UNIVERSITÀ DEGLI STUDI DI CATANIA
Dipartimento di Ingegneria Elettrica, Elettronica ed Informatica

Dottorato di Ricerca in Ingegneria Informatica e delle Telecomunicazioni
XXVIII Ciclo

SUSTAINABLE DEVELOPMENT OF NETWORKS AND TELECOMMUNICATION SERVICES

Ing. Vincenzo Mario Riccobene

Coordinatore	Tutor
Chiar.ma Prof.ssa V. Carchiolo	Chiar.mo Prof. G. Schembra

To my family,
always encouraging me
to strive to do my best.

To Federica,
always bearing me
and my wacky ideas.

Imagination is more
important than knowledge.
Knowledge is limited.
Imagination encircles the world.
ALBERT EINSTEIN

SOMMARIO

Lo sviluppo sostenibile delle reti di Telecomunicazioni comporta la messa in opera di architetture e servizi che tengano in considerazione le esigenze presenti e future del pianeta e dei suoi abitanti. Questo lavoro di tesi ha lo scopo di analizzare gli aspetti collegati allo sviluppo sostenibile e studiarne le relazioni con il mondo delle telecomunicazioni: come gli odierni servizi di telecomunicazioni influiscono sullo sviluppo sostenibile e quali linee e' necessario seguire al fine di progettare i servizi e le reti del futuro.

Il lavoro di tesi comincia con un'analisi degli studi effettuati sullo sviluppo sostenibile negli ultimi dieci anni dalle Nazioni Unite. Una specializzazione di questi concetti per quello che concerne Internet e i servizi di telecomunicazioni e' stata possibile grazie ad un'ulteriore analisi del lavoro portato avanti dalla Internet Society, attraverso la quale sono stati individuati diversi gap che, a mio parere, la ricerca deve affrontare da un punto di vista tecnologico per andare in contro ad uno sviluppo sostenibile dei servizi di telecomunicazioni partendo da odierni servizi e infrastrutture. Essi sono:

- Riduzione dei costi
- Riduzione dell'impatto sull'ambiente.
- Incremento di flessibilita' e scalabilita'
- Incremento dell'affidabilita' e stabilita'

Il risultato complessivo delle attivita' di ricerca descritte in questa tesi puo' essere riassunto nella proposta di una nuova architettura di alto livello della rete che abbraccia l'intero panorama, ma che allo stesso tempo non si vuole proporre come una soluzione generale e omnicomprensiva, ma al contrario, vuole essere un punto di partenza per integrare nuove tecnologie e favorire l'ingresso di ulteriori nuove tecnologie attraverso l'uso di interfacce standard, API e controller centralizzati. In questa figura vengono inglobati e tenuti in considerazione tutti i livelli della rete: dalle reti di accesso alla core, includendo anche i Data Center. Tale architettura si propone come un approccio di alto livello per una maggiore integrazione e coordinazione tra le diverse tecniche che stanno nascendo negli ultimi anni indipendentemente e che se combinate possono aiutare a colmare le correnti lacune di cui la rete soffre da un punto di vista dello sviluppo sostenibile.

L'architettura proposta e' supportata da modelli matematici e Proof-of-Concepts (PoCs) che ne dimostrano i principi e la fattibilita'.

Quattro importanti driver tecnologici vengono presi in considerazione nell'architettura proposta:

- Efficienza Energetica dei dispositivi di rete
- Software Defined Networking

- Virtualizzazione delle funzioni di rete
- Cloud e Fog computing

L'impatto che l'efficienza energetica ha sui gap indicati all'inizio di questa sezione e' duplice, in quanto contribuisce sia alla riduzione dei costi di servizio che alla riduzione dell'impatto sull'ambiente. Se si vogliono ridurre i consumi energetici degli apparati di rete e' fondamentale lavorare sia sul Data Plane che sul Control Plane. La capacita' di un device di modificare la propria frequenza di clock o di accendere e spegnere periferiche a seconda che siano necessarie in quel momento o meno puo' giocare un ruolo fondamentale. Diverse tecniche sono state analizzate, progettate ed implementate, prendendo come riferimento la NetFPGA board: una piattaforma programmabile a livello hardware grazie alla presenza di un chip FPGA, corredata da moduli hardware aggiuntivi che ne fanno un Router programmabile. Grazie all'uso di questa scheda e' stato possibile misurare fisicamente le principali fonti di spreco ed implementare prototipi per validare nuove tecniche che agiscono direttamente a livello hardware, misurandone l'impatto reale sulle prestazioni e sui consumi energetici. E' stata aggiunta al Reference Router della NetFPGA la possibilita' di scalare la frequenza dell'unita' di processamento del traffico tra 125Mhz (frequenza con cui e' possibile gestire fino a 4Gbps di traffico) e 62.5MHz (con cui e' possibile gestirne 2Gbps). Uno degli elementi esplorati nelle attivita' di ricerca e' il design di una politica di switching che consente di ridurre il consumo energetico della scheda supportando al qualita' del servizio richiesta. Inoltre, e' stata esplorata la possibilita' di implementare sulla scheda un numero maggiore di frequenze di clock, in maniera da seguire l'andamento del traffico in modo piu' fedele. Per questo e' stato progettato un modello analitico in grado di

- aggiustare i parametri significativi del modello,
- prevedere quale sarebbe stato il comportamento del sistema con un numero maggiore di frequenze e
- calcolare la configurazione di frequenze piu' appropriata per la gestione del traffico.

I componenti sono stati integrati sulla NetFPGA, realizzando una architettura hardware/software in cui il Router Governor, e' in grado di ridurre i consumi energetici del dispositivo in base al traffico in ingresso, variando gli stati di funzionamento dell'hardware e riducendo conseguentemente il consumo complessivo dell'apparato, mantenendo la Qualita' di Servizio entro i livelli desiderati.

Un aspetto fondamentale rimane quello della comunicazione tra Data Plane (DP) e Control Plane (CP): il Control Plane puo' agire in modo energy-aware solo se e' a conoscenza delle capacita' che il Data Plane ha di variare il proprio stato di lavoro. Una soluzione a tale problema e' il Green Abstraction Layer (GAL): un framework partorito all'interno del progetto EcoNET, in stretta collaborazione con partner industriali e produttori di apparati di rete. Questo framework viene proposto come soluzione per colmare questa lacuna attualmente presente nella rete.

Software Defined Networking si riferisce alla separazione fisica tra Control Plane e Data Plane degli apparati di rete. Piu' in particolare, il Data Plane viene implementato all'interno del device che si occupa di fare il forwarding dei dati, mentre il Control Plane e' separato e viene gestito esternamente. Una delle piu' comuni soluzioni da questo punto di vista consiste nel centralizzare il Control Plane, fornendo un unico nodo (opportunamente replicato per supportare fault-tolerant) che si occupa della reale implementazione del Control Plane di tutti i nodi presenti in una area della rete. Questa separazione, quindi, introduce la necessita' di implementare la comunicazione tra il nodo che implementa il CP e quello che implementa il DP. Tra i protocolli che risolvono questo problema, quello che ha raggiunto maggiore maturita' e adozione nel mondo industriale e della ricerca e' OpenFlow. OpenFlow e' un protocollo open che fornisce delle primitive di comunicazione tra il controller e i device (chiamati switch OpenFlow). Il contributo apportato da questo pilastro alla realizzazione di una rete in linea con i principi dello sviluppo sostenibile e' abbastanza generale, in quanto contribuisce su diversi fronti:

- incrementa la flessibilita', introducendo la possibilita' di variare dinamicamente la distribuzione dei flussi in tempo reale partendo da una visione globale della rete;
- incrementa la stabilita', dato che e' possibile implementare meccanismi di self-healing, basati sul riconoscimento automatico di malfunzionamenti ed eventuali meccanismi, anch'essi automatici, di recovery;
- consente di introdurre logiche nuove nella rete e a livello globale che possono ridurre i consumi energetici degli apparati, riducendo quindi sia l'impatto ambientale che i costi della rete.

Durante le attivita' di ricerca e' stato dimostrato come l'uso di SDN puo' consentire alla rete di beneficiare di questi aspetti. E' stata proposta una estensione di OpenFlow, basata sulle primitive del Green

Abstraction Layer ed e' stata fornita una Network Control Policy di esempio che dimostra come sia possibile ridurre i costi mantenendo alta la QoS.

L'uso di SDN e' stato inoltre proposto in combinazione con NFV e Cloud/Fog computing nella architettura di NetFATE, discussa in seguito. La virtualizzazione delle funzioni di rete, anche conosciuta come NFV (Network Function Virtualization) consiste nel virtualizzare la componente Data Plane di una funzione di rete e consentire la separazione tra la funzione di rete stessa e l'hardware sul quale essa esegue. Il trend che si sta affermando negli ultimi anni consiste nella realizzazione di "virtual appliance" basate su diversi meccanismi di virtualizzazione (hypervisors, container, ecc) che introducono nel mondo delle telecomunicazioni la possibilita' di usufruire dei vantaggi di cui si e' finora usufruito solo nel mondo IT, principalmente attraverso approcci come il Cloud Computing. Questo approccio rappresenta un grande passo verso la riduzione dei costi e del time to market di nuovi servizi: da un lato, gli operatori non saranno piu' costretti ad investire nell'acquisto di nuovo hardware ogni qualvolta sara' necessario introdurre una nuova funzione di rete per supportare nuovi servizi; inoltre questo significa che l'arrivo sul mercato di nuovi prodotti sara' guidato da processi che si muovono principalmente nell'ambito del software, piuttosto che dell'hardware, riducendo i tempi necessari allo sviluppo del prodotto e al deployment dei servizi. Nell'architettura proposta come risultato delle attivita' di ricerca, l'utilizzo di NFV e' consigliato in cooperazione con SDN, Cloud and Fog Computing, come descitto di seguito.

Il numero di utenti della rete e la tipologia di traffico che deve essere gestita dalla odierna infrastruttura sta cambiando molto rapidamente e la mancanza di flessibilita' delle attuali architetture e protocolli di rete si riscontra nelle grandi difficolta' nello scalare le prestazioni dei servizi. Il Cloud computing ha rappresentato negli ultimi anni una delle possibilita' piu' grandi in termini di scalabilita'. Sebbene le tecnologie correlate sono nate sotto il cappello dell'IT (Information Technology), il suo contributo all'evoluzione di Internet e del mondo delle telecomunicazioni e' stato incredibilmente dirompente. In risposta all'incremento di utenti di internet, al cambiamento dei paradigmi di traffico, all'incremento del numero di device costantemente connessi alla rete, anche il Fog computing sta prendendo piede. Inizialmente introdotto da Cisco, il Fog computing rappresenta una versione del Cloud piu' vicina all'utente. Questo consente di incrementare le performance di alcuni servizi, riducendo anche la mutua influenza tra i servizi stessi. La combinazione di questi fattori con SDN ed NFV conduce verso scenari incredibilmente interessanti che possono stravolgere il panorama dei servizi di telecomunicazioni, creando nuove opportunita', ma soprattutto introducendo meccanismi capaci di sostenere i nuovi trend di traffico e le nuove necessita' degli utenti, riducendo ulteriormente i costi di servizio. Queste tecniche insieme possono rappresentare una valida alternativa all'over-provisioning applicato finora. Come facilmente intuibile, il mantenimento di una architettura fisica complessa e sovradimensionata influisce pesantemente sia sui costi di investimento che sui costi di esercizio. Partendo da questo contesto e sfruttando questi principi nell'ottica di realizzare un infrastruttura piu' efficiente in termini di gestione delle risorse e piu' flessibile in termini di adattamento alle richieste, li ho studiati per comprenderne gli spunti innovativi e possibili punti di integrazione. Questo lavoro ha contribuito a dimostrare come sia possibile sfruttare le nuove capacita' di calcolo che stanno nascendo (o che sono gia' presenti) ai confini delle reti per incrementare l'adattamento della rete ai cambiamenti e ai nuovi pattern di traffico.

Tutti i concetti discussi sono stati integrati in una architettura eterogenea che si pone come obiettivo quello di includere dispositivi di rete tradizionali (non capaci di supportare paradigmi come SDN ed NFV) e nuovi device progettati per supportare nuovi protocolli e tecnologie in modo da ottenere un nuovo sistema di telecomunicazioni in grado di supportare i requisiti delle attuali generazioni e di quelle future, seguendo le indicazioni fornite dalle Nazioni Unite e dall'Internet Society per uno sviluppo sostenibile.

ABSTRACT

Sustainable development of telecommunication network is related to the provisioning of new architectures and services aware of current and future needs of people. This reaserch work wants to analyze aspects related to the sustainable development in general and their relation with the telecommunication networks and services in order to help designing the Internet of the future.

The first step has been related to a deep analysis on the sustainable development and the work carried on by the United Nations in the last few years. This concept has been enreached thanks to the work done by the Internet Society which identified gaps to be bridged from different perspectives. The main goal of this work is to explore technological aspects that could help out the future Internet and the telecommunication world to meet requirements supporting the sustainable development. The gaps identified during this analysis were the followings:

- High costs
- High environmental impact

- Low flexibility and scalability
- Low reliability and relisilience

The main results of the research activity here discussed is the proposal of a new high level architecture that integrates new kind of devices and embraces the network as a whole. To properly support the sustainable development, four technological drivers have been individuated and they are briefly listed below:

- Energy efficiency of network devices
- Software Defined Networking (SDN)
- Network Function Virtualization (NFV)
- Cloud and Fog computing

The proposed architecture shows how new technologies (such as energy efficiency Software Defined Networks and Network Function Virtualization) can be integrated within the network along with energy efficiency techniques in a way that is inspired to the Cloud computing (bringing orchestration mechanisms within the network) and it is supported by the Fog computing approach (distributing computational power across the whole network, not only in the Data Centers). The proposed architecture is then here supported by mathematical models and Proof-of-Concepts (PoCs) demonstrating the feasibility of such evolution.

The energy efficiency has a twofold impact since it contributes to the reduction of the environmental impact and also the general reduction of the costs. Work is required at both the Data Plane and the Control Plane of the network. From this perspective it is necessary for the devices to expose internal energy states available that could correspond to different levels of power consumption, as well as different levels of performance. Those states, mainly related to the Data Plane, have to be exposed to the Control Plane in order to apply energy-aware network policies. In this work, the candidate for this is the Green Abstraction Layer (GAL): a framwork designed in the FP7 European founded project EcoNET.

SDN is about decouplong the Control Plane and the Data Plane of the network. Usually this is implemented using a remote centralized controller that programs the Contro Plane for network devices that only implement the Data Plane. This introduces the requirement of a stable and reliable connection between controller and network devices. For this purpose, the procol gaining momentum is OpenFlow: it provides primitives for the communication between the controller and the so called OpenFlow switches. SDN provides to the infrastructure a set of benefits in line with the sostainable development, like:

- Higher flexibility, due to the dynamic configurability of traffic paths using a global view of the network;
- Higher stability, due to the eventual support of self-healing mechanisms based on automatic recognition of outages and overloadings;
- Allows the introduction of new Control Plane stategies at runtime and in software, avoiding the replacement of devices.

During the research activities an extension of OpenFlow has been proposed in order to support the GAL primitives and a Network Control Policy (NCP) has been designed in order to demonstrate the feasibility of such an approach.

The usage of SDN has also been proposed in combination with NFV, Cloud and Fog computing in the design of NetFATE (Network Functions At The Edge) to afford the increasingly higher amount of traffic of the Internet and the new traffic patterns. More in details, NFV consists of the separation of network functions from the hardware they run on top of and that is usaully achieved by mean of virtualization. This represents a step further towards the cost and time to market reduction of new services: on the one hand the operators will not be required to spend money buying new network dedicated hardware any time an update of the network is required; moreover the development of new services will be driven by software business model (not hardware) which will further reduce the time of service development.

The proposed integrated high level architecture includes all the discussed elements and shows a way to integrate them together in the Internet of the future.

In Chapter 1 the sustainable development arguments have been discussed: main challenges and motivations are discussed analyzing the work from the United Nations.

In Chapter 2 the concept of sustainable development is extended to the telecommunication world and main gaps are identified and discussed, along with technological drivers listed below.

In Chapter 3 the state of the art related to the technological drivers to be used is presented.

In Chapter 4 the description of the high level architecture for the sustainable development is proposed, along with some motivations.

In Chapter 5 an example of design methodology for a Local Control Policy (LCP) is proposed along with the exposure of internal energy aware states. The LCP has been designed and implemented for a real device (the NetFPGA board) and an analytical model has been designed to support the estimation of the achievable power consumption with a higher number of frequencies.

In Chapter 6 the design of a Network Control Policy (NCP) is proposed. This NCP exploits the OpenFlow protocol and is based on an extension of OpenFlow itself supporting the GAL. The NCP calculates the optimal (or sub-optimal) allocation of traffic flows across the network.

In Chapter 7 the integration of an Orchestrator managing the network is considered, specially at the edge of the network, where also Fog Computing, Network Function Virtualization and Software Defined Networks are involved. The implementation of a prototype is proposed to demonstrate the feasibility of this approach and an analytical Markov model is presented to estimate the end-to-end delay of flows.

PUBLICATIONS

Conference Proceedings

- A Lombardo, D Reforgiato, V Riccobene, G Schembra. "Modeling Temperature and Dissipation Behavior of an Open Multi-Frequency Green Router." IEEE online Greencomm 2012. IEEE, 2012.

- A Lombardo, D Reforgiato, V Riccobene, G Schembra. "A Markov model to control heat dissipation in open multi-frequency green routers." Sustainable Internet and ICT for Sustainability (IEEE SustainIT), 2012. IEEE, 2012.

- A Lombardo, D Reforgiato, V Riccobene, G Schembra. "Designing a Governor Policy for Energy Saving and Heat Control in Frequency-Scaling Green Routers." IEEE online Greencomm 2013. IEEE, 2013.

- A Lombardo, A Manzalini, V Riccobene, G Schembra, "An Analytical Tool for Performance Evaluation of Software Defined

Networking Services", SDNMO 2014, Krakow, 5-9 May 2014. V Riccobene, A Lombardo, A Manzalini, G Schembra, "Network Functions At The Edge (NetFATE): Design and Implementation Issues", GTTI, Palermo, 18-20 Giugno 2014.

- R. Bolla, R. Bruschi, F. Davoli, A. Lombardo, C. Lombardo, G. Morabito, V. Riccobene, "Green Extension of OpenFlow," Proc. of the ITC 2014 EPFI Workshop, to appear.

Journal Proceedings

- A Lombardo, D Reforgiato, V Riccobene, G Schembra, "NetFPGA Hardware Modules for Input, Output and EWMA Bit-Rate Computation." International Journal of Future Communication and Networking,

- A Lombardo, D Reforgiato, V Riccobene, G Schembra, "Design of a Temperature-Constrained Governor to Save Energy in an Open Multi-Frequency Green Router", International Journal of Business Data Communications and Networking.

- A Lombardo, V Riccobene, and G Schembra. "Design of a Traffic-Aware Governor for Green Routers.", Journal of Electrical and Computer Engineering, Volume 2014 (2014), Article ID 683408, 12 pages.

- R Bolla et. all, "A Northbound Interface for Power Management in Next Generation Network Devices", IEEE Communications Magazine, Volume 52 , Issue 1, 2014.

Workshops and Posters

- R Bolla et. all "Exposing Energy-Aware Capabilities in Next Generation Network Devices" IEEE International Conference on Communications Workshop 2013, 9-13 June 2013, Budapest.
- D Reforgiato, V Riccobene "FREE - Fast Reroute for Energy Efficiency" Poster, IEEE Infocom 2013.
- A Lombardo, V Riccobene, G Schembra. "An Analytical Tool to Design a Router Governor Controlling Energy Consumption and Temperature", 24th Tyrrhenian International Workshop on Digital Communications.
- Alfio Lombardo, Antonio Manzalini, Vincenzo Riccobene, Giovanni Schembra, "An Open Architecture for Software Defined Services at the Edge", EuCNC 2014, Bologna, Italy, 23-26 June 2014.
- McGrath, M. J., Riccobene, V., Petralia, G., Xilouris, G., Kourtis, M. A. (2015, May). "Performant deployment of a virtualised network functions in a data center environment using resource aware scheduling". In Integrated Network Management (IM), 2015 IFIP/IEEE International Symposium on (pp. 1131-1132). IEEE.

Whitepapers

- A Manzalini, R Saracco, C Buyukkoc, P Chemouil, S Kuklinski, A Gladisch, M Fukui, W Shen, E Dekel, D Sol-

dani, M Ulema, W Cerroni, F Callegati, G Schembra, V Riccobene, C Mas Machuca, A Galis, J Mueller "Software-Defined Networks for Future Networks and Services", White Paper based on the IEEE Workshop SDN4FNS (Trento 11th-13th November, 2013) 29th January 2014, available on line at: http://sites.ieee.org/sdn4fns/files/2014/02/White-Paper-IEEE-SDN4FNS-2014˙02˙05.pdf

CONTENTS

LIST OF FIGURES

CHAPTER

ONE

INTRODUCTION

It does not sound as a new concept and, in fact, it is not. The words "sustainable development" are being on everyone's lips from many years now, especially since the 1987, when United Nations (UN) defined the sustainable development and the initial goals around it in the report "Our Common Future - Towards Sustainable Development" [1]. This report define the Sustainable development as follows:

"Sustainable development is development that meets the needs of the present without compromising the ability of future generations to meet their own needs. It contains within it two key concepts:

- *the concept of 'needs', in particular the essential needs of the world's poor, to which overriding priority should be given; and*
- *the idea of limitations imposed by the state of technology and social organization on the environment's ability to meet present and future needs"*

New approaches to development have been pursued and defined in the last thirty years. In 2000, eight Millennium Development Goals (MDGs) [2] have been defined and adopted by UN and set targets to reduce poverty and assure satisfied basic needs (such as food, water, health and education) all over the world.

There is still a lot of work to do towards the achievement of the goals: this year the United Nations adopts a Post-2015 Development Agenda, based around new Sustainable Development Goals (SDGs) [3], covering the plan for the next fifteen years, up to 2030: this will make start a new era of the international development.

One of the most important and pervasive argument related to this is the call for public-private cooperation to "make available the benefits of new technologies, especially ICTs" [4]. In the last decades ICT has strongly contributed to the support of the previously mentioned goals enabling access to information, increasing the distribution of educational resources, improving food production, facilitating decision-making helping vulnerable communities.

Further developments now underway mean that ICTs, and particularly Internet, will have even greater impact on development implementation and outcomes over the next fifteen years. In [5] the Internet Society [6] (ISoc, an international organization that promotes Internet use and access) refers to these, highlighting the implementation and the monitoring of the sustainable development, the exploitation of big data for development and the definition of new multi-stakeholder approaches to the sustainable development.

However, if on one hand Internet and the telecommunication technologies will play a crucial role in support the future sustainable development, on the other hand it is necessary to follow for these technologies a development strategy which is sustainable it-self. In other words, the question is: how will be possible to develop Internet and ICT satisfying *"the needs of the present without compromising the ability of future generations to meet their own needs"*? In order to find an answer to this research question, many aspects would need to be discussed and analyzed: they would relate to the social, political, economic and technological spheres. In this thesis the focus is on the technological aspects, which will be discussed and analyzed starting from the new requirements of the Internet, discussing issues, challenges and proposing solutions related to them.

CHAPTER

TWO

SUSTAINABLE DEVELOPMENT IN TELECOMMUNICATION NETWORKS

In the last years, ISoc spent a lot of effort on the analysis of the most important factors that can lead to a sustainable development of the ICT and Internet. Among the important results achieved, which are discussed in [5], there is also the identification of a set of priorities that should be taken into account by the service providers and the operators: they are related to social and political aspects, and some of them can also benefit from the technology. The latter are of interest for this work and they are listed and described below:

- Environmental impact of Internet,
- Connectivity and access for all
- Reliability and resilience,
- Affordability.

The reduction of the environmental impact of Internet is crucial in order to preserve the current state of the environment for the future generations. The concept of connectivity for all is also essential to support equality and help the poor populations to increase quality of life and this pass through an affordable, reliable and resilient network connection.

2.1 Environmental impact of the Internet

Future generations need a sustainable society and the environmental impact is one of the most important aspects to care of and primarily impact on the quality of life. From this perspective the production of energy is one of the most impactful causes of the greenhouse gas (GHG) emissions in the world: Internet and ICT, on one hand, enable energy savings through improved efficiency, due to the virtualization of goods and services and the provision of smart systems to manage productive processes; on the other hand, ICT world is one of the fastest growing consumer of energy, (today's global Internet energy consumption is about 8% of the global production [7]) and this is going to be even worse as new paradigms, like Cloud Computing, Internet of Things, Big Data and so forth are gaining diffusion. It is therefore extremely important to better understand, quantify and qualify the impact that ICT has in terms of energy consumption and this means to plan research activities driving the reduction of energy consumption of the hardware devices implementing Internet. Some research studies carried on in the last years demonstrate that a big quantity of energy currently used by the network devices is wasted, since they are usually overprovisioned. In fact today's most telecommunications networks are provisioned for worst-case or the busy-hour load, and this load typically exceeds their median long-term utilization by a wide margin; moreover, as shown in [8], current network nodes have a power consumption that is practically constant and does not depend on the actual traffic load they face. The implication of these factors is that most of the energy consumed in networks is wasted [9] [10]. For all these reasons, addressing energy efficiency in the Internet is receiving considerable attention in the literature today [11] [12] [13] [14] [15] [16] [17] and many research projects have been working on this topic (see for example [18] [19] [20]). The novel approaches for networking should involve, besides typical performance parameters, as for example throughput, latency, jitter and packet loss probability, also the amount of consumed energy and this should be taken into account even during both the design of the devices (in terms of hardware and software) and the design of the networks (in terms of topologies, provisioning strategies, and so forth). Energy efficiency of the networks is a foundamental ingredient to drive the sustanable development of Internet and ICT.

2.2 Connectivity access for all

In the era of digital economy, participation is the keywork. A wider adoption and diffusion of Internet and ICT technologies is required for people to "participate" and access information to support the evolution and improvement of life quality. Poorer individuals have to be able to communicate and take advantage from internet-enabled devices and this requires for Internet to have the proper technological support for the really fast increasing number of users and general amount of traffic to be carried by the network.

Over the last decade Internet transformed the telecom industry and it will probably continue its great impact, as the number of connected devices will increase and new forms of communication will change how people interact with each other. Changes have occurred across the globe and new changes are expected. Widespread broadband connections are being driven by the adoption of xDSL, cable and fiber-based broadband services, thanks to projects on going around the globe, like Wacscable [21]. Moreover the massive growth of data-enabled mobile devices is dramatically increasing the number of "always-on network" endpoints. To cope with this increasing amount of traffic, Service Providers (SP) have adopted so far overprovisionig approaches, allowing the network to manage the peak-hour traffic load with no issues. However, the IP Traffic forecasts provided by CISCO in the Visual Networking Index [22] include a set of projections that are briefly mentioned below for completeness and that explain the reasons why it is no longer possible to adopt over-provisioning:

- Access Network
  - The annual global IP traffic will surpass the zettabyte (1000 exabytes) threshold in 2016, and the two zettabyte threshold in 2019 when will be threefold with respect the current traffic amount (CAGR of 23% from 2014 to 2019).
  - Broadband speeds will double by 2019. By 2019, global

fixed broadband speeds will reach 43 Mbps, up from 20 Mbps in 2014.

- Metro/Transport Network

  - The busy-hour (the busiest 60 minute period in a day) Internet traffic grows faster than average Internet traffic. It will increase by a factor of 3.4 between 2014 and 2019, while average Internet traffic will increase 2.8-fold. This means that the variation over the day will represent a bigger problem than the normal management of the network and requires an increment of dynamicity of the orchestration of physical network resources.
  - Metro traffic will surpass long-haul traffic in 2015, and will account for 66 percent of total IP traffic by 2019.

- Mobile Networks, Machine to Machine (M2M) and Internet of Things

  - The 67% of IP traffic will be generated by non-PC devices (such as TVs, tablets, smartphones, and M2M modules) by 2019, (with respect the 40% of 2014). Moreover, traffic from wireless and mobile devices will exceed traffic from wired devices by 2019.
  - The number of devices connected to IP networks will be three times as high as the global population in 2019, which means about three networked devices per capita.

Provide the proper infrastructure and services for future users means to provide not only a carrier that moves the user data across the world, but also to process, store and analyse such a data. It will be required a flexible and scalable infrastructure capable of adact its processing capabilites according to the user request, keeping the costs low. To support this evolution, the cloud computing paradigm is gaining momentum in the last decades and relies on concepts like virtualization, resource sharing and multitenancy. In the last years, the concept of Fog Computing is also gaining attention: it consists in moving the computational resources of the cloud computing towards the user's location, outside the Data Centers (DC). This allows to reduce the latency between the source of the data (the user) and the processing of data (the DC). Those approaches are considered here crucial bricks to be used to build the wall of the sustainable development of Interenet and their features and evolution will be further discussed in the rest of this work.

2.3 Reliability and resilience

Another very important aspect is related to the reliability and resilience of the network services. For both fixed and mobile networks, in fact, the workload deployment orientation is changing from communication-oriented to cloud-centric. For example, Internet video to TV doubled in 2014 and will continue to grow, increasing fourfold by 2019. It will be 17% of consumer Internet video traffic by 2019, (up from 16 percent in 2014) and over half of this will be delivered by Content Delivery Network (CDN) applications [22]. Cumulatively, all these changes are having a deep technical and economic impact on transformation of telecom networks, by redefining many of the parameters that governed network design. Therefore, the "New-IP" comes with a set of new requirements to be satisfied and new conditions to cope with:

- a shift to large numbers of devices, very data consuming, geographically-shifting, and time-sensitive workloads on demand which require more efficient and dynamically adaptable networks.

- an increasing diffusion of cloud services, requiring both vertical and horizontal scaling of network functions (which respectively means to add new resources and locations) and their interconnection in a high flexible manner.

- Increment of media quality expectations and consumption patterns, necessitating new content distribution architectures (such as CDN) and more flexible flow controls at the edge of the network to improve and control performance, offer flexible value-added network services and reduce network costs.

- Increment of usage of mass scale of signalling-heavy applications, (such as M2M or IoT) which require an efficient scaling and needs to be supported by new network architectures, that can seamlessly trade off data plane and signalling plane workloads.

What it is required to scale is not only related to the bandwidth (already supported by the spread diffusion of the broadband), but it has to be mainly supported by the modification to the network architecture, according to the required dynamicity and flexibility to manage the diversity and the adaptation, exploit smart orchestration techniques and mechanisms. Traditional architecture of computer networks is now unable to cope with the rapidly increasing flow and speed of information transmission. The sustainable development of Internet requires ways to quickly scale up and manage increasing quantity and diversity of traffic demand dynamically and very fast.

For this set of actions to be successful, it is required a valuable proposition for both customers and shareholders, to open the industry to new investments. The foundation of this crucial change is next-generation IP transformation which seems to be driven by Software Defined Networking (SDN) [23], a new paradigm moving the network towards the acquisition of agility, automation and self healing features: SDN is about decoupling the software control plane from hardware data plane (packets forwarding) and moving its logic (and states) to logically centralized controllers. This, on the one hand, will increase the reliability and the resilience of the network, improving also its flexibility; on the other hand it will reduce the costs thanks to a centralized and more efficient resource management which will evolve more easily and reducing capital investment in buying new hardware. This paradigm and its more important features will be further discussed later on in this work.

2.4 Affordability

Network access for all is not only related to the performance and the agility of the network. The network connection must to be affordable for the final user that needs to relies on a stable, efficient and low cost Internet connection. The reduction of the costs is therefore a very important aspect to focus on towards the sustainable development of Internet.

There are different aspects to include into the equation, but in general they can all be included in CAPital EXpenditures (CAPEX) and OPerational EXpenditures (OPEX). CAPEX includes the acquisition of new hardware or the update of the one in place, increase the size of the Data Centers, and so forth. OPEX is about the cost to be invested in maintaining the service up and running, which includes salary of people working on the infrastructure, hardware and software maintainance, and so forth. For instance, automated operations processes (e.g. configuration of network and service equipment) could limit human intervention, reducing also wrong operations; a flexible and optimal provisioning of network functions and services could reduce personnel and equipment costs and allows to postpone or avoid undesired investment costs.

Another very important aspect to be considered is the rapid evolution of the user needs. The ossification of Internet and telecom networks is creating difficulties for Service Providers and Network Operators to develop and deploy new network functionalities, services, and management policies, which are essential to cope with the increasing dynamicity of the ICT markets: launching new services is still requiring time-consuming and expensive efforts (impacting both CAPEX and OPEX) and are preventing a rapid roll out of new businesses opportunities that would support the increasing needs. Innovation cycles of networks should be supported by processes and technologies that can make them faster, simpler and, above all, cheaper.

New paradigms like Network Function Virtualization (NFV) [24], coupled with the mentioned Software Defined Networking, if properly integrated into the current network environment, could help in fulfilling the aformentioned requirements, based on a deeper integration of the network and IT domains and all the involved operations: this could bring a deep impact on the evolution of telco operators networks and the technology providers ecosystems. In particular NFV implies virtualizing some network functions that can run on standard HW, and that can be moved and instantiated in various locations of the network.

Different deployment scenarios of SDN and NFV could be envisioned, depending on the network segments (e.g., core or edge) and, consequently, on the time horizon (e.g., medium-long term or short term). Nevertheless this overall trend of network "softwarization" is unstoppable, because of the continuous technology evolution and costs reductions, thus enabling the transition from the economy of resources to the economy of information. In the short term, shifting the focus from the core network to the edges (i.e., in the aggregation, access even up to the Users' terminals) makes this evolution possible starting by the micro scale, as it costs less, it scales smoothly and it leads to immediate revenues for operators. This creates a volume and an economic market that will drive investment outside of the network infrastructure boundary and support the advent of new communications paradigms.

All the aspects discussed in this chapter play a very important role in the network ecosystem and will be really influencing the future evolution of the network. The author believes they will drive the network and Internet across a sustanable development and the mentioned technologies and paradigms will strongly contribute to go towards that direction. The rest of the thesys will discuss the technical aspects related to them, envisioning how the network evolution could exploit that benefits from all of them, realising a heterogeneous environment. Moreover all these aspects will be further investigated and discussed lated on in this work, bringing examples and technical expressions of how to bring the adoption of these technologies into the sustainability concept.

CHAPTER

THREE

STATE OF THE ART

3.1 Energy Efficiency of the Internet

The concept of energy efficiency is not new in general-purpose computing systems. The first support of power management has been introduced with the Intel 486-DX processor and the first official version of the ACPI (Advanced Configuration & Power Interface) standard [25] has been published in 1996. More energy-saving mechanisms were introduced in the last decades and HW enhancements were made, so that general purpose CPUs could consume less power and be more efficient.

Talking about network specific solutions, a large part of network devices comes from computer technologies. The evolution has proceeded by including each time new ad-hoc hardware engines and customized silicon elements for offloading the most complex and time-critical traffic processing operations. As a consequence, it is not possible to apply generic energy-saving techniques/technologies due to the fact that network device internal elements have very different features, architectures and requirements with respect to general-purpose hardware.

From a research perspective, some work on the energy consumption of the Internet has been conducted by Gupta et al. [26] already in 2003, by Christensen et al. in 2004 [27], showing that this is a crucial aspect to work on. More recently (since about 2008) researchers, operators, and device manufacturers dramatically increased effort spent in this direction. So far, all these first efforts mainly resulted in patchy technologies and solutions, which refer to specific environments and/or protocols, and do not permit a fast and effective development and large-scale spreading of energy-awareness in telecommunication devices and infrastructures. Moreover, the lack of a standardized approach and of support of legacy technologies for network energy efficiency makes related industrial initiatives extremely costly and economically not viable.

3.1.1 Power Consumption characterization

A characterization of the energy consumption of network devices is really important in order to find potential sources of waste and design techniques that can avoid them. A paper from researchers of the University of Genoa proposes a comprehensive survey related to the power consumption of network devices [28]. In that survey, the energy consumption in wire-line networks is expressed in line to the network devices working in the different network portions (access, metro, transport and core network), since the overall energy consumption in networks arises from their operational power requirements and their density.

An interesting generalization of all the techniques analysed in that paper indicate the existence of three big classes of energy efficiency approaches in telecommunication devices and networks:

- Re-engineering:

  The re-engineering approaches aim at introducing and designing more energy-efficient elements for network device architectures, at suitably dimensioning and optimizing the internal organization of devices, as well as at reducing their intrinsic complexity levels. Novel energy-efficient technologies mainly consist of new silicon, memory technologies for packet processing engines, and novel media/interface technologies for network links. This set of approaches is mainly based on the hardware design of the components and, for this reason, it is required a very special equipment in order to verify and analyse this kind of approaches.

- Dynamic adaptation:

  Dynamic adaptation approaches are aimed at modulating capacities of network device resources according to the current traffic load and service requirements. They are generally founded on two main kinds of power management capabilities provided by the hardware level: namely power scaling states and idle states. Nowadays, the largest part of current network equipment does not include such hardware capabilities, but power management is a key feature in today's processors across all market segments, and it is rapidly evolving also in other hardware technologies

[29]. More in detail, power scaling capabilities allow the dynamic reduction of the working rate of processing engines or of link interfaces. This is usually accomplished by tuning the clock frequency and/or the voltage of processors, or by throttling the CPU clock (i.e., the clock signal is gated or disabled for some number of cycles at regular intervals).

- Sleeping/standby:

  The sleeping/standby approaches are used to smartly and selectively drive unused network/device portions to low power standby modes, and to wake them up only if necessary. However, since today's networks and related services and applications are required to be continuously available, standby modes needs to be supported with special proxy techniques able to maintain the "network presence" of sleeping nodes/components. These approaches are not mutually exclusive, and research efforts will be probably requested in all such directions in order to effectively develop new-generation green devices.

A high number of approaches based on dynamic power management techniques have been proposed to reduce the energy consumption of telecommunication networks and devices and many of them are discussed in the survey [28]. They are able to optimize the tradeoff between network performance and energy requirements. However, in order to properly exploit those techniques and technologies, it is necessary to provide a broader view of how they can be exploited in real networks, scaling the high number of devices that constitute the current network environment.

The "green" approaches presented before can be extended to the whole network by mean of Local Control Policies (LCPs) together with energy-aware routing and traffic engineering. However, the lack of a standardized representation of the energy-aware capabilities of heterogeneous networking equipment makes their deployment confusing and impractical. To this aim, for the purpose of this work it is proposed the usage of a novel framework, the Green Abstraction Layer (GAL), whose purpose is to define a multi-layered abstraction interface for the hardware and physical resources, where energy management actions are directly performed. Therefore, the GAL syntax can be exposed to the platform-independent logical representation commonly used in network control protocols. Given the internal architectural complexity and heterogeneity of many network devices, the GAL approach is based on a hierarchical decomposition, where each level provides an abstract and aggregated representation of internal components. It is also considered the existence of logical resources which can be exposed to the Local Control Policies in order to reduce the complexity of the algorithms and mask the internal hardware details of the devices.

In Chapter 5 the design of a Local Control Policy is presented: this LCP exploits a green technique exposed by the underlying hardware, which is based on CPU frequency switching. In the same section, an analytical model is provided in order to support the re-engineering of the hardware and propose the increment of the number of clock frequencies provided by the hardware device. The technique has been thus implemented using a NetFPGA board.

In Chapter 6 the Green Abstraction Layer is deeply discussed and an example of its application is presented.

3.2 Cloud and Fog Computing

In the provision of reliable and stable Internet connection and services it is extremely important to provide not only the network to move data from one point to the other of the globe, but also robust, reliable and really efficient data processing and storing mechanism supporting the high level services in a flexible and scalable manner. In this context, the cloud computing is gaining attention, as well as the Fog Computing. In this section these approaches are deeper analysed in order to better understand how they provide the base support to reduce the costs and increase the flexibility necessary to cope with the increasing number of Internet users.

3.2.1 Cloud Computing

Cloud computing is a model that is gaining momentum in the last decades. It enables ubiquitous network access to a shared pool of configurable computing resources. Cloud computing provides to the users various capabilities to store and process data hosting functionalities and features within Data Centers. One of the most important features that comes with the cloud computing is the resource sharing: this enable coherence and economies of scale, which therefore become accessible through the network. One of the goals of the Data Center owners is focused today on the maximization of the effectiveness of resource sharing: cloud resources are usually not only shared by multiple users but are also dynamically reallocated on demand: this resource sharing can therefore be converted in cost reduction. Today's cloud computing architectures are based on virtualization. The concept of virtualization it-self is not new to the ITC world. The first component subject to the virtualization was the memory, since in the 1970s memory was very expensive as computer component. In the last decades, the concept has been extended to entire desktops and it leaded, by the way, to the diffusion of cloud computing system and services. Today, there are many reasons that lead to recur to the virtualization of resources:

- Resource Sharing/Aggregation: a big physical resource can be split in smaller virtual pieces and sold to many customers or many small resources can be aggregated and sold as a bigger resource to provide higher capacity. The physical CPU of a server, for instance, is split in many virtual CPUs and serves multiple appliances from different customers, many physical hard disks can be aggregated into one single bigger virtual hard disk, etc.;

- Resource Isolation: the usage of the same resource from multiple customers could bring security issues and customers could not trust each other. For this reason is very important to provide isolation between the different appliances;

- Resource Scaling: the user requirements can change during the service provisioning (mainly due to the dynamicity of the user request) and virtualization provides an easier way (with respect the physical resources) to scale up and down resources according to the Service Level Agreement between the customer and the service provider;

- Resource Usage optimization: the dynamic adjustment can be exploited to apply usage policies according to the infrastructure costs and policies like consolidation (use the minimum number of physical resources) or distribution (balance the load equally across all the resources) can be applied;

- Resource Abstraction: virtualization aims to reduce the complexity abstracting from the underlying hardware and providing a common view of the virtual resources in a way that the customer does not need to be aware of the physical infrastructure behind the resources them-self.

The two candidates approaches to provide the features listed above in a NFV environment are: virtual machine monitors and hardware partitions.

Virtual machine monitors

A Virtual Machine (VM) is an independent software entity sharing the hardware resources with other VMs (potentially belonging to different customers), isolating the Operating System to make each appliance independent from the others. The management of VMs is mainly realized through a software program called Virtual Machine Monitor (VMM), or hypervisor: it is a program allowing multiple operating systems to share physical resources. The abstraction provided by the hypervisor gives to the VMs the illusion to have dedicated hardware on top of which the guest operating system is installed. For this reason, for each physical resource, such as physical CPU cores (pCores), physical Network Interface Cards (pNICs), physical Disks (pDisk), etc., there is a correspondent virtual resource, (like vCores, vNICs, vDisk). There are three main different approaches that can be used by a hypervisor: Full virtualisation - the hypervisor emulates the hardware device providing an emulated virtual version of it; the guests Operating Systems (OSs) interact with virtual devices providing the same functionalities and interfaces than a real one and the hypervisor is in charge of the binary translation of the instructions and operations required by the VMs against the virtual devices in real instructions and operations to be executed on the physical hardware. This often means to introduce a negatively influence on the performance.

- Para-virtualisation: the hypervisor modifies guest operating systems in such a manner that the binary translation is no more needed. The guest OS is recompiled prior to installation inside a virtual machine. If on the one hand this approach can offer potential performance advantages, on the other hand it requires the use of specially modified operating system kernels and is only suited to open source OSs;

- Hardware-assisted virtualisation: the hypervisor is supported by the underlying hardware that provides specific instructions for virtualisation support. This solution is simpler and gives better performance in comparison to other solutions, but it requires hardware with specific capabilities to be used.

Some examples of commercial and open source hypervisors are: VMware ESXi [30], KVM [31], Xen [32], Microsoft HyperV [33].

Hardware partitions

The hardware partitioning splits physical machines into different partitions and each one can execute a different appliance. This approach has its major application in the container-based virtualization in which the operating system's kernel runs on the physical hardware and several isolated VMs rely on top of it. The isolated guests also are called cells or containers. All the guest systems use the same operating system as the host OS, but each guest has its own set of resources and runs in isolation with respect the others. One of the main advantages of this approach is that the overhead due to the hypervisor is avoided. On the other side, this approach is not generally applicable to all the circumstances and introduces some security issues related to the fact that all the virtual appliances share the same kernel instance. Some important emerging technologies that use this approach are LXC [34], Docker [35] and Jailhouse [36]. Research works has been conducted to compare the hypervisors and the containers [37] , [38].

Software Defined Infrastructure

Nowadays a new model of cloud computing is gaining momentum: it goes under the name of Software Defined Infrastructure (SDI). SDI technically means a computing infrastructure which is entirely under the control of software and operator or human interventions are not required. In this model three main domains have been individuated and they are briefly discussed in the followng:

- Software Defined Compute (SDC) - SDC is related to virtual compute resources. Particularly, it is intended as an open and smart control plane for compute resources that can be discovered, monitored and assigned to specific VNFs in order to meet the SLAs and satisfy the Infrastructure provider requirements. It is supported by centralized compute control capabilities in charge of interaction with compute resources and dynamic allocation of VNFs to them in order to meet the service requirements.

- Software Defined Network (SDN) - Originally, SDN focused on separation of the Control Plane (CP) of the network, (which makes decisions about traffic steering across the network), from the Data Plane, (which is in charge of the actual forwarding of packets from place to place). In an SDN deployment, rules for packet handling are sent to the switch from a centralized controller; each switch queries the controller for guidance and information about traffic to be handled and network resource configuration, usage and statistics. Today, this allows network automation and supports network resource orchestration, opening up the network domain to a higher level of abstraction.

- Software Defined Storage (SDS) - SDS is an approach to data storage in which the entity that controls storage tasks is decoupled from the physical storage hardware. It allows storage resources to be used more efficiently and their administration can be simplified by automated policy-based management. Potentially, a single software interface could be used to manage a shared storage pool that runs on commodity hardware.

The convergence of those three domains is required in order to provide a complete management and orchestration of the infrastructure resources in a synchronized and automated fashion and this is probably one of the most crucial and complex challenges to address for this new way to provide cloud computing in order for the infrastructure providers to satisfy the user requirements, reducing the costs and maximizing the resource utilization.

3.2.2 Fog Computing

Fog Computing is a paradigm that extends Cloud computing and services to the edge of the network. Similarly to what Cloud does, Fog provides data, compute, storage, and application services to end-users [39]. This vision has been recently released by Cisco to enable applications on billions of connected devices, part of the Internet of Things (IoT), to run directly at the network edge [40]. Customers can develop, manage and run software applications at the edge. The concept of edge is still very wide: some people consider the possibility to run such custom application on top of networked devices, like router and switches, others propose the creation of so called "micro data centre": very small DCs composed by one or two racks that offer storage, compute and network functionalities. This open application environment encourages more developers to bring their own applications and connectivity interfaces close to the user, with the great advantage of reducing the latency between the data source and consumer (the user) and the place where the data is processed and stored. This infrastructure allows applications to run as close as possible to sensed actionable and massive data, coming out of people, processes and things. Therefore, Fog can be distinguished from Cloud by its proximity to end-users, the dense geographical distribution and its support for mobility [41]. As Fog computing is implemented at the edge of the network, it provides low latency, location awareness, and improves quality-of-services (QoS) for streaming and real time applications. Moreover, this new infrastructure supports heterogeneity as

Fog devices include end-user devices, access points, edge routers and switches. The Fog paradigm is well positioned also for real time big data analytics, supporting densely distributed data collection points.

In Chapter 7 it is discussed an example of how the combination of Cloud and Fog computing and other new approaches related to the network can be exploited to improve the flexibility, reduce the latency, increasing the Quality of Service and reducing the costs of the connection.

3.3 Network Function Virtualization

Network Function Virtualization (NFV) [42] has been recently proposed to increase the network service provisioning flexibility and manageability, reducing time to market and increasing service deployment automation. At the basis of this new paradigm, virtualization technologies and hardware commoditization decouple the specific software implementation of the network functions from the hardware on which those functions run. More in general, NFV embraces both networks and DCs, whose infrastructures have become increasingly complex to be managed and extremely heterogeneous over the last decade. Many operations are still performed manually, increasing the time needed to satisfy customers by weeks. Additionally, there is a lack of particular knowledge of which resources are involved and how in such operations. In this scenario, automation of management and orchestration within such an infrastructure plays an increasing important role and is one of the main advantages that NFV is bringing. All this includes an understanding of business application requirements and the implementation of strategies and methodologies to satisfy those requirements in real time. This is one of the main reasons leading the current evolution of the networks towards a "softwarization" of the services, also supported through open standards, and adaptive policy-based automation across compute, storage, and network resources: with a software-defined environment it will be possible to formally describe the customer expectations and translate them into automated deployments and lifecycle management of the services, also meeting peak demand and preventing performance degradation. This software-defined environment will ultimately provide an application-aware infrastructure capable to capture workload requirements, provide policy-based automation and optimize resource scheduling and scaling. To build this vision, the three most important pillars are:

- An open virtualization environment
- Mechanism for scheduling and elastic scaling optimization
- Application-aware infrastructure

3.3.1 Open Virtualization Environment

Opening up hardware capabilities exposing them into open frameworks is the first step in order to build an agile, responsive, and flexible IT infrastructure to support VNFs: it is important to embrace and manage all compute, storage, and networking resources and provide open interfaces for discovery and management, defining an integrated framework to interact with them: the domain integration has to converge into software-defined platforms that enable choice, flexibility, and interoperability across the whole network. Three main domains to be integrated in such a software-defined model are the compute, the network and the storage in the form of the SDC, SDN, and SDS as already stated before.

3.3.2 Scheduling and elastic scaling optimization

Since the real time automation of workloads is on the road to support real time adaptation, the fluidity of the network is becoming very important: the environment has to be ready for fluidly adjust the components building services according to changing demands, which means increase dynamicity through auto-adaptation. This involves many aspects related to the lifecycle of the services and the infrastructure: ultimately, an NFV infrastructure has to instantiate VNFs in the right locations and at the right time, scaling the performance according to the requests and mapping them on the available hardware resources. For networking applications, taking into account network performance parameters, such as latency and throughput, is crucial and makes the management even more complex, given to the fact that the Infrastructure could be distributed across wide geographic areas.

3.3.3 Application-aware infrastructure

Once organizations have an elastic, scalable infrastructure, they can start applying new higher level methodologies to define their workloads in terms of components (such as application servers and databases) and infrastructure (firewalls, virtual machines, storage, etc.) which means introduce the concept of application awareness. Application awareness is about the optimization of the network resources in order to meet the specific performance requirements (bandwidth, delay, jitter, etc.) of the overlay applications. Good examples of such optimization could be: the usage of dedicated hardware resources, the choice of specific paths for mission critical and delay-sensitive applications, the provisioning of high-priority Service Levels Agreements (SLAs) for video conferencing and medium and low-priority SLAs for email traffic. While application awareness for Layers 4-7 already exists within appliances focused on applications, such intelligence in basic OSI layer network elements (Layers 1-3) is missing. The business models that separate application providers (like Netflix, Google, Yahoo, etc.) from the service providers owning the network infrastructure is now rising as a problem for the service providers: the network infrastructure investments made by the latter are actually monetized by the former. To support the application-awareness, this trend needs to be inverted and the infrastructure needs to be aware of the application specific requirements on top of it: this will give to the infrastructure the chance to exploit the open virtualization environment provided by SDC, SDN and SDS in order to meet the user requirements in a more efficient manner. One of the most appropriate mechanisms to bring this vision into reality is the inclusion of intelligent orchestration mechanims into the network enviroment.

3.4 Software Defined Networks

Today's applications that commonly access geographically distributed databases and servers through public and private clouds require extremely flexible traffic management and access to bandwidth on demand. Moreover the explosion of mobile traffic is leading to new requirements: the Bring Your Own Device (BYOD) trend, among the others, that requires flexible and secure networks from which users expect on-demand access to applications, infrastructure, and other IT resources. To be included into the equation also the new paradigms that are entering the network environment, like big data and Internet of Things: more bandwidth will be required which in turn will require massive parallel processing providing a constant demand for additional capacity and any-to-any connectivity. In trying to meet these networking requirements by evolving computing trends, network designers find themselves constrained by the limitations of current networks: the ossification of the network, the inability to scale according to the increasement of traffic requirement and the vendor dependence due to the lack of open interfaces to tailor the network to their specific environments are only examples. The sustainable development of the network is based on the provision of a more scalable and flexible environment, which needs to be supported by the proper technological solutions. Software-Defined Networking (SDN) [43] is an emerging network approach which provides to the network dynamicity, manageabilty, cost-effectiveness, and adaptability in order to make the network able to deal with fast changing traffic patterns, depending on the dynamic nature of today's applications. The main novelty introduced by SDN is about decoupling the network Control Plane (CP) and forwarding functions (Data Plane) enabling the network control to become directly programmable and the underlying infrastructure to be abstracted for applications and network services. Coupling SDN with NFV, extend the programmability feature to the Data Plane as well, which means to increase the network flexibility. In an SDN network, control is directly programmable because it is decoupled from the Data Plane. The abstraction of the control from the forwarding lets efficient algorithms based on centralised policies dynamically adjust network-wide traffic flows to meet changing needs. The network intelligence is further improved by the provision of a global and complete view of the network to the network applications through the controller it-self. In order to implement the communication between the controller and the network devices needs to be implemented in a proper manner, in order to guarantee a secure and stable communication and to allow the controller to program the devices with the required level of flexibility. When this is implemented through open standards, SDN simplifies network design and operations because instructions are provided by SDN controllers instead of multiple, vendor-specific devices and protocols. From this perspective, the OpenFlow protocol [44] is the one that gained momentum in the last few years: it is an open protocol defined and supported by the Open Networking Foundation [45], which provides the specification for the architecture of the network devices (called "OpenFlow switches") and for the messages to be exchanged between controller and switches in order to configure the traffic flows across the network.

There are still many open questions from a research perspective that are related to the SDN domain which are part of the ongoing research. A comprehensive survey about SDN which includes also the research challanges is presented in [46].

CHAPTER

FOUR

A NEW ARCHITECTURE FOR THE SUSTAINABLE DEVELOPMENT

In this section it is proposed an architecture that includes all the technologies and the approaches mentioned in the previous section and suggest a possible way to integrate them. This can transform the network making its asset closer to the principles of the sustainable development. This section will focus on the explanation of the general architecture starting from the single elements to be integrated, and proposing at the end the integrated architecture. It will be further supported by the following chapters that will be Proof-of-Concepts and analytical models demonstrating the feasibility of the architecture here exposed.

4.1 Green Mechanisms in today's network devices

In the next future "green" network devices are expected into the market to allow on the network devices the presence of different power states [47]. These power states will change according to the input traffic: the devices will be characterized by a high energy consumption when they will need to cope with high traffic volumes. Conversely, the power consumption will be lower when the input traffic is lower. Different approaches have been proposed in literature to reduce the energy consumption of network components and they have been analysed and discussed in previous sections. The three big categories individuated so far are: re-engineering the circuitry, introducing sleeping states and introducing power scaling mechanisms as discussed in Chapter 3. The first category is delegated to chips manufacturer and it is considered an increasingly complex activity, since if on the one side the technology improves, on the other side the request of reducing the power consumption push even stronger. In fact the increasing user traffic and router capacities are two of the main reasons for the high power consumption of the network and they are not compensated by a corresponding increase in silicon energy efficiency. The increasing energy consumption in the last few years essentially depends on both new services being offered and incressing data traffic volume, which follows Moore's law, by doubling every 18 months [48]. To support new generation network infrastructures and related services, the number of devices and the architectural complexity increase (for instance, high-end IP routers have a capacity increase factor of 2.5 every 18 months

[49]). On the other side, silicon technologies (e.g., CMOS) improve their energy efficiency with a lower rate (increasing of a factor 1.65 every 18 months, as suggested by Dennard's scaling law [50]).

About the other two classes of mechanisms, one is based on putting network components to sleep during idle intervals, reducing energy consumed in the absence of traffic, which is related to the presence of so called Low Power Idle states in the hardware device; the other one is based on adapting the rate of network operations to the amount of traffic really processed, reducing the energy consumed when actively processing packets, which relies on hardware Power Scaling states. The hardware components (the processing units, the NICs, etc) can provide different power states. Rate adaptation in particular is usually achieved by scaling the processing power according to the data rate a network device has to process and or manage; to this purpose the clock frequency driving the network device processing units can be modified according to the input data rate [17]. The energy aware technique to be used in a green device depends on a number of parameters, including the role of the device in the network, the incoming traffic profile, the hardware complexity and the related costs with respect to the energy we can potentially save and the QoS we want to guarantee to the users [51]. The only existence of power scaling and low power idle hardware states, however, do not guarantee a good enough solution from a power management perspective: it is also necessary for the system to change state in the right way and at the right moment and to achieve this target it could be necessary the usage of smart control policies. For very simple devices it is easy to think to possible solutions, but a scalable solution would be required for big and complex devices that expose a huge number of different components and each of them with different power states of different kinds, correspondent to different performance level. A proper definition and implementation of Local Control Policies is required. They are algorithms running locally on the device in order to implement policies that control the current state of the hardware components. A concrete example of these is reported in Chapter 5 where the design of a Local Control Policy to manage internal power states is proposed and the implementation on a real network device, the NetFPGA, has been proposed as PoC to show how it is possible to provide a clock frequency scaling mechanism and manage different traffic levels with different performance and related power consumption.

Generally speaking, a LCP needs to have a clear view of the current status of all the components and also of the ones available in the hardware platform, then it will implement the mechanism to change the state of the system between the available once.

In order to address this problem, the Green Abstraction Layer (GAL) has been proposed. The GAL is a standard that defines a set of primitives and recommendation for the exposure of the "green" capabilities of a device. Further details are discussed in Section 4.1.1.

4.1.1 Green Abstraction Layer

The adaptation of the power consumption profile to the actual traffic the devices have to process can be realized by providing power management primitives into the hardware platforms of networking devices which mainly represents hooks to switch the state of the system. This is very similar to features currently available in general-purpose computing systems (e.g., ACPI (Advanced Configuration and Power

Interface) [25]), which interact with control policies provided by the Operating System whose goal is to find a system configuration that consumes no more power than necessary (i.e. Linux governors).

In order to introduce the GAL nomenclature, an Energy-Aware Entitiy (EAE) is a device or, more generically, a hardware component supporting different power states, called Energy-Aware states (EASs). An EAS is an operational power profile mode provided by the entity that can be configured by control plane processes. To assure the correct exchange of information between the entity and control plane processes, the EAS must be represented as a complex data type, which contains information about power consumption, performance, available functionalities, and responsiveness of the entity when working in such a configuration. This information is obviously necessary to higher level control policies in order to decide which EAS provides the desired trade-off between power consumption and network performance.

The GAL architecture was conceived as a modular and easily extendable software framework, which provides interface capabilities towards heterogeneous hardware and multiple hierarchical interfaces towards control processes, in order to set energy configurations of EAEs at various levels of the internal architecture.

This architecture allows the reduction of the power consumption of the network, avoiding eventual infuence on the performance, and it applies network wide mechanisms that scale in a hyerarchical manner reaching a reduction of the power consumption. Telling such an architecture using a bottom up approach, the power management primitives that reside at the lowest levels of the hierarchy are the ones responsible for managing the desired hardware operating behaviour. Thus, a first instance of the GAL should be used for interfacing the lowest level

EAEs with some control-plane processes implementing HW-specific optimization strategies. Then, at intermediated levels (e.g., line-card or chassis), new LCPs (one for each entity at that level) are needed to orchestrate the settings and the operating behaviours of underlying EAEs, and to expose a synthetic and aggregate set of operating characteristics and available configurations (of the line-card, or of the chassis) to higher levels. This process terminates at the device level, where the highest LCP handles the high-level configuration of the device and exposes a simplified view of it at a higher level.

The result of such hierarchical approach consists of a tree, whose root node is represented by a high-level LCP, whereas leaf nodes correspond to HW elements. The GAL is responsible of the interface among the tree nodes.

It is envisioned that each device will be capable to implement its-own independent control loops, (i.e. LCPs). According to the specific features of the local device, the LCP may dynamically orchestrate the configuration of internal components (e.g., line-cards, link interfaces, network processors, etc.) to meet the desired QoS with the minimum consumption. However, when each device independently performs energy optimizations, the overall network consumption might result much higher than in the case of cooperation among nodes and/or end-to-end network QoS requirements might be not satisfied. For instance, moving traffic flows along different network paths could reduce the load of some nodes and trigger LCPs in a more efficient manner. This issue can be overcome introducing Network Control Policies (NCPs) which are policies that refers to a bigger part of the network, not only to the device. A NCP could be implmemented for instance either as routing protocols or as centralized traffic engineering controllers.

From a functional point of view, NCPs can make decisions about the power profiles that are available in all network nodes, and that are locally provided by lower level LCPs through the GAL. Since, as easily arguable, the Network Control Policies could be managing a very high level of complexity, managing the traffic arrangement and also all the power states of the devices, it is necessary to propose a hyerchical organization of the archtiecture in a way that the Local Control Policies running on top of the single devices can be triggered by the Network Control Policies that manages the whole network. This also brings on the table the need to provide abstraction of the power consumption states in order to provide to the NCPs a form of aggregation of the power states available on the device in order to reduce the complexity.

When a NCP (e.g., a routing protocol) decides to change the configuration of a logical link (e.g., at the IP layer), by setting it into sleep mode, a command is sent to the device-level LCP. The highest level LCP interprets the command and looks for the physical resources related to that logical link, e.g. the physical port and all the other internal components involved in the configuration change.

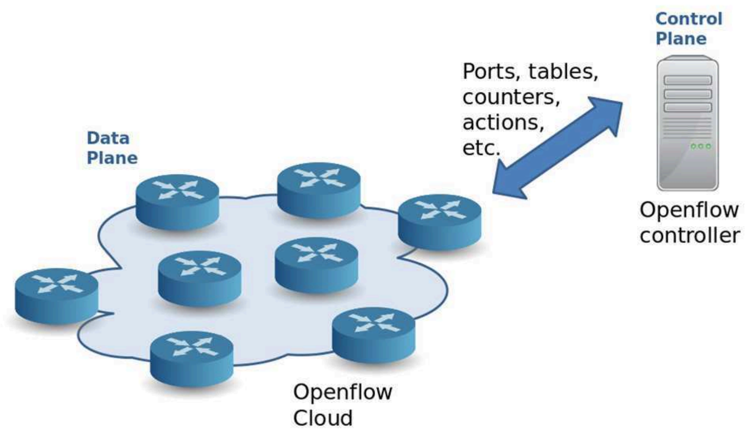
The Green Abstraction Layer has been standardized by ETSI in March 2014 [52]. Further details about the GAL can be found in [53]. The Green Abstraction Layer is a very important portion of the proposed architecture and it is going to be used for controlling the power consumption of the legacy network equipment (non-SDN) and also the SDN switches, as demonstrated in Chapter 6. In order to implement this from an SDN perspective, another given level of complexity needs to be included and this will be discussed in the next section.

4.2 Network Control Policies through Software Defined Networks

Network Control Policies could be implemented in the form or routing protocols, which means that they process global information but act locally, or by means of Software Defined Controllers, which means that a centralized controller collects information about the network devices and makes decision for them. OpenFlow (OF) [54] is a Layer 2 communication mechanism implementing the Software Defined Network (SDN) paradigm. It basically allows to determine the path of packets through the network by software. The main advantage of SDN approach is in fact the separation of the control from the data. OpenFlow provides an open protocol to program the flow tables in different switches, so a network administrator can partition the traffic in different flows, characterized by specific header field values, and each of them will receive a particular treatment in terms of routing and processing inside the network device. A common scenario of an OpenFlow network is depicted in Figure 4.1: a centralized Controller manages a set of OpenFlow switches. The former handles the control plane of the network whereas the latter manage the data plane of the network using a flow based mechanism. This architecture provides flexibility to the traffic management procedure, since the behaviour of the network can be changed, modifying the Controller behaviour.

The Controller is in charge to configure the OF switches in order to assure the end-to-end connectivity, guarantee a given bandwidth and a given set of processing actions (i.e. cryptography module, forwarding tables, VLAN tag extractor, compression module, etc.) for

Figure 4.1: *OpenFlow most frequent deployment*



each service. The flexibility provided by this approach is based on the Controller technology: it has a set of software modules which define flow allocation criteria. With all this in mind, here a way to extend OpenFlow protocol in order to carry information about the energy consumption of the devices is proposed. To the best of author's knowledge, there is no technique or metric which makes the controller aware of the power consumption of the internal hardware components of the switches: for the purpose of this work, the candidate is again the Green Abstraction Layer and its specifications.

On top of this approach, a "in-network consolidation" problem can be solved in order to to allocate traffic flows and manage different operational states of the OpenFlow switches, minimizing the power consumption of the network and, at the same time, guaranteeing a given bandwidth allocation to each service.

The proposed solution is based on defining a set of OpenFlow messages, which implement the information exchange between the switch and the Controller, and providing a possible way to calculate an optimal (or nearly optimal) solution to the ”in-network consolidation” problem.

The GAL definition also supports the possibility to define logical resources and is based on the fact that the largest part of network control protocols generally work on top of logical resources, often losing the details on how they can be mapped on internal hardware components. So, the GAL maps these logical resources on the logical entities which are exposed and are manageable by a network controller. Power management operations executed at logical resource level may involve different physical elements. Each logical entity has different Energy-Aware States and each of them is characterized by a given power consumption and a maximum supported data rate. Generically, the data rate is the maximum amount of data per second that the entity is able to manage.

In Chapter 6 details about the complete approach are presented to integrate the Green Abstraction Layer into the SDN primitives. This will be used to extend SDN in order to carry also Energy Aware States of the OpenFlow switches and communicate to the controller this information in order to combine network performance optimization with the power consumption optimization.

4.3 Network Functions at the edge

The combination of Cloud Computing, Fog Computing and Network Function Virtualization (NFV) together with SDN is the last required ingredient. The reduction of the costs and the increment of processing capacity available across the network are two really important features for the future Internet and the combination of the factor listed here can help reaching this.

Those emerging paradigms could bring a deep impact on the evolution of Telco Operators networks and the Technology Providers ecosystems, reducing costs and increasing the flexibility and the agility of the network it-self. Whereas, SDN aims at decoupling the Control Plane from the hardware Data Plane (which means the actual packet forwarding) moving its logic (and states) to centralized controllers, NFV implies virtualizing Network Functions (NFs) that can run on standard hardware and that can be instantiated and migrated in various locations of the network. Different deployment scenarios of SDN and NFV together could be envisioned, depending on the network segments (e.g., core or edge) and, consequently, on the exploitation time horizon (e.g., medium-long term or short term). In fact in the short term, shifting the focus from the core network to the edges (i.e., to the aggregation, the access or even up to the users' terminals) makes this evolution possible starting by the micro scale, as it costs less, scales smoothly, and leads to immediate revenues. This creates a volume and an economic market that will drive investment outside of the network infrastructure boundary and stimulate the advent of new communications paradigms.

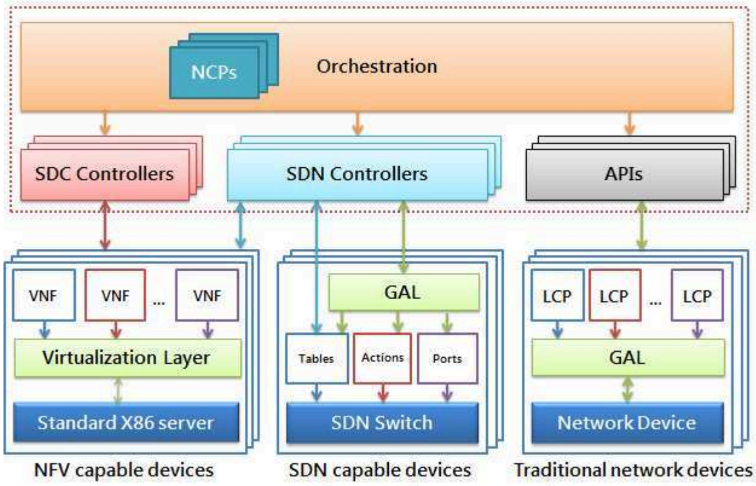
It is argued that in about five years the edges will be transformed into sort of distributed Data Centers consisting of a number of diverse and autonomous, but interrelated nodes, devices, machines [55]. For example, it will be possible to create, program, instantiate or migrate dynamically different types of virtual networks, network functions, data and services. This will allow to overcome current ossified architectures, towards the operations of sort of ephemeral (temporary) virtual networks of resources capable of self-adapting elastically and flexibly to application dynamics. To turn this vision into reality implies facing a number of challenges, for example the capability of orchestrating network functions as applications and the capability of instantiating, relocating multiple Virtual Machines (VMs) dynamically across various network locations. In this direction, an analytical approach to model and investigate the allocation of network functions on a Telco Operator's network is discussed in Chapter 7.

The author strongly believes that SDN and NFV, combined with those new Cloud/Fog computing trends can make the network application deployment more flexible and cheap. Such applications are currently provisioned as service chains, defined as sets of different devices (e.g. middleboxes [56]) connected in cascade, that process the traffic as needed. For instance, email traffic should be processed by devices that perform virus, spam and phishing detection; video and voice traffic would be processed by traffic shapers and video transcoders; web traffic by virus scanning devices, and so forth.

As evolution of such a scenario, NFV could decouple Network Functions (NF), e.g. middlebox functionalities, from dedicated hardware devices, and put them into Virtual Machines (VMs), also enabling the migration of them over the network, and in particular toward the edges of the network. Using SDN, instead, management functions (like routing) could be moved out of the network nodes and placed in centralized controller software. Coupling those approaches, a centralized entity, called Orchestrator, that has an overall and complete real-time view of the network, can manage both the NF allocation and the traffic paths between VMs and, as a result, entire service chains can be provisioned and dynamically reconfigured at runtime. Moreover, those approaches can also improve the performance of the network. Two of the major factors that affect the delay over the Internet are the propagation time that basically depends on the geographic distance, and the traffic load on links and nodes that could lead to further delay or to loss packets. Thus, moving SDN and NFV to the edge can impact also on the Quality of Experience (QoE) of the customers: shift the (Virtual) Network Functions toward the user could lead to the reduction of the propagation delay and reduce the number of devices involved in the network path, depending on the network traffic requirements.

It is going to be necessary to disentangle this scenario in which aforementioned paradigms would be deployed at the edges of the network. To this end, Chapter 7 presents an architecture that supports virtual function allocated as close as possible to the users, in order to minimize service provision latency. This architecture and its architectural components have been called NetFATE (Network Function At The Edges). In order to show the real applicability of the proposed architecture, a Proof-of-Concept has been also implemented and discussed. Moreover an analytical Markov model is proposed to estimate how this change of paradigm could positively impact the end-to-end delay for a given topology and traffic distribution, in order to support the Orchestrator work in allocating and migrating VMs on network

Figure 4.2: *Proposed architecture to support a sustainable development of Internet.*



nodes. The tools introduced by this work can strongly help network managers to improve the QoE, the flexibility and the runtime customization of the network services.

4.4 Proposed architecture of the future Internet

In this section the architecture that integrates all the technologies and protocols here discussed is presented. The architecture is shown in Figure 4.2.

The lower part of the architecture shows different kind of devices:

- the traditional network devices: which are the devices that are currently available in networks and does not support any SDN, NFV functionality;

- the SDN-capable network devices: which are the devices that supports SDN aproach (e.g. OpenFlow switches);

- the NFV-capable network devices: which are the devices that support virtualization of network functions and networking support for them, for instance through virtual switching.

Both the traditional and the SDN devices expose green capabilities in order to allow a dynamic state adaptation. They are managed through the Green Abstraction Layer. More specifically, the traditional network devices have to provide two or more Energy-Aware States and expose them through the GAL to LCPs executing locally on the device. The SDN devices will provide a set of logical resources exposing energy aware states through the GAL to the SDN controller: this will allow the Orchestrator (in the higher part of the architecture) to take it as input and manage the application of Network Control Policies on the network. The NFV devices provide a virtualization layer to support virtualization of network functions and it is managed through a Software Defined Compute (SDC) controller (e.g. OpenStack Nova).

In the higher part of the architecture, the Orchestration module orchestrates resources across the whole network infrastructure through SDC controllers, SDN controllers and ad-hoc APIs (if available for the specific traditional device) in order to apply the high level Network Control Policies.

This concept is further supported by a set of mathematical models, scientific works and Proof-of-Concepts that are described in next chapters.

CHAPTER

FIVE

DESIGN OF A LOCAL CONTROL POLICY

In the next future ”green devices” are expected in order to support different power states [47] selecting the most propriate one according to the input traffic.

Two main approaches has been proposed in literature to reduce the energy consumption of network components [12]: one is based on putting network components to sleep during idle intervals, reducing energy consumed in the absence of traffic, which is related to the presence of so called Low Power Idle states in the hardware device; the second one is based on adapting the rate of network operations to the amount of traffic really processed, reducing the energy consumed when actively processing packets, which relys on hardware Power Scaling states.

This section focuses on routers that achieve energy saving by applying the power scaling approach. Rate adaptation in particular is usually achieved by scaling the processing power according to the data rate the router has to manage; to this purpose the clock frequency driving the router processes can be modified according to the input data rate [17]. The energy aware technique to be used in a green router depends on a number of parameters, including the role of the router in the network, the incoming traffic profile, the hardware complexity and the related costs with respect to the energy we can potentially save and the QoS we want to guarantee to the users [51].

In order to provide an example of mechanisms that can help to control the energy consumption of physical network devices, here a frequency scaling router governor is proposed. As the governor currently managing the power states of the Linux kernel [57], the proposed one aims to manage the power states of a network router, dynamically changing the clock frequency of the packet processing engine, by mean of a parametric policy. Along with the governor, an analytical model is also proposed in order to support the design parameters and evaluate the governor performance.

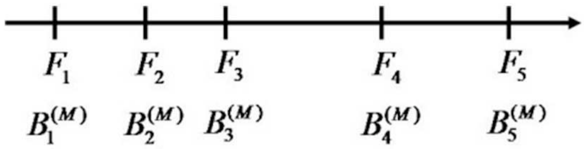
The energy aware technique to be used in a green router depends on a number of parameters, including the role of the router in the network, the incoming traffic profile, the hardware complexity and the related costs with respect to the energy that can be potentially saved and the QoS to be guaranteed to the users [51]. For the implementation of power scaling states, one of the most common ways is the clock frequency scaling of the processing unit of a router according to the data rate the router has to manage [17]. This section is mainly focused on this aspect related to the frequency scaling of the network processors, based on the incoming traffic and the consequent reduction of power consumption when low traffic rates has to be managed.

A lot of work has been done at the hardware level in order to guarantee the capability to dynamically switch the clock frequency of the hardware [58], but it is still necessary the definition of policies that make the decision about whether and when to scale the frequency. For this reason, a Governor policy is proposed in this section to support the dynamic selection of the frequency to switch to. The policy takes into account that for all the system characterized by clock frequency switching hardware, each frequency switch comes with a specific cost for the QoS, which is defined by the following parameters:

- Packet loss,
- Mean delay,
- Energy waste during frequency switching intervals.

The proposed approach aims at finding the best trade-off between energy efficiency and QoS and is very general since it can be used to limit such a router QoS degradation by only changing the particular target QoS parameter (e.g. loss probability, mean delay or energy consumption during the switching periods). For this reason, the number of transition states should be reduced. Traditional QoS parameters characterizing the router, like for example packet loss probability for output queues overflow and queueing delay, are not considered here because they are not altered by the presence of the proposed Router Governor. Additionally, a multi-dimensional discrete-time Markov model has been defined, in order to evaluate the performance of the governor by mean of analytical framework on top of the NetFPGA real case. The model will also be used to perform a parameter tuning, adapting the behaviour of the policy according to the user requirements.

Figure 5.1: *Mathematical notation example.*



5.1 Policy definition

The policy is defined considering a generic number of available frequencies, since each system could have a different number of frequencies. Let $\overline{\Phi}$ be the set of clock frequencies supported by the router CPU, and F_i be the generic i-th CPU clock frequency. For the sake of simplicity, the frequencies are sorted in such a way that $F_i < F_{i+1}$. Let us indicate the maximum bit rate that can be supported with no loss when the CPU is working at the frequency F_i as $B_i^{(M)}$. These values are sketched in Figure 5.1.

An important observation that is at the basis of this approach is that the greater the cardinality of $\overline{\Phi}$, (i.e. the greater the number of available frequencies), the higher the ability to follow the input traffic behavior with the most appropriate clock frequency, and consequently the higher the energy saving gain. However, a high number of clock frequencies could cause too frequent switches and therefore QoS degradation. For this reason, the best tradeoff between energy saving and QoS performance can be achieved by using an appropriate set Φ of clock frequencies that is a subset of $\overline{\Phi}$. For very simple processing units in which the number of frequencies is very small (two or three) the subset of frequencies to use will usually be coincident with the whole set of frequencies available. In addition, it is also important to take into account that the choice of the particular subset Φ depends on the input traffic, i.e. its mean value, its variance and its autocorrelation. In fact, if the input bit rate, due to its first- and second-order statistics, too frequently crosses the value $B_i^{(M)}$ associated to the clock frequency F_i, this clock frequency should not be used. This is another important advantage introduced by this policy: the set of frequencies can be selected dynamically and according to the current traffic statistics.

Once the set of active frequencies Φ is decided, the Router Governor has to work controlling that the QoS requirements are respected. To achieve this goal, indicating the generic i-th clock frequency in the set Φ as F_i, the Router Governor policy is defined as follows:

- RULE 1: if the clock frequency was previously set to F_i (see Figure 5.2, where $i = 3$) and the current input bit rate B_{IN} is greater than $B_i^{(M)}$ ($B_3^{(M)}$ in Figure 5.2), then the clock frequency is switched to the minimum clock frequency belonging to Φ that does not cause losses (F_4 in Figure 5.2);

- RULE 2: if the clock frequency was previously set to F_i (see Figure 5.3 where $i = 4$) and the current input bit rate B_{IN} is lower than $B_{i-1}^{(M)}$ (e.g. lower than $B_3^{(M)}$ in Figure 5.3), then it can be switched down to a value F_k less than F_i, but not less than the minimum clock frequency belonging to Φ that does not cause losses (i.e. F_i in Figure 5.3).

Figure 5.2: *Visual representation of first policy rule.*

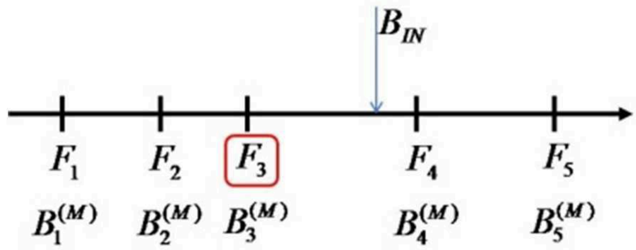
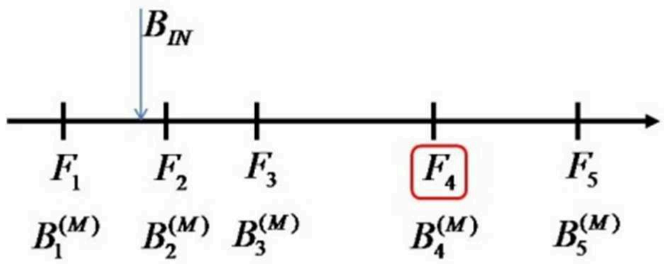


Figure 5.3: *Visual representation of second policy rule.*



However, since, as already mentioned, a frequency switch causes a QoS degradation, the switching is triggered by a probabilistic mechanism base on probability $p_G(B_{IN}, I, k)$ which is adaptive to the current input bitrate B_{IN}: the greater the distance between B_{IN} and the maximum bit rate that can be supported by the new clock frequency, the lower the risk of a new frequency switch. To this purpose, referring to the example illustrated in Figure 5.3, the switching probability is defined as follows:

- the new clock frequency is set to F_2 with a probability:

$$p_G(B_{IN}, 4, 2) = \delta \frac{B_2^{(M)} - B_{IN}}{B_4^{(M)} - B_{IN}} \tag{5.1}$$

- if the result of the previous draw were negative, and so the clock frequency were not set to F_2, the new clock frequency is set to F_3 with a probability:

$$p_G(B_{IN}, 4, 3) = \delta \frac{B_3^{(M)} - B_{IN}}{B_4^{(M)} - B_{IN}} \tag{5.2}$$

- if the previous draw is negative again, (i.e. the clock frequency is not set to F_3), the clock frequency remains F_4.

Generally speaking, if the current clock frequency is F_i and the input bit rate B_{IN} is lower than $B_{i-1}^{(M)}$, the clock frequency can be changed in the set $\{F_j, ..., F_i\}$, where F_j is the minimum clock frequency of Φ not causing loss. More specifically, the clock frequency is set to F_k, with $k \in [j, i]$, with a probability:

$$p_G(B_{IN}, i, k) = \left[\prod_{h=j}^{k-1} \left(1 - \delta \frac{B_h^{(M)} - B_{IN}}{B_i^{(M)} - B_{IN}} \right) \right] \cdot \begin{cases} \delta \frac{B_k^{(M)} - B_{IN}}{B_i^{(M)} - B_{IN}} & if k < i \\ 1 & if k = i \end{cases} \tag{5.3}$$

The term $\delta \in [0, 1]$ allows the designer to make clock frequency switches more or less rare. It is easy to argue that its value plays a very important role in the router performance. The design of the clock frequency subset Φ and the parameter δ will be assisted by the analytical model that will be described in Section 5.2. In order to follow variations of traffic statistics in a long-term time scale, they can be modified runtime according to continuous measurements done by the Router Governor.

5.1.1 The NetFPGA use case

In order to evaluate the performance of the proposed Governor, a real device, the NetFPGA, has been taken into account. The NetFPGA is an open-hardware network device that allows the user to change the bit stream, changing the behavior of the device at the hardware level. This also allows to physically measure the power consumption of the device and implementing the policy on top of the open source Reference Router project. From the power scaling perspective, the NetFPGA has only two clock frequencies available: 62.5 MHz and 125 MHz. The proposed policy has been first applied to the simple NetFPGA case, then it has been applied to a more complex case, built around an extension by means of interpolation of the power consumption base model of the board, to a case with eight different frequencies.

The NetFPGA board [59] is an accelerated network hardware that augments the functions of a standard computer. A user-programmable FPGA (with two PowerPC processors) is hosted on the board together with SRAM, DRAM, and four 1 Gbps Ethernet ports. The FPGA directly handles all data-path switching, routing, and processing operations of Ethernet frames and Internet packets, leaving software to handle control-path functions only. The open router considered in this work is setup using the Reference Router [60] implemented on the NetFPGA platform (as it is in the version 1Gbps). The reference pipeline consists of the user data path, eight receive queues and eight transmit queues (each port has a CPU and a MAC queue). The Reference Router exposes a set of hardware registers with the purpose of interacting with the hardware platform from the software: one of them is the CPCI_CNET_CLK_SEL_REG register that allows users to change the NetFPGA clock rate from 125 MHz to 62.5 MHz and vice versa. This clock frequency variation affects only a part of the NetFPGA board. The physical port module still runs at 125 MHz. However, there are shallow (2KB) FIFOs between the two clock domains that manage the clock domain crossing. This clock variation facility constitutes the base of this work. As stated in [61], when the Reference Router works at 62.5 MHz, it is able to forward traffic without loss only when the bitrate is less than or equal to 2 Gbps. The current implementation of the switching, performed by software, leads to a reset of the board for 2 microseconds, which causes the loss of incoming packets during that time window. The development of a hardware version of the frequency switch is possible in order to avoid the reset and the consequent packet loss. In any case, for all the system characterized by clock frequency switching hardware, it is possible to associate to each clock frequency switch a cost that could be in terms of either (i) loss probability, (ii) delay or (iii) energy consumption peaks. The goal of this work is to define a Governor as a software entity deciding when to switch clock frequency and to which value reducing that cost under a given threshold. The proposed model is used to both analyze the performance and determine the system parameters. In order to decrease energy consumption, but at the same time control the QoS maintaining it acceptable, to avoid waste of energy when the input traffic is less than 2 Gbps the governor receives the run-time value of the input bit-rate by a Traffic Monitor module developed on the board, and decides the clock frequency of the board according to the power management policy described below. This policy is defined taking into account that a clock frequency variation, as said so far, comes with a cost, which in case of the NetFPGA board means loss of traffic entering the router during the switching interval.

5.1.2 Power Consumption Measurement

In order to apply the model to the NetFPGA and evaluate the results, a set of power consumption measurements have been evaluated using the following configuration:

- the Reference Router project was loaded on the NetFPGA card;
- SCONE (the NetFPGA control software) was started on the hosting PC;
- Ultraview PCI Smart Extender PCIEXT-64UB card installed under the NetFPGA to access information about the power consumption, (by isolating it from the consumption of the NetFPGA host computer);
- Agilent MSO7054A series 7000 InfiniiVision oscilloscope connected to the Extender for power measurements;
- IXIA traffic generator to generate traffic.

As demonstrated in [61], the consumed power can be modeled as follows:

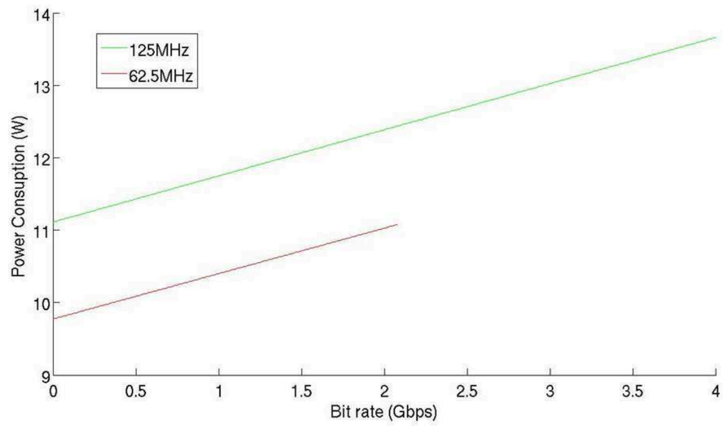
$$\Psi(f_c, B_{IN}) = P_c(f_c) + KP_E(f_c) + N_I(B_{IN}) \cdot E_p(f_c) + \\ R_I(B_{IN}) \cdot E_r(f_c) + R_O E_t(f_c) \tag{5.4}$$

where f_c is the CPU clock frequency, whereas B_{IN} is the bit rate of the router input traffic. The term $P_C(f_c)$ is the constant baseline power consumption of the NetFPGA card (without any Ethernet ports connected); $P_E(f_c)$ is the power consumed by each Ethernet port (without any traffic flowing); $E_p(f_c)$ is the energy required to process each packet (parsing, routing lookup, etc.); $E_r(f_c)$ is the energy required to receive, process and store a byte on the ingress Ethernet interface; $E_t(f_c)$ is the energy required to store, process and send a byte on the egress Ethernet interface; K is the number of Ethernet ports connected (1 to 4); $N_1(BIN)$ is the input traffic bit rate to the NetFPGA card in packets-per-second (pps); $R_1(B_{IN})$ is the input rate to the NetFPGA card in bytes-per-second; $R_0(B_{IN})$ is the output rate from the NetFPGA card in bytes-per-second.

In Figure 5.4 the resulting model of power consumption is shown, varying the input bit rate for the two available clock frequencies. When the bit rate is lower than or equal to 2 Gbps (in such a case the clock frequency 62.5 MHz may be used without any packet loss), the consumed power is less than 11 W; for bit rates higher than 2 Gbps, the clock frequency of 125 MHz has to be used, and the power consumption goes up to 14 W.

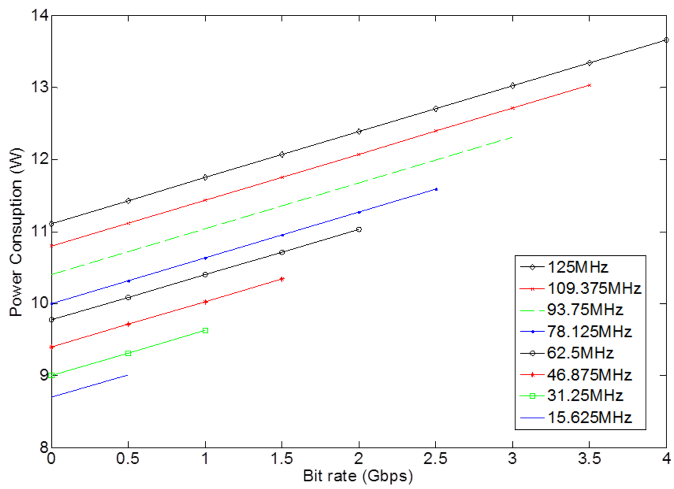
In the following sections the results obtained with the 2-frequencies NetFPGA model are be discussed, along with the results obtained on an extension of those measurements, calculated by interpolation. More specifically, the following eight clock frequencies are considered:

Figure 5.4: *Power consumption model of the NetFPGA with 2 frequencies.*



- F_1 =15.625 MHz,
- F_2 =31.25 MHz,
- F_3 =46.875 MHz,
- F_4 =62.5 MHz,
- F_5 =78.125 MHz,
- F_6 =93.75 MHz,
- F_7 =109.375 MHz,
- F_8 =125 MHz.

Figure 5.5: *Power consumption model of NetFPGA with 8 frequencies.*



The resulting power consumption model is depicted in Figure 5.5

5.2 Markov Model

In this section a discrete-time model of the system described so far is defined in order to capture the behavior of the governor policy, the power consumption of the router and the effect of the policy on the reduction of power consumption. Since it depends on the input traffic bit rate according to the Router Governor policy, the generic Markov model state is defined as:

$$S^{(\Sigma)}(n) = (S^{(C)}(n), S^{(I)}(n), S^{(S)}(n)) \tag{5.5}$$

where:

- $S^{(C)}(n) \in \xi^{(C)}$ is the clock frequency process at the generic slot n;
- $S^{(I)}(n) \in \xi^{(I)}$ represents the quantized input traffic bit rate at the generic slot n;
- $S^{(S)}(n) \in \xi^{(S)} = 0, 1$ is the indicator variable of a switch at the generic slot n: $S^{(S)}(n) = 1$ if, in the slot n, the router is switching its clock frequency.

The set of states $\xi^{(C)}$ contains the active frequencies, (i.e. all the clock frequencies belonging to the set Φ). The set $\xi^{(I)}$ contains the considered quantized input traffic values. Let us define the slot duration as the interval between two consecutive observations of the input bit rate, it will be indicated as Δ. In order to define the model time diagram, let us consider two generic states:

- $\underline{S}_{\Sigma 1} = (S_{C1}, S_{I1}, S_{S1})$ in the slot n, and
- $\underline{S}_{\Sigma 2} = (S_{C2}, S_{I2}, S_{S2})$ in the slot $n + 1$.

Let us assume the following event sequence:

- The first action at the beginning of the slot $n + 1$ is the evaluation of the new value of the input traffic bit rate. This value is obtained by sampling the bit rate values and smoothing the obtained sequence with a EWMA filter with a time constant equal to the time slot Δ.

- Then, according to the new value of the input traffic bit rate, the Governor decides the clock frequency for the new slot. Let us recall that, as said so far, a clock frequency modification determines that the router enters in the switching interval, during which some performance degradation occurs; all the clock frequency switching slots will be characterized by the state variable $S^{(S)}(n) = 1$. Let T_F be the duration of this period.
- Then, at the end of the slot $n+1$, the system state variables are observed.

After that, the generic element of the state transition probability matrix can be defined as follows:

$$\begin{aligned} Q^{(\Sigma)}_{[s_{\Sigma 1}, s_{\Sigma 2}]} &= Prob\left\{ S^{(\Sigma)}(n+1) = S_{\Sigma 2} | S^{(\Sigma)}(n) \right\} \\ &\qquad\qquad\qquad\qquad = s_{\Sigma 1} \\ &= Q^{(I)}_{[s_{I1}, s_{I2}]} \cdot \eta^{(C)}_{[s_{C1}, s_{C2}]}(S_{I2}) \cdot Q^{(S)}_{[s_{S1}, s_{S2}]}(s_{C1}, s_{C2}) \end{aligned} \tag{5.6}$$

$Q^{(S)}_{[s_{S1}, s_{S2}]}(s_{C1}, s_{C2})$ is the transition probability of the clock frequency switch indicator variable. It is defined as follows:

$$Q^{(S)}_{[s_{S1}, s_{S2}]}(s_{C1}, s_{C2}) = \begin{cases} 1 & if(s_{C2} \neq s_{C1}, s_{S1} = 0, s_{S2} = 1) \\ 1 & if(s_{C2} = s_{C1}, s_{S1} = 0, s_{S2} = 0) \\ \Delta/\overline{T}_F & if(s_{S1} = 1, s_{S2} = 0) \\ 1 - \Delta/\overline{T}_F & if(s_{S1} = 1, s_{S2} = 1) \\ 0 & otherwise \end{cases} \tag{5.7}$$

where the term Δ/T_F is the probability that the router leaves the switching period. The first two probabilities are set to 1 because they represent the probability of changing the state variable $S^{(S)}(n)$ from 0 to 1 when a clock frequency switch occurs, and the probability of maintaining $S^{(S)}(n)$ equal to 0 when the router works normally. $\eta^{(C)}_{[s_{C1},s_{C2}]}(S_{I2})$ gives the probability of a clock frequency switch depending on the clock frequency switching law used by the Governor to decide the clock frequency according to the input traffic bit rate. It is set to 0 when, according to the clock frequency switching law, it is not possible that the Governor sets the value of s_{C2} when the input traffic value is s_{I2} and the current clock frequency is s_{C1}. Following the Governor policy illustrated in Section 5.1, it is defined as follows:

$$\eta^{(C)}_{[s_{C1},s_{C2}]}(s_{I2}) = \begin{cases} 1 & if s_{I2} > B^{(M)}_{s_{C1}} and B^{(M)}_{s_{C2}} = s_{I2} \\ 1 & if s_{I2} = B^{(M)}_{s_{C1}} and s_{C2} = s_{C1} \\ p_G(s_{I2}, s_{C1}, s_{C2}) & if s_{I2} < B^{(M)}_{s_{C1}} and \\ & s_{I2} \leq B^{(M)}_{s_{C2}} \leq B^{(M)}_{s_{C1}} \\ 0 & otherwise \end{cases} \tag{5.8}$$

The term $p_g(s_{I2}, s_{C1}, s_{C2})$ is the frequency clock switching probability defined as in (1). As already specified in Section 5.1, it is adaptive with the current value of the input bit rate; $Q^{(I)}$ is the state transition probability matrix for the quantized input traffic. It is an input of the problem, because it characterizes the traffic crossing the router.

Now, from the matrix $Q^{(\Sigma)}$ it is possible to derive the system steady-state probability array $\underline{\pi}^{(\Sigma)}$ by solving the following system:

$$\begin{cases} \pi^{(\Sigma)} Q^{(\Sigma)} = \pi^{(\Sigma)} \\ \pi^{(\Sigma)} \cdot \underline{1}^T = 1 \end{cases} \tag{5.9}$$

where 1^T is a column array with all the elements equal to one. Its generic element, $\pi^{(\Sigma)}_{[\underline{s}_\Sigma]}$, is the steady-state probability of the state $\underline{S}_\Sigma = (s_C, s_I, s_S)$.

5.3 Performance parameter derivation

In this section the main QoS parameters are derived, with the purpose of both evaluating router performance and supporting Router Governor Design. First of all, let us calculate the mean power consumed by the router when the Governor applies the proposed policy:

$$P_{MEAN} = \sum_{\forall s_C \in \xi^{(C)}} \sum_{\forall s_I \in \xi^{(I)}} \Psi(s_C, s_I) \cdot \sum_{\forall s_S \in \xi^{(S)}} \pi^{(\Sigma)}_{[s_C, s_I, s_S]} \tag{5.10}$$

where the term $\Psi(s_C, s_I)$ in Equation 5.10 is a model input, and represents the power consumed when the router is loaded with an input traffic bit rate of s_I and the clock frequency is s_C. Now let us calculate the QoS parameters that can be degraded during clock frequency switching periods, according to the switching technique applied by the green router. The following three relevant cases will be considered: if the router remains frozen during the switching period and all the traffic arrived in that period is lost, as for example in the green NetFPGA reference router case, the QoS parameter to be considered is the probability of loss occurring during the switching periods. It is defined as:

$$P_{loss} = \lim_{x \to +\infty} \frac{L(m)}{V(m)} = \frac{\overline{L}}{\overline{V}} = \frac{\sum_{s_C \in \xi^{(C)}} \sum_{s_I \in \xi^{(I)}} s_I \pi^{(\Sigma)}_{[s_C, s_I, 1]}}{\sum_{\underline{s}_\Sigma \in \xi^{(\Sigma)}} S_I \pi^{(\Sigma)}_{[\underline{s}_\Sigma]}} \qquad (5.11)$$

where $L(m)$ and $V(m)$ are the cumulative number of lost and arrived bits in m consecutive slots, respectively. The term $\overline{V}$ is the mean value of arrived bits per slot, while the term $\overline{L}$ represents the mean value of bits lost per slot.

If the router remains frozen during the switching period and all the traffic arrived in that period is buffered, the QoS parameter to be considered is the mean delay suffered by the traffic arrived during the switching periods. It can be represented by the mean number of packets that arrive during a switching period:

$$\overline{D} = \frac{\overline{T}_F}{\Delta} \left[\sum_{\forall s_C \in \xi^{(C)}} \sum_{\forall s_I \in \xi^{(I)}} s_I \cdot \pi^{(\Sigma)}_{[s_C, s_I, 1]} \right] \qquad (5.12)$$

where the term in squared brackets represents the mean traffic loading the router during a switching period, while $\overline{T}_F/\Delta$ represents the mean duration of the switching period expressed in slots.

If a clock frequency switch causes a peak of energy consumption the QoS parameter to be considered is the total mean power consumption, $P^{(switch)}_{MEAN}$, defined as the sum of the mean value of the consumed power not considering the switching events, P_{MEAN}, and the mean power caused by the switches. Indicating the power consumed during a switch period as P_{switch}, and taking into account that a switch lasts for $\overline{T}_F/\Delta$ slots, the overall mean power can be calculated as follows:

$$P_{MEAN}^{(switch)} = P_{MEAN} + \frac{P_{switch}}{\overline{T}_F/\Delta} \sum_{\forall s_C \in \xi^{(C)}} \sum_{\forall s_I \in \xi^{(I)}} \pi_{[s_C,s_I,1]}^{(\Sigma)} \tag{5.13}$$

The term P_{switch} is an input of the problem, while P_{MEAN} has been derived in Equation 5.10. Another important parameter that can be derived by the mean consumed power calculated as in 5.13 is the power saving percentage achieved by using the proposed Governor policy. Depending on whether the power consumed during switches is considered or not, it can be calculated as follows:

$$\rho = (P_{MAX} - P_{MEAN})/P_{MAX} \cdot 100\% \tag{5.14}$$

$$\rho = (P_{MAX} - P_{MEAN}^{(switch)})/P_{MAX} \cdot 100\% \tag{5.15}$$

where P_{MAX} is the power consumed if no saving policy is applied.

5.3.1 Results for the default NetFPGA

In this section the results obtained for the NetFPGA are discussed. In order to also analyze which is the impact of the traffic correlation on the achieved performance, three different traffic traces are generated from a real traffic trace recorded at the ingress of the DIEEI lab has been quantized in eight different bit rate levels, ranging from 0.4 Gbit/s to 3.9 Gbit/s with steps of 0.5 Gbit/s. First- and second-order statistics of the trace, in terms of probability density function (pdf) and autocorrelation function (acf), are represented in Figures 5.6 and 5.7. Then, solving an inverse eigenvalue problem, the input traffic Markov model characterized by the transition probability matrix $Q^{(I)}$

H1	H2	H3
	(1,1) - 9.9990e-001	(1,2) - 1.0000e-004
(2,1) - 3.1569e-005	(2,2) - 9.9993e-001	(2,3) - 3.5098e-005
(3,2) - 6.7811e-006	(3,3) - 9.9994e-001	(3,4) - 4.8774e-005
(4,3) - 4.1255e-005	(4,4) - 9.9995e-001	(4,5) - 6.3636e-006
(5,4) - 1.9848e-005	(5,5) - 9.9994e-001	(5,6) - 3.8975e-005
(6,5) - 3.8314e-005	(6,6) - 9.9992e-001	(6,7) - 3.8609e-005
(7,6) - 1.3970e-005	(7,7) - 9.9990e-001	(7,8) - 8.6030e-005
(8,7) - 1.4286e-004	(8,8) - 9.9986e-001	

Table 5.1: *Transition probability matrix.*

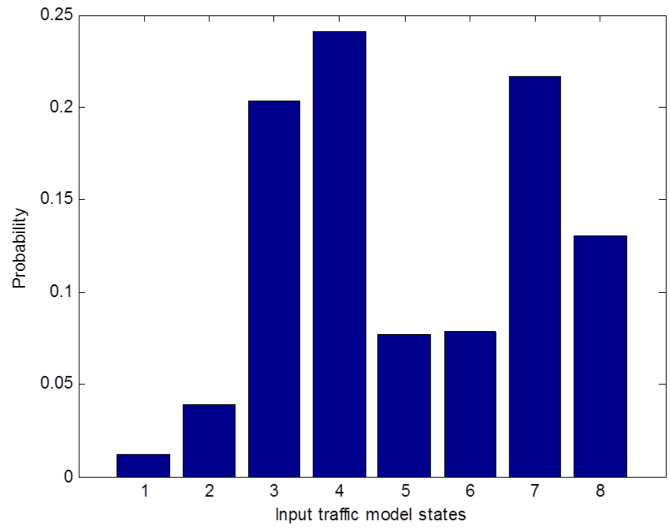
has been derived, which is a tri-diagonal matrix whose non-null elements are listed in Table 5.1. The considered traffic has a mean value of 2.54 Gbit/s and a standard deviation of 0.965 Gbit/s.

The three different cases of input traffic considered are characterized by transition probability matrices derived from the one listed in Table 5.1 by multiplying the terms of the pseudo-diagonals by a coefficient $\alpha = 10^{-2}$ and $\alpha = 10^{-3}$. The terms of the main diagonals are calculated such that the sum of each row is equal to one. In this way first-order statistics remained unchanged, while traffic becomes more correlated, as shown in Figure 5.7.

Probability distribution array of the considered traffic has a mean value of 2.54 Gbit/s and a standard deviation of 0.965 Gbit/s.

Figure 5.8 shows the relation between the loss probability and the energy saving gain obtained by applying the model to the input data described above. Clearly, when the p_G increases, the energy saving gain increases as well because higher loss probability means that the

Figure 5.6: *Probability Density Function of the traffic trace.*

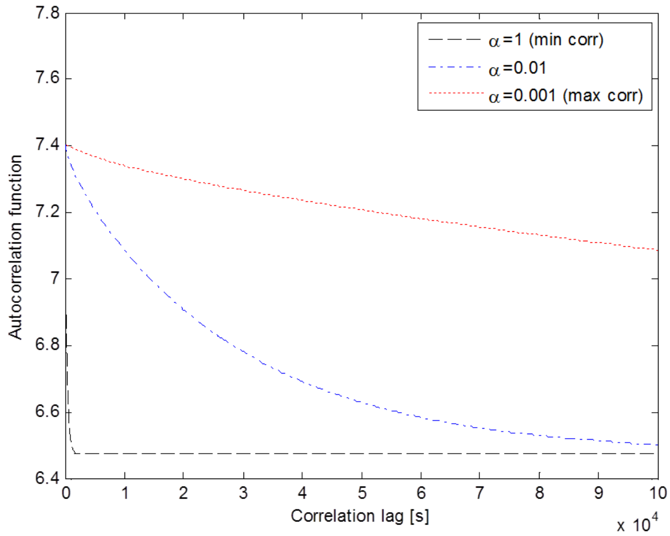


router is operating in lower frequency clock domains. These results allow to design the Governor law in order to maintain a known interval range of loss probability. These results allow to design the Governor law to maintain the loss probability within a given interval range according to the Service Level Agreement, and at the same time, to save energy up to 15%.

5.3.2 Results for the extended NetFPGA

As mentioned above, this second part of the work has the goal to design the clock frequency subset Φ and the δ probability term to be used in Equation 5.3. Applying such a switching probability, the greater the value of the δ parameter, the more accurate is the Router Governor

Figure 5.7: *Autocorrelation Function of the traffic trace.*

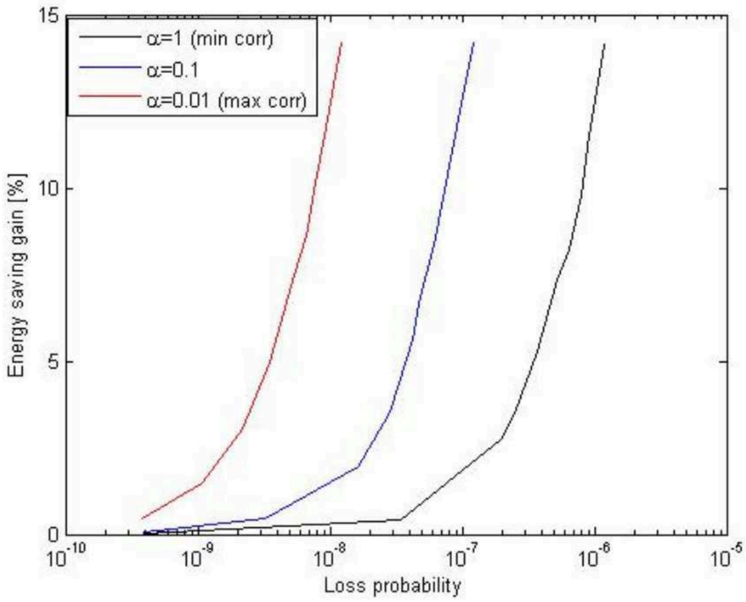


in following the input traffic bit rate variations, so obtaining higher power saving, but consequently increasing the loss probability, or more generically, the cost that comes with the clock frequency switch.

The proposed model is in this case used to solve an optimization problem, finding the subset Φ of active clock frequencies and the probability term δ which maximize the power saving gain ρ, subject to the constraint $P_{Loss} <= P_{Loss}^{(T)}$, where $P_{Loss}^{(T)}$ is the upper bound for the switching loss probability that can be tolerated, hereinafter also called target loss probability.

The same traffic traces as in the previous section have been used. In order to analyze the impact of the choice of the number of CPU clock frequencies that can be selected by the Governor, four different sets of clock frequencies ash been considered:

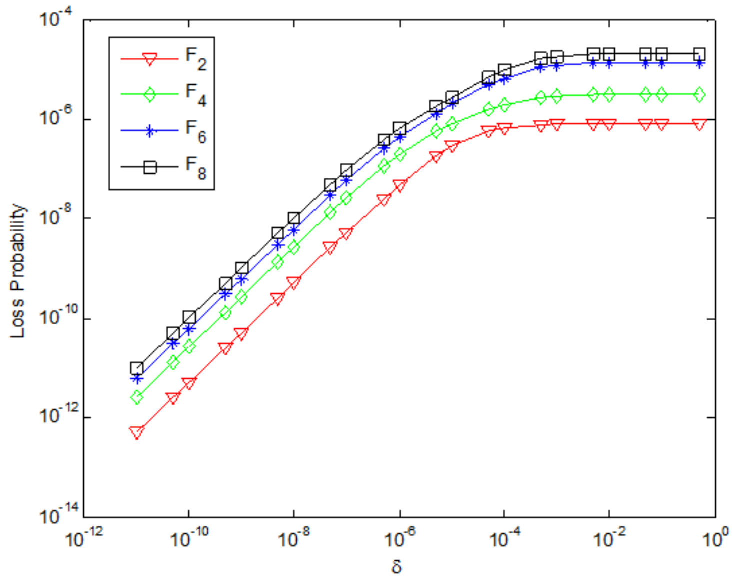
Figure 5.8: *Energy Saving vs. Loss Probability*



- F2: 62.5, 125
- F4: 31.25, 62.5, 78.125, 125
- F6: 15.625, 31.25, 62.5, 78.125, 109.375, 125
- F8: 15.625, 31.25, 46.875, 62.5, 78.125, 93.75, 109.375, 125

The first step of this analysis is the evaluation of the considered performance parameters loss probability and power saving against the probability term δ. They are shown in Figures 5.9 and 5.10. In Figure 5.9 we first notice the strong impact that has the probability term δ on these parameters. More specifically we can observe how increasing δ, i.e. making the Governor more flexible and able to closely follow

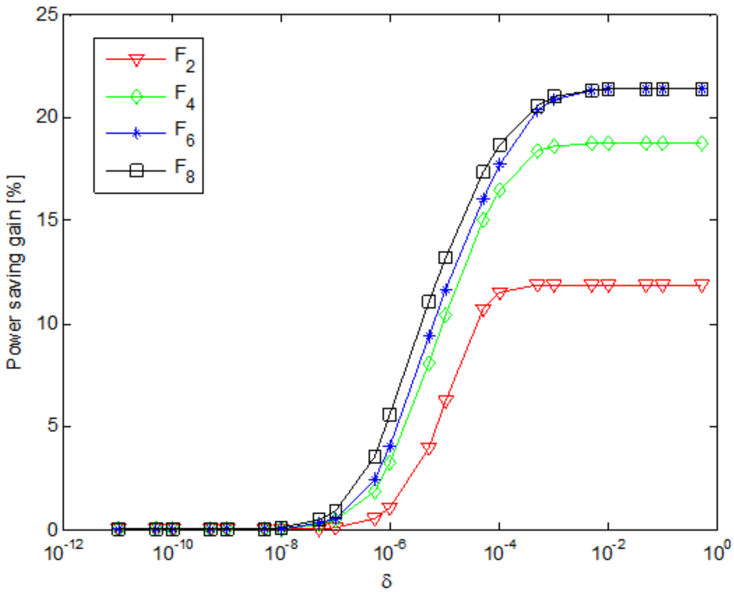
Figure 5.9: *Loss probability vs. δ*



the input traffic bit rate, a greater energy saving is achieved, but with an increasing loss probability during the switching intervals. In the same figure we can observe the impact of the choice of the number of frequencies: for example, by using only two frequencies, the energy saving gain cannot exceed 12%, while a 22% gain is obtained by using six or eight frequencies. The mean temperature follows the same behavior since it is able to reach the lowest values only by using eight clock frequencies. However, using a greater number of frequencies causes a greater loss probability that can be reduced by decreasing the value of δ.

Now, in order to simultaneously account for these performance parameters, in Figure 5.11 plotting the energy saving gain vs. loss

Figure 5.10: *Power saving gain vs. δ*

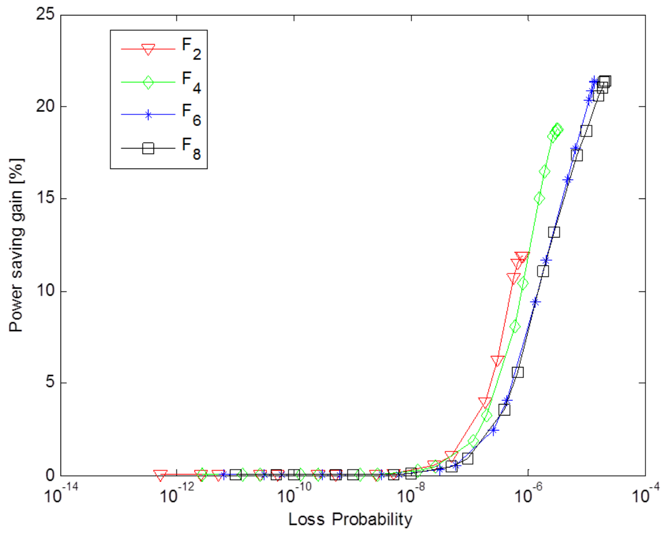


probability have been shown. The graphic is shown for the four cases of CPU clock frequency sets. The router Governor designer can use Figure 5.11 to know the maximum energy saving gain that can be achieved for a given loss probability target. Finally, in order to support the choice of the number of clock frequencies, the above results have been summarized in the bar plot shown in Figure 5.12 where each group of four bars represents the maximum energy saving gain obtained for each loss probability target. From this figure we can deduce that the set of two frequencies is more suitable for very stringent loss probability targets because it provides the best energy saving gain. On the contrary, when we need a greater energy saving gain at the expenses of loss probability, six frequencies are more appropriate.

To extend the results obtained so far, all the possible frequencies and combinations have been considered in order to check if the best configuration is different than the configuration considered so far. For this purpose, the traffic model used is constituted by a transition probability matrix $Q^{(I)}$ slightly different than the one in Table 5.1. The considered traffic has a mean value of 2.66 Gbit/s and a standard deviation of 0.946 Gbps. This traffic matrix is based on a bit rate array $\Gamma^{(I)} = [0.25, 0.75, 1.25, 1.75, 2.25, 2.75, 3.25, 3.75]$ Gbps.

From the traffic model $(Q^{(I)}, \Gamma^{(I)})$ a set of ten different models have been derived, obtained as follows: $T_i = (Q^{(I)}, \Gamma^{(I)})$, with $1 <= i <= 5$, characterized by a transition probability matrix $Q_i^{(I)}$ derived from $Q^{(I)}$ by multiplying the terms of the two pseudo-diagonals by a coefficient $\alpha \in 10^4, 10^2, 10^0, 10^{-2}, 10^{-4}, 10^{-6}$. The terms of the main diagonals are then calculated such that the sum of each row is equal to one. In this way the traffic modeled by T_1 and T_2 result less correlated than the measured traffic, the traffic modeled by T_3 coincides with the

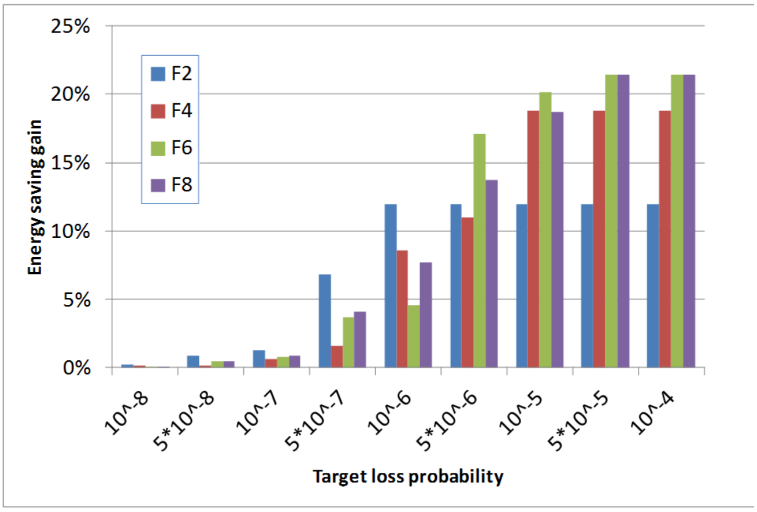
Figure 5.11: *Energy saving gain vs. loss probability for 4 set of frequencies.*



real traffic, while the other models represent more correlated traffic. $T_i = (Q^{(I)}, \Gamma^{(I)})$, with $6 <= i <= 10$, characterized by the same five transition probability matrices of the previous case, i.e. $Q_i^{(I)} = Q_{i-5}^{(I)}$, but with a bit rate array $\overline{\Gamma}^{(I)}$ achieved by mirroring the array $\Gamma^{(I)}$ of the previous case. By so doing, the new pdf is the mirror of the one previosly shown and the new mean value is equal to 1.33 Gbit/s.

The same analysis as before has been applied to the system with this new set of traffic matrices, using the analytical system model defined in Section 5.2, the loss probability and the power consumption of the router architecture discussed so far have been analyzed. More in details, 127 different frequency sets Φ have been considered, achieved by choosing from the whole set $\overline{\Phi}$ all the possible subsets containing the

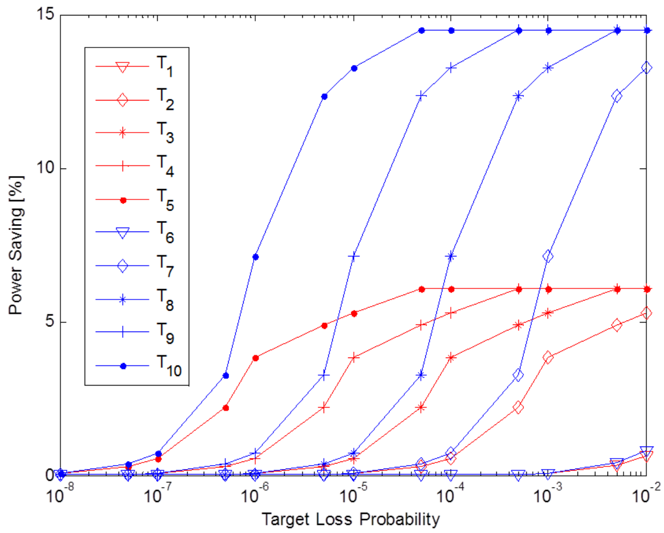
Figure 5.12: *Energy saving gain for different loss probability targets.*



highest frequency, i.e. F_8 =125 MHz. In other words, the considered subsets are: F_1, F_8 , F_2, F_8 , F_3, F_8, ..., F_1, F_2, F_8, F_1,F_3,F_8 , F_1, F_4, F_8, ..., F_1, F_2, F_3, F_4, F_5, F_6, F_7, F_8.

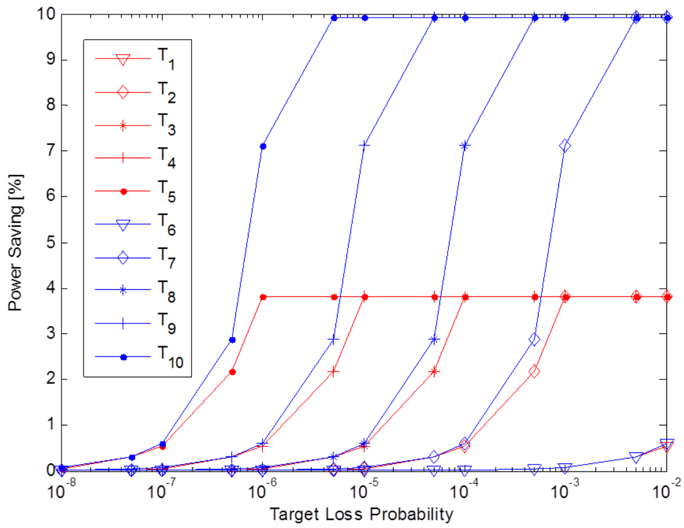
The optimization problem stated at the beginning of this section has been solved for each of the considered ten traffic models, and versus the target loss probability $P_{loss}^{(T)}$. The results are shown in Figure 5.13, where each point corresponds to the configuration (δ, Φ) that provides the highest power saving for each target loss probability and traffic model. The reader can notice that, when the value of $P_{loss}^{(T)}$ increases, the power saving for all the curves tends to an asymptotic value which mainly depends on the mean value of the input traffic. Therefore this result highlights that the maximum achievable power saving is influenced by the mean value of the input traffic bit rate.

Figure 5.13: *Maximum power saving due to the Router Governor given a maximum loss probability.*



Moreover, in the same figure it is possible to notice also that the higher the autocorrelation of the traffic, the higher the power saving for a given target loss probability. It is caused by the fact that, when the traffic autocorrelation is higher, the Router Governor can follow the traffic profile with more rare switches of the clock frequency. Now, in order to evaluate the impact that the used frequencies have on the router performance (target loss probability and power consumption), the optimization problem has be solved for fixed number of frequencies, leaving the system free to choose the best value of δ and the best set Φ of a given size. Figures 5.14, 5.15 and 5.16 show the results for the cases of two, four and eight frequencies, respectively. The reader can also notice that again the maximum achievable power saving is higher

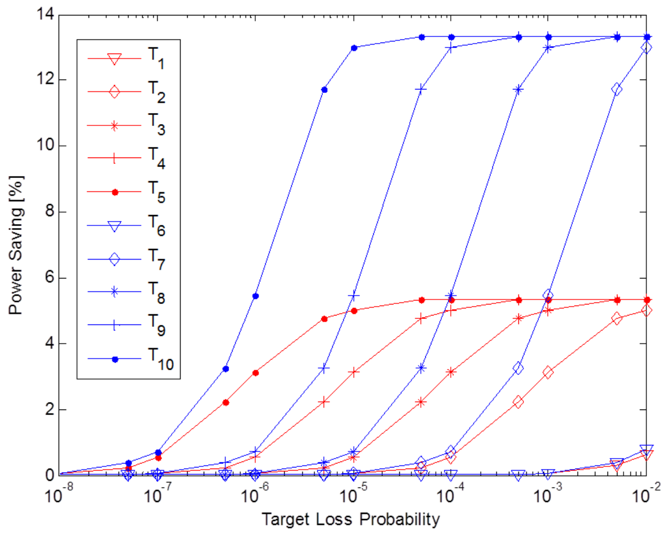
Figure 5.14: *Power saving vs. target loss probability resulted by optimization problem with 2 frequencies.*



using a higher number of frequencies; this is because in this case the router processor is able to follow the input traffic more accurately.

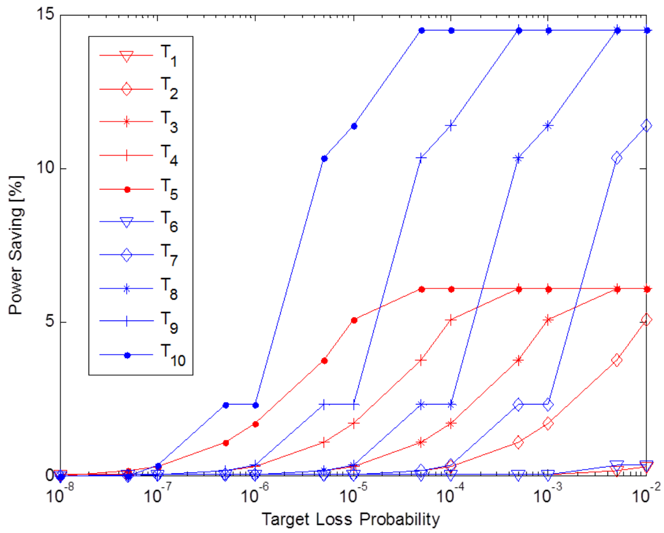
To better investigate the behavior of the Router Governor varying the frequency set and the δ parameter, Figures 5.17, 5.18 and 5.19 shows a detailed view of a subset of the cases already represented in Figures 5.13, 5.14 and 5.15. In particular, the cases corresponding to a loss probability target of 10^{-6} are taken into account. Figure 5.17 shows the results of the most general optimization problem, solved over the 127 frequency sets described so far. Figures 5.18 and 5.19 present the results achieved for the two optimization problems characterized by two and four frequencies, respectively. Such figures explore the frequency configurations and the δ parameter value selected by the

Figure 5.15: *Power saving vs. target loss probability resulted by optimization problem with 4 frequencies.*



optimization algorithm, also showing the power gain of each case. The above figures show that the Router Governor changes the subset of used frequencies according to both the mean and the autocorrelation of the input traffic. To better understand the results of the optimization algorithm, it is necessary a further analysis of those figures and, doing that, it is important to take into account that for cases T_1 and T_6, the traffic is uncorrelated and so the achievable power consumption is very low (see Figures 5.13, 5.14 and 5.15). So, the Router Governor selects a low value of δ to avoid too frequent clock frequency switches. As far as the other cases are concerned, when the traffic autocorrelation increases the number of frequencies can be augmented: in fact, if the Router Governor sets the clock frequency to a value that supports the

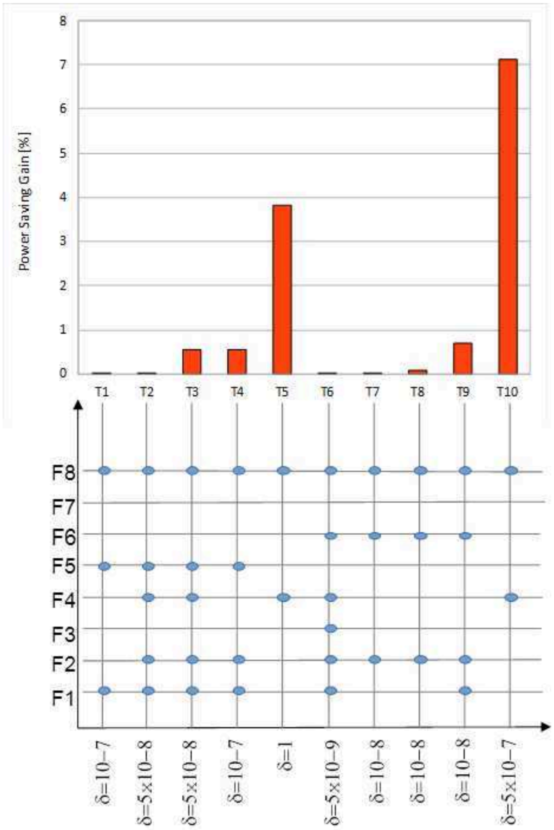
Figure 5.16: *Power saving vs. target loss probability resulted by optimization problem with 8 frequencies.*



current input traffic and such frequency remains unchanged for a given amount of time, both the loss probability and the power consumption will be positively influenced.

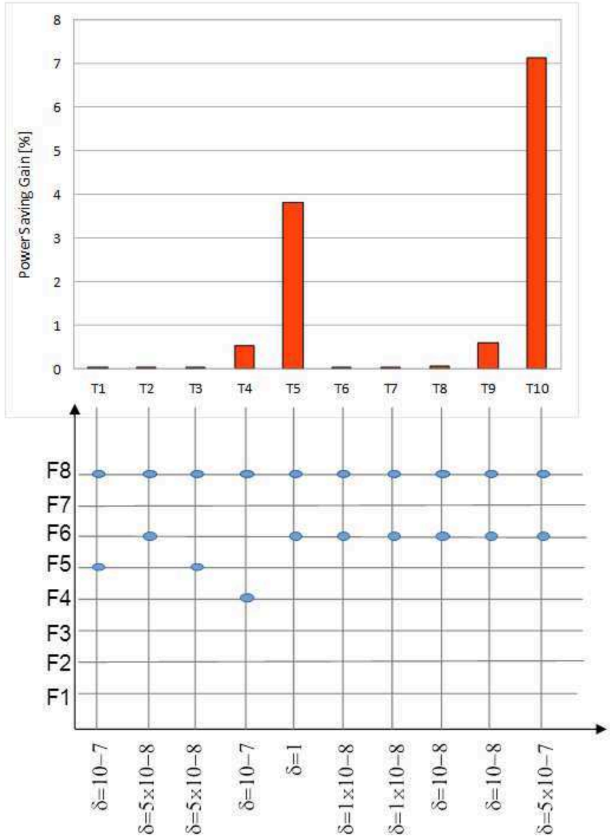
In Figure 5.17 the reader can notice that for T_2 and T_3 the optimization problem selects five frequencies and δ is equal to $5 \cdot 10^{-8}$, whereas for the T_4 case four frequencies have been selected, but the clock frequency is more free to follow the input traffic variations, since δ is equal to 10^{-7}. Instead, in the T_5 case, where the autocorrelation of the input traffic is very high, the algorithm selects only two frequencies but, since $\delta = 1$ leaves the system completely free to change between them every time the input traffic varies. Regarding Figure 5.18, same considerations can be formulated, but here we can found a

Figure 5.17: *Power saving and selected configuration (δ, Φ) corresponding to a target loss probability of* 10^{-6}.



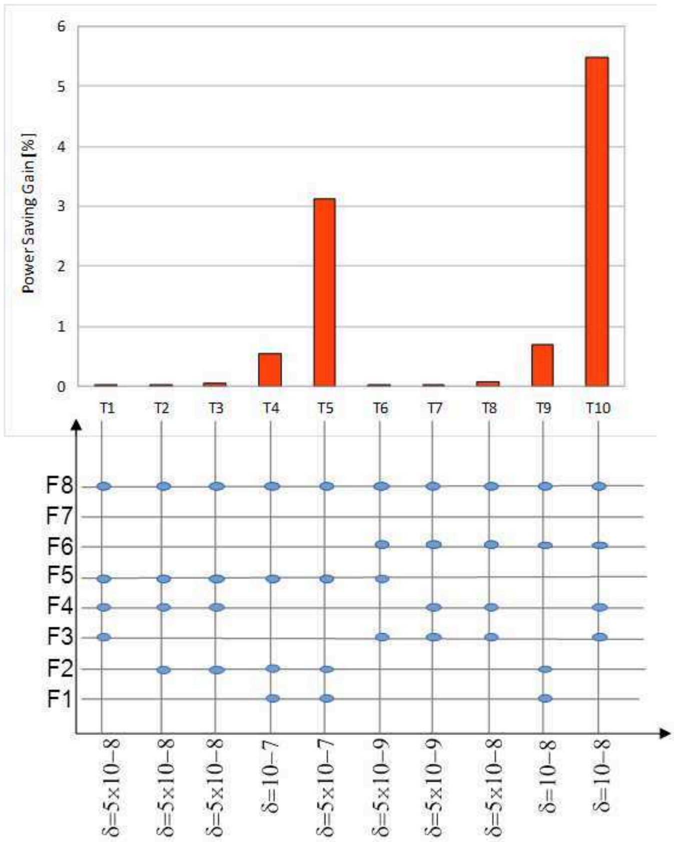
much more evident result for cases T_2, T_3 and T_4: in fact, the higher the autocorrelation of the traffic, the lower the frequencies we can use and therefore the higher the power saving the system can achieve. Also in the same figure, we can notice, for the T_5 case, that the system

Figure 5.18: *Power saving and selected configuration (δ, Φ) corresponding to a target loss probability of* 10^{-6}*, with 2 frequency subsets.*



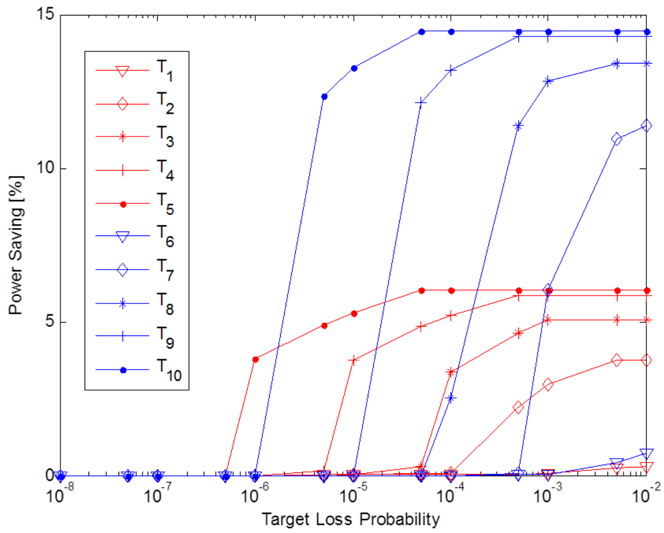
is free to change the clock frequency following the input traffic (δ is equal to 1). In Figure 5.19 the optimization algorithm selects four frequencies for each case: for both the cases T_2 and T_3 the δ parameter is equal to $5 \cdot 10^{-8}$, whereas for T_4 and T_5 lower frequencies are selected

Figure 5.19: *Power saving and selected configuration (δ, Φ) corresponding to a target loss probability of* 10^{-6}*, with 4 frequency subsets.*



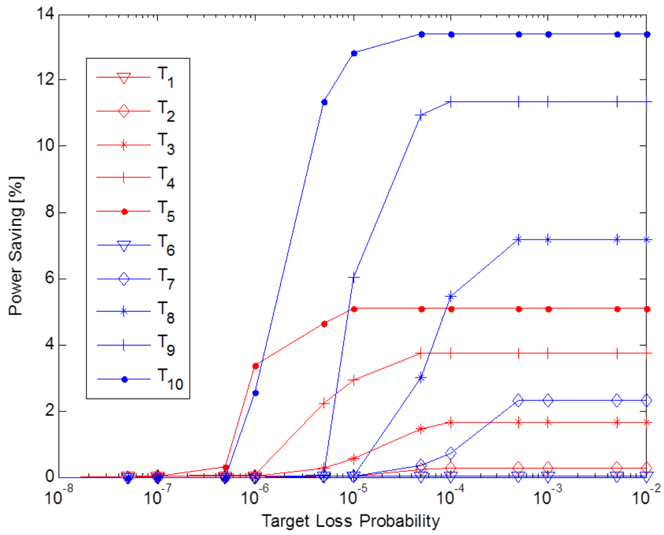
and the δ parameter leaves the Governor free to change the clock frequency more often, increasing the power saving and maintaining the same loss probability. Finally, in order to evaluate the impact of δ on the performance, the solution for the optimization problem for all

Figure 5.20: *Power saving vs. target loss probability resulted by optimization problem with* $\delta = 10^{-4}$.



the 127 sets described so far has been calculated, but for two given values of δ, i.e. 10^{-4} and 10^{-6}. The respective results are shown in Figure 5.20 and Figure 5.21. First of all, it is easy to notice that the higher the value of δ, the higher the power saving, since the Router Governor can follow the input traffic more accurately: in fact, we can achieve a higher power saving using δ equal to 10^{-4} rather than 10^{-6}. Finally, let us note that all the above figures have been presented to evaluate the impact of the traffic behavior, the parameter δ and the set Φ on the power consumption and the system performance, but the same figures can also be used by the system designer to choose suitable values of those parameters according to the input traffic, looking for the best trade-off between power saving and loss probability.

Figure 5.21: *Power saving vs. target loss probability resulted by optimization problem with* $\delta = 10^{-6}$.



CHAPTER

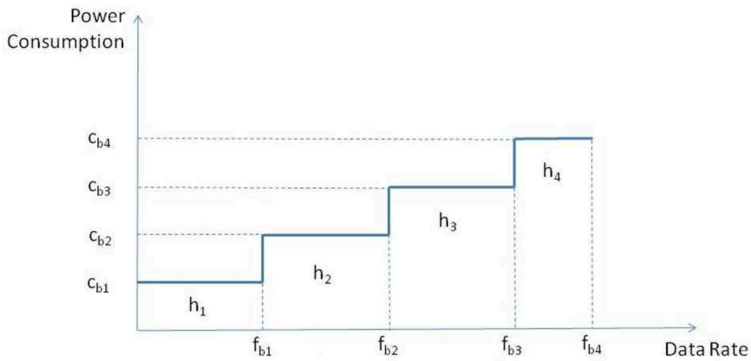
SIX

NETWORK CONTROL POLICY THROUGH SDN

In this section the design of a Network Control Policy is discussed. An algorithm to calculate the optimal traffic path allocation is proposed. This algorithm is based on the knowledge of the internal details of the OpenFlow switches and the power states of the internal entities. To each state is associated not only a given level of power consumption, but also a given level of performance. This information is collected locally by the switch and transmitted to the controller using the Green Abstraction Layer primitives.

An OF switch can be represented, from a network point of view, by a set of ports (Physical, Logical and Reserved), a set of actions that could be performed on traffic flows (the list of actions of a switch depends on the specific hardware) and one or more flow tables, which contains instructions about actions and forwarding for specific traffic

Figure 6.1: *Generic curve that describe the power consumption of a GAL Logical Entity*



flows. An OpenFlow version of the GAL can consider as logical entities the ports, the actions and the tables of all the OF switches. Since there is no constraint about how those resources have to be implemented inside the device, manufacturers can use their own technologies and solutions and thank to the GAL each resource can be defined in terms of power consumption with respect to the offered load. In this way, all the entities of the network can be completely described, from the energy perspective, by a set of EAS. Figure 6.1 provide an example of a generic module that has four different EAS, where cvh is the power consumption related to a generic state h and f_{vh} is the correspondent maximum supported data rate.

6.1 OpenFlow Protocol Extension

As mentioned before, the extension of OpenFlow protocol to make the network efficient using the GAL requires both the definition of new OpenFlow messages and the design of a new module inside the latter. As far as the new message definition is concerned, the switches have to send the description of its own logical resources and related EAS to the controller. In this direction, it has been defined a set of new messages to implement the main GAL primitives: discovery, provisioning and monitoring. On the other hand, a new NCP called Energy Optimizer is proposed: it is a possible demonstration of how the controller can exploit the information provided by the GAL to reduce the power consumption of the network. In fact, the Controller gathers information about the EASs and the Energy Optimizer module calculates a nearly optimal configuration to process a given traffic demand consuming the minimum energy.

The discovery primitive allows the Controller to get information about the power consumption of each entity. The related packet from the switch to the Controller contains the complete description of EAS: an identifier of the EAS, the power gain with respect the maximum power consumption and the throughput in terms of packets per second and bits per second. In order to completely support the GAL the message contains also wake-up time, sleep time and transition power. The provisioning primitive allows the Controller to set the state of an OF switch. The related packet from the Controller to the switch contains the logical resource identifier and the EAS identifier. The monitor state and monitor consumption primitives allows the Controller to get information respectively about the current EAS set on a logical entity and the current power consumption of the entity. The related packets from the switch to the Controller contains the identifier of the entity and, in the first case the identifier of the current state, in the second case the power consumption in watt of the entity.

6.2 The Energy Optimizer Controller module

In this section a possible algorithm to calculate the solution of the "in-network consolidation" problem is provided. Consider a network represented by a directed graph G(V, E) where V is a set of nodes (OpenFlow switches) and E is a set of links among them. Each node $v \in V$ has P_v ports, implements A_v actions and uses T_v tables. Without loss of generality, we can consider that each switch has a single table t_v that contains all the entries: in fact the power consumption of such a table is proportional to the number of flows the node have to forward or process. Each link $e(v, w) \in E$ between v and w has a maximum allowed half duplex capacity $M_{ij} \in [0, 1]$. Moreover, $I(v, a)$ is defined, which is equal to 1 if node v implements action a, 0 otherwise. The traffic demand is represented by a set of S services, each of them defined by:

- $src_s \in V$ is the source node,
- $dst_s \in V$ is the destination node,
- f_s is the maximum data rate required by the service.

- act_s is the set of actions required by the service. Each service requires W_s actions.

Here a formal definition of a problem is proposed, which takes as input the topology, the EASs of each logical resource and the traffic demand to be satisfied and solve the problem, allocating the traffic over the network and the actions on specific nodes. The objective is therefore to find the flows and actions configuration that satisfies the traffic demand and minimizes the power consumption of the entire network. The problem discussed so far can be formalized using the following Integer Linear Programming (ILP) formulation:

$$min(\sum_{b=1}^{B} x_{bh} \cdot c_{bh}) \tag{6.1}$$

subject to

$$x_{bh} \geq 0 \tag{6.2}$$

$$x_{bh} \leq 0 \tag{6.3}$$

$$y_{ij}^{s} \geq 0 \tag{6.4}$$

$$y_{ij}^{s} \leq 1 \tag{6.5}$$

$$\sum_{s=1}^{S} y_{ij}^{s} \cdot f_s \leq M_{ij} \forall I \in B, \forall j \in B \tag{6.6}$$

$$\sum_{j=1}^{B} y_{ij}^{s} = \sum_{j=1}^{B} y_{ji}^{s}; \forall I \in B, \forall s \in S \tag{6.7}$$

$$\sum_{j=1}^{B} y^{s}_{s_s j} = 1 \forall s \in [1, S] \tag{6.8}$$

$$\sum_{i=1}^{B} y^{s}_{i d_s} = 1 \tag{6.9}$$

$$\sum_{i=1}^{B} \sum_{j=1}^{B} y^{s}_{ij} \cdot a^{t}_{s} \cdot I^{t}_{i} = 1 \tag{6.10}$$

$$\sum_{i=1}^{B} \sum_{s=1}^{S} y^{s}_{ij} \cdot f_s \leq \sum_{b=1}^{B} \sum_{h=1}^{H} x_{bh} \cdot f_{bh} \forall j \in [1, B] \tag{6.11}$$

$$\sum_{h=1}^{H} x_{bh} = 1; \forall b \in [1, B] \tag{6.12}$$

where the variables of the problem denote:

- x_{bh}, a binary variable which is equal to 1 if the sub-module b is in state h;
- y^{S}_{ij} , a binary variable which is equal to 1 if the flow correspondent to service s pass through the link from i to j.

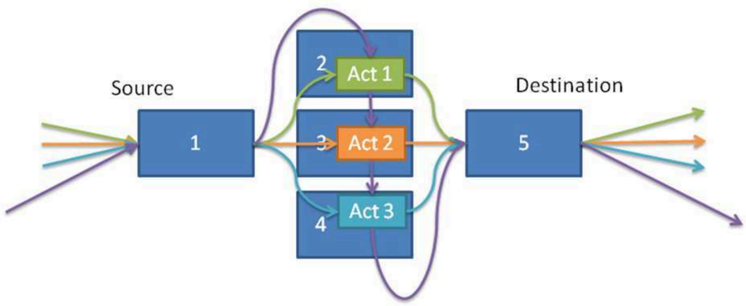
Expression 6.1 represents the overall power consumption of every logical entity of the network: it is minimized by the problem. Constraints 6.2, 6.3, 6.4 and 6.5 are lower and upper bound of the variables. Constraint 6.6 ensures that no link carries more traffic flow than its capacity. A flow-conservation constraint 6.7 ensures that no flow is lost or created except at the source and destination. Constraints 6.8 and 6.9 ensure that the sum of the flows leaving the source, or entering the destination of a service sums to 1. Constraint 6.10 guarantees that traffic flows pass through nodes which implement actions required by each service. Given that x_{bh} variables depends on flows which pass through the nodes of the network, it is necessary to add also constraint 6.11, which include a dependency among x_{bh} and y^s_{ij}. Finally, constraint 6.12 guarantees that each logical entity works just in a single EAS. It is required to use an Integer formulation to guarantee that variables x_{bh} assume values 0 or 1. At the same time, the usage of an Integer formulation assure that the packets belonging to each traffic flow are sent along the same path, since y^s_{ij} takes binary values. This formulation is NP-hard and to make the problem tractable in the next section a relaxation is introduced based on a Continuous Linear Programming (CLP) definition of the problem and a heuristic that can easily calculate a solution which achieve configurations comparable to the original one. Using a CLP relaxation, the packets belonging to a flow might not follow the same path, but could be split over different paths. The reader notice that the ILP definition described above takes as input a network of logical resources B and it does not take into account the nodes of the network. To obtain the entities topology it has been defined a transformation algorithm which converts the original nodes topology to the entities topology. This algorithm is deeply explained in appendix.

6.3 Problem relaxation

In this section a Continuous Linear Programming (CLP) solution of the previous problem is proposed as a relaxation of the model described in Section 6.2 coupled with a heuristic to find an integer solution. Starting from the problem defined above, constraint 6.12 is removed and the meaning of the variables y^s_{ij} is changed. They are continuous and are defined as follows: y^s_{ij} is the part (percentage) of f_s allocated on the link from i to j. Since x_{bh} variables assume integer values, the usage of a heuristic is proposed, which approximate such variables to 0 or 1. The algorithm behind that is composed by three phases: in the first one a set of constraint are added to the problem, in order to set all the logical entities to the EAS with lower supported data rate and lower power consumption; in the second phase the algorithm calculates the solution of the CLP problem, (the one defined above); so, at the third step, the algorithm checks the results, and selects the entity which presents the higher value of $\sum_{h=1}^{H} x_{bh} - 1$; if setting the current EAS the entity cannot process the assigned traffic flow, the next state is activated and the lower states are constrained to 0. Let OP denote the formulation of the optimization problem in Section 6.12 and let H˙max the maximum number of EAS which of an entity and let $i = 1$; the algorithm executes the following steps:

- Step 1 - Add temporary constraints $x_{b1} = 1$ for all $b \in B$ to OP.
- Step 2 - Solve the optimization problem OP. Calculate the optimal values of the variables y^s_{ik} and x_{bh}
- Step 3 - If all x_{bh} variables take integer values, exit. Otherwise, check the results of Step2:
  - Take the entity which has the maximum value of $\sum_{h=1}^{h} x_{bh} - 1$

Figure 6.2: *Testing topology*



- Add a new constraint $x_{b_i+1} = 1$ to OP;
- Add a new constrating $x_{b_j} = 0$ for each j=1, ..., i to OP;
- execute Step 2 again.

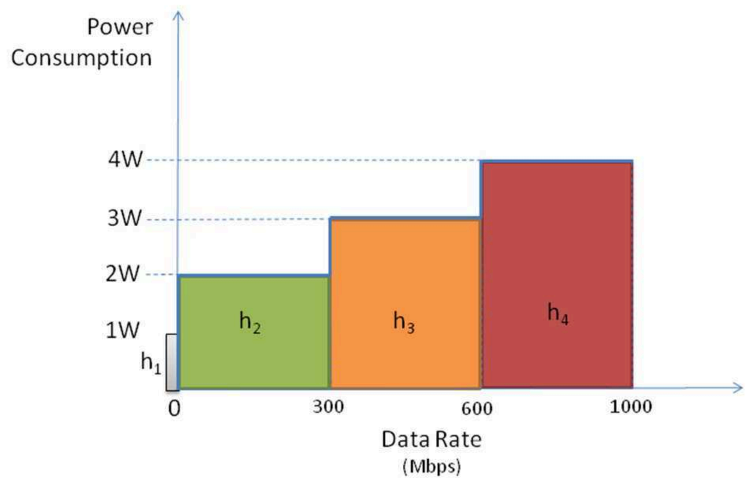
6.4 Performance Analysis

In this section the obtained results are shown and discussed for a simple topology, using the proposed approach and varying the traffic demand. The topology is the one shown in Figure 6.2, which is composed by five nodes (and correspondent forty logical entities): Node 1 is the source of services, Nodes 2, 3 and 4 implements respectively Action 1, 2 and 3, whereas Node 5 is the destination.

For the sake of simplicity, each logical entity (ports, tables and actions) have the same power consumption profile: they have a standby state h1 which consumes only 1 Watt and other three active state h2, h3, h4 which consume respectively 2, 3 and 4 Watt.

Our tests aim to analyse the behaviour of the Energy Optimizer Controller module varying the number of services and the bandwidth

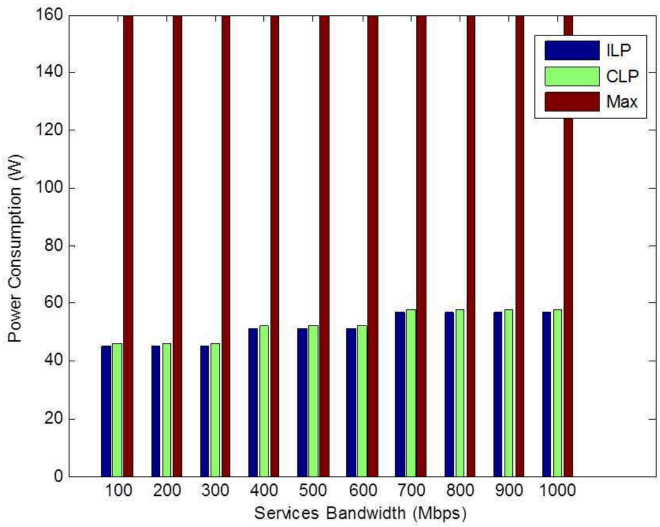
Figure 6.3: *Power consumption characterization of each logical entity in the network*



that the latter request to the network. So four different scenarios on the same topology are defined:

- Only a service is activated and it requires to be processed by Action 1;
- Two services are activated, Service 1 requires to be processed by Action 1 whereas Service 2 requires to be processed by Action 2;
- Three services are activated, Service 1, 2 and 3 require to be processed respectively by Action 1, 2 and 3.
- Four services are activated, Service 1, 2 and 3 require to be

Figure 6.4: *Power consumption vs. Service Bandwidth for Case 1.*

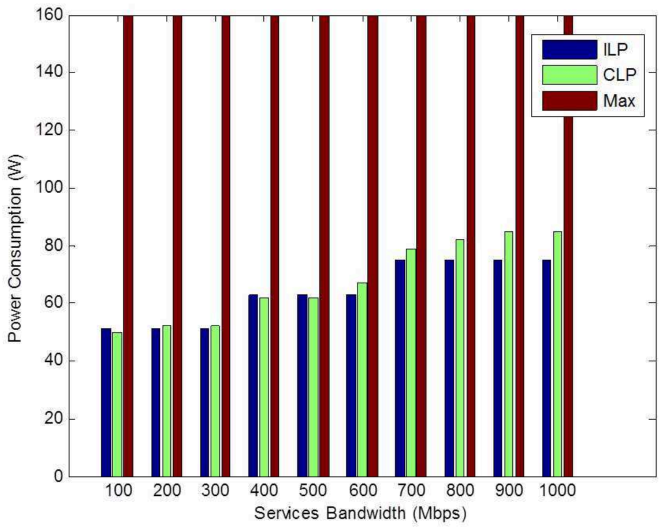


processed respectively by Action 1, 2 and 3, whereas Service 4 requires all Action 1, 2 and 3.

In Figures 6.4, 6.5, 6.6 and 6.7 the power consumption of all four different scenarios is shown varying the bandwidth. The reader notice that all the services require the same bandwidth and the x axis reports the bandwidth required by a single service.

The blue bars represent the power consumption of the solutions calculated by the Integer Linear Programming, the green bars represent the power consumption of the solutions calculated by the Continuous Linear Programming and the brown bars represent the maximum power consumption of the network (when no energy efficiency techniques are used). In Figures 6.4 and 6.5 (Scenarios 1 and 2) it is clear

Figure 6.5: *Power consumption vs. Service Bandwidth for Case 2*



that the power consumption of solutions calculated by ILP and CLP is comparable, also varying the bit rate. In Figure 6.6 (Scenario 3) the ILP obtain better results, in particular with high service bandwidth. The reason behind that is that the system is too constrained and the heuristic, in such condition, will not lead to an optimal solution. In Figure 6.7 the situation is the opposite: removing the constraint to manage the entire packet belonging to the same traffic flow on a single path the CLP is more flexible and it is able to find more efficient solutions. The reader notices that, in Scenario 4, if the bandwidth requested by each service is higher than 600 Mbps the problem is infeasible.

Figure 6.6: *Power consumption vs. Service Bandwidth for Case 3*

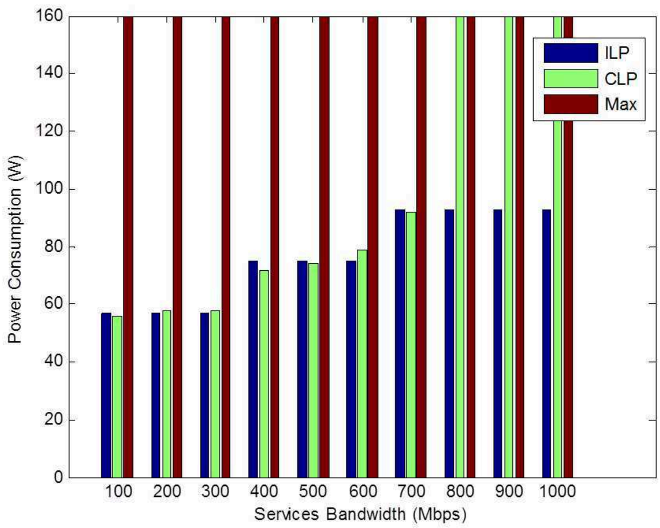
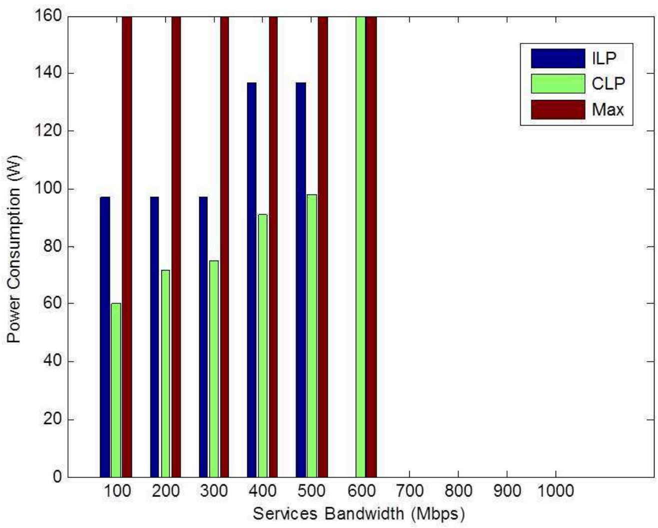


Figure 6.7: *Power consumption vs. Service Bandwith for Case 4*



CHAPTER

SEVEN

VIRTUAL NETWORK FUNCTIONS AT THE EDGES

7.1 Introduction

Emerging paradigms such as Software Defined Network (SDN) [43] and Network Function Virtualization (NFV) [42] could bring a deep impact on the evolution of Telco Operators networks and the Technology Providers ecosystems. As already discussed in previous sections, they can have a huge impact on the networks reducing costs and increasing the flexibility and the agility of the network it-self. Whereas, SDN aims at decoupling the Control Plane from the hardware Data Plane (which means the actual packet forwarding) moving its logic (and states) to centralized controllers, NFV implies virtualizing Network Functions (NFs) that can run on standard hardware and that can be instantiated and migrated in various locations of the network.

This section proposes an architecture that supports virtual function allocation as closer as possible to the users, in order to minimize service provision latency. This architecture and its architectural components have been called NetFATE (Network Function At The Edges). In order to show the real applicability of the proposed architecture, a Proof-of-Concept has been implemented and it is discussed later on in the section. An analytical Markov model of the NetFATE network is also proposed to estimate end-to-end delay for a given topology and traffic distribution, in order to support the Orchestrator work in allocating and migrating VMs on network nodes. The tools introduced by this work can strongly help network managers to improve the QoE, the flexibility and the runtime customization of the network services.

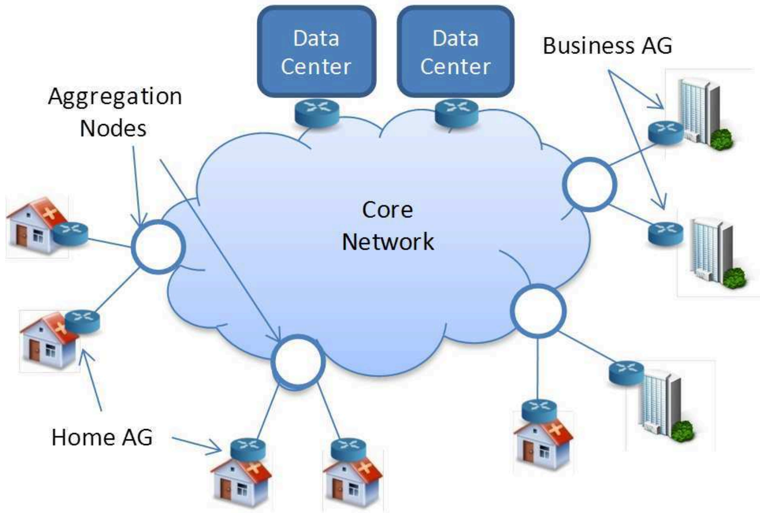
7.2 Reference scenario

It worth spending a section on the definition of the scenario to which the discussion is pointing to. The reference scenario is depicted in Figure 7.1, where a simple schematization of a common telco operator network topology is shown: the main elements of the considered architecture are: the Data Centers (DC), with high computational and storage capabilities, the Core Network, characterized by high speed WAN connections and high performance forwarding devices, the Aggregation Nodes (ANs), each aggregating traffic of thousands of users, and the Access Gateways (AGs), that can be either Home or Business Gateways, according to the kind of connected customers.

Here it is assumed that in this network scenario some nodes are able to support virtual Network Functions (NFs), thanks also to a suitable hardware and software architecture where functions are implemented as separated virtual machines (VMs) on network nodes; in the following these nodes will be referred to as NFV nodes. NFV nodes provide more processing power than normal routers or network devices and allow VM migration. All other nodes that do not implement NFV features will be referred to as non-NFV nodes. Since one of the goals of this work is to analyze the effects of SDN and NFV distributed at the edge of the network on the performance, a mathematical tool has been designed to estimate one of the major factors that affect the QoE, i.e. network latency.

The end-to-end network latency is caused by different factors, the most important of which are the propagation delay and the link bandwidths. Another important aspect to consider is the possible loss of packets that leads to QoS reduction (i.e. UDP-based services) or retransmissions (TCP-based services). Losses are mainly caused by the node overload. With the aim of catching all these elements, an analytical model will represent the nodes as a set of queues, as shown later in Figures 7.5 and 7.6, where the architecture of both NFV and non-NFV nodes is sketched: a non-NFV node can be represented by the set of queues associated to its output Network Interface Cards (NICs); packets are queued in those queues to be transmitted on output links and so they are served with a rate equal to the packet transmission rate of the output link. Instead, each NFV node, besides the previous queues, contains a set of queues to manage function processing: one queue for each function run by the NFV node. The queue associated to a generic function contains packets waiting to be processed by the CPU of the NFV node to receive the specific service of that function, and so it is served at the frequency of the CPU quota assigned to

Figure 7.1: *Proposed deployment architecture of SDN/NFV solution*



the relative VM. Upon the above observations, a DC is represented as a very powerful NFV node because the provided processing power is very high; on the contrary, nodes at the edge of the network have a lower processing power.

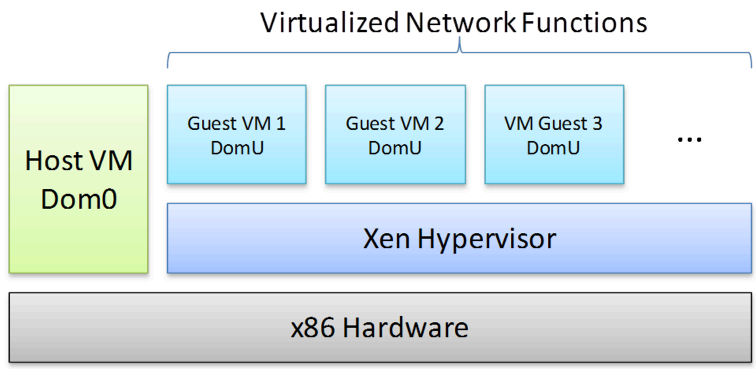
7.3 Proposed architecture and implementation

This section describes the proposed NetFATE architecture. It is a distributed architecture composed by NFV nodes and an Orchestrator: NFV nodes are capable to support both NFV and SDN functionalities locally, whereas the Orchestrator is able to manage Network Functions and service chaining according to the position of the Network Functions (NFs) and the requirements of both telco operator and users.

7.3.1 NFV nodes

An NFV node, from a hardware perspective, is a device equipped with a set of Network Interface Cards (NICs) that support virtualization and can provide efficient implementation of the internal networking operations. Regarding the software architecture, it has to support both the virtualization of Network Functions and the communication among the VMs: the former has been put in place using a virtualization hypervisor, whereas the latter has been implemented thanks to virtual switches that support OpenFlow protocol. Those aspects are discussed in the following. The local support of NFV is realized using the well-known Virtual Machine (VM) Hypervisor Xen [32], as shown in Figure 7.2, that allows the execution of different isolated VMs inside the same hardware host. Xen is a "bare-metal" hypervisor, which means that it runs directly on top of the physical machine rather than within an operating system. It needs a host machine, called Dom0, to be managed (see Figure 7.2). In addition, a set of guest VMs are executed on the same host to implement the Network Functions. In order to provide an NFV node with SDN, OpenvSwitch has been installed on Dom0: OpenvSwitch (OvS) [62] is a multilayer open-source virtual switch that allows to automate the network configuration programmatically. It also supports the OpenFlow protocol and exploits this feature to implement SDN capabilities on the NFV node. In Figure 7.3 the network communication architecture among virtual machines is shown. Using OpenvSwitch it is possible to setup one (or more) vir-

Figure 7.2: *NFV node virtualization architecture*

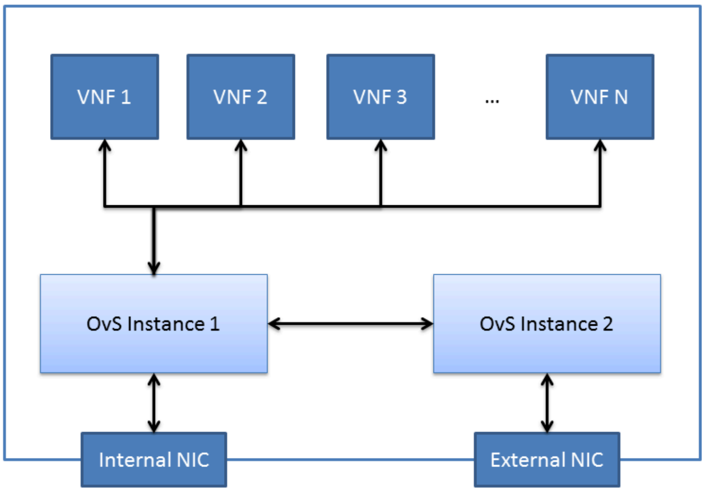


tual switches to implement communications among VMs. This switch receives forwarding instructions from the SDN controller residing in the Orchestrator. Dom0 covers a local coordination and is also in charge of managing the Network Functions that are running on the NFV node.

7.3.2 Orchestrator

Orchestration is the automatic coordination and management of the services over the network. If on the one hand the introduction of SDN and NFV introduces high flexibility from the orchestration perspective, on the other hand it leads to a high complexity, due to the need of allocating, migrating and destroying virtual network functions, and

Figure 7.3: *NFV node VMs communication architecture*

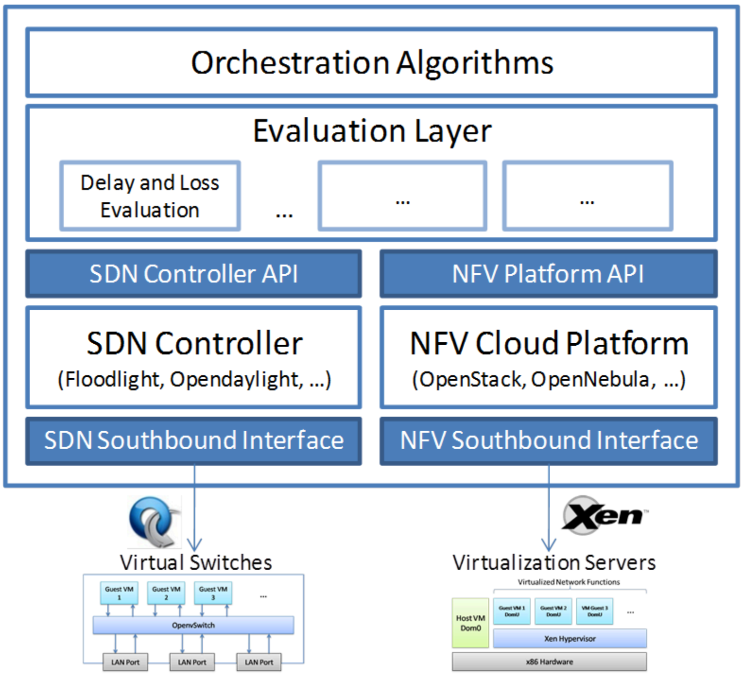


consequently controlling the traffic paths according to the run-time evolution of the network. Therefore, the Orchestrator is in charge of managing both Network Function allocation and traffic routing among Virtual Machines for all the NFV nodes of the NetFATE network, dynamically implementing the service chains according to the operator needs and to the user requirements. As mentioned before, to realize all the above functionalities, it is required to use both SDN and NFV functionalities. So the proposed architecture for the Orchestrator is the one shown in Figure 7.4. Thanks to the SDN Controller, the Orchestrator is able to implement both routing control and communication with the NFV nodes (using the OpenFlow protocol). Instead, with NFV Cloud Platform, the Orchestrator can manage the VM life cycle. The Orchestrator runs on a dedicated server and communicates with each NFV node through the telco operator network. The SDN Southbound Interface is provided by the OpenFlow protocol [44], whereas the NFV Southbound Interface is provided by the Xen API (Application Program Interface) [63]. Both SDN Controller and NFV Cloud Platform implements APIs in order to be managed by upper layer and to provide information to it. The Orchestration Layer contains a set of boxes that help to orchestrate the network resources: the Delay and Loss Evaluation block provides information regarding the network delay and the possibility that a loss occurs. This module contains some formulas from the model described in the next section. The Orchestration module makes decisions regarding the actual allocation of both virtual resources and traffic paths.

7.4 A model for the orchestrator

In this section it is defined a mathematical tool that can be very useful for orchestration algorithms and strategies that want to minimize the delay and increase QoE for the users. First, each single node of the network is modeled, either NFV node or non-NFV node, as a set of queues; then, some parameters will be defined in order to evaluate the performance of the network, varying the configuration setup. The queues correspondent to the output links will be referred as output (OUT) queues, and to the ones associated to the functions as network function (NF) queues.

Figure 7.4: *Orchestration architecture*



Let N be the number of nodes in the network, that can be subdivided in N_{NFV} NFV nodes, and N_{nNFV} non-NFV nodes. The number of functions implemented by the i-th node is defined as $L_i^{(F)}$, and the number of output NICs in the same node as $L_i^{(OUT)}$. Therefore, as shown in Figure 7.5, the NFV node i has a set of $L_i^{(F)}$ NF queues and a set of $L_i^{(OUT)}$ OUT queues. Let $Q_{i,j}^{(F)}$ and $Q_{i,h}^{(OUT)}$ be the generic NF queue and OUT queue of the node i, with $i \in [1, N_{NFV}]$, $j \in [1, L_i^{(F)}]$, and $h \in [1, L_i^{(OUT)}]$. The generic NF queue $Q_{i,j}^{(F)}$ is loaded with an input rate $\Lambda_{i,j}^{(F)}$ and served with a service rate $\mu_{i,j}^{(F)}$. Instead, the generic OUT queue $Q_{i,h}^{(OUT)}$ is loaded with an input rate $\Lambda_{i,h}^{(OUT)}$ and served

Figure 7.5: *NFV node model*

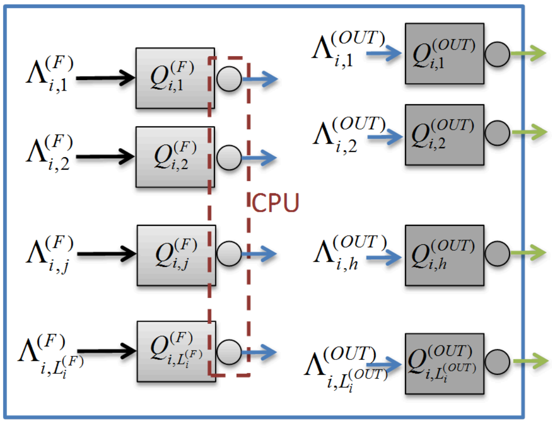
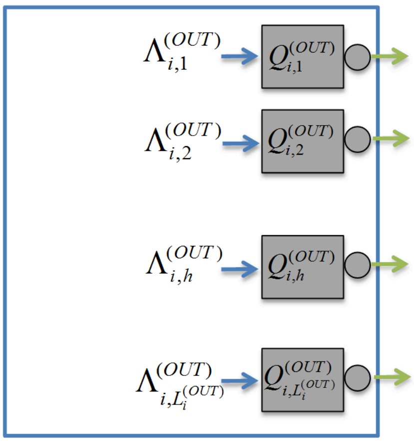


Figure 7.6: *Non-NFV node model*



with a service rate $\mu_{i,h}^{(OUT)}$. Of course, we have:

$$\Lambda_{i,j}^{(F)} = \sum_{\forall k \in \Phi_{i,j}} \lambda_k \tag{7.1}$$

$$\Lambda_{i,h}^{(OUT)} = \sum_{\forall k \in \Psi_{i,h}} \lambda_k \tag{7.2}$$

$$\mu_{i,h}^{(OUT)} = C_{i,h}^{(NIC)} \tag{7.3}$$

where: λ_k is the mean bit rate of the generic flow k; $\Phi_{i,j}$ is the set of flows routed through the node i and requiring the function j; $\Psi_{i,h}$ is the set of flows crossing the node i and leaving it through the NIC h; $C_{i,h}^{(NIC)}$ is the output packet transmission rate of the h-th output link of the i-th NFV node. In order to calculate the service rate of $Q_{i,j}^{(F)}$, the method of Lagrange multipliers can be applied [64], assuming that the processing capacity of the node is shared among those queues using a Generalized Processor Sharing and a Weighted Round Robin scheduling. In this way, when a given function queue of a node is empty, its processing capacity is shared among all the other function queues of the same node. We have:

$$\mu_{i,j}^{(F)} = \Lambda_{i,j}^{(F)} + \frac{\sqrt{\Lambda_{i,j}^{(F)}}}{\sum_{f=1}^{L_i^{(F)}} \sqrt{\Lambda_{i,f}^{(F)}}} \cdot (C_i^{(CPU)} - \sum_{f=1}^{L_i^{(F)}} \sqrt{\Lambda_{i,f}^{(F)}}) \tag{7.4}$$

where $C_i^{(CPU)}$ is the mean packet processing rate of the processor in the i-th NFV node. As expected, the service rate of $Q_{i,j}^{(F)}$ in Equation 7.4 results as the sum of the input rate of the same queue and an addition computation capacity that is the CPU quota not used by the other queues. The same notation can be used for the non-NFV nodes, with the only difference that only OUT queues are present in them. Now it is possible to model the whole network, constituted by both NFV and non-NFV nodes. The model inputs are the network topology, the allocation of the functions on the nodes and the traffic characterization. More specifically, each traffic flow represents an aggregate of traffic streams with the same ingress node, the same egress node, not necessarily coinciding with the ingress node, and requiring the same network functions. So the traffic characterization consists in the description of all the flows, in terms of ingress and egress nodes, list of required functions, and mean bit rate. Starting from this information, a routing algorithm calculates the end-to-end path for each flow. We model the whole network with an N-dimensional continuous-time Markov chain whose state is defined as follows:

$$S^{(\Sigma)}(t) = (\underline{S}_1(t), K, \underline{S}_N(t)) \tag{7.5}$$

where $\underline{S}_i(t)$ represents the state of the generic node i, that is:

$$\underline{S}_i = (S_{i,j}^{(F)}, K, S_{i,L_i^{(F)}}(t)) \tag{7.6}$$

if the node i is an NFV node, or

$$\underline{S}(t) = (S_{i,1}^{(OUT)}(t), K, S_{i,L_i^{(OUT)}}^{(OUT)}(t)) \tag{7.7}$$

if the node i is a non-NFV node. The state of the generic queue represents the number of packets in the queueing system, i.e. in both the queue and in the service facility. Assuming that all the flows are characterized by an exponential distribution of the inter-arrival times, and that the service times are exponentially distributed as well in both

NF and OUT queues, and assuming that the routing algorithm is able to avoid closed loops, we match the hypotheses of the Jackson theorem for queuing networks [65], and therefore the equilibrium probability distribution is particularly simple to compute as the network has a product-form solution. To this purpose, let us indicate for node i the utilization coefficient of the j-th NF queue as:

$$\rho_{i,j}^{(F)} = \frac{\Lambda_{i,j}^{(F)}}{\mu_{i,j}^{(F)}} \tag{7.8}$$

and the utilization coefficient of the h-th OUT queue as:

$$\rho_{i,j}^{(OUT)} = \frac{\Lambda_{i,j}^{(OUT)}}{\mu_{i,j}^{(OUT)}} \tag{7.9}$$

Now, according to the classical queueing theory, it is possible to calculate for the queueing systems $Q_{i,j}^{(F)}$ and $Q_{i,j}^{(OUT)}$:

- the mean number of packets in the queuing systems:

$$\nu_{i,j}^{(F)} = \frac{\rho_{i,j}^{(F)}}{1 - \rho_{i,j}^{(F)}} \tag{7.10}$$

$$\nu_{i,h}^{(OUT)} = \frac{\rho_{i,h}^{(OUT)}}{1 - \rho_{i,h}^{(OUT)}} \tag{7.11}$$

- the mean sojourn time in the queueing systems:

$$W_{i,j}^{(F)} = \frac{\nu_{i,j}^{(F)}}{\Lambda_{i,j}^{(F)}} \tag{7.12}$$

$$W_{i,h}^{(OUT)} = \frac{\nu_{i,h}^{(OUT)}}{\Lambda_{i,h}^{(OUT)}} \tag{7.13}$$

Finally, the end-to-end delay for each flow is derived. It depends on the paths imposed by the routing algorithm:

$$\begin{aligned} W_k^{e2e} = & \sum_{i=1}^{N} [\sum_{j=1}^{L_i^{(F)}} W_{i,j}^{(F)} \cdot I_{i,j}^{(F)}(k) + \\ & + \sum_{h=1}^{L_i^{(OUT)}} W_{i,h}^{(OUT)} \cdot I_{i,h}^{(OUT)}(k) + \\ & + \sum_{h=1}^{L_i^{(OUT)}} W_{i,h}^{(PROP)} \cdot I_{i,h}^{(OUT)}(k)] \end{aligned} \tag{7.14}$$

where: the term $I_{i,j}^{(F)}(k)$ is a Boolean indicator function that is equal to 1 if and only if Flow k uses the function j in Node i, otherwise is null; the term $I_{i,h}^{(OUT)}(k)$ is a Boolean indicator function that is equal to 1 if and only if Flow k leaves Node i through NIC h. the term $W_{i,h}^{(PROP)}$ is the propagation delay due to the link that corresponds to Interfaces h of Node i.

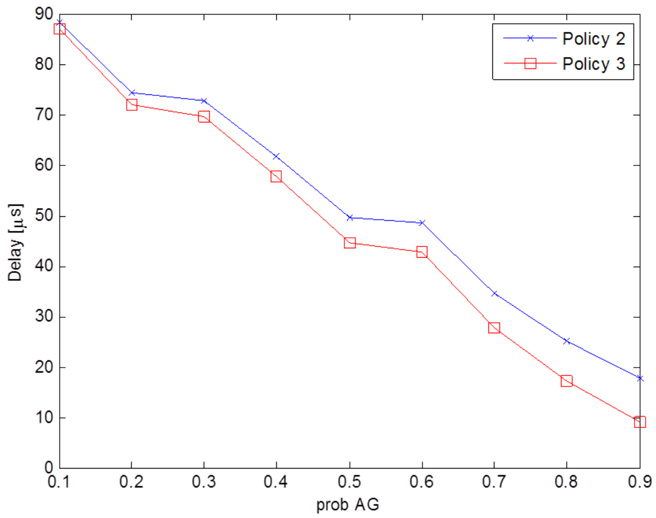
7.5 Case study

In this section a case study is discussed in order to estimate the latency that can be obtained by the function allocation strategy. In this direction, a specific topology, starting from the one sketched in

Figure 7.1 has been considered with a fixed set of parameters (like link bandwidths, distance between nodes, traffic requirements, and so forth). The results shown in this section have been calculated through the analytical model defined in Section 7.4. The topology is constituted by one Data Center (DC), ten Aggregation Nodes (ANs) and two hundreds Access Gateways (AGs): the AGs are randomly connected to ANs, whereas the DC and the ANs are connected to each other through the core network. DC, ANs and AGs can run VMs that contain network functions and then they are considered as NFV nodes, supposing that ANs and AGs are equipped as "micro Data Centers", with much smaller capacity than the Data Center. In particular, we consider the processing capacity of the DC high enough to support the processing capacity of all the functions needed by the whole network, whereas the processing capacity devoted to run the VMs of the ANs is ten time higher than the AGs one. The bandwidths of the links between AG and AN is 100 Mbps and the distance is 1 km, whereas the bandwidth of the links between ANs is 10 Gbps and the distance is 5 km. The propagation delay depends on the distance: it is calculated as the time needed to traverse a cable of a given length at the speed of light inside a fiber-optic channel [66]. The distance from the AN and DC is set to 2000 km which corresponds to a one way propagation delay of 10 milliseconds. Thousand flows enter the network from AGs (at user premises) and are characterized by a source node, a destination node (both are AGs), a bit rate and a list of functions that should process the flow itself in order to build the desired service chain. We assume that the network provides six kinds of functions and both the source node of each flow and the functions requested by it are randomly selected in the case study; the bit rate is set to the same value for all the flows, in the range from 100 Kbps to 2 Mbps, whereas the destination node is set with a specific criteria described in the following. First we need to specify that the destination can be the same of the source: it could happen, for example, in some scenarios of Internet of Things and Big Data services, where the source and the destination of a stream of data reside in the same place and are connected to Internet through the same AG. If the source and the destination of a flow are connected to the same AG, the flow path needs to traverse the nodes that run the required functions and come back to reach the destination. Since performance strongly depends on the mutual position of source and destination, we have defined a parameter called prob_AG, which is the probability that the sources and the destinations of flows are attached to the same AG, and we have ran the model varying this parameter from 0.1 to 0.9. Therefore, once the topology and the characteristics of the flows have been fixed, as a first step we have randomly chosen all the destinations with prob_AG equal to zero, i.e. drawing, for each flow, an AG different from the AG of the source. Then we have increased prob_AG by steps of 0.1, increasing, step by step, always in a random way, the number of flows that enter and exit the network by the same AG. In order to emphasize how the function allocation can influence the network performance, we define three different policies: with Policy 1 all the functions are allocated on the DC; with Policy 2 all the function are allocated on the AN and with Policy 3 all the functions are allocated on the AG. It is worth noting that each flow is processed by the first function of each kind that finds on its path. Using Policy 1 the delay of the network is constant varying the prob_AG, because all the flows start from the source, reach the Data Center where they are processed, and come back again to the destination. Using instead Policy 2 and Policy 3 an interesting variation can be observed. In the following figures the curve relative to Policy 1 are not shown because the delay is constant and very high with respect to the other two curves. In particular, it is in the magnitude of milliseconds, whereas for Policy 2 and 3 it is in the magnitude of microseconds. A comparison between Policy 2 and Policy 3 is shown in Figure 7.7, that shows the average delay suffered by a flow when the bit rate of each flow is 100 Kbps, and where it is clear that high values of prob_AG correspond to low values of delay. From that figure we note that th propagation delay is the major component of the delay suffered by flows over the whole network. A further reinforcement of this consideration is given by Figure 7.8, where the contribution of the propagation delay is removed and only the delay caused by queueing over links and functions is shown. Policy 3 is slightly better than Policy 2 in terms of delay and the difference between the two lines increase along with prob_AG, for this specific case study. First of all this delay component is very low with respect to the total delay. Whereas all the other curves are constant, the link delay obtained with Policy 3 slightly decrease when the prob_AG increases: this is because more flows are processed directly in AGs and go to the destination, avoiding the queueing delay encountered to reach the ANs.

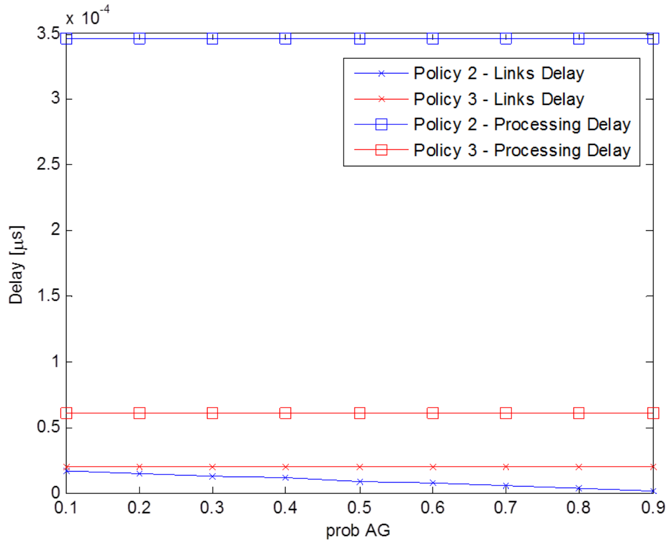
Moreover, the model also takes into account the losses caused by a given function allocation policy. Fgures 7.9 and 7.10 show the average delay suffered by each flow when the bit rate vary from 100 Kbps to 2 Mbps when the prob_AG is respectively 0.1 and 0.9. The peaks in those figures mean that the delay increases up to infinite values: it depends on the fact that the utilization coefficient ρ for one or more

Figure 7.7: *Average delay per flow varying prob_AG, the bit rate is 100 Kbps*



queues is higher than 1, and it causes queue explosion. The values in the flat interval of those curves are dominated by the propagation delay, and correspond to the ones depicted in Figure 7.7 for the related values of prob_AG. It is clear, also from Figure 7.7 that, for both the cases, Policy 3 is slightly better than Policy 2 in terms of delay. But, observing these last two figures, it is possible to see that, if the bit rate of the flows is higher than 996 Kbps, Policy 3 is not able to work, while if the bit rate is higher than 1.8 Mbps, also Policy 2 is not applicable: it means that, in order to minimize the delay of this case study, the application of Policy 3 is indicated only when the bit rate is lower than 996 Kbps, if the bit rate is in the range from 996 to 1.8 Mbps the functions should be migrated on the ANs, and finally, when the

Figure 7.8: *Average queuing delay suffered by each flow varying prob_AG*



bit rate is reaching 1.8 Mbps, the functions should be migrated on the DC.

Figure 7.9: *Average delay suffered by each flow varying the bit rate when prob_AG = 0.1*

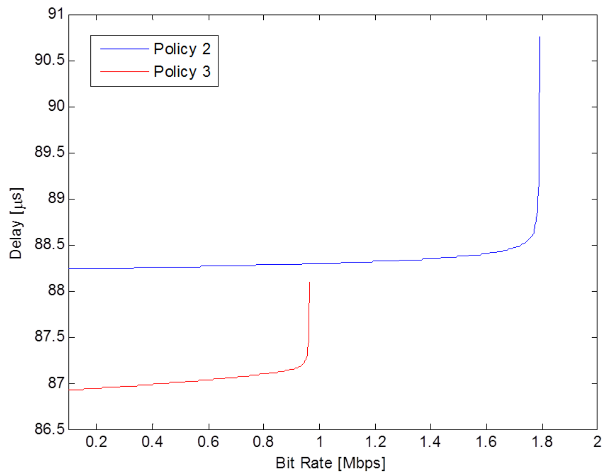
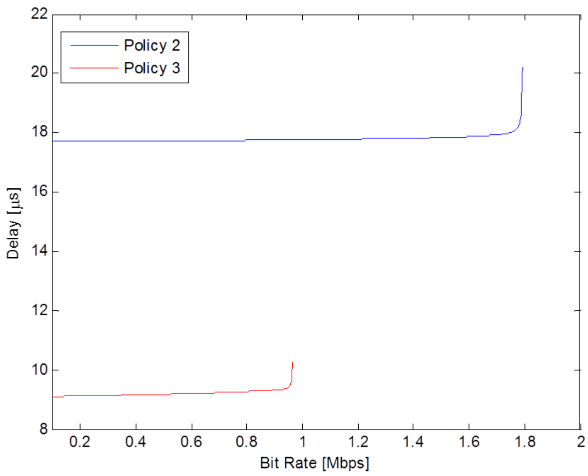


Figure 7.10: *Average delay suffered by each flow varying the bit rate when prob_AG = 0.9*



CONCLUSIONS

In this thesys the concept of sustainable development has been analyzed in the context of the Internet for what concerns the network infrastructure and services.

In Chapter 1 the concept of sustainable development has been analyzed according to the work done by the United Nations in the last decades.

In Chapter 2 this concept has been extended to the telecommunication networks and services thanks to the analysis of the work done by the Internet Society. Four main gaps have been identified for the current network environment.

- High costs
- High environmental impact
- Low flexibility and scalability
- Low reliability and relisilience

According to them, new technological drivers have been individuated in order for them to be integrated into the new Internet.

In Chapter 3 a state of the art related to those technological drivers has been discussed. Specifically, energy efficiency features of network devices, inclusion of Software Defined Networking, virtualization of Network Functions, adoption of Fog Computing over the network and extension of Cloud Computing orchestration mechanisms to the network domain have been taken into account.

In Chapter 4 discussed the proposed architecture for the future Internet in order to support the sustainable development. This high level architecture integrates new kind of devices and embraces the network as a whole.

In the proposed architecture the usage of the Green Abstraction Layer has been envisioned: a physical device provides different power states and they are mapped within the Green Abstraction Layer and exposed to smarter local controllers called Local Control Policies.

In Chapter 5 the design of a Local Control Policy is discussed, as long as the provision of different power states.

At the higher layer Network Control Policies are managed by the Orchestrator and reflected on the network devices through controllers and APIs.

In Chapter 6 an example of Network Control Policy is provided. A Network Control Pocily for SDN devices has been designed and applied from an Orchestrator to the network through the SDN controller.

In Chapter 7 a new scenario is discussed where Fog Computing is envisioned to be massively used over the network in combination with NFV and SDN. A prototype of that technique has been implemented and a mathematical model has been proposed in order to support the Orchestrator to forecast the end-to-end delay of the flows.

The adoption of those techniques and drivers will lead to a more appropriate network environment and will help the United Nations and the world to further move towards a more sustainable development.

BIBLIOGRAPHY

[1] "Our common future - towards sustainable development, http://www.un-documents.net/ocf-02.htm," 1987.

[2] "The millennium development goals report, http://www.un.org/millenniumgoals/2015˙mdg˙report/pdf/mdg 2015 rev (july 1).pdf," 2015.

[3] "Transforming our world: the 2030 agenda for sustainable development, https://sustainabledevelopment.un.org/content/documents/789-1transforming our world.pdf," 2015.

[4] "Millennium development goals, http://www.un.org/millenniumgoals/."

[5] "The internet and sustainable development, http://www.internetsociety.org/sites/default/files/isoc-icts-sdgs-201506-1.pdf," 2015.

[6] "Internet society, http://www.internetsociety.org."

[7] M. Pickavet, W. Vereecken, S. Demeyer, P. Audenaert, B. Vermeulen, C. Develder, D. Colle, B. Dhoedt, and P. Demeester, "Worldwide energy needs for ict: The rise of power-aware networking," in *Advanced Networks and Telecommunication Systems, 2008. ANTS'08. 2nd International Symposium on*, pp. 1–3, IEEE, 2008.

[8] J. Chabarek, J. Sommers, P. Barford, C. Estan, D. Tsiang, and S. Wright, "Power awareness in network design and routing," in *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*, IEEE, 2008.

[9] I. Sprint, "The applied research group.," 2007.

[10] A. P. Jardosh, G. Iannaccone, K. Papagiannaki, and B. Vinnakota, "Towards an energy-star wlan infrastructure," in *Mobile Computing Systems and Applications, 2007. HotMobile 2007. Eighth IEEE Workshop on*, pp. 85–90, IEEE, 2007.

[11] M. Gupta and S. Singh, "Greening of the internet," in *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, pp. 19–26, ACM, 2003.

[12] S. Nedevschi, L. Popa, G. Iannaccone, S. Ratnasamy, and D. Wetherall, "Reducing network energy consumption via sleeping and rate-adaptation.," in *NSDI*, vol. 8, pp. 323–336, 2008.

[13] L. Benini, A. Bogliolo, G. Paleologo, G. De Micheli, *et al.*, "Policy optimization for dynamic power management," *Computer-Aided*

Design of Integrated Circuits and Systems, IEEE Transactions on, vol. 18, no. 6, pp. 813–833, 1999.

[14] L. Benini, P. Glynn, and G. De Micheli, "Event-driven power management," 2001.

[15] H. Jung and M. Pedram, "Dynamic power management under uncertain information," in *Design, Automation & Test in Europe Conference & Exhibition, 2007. DATE'07*, pp. 1–6, IEEE, 2007.

[16] Q. Qiu, Y. Tan, and Q. Wu, "Stochastic modeling and optimization for robust power management in a partially observable system," in *Proceedings of the conference on Design, automation and test in Europe*, pp. 779–784, EDA Consortium, 2007.

[17] R. Bolla, R. Bruschi, F. Davoli, and F. Cucchietti, "Energy efficiency in the future internet: a survey of existing approaches and trends in energy-aware fixed network infrastructures," *Communications Surveys & Tutorials, IEEE*, vol. 13, no. 2, pp. 223–244, 2011.

[18] "Econet fp7 project, http://www.econet-project.eu," 2010.

[19] "Trend fp7 project, http://www.fp7-trend.eu/," 2010.

[20] "Greentouch, http://www.greentouch.org," 2011.

[21] "Wacs, west africa cable system, http://wacscable.com/."

[22] "Visual networing index, http://www.cisco.com/c/en/us/solutions/service-provider/visual-networking-index-vni/index.html," 2015.

[23] "Software-defined networking: The new norm for networks, white paper. [online], https://www.opennetworking.org/."

[24] "Network functions virtualisation, white paper. [online], http://portal.etsi.org/nfv/nfv˙white˙paper.pdf."

[25] "Acpi, http://www.acpi.info/."

[26] M. Gupta and S. Singh, "Greening of the internet," in *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, pp. 19–26, ACM, 2003.

[27] K. Christensen, B. Nordman, and R. Brown, "Power management in networked devices," *Computer*, vol. 37, no. 8, pp. 91–93, 2004.

[28] R. Bolla, R. Bruschi, F. Davoli, and F. Cucchietti, "Energy efficiency in the future internet: a survey of existing approaches and trends in energy-aware fixed network infrastructures," *Communications Surveys & Tutorials, IEEE*, vol. 13, no. 2, pp. 223–244, 2011.

[29] J. Noguera and I. O. Kennedy, "Power reduction in network equipment through adaptive partial reconfiguration," in *Field Programmable Logic and Applications, 2007. FPL 2007. International Conference on*, pp. 240–245, IEEE, 2007.

[30] "Vmware. http://www.vmware.com,"

[31] "Kvm hypervisor, http://www.linux-kvm.org/page/main˙page."

[32] "Xen hypervisor, http://www.xenproject.org/."

[33] "Microsoft hyperv, http://www.microsoft.com/en-us/server-cloud/solutions/virtualization.aspx."

[34] "lxc, https://linuxcontainers.org/."

[35] "Docker, https://www.docker.com/."

[36] "jailhouse, https://github.com/siemens/jailhouse."

[37] W. Felter, A. Ferreira, R. Rajamony, and J. Rubio, "An updated performance comparison of virtual machines and linux containers," *technology*, vol. 28, p. 32, 2014.

[38] "Docker containers performance in vmware vsphere. vmware vroom, http://blogs.vmware.com/performance/2014/10/docker-containers-performance-vmware-vsphere.html."

[39] I. Stojmenovic and S. Wen, "The fog computing paradigm: Scenarios and security issues," in *Computer Science and Information Systems (FedCSIS), 2014 Federated Conference on*, pp. 1–8, IEEE, 2014.

[40] F. Bonomi, "Connected vehicles, the internet of things, and fog computing," in *Proceedings of the Eighth ACM International Workshop on VehiculAr Internetworking (VANET'2011), Las Vegas, USA*, 2011.

[41] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*, pp. 13–16, ACM, 2012.

[42] “Network functions virtualisation, [whitepaper] https://portal.etsi.org/portals/0/tbpages/nfv/docs/nfv˙white˙paper3.pdf.”

[43] O. Committee *et al.*, “Software-defined networking: The new norm for networks,” *Open Networking Foundation*, 2012.

[44] “Openflow protocol, http://archive.openflow.org/.”

[45] “Open networking foundation, https://www.opennetworking.org/.”

[46] D. Kreutz, F. M. Ramos, P. Esteves Verissimo, C. Esteve Rothenberg, S. Azodolmolky, and S. Uhlig, “Software-defined networking: A comprehensive survey,” *proceedings of the IEEE*, vol. 103, no. 1, pp. 14–76, 2015.

[47] “Cisco energywise, http://www.cisco.com/,” 2009.

[48] G.-Q. Zhang, G.-Q. Zhang, Q.-F. Yang, S.-Q. Cheng, and T. Zhou, “Evolution of the internet and its cores,” *New Journal of Physics*, vol. 10, no. 12, p. 123027, 2008.

[49] D. T. Neilson, “Photonics for switching and routing,” *Selected topics in quantum electronics, IEEE Journal of*, vol. 12, no. 4, pp. 669–678, 2006.

[50] M. Bohr, “A 30 year retrospective on dennard’s mosfet scaling paper,” *Solid-State Circuits Society Newsletter, IEEE*, vol. 12, no. 1, pp. 11–13, 2007.

[51] C. Hu, C. Wu, W. Xiong, B. Wang, J. Wu, and M. Jiang, “On the design of green reconfigurable router toward energy efficient inter-

net," *Communications Magazine, IEEE*, vol. 49, no. 6, pp. 83–87, 2011.

[52] "Power management capabilities of the future energy telecommunication fixed network nodes, http://www.etsi.org/deliver/etsi˙es/203200˙203299/203237/01.01.01˙60/es˙203237v010101p.pdf."

[53] R. Bolla, R. Bruschi, F. Davoli, P. Donadio, L. Fialho, M. Collier, A. Lombardo, D. Reforgiato, V. Riccobene, and T. Szemethy, "A northbound interface for power management in next generation network devices," *Communications Magazine, IEEE*, vol. 52, no. 1, pp. 149–157, 2014.

[54] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "Openflow: enabling innovation in campus networks," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 2, pp. 69–74, 2008.

[55] A. Manzalini and R. Saracco, "Software networks at the edge: a shift of paradigm," in *Future Networks and Services (SDN4FNS), 2013 IEEE SDN for*, pp. 1–6, IEEE, 2013.

[56] "Middleboxes: Taxonomy and issues, https://tools.ietf.org/html/rfc3234," 2002.

[57] "Linux cpu freqs governors, https://www.kernel.org/doc/documentation/cpu-freq/governors.txt."

[58] D. R. Suleiman, M. A. Ibrahim, and I. I. Hamarash, "Dynamic voltage frequency scaling (dvfs)

for microprocessors power and energy reduction, http://www.emo.org.tr/ekler/035226640b6b89f˙ek.pdf.”

[59] G. Gibb, J. W. Lockwood, J. Naous, P. Hartke, and N. McKeown, “Netfpga - an open platform for teaching how to build gigabit-rate network switches and routers,” *Education, IEEE Transactions on*, vol. 51, no. 3, pp. 364–369, 2008.

[60] “Netfpga reference router, https://github.com/caustic/netfpga-wiki/wiki/referencerouterwalkthrough.”

[61] A. Lombardo, C. Panarello, D. Reforgiato, and G. Schembra, “Power control and management in the netfpga gigabit router,” in *Future Network & Mobile Summit (FutureNetw), 2012*, pp. 1–8, IEEE, 2012.

[62] “Openvswitch, http://openvswitch.org/.”

[63] “Xenapi, http://www.xenproject.org/developers/teams/xapi.html.”

[64] M. Ahmadi, *High-performance Processing in Networked and Grid Environments*. PhD thesis, TU Delft, Delft University of Technology, 2010.

[65] *Queueing Systems. Volume 1: Theory. (Book).*

[66] “What is network latency and why does it matter? (whitepaper), https://www.o3bnetworks.com/wp-content/uploads/2015/02/white-paper˙latency-matters.pdf.”