IEEE/ACM Trans. Netw.,* vol. 20, n^{o} 1, p. 206-219.

Cittadini, Luca, Giuseppe Di Battista et Maurizio Patrignani. 2013. « MPLS Virtual Private Networks ». *Recent Advances in Networking*.

Cui, Z., L. Chen, M. Chen, Y. Bao, Y. Huang et H. Lv. 2012. « Evaluation and Optimization of Breadth-First Search on NUMA Cluster ». In *2012 IEEE International Conference on Cluster Computing*. (24-28 Sept. 2012), p. 438-448.

Detti, A., A. Caricato et G. Bianchi. 2010. « Fairness-oriented Overlay VPN topology construction ». In *2010 17th International Conference on Telecommunications*. (4-7 April 2010), p. 658-665.

Dumka, A., H. L. Mandoria, K. Dumka et A. Anand. 2015. « MPLS VPN using IPv4 and IPv6 protocol ». In *2015 2nd International Conference on Computing for Sustainable Global Development (INDIACom)*. (11-13 March 2015), p. 1051-1055.

eTutorials.org. 2017. « Overlay and Peer-to-peer VPN Model ». On line. </http://etutorials.org/Networking/MPLS+VPN+Architectures/Part+2+MPLS-based+Virtual+Private+Networks/Chapter+7.+Virtual+Private+Network+VPN+Implementation+Options/Overlay+and+Peer-to-peer+VPN+Model/>. Accessed May 2017.

Galán, Fermín. 2008. « NSF Network with 14 nodes ». On line. </http://www.dit.upm.es/vnumlwiki/index.php/Example-NSF-14_1.8/>.Accessed May 2017.

Ghazisaeedi, E., C. Huang et J. Yan. 2015. « Off-peak energy-wise link reconfiguration for virtualized network environment ». In *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*. (11-15 May 2015), p. 814-817.

Ghosal, S., et S. C. Ghosh. 2014. « A Probabilistic Greedy Algorithm with Forced Assignment and Compression for Fast Frequency Assignment in Cellular Network ». In *2014 IEEE 13th International Symposium on Network Computing and Applications*. (21-23 Aug. 2014), p. 189-196.

Gouveia, R., J. Aparício, J. Soares, B. Parreira, S. Sargento et J. Carapinha. 2014. « SDN framework for connectivity services ». In *2014 IEEE International Conference on Communications (ICC)*. (10-14 June 2014), p. 3058-3063.

Hachimi, M. el, et M. Bennani. 2007. « Mobile agents based approach for P2MP VPLS ». In *IEEE International Conference on Pervasive Services*. (15-20 July 2007), p. 427-430.

Hong, Chi-Yao, Srikanth Kandula, Ratul Mahajan, Ming Zhang, Vijay Gill, Mohan Nanduri et Roger Wattenhofer. 2013. « Achieving high utilization with software-driven WAN ». *SIGCOMM Comput. Commun. Rev.,* vol. 43, n^{o} 4, p. 15-26.

Hu, F., Q. Hao et K. Bao. 2014. « A Survey on Software-Defined Network and OpenFlow: From Concept to Implementation ». *IEEE Communications Surveys & Tutorials,* vol. 16, n^{o} 4, p. 2181-2206.

Hu, J. W., C. S. Yang et T. L. Liu. 2016. « L2OVX: An On-demand VPLS Service with Software-Defined Networks ». In *2016 30th International Conference on Advanced Information Networking and Applications Workshops (WAINA)*. (23-25 March 2016), p. 861-866.

Jain, Sushant, Alok Kumar, Subhasree Mandal, Joon Ong, Leon Poutievski, Arjun Singh, Subbaiah Venkata, Jim Wanderer, Junlan Zhou, Min Zhu, Jon Zolla, Urs H, #246, lzle, Stephen Stuart et Amin Vahdat. 2013. « B4: experience with a globally-deployed software defined wan ». *SIGCOMM Comput. Commun. Rev.,* vol. 43, n^{o} 4, p. 3-14.

Jerkins, J. A. 2015. « Detecting MPLS L3 VPN misconfiguration with the MINA algorithm ». In *2015 International Conference and Workshop on Computing and Communication (IEMCON)*. (15-17 Oct. 2015), p. 1-5.

Kodaganallur, V., et A. K. Sen. 2010. « Greedy by Chance - Stochastic Greedy Algorithms ». In *2010 Sixth International Conference on Autonomic and Autonomous Systems*. (7-13 March 2010), p. 182-187.

Kreutz, D., F. M. V. Ramos, P. E. Veríssimo, C. E. Rothenberg, S. Azodolmolky et S. Uhlig. 2015. « Software-Defined Networking: A Comprehensive Survey ». *Proceedings of the IEEE,* vol. 103, n^{o} 1, p. 14-76.

Lara, A., A. Kolasani et B. Ramamurthy. 2014. « Network Innovation using OpenFlow: A Survey ». *IEEE Communications Surveys & Tutorials,* vol. 16, n^{o} 1, p. 493-512.

Liu, Y., D. Niu et B. Li. 2016. « Delay-Optimized Video Traffic Routing in Software-Defined Interdatacenter Networks ». *IEEE Transactions on Multimedia,* vol. 18, n^{o} 5, p. 865-878.

Lu, Feng, et Lei Xu. 2010. « Power data network VPN deployment based on PBB and H-VPLS ». In *2010 Second Pacific-Asia Conference on Circuits, Communications and System*. (1-2 Aug. 2010) Vol. 2, p. 174-177.

Malis, A. G. 2012. « MPLS-TP: Where are we? ». In *OFC/NFOEC*. (4-8 March 2012), p. 1-3.

MikroTik. 2010. « Manual:Layer-3 MPLS VPN example ». On line. </https://wiki.mikrotik.com/wiki/Manual:Layer-3_MPLS_VPN_example/>.Accessed May 2017.

Nonde, L., T. E. H. El-Gorashi et J. M. H. Elmirghani. 2015. « Energy Efficient Virtual Network Embedding for Cloud Networks ». *Journal of Lightwave Technology,* vol. 33, n^{o} 9, p. 1828-1849.

Nunes, B. A. A., M. Mendonca, X. N. Nguyen, K. Obraczka et T. Turletti. 2014. « A Survey of Software-Defined Networking: Past, Present, and Future of Programmable Networks ». *IEEE Communications Surveys & Tutorials,* vol. 16, n^{o} 3, p. 1617-1634.

Ouerfelli, L., et H. Bouziri. 2011. « Greedy algorithms for dynamic graph coloring ». In *2011 International Conference on Communications, Computing and Control Applications (CCCA)*. (3-5 March 2011), p. 1-5.

Parra, O. J. Salcedo, G. L. Rubio et L. Castellanos. 2012. « MPLS/VPN/BGP Networks evaluation techniques ». In *2012 Workshop on Engineering Applications*. (2-4 May 2012), p. 1-6.

Rawat, D. B., et S. R. Reddy. 2017. « Software Defined Networking Architecture, Security and Energy Efficiency: A Survey ». *IEEE Communications Surveys & Tutorials,* vol. 19, n^{o} 1, p. 325-346.

Sassi, Haifa, et Tara Nath Subedi. 2017. « Enhancing Availability in a Huge Network Infrastructure: The Google Experience ». *Substance ETS*.

Strelkovskaya, I., I. Solovskaya et S. Paskalenko. 2015. « Solution to a problem of routing in MPLS-TE network with additional directions of traffic transmission ». In *2015 Second International Scientific-Practical Conference Problems of Infocommunications Science and Technology (PIC S&T)*. (13-15 Oct. 2015), p. 54-57.

Su, S., Z. Zhang, A. X. Liu, X. Cheng, Y. Wang et X. Zhao. 2014. « Energy-Aware Virtual Network Embedding ». *IEEE/ACM Transactions on Networking,* vol. 22, n^{o} 5, p. 1607-1620.

Sun, M. S., et W. H. Wu. 2012. « Engineering analysis and research of MPLS VPN ». In *2012 7th International Forum on Strategic Technology (IFOST)*. (18-21 Sept. 2012), p. 1-5.

Trevisan, Luca. 2011. « CS261: Optimization ». On line. </ https://people.eecs.berkeley.edu/~luca/cs261/lecture16.pdf/>. Accessed May 2017.

TS-007, ONF. 2012. « OpenFlow Switch Specification ». On line. </https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf/specifications/openflow/openflow-spec-v1.3.1.pdf/>. Accessed May 2017.

Ueno, K., T. Suzumura, N. Maruyama, K. Fujisawa et S. Matsuoka. 2016. « Extreme scale breadth-first search on supercomputers ». In *2016 IEEE International Conference on Big Data (Big Data)*. (5-8 Dec. 2016), p. 1040-1047.

Vishwanath, A., K. Hinton, R. W. A. Ayre et R. S. Tucker. 2014. « Modeling Energy Consumption in High-Capacity Routers and Switches ». *IEEE Journal on Selected Areas in Communications,* vol. 32, n^{o} 8, p. 1524-1532.

Wang, J. M., Y. Wang, X. Dai et B. Bensaou. 2014. « SDN-based multi-class QoS-guaranteed inter-data center traffic management ». In *2014 IEEE 3rd International Conference on Cloud Networking (CloudNet)*. (8-10 Oct. 2014), p. 401-406.

Wang, Z., et X. Guo. 2012. « The simulation and improvement of Diff-Serv model based on OPNET ». In *Proceedings of 2012 2nd International Conference on Computer Science and Network Technology*. (29-31 Dec. 2012), p. 992-995.

Winter, R. 2011. « The coming of age of MPLS ». *IEEE Communications Magazine,* vol. 49, n^{o} 4, p. 78-81.

Yildirim, A. S., et T. Girici. 2014. « Cloud Technology and Performance Improvement with Intserv over Diffserv for Cloud Computing ». In *2014 International Conference on Future Internet of Things and Cloud*. (27-29 Aug. 2014), p. 222-229.

Yunos, R., S. A. Ahmad, N. M. Noor, R. M. Saidi et Z. Zainol. 2013. « Analysis of routing protocols of VoIP VPN over MPLS network ». In *2013 IEEE Conference on Systems, Process & Control (ICSPC)*. (13-15 Dec. 2013), p. 139-143.

Zemmouri, S., S. Vakilinia et M. Cheriet. 2016. « Let's adapt to network change: Towards energy saving with rate adaptation in SDN ». In *2016 12th International Conference on Network and Service Management (CNSM)*. (Oct. 31 2016-Nov. 4 2016), p. 272-276.

Zhang, Q., Q. Zhu, M. F. Zhani et R. Boutaba. 2012. « Dynamic Service Placement in Geographically Distributed Clouds ». In *2012 IEEE 32nd International Conference on Distributed Computing Systems*. (18-21 June 2012), p. 526-535.

Zhang, Y., P. Chowdhury, M. Tornatore et B. Mukherjee. 2010. « Energy Efficiency in Telecom Optical Networks ». *IEEE Communications Surveys & Tutorials,* vol. 12, n^{o} 4, p. 441-458.