Escola Tècnica Superior d'Enginyeria de Telecomunicació de Barcelona
UNIVERSITAT POLITÈCNICA DE CATALUNYA

PROJECTE FINAL DE MÀSTER

Embedding of Virtual Network requests over heterogeneous wireless and wired networks

Estudis: *Màster en Enginyeria Telemàtica*
Autora: *Montserrat Bellot Aubanell*
Directora: *Cristina Cervelló i Pastor*

Castelldefels, Març 2014

DEDICATORIA

Al meu marit Luis per la paciència que ha tingut i tot el suport que m'han donat.

A la meva família i amics que sempre han estat al meu costat.

AGRAÏMENTS

Vull agrair especialment a la Doctora Cristina Cervelló i Pastor el temps que ha dedicat a ajudar-me en aquest projecte, des de un inici m'ha guiat en el procés d'elaboració, contribuint amb les seves idees i supervisant els resultats.

També vull agrair al departament d'Enginyeria Telemàtica que em va ajudar a assolir els estudis del màster.

TABLE OF CONTENTS

FIGURE INDEX

RESUM DEL PROJECTE

La investigació en l'àrea de xarxes inalàmbriques s'ha incrementat en els darrers anys a causa del seu potencial per a proveir accés a Internet a la que s'anomena "l'última milla". Concretament, en campus, àrees rurals, àrees metropolitanes, etc. A pesar d'haver obtingut avenços d' investigació en aquesta àrea, encara queden reptes pendents com la virtualització de xarxes inalàmbriques.

En el present projecte integrem els principis de "Software-Defined Networking" (SDN) i virtualització de xarxes (NFV) en una xarxa inalàmbrica connectada a una xarxa cablejada formada per commutadors OpenFlow. SDN i NFV són una nova forma de desplegar infraestructures de xarxa. Aquestes tecnologies poden permetre la gestió dinàmica de recursos i serveis. SDN separa la capa de control de la capa de transmissió i proporciona una visió centralitzada de la xarxa distribuïda. SDN contribueix a organització i automatització del trànsit de la xarxa permetent prendre decisions a partir de polítiques establertes. NFV se centra en l'optimització de la gestió dels serveis de la xarxa a partir de la virtualització. La virtualització és una tecnologia que permet l'execució de múltiples arquitectures de xarxa heterogènies en una xarxa de substrat compartit . Un dels components més importants per a la virtualització de la xarxa és la inserció d'una xarxa virtual, que proporciona una forma d'assignar a cada petició virtual els recursos físics de la xarxa (p.e. CPU,BW).

Aquest projecte es centra en l'algorisme del controlador SDN que, dinàmicament, aprovisiona i gestiona els recursos virtuals. L'objectiu és integrar les peticions de xarxes virtuals sobre una xarxa heterogènia integrada. La funció d'orquestració del controlador treballa conjuntament amb el servidor NFV per gestionar i administrar els recursos de xarxa, amb l'objectiu de disposar d'un control coordinat extrem a extrem.

El projecte s'organitza de la següent manera. Inicialment es presenta un estudi de les xarxes inalàmbriques "wireless mesh networks (WMN)" per familiaritzar als lectors amb aquestes xarxes. A continuació proposem un algorisme dinàmic basat en el de Dijkstra que, de manera eficient, utilitza els recursos de la xarxa física (substrat) per servir peticions d'usuaris i trobar la millor trajectòria al llarg de una xarxa heterogènia (inalàmbrica i fixa). La mètrica utilitzada en la xarxa inalàmbrica pren en consideració les interferències provocades per altres canals "Interflow" i les interferències produïdes per un mateix canal al llarg del camí "IntraFlow". A causa de la naturalesa no isotònica de la interferència "IntraFlow" aquesta mètrica no és compatible amb l'algorisme de Dijkstra. Per solventar aquest problema, s'implementa un mecanisme per convertir la mètrica no isotònica en isotònica. Amb l'objectiu d'avaluar aquesta proposta, es programa en java un marc que ens permet simular l'algorisme que hem dissenyat i estudiar el seu rendiment a través d'extenses simulacions en diferents escenaris. Els resultats obtinguts els hem analitzat i presentat en gràfiques excel. Per acabar, hem suggerit pautes per al disseny d'aquest tipus de xarxes i s'han proposat treballs futurs.

RESUMEN DEL PROYECTO

La investigación en el área de redes inalámbricas se ha incrementado en los últimos años, debido a su potencial para proveer acceso a Internet a la que se denomina "la última milla". Concretamente, campus, áreas rurales, áreas metropolitanas, etc. A pesar de haber obtenido significativos avances de investigación, todavía quedan retos pendientes como la virtualización de redes inalámbricas.

En el presente proyecto integramos los principios de " Software-Defined Networking " (SDN) y virtualización de redes (NFV) en una red inalámbrica conectada a una red cableada formada por conmutadores Openflows. SDN y NFV son una nueva forma de desplegar infraestructuras de red. Estas tecnologías pueden permitir la gestión dinámica de recursos y servicios. SDN separa la capa de control de la capa de transmisión y proporciona una visión centralizada de la red distribuida. SDN contribuye a la organización y automatización del tráfico de la red permitiendo tomar decisiones a partir de políticas establecidas. NFV se centra en la optimización de la gestión de los servicios de la red a partir de la virtualización. La virtualización es una tecnología que permite la ejecución de múltiples arquitecturas de red heterogéneas en una red de sustrato compartido. Uno de los componentes más importantes para la virtualización de red es la inserción de una red virtual, que proporciona una forma de asignar a cada petición virtual los recursos físicos de la red (p.e. CPU, BW).

Este proyecto se centra en el algoritmo del controlador SDN que, dinámicamente, provisiona y gestiona los recursos virtuales. El objetivo es dar servicio a peticiones de redes virtuales sobre una red inalámbrica y fija integrada. La función de orquestación del controlador trabaja conjuntamente con el servidor NFV para gestionar y administrar los recursos de red, con el objetivo de disponer de un control coordinado extremo a extremo.

La organización del proyecto es la siguiente. Inicialmente se presenta un estudio de las redes inalámbricas "wireless mesh networks (WMN)" para familiarizar a los lectores con estas redes. A continuación, proponemos un algoritmo dinámico basado en el de Dijkstra que de manera eficiente utiliza los recursos de la red física (sustrato) para servir peticiones de usuarios y encontrar la mejor trayectoria a lo largo de una red inalámbrica y fija integrada. La métrica utilizada en las redes inalámbricas toma en consideración las interferencias debidas a otros canales "Interflow" y las interferencias producidas por un mismo canal a lo largo del camino "IntraFlow”. Esta métrica no es compatible con el algoritmo de Dijkstra, debido a la naturaleza no isotónica de la interferencia "IntraFlow". Para solucionar este problema implementamos un mecanismo para convertir esta métrica no isotónica en isotónica. Con el objetivo de probar nuestra propuesta se programa en java un marco que nos permite simular el algoritmo que hemos diseñado y evaluar su rendimiento a través de extensas simulaciones en diferentes escenarios. Los resultados obtenidos los analizamos y presentamos en gráficas excel. Por último, sugerimos pautas para el diseño de este tipo de redes y proponemos trabajos futuros.

ABSTRACT

Research in the area of wireless mesh networks (WMN) has increased in the last few years, due to its potential for the "last few miles" and the possibility of a large variety of applications in personal, local, campus, and metropolitan areas. Despite the advances in WMN, many research challenges still remain; one of these challenges is the use of virtualization on WMN.

We propose to integrate Software Defined Networking (SDN) and network functions virtualization (NFV) principles in a WMN connected to a wired network formed by OpenFlow switches. SDN and NFV are a new way of deploying network infrastructure. These technologies can enable resources and services orchestration with dynamic provisioning. SDN separates the network's control and forwarding layers and provides a centralized view of the distributed network. It contributes to the network automation that enables policy-based decisions to orchestrate the network traffic. NFV focuses on optimizing the network services building virtualized network functions.

Virtualization is a technology of running multiple heterogeneous network architectures on a shared substrate network. One of the most important components in network virtualization is the virtual network embedding, which provides a way to allocate physical network resources (e.g. CPU and link BW) to virtual network request.

This project is specifically focused on the algorithm of the SDN controller that dynamically provision and manage virtual resources. The goal is to embed Virtual Network requests over an integrated heterogeneous network. The controller's orchestrator function interworks with the NFV server to manage network resources for the coordinated control of the end-to-end infrastructure.

The project is organized as follows. We begin with a survey of WMNs. Then, we propose a global embedding algorithm, based on Dijkstra's, which efficiently uses the resources of the physical substrate network to find the best path along an integrated wireless and wired network. In WMNs, we use a metric that takes into consideration the intraflow and interflow interferences. Due to the non-isotonic nature of the intraflow interference this metric is not compatible with Dijkstra's algorithm. So, to tackle with this problem, we implement a mechanism to convert our non-isotonic metric into an isotonic one. Next, to test this proposal, we program a framework in java that allows us to simulate our embedding algorithm. We evaluate its performance through extensive simulations and under several scenarios. Following, we present the results in graphics and the obtained values are analyzed. Finally, we suggest guidelines for network planning and future works.

Keywords: Wireless Mesh Networks**,** Software Defined Networking, Network Functions Virtualization, Virtual Network Embedding, Non-Isotonic Metrics.

GLOSSARY

AODV	Ad-hoc on Demand Distance Vector
AP	Access Point
BW	Bandwidth
BS	Base Station
CPU	Central Processing Unit
CSC	Channel Switching Cost
DSR	Dynamic Source Routing
DSDV	Distance-Vector Routing
ETX	Expected Transmission Count
ETT	Expected Transmission Time
FIFO	First In First Out
GSR	Global State Routing
IGW	Internet Gateway
IRU	Interference aware Resource Usage
LOS	Line Of Sight
MAC	Medium Access Control
MANET	Mobile Ad hoc Network
MAP	Mesh access point
MIC	Metric of Interference and Channel Switching
NFV	Network functions virtualization
NIC	Network Interface card
NLOS	Non-line-of-sight

NM	Network Manager
NV	Network Virtualization
PC	Portable Computer
QoS	Quality of Service
SDN	Software Defined Networking
VoD	Video on Demand
VN	Virtual Network
VNE	Virtual Network Embedding
VAN	Virtual Access Network
WAN	Wide Area Network
WCETT	Weighted Cumulative ETT
WMN	Wireless Mesh Network

1 Introduction and Objectives

Wireless mesh network (WMN) is a useful wireless technology that is used to provide high bandwidth and expanded coverage access to Internet, to wireless clients. It is used for numerous applications, e.g., broadband home networking, community and neighborhood networks, enterprise networking, building automation, etc. Additionally, there are various attractive qualities as low-cost deployment, robustness and its inheritance of useful characteristics from both the ad-hoc networking paradigm and the traditional wired infrastructure paradigm.

The quantity of research being conducted in the area of wireless mesh networks has increased in the past few years. This recent job has provided excellent surveys focused on: interference modeling, power control, topology control, link scheduling and routing. Despite these significant research efforts on wireless mesh networks, little attention has been paid to virtual network embedding in heterogeneous networks composed by wireless mesh networks interconnected with wired networks.

Network virtualization of a wireless mesh network is an economical way for different subscribers to customize their exclusive access networks through a common network infrastructure. The shared infrastructure is referred to as the physical network or the substrate network. Network virtualization allows a physical network to support multiple virtual networks simultaneously. Such access networks works as a virtual access networks (VAN) over the same WMN infrastructure to satisfy diverse end-to-end requirements. Each virtual access network works independently without any interference with each other. So, each client gets an impression that has a dedicated access network that satisfies its requirements.

The most critical task of network virtualization is the virtual network embedding (VNE) that handles efficient mapping of virtual nodes and virtual links onto physical nodes and links. The characteristics of WMNs make virtual network embedding become a challenging problem. Due to the broadcast nature of wireless links, the link mapping needs to consider the specific interference model of a WMN, and determine the resource allocation for each VAN accordingly to client requirements and guarantee the independence of each VAN.

This project addresses a streaming service in a local neighborhood community, over a heterogeneous network that integrates a multi-channel wireless mesh network with a wired network. We propose to integrate Software Defined Networking (SDN) and network functions virtualization (NFV) principles in Wireless Mesh Networks (WMN) connected to a wired network formed by OpenFlow switches. SDN and NFV are a new way of deploying network infrastructure. These technologies can enable resource and service orchestration with dynamic provisioning. SDN separates the network's control and forwarding layers and provides a centralized view of the distributed network. It contributes to the network automation that enables policy-based decisions to orchestrate network traffic. NFV focuses on optimizing the network services building virtualized network functions.

The use of a centralized network controller and the ability to setup arbitrary paths for data flows make SDN a handy tool to deploy efficient routing algorithms in WMNs. The SDN controller knows the topology and network resources, and can build an efficient path and program the network on an on-demand basis. The controller configures the forwarding behavior of the wired switches by setting rules in their flow tables, using a South-bound protocol. We propose to balance traffic among the gateways of a Wireless Mesh Network, implementing the gateway balancing logic in the controller.

This project is specifically focused on the algorithm of the SDN controller that dynamically provision and manage virtual resources. The controller's orchestrator function interworks with the NFV server to manage network resources for the coordinated control of the end-to-end infrastructure. To perform this, we have designed a virtual access network embedding algorithm that can ensure the coexistence of different requests on the same substrate and provide the most suitable network resources according to the requests requirements, networks constraints (CPU, BW), and network load-balance. Since the virtual access network embedding problem is NP-hard, a heuristic algorithm has been developed based on Dijkstra's algorithm, to find the minimum path weight in both wired and wireless networks. In WMNs we have considered a metric that takes into account intraflow and interflow interferences. Due to the non-isotonic nature of the intraflow interference value, we have defined and implemented a mechanism to convert non-isotonic metrics into isotonic ones.

To test this proposal we have programmed in java a framework that allows us to simulate our global embedding algorithm. This framework provides a powerful tool to test the performance of our proposal through random networks, and virtual network requests. Next we have analyzed and represented the results in excel graphics.

The remainder of this project report is organized as follows:

Chapter 1 contains the introduction and objectives. **Chapter 2** presents a survey of wireless mesh networks, which identify the WMN main features. This chapter aims to help readers getting familiar with WMN.

Chapter 3 is devoted to the project development. First, we describe the project scenario, the starting assumptions, the network operation and the challenges. Then, we focus on the embedding algorithm and the efficient Dijkstra's algorithm, adapted to find the minimum weight path in wired and wireless networks. Next, we study in depth the metric used in Dijkstra that is calculated taking into account intraflow interferences (between nodes on path of the same flow) and interflow interferences (between neighboring nodes competing for the same channel). Following, we tackle the challenge related to the intraflow interference nature that makes our metric become non-isotonic and not compatible with Dijkstra, and therefore, being necessary to convert the non-isotonic metric into isotonic. To solve this problem, we decompose the real network into a virtual one, which provides a solution that allows the use of non-isotonic additive metrics in our algorithm.

Chapter 4 describes the software implementation of the proposed global embedding algorithm. We use Java to program a framework. This framework allows us the evaluation of our embedding algorithm performance on several scenarios of random heterogeneous wireless and wired networks combined with random requests.

Chapter 5 contains the results obtained from the tests performed on the developed application. We modify the following parameters during the tests: networks size (wireless and wired), available resources (BW, CPU, number of GW), number of requests and its requirements parameters. We present the results in excel graphs and we analyze them statistically.

Finally**, Chapter 6** indicates the conclusions about the simulations results. We conclude that the global virtual access network embedding algorithm and its developed software simulator works efficiently and effectively, and detects weaknesses that should be taken into account in network design. In this section, we also provide proposals for future works.

1.1 Objectives

The main objective of this project is to design and implement a global embedding algorithm framework that serves virtual network requests using efficiently the resources of an integrated wired and wireless mesh network

To achieve this goal are necessary several steps and intermediate objectives, which can be classified as follows:

- Study of the state of art of WMNs.
- Study of Network virtualization techniques.
- Design and implementation of an efficient embedding algorithm that first checks the available resources of the network. Then search for multiple VN request embedding candidates. Next, evaluate the embedding candidates, and finally, take those with higher quality.
- Implementation of load-balanced feature in the algorithm.
- In WMNs. Study of different WMNs metrics and selection of a metric model that take into account intraflow and interflow interference.
- In WMNs. Study of the non-isotonic feature in metrics due to the intraflow nature, and search for a mechanism to converts non-isotonic metrics into isotonic ones.
- In WMNs. Use of multiradio, multichannel infrastructure in order to minimize interference.
- Implementation of the global embedding algorithm framework in java.
- Definition of different scenarios and evaluation of our embedding algorithm performance.
- Graphic representation and data analysis.
- Conclusions and guidelines for network designing and future works.

2 Wireless Mesh Networks

Wireless Mesh Network is emerging as a solution for providing last few miles connectivity. There are different qualities that make this paradigm attractive; these are low-cost deployment, robustness and its inheritance of useful characteristics from the ad-hoc networking paradigm and the wired paradigm. WMNs deployment has been proposed as reliable and affordable access networks in metropolitan and also in underdeveloped regions. Here, the aim is to design a network with low-cost access (often by Internet Service Providers) to aid the development of communities. This paradigm has been broadly described by different works [1].

In the present section, we present a brief survey about WMNs. This chapter aims to help readers getting familiar with WMNs.

2.1 Wireless Mesh Networks Architecture

Wireless mesh network is a communications network made up of wireless mesh routers and wired/wireless clients and gateways, for an example of WMN architecture see Fig. 2. 1.

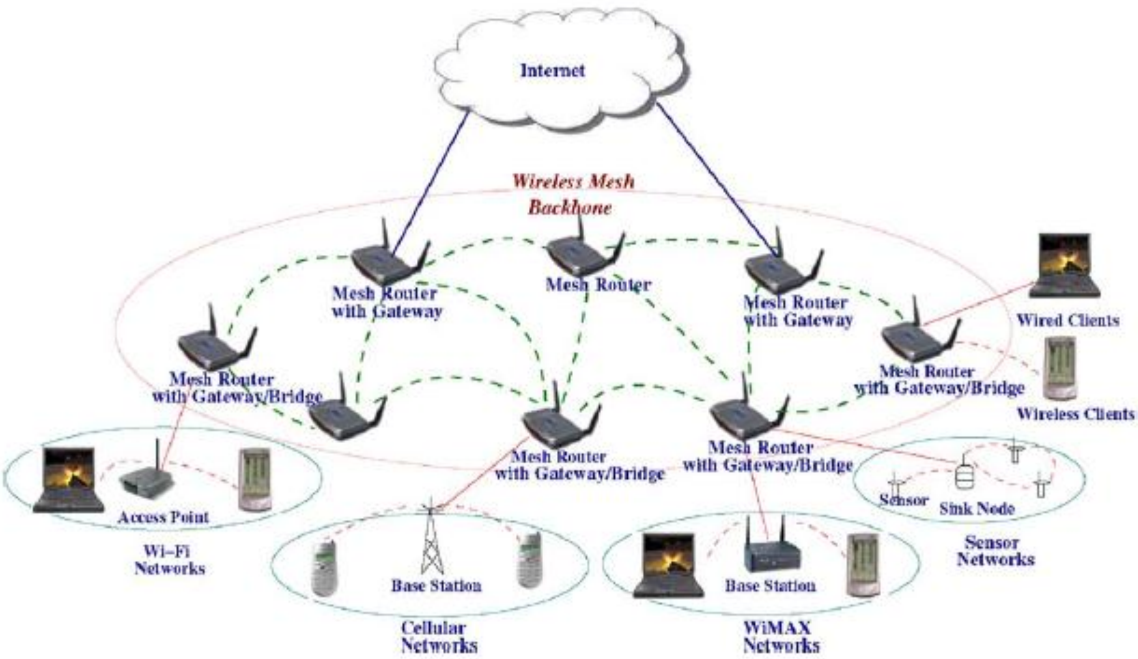


Fig. 2. 1. Wireless Mesh Architecture. Extracted from [1].

A mesh router is usually equipped with multiple wireless interfaces built on either the same or different wireless access technologies. Mesh routers can be built based on dedicated computer systems (e.g., embedded systems) and look compact or they can also be built based on general-purpose computer systems (e.g., laptop/ desktop PC). Each mesh router could operate as a backbone router forwarding packets on behalf of other nodes and additionally, as an access point (AP) providing access service for wireless clients.

The mesh clients can be either stationary or mobile, they are often laptops, cell phones and other wireless devices equipped with wireless network interface cards (NICs) that can connect directly to wireless mesh routers. Customers without wireless NICs can access WMNs by connecting to wireless mesh routers through, for example, Ethernet. The hardware platform and the software for mesh clients can be much simpler than those for mesh routers.

To enable the integration of WMNs with various existing wireless networks or fixed networks, some mesh routers include the gateway/bridge functionalities. A mesh router that is connected to the wired Internet is referred to as an Internet gateway (IGW). Multiple IGWs can be deployed in a WMN to increase the network capacity.

After a client is associated with an AP, the packets of the client are forwarded by the wireless multi-hop backbone between the AP and IGW. This general form of WMNs can be visualized as an integration of two planes where the access plane provides connectivity to the clients while the forwarding plane relays traffic between the mesh routers. This is a convenient and low-cost approach to provide Internet access.

The architecture of WMNs according to [1] can be classified into three main groups based on the functionality of the nodes:

Infrastructure/Backbone WMNs. This architecture is the most commonly used and it is shown in Fig. 2. 1, where dash and solid lines indicate wireless and wired links, respectively. Mesh routers are static and form an infrastructure of wirelessly connected routers to serve the users. The routers have multiple radio interfaces for wireless backhaul and one radio interface for connection with end users. The mesh routers form a mesh of self-configuring, self-healing links among themselves. With gateway functionality, mesh routers can be connected to wire networks. Clients with Ethernet interface can be connected to mesh routers via Ethernet links. Clients with the same radio technologies as mesh routers, they can directly communicate with mesh routers. If different radio technologies are used, clients must communicate with the base stations that have Ethernet connections to mesh routers. The WMN infrastructure/ backbone can be built using various types of radio technologies; the mostly used is IEEE 802.11 technologies. An example of this architecture is a community and neighborhood networks that can be built using infrastructure meshing. The mesh routers are placed on the roof of houses in a neighborhood, which serve as access points for users inside the homes and along the roads. Typically, two types of radios are used in the routers, i.e., for backbone communication and for user communication, respectively. The mesh backbone communication can be established using long-range communication techniques including directional antennas.

Client WMNs. Client meshing provides peer-to-peer networks among client devices. In this type of architecture, client nodes constitute the actual network to perform routing and configuration functionalities as well as providing end user applications to customers. Hence, a mesh router is not required for these types of networks. The basic architecture is shown in Fig. 2. 2. In Client WMNs, a packet destined to a node in the network is sent through multiple nodes to reach the destination. Client WMNs are usually formed using one type of radios on devices. Moreover, the requirements on end-user devices are increased when compared to infrastructure meshing, since, in Client WMNs, the end-users must perform additional functions such as routing and self-configuration.

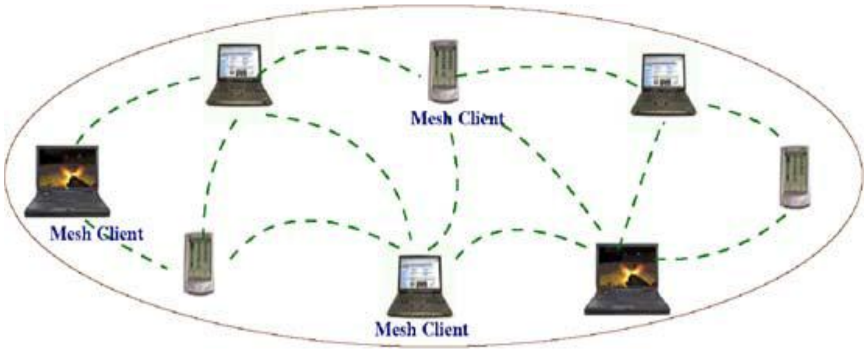


Fig. 2. 2. Client WMNs. Extracted from [1]

Hybrid WMNs. This architecture is the combination of infrastructure and client meshing as shown in Fig. 2. 3**.** Mesh clients can access the network through mesh routers as well as directly meshing with other mesh clients. While the infrastructure provides connectivity to other networks such as the Internet, Wi-Fi, WiMAX, cellular, and sensor networks; the routing capabilities of clients provide improved connectivity and coverage inside the WMN.

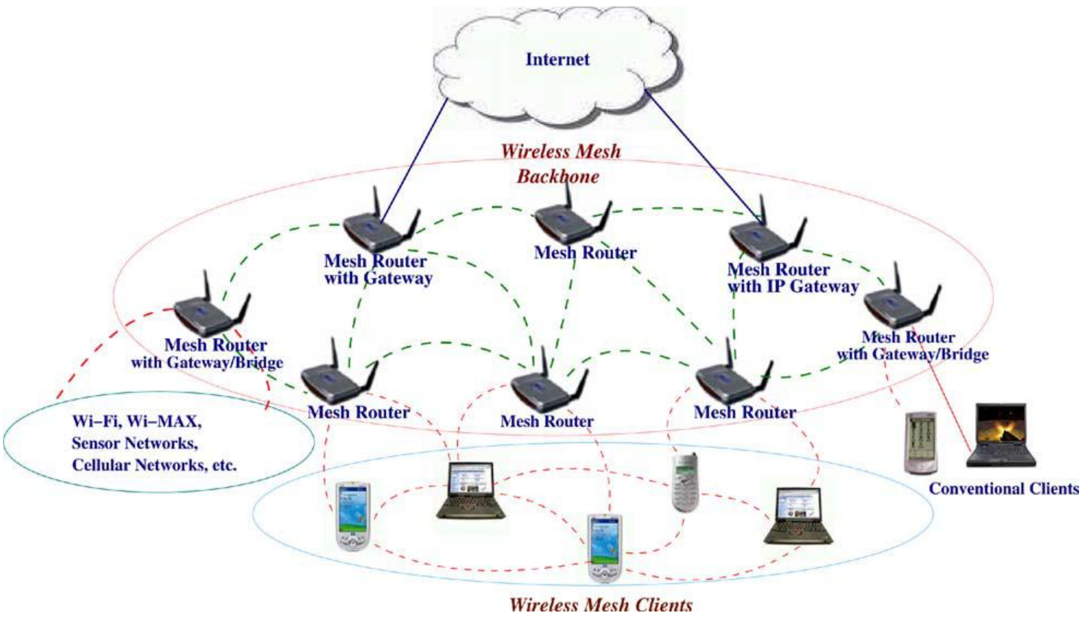


Fig. 2. 3. Hybrid WMNs. Extracted from [1]

2.2 Wireless Mesh Networks vs Ad-Hoc Network

A wireless mesh network can be seen as a special type of wireless ad-hoc network, but there are some differences that are listed below.

First of all, ad-hoc network nodes are often constrained by energy resources in front of mesh routers that have no limitations regarding energy consumption and can perform more complex functions.

Additionally, wireless mesh networks often have a more planned configuration, and may be deployed to provide dynamic and cost effective connectivity over a certain geographic area.

Wireless mesh networks have a relatively stable topology except for the occasional failure of nodes or addition of new nodes. The path of traffic, being aggregated from a large number of end users, changes infrequently. Practically all the traffic in an infrastructure mesh network is either forwarded to or from a gateway, while in ad hoc networks or client mesh networks the traffic flows between arbitrary pairs of nodes.

2.3 WMN Management

Network management, can be defined as the deployment, coordination and integration of hardware, software and human elements to monitor, analyze, evaluate and control network resources in order to meet the operational performance and QoS requirements in a network.

WMN management could be decentralized (network management is distributed among various network elements) or centrally managed (with a central network manager).

2.4 WMNs Characteristics and benefits

WMNs provide non-line-of-sight (NLOS) connectivity among the users without direct line-of-sight (LOS) links, extending the coverage range of current wireless networks without sacrificing the channel capacity.

WMNs enhance network performance, because of flexible network architecture, easy deployment and configuration, fault tolerance, and mesh connectivity, i.e., multipoint-to-multipoint communications. WMNs provide support for ad hoc networking, and capability of self-forming, self-healing, and self-organization.

Mesh routers usually do not have strict constraints on power consumption. However, mesh clients may require power efficient protocols. Mesh routers usually have minimal mobility, while mesh clients can be stationary or mobile nodes.

Mesh routers can be equipped with multiple radios to perform routing and access functionalities. This enables separation of two main types of traffic in the wireless domain. While routing and configuration are performed between mesh routers, the access to the network by end users can be carried out on a different radio. This significantly improves the capacity of the network. On the other hand, in ad hoc networks, these functionalities are performed in the same channel, and as a result, the performance decreases.

WMNs provide compatibility and interoperability with existing wireless networks. Wireless mesh networks can be implemented with various wireless technology including 802.11,802.15,802.16, cellular technologies or combinations of more than one type. Moreover WMNs enable integration of various existing networks such as Wi-Fi, the Internet, P2P, cellular and sensor networks providing end-users with services of this WMNs.

Finally, WMNs benefit from incremental expansion, because their ease of deployment, affordable cost and their robustness. These benefits of WMNs consistently motivate researchers to study their characteristics for better performance.

2.5 WMN Application Scenarios

Mesh networks may involve different solutions depending on communication requirements, According to [1], the following scenarios are considered.

Broadband home networking. A home (even a small one) usually has many dead zones without service coverage. Mesh networking, as shown in Fig. 2. 4, can resolve all these issues. The access point could be wireless mesh routers with mesh connectivity established among them. Therefore, the communication between these nodes becomes much more flexible and more robust to network faults and link failures. Communication within home networks can be realized through mesh networking without going back to the access hub all the time. Thus, network congestion due to backhaul access can be avoided. In this application, wireless mesh routers have no constraints on power consumptions and mobility.

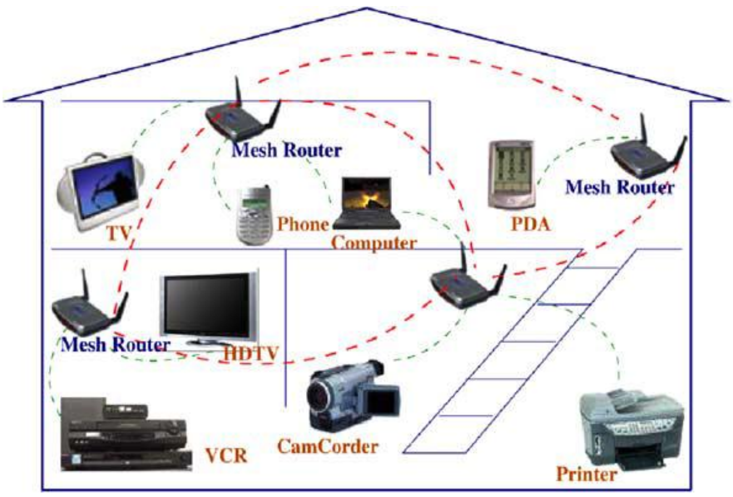


Fig. 2. 4. WMNs for broadband home networking. Extracted from [1].

Community and neighborhood networking (Fig. 2. 5). In a community, the common architecture for network access is based on cable or DSL connected to the Internet, and the last-hop is wireless by connecting a wireless router to a cable or DSL modem. This type of network access has several drawbacks:

- Even if the information must be shared within a community or neighborhood, all traffic must flow through Internet. This significantly reduces the network resource utilization.
- Large percentage of areas in between houses is not covered by wireless services.
- An expensive but high bandwidth gateway between multiple homes or neighborhoods may not be shared and wireless services must be set up individually. As a result, network service costs may increase.
- Only a single path may be available for one home to access the Internet or communicate with neighbors.

WMNs mitigate the above disadvantages through flexible mesh connectivity between homes, WMNs can also enable many applications such as distributed file storage, distributed file access, and video.

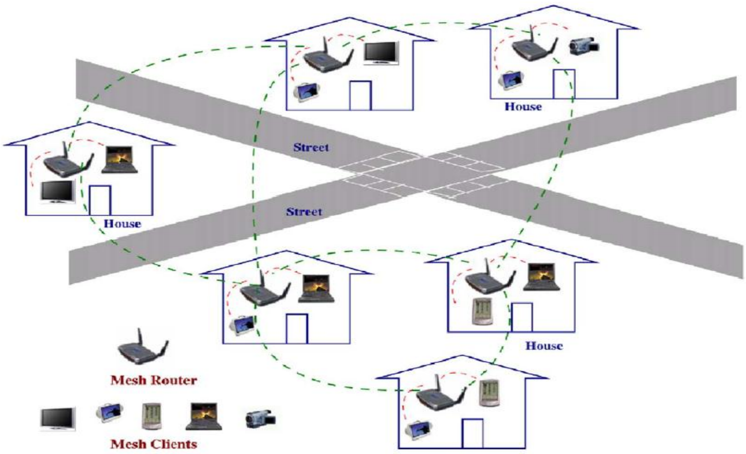


Fig. 2. 5. WMNs for community networking. Extracted from [1].

Enterprise networking (Fig. 2. 6). This can be a small network within an office or a medium-size network for all offices in an entire building, or a large scale network among offices in multiple buildings. The service model of enterprise networking can be applied to many other public and commercial service networking scenarios such as airports, hotels, shopping malls, convention centers, sport centers, etc.

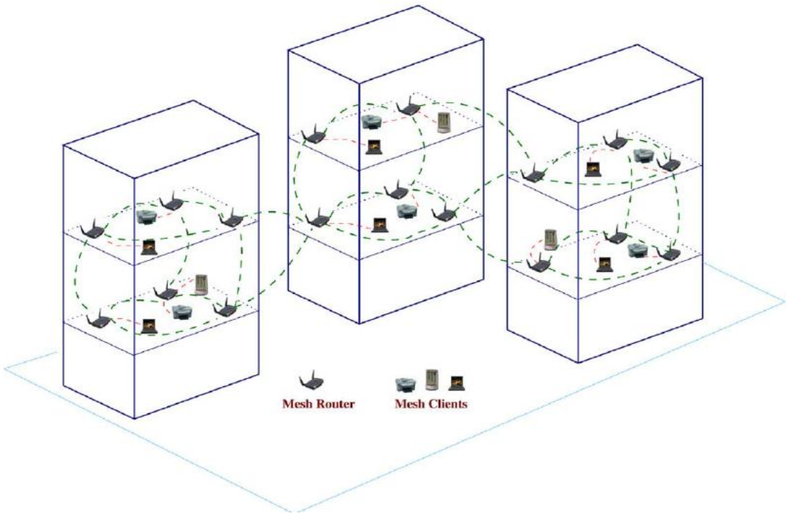


Fig. 2. 6. WMNs for enterprise networking. Extracted from [1].

Metropolitan area networks (Fig. 2. 7). Wireless mesh MAN is an economic alternative to broadband networking, especially in underdeveloped regions. It covers a potentially much larger area than home, enterprise, building, or community networks, as shown.

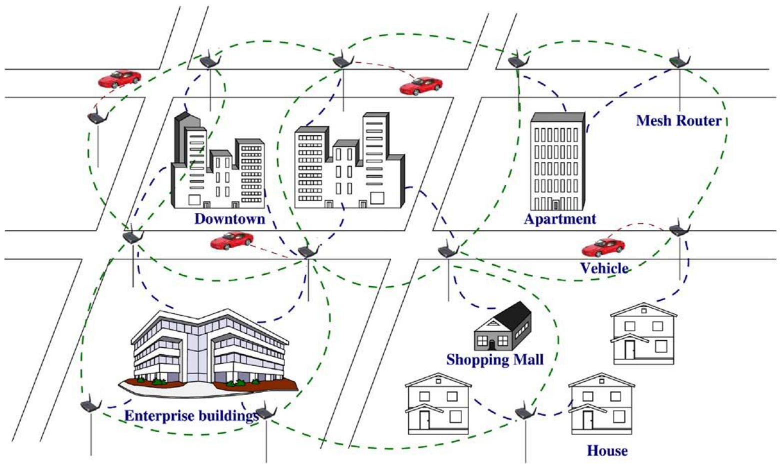


Fig. 2. 7. Metropolitan area Networks. Extracted from [1]

Transportation systems (Fig. 2. 8)**.** Instead of limiting IEEE 802.11 or 802.16 accesses to stations and stops, mesh networking technology can extend access into buses, ferries, and trains. Thus, convenient passenger information services, remote monitoring of in-vehicle security video and driver communications can be supported. To enable such mesh networking for a transportation system it is needed the technique of the high-speed mobile backhaul from a vehicle (car, bus, or train) to the Internet and mobile mesh networks within the vehicle, as shown in Fig. 2. 8.

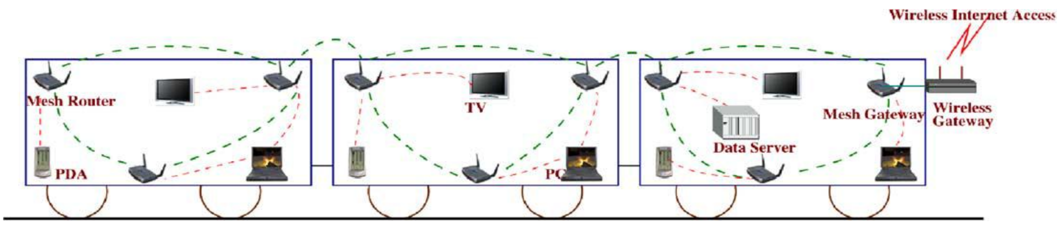


Fig. 2. 8. WMNS for Transportation Systems. Extracted from [1].

Building automation (Fig. 2. 9). WMNs could provide a solution to monitor and control in a building, various electrical devices including power, light, elevator, air conditioner, etc.,

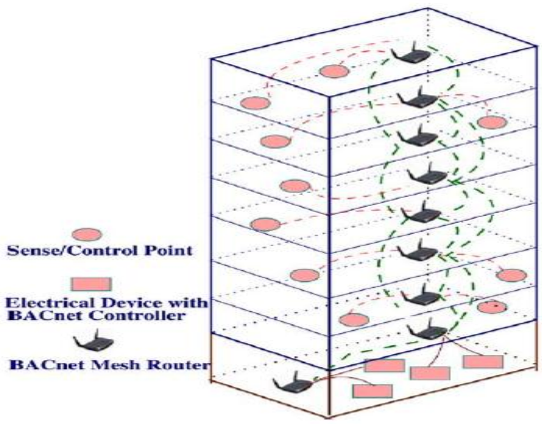


Fig. 2. 9. WMNs for building automation. Extracted from [1].

Health and medical systems. In a hospital or medical center, monitoring and diagnosis data need to be processed and transmitted from one room to another for various purposes.

Security surveillance systems. For enterprise buildings, shopping malls, grocery stores, etc.

In addition to the above applications, WMNs can also be applied to Spontaneous (Emergency/ Disaster) Networking and P2P Communications.

2.6 Design Challenges

An overview of the fundamental problems and design objectives that affect the performance of WMNs is described below.

Interference measurement and modeling. Every transmission between wireless mesh routers creates interference in its neighborhood, this is the major challenge in the design of a WMNs. It is needed a Realistic estimation of interference.

Power Control. Power control and topology control mechanisms in WMNs mainly deal with assigning transmission power levels to nodes such that the traffic demands are satisfied with better overall throughput. High power level for transmission is necessary for successful reception at the receiver, but, high power transmission causes high interference and MAC layer collisions at other unintended receivers and decreases the achievable network capacity. In WMN Energy conservation is not an objective.

Topology Control. Power control and topology control mechanisms determine the network connectivity and underlying physical layer topology. All links of such a topology can carry the traffic between the nodes, and the reception rate depends on the quality of the link.

Link scheduling. Link scheduling strategies estimate transmission conflicts between links of the routing paths using the interference model and try to achieve a conflict-free feasible transmission schedules. A scheduling scheme has to solve: medium access, collision detection/avoidance and transmission scheduling techniques. Various characteristics of links such as quality, stability and reliability play an important role in routing metric design which is used by the routing protocol.

Channel/radio assignment. To minimize the interference effects between interfering links. There are used multiple channels, which are assigned to single or multiple radios at nodes. Additionally, to further mitigate the interference effects, interfering links are sometimes separated in the frequency domain. Channel/radio assignment schemes try to arrange nearby transmissions on orthogonal or minimally overlapping channels in single or multi-radio WMNs.

Routing. Routing strategy determines reliable and high throughput end-to-end paths between the source and destination of data. Various characteristics of links such as quality, stability and reliability play an important role in routing metric design which is used by the routing protocol. Namely, low inter-path and intra-path interference, load balancing, higher reliability and throughput.

The problems described above are highly interrelated. For example, it may happen that link scheduling does not yield a high throughput schedule because of the existence of high interference links in the network. This may require the traffic of such links to be re-routed on shorter and lower interference links. This points the way to treating link scheduling and routing as a joint problem.

2.7 Interference in wireless mesh networks

Interference is a major performance limiting factor in wireless mesh networks due to the shared nature of the wireless medium. Interference occurs since a wireless link does not have a dedicated bandwidth. Transmissions from neighboring nodes may compete for the same bandwidth and hence interfering with the transmissions of neighboring links.

Interference can be classified as intraflow interference (between nodes on path of the same flow), and interflow interference (between neighboring nodes competing for the same channel).

The interference may also be distinguished into two categories:

Radio interference or physical interference: It is the superposition of signals which results in bits alterations of original signal causing packet losses. It occurs after the data transmission.

Channel contention interference or logical interference. Is that due to the medium access protocols where the station needs to wait because the medium is occupied. Occurs before the data transfer happens.

2.7.1 Interflow and Intraflow interference

Interflow interference. It occurs when a flow through the link not only consumes the bandwidth of the nodes along its path, it also contends for bandwidth with the nodes that are in the neighboring area of its path. For instance, in 2.10, an effective routing metric should give to the path A->B->D a lower weight than to the path A->C->D; since path A->B->D has much less inter-flow interference than path A->C->D

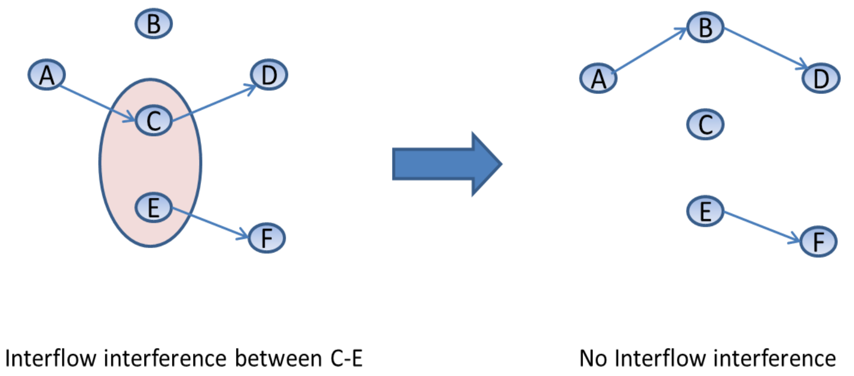


Fig. 2. 10. Example of Interflow interference.

Intra-flow interference

It is produced when nodes on the path of the same flow compete with each other for channel bandwidth. Such intra-flow interference increases the bandwidth consumption of the flow at each of the nodes along the path and causes the throughput of the flow to degrade sharply and the delay at each hop to increase dramatically as the hop count of the flow increases. Therefore, the potential of increased congestion levels due to such intra-flow interference must be considered when designing a routing metric for mesh networks. For example, as shown in Fig. 2.11, an interference aware metric should give path A->B->D a higher weight than path A->C->D, since the reuse of channel 1 on A->B->C creates much more intra-flow interference than that in path A->D->C.

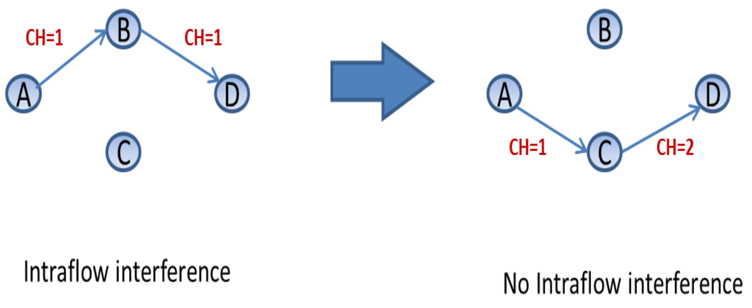


Fig. 2. 11. Example of intraflow interference.

2.7.2 Number of radios in a wireless mesh router

The number of radios in a wireless mesh router, affects the amount of interference in the network. Depending on the number of radios, the network is categorized on three types:

Single radio wireless mesh networks. It is the basic wireless mesh network configuration, where each node has a single radio and all the radios are tuned to the same frequency. It has important limitations due to the channel access procedure of several nodes contending for the same channel and causing interferences.

Dual radio wireless mesh networks. These mesh routers have different radios for client access and backhaul. Routers have two radios operating in two different frequencies. The amount of interference is lesser than single radio mesh networks. But the use of a single frequency for backhaul causes interference among routers.

Multi radio wireless mesh networks have the same features of dual radio mesh networks with different frequencies for client access and backhaul access. The backhaul nodes have multiple radios and hence interference among backhaul routers is considerably reduced.

2.7.3 Interference measurement and modeling

Interference is unpredictable, which challenges the design of all upper-layer protocols. There have been proposed various ways to model the impact of interference, the most important models are discussed below:

Protocol Interference Model [2][31]: Communication between nodes *u* and *v* results in collision-free data reception at node *v* if no other node within a certain interference range from *v* is transmitting simultaneously. This model has been further extended to consider link layer reliability using acknowledgments in which interference range of node u is also counted for interference. This is often referred as disk model (or double disk model) where interference is assumed to be a binary phenomenon developed in certain fixed distance from the source and the destination of any active link. Such Interference range of any node is often assumed to be a constant times larger than its communication range.

Physical Interference Model [2][31]: Communication between nodes u and v results in collision-free data reception at node v if SINR (Signal to Interference and Noise Ratio) at node v is above a certain threshold β. If P_{vu} is the signal power received at node v from u, a packet from node u is successfully received at node v *if*:

$$\frac{P_{vu}}{N+\sum_{i\in I} P_{vi}} \geq \beta \quad \text{(Eq 2.1)}$$

where, I is a set of nodes simultaneously transmitting. N is the background noise and β is a physical layer dependent constant. The threshold-based version of this SINR model was extended to a more general graded probabilistic SINR model [81] which also considers SINR lesser than the threshold and predicts the probability of successful reception.

K−hop Interference Model [2][33]: Two links within K hops distance from each other cannot successfully transmit at the same time. The simplest case of such a model (with K = 1) is often referred as node-exclusive interference model where only restriction imposed by interference is that a node cannot transmit and receive on two separate links concurrently.

These interference models can be represented using a conflict graph. In a conflict graph, every link in the network is represented as a vertex and two vertices share an edge if and only if the corresponding edges interfere with each other. Depending on the interference model and its directionality characteristics, the resultant conflict graph can be undirected (double-disk model or k-hop interference model) or directed (physical interference model).

The above models assume the use of Omni-directional antennas. To increase the throughput capacity it could be used directional antennas which cause less interference. For this kind of antennas it is needed to improve network design due to their directionality characteristics.

Modeling link quality, capacity and the effect of interferences can be very difficult as the wireless environment is often a complex combination of various parameters. Researchers have proposed to rely on realistic measurements like link quality, packet delivery rate, etc., to capture the effects of interferences.

Scalability is an important factor since large number of measurements can become intractable. It is necessary that such measurements are accurate and there exists an efficient way to collect them periodically without incurring much overhead.

The above mentioned problems are highly interrelated. For example, it may happen that link scheduling does not obtain a high throughput schedule because of the existence of high interference links in the network. This may require the traffic of such links to be re-routed on shorter and lower interference links.

2.8 Routing protocols for mesh networks

It is necessary to first understand what routing protocols best fit mesh networks to understand the necessary properties of routing metrics to support effective routing in mesh networks. Depending on when routes are calculated, the possible routing protocols for mesh networks can be divided into two categories.

On-demand Routing: Originally proposed for ad hoc networks, on-demand or reactive routing protocols (e.g., DSR [11], AODV [12], MCR [13], LBAR [14], and DLAR [15],) only create a route between a pair of source and destination nodes when the source node needs to send packets to the destination. Network wide flooding is usually used to discover routes when they are needed. For ad hoc networks, since there are frequent link breaks caused by the mobility of nodes, flooding-based route discovery provides high network connectivity and relatively low message overhead compared to proactive routing protocols. However, in mesh networks, links usually have much longer expected lifetimes due to the static nature of nodes. Since the frequency of link breaks is much lower than the frequency of flow arrivals in mesh networks, flooding-based route discovery is both redundant and very expensive in terms of control message overhead. Therefore, on-demand routing protocols are generally not scalable or appropriate for mesh networks.

Proactive Routing: Based on how packets are routed along the paths, in proactive routing protocols, each node maintains one or more tables containing routing information to every other node in the network. All nodes update these tables to maintain a consistent and up-to-date view of the network. When the network topology changes, the nodes propagate update messages throughout the network to maintain consistent and up-to-date routing information about the whole network. Based on how packets are routed along the paths, proactive routing can further be divided into two subcategories source routing and hop-by-hop routing.

1. ***Source Routing:*** Source routing, such as LQSR [25], imposes minimal burden on relaying nodes since the source node calculates the route for a flow and puts the entire path of the flow in the packet headers. Intermediate nodes only need to relay packets based on the paths in the packet headers. Considering that the packet size in mesh networks is usually very small to cope with the high bit error rate of wireless channels, putting the entire path in the packet header imposes expensive message overhead.
.
2. ***Hop-by-hop Routing:*** In hop-by-hop routing, every node maintains a routing table that indicates the next hops for the routes to all other nodes in the network. For a packet to reach its destination, it only needs to carry the destination address. Intermediate nodes forward the packet along its path based only on the destination address. Due to its simple

forwarding scheme and low message overhead, hop-by-hop routing is dominant in wired networks. Similar reasons also make hop-by-hop routing the most preferable for mesh networks. Hop-by-hop routing requires careful design of its routing metrics to ensure loop-free packet forwarding.

2.9 Routing metrics for wireless mesh networks

Designing routing metrics is critical for performance in wireless mesh networks. A good routing metric should capture the quality of the wireless links to meet the required criteria and to aid in selecting a better path. In this section we provide a survey of the requirements and routing metrics.

2.9.1 Factors affecting routing metrics

To do an effective analysis, it is necessary to take into account the following factors.

Different routing protocols may impose different costs in terms of message overhead and management complexity, so it is important to be sure that the design of the routing metrics is compatible with effective routing protocols.

Due to the shared nature of the wireless medium, a wireless link in a mesh network does not have dedicated bandwidth so, the transmissions of neighboring nodes may also contend for the same bandwidth.

To reflect the quality of a link, an effective routing metric must be able to capture the interference between competing flows.

Since current wireless cards can be configured to different channels, wireless links that are configured to different channels may not interfere with each other even if they are physically located near each other. Therefore, effective routing metrics must be able to consider the channel assignments of links to understand the impact of interference on the performance of paths.

Different routing protocols may impose different requirements on the design of their routing metrics.

2.9.2 Routing metrics requirements

To ensure good performance, routing metrics must satisfy four requirements.

1. Must not cause frequent route changes to ensure the stability of the network. The stability of path weights is determined by the type of path characteristics that are captured by the routing metrics, which can be either load-sensitive or topology-dependent.
2. Must ensure that minimum weight paths can be found by efficient algorithms with polynomial complexity.

3. A metric must capture the characteristics of mesh networks to ensure that minimum weight paths have good performance and assign a weight to a route based on the traffic load on the route. Some examples are Interface Switching Cost and Number of Congested Nodes. Under load sensitive metrics, the weight of a route may change frequently as flows arrive and depart.
4. Must ensure that forwarding loops are not formed by routing protocols.

2.9.3 Performance and routing metrics

The goal of the routing protocols is to route packets through minimum weight paths in terms of certain routing metrics. To ensure that the resources of mesh networks are utilized efficiently, the minimum weight paths selected by these routing protocols must have good performance in terms of high throughput and low packet delay. The routing metrics must be able to capture the characteristics of mesh networks that impact the performance of paths. These characteristics are:

Path length. Each hop introduces extra delay and potentially more packet loss, a longer path usually increases the end-to-end delay and reduces the throughput of a flow. So, a routing metric should increase the weight of a path when the path's length increases.

Link capacity. Unlike a wired link, whose capacity is independent of the physical distance between the link's end points, the maximum transmission rate between two neighboring wireless nodes (i.e., the link capacity between the two nodes) is directly related to the physical distance between the two nodes. In general, as the distance between two nodes increases, the channel quality degrades. So, although the effect of path length seems to favor paths with smaller hop count, the relationship between distance and link capacity counteracts this effect by favoring paths with larger hop count but higher link capacities. Therefore, when designing routing metrics, a trade-off must be found between these two trends.

Packet loss ratios. Different wireless links may have different packet loss ratios. A node may need to retransmit a packet multiple times on a link with a high packet loss ratio, which affects both the throughput and the delay of any flow that goes through the link. Hence, a routing metric must capture the packet loss ratios to ensure good performance for the minimum weight path.

Interference. To find minimum weight paths with good performance, routing metrics must capture both intraflow and interflow interference. It is not easy to capture interference using routing metrics since both the channel used by a link (the channel assignment of the link) and the capacity of the link are related to the amount of intra-flow and inter-flow interference that the link may impose on its neighborhood. In terms of the impact of link capacity, a packet that is transmitted over a 1Mbps link consumes more channel time at its neighbors than if it is transmitted over a 10Mbps link, hence resulting in more intra-flow and inter-flow interference.

The channel assignment may impact the interference level since neighboring nodes may use different channels or radio technologies so that they do not interfere with each other. The IEEE 802.11b/g standards and the IEEE 802.11a standard provide 3 and 12 non-overlapped frequency channels respectively and the IEEE 802.11b/g and the IEEE 802.11a operate on different frequency bands (2.4 GHz and 5 GHz, respectively). Hence, both the diversity of channel assignments and the link capacity need to be captured when the routing metrics considers the interference of a path.

2.9.4 Efficient algorithms to calculate minimum weight paths

The routing protocols rely on certain forms of efficient algorithms, such as the Bellman-Ford or Dijkstra's algorithms, to compute the minimum weight paths. Even if a routing metric ensures that its minimum weight paths have good performance, there is no guarantee that a routing protocol can have good performance if there does not exist an efficient algorithm to calculate the minimum weight paths based on the routing metric. According to [13] the necessary and sufficient condition for the existence of such efficient algorithms is that the routing metrics must have a property called **isotonicity**. If a routing metric is not isotonic, only algorithms with exponential complexity can calculate minimum weight paths based on this routing metric which are only tractable on networks of moderate size.

2.9.4.1 Isotonicity

The isotonic property means that a metric should ensure that the order of the weights of two paths is preserved if they are appended or prefixed by a common third path. Assume that for any path *a* its weight is defined by a routing metric, which is a function of *a*, denoted as *W(a).* Denoting the concatenation of two paths *a* and *b,* by *a+b*, the definition of isotonicity is:

Definition 1: A routing metric *W(.)* is isotonic if $W(a) \leq W(b)$ implies both $W(a+c) \leq W(b+c)$ *and* $W(c'+a) \leq W(c'+b)$, for all a; b; c; c'. See Figure 2.12.

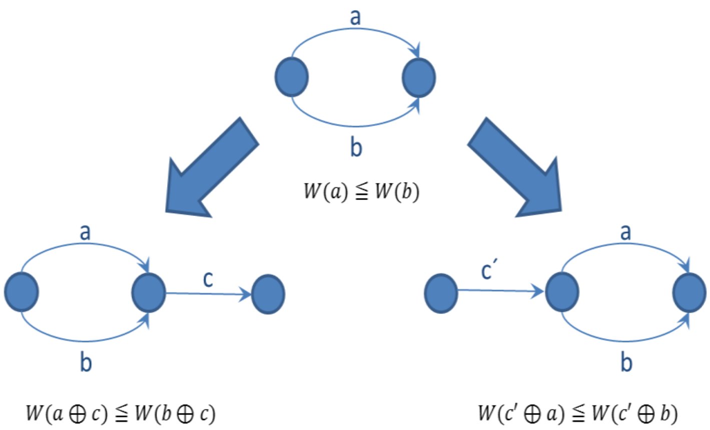


Fig. 2. 12. Example of Isotonicy.

In the work of Sobrinho [19] [20] [13] there exists a relation between the isotonicity property and the optimality of the Bellman-Ford and Dijkstra's algorithms

Theorem 1: Isotonicity is a sufficient and necessary condition for both the Bellman-Ford and Dijkstra's algorithm to find minimum weight paths.

Theorem 1 implies that if a routing metric is not isotonic, routing protocols based on the Bellman-Ford or Dijkstra's algorithm may not find the minimum weight path between two nodes. The resulting sub-optimal paths may degrade network performance. Therefore, routing metrics must be either isotonic or be able to transfer to some isotonic forms to ensure good network performance.

2.9.4.2 Loop-free routing

According to [13], to determine whether minimum weight paths can be calculated efficiently, it is also needed to ensure loop-free routing. As shown by Sobrinho's work [19], a metric must be isotonic to ensure that no routing loops can be formed when hop-by-hop routing is combined with Dijkstra's algorithm:

Theorem 2: If Dijkstra's algorithm is used in hop-by-hop routing, isotonicity is a sufficient and necessary condition for loop-free forwarding.

2.9.5 Routing metrics

In this section, we discuss briefly the routing metrics proposed for wireless mesh networks. These metrics are: hop count, ETX [23], [24], ETT [25], WCETT [25] and MIC [26]. All routing metrics are topology-dependent and each routing metric was proposed as an improvement over the previous one.

2.9.5.1 Hop count

Hop count is the most commonly used routing metric in existing routing protocols such as DSR [11], AODV [12], DSDV [21] and GSR [22]. It reflects the effects of path lengths on the performance of flows.

Advantages: Hop count metric is isotonic, so, efficient algorithms can find loop-free paths with minimum hop count.

Drawbacks: Hop count does not consider the differences of the transmission rates and packet loss ratios between different wireless links, or the interference in the network. Hence, using a hop count metric may not result in good performance.

2.9.5.2 Expected Transmission Count (ETX)

ETX is defined as the estimated number of MAC transmissions (including retransmissions) required for successfully delivering a packet through a wireless link. The weight of a path is defined as the summation of the ETX's of all links along the path. Since both long paths and lossy paths have large weights under ETX, the ETX metric captures the effects of both packet loss ratios and path length.

Mathematical description: ETX value can be defined as follows

$$ETX = \frac{1}{d_f x d_r} \qquad \text{(Eq 2.2)}$$

d_f : link forward delivery (probability that data packet successfully arrives at receiver)
d_r: backward delivery ratio (probability that ACK is received by sender)
$d_f \times d_r$: shows that a packet is transmitted with success in forward direction and ACK is also successfully received in backward direction.

The total ETX of a path is the summation of the ETX of all links on the path.
Advantages: ETX is an isotonic routing metric, which guarantees easy calculation of minimum weight paths and loop-free routing under all routing protocols.

Drawbacks: ETX does not consider interference or the fact that different links may have different transmission rates.

2.9.5.3 Expected Transmission Time (ETT)

The ETT routing metric improves ETX by considering bandwidth also while assigning metric to a link. The ETT of a link is defined as the expected MAC layer duration for a successful transmission of a packet at link. The weight of a path p is simply the summation of the ETT's of the links on the path. The relationship between the ETT of a link and ETX can be expressed as:

Mathematical description: ETT value can be defined as follows:

$$ETT = ETX \times \frac{S}{B}, \qquad \text{(Eq 2.3)}$$

where B is the bandwidth of link and S is the packet size.

Advantages: The ETT metric captures the impact of link capacity on the performance of the path. Similar to ETX, ETT is also isotonic.

Drawbacks: The remaining drawback of ETT is that it still does not fully capture the intraflow and interflow interferences in the network. For example, ETT may choose a path that only uses one channel, even though a path with more diversified channels has less intra-flow interference and hence higher throughput.

2.9.5.4 Weighted Cumulative ETT (WCETT)

WCETT improves ETT by considering the channel diversity along the path. The reduction of the intraflow interference is done by reducing the number of nodes on the path of a flow that transmits on the same channel.

Mathematical description: As different links on a path might have different channels assigned to it, it is important to capture the effect of the sum of the transmission times of links on every channel. Let X_j be the sum of the transmission times of links on channel j as follows:

$$X_j = \sum_{link\ i\ is\ on\ channel\ j} ETT_i \qquad 1 \le j \le k \qquad \text{(Eq 2.4)}$$

WCETT for a path p is defined as:

$$WCETT(p) = (1-\beta)\sum_{i=1}^{n} ETT_i + \beta max_{1\le j\le k} X_j \qquad \text{(Eq 2.5)}$$

β is a tunable parameter subject to $0 \le \beta \le 1$.
X_j is the number of times the channel j is used along the path p and captures the intra-flow interference.
$max_{1\le j\le k}$ X_j component counts the maximum number of times that the same channel appears along a path. It captures the intra-flow interference of a path since it essentially gives low weights to paths that have more diversified channel assignments on their links and hence lower intra-flow interference.

Advantages: WCETT finds routing paths with least ETT values and highest channel diversity.

Drawbacks: WCETT is proven to be non-isotonic. Due to WCETT's lack of isotonicity, there is no efficient algorithm with polynomial complexity to calculate minimum weight paths. In addition, the non-isotonicity of WCETT makes it unusable for link-state routing. To ensure loop-free routing, WCETT can only be used in on-demand routing, source routing such as LQSR or distance-vector routing. Moreover, WCETT does not explicitly consider the effects of inter-flow interference, although it does capture intraflow interference. Therefore, WCETT may route flows to dense areas where congestion is more likely and may even result in starvation of some nodes due to congestion.

2.9.5.5 Metric of Interference and Channel-switching (MIC)

MIC metric [10], improves WCETT by considering inter-flow and intra-flow interference using IRU (Interference aware Resource Usage) and CSC (Channel Switching cost) components of links. IRU of a link *ij* on channel *c* also includes its ETT and can be defined as below**:**

$$IRU_{ij}(c) = ETT_{ij}(c)\ X\ \left|N_i(c) \cup N_j(c)\right| \qquad \text{(Eq 2.6)}$$

$|N_i(c) \cup N_j(c)|$ is the number of neighboring nodes interfered due to activity of a link *ij* on channel *c*. To consider intraflow interference, every node on the routing path has assigned a CSC value. CSC (*Channel Switching Cost*) is defined as:

$$CSC_i = \begin{cases} w_1 & if\ CH(prev(i)) \neq CH(i) \\ w_2 & if\ CH(prev(i)) = CH(i) \end{cases} \ldots\ldots\ldots 0 \leq w_1 < w_2 \qquad \text{(Eq 2.7)}$$

CSC of a node *x* is low if the previous link where *x* was the receiver and the next link where *x* is the sender are on different channels. CSC value is high if both incoming and outgoing links are on the same channel as it introduces more intra-flow interference. MIC of a routing path *p* can be expressed as below solving its problems of non-isotonicity and the inability to capture inter-flow interference. The MIC metric of a path p is defined as follows:

$$MIC(p) = \frac{1}{N \times \min(ETT)} \sum_{link\ ij \in p} IRU_{ij} + \sum_{node\ i \in p} CSC_i, \qquad \text{(Eq 2.8)}$$

where $\frac{1}{N \times \min(ETT)}$ tries to balance the load in the network.

Essentially, the physical meaning of the IRU component is the aggregated channel time consumed by neighboring nodes that transmitting on the link. This metric captures the interflow interference and favors a path that consumes less channel time at its neighboring nodes. The CSC part of MIC represents the intra-flow interference; the CSC weight for paths with consecutive links using the same channel is higher than the weight for paths that alternate their channel assignments. The CSC favors paths with more diversified channel assignments.

3 Project design

The main objective of this project is to design and implement a global embedding algorithm for virtual network requests over a heterogeneous integrated wireless mesh network and wired network for a video on demand service.

We propose to integrate Software Defined Networking (SDN) and network functions virtualization (NFV) principles in heterogeneous networks. These technologies can enable resource and service orchestration with dynamic provisioning. SDN separates the network's control and forwarding layers and provides a centralized view of the distributed network. NFV focuses on optimizing the network services building virtualized network functions.

This proposal is based on virtualization techniques. The concept of network virtualization allows the setup of separate virtual networks (VN) running simultaneously on a shared substrate network. Network virtualization is achieved via logical segmentation of the underlying physical network. A virtual network is a group of virtual nodes interconnected via dedicated Virtual Links. The objective of virtualization is to make each VN appear to a customer as a dedicated physical network with dedicated resources. VNs can simultaneously support multiple network architectures (wireless mesh networks, wired networks, etc.).

This project is specifically focused on the algorithm of the SDN controller that dynamically provision and manage virtual resources. The controller's orchestrator function interworks with the NFV server to manage network resources for the coordinated control of the end-to-end infrastructure. We propose to balance traffic among the gateways of a Wireless Mesh Network, implementing the gateway balancing logic in the controller.

To test this proposal we have programmed in java a framework that allows us to simulate our embedding algorithm, its performance is evaluated through extensive simulations under several scenarios. The results are discussed and presented in excel graphics.

3.1 Scenario

Our scenario is based on a video on demand service located in a wired network (Internet), where requests are received and served at a specific time. Clients are located in a wireless mesh network in a neighboring community. They are not directly connected to the wired network so if users want to access to a video server, they have to connect to the nearest wireless mesh router using a wireless or wired link. In the most common form of WMNs, every router performs relaying of data for other mesh routers until they connect to a wireless mesh router, which has the additional capability of being Internet gateways. Such gateways routers carry the traffic between the wireless mesh networks and the Internet. We can visualize this, as the integration of two planes where the access plane provides connectivity to the clients while the forwarding plane relays traffic between the wireless mesh routers and Internet routers.

In this project, we use multi-radio mesh routers with two radios for backbone communication and one radio for user communication. The mesh backbone communication can be established using long-range communication techniques including directional antennas.

Provisioning is performed by a centralized Controller that receives virtual network requests from the clients, through the Gateways. The controller's orchestrator function interworks with the NFV server to manage network resources for the coordinated control of the end-to-end infrastructure. The controller is responsible of provisioning the VNs through matching, embedding and allocating available substrate resources to set up the required VNs.

An example of this architecture is a neighborhood community network that can be built using infrastructure meshing. The mesh routers are placed on the roof of houses, which serve as access points for users inside their homes and along the roads.

3.2 Assumptions

The following assumptions are made for the project design and implementation:

Wireless Network

WMNs clients want to connect to a service that is provided in a wired network. The clients are often laptops, cell phones and other wireless devices equipped with wireless network interface cards (NICs) that can connect directly to wireless mesh routers. Customers without wireless NICs can access WMNs by connecting by cable to wireless mesh routers access points. Mesh clients can be either stationary or mobile, but once they access to an access point, they are stationary for the whole transmission.

The VN requests of the clients are collected by the Gateways, which in turn collects network management information from the network devices and forwards the data to the Network Manager.

The controller collects all the requests at the same time, and schedules them in FIFO order.

Each mesh router could operate as a router and also as a host forwarding packets on behalf of other nodes that may not be within direct wireless transmission range of their destinations. In WMN, mesh routers establish a multi-hop backbone through wireless links.

A wireless mesh node is equipped with three radios; two radios are exclusively used as backhaul and download of information, and one radio for client access and management information.

These radios are configured to different static channels in order to prevent interferences. Several existing algorithms [6] can be used to pre-configure the channels for the radios and the discussion of channel assignment algorithms is beyond the scope of this project.

Mesh routers have no limitations regarding energy consumption. Also, the pattern of traffic between these routers is assumed to be fairly stable over time.

To enable the integration of WMNs with various existing wireless networks or fixed networks, some mesh routers includes the gateway/bridge functionalities. A mesh router that is connected to the wired Internet is referred to as an Internet gateway (IGW). Multiple IGWs can be deployed in a WMN to increase the network capacity.

In wireless mesh networks the information about the wireless network is gathered by the gateways.

Wired Network

The SDN controller gathers all the information of the wired and wireless mesh network infrastructure. It receives the requests from the wireless mesh clients, looks for the best path (using efficiently the available resources) and provides wireless mesh clients with the service required in Internet, according to its requirements.

There is only one media server that provides video service to all the WMNs clients.

3.3 Challenges

The following key challenges are detected:

- Design and implementation of an efficient embedding algorithm that first checks the available resources of the network, then search for multiple candidates embeddings for a VN request, next, evaluate the candidates embeddings based on a comparison metric, and take the highest –quality embedding candidate.
- Implement load-balanced functionality in the algorithm.
- In WMNs, use a metric model that takes into account intraflow and interflow interferences in wireless mesh networks.
- In WMNs, solve the non-isotonic problem of the metric model, caused by the nature of the intraflow interference.
- In WMNs, use of multi-radio, multichannel infrastructure in order to minimize interference.
- In Wired and wireless networks, select and efficient algorithm to find the shortest path.
- Provide a global solution that involve wireless wired network.

- Design of an embedding algorithm for integrated wired and wireless mesh networks.
- Software implementation of the framework, to test our algorithm performance.
- Graphic representation and data analysis.

3.4 Network Operation

The network operation follows the steps described below.

This general form of WMNs can be visualized as an integration of two planes where the access plane provides connectivity to the clients while the forwarding plane relays traffic between the mesh routers. This design has become more and more popular due to the increasing usage of multiple radios in mesh routers and virtual wireless interfacing techniques.

Discover phase: Wireless mesh routers send network information to the wireless mesh gateways, which in turn send the network information to the SDN controller located in the wired network.

Request phase: When a client, located in a wireless mesh network, wants to access to a service located into the wired network, it sends requests to the wireless mesh gateways and the wireless Mesh gateways resends the request to the controller.

Resource selection: At a specific time the controller gathers the entire requests and by FIFO order**.** The controller checks the availability of the service and look for the best path to interconnect wireless mesh clients with the video server located in the wired network.

Binding of configuration: The controller configures all the equipment involved in the transmission (wired routers, wireless mesh routers, and gateways).

Service provision: According to the path selected and configured, video server packets are forwarded through the selected wired routers, wireless mesh gateway and wireless Mesh Routers to the wireless client.

3.5 Wireless VN Embedding algorithm solution

In this section we model the substrate network, virtual network and VN assignment. Next it is described the embedding algorithm that we use in this project, and finally, the solution provided to convert a non-isotonic metric into isotonic is discussed.

3.5.1 Model Formulation

Network model: The real network with virtual resources is modeled in the following way:

SN: substrate network denoted by an undirected graph

$$G^S = \left(N^S, L^S, A_N^S, A_{L,}^S, I\right) \qquad \text{(Eq. 3.1)}$$

N^S= set of nodes

L^S = Set of links

A_N^S = Set of nodes resources (CPU)

$A_{L,}^S$ = Set of link resources (we model the link capacity to be fixed, time average value, over time-varying link channels, reasonable because time-scale of embedding arrival and departure much slower than channel variations.)

$CPU^S(n)$= amount of CPU resource of node $n \in N^S$

$CAP^S(l)$= capacity of wireless link $l \in L^S$

P^S= set of all paths in G^S

Virtual Network model:

The Virtual Network is modeled as follows:

VN: virtual network denoted by an undirected graph

$$G^V = (N^V, L^V) \qquad \text{(Eq. 3.2)}$$

N^V= set of nodes

$L^V = set\ of\ links$

Requirements:

C_N^V = Set of nodes requirements (CPU)

C_L^V = Set of link requirements

$CPU^V(n)$= amount of CPU requirement for the virtual node $n \in N^V$

$BW^V(l)$= capacity requirement for the virtual wireless link $l \in L^V$

These requirements could be interpreted as constraints for the embedding.

D^V= The request should be served for this duration

L^N= set of links connected to the node n in both G^V and G^S

Virtual Network Embedding:

We define a mapping from G^V to a subset of G^S that complies with the requirements:

$$E: G^V \rightarrow (N^*, P^*) \qquad\qquad N^* \in N^S P^* \in P^S$$

3.5.2 Global Embedding Algorithm

In this project, we study the embedding of Virtual Network Requests over an integrated wired and wireless mesh network. To perform this, we have designed a virtual access network embedding algorithm that can ensure the coexistence of different requests on the same substrate and provide the most suitable network resources according to the requests requirements, networks constraints (CPU, BW) and network load-balance. The main steps of the algorithm are described below.

Previously to run the algorithm, we generate randomly wireless and wired networks and the reception of a set of VN requests.

Step0. For each request, test if there are enough resources in the wireless network.

Step1. Isotonic conversion. Decomposition of the wireless real network, into a virtual one, to convert non-isotonic metric into isotonic metric.

Step2. For each request, use of the adapted Dijkstra's algorithm to look for the minimum weight path to each gateway. It is used intraflow and interflow interference as weight.

Step3. Selection of the most suitable path based on the weights comparison.

Step4. Control of the available resources in the wired network.

Step5. Taking as starting point the Wireless Mesh Gateway of the selected path, use of Dijkstra's algorithm to find the minimum weight path to the video server.

Step6. For each request, provision of a global integrated wireless and wired path, to access to the video server.

3.5.3 Algorithm for finding minimum weight paths

In this project we use Dijkstra's algorithm that finds the shortest path from the starting point to all other nodes in the network. As metrics, we use different types, namely: **Concave metrics**: BW,CPU and the **Non-isotonic additive metric,** composed by interflow and intraflow interference, used in the wireless network to model network interferences.

In our embedding algorithm first we search the path that agrees with the requirements of BW and CPU, and for the selected paths we apply an algorithm to find an optimized path. In the wireless network this path is the one that produces less interference, in wired network the path with less weight.

3.5.4 Solution to use Non-Isotonic Metrics in Dijkstra's Algorithm

Due to the fact that our wireless metric model is based on an additive non-isotonic metric, and we want to use Dijkstra's algorithm, which is not compatible with non-isotonic metrics, it is necessary to use a mechanism to convert our non-isotonic metric into an isotonic one.

The decomposition of our metric model is based on a deep understanding of why our metric model is not isotonic. Fig. 3. 1. demonstrates that the non-isotonic behavior of our metric model is due to the fact that the additional weight that link (Y,Z,1) brings to a path not only depends on link (Y,Z,1)'s own status, but is also related to the channel assignment of the link that precedes link (Y,Z,1). Due to the common channel used by links (X,Y,1) and (Y,Z,1), adding link (Y,Z,1) to the path (X,Y,1) introduces a higher cost than adding link (Y,Z,1) to path (X,Y,2). Hence, even though MIC((X,Y,1)) < MIC((X,Y,2)), MIC((X,Y,1) $\oplus$ (Y,Z,1))>MIC((X,Y,2) $\oplus$ (Y,Z,1)).

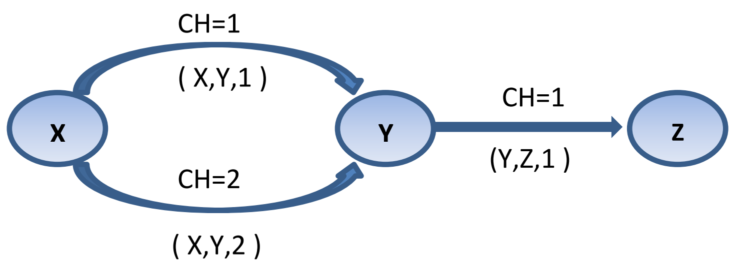


Fig. 3. 1. Non-Isotonicy of the Wireless Metric.

To solve this problem, we use a solution based on [6]. We map a real network into a virtual network, and decompose the metric into isotonic link weight assignments in the virtual network. The goal of this decomposition is to ensure that the algorithm that we use, Dijkstra, could find efficient minimum weight paths without creating forwarding loops.

The decomposition of the metric is based on the fact that the non-isotonic behavior of the metric is caused by the different increments of path weights due to the addition of a link on a path. Whether a cost increment will be different by adding a link is only related to the channel assignment of the previous link on the path. This is due to the fact that the possible channels assignments for the precedent link are limited by the number of radios that a node has, and since each radio is only configured to one channel. Then, by introducing several virtual nodes to represent these possible channel assignments for the precedent link, we can translate our wireless metric model into isotonic weight assignments to the links between these virtual nodes.

More specifically, for every channel *c* assigned to a node, a radio is configured into the node and two virtual nodes $Xi(c)$ and $Xe(c)$ are introduced. $Xi(c)$ represents that the *previous* node (X) that transmits to node X on channel *c*. $Xe(c)$ indicates that node X transmits to its next hop on channel *c*. The subscript *i* stands for "ingress" and the subscript *e* stands for "egress". Links between the virtual nodes belonging to different real nodes are used to capture the interflow weight. Intraflow weight is captured by the channel switching cost assigned to the virtual links, within a virtual Node.

For example, we map the original real paths in Fig. 3. 2 to the virtual network in Fig. 3.3**.** We assume that w2 is the weight of not changing the channel, and w1 is the weight due to the use of the same channel. The real path $(X,Y,1) \oplus (Y,Z,1)$, which has the metric weight $interflow_{XY}(1)$ +interflow$_{YZ}(1) + w_2$, is mapped into a virtual path $X_e(1) \rightarrow Y_i(1) \rightarrow Y_e(1) \rightarrow Z_i(1)$, whose aggregated link weight is also $interflow_{XY}(1) + interflow_{YZ}(1) + w_2$. The real path $(X,Y,2) \oplus (Y,Z,1)$ with metric weight $interflow_{XY}(2) + interflow_{YZ}(1) + w_1$ is mapped to path $X_e(2) \rightarrow Y_i(2) \rightarrow Y_e(1) \rightarrow Z_i(1)$, whose aggregated link weight is also $interflow_{XY}(2) + interflow_{YZ}(1) + w_1$.

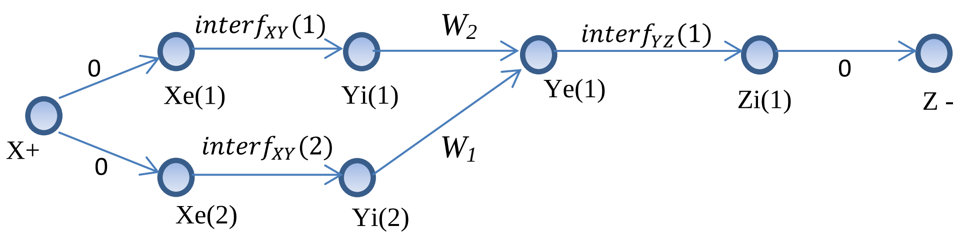


Fig. 3. 2. Metric Decomposition.

$$interflow = \propto IRU_{Source-Destination}(channel)$$

Below in Figure 3.3, we show a more complete example, in which all the wireless mesh nodes are additionally source and sink nodes, and are represented by + and −.We have three Virtual nodes X, Y and Z. There are added links from the ingress virtual nodes to the egress virtual nodes. The weights of these links are assigned to capture different CSC costs. Link $(X_i(c), X_e(c))$ represents that node X does not change channels while forwarding packets and hence weight w_2 is assigned to this link. Similarly, weight w_1 is assigned to link $(X_i(c), X_e(c1))$, where $c \neq c_1$, to represent the low cost of changing channels while forwarding packets. Links between the virtual nodes belonging to different real nodes are used to capture the interflow weight. If node X is able to communicate with its neighbor node Y through channel c, two links $(X_e(c), Y_i(c))$ and $(Y_e(c), X_i(c))$ are added for each common channel between nodes X and Y (e.g., links $(X_e(1), Y_i(1))$ and $(Y_e(1), Xi(1))$). The weights of these links are $InterflowIRU_{XY}(c)$ and $InterflowIRU_{YX}(c)$, respectively.

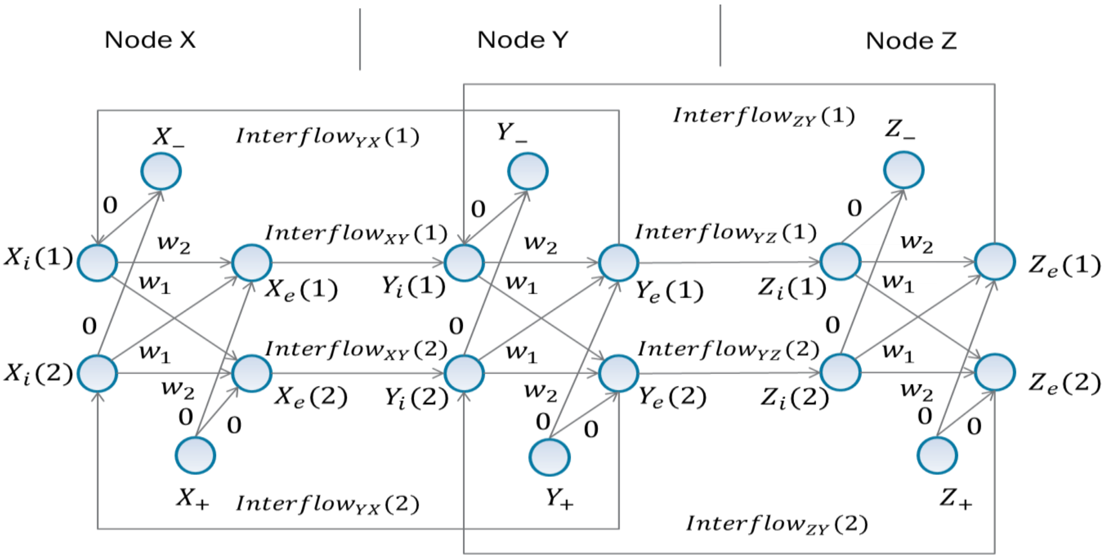


Fig. 3. 3. Decomposition into Virtual Nodes

By building the virtual network from a real network, we essentially decompose the metric of the real network into weight assignments to the links between virtual nodes. This is because the metric weight of a real path in a real network can be reconstructed by aggregating all of the weights of the virtual links on the corresponding virtual path. This model could be applied to the metric MIC due to its additive characteristic, but it cannot be applied to WCETT.

4 Simulator development

We have developed a simulator (attached to this project) of our framework to evaluate the proposed global embedding algorithm and analyze the impact of its key features.

4.1 Software features

We have used Java Netbeans to develop the global embedding algorithm simulator. The developed source code is attached to this project file. The simulator is composed of the following applications:

Random requests generator. This application randomly generates requests. The user determines the source and the sink and the number of requests. The CPU and BW requirements are randomly generated within a predefined range.

Random networks generator. This application randomly generates wireless and wired networks. The user determines: number of wireless nodes and links, number of wired nodes and links and wired network degree. The CPU of the nodes and the BW of the links are randomly generated within a predefined range. The intraflow and interflow values are also randomly generated within a predefined range.

Wireless network embedding algorithm. This is the first part of the global embedding algorithm, this application reads the network file, and creates a wireless network graph. Next it reads the requests file and check the feasibility of serving each request. Then, it decompose the wireless real graph into a virtual graph that allows to convert our non-isotonic metric model into isotonic one. Next, it applies Dijkstra's algorithm, and find the minimum weight path to each available gateway. Between the candidates gateways, it is selected the one with best metric in terms of interference. Finally, it subtracts the resources used by the selected path.

Wired network embedding algorithm. This is the second part of the global embedding algorithm, and it is executed only in case that there is a feasible wireless path. This application reads the network file, and creates a wired network graph. Next reads the requests file and check the feasibility of serving each request. Then taking the gateway selected as the starting point, it applies Dijkstra's algorithm, and find the minimum weight path to destination (video server). Finally, subtracts the resources used by the selected path.

Results database. For each request the following information is collected: time required, number of hops, total weight, wireless weight, wired weight, and path. All the data obtained is collected, processed and analyzed with an EXCEL application.

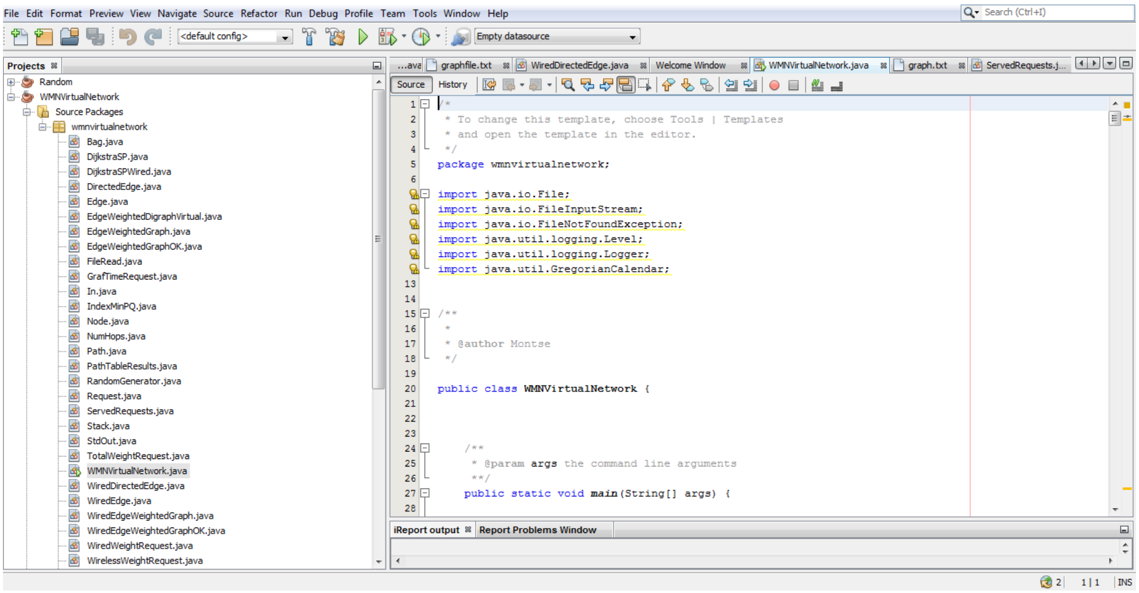


Fig. 4. 1 View of the Java Netbeans with the developed software

4.2 Software Test

To evaluate the correct operation of the implemented software, first we have selected a scenario with a small wireless mesh network to make the analysis of the results much easier. In the following subsections we will evaluate more complex scenarios.

In this case, we have tested the following main steps, being the most critical part of the implementation the wireless mesh network.

1. Configuration of the framework parameters.
2. Generation of the random Requests file.
3. Generation of the random wireless and wireless network graphfile.
4. Wireless Network: generation of the wireless graph, decomposition into Virtual Nodes to convert the non-isotonic metric, run Dijkstra's algorithm, selection of the best wireless Mesh Gateway.
5. Wired Network: generation of the wired graph. Run Dijkstra's algorithm.
6. Results Collection: Time Elapsed, number of hops, weight, detailed path file.

4.2.1 Network parameters

The application is configured with the following parameters:

Wireless Network	
Number of Nodes	6
Number of Links	8
Degree	2

Number of GW	2
CPU	10-20
BW	4

Fig. 4. 2 Wireless Network Parameters

Wired Network	
Number of Nodes	6
Number of Links	9
CPU	20-50
BW	10

Fig. 4. 3 Wired Network Parameters

4.2.2 Requests file

With the objective to verify the correct operation of the network embedding algorithm, we use a requests file in which, all the requests have the same source and destination.

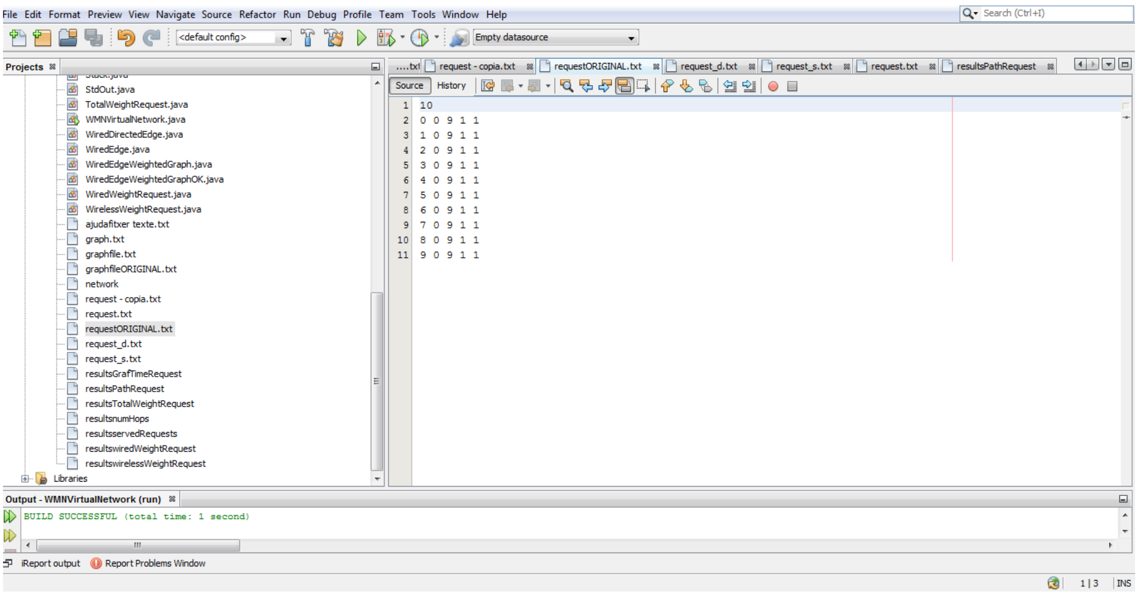


Fig. 4. 4 Requests File

4.2.3 Wireless and Wired Network Graphfile

The Random generator randomly creates a text file that contains the data of the wireless and wired network. The user can configure, in the generator, the following parameters: number of wireless nodes and links, number of wired nodes and links and wired network degree. The CPU of the nodes and the BW of the links are randomly generated within a predefined range. The channel, intraflow and interflow values are also randomly generated within a predefined range. Below there is a capture of the graphfile:

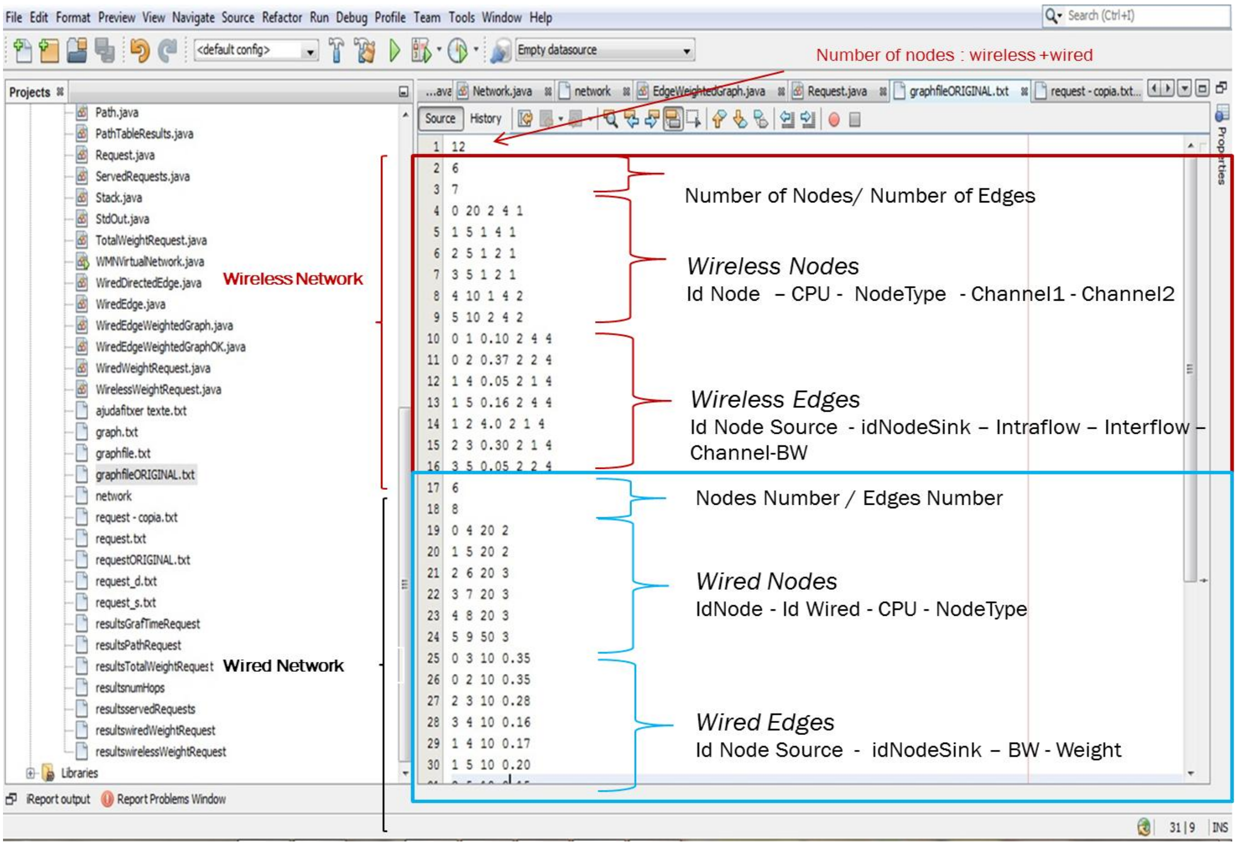


Fig. 4. 5 Graphfile

The wireless mesh network topology of the graphfile is the following one:

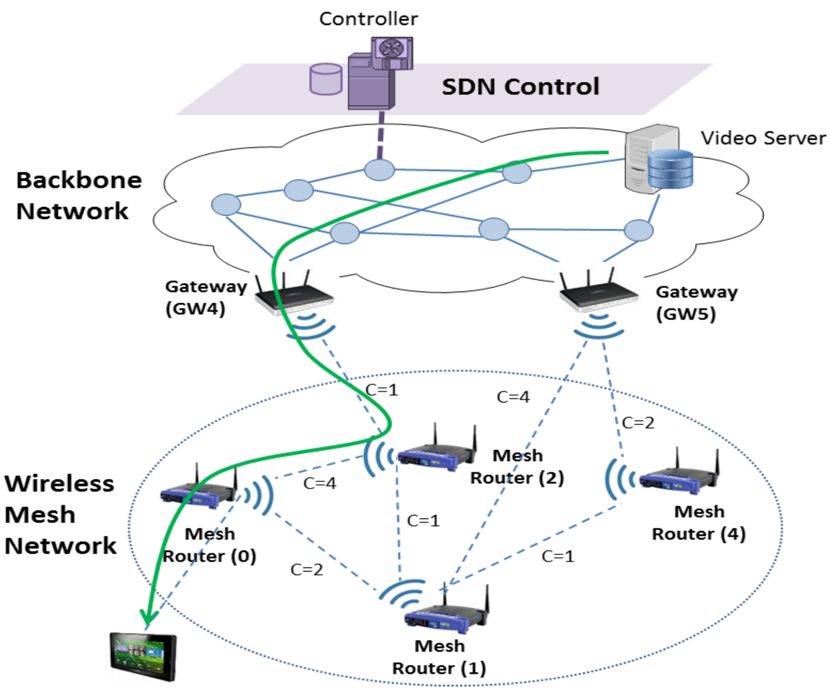


Fig. 4. 6 Wireless Mesh Network Topology connected to the SDN-controlled backbone

4.2.4 Decomposition of the metric into Virtual Nodes.

Due to the incompatibility of our non-isotonic metric with Dijkstra's algorithm, we need to decompose the wireless real graph into a wireless virtual graph to convert our non-isotonic metric model into an isotonic one. The following graph represents the real network decomposition into the Virtual Network. Each node is provided with two radio interfaces, so each real node is decomposed in two egress virtual nodes, and two ingress virtual nodes, one for each channel. In addition if the node has sink and source functionalities, we have to add one virtual node for the sink, and another virtual node for the source node. The interflow interference is assigned to the virtual links between virtual nodes, and the intraflow interference is assigned to the virtual links located inside the virtual node. The following Fig. 4. 7 is a simplified representation of the converted network in order to promote a better understanding of the solution.

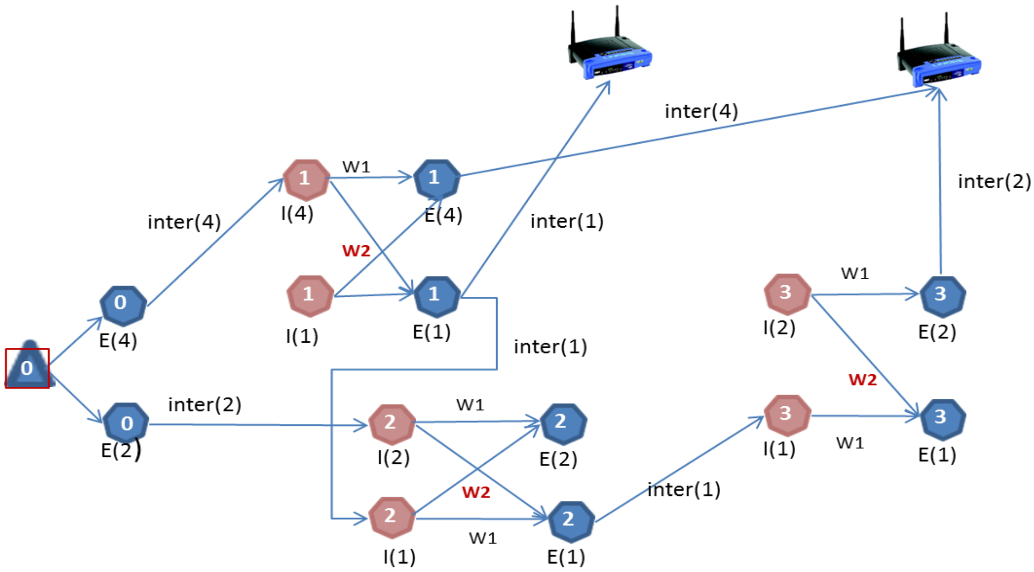


Fig. 4. 7 Example of Network Decomposition into VN

4.2.5 No Load Balancing Test

We first consider the test of the embedding algorithm configured with no load balancing functionality. In test 2 the embedding algorithm with load balancing option is configured.

To check the correct operation of the simulator we analyze the results file in detail. In this file we can observe the following:

Minimum weight path selection. We can observe that in the wireless network the weight increases as request are served. Wired network weight depends on the gateway selected, depending on the gateway location the wired weight could be increased or decreased but taking as starting point a concrete gateway. Dijkstra's algorithm will find the minimum weight path to destination. By the figure below, we can observe that the priority of the global embedding algorithm is to find the minimum weight wireless path, and the wired path is set depending on the wireless mesh Gateway used.

Resource management, involves feasibility control of the requests and resources subtraction after their assignation. Examining the file below we can observe that from request 0 to request 3 it is used the wireless path 0-1 1-4 until the BW resource is exhausted, and then from request 4 to request 7, it is used the wireless path 0-2 2-3 3-5, until it is also exhausted. At this point no more requests are served.

Intraflow penalization, a path that does not change its channel hop by hop, gets its weight substantially increased due to the intraflow interference.

```
REQUESTS RESULTS
************************************************
************************************************
REQUEST ID: 0
Request idOrigin: 0
Request idDestination: 9
Request CPU: 1
Request BW: 1
Request Served: 1
Time Elapsed: 140
Id GW Selected: 4
Path Weight: 0.65
WIRELESS
Wireless Weight: 0.15000000000000002
Wireless Path:
Edge: 0-1 interflow: 0,10000 intraflow: 0,00000 weight: 0,10000 channel: 4 BW: 3 BWinit: 4
Edge: 1-4 interflow: 0,05000 intraflow: 0,00000 weight: 0,05000 channel: 1 BW: 3 BWinit: 4
WIRED
Wired weight: 0.5
Wired Path:
 Edge: 4-6  weight:0,35000 BW:9 BWinit:10
 Edge: 6-9  weight:0,15000 BW:9 BWinit:10
************************************************
************************************************
REQUEST ID: 1
Request idOrigin: 0
Request idDestination: 9
Request CPU: 1
Request BW: 1
Request Served: 1
Time Elapsed: 126
Id GW Selected: 4
Path Weight: 0.65
WIRELESS
Wireless Weight: 0.15000000000000002
Wireless Path:
Edge: 0-1 interflow: 0,10000 intraflow: 0,00000 weight: 0,10000 channel: 4 BW: 2 BWinit: 4
Edge: 1-4 interflow: 0,05000 intraflow: 0,00000 weight: 0,05000 channel: 1 BW: 2 BWinit: 4
WIRED
```

Wired weight: 0.5
Wired Path:
Edge: 4-6 weight:0,35000 BW:8 BWinit:10
Edge: 6-9 weight:0,15000 BW:8 BWinit:10
****
****
REQUEST ID: 2
Request idOrigin: 0
Request idDestination: 9
Request CPU: 1
Request BW: 1
Request Served: 1
Time Elapsed: 112
Id GW Selected: 4
Path Weight: 0.65
WIRELESS
Wireless Weight: 0.15000000000000002
Wireless Path:
Edge: 0-1 interflow: 0,10000 intraflow: 0,00000 weight: 0,10000 channel: 4 BW: 1 BWinit: 4
Edge: 1-4 interflow: 0,05000 intraflow: 0,00000 weight: 0,05000 channel: 1 BW: 1 BWinit: 4
WIRED
Wired weight: 0.5
Wired Path:
Edge: 4-6 weight:0,35000 BW:7 BWinit:10
Edge: 6-9 weight:0,15000 BW:7 BWinit:10
****
****
REQUEST ID: 3
Request idOrigin: 0
Request idDestination: 9
Request CPU: 1
Request BW: 1
Request Served: 1
Time Elapsed: 76
Id GW Selected: 4
Path Weight: 0.65
WIRELESS
Wireless Weight: 0.15000000000000002
Wireless Path:
Edge: 0-1 interflow: 0,10000 intraflow: 0,00000 weight: 0,10000 channel: 4 BW: 0 BWinit: 4
Edge: 1-4 interflow: 0,05000 intraflow: 0,00000 weight: 0,05000 channel: 1 BW: 0 BWinit: 4
WIRED
Wired weight: 0.5
Wired Path:
Edge: 4-6 weight:0,35000 BW:6 BWinit:10
Edge: 6-9 weight:0,15000 BW:6 BWinit:10
****
****
REQUEST ID: 4
Request idOrigin: 0
Request idDestination: 9
Request CPU: 1
Request BW: 1
Request Served: 1
Time Elapsed: 76
Id GW Selected: 5
Path Weight: 0.9199999999999999

WIRELESS
Wireless Weight: 0.72
Wireless Path:
Edge: 0-2 interflow: 0,37000 intraflow: 0,00000 weight: 0,37000 channel: 2 BW: 3 BWinit: 4
Edge: 2-3 interflow: 0,30000 intraflow: 0,00000 weight: 0,30000 channel: 1 BW: 3 BWinit: 4
Edge: 3-5 interflow: 0,05000 intraflow: 0,00000 weight: 0,05000 channel: 2 BW: 3 BWinit: 4
WIRED
Wired weight: 0.2
Wired Path:
Edge: 5-9 weight:0,20000 BW:9 BWinit:10
****
****
REQUEST ID: 5
Request idOrigin: 0
Request idDestination: 9
Request CPU: 1
Request BW: 1
Request Served: 1
Time Elapsed: 76
Id GW Selected: 5
Path Weight: 0.9199999999999999
WIRELESS
Wireless Weight: 0.72
Wireless Path:
Edge: 0-2 interflow: 0,37000 intraflow: 0,00000 weight: 0,37000 channel: 2 BW: 2 BWinit: 4
Edge: 2-3 interflow: 0,30000 intraflow: 0,00000 weight: 0,30000 channel: 1 BW: 2 BWinit: 4
Edge: 3-5 interflow: 0,05000 intraflow: 0,00000 weight: 0,05000 channel: 2 BW: 2 BWinit: 4
WIRED
Wired weight: 0.2
Wired Path:
Edge: 5-9 weight:0,20000 BW:8 BWinit:10
****
****
REQUEST ID: 6
Request idOrigin: 0
Request idDestination: 9
Request CPU: 1
Request BW: 1
Request Served: 1
Time Elapsed: 82
Id GW Selected: 5
Path Weight: 0.9199999999999999
WIRELESS
Wireless Weight: 0.72
Wireless Path:
Edge: 0-2 interflow: 0,37000 intraflow: 0,00000 weight: 0,37000 channel: 2 BW: 1 BWinit: 4
Edge: 2-3 interflow: 0,30000 intraflow: 0,00000 weight: 0,30000 channel: 1 BW: 1 BWinit: 4
Edge: 3-5 interflow: 0,05000 intraflow: 0,00000 weight: 0,05000 channel: 2 BW: 1 BWinit: 4
WIRED
Wired weight: 0.2
Wired Path:
Edge: 5-9 weight:0,20000 BW:7 BWinit:10
****
****
REQUEST ID: 7
Request idOrigin: 0
Request idDestination: 9

Request CPU: 1
Request BW: 1
Request Served: 1
Time Elapsed: 98
Id GW Selected: 5
Path Weight: 0.9199999999999999
WIRELESS
Wireless Weight: 0.72
Wireless Path:
Edge: 0-2 interflow: 0,37000 intraflow: 0,00000 weight: 0,37000 channel: 2 BW: 0 BWinit: 4
Edge: 2-3 interflow: 0,30000 intraflow: 0,00000 weight: 0,30000 channel: 1 BW: 0 BWinit: 4
Edge: 3-5 interflow: 0,05000 intraflow: 0,00000 weight: 0,05000 channel: 2 BW: 0 BWinit: 4
WIRED
Wired weight: 0.2
Wired Path:
Edge: 5-9 weight:0,20000 BW:6 BWinit:10
****
****
REQUEST ID: 8
Request idOrigin: 0
Request idDestination: 9
Request CPU: 1
Request BW: 1
Request Served: 0
Time Elapsed: 16
Id GW Selected: 0
Path Weight: 0.0
WIRELESS
Wireless Weight: 0.0
Wireless Path:
WIRED
Wired weight: 0.0
Wired Path:
****
****
REQUEST ID: 9
Request idOrigin: 0
Request idDestination: 9
Request CPU: 1
Request BW: 1
Request Served: 0
Time Elapsed: 18
Id GW Selected: 0
Path Weight: 0.0
WIRELESS
Wireless Weight: 0.0
Wireless Path:
WIRED
Wired weight: 0.0
Wired Path:

4.2.5.1 Graphics

For a better understanding of the results, we use EXCEL for the graph presentation of the data obtained. In Fig. 4. 8, we represent the time elapsed per request. We observe an initial peak that corresponds to the initial network parameters calculation, which is the highest one. As requests are served, the time elapsed to serve a request decreases. We assume that is due to the subtraction of available resources, which decreases the time for path calculation.

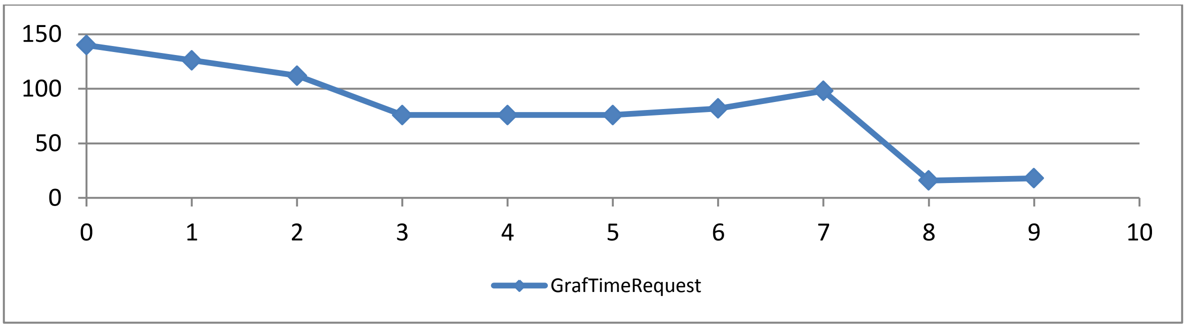


Fig. 4. 8. Time Elapsed Graphic

In Figure 4.9, we represent the weight per request. For a better understanding of the data represented in the graph, when a request is not served, the weight of this request is set to zero. A request is not served when there are no resources available. In this figure, we observe that the priority of the implemented algorithm is to find the wireless path with less weight and consequently the path with less interference, the wired path depends on the wireless path. Moreover, we can see that in the wireless network the weight increases as requests are served, as in the wired network the weight depends on the gateway selected, so its value could be increased or decreased depending on the gateway selected.

Fig. 4. 9. Weight per Request.

In Fig. 4. 10, we represent the number of hops per request. For a better graphic representation, when a request is not served the number of hops of this request is set to zero. A request is not served if there are no resources available. In the figure we observe the point at which the wireless path changes to another gateway. The path is the same from request 0 to 3, they all use GW4, and from request 4 to 7 it is used GW5, In this case, we can notice that for wired network it is better to use the GW5, but as our global embedding algorithm has the priority to minimize the wireless weight, the gateway is selected according to wireless optimal values.

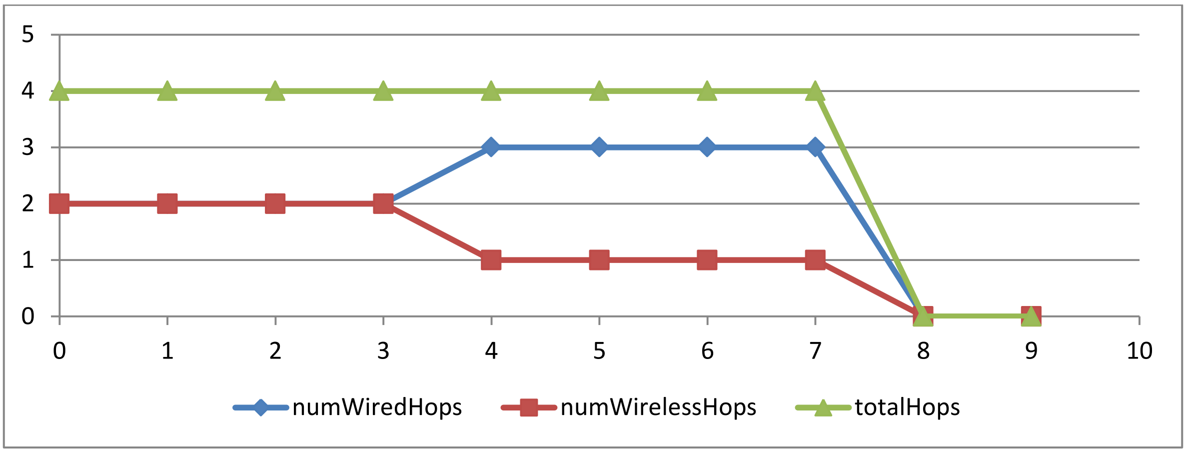


Fig. 4. 10. Number of Hops.

4.2.6 Load Balancing Test

Our Global embedding algorithm has the load balancing functionality. To evaluate the performance of the algorithm with this feature, it has been tested this option, with the same scenario and configuration parameters of the test 1, where the global embedding algorithm is used without the load balancing functionality.

In order to implement the load balancing functionality in the algorithm, each time that a link is used for a path, the interflow value of this link is increased. It is used the following mathematical formula to recalculate the interflow.

$$interflow = interflow + \frac{BWinit - BW}{BWinit} \qquad \text{(Eq 4.1)}$$

To check the correct operation of this load balancing functionality, we analyze the results file in detail. We observe that from request 0 to 1, it is used a path that we will denote as Path 1 (0-1 1-4); at this point this path is the one that has lower weight. Each time that a link is used for a path, its weight increases according to the Eq. 4.1. At request 2, is selected the path 2 (0-2 2-3 3-5). Then, across the following requests, the use of path 1 and path 2 is alternated; due to the load balancing that produces the increasing of the weight of the used link. At the end, the resources are exhausted and no more requests are served.

REQUESTS RESULTS FILE
****
****
REQUEST ID: 0
Request idOrigin: 0

```
Request idDestination: 9
Request CPU: 1
Request BW: 1
Request Served: 1
Time Elapsed: 172
Id GW Selected: 4
Path Weight: 0.65
WIRELESS
Wireless Weight: 0.15000000000000002
Wireless Path:
Edge: 0-1 interflow: 0,10000 intraflow: 0,00000 weight: 0,10000 channel: 4 BW: 3 BWinit: 4
Edge: 1-4 interflow: 0,05000 intraflow: 0,00000 weight: 0,05000 channel: 1 BW: 3 BWinit: 4
WIRED
Wired weight: 0.5
Wired Path:
 Edge: 4-6  weight:0,35000 BW:9 BWinit:10
 Edge: 6-9  weight:0,15000 BW:9 BWinit:10
**************************************************
**************************************************
REQUEST ID: 1
Request idOrigin: 0
Request idDestination: 9
Request CPU: 1
Request BW: 1
Request Served: 1
Time Elapsed: 103
Id GW Selected: 4
Path Weight: 1.15
WIRELESS
Wireless Weight: 0.6499999999999999
Wireless Path:
Edge: 0-1 interflow: 0,35000 intraflow: 0,00000 weight: 0,35000 channel: 4 BW: 2 BWinit: 4
Edge: 1-4 interflow: 0,30000 intraflow: 0,00000 weight: 0,30000 channel: 1 BW: 2 BWinit: 4
WIRED
Wired weight: 0.5
Wired Path:
 Edge: 4-6  weight:0,35000 BW:8 BWinit:10
 Edge: 6-9  weight:0,15000 BW:8 BWinit:10
**************************************************
**************************************************
REQUEST ID: 2
Request idOrigin: 0
Request idDestination: 9
Request CPU: 1
Request BW: 1
Request Served: 1
Time Elapsed: 138
Id GW Selected: 5
Path Weight: 0.9199999999999999
WIRELESS
Wireless Weight: 0.72
Wireless Path:
Edge: 0-2 interflow: 0,37000 intraflow: 0,00000 weight: 0,37000 channel: 2 BW: 3 BWinit: 4
Edge: 2-3 interflow: 0,30000 intraflow: 0,00000 weight: 0,30000 channel: 1 BW: 3 BWinit: 4
Edge: 3-5 interflow: 0,05000 intraflow: 0,00000 weight: 0,05000 channel: 2 BW: 3 BWinit: 4
WIRED
Wired weight: 0.2
```

Wired Path:
Edge: 5-9 weight:0,20000 BW:9 BWinit:10
****
****
REQUEST ID: 3
Request idOrigin: 0
Request idDestination: 9
Request CPU: 1
Request BW: 1
Request Served: 1
Time Elapsed: 122
Id GW Selected: 5
Path Weight: 1.67
WIRELESS
Wireless Weight: 1.47
Wireless Path:
Edge: 0-2 interflow: 0,62000 intraflow: 0,00000 weight: 0,62000 channel: 2 BW: 2 BWinit: 4
Edge: 2-3 interflow: 0,55000 intraflow: 0,00000 weight: 0,55000 channel: 1 BW: 2 BWinit: 4
Edge: 3-5 interflow: 0,30000 intraflow: 0,00000 weight: 0,30000 channel: 2 BW: 2 BWinit: 4
WIRED
Wired weight: 0.2
Wired Path:
Edge: 5-9 weight:0,20000 BW:8 BWinit:10
****
****
REQUEST ID: 4
Request idOrigin: 0
Request idDestination: 9
Request CPU: 1
Request BW: 1
Request Served: 1
Time Elapsed: 88
Id GW Selected: 4
Path Weight: 2.15
WIRELESS
Wireless Weight: 1.65
Wireless Path:
Edge: 0-1 interflow: 0,85000 intraflow: 0,00000 weight: 0,85000 channel: 4 BW: 1 BWinit: 4
Edge: 1-4 interflow: 0,80000 intraflow: 0,00000 weight: 0,80000 channel: 1 BW: 1 BWinit: 4
WIRED
Wired weight: 0.5
Wired Path:
Edge: 4-6 weight:0,35000 BW:7 BWinit:10
Edge: 6-9 weight:0,15000 BW:7 BWinit:10
****
****
REQUEST ID: 5
Request idOrigin: 0
Request idDestination: 9
Request CPU: 1
Request BW: 1
Request Served: 1
Time Elapsed: 90
Id GW Selected: 5
Path Weight: 3.17
WIRELESS
Wireless Weight: 2.9699999999999998

Wireless Path:
Edge: 0-2 interflow: 1,12000 intraflow: 0,00000 weight: 1,12000 channel: 2 BW: 1 BWinit: 4
Edge: 2-3 interflow: 1,05000 intraflow: 0,00000 weight: 1,05000 channel: 1 BW: 1 BWinit: 4
Edge: 3-5 interflow: 0,80000 intraflow: 0,00000 weight: 0,80000 channel: 2 BW: 1 BWinit: 4
WIRED
Wired weight: 0.2
Wired Path:
Edge: 5-9 weight:0,20000 BW:7 BWinit:10
****
****
REQUEST ID: 6
Request idOrigin: 0
Request idDestination: 9
Request CPU: 1
Request BW: 1
Request Served: 1
Time Elapsed: 64
Id GW Selected: 4
Path Weight: 3.6500000000000004
WIRELESS
Wireless Weight: 3.1500000000000004
Wireless Path:
Edge: 0-1 interflow: 1,60000 intraflow: 0,00000 weight: 1,60000 channel: 4 BW: 0 BWinit: 4
Edge: 1-4 interflow: 1,55000 intraflow: 0,00000 weight: 1,55000 channel: 1 BW: 0 BWinit: 4
WIRED
Wired weight: 0.5
Wired Path:
Edge: 4-6 weight:0,35000 BW:6 BWinit:10
Edge: 6-9 weight:0,15000 BW:6 BWinit:10
****
****
REQUEST ID: 7
Request idOrigin: 0
Request idDestination: 9
Request CPU: 1
Request BW: 1
Request Served: 1
Time Elapsed: 131
Id GW Selected: 5
Path Weight: 5.42
WIRELESS
Wireless Weight: 5.22
Wireless Path:
Edge: 0-2 interflow: 1,87000 intraflow: 0,00000 weight: 1,87000 channel: 2 BW: 0 BWinit: 4
Edge: 2-3 interflow: 1,80000 intraflow: 0,00000 weight: 1,80000 channel: 1 BW: 0 BWinit: 4
Edge: 3-5 interflow: 1,55000 intraflow: 0,00000 weight: 1,55000 channel: 2 BW: 0 BWinit: 4
WIRED
Wired weight: 0.2
Wired Path:
Edge: 5-9 weight:0,20000 BW:6 BWinit:10
****
****
REQUEST ID: 8
Request idOrigin: 0
Request idDestination: 9
Request CPU: 1
Request BW: 1

Request Served: 0
Time Elapsed: 12
Id GW Selected: 0
Path Weight: 0.0
WIRELESS
Wireless Weight: 0.0
Wireless Path:
WIRED
Wired weight: 0.0
Wired Path:

****
****
REQUEST ID: 9
Request idOrigin: 0
Request idDestination: 9
Request CPU: 1
Request BW: 1
Request Served: 0
Time Elapsed: 19
Id GW Selected: 0
Path Weight: 0.0
WIRELESS
Wireless Weight: 0.0
Wireless Path:
WIRED
Wired weight: 0.0
Wired Path:

4.2.6.1 Graphics

As in the test 1 in the Fig. 4. 11, we have represented the time elapsed per request. We see an initial peak, which corresponds to the initial network calculation. Due to this fact, this value is the highest one. As requests are served, time elapsed between requests decreases; we assume that is due to the subtraction of available resources of the network, this fact facilitates the calculation of the path.

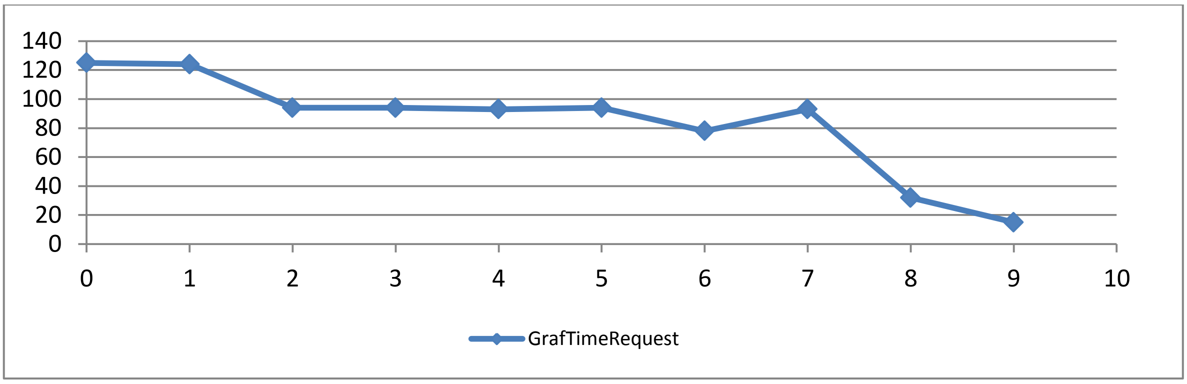


Fig. 4. 11. Time per Request

In the Fig. 4. 12, we represent the weight per request. For a better graphic representation, when a request is not served, the weight of this request is set to zero. A request is not served when there are no resources available. In the graphic we observe that unlike test1 (no load balancing test), in the wireless network, the weight increases steadily as requests are served; this is due to the fact that the wireless weight, increases moderately after the use of a link. Again we observe that the priority of the implemented algorithm is to find the wireless path with less weight, and the wired path depends of the wireless path.

Fig. 4. 12. Weight per Request

In Fig. 4. 13, we represent the number of hops per request, for a better graphic representation, when a request is not served the number of hops of this request is set to zero. A request is not served when there are no resources available. In the figure, we observe the point at which the wireless and consequently the wired path changes of gateway, unlike test1, request 0 and 1 use GW4, request 2 and 3 use GW5. Then, across the following requests, the use of one gateway or another is alternated; due to the load balancing that produces the increasing of the weight of the used link. At the end, the resources are exhausted and no more requests are served. Again, in this case it can be appreciated that for wired network it is better to use the GW5, but as the priority of our global embedding algorithm is to minimize the wireless weight, the gateway is selected according to the wireless optimal results.

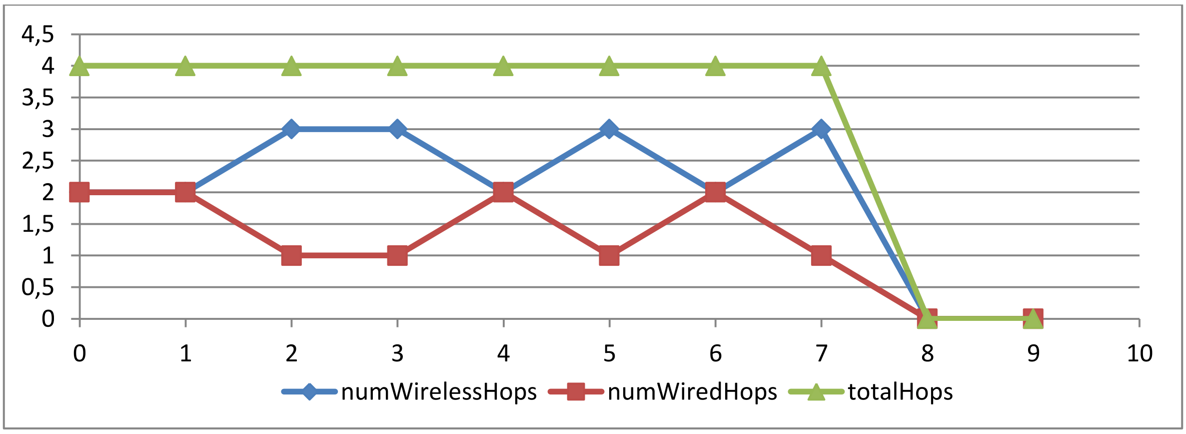


Fig. 4. 13. Number of Hops.

4.3 Conclusions of the test results

The proposed global embedding algorithm efficiently uses the resources of the physical substrate network to find an effective global path for an integrated wireless and wired network.

Due to the critical problem with interferences in the WMNs, the global embedding algorithm prioritizes the WMNs, and finds the most efficient path in terms of less interference.

Once the minimum weight path for the wireless network is found, it is calculated the minimum weight path of the wired network, but taking as origin the selected wireless gateway.

The load balance functionality, balances the requests along the network, and makes a more equilibrated use of the resources.

In the following sections we will use a higher scenario. We consider that it is necessary to analyze the confidence level of our results and at the same time, to simulate a network with requests with different sources (all of them located in wireless network) but the same destination (located in wired network) to study the network behavior in a more realistic scenario.

5 Simulations and Results

This chapter aims to evaluate the performance of our global embedding algorithm proposal. In order to do this, we have structured it into different parts. First, there are described the scenarios created to test the algorithm under different network configurations. Secondly, in order to establish a performance comparative, we present the results using the EXCEL application, for each graph, we make an analysis of the results obtained. Next, we perform a statistical analysis of the data obtained. Finally, we will discuss results and conclusions.

5.1 Scenarios

In the section Project Design, we have detailed the features used for the project scenario. Our scenario is based on a video on demand service located in a wired network (Internet), where requests are received and served at a specific time. Clients are located in a neighboring community-based wireless mesh network. Clients are not directly connected to the wired network; to access to the video server, they have to connect to the nearest wireless mesh router. Wireless mesh routers are equipped with two radios for the backbone communication and one radio for the user's communication. An example of this architecture is a neighborhood community network that can be built using infrastructure meshing. The mesh routers are placed on the roof of the houses, which serve as access points for users inside their homes and along the roads.

We classify the scenarios into two main groups. The first one, where there is a unique origin for all the requests, and the second one, more realistic; where there are different random origins in the requests file. In both groups all the nodes generate traffic to the same destination, a video server located in the wired network.

In the table below, different potential scenarios for simulation purposes are described. To be able to analyze the results statistically, more than 30 simulations are made per each selected scenario.

Network Type	Load Functionality	Num. of GW	Num. of Requests	Test
Small Network (20)	No load	2 GW	Request 110	Wireless network saturation: GW
		5 GW	Request 110	
	Load	2GW	Request 110	Wireless network saturation :GW
		5GW	Request 110	
Large Network (100)	No Load	2GW	Request 110	Wireless network saturation : GW
		20GW	Request 110	Wired network saturation: Server
	Load	2 GW	Request-110	Wireless network saturation : GW
		20GW	Request-110	Wired network saturation: Server

Fig. 5. 1. Possible Scenarios for Testing.

5.2 Simulation settings

This section includes the most relevant parameters configured on the developed simulator to evaluate the proposed global embedding algorithm.

Wireless Network	Big	Small
Number of Nodes	100	20
Number of Links	200	40
Degree	2	2
Number of GW	20/2	5/2
CPU	1-10	1-10
BW	1-10	1-10
Number of Radios	2	2
Number of available channels	4	4

Fig. 5. 2. Wireless Network Parameters

Wired Network	
Number of Nodes	400
Number of Links	1600
Degree	4
CPU	50-100 units
BW	50-100 units

Fig. 5. 3. Wired Network Parameters

5.3 Statistical Hypothesis Testing

Investigations depart from the knowledge provided by the data collected from a sample and the target is to infer population characteristics of the collected data. A statistical hypothesis test is a method of statistical inference using data from a scientific study. In statistics, a result is called statistically significant if it is unlikely to have occurred by chance alone, according to a pre-determined threshold probability called the significance level.

One use of hypothesis testing is deciding whether experimental results contain enough information to cast doubt on conventional wisdom. In frequency probability, decisions are almost always made using null-hypothesis tests (i.e., tests that answer the question : assuming that the null hypothesis is true, what is the probability of observing a value for the test statistic that is at least as extreme as the value that was actually observed?).

The critical region of a hypothesis test is the set of all outcomes which, if they occur, will lead us to decide that there is a difference. That is, cause the null hypothesis to be rejected in favor of the alternative hypothesis.

Testing Hypotheses about Proportions.

The confidence interval of a population parameter is defined as the range of values obtained from the statistical sample, bounded by its lower and upper limits. This interval will cover the value of the population parameter with a probability $1-\alpha$, called "confidence level". Specifically, the confidence interval of the proportion defines us between what two values will be the population proportion, with the probability or confidence level that we have previously set.

The proportion or relative frequency of occurrence of an observation (p) is the ratio between the number of times that the observation appears and the total number of observations. The sampling distribution of the proportion is a binomial distribution that approaches to the normal distribution when the sample size is large (n >25). Mean and standard deviation (standard error of proportion) are worth:

$$\mu_p = \pi \qquad \text{(Eq. 5.1)}$$

$$\sigma_p = \sqrt{\frac{\pi(1-\pi)}{n}} \qquad \text{(Eq. 5.2)}$$

From this assumption and considering that the proportion observed in the sample, p, is the unbiased estimator of the population proportion, π_0, the confidence interval of the proportion in a sample is obtained adding the maximum estimation error to the observed proportion in the sample, the lower limit and the upper limit are worth:

$$l_{lower} = p - Z_{\frac{\alpha}{2}} \times \sigma_p = Z_{\frac{\alpha}{2}}\sqrt{\frac{p(1-p)}{n}} \qquad \text{(Eq. 5.3)}$$

$$l_{upper} = p - Z_{1-\frac{\alpha}{2}} \times \sigma_p = Z_{1-\frac{\alpha}{2}}\sqrt{\frac{p(1-p)}{n}} \qquad \text{(Eq. 5.4)}$$

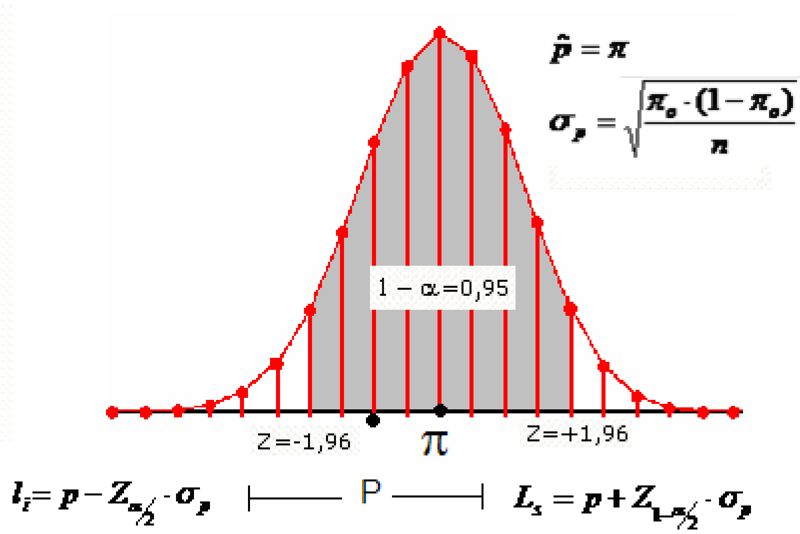


Fig. 5. 4. Confidence Intervals for a confidence level of 95%.

To test a hypothesis relating to a value hypothetically established as population proportion, π, we can determine the confidence interval and check if the value proposed in the null hypothesis, is included or not in the interval.

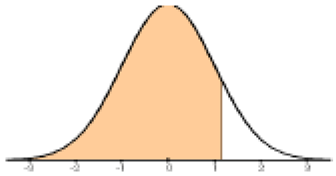
Similarly, it can be determined a statistic test to quantify the discrepancy between the value observed in the sample and the one set in the null hypothesis. Taking into account that the sampling distribution of the statistic, p, is close to normal when the samples are large (n >25) this statistic is:

$$Z = \frac{p-\pi_0}{\sigma_p} = \frac{p-\pi_0}{\sqrt{\frac{\pi_0(1-\pi_0)}{n}}} \qquad \text{(Eq. 5.5)}$$

If the null hypothesis is false this discrepancy must exceed the critical value of the sampling distribution. Similarly, the critical level p associated with this discrepancy must be less than the significance level, α, to reject the null hypothesis. Otherwise, we will not have sufficient evidence to reject this null hypothesis.

Standard normal distribution table

Inner values represent the probability of obtaining values of Z less than or equal to the standard score, z, defined by the intersection of the row with the column indicative of the second decimal. For example, the probability of scoring less or equal to 1.05 is 0.8531. That is P (Z≤ 1.05) = 0.8531.



P(Z ≤ z)

z	0	0,01	0,02	0,03	0,04	0,05	0,06	0,07	0,08	0,09
0,00	0,5000	0,5040	0,5080	0,5120	0,5160	0,5199	0,5239	0,5279	0,5319	0,5359
0,10	0,5398	0,5438	0,5478	0,5517	0,5557	0,5596	0,5636	0,5675	0,5714	0,5753
0,20	0,5793	0,5832	0,5871	0,5910	0,5948	0,5987	0,6026	0,6064	0,6103	0,6141
0,30	0,6179	0,6217	0,6255	0,6293	0,6331	0,6368	0,6406	0,6443	0,6480	0,6517
0,40	0,6554	0,6591	0,6628	0,6664	0,6700	0,6736	0,6772	0,6808	0,6844	0,6879
0,50	0,6915	0,6950	0,6985	0,7019	0,7054	0,7088	0,7123	0,7157	0,7190	0,7224
0,60	0,7257	0,7291	0,7324	0,7357	0,7389	0,7422	0,7454	0,7486	0,7517	0,7549
0,70	0,7580	0,7611	0,7642	0,7673	0,7704	0,7734	0,7764	0,7794	0,7823	0,7852
0,80	0,7881	0,7910	0,7939	0,7967	0,7995	0,8023	0,8051	0,8078	0,8106	0,8133
0,90	0,8159	0,8186	0,8212	0,8238	0,8264	0,8289	0,8315	0,8340	0,8365	0,8389
1,00	0,8413	0,8438	0,8461	0,8485	0,8508	0,8531	0,8554	0,8577	0,8599	0,8621
1,10	0,8643	0,8665	0,8686	0,8708	0,8729	0,8749	0,8770	0,8790	0,8810	0,8830
1,20	0,8849	0,8869	0,8888	0,8907	0,8925	0,8944	0,8962	0,8980	0,8997	0,9015
1,30	0,9032	0,9049	0,9066	0,9082	0,9099	0,9115	0,9131	0,9147	0,9162	0,9177
1,40	0,9192	0,9207	0,9222	0,9236	0,9251	0,9265	0,9279	0,9292	0,9306	0,9319
1,50	0,9332	0,9345	0,9357	0,9370	0,9382	0,9394	0,9406	0,9418	0,9429	0,9441
1,60	0,9452	0,9463	0,9474	0,9484	0,9495	0,9505	0,9515	0,9525	0,9535	0,9545
1,70	0,9554	0,9564	0,9573	0,9582	0,9591	0,9599	0,9608	0,9616	0,9625	0,9633
1,80	0,9641	0,9649	0,9656	0,9664	0,9671	0,9678	0,9686	0,9693	0,9699	0,9706
1,90	0,9713	0,9719	0,9726	0,9732	0,9738	0,9744	0,9750	0,9756	0,9761	0,9767
2,00	0,9772	0,9778	0,9783	0,9788	0,9793	0,9798	0,9803	0,9808	0,9812	0,9817
2,10	0,9821	0,9826	0,9830	0,9834	0,9838	0,9842	0,9846	0,9850	0,9854	0,9857
2,20	0,9861	0,9864	0,9868	0,9871	0,9875	0,9878	0,9881	0,9884	0,9887	0,9890
2,30	0,9893	0,9896	0,9898	0,9901	0,9904	0,9906	0,9909	0,9911	0,9913	0,9916
2,40	0,9918	0,9920	0,9922	0,9925	0,9927	0,9929	0,9931	0,9932	0,9934	0,9936
2,50	0,9938	0,9940	0,9941	0,9943	0,9945	0,9946	0,9948	0,9949	0,9951	0,9952
2,60	0,9953	0,9955	0,9956	0,9957	0,9959	0,9960	0,9961	0,9962	0,9963	0,9964
2,70	0,9965	0,9966	0,9967	0,9968	0,9969	0,9970	0,9971	0,9972	0,9973	0,9974
2,80	0,9974	0,9975	0,9976	0,9977	0,9977	0,9978	0,9979	0,9979	0,9980	0,9981
2,90	0,9981	0,9982	0,9982	0,9983	0,9984	0,9984	0,9985	0,9985	0,9986	0,9986
3,00	0,9987	0,9987	0,9987	0,9988	0,9988	0,9989	0,9989	0,9989	0,9990	0,9990
3,10	0,9990	0,9991	0,9991	0,9991	0,9992	0,9992	0,9992	0,9992	0,9993	0,9993
3,20	0,9993	0,9993	0,9994	0,9994	0,9994	0,9994	0,9994	0,9995	0,9995	0,9995
3,30	0,9995	0,9995	0,9995	0,9996	0,9996	0,9996	0,9996	0,9996	0,9996	0,9997
3,40	0,9997	0,9997	0,9997	0,9997	0,9997	0,9997	0,9997	0,9997	0,9997	0,9998
3,50	0,9998	0,9998	0,9998	0,9998	0,9998	0,9998	0,9998	0,9998	0,9998	0,9998

Fig. 5. 5. Standard normal distribution table.

5.4 Simulations Results

This section presents the results obtained from the simulations. We use the EXCEL application for the graphic generation.

As it was indicated at the beginning of this section, we classify the scenarios into two main groups. In the first one, there is a unique origin for all the requests, and in the second one, which is more realistic, there are different random origins in the requests file. In both groups all the nodes generate traffic to the same destination, i.e. a video server located in the wired network. To be able to analyze the results statistically, more than 30 simulations are made per request on a random generated network of similar characteristics according to a selected scenario.

We have observed that the behavior and so the conclusions are similar to the ones obtained with the simple scenario of Chapter 4. For this reason, some comments are quite parallel.

5.4.1 Unique Source

We select the following scenarios for the configuration of unique origin for all requests.

Type of Wireless Networks	Load Functionality	Num. of GW	Num. of Requests	Objective
Large Network (100 nodes)	No Load	10 GW	25 requests	Wireless network saturation: source resources
	Load	10 GW	25 requests	Wireless network saturation: source resources

Fig. 5. 6. Unique Source tests.

The following figures are generated:

- Time Elapsed per request.
- Weight per request.
- Number of Hops per request.
- Number of requests served.
- Proportion of requests served according to its request order.

5.4.1.1 No load Balancing Configuration

In this test, we have configured the global embedding algorithm, without the functionality of load balancing. We use a large wireless network composed of 100 nodes, with 10 GW, and a network manager which serves 25 requests each time. Due to unique origin configuration, and the availability of plenty wireless and wired network resources, this tests aims to highlight the limitations of the source node either by its processing capacity as for its links.

In Fig. 5. 7, we represent the average time elapsed per request. We observe an initial peak, which is the highest one, which corresponds to the initial network parameters calculation. As requests are served, the time elapsed to serve another decreases. We assume that is due to the subtraction of available resources, which decreases the time for path calculation.

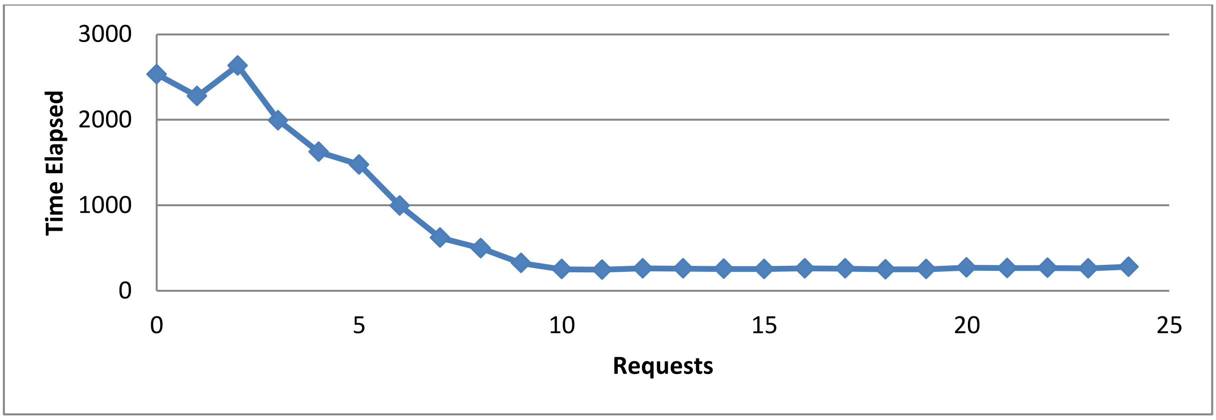


Fig. 5. 7. Average Time Elapsed per Request. Unique Source. No load balancing.

In Fig. 5. 8, we represent the average weight per request. For a better understanding of the data represented in the graph, when a request is not served, the weight of this request is set to zero. A request is not served when there are no resources available. In the figure, we can observed that the priority of the implemented algorithm is to find the wireless path with less weight, and consequently the path with less interference. The wired path depends on the wireless path. Moreover, we can observed that in the wireless network, there is the tendency of increasing the weight as request are served. Provided that wired network weight depends on the gateway selected, its value could be increased or decreased depending on the gateway selected.

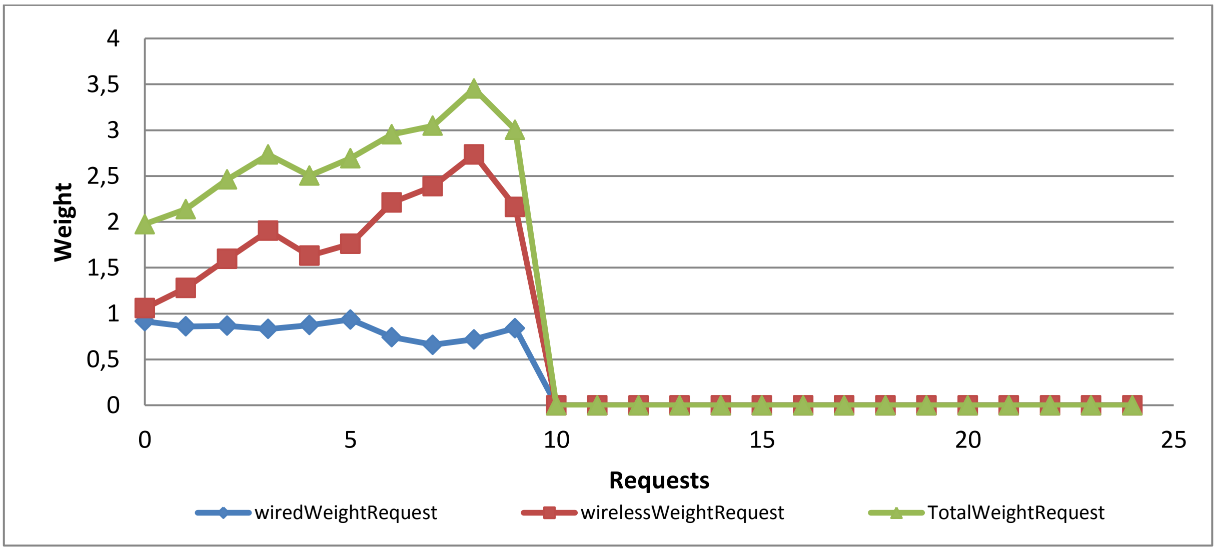


Fig. 5. 8. Average Weight per Request. Same Source. No Load.

In Fig. 5. 9, we represent the average number of hops per request, for a better graphic representation, when a request is not served the number of hops of this request is set to zero. A request is not served if there are no resources available. In the figure, we can observed that, in average, the number of wireless hops increases with the number of requests served. This is logical if one takes in mind that usually the more hops a path has, the more interference this path has.

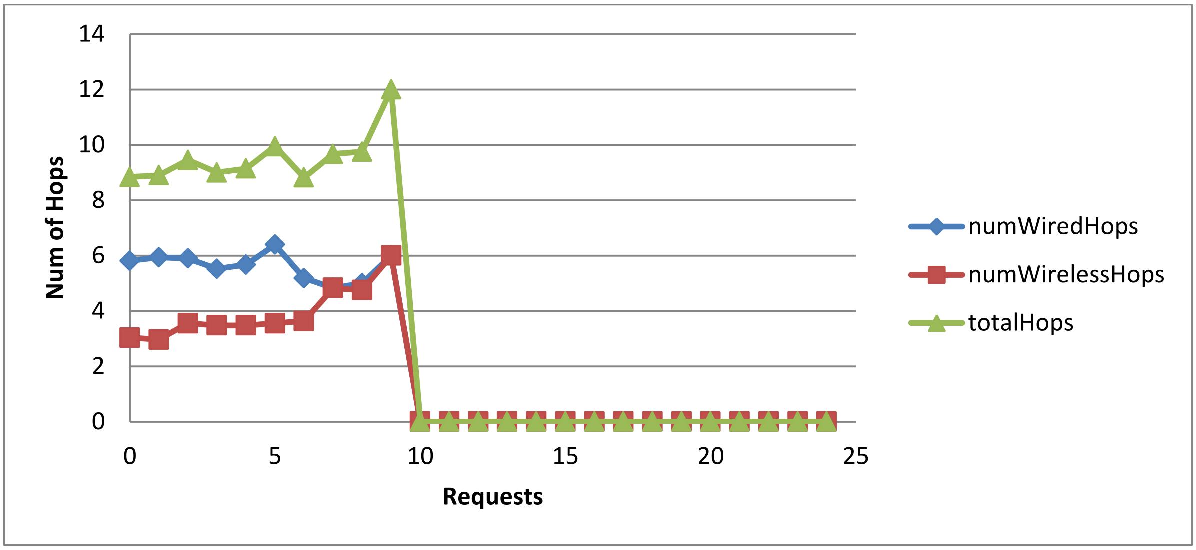


Fig. 5. 9. Average Number of Hops per Request. Same Source. No Load.

In Fig. 5. 10, we represent the proportion of requests served according to its request order in the scheduler list. For a group of 25 requests served by the SDN controller, each request is simulated 31 times on a random network of similar characteristics. When the position of a request in the scheduler list increases, there are less networks that are able to serve this request.

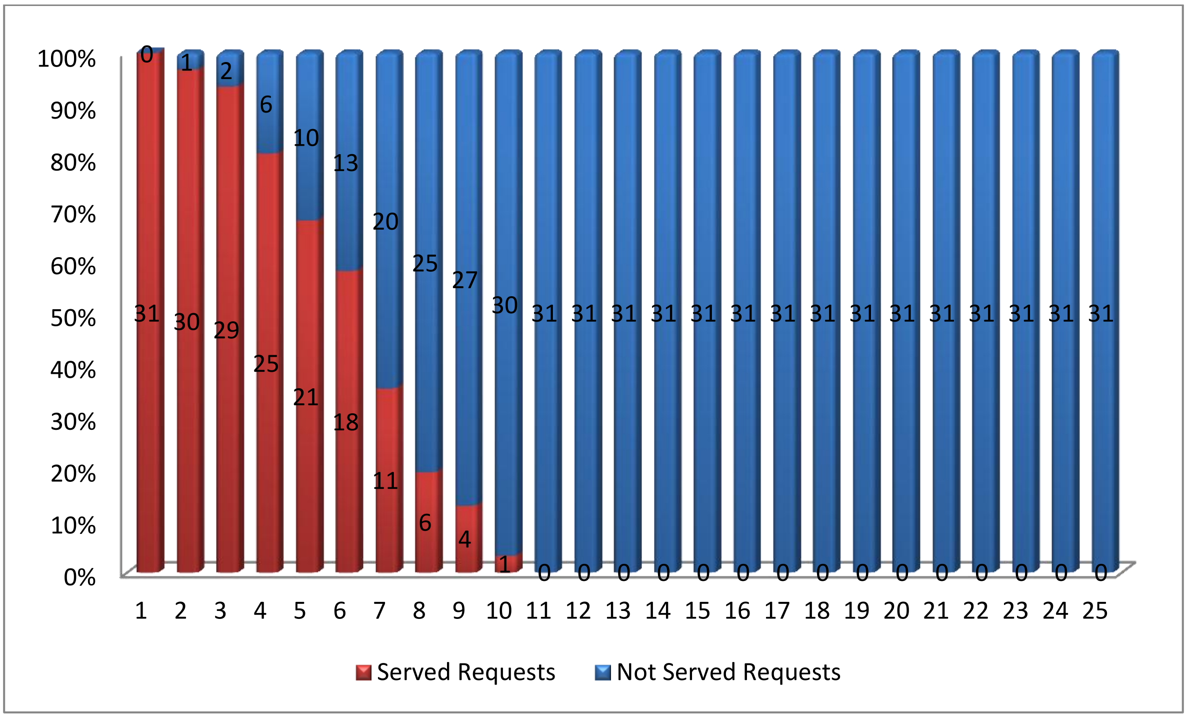


Fig. 5. 10. Proportion of requests served according to its request order.

To evaluate these results statistically, we state as Null hypothesis that:" The proportion of serving a concrete request, evaluated in more than 30 different networks with similar features, is less than 50%".

$$H_0: \pi_0 < 0,5$$

$$H_1: \pi_0 \geq 0,5$$

We calculate the discrepancy between p and π_0 measured in units of typical proportion error (assuming H_0 true).

$$Z = \frac{p - \pi_0}{\sigma_p} = \frac{p - \pi_0}{\sqrt{\frac{\pi_0(1 - \pi_0)}{n}}}$$

Serving 25 requests each time, the 6 first requests of the list are successfully served equal or more than 50, in more than 30 random networks with similar topology, (over 15 times served a request, in 30 different networks).

$$p = \frac{6}{25} = 0,24$$

According to the standard normal distribution table with a confidence level of 95%, the critic value to reject $H_0\ is\ z = 1,64$.

$$P(0.95) \rightarrow z = 1,64 \text{ Critic value}$$

$$Z = \frac{p - \pi_0}{\sigma_p} = \frac{p - \pi_0}{\sqrt{\frac{\pi_0(1 - \pi_0)}{n}}}$$

$$Z = \frac{0.24 - 0.5}{\sqrt{\frac{0.5(1 - 0.5)}{30}}} = \frac{-0.26}{0.091} = -2.85$$

With a value of Z=-2.85 we couldn't reject the null hypothesis. At this point we have two options, if we want that the half or more of the requests are served, each time. We can decrease the probability or decrease the number of requests served each time.

According to the standard normal distribution table with a confidence level of 95% the critic value to reject $H_0\ is$:

$$P(0.95) \rightarrow z = 1,64 \text{ Critic value}$$

If we decrease the number of requests, served each time and select a file of 7 requests:

$$p = \frac{6}{7} = 0.85$$

Then,

$$Z = \frac{0.85 - 0.5}{\sqrt{\frac{0.5(1-0.5)}{30}}} = \frac{0.35}{0.091} = 3.84$$

With 7 requests we could reject the null hypothesis and say that with a confidence of 95% equal or more than 50% of the requests will be served.

For the following tests we will take the confidence value of 95%

5.4.1.2 Load balancing Configuration

For the following test, we have configured the global embedding algorithm with the functionality of load balancing. The load balancing functionality produces that each time that a link is used for a path, the interflow value of this link is increased. We use a large wireless network composed of 100 nodes, with 10 GW, and a SDN controller which serves 25 requests each time. Due to configuration of a unique origin, and the availability of plenty wireless and wired network resources, this test aims to highlight the limitations of the source node either by its processing capacity as for its links. Besides, this test also wants to evaluate the behavior of the load balancing functionality. For a better graphic representation, when a request is not served, the weight or number of hops of this request is set to zero. A request is not served when there are no resources available

In Fig. 5. 11, we represent the average time elapsed per request, we can see an initial peak (the highest one) that corresponds to the initial network parameters calculation. As requests are served, the time elapsed to serve another decreases. We assume that is due to the subtraction of available resources, which decreases the time for path calculation.

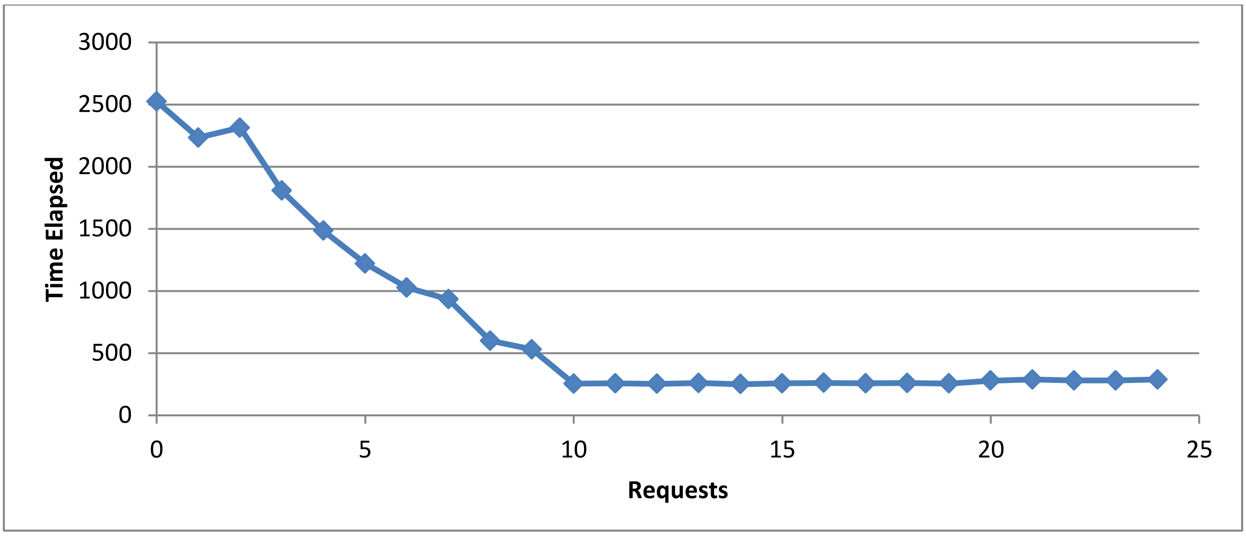


Fig. 5. 11. Average Time per Request. Same Source. Load balancing.

In Fig. 5. 12, we represent the average weight per request. In the figure we observe that, unlike the “no load-balancing test”, in the wireless network the weight increases steadily as requests are served. This is due to the fact that the wireless weight increases moderately after the use of a link.

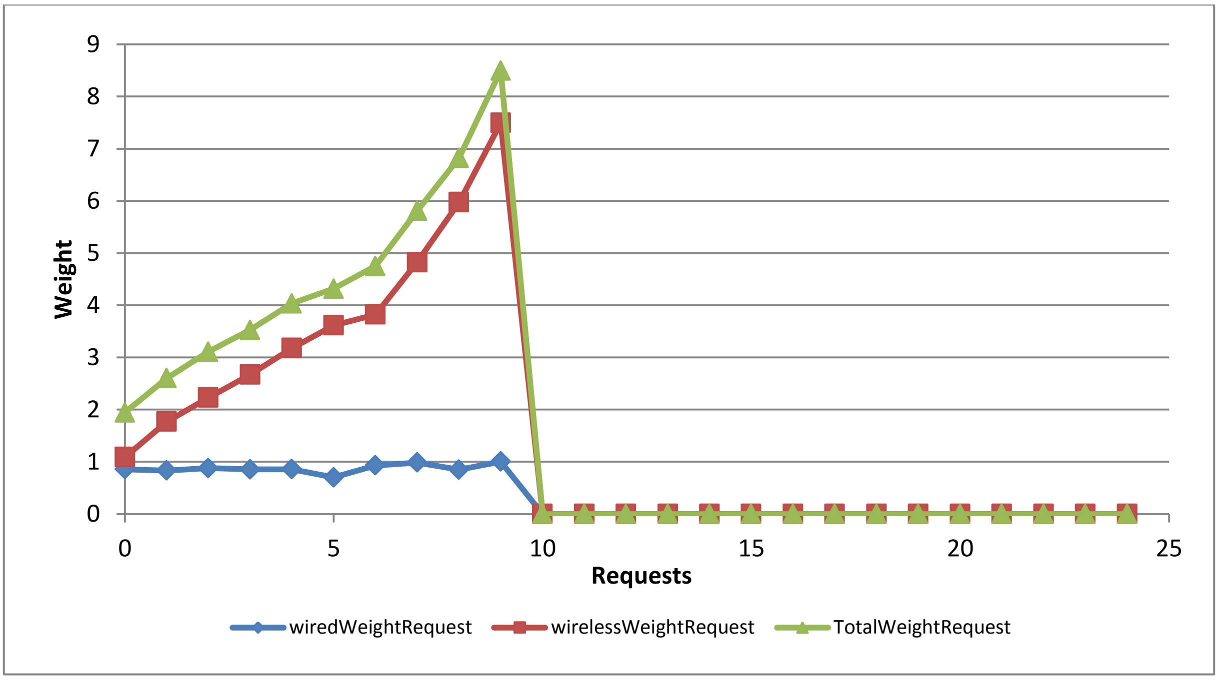


Fig. 5. 12. Average Weight per Request. Same Source. Load balancing.

In Fig. 5. 13, we represent the average number of hops per request. In the figure, we can see that, in average, the number of wireless hops increases with the number of requests served. This is coherent if one takes in mind that usually the more hops a path has, the more interference it has.

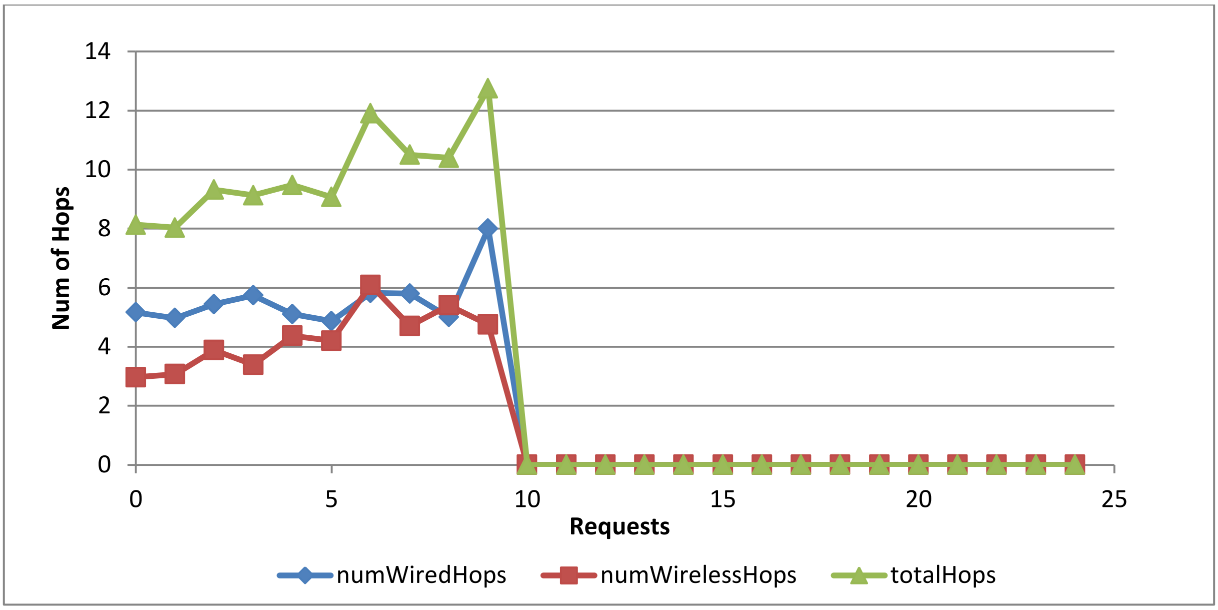


Fig. 5. 13. Average Number of Hops per Request. Same origin. Load balancing.

In Fig. 5. 14, we represent the proportion of requests served according to its request order in the scheduler list. We observe that for a group of 25 requests served by the controller each request is simulated 30 times on a random network of similar characteristics. According as request order increases there is less proportion of served requests.

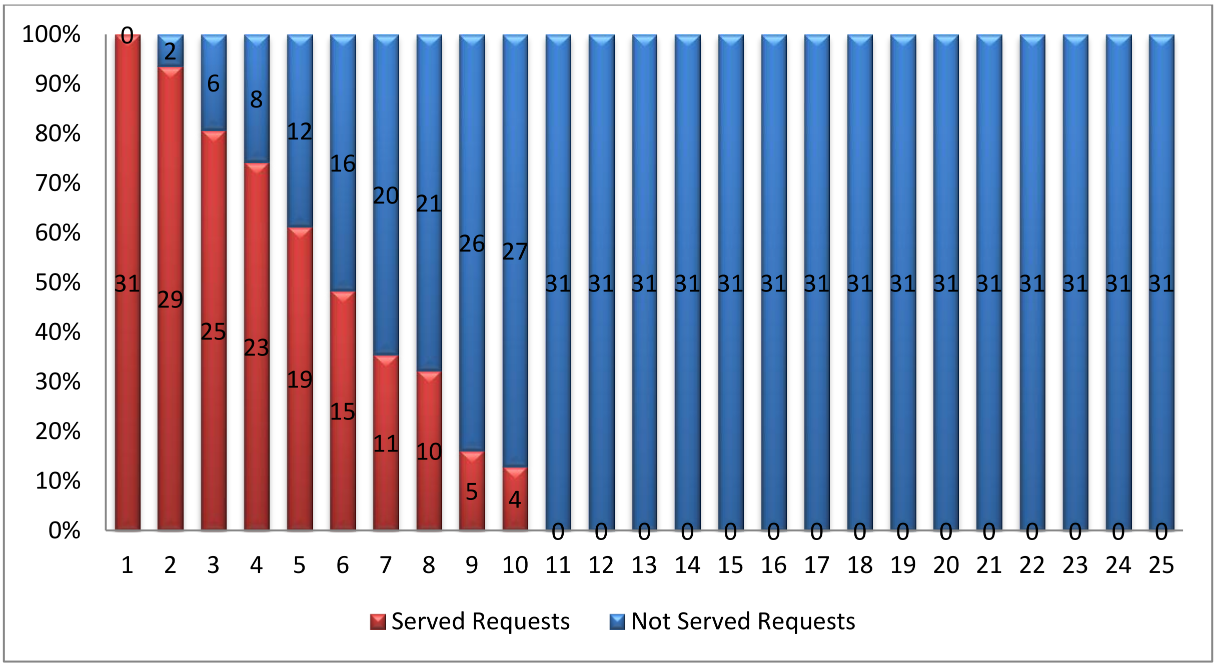


Fig. 5. 14. Proportion of requests served according to its request order.

To evaluate these results statistically, we state as Null hypothesis that: "The proportion of serving a concrete request, evaluated in more than 30 different networks with similar features, is less than 50%".

$$H_0: \pi_0 < 0{,}5$$

$$H_1: \pi_0 \geq 0{,}5$$

We calculate the discrepancy between p and π_0 measured in units of typical proportion error (assuming Ho true).

$$Z = \frac{p - \pi_0}{\sigma_p} = \frac{p - \pi_0}{\sqrt{\frac{\pi_0(1-\pi_0)}{n}}}$$

According to the Standard normal distribution table with a confidence level of 95%, and 80% the critic value to reject $H_0\ is$:

$$P(0.95) \rightarrow z = 1{,}64 \text{ Critic value}$$

$$P(0.80) \rightarrow z = 0{,}85 \text{ Critic value}$$

Serving 10 requests each time, the 6 first requests of the list are successfully served equal or more than 50%, in more than 30 random networks with similar topology, (over 15 times served a request, in 30 different networks).

$$p = \frac{6}{10} = 0.6$$

Then

$$Z = \frac{0.6 - 0.5}{\sqrt{\frac{0.5(1-0.5)}{30}}} = \frac{0.1}{0.091} = 1.09$$

With a value of Z=1.09 we couldn't reject the null hypothesis with a confidence of 95%, but we could reject the null hypothesis and say that with a confidence of 80% equal or more than 50% of the requests will be served. Finally, if we want a confidence of 95% we have to take fewer requests per time. If we decrease the number of requests served each time and select a file of 7 requests:

$$p = \frac{6}{7} = 0.85 \qquad Z = \frac{1-0.5}{\sqrt{\frac{0.5(1-0.5)}{6}}} = \frac{0.35}{0.091} = 3.84$$

With 6 requests we could reject the null hypothesis and say that with a confidence of 95% Serving 6 requests each time, the proportion of serving a concrete request, evaluated in more than 30 different networks with similar features, is equal or more than 50%

5.4.2 Different sources

As it was indicated at the beginning of this section, we classify the scenarios into two main groups. In the first one, there is a unique origin for all the requests, and the second one, more realistic, there are different random origins in the requests file. All the requests have the same destination. This setting is more in line with the topology of a video on demand service. Different customers located in a wireless network, request to access to a video server located on a wired network. In this tests series, it has been configured the global embedding algorithm with the functionality of load balancing. We test the behavior of the algorithm in the following fields: Wired Network saturation, GW saturation and Server saturation.

Type of Wireless Networks	Load Functionality	Num. of GW	Num. of Requests	Objective
Medium Network (50 nodes)	Load	2 GW	150 requests	Wireless network saturation: GW
Large Network (100 nodes)	Load	20GW	150 requests	Wired network saturation: Server

Fig. 5. 15. Different sources tests.

The following graphs are generated:

- Time Elapsed per request.
- Weight per request.
- Number of Hops per request.
- Number of requests served.
- Proportion of requests served according to its request order.

5.4.2.1 Large Network with 20GW

In this test, we have configured the global embedding algorithm with the functionality of load balancing. We use a large wireless network composed of 100 nodes, with 20 GW, and a network manager which serves 130 requests each time. Due to the different origin of the clients, and the availability of plenty wireless and wired network resources, this tests aims to highlight the limitations of the video server node either by its processing capacity as for its links. For a better graphic representation, when a request is not served, the weight or number of hops of this request is set to zero. A request is not served when there are no resources available.

In Fig. 5. 16, we represent the average time elapsed per request, We see an initial peak, which is the highest one, that corresponds to the initial network parameters calculation. As requests are served, the time elapsed to serve a request decreases. We assume that is due to the subtraction of available resources, which decreases the time for path calculation.

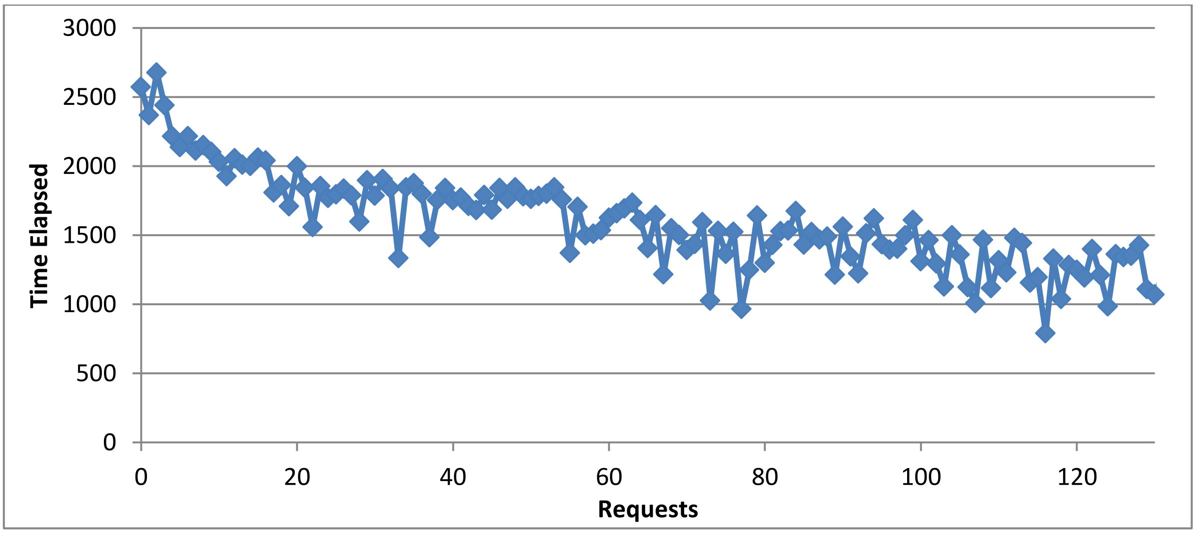


Fig. 5. 16. Average Time per Request. Different Sources. Large Network. 20GW

In Fig. 5. 17, we represent the average weight per request. In the figure we can observe that with different sources, the network is able to serve more requests each time. This contrasts with the previous tests with the same origin. This feature limited the number of requests that were served each time due to the restricted availability of the resources of the source node; if the sources are distributed this restriction decreases. We can also observe that the weight increases steadily as requests are served. This is due to the fact that the wireless weight increases moderately after the use of a link.

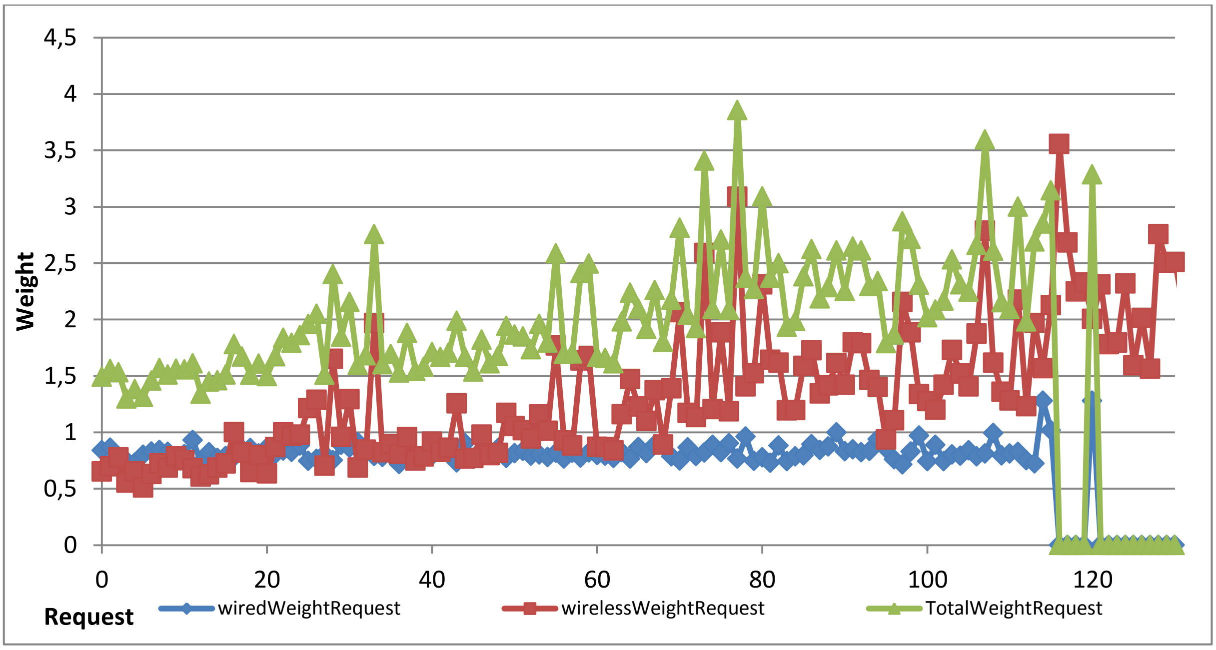


Fig. 5. 17. Average Weight per Request. Different Sources. Large Network. 20GW

In Fig. 5. 18, we represent the average number of hops per request. We observe that due to the fact that in the wireless network there are different sources, there is no clear tendency in the number of hops. In the wired network, we can see a slight tendency to increase the number of hops. This is because, as requests are served, the network resources are exhausted, and the paths need more hops to reach the destinations

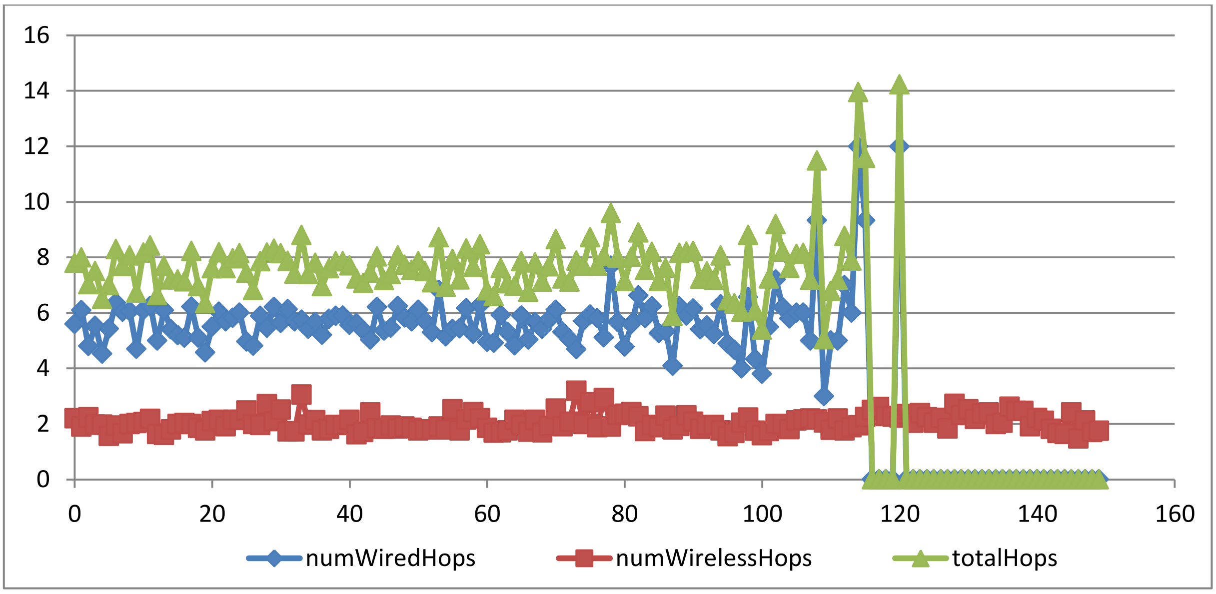


Fig. 5. 18. Average Num of Hops. Different Sources. Large Network. 20GW

In Fig. 5. 19, we represent the proportion of requests served according to its request order in the scheduler list. We observe that for different-source requests, the number of requests served increases dramatically in front the same-source requests. This is due to the resource restrictions of the source. Moreover, it can be noticed that as the requests are served, the proportion of served requests decreases; this is due to the network resource exhaustion. In this scenario, the constraining resource is the video server, and the requests are not served when its capacity, randomly set between 50 -100, is exhausted.

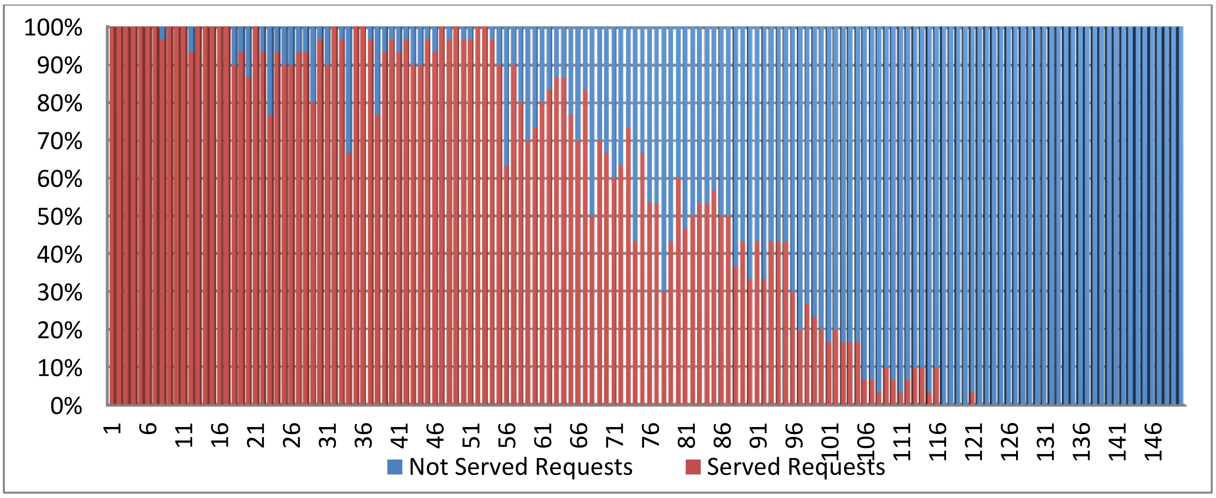


Fig. 5. 19. Proportion of requests served according to its request order

To evaluate these results statistically, we state the Null hypothesis as follows: “The proportion of serving a concrete request, evaluated in more than 30 different networks with similar features, is less than 50%”.

$$H_0: \pi_0 < 0{,}5$$

$$H_1: \pi_0 \geq 0{,}5$$

We calculate the discrepancy between p and π_0 measured in units of typical proportion error (assuming Ho true).

$$Z = \frac{p - \pi_0}{\sigma_p} = \frac{p - \pi_0}{\sqrt{\frac{\pi_0(1-\pi_0)}{n}}}$$

According to the Standard normal distribution table with a confidence level of 95%, the critic value to reject $H_0\ is$:

$$P(0.95) \rightarrow z = 1{,}64 \text{ Critic value}$$

If we take 150 requests, each time:

$$p = \frac{88}{150} = 0.586$$

Then

$$Z = \frac{0.586 - 0.5}{\sqrt{\frac{0.5(1-0.5)}{30}}} = \frac{0.086}{0.091} = 0.94$$

With a value of Z=0.94 we couldn’t reject the null hypothesis. At this point we have two options, if we want that the half or more of the requests are served, each time. We can decrease the probability or decrease the number of requests served each time. If we take 100 requests, each time:

$$p = \frac{88}{100} = 0.88$$

Then

$$Z = \frac{0.88 - 0.5}{\sqrt{\frac{0.5(1-0.5)}{30}}} = \frac{0.38}{0.091} = 4.17$$

With 100 requests we could reject the null hypothesis and say that with a confidence of 95% serving 100 requests each time, the proportion of serving a concrete request, evaluated in more than 30 different networks with similar features, is equal or more than 50%.

5.4.2.2 Medium Network with 2GW

In this test, it has been configured the global embedding algorithm with the functionality of load balancing. It is used a medium wireless network composed of 100 nodes, with 2 GW, and a controller which serves 150 requests each time. Due to the scarce number of GWs, this tests aims to highlight the limitations of a scarce number of GWs in the network design. For a better graphic representation, when a request is not served, the weight and the number of hops of this request is set to zero. A request is not served when there are no resources available.

In Fig. 5. 20, we represent the average time elapsed per request, we see an initial peak (the highest one), that corresponds to the initial network parameters calculations. As requests are served, the time elapsed to serve a request decreases. We assume that is due to the subtraction of available resources, which decreases the time for path calculation.

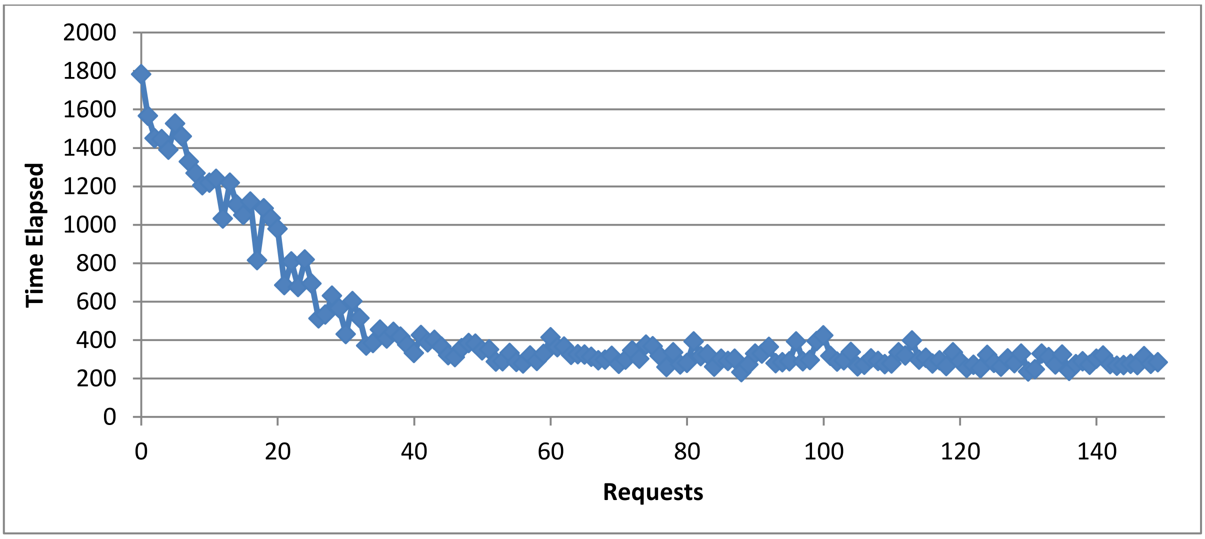


Fig. 5. 20. Average Time per Request. Different Sources. Medium Network. 2GW

In Fig. 5. 21, we represent the average weight per request. We observe that with different sources, the network is able to serve more requests each time, but due to the scarce quantity of GWs the number of served requests decreases dramatically. It can be also noticed that at first when there are plenty of wireless resources, the weight increases steadily as requests are served, this is due to the fact that wireless weight increases moderately after the use of a link. After a while, the wireless network resources, like the links BW of the both GWs are exhausted. For this reason, depending on origin of the request we could find a viable path or not.

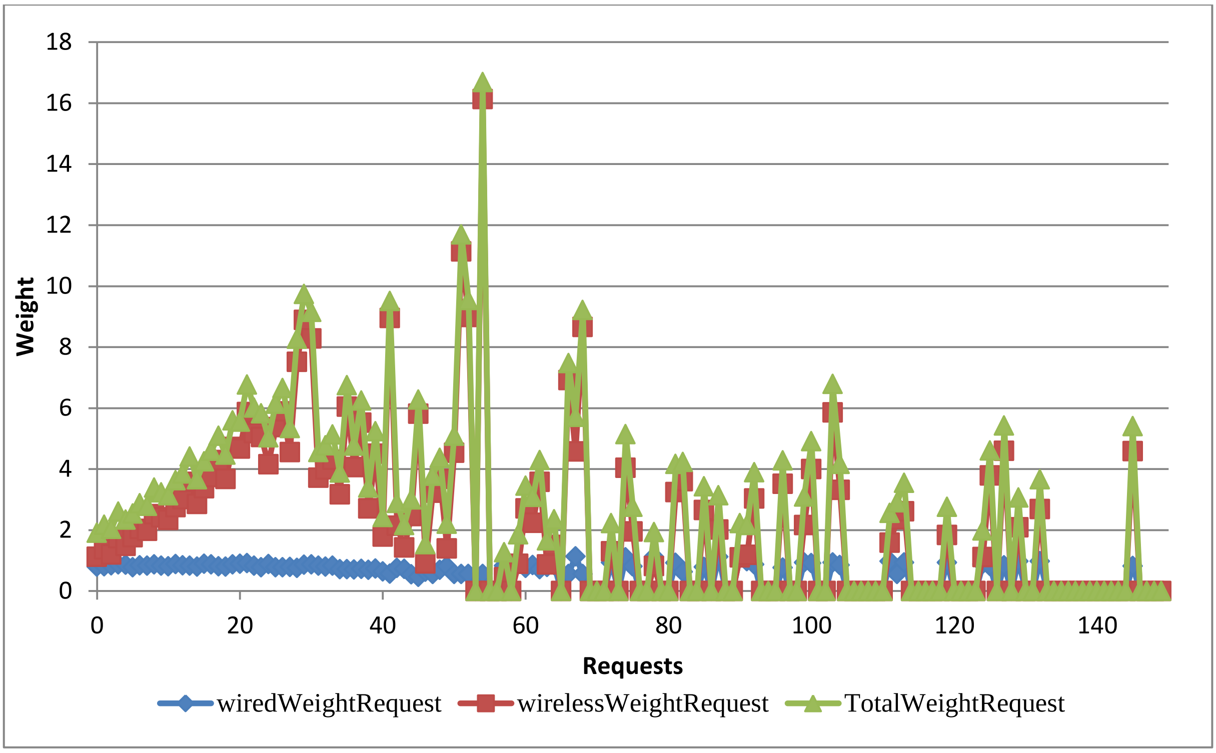


Fig. 5. 21. Average Weight per Request. Different Sources. Medium Network. 2GW

In Fig. 5. 22, we represent the average number of hops per request. We can observe that due to the fact that in the wireless network there are different sources, there is no a clear tendency in the number of hops. In the wired network, we can see a slight tendency to increase the number of hops. This is because, as the requests are served, the network resources are exhausted, and the paths need more hops to reach the destination.

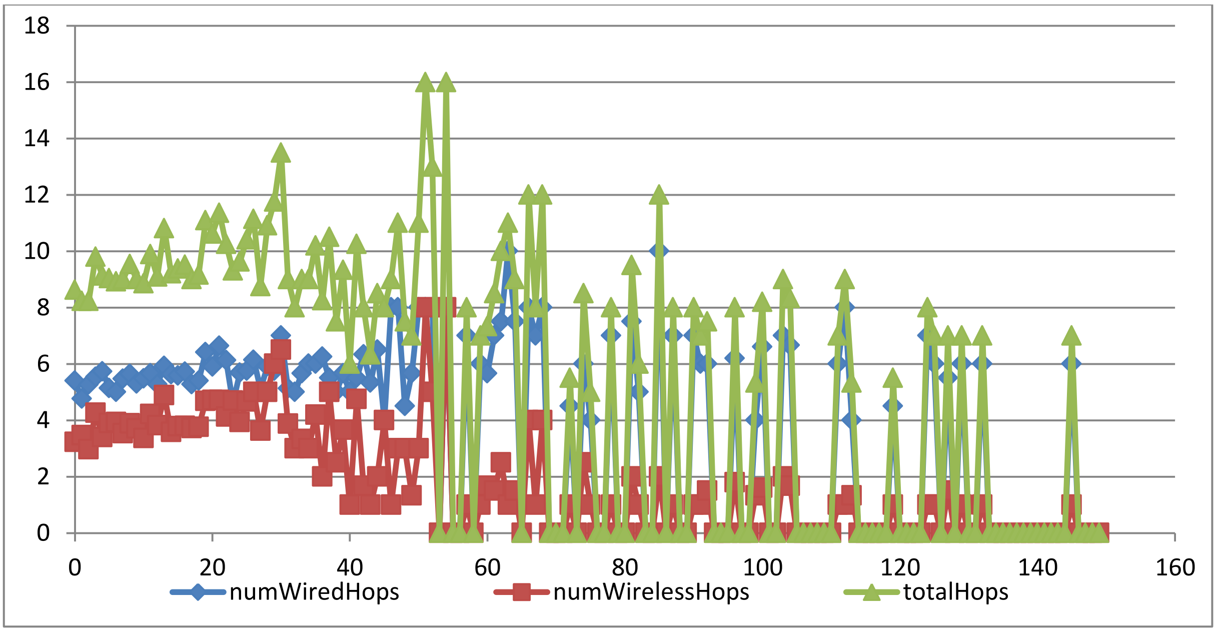


Fig. 5. 22. Average Number of hops. Different Sources. Medium Network. 2GW.

In Fig. 5. 23, we represent the proportion of requests served according to its request order in the scheduler list. We can observe that due to the scarce number of GWs, the average number of requests served decreases dramatically. Moreover, it can be noticed that as the requests are served the proportion of served requests decreases, due to the network resource exhaustion.

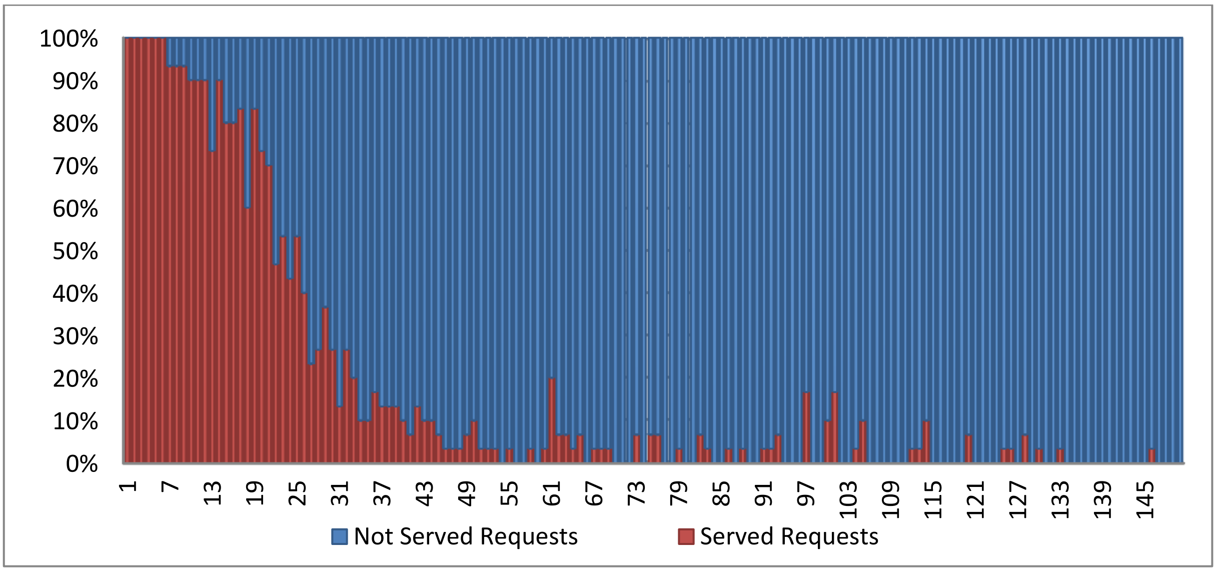


Fig. 5. 23. Proportion of requests served according its request order

To evaluate these results statistically, we state as Null hypothesis that: The proportion of serving a concrete request, evaluated in more than 30 different networks with similar features, is less than 50%.

$$H_0 : \pi_0 < 0,5$$

$$H_1 : \pi_0 \geq 0,5$$

We calculate the discrepancy between p and π_0 measured in units of typical proportion error (assuming Ho true)

$$Z = \frac{p - \pi_0}{\sigma_p} = \frac{p - \pi_0}{\sqrt{\frac{\pi_0(1-\pi_0)}{n}}}$$

According to the standard normal distribution table with a confidence level of 95%, the critic value to reject $H_0\ is:\ P(0.95) \rightarrow z = 1,64$.

If we take 100 requests, each time:

$$p = \frac{20}{100} = 0.2$$

$$Z = \frac{0.2 - 0.5}{\sqrt{\frac{0.5(1-0.5)}{30}}} = \frac{-0.3}{0.091} = -3.29$$

With a value of Z = -4 we couldn't reject the null hypothesis with a confidence of 95%.We have to take less requests per time

$$p = \frac{20}{30} = 0.66$$

$$Z = \frac{0.66 - 0.5}{\sqrt{\frac{0.5(1-0.5)}{30}}} = \frac{0.166}{0.091} = 1.824$$

With 30 requests we could reject the null hypothesis and say that, with a confidence of 95%, serving 30 requests each time, the proportion of serving a concrete request evaluated in more than 30 different networks with similar features, is equal or more than 50%.

5.5 Conclusions of the simulations results

In this subsection, we summarize the conclusions obtained from the different simulations, identifying the main points to take into account.

Impact of the requests origin. From the graphics, if the tests of the same origin and the tests of different origins are compared, we see that due to the scarce resources of the clients located in neighborhoods, in contrast of resources usually available on wired and wireless mesh backbones or the video server; when requests have a common origin, less requests are served than when requests are served from different sources. Therefore, if a client wants to perform multiple service requests at once, it should properly size their resources, e.g. the number of radios, capacity, etc.

Impact of wireless network priority for minimum weight path calculation. From the tests with same origin requests, we can evaluate the weight behavior. We observe that in wireless networks, the weight increases as requests are served. Due to the critical problem with the interferences in the WMNs, the global embedding algorithm prioritizes the WMNs, and finds the most efficient path in terms of less interference. Once the minimum weight path for the wireless network is found, it is calculated the minimum weight path of the wired network taking as origin the selected wireless gateway.

Impact of substrate network size. From the graphics, in the average time elapsed per request, we notice an initial peak that corresponds to the initial network parameters calculation that is the highest one in the graph. As requests are served, the time elapsed to serve a request decreases. We assume that is due to the subtraction of available resources, which decreases the time for path calculation. In the tests, we have evaluated three types of networks, small, medium and large, and we can observe that as network density increases the average time elapsed increases.

Impact of number of GW. From the test of the Large Network size with 20 GW at point 5.4.3.1 and the test of the Medium Network size with 2 GW at point 5.4.3.2. We notice that gateways, that interconnect wireless with wired networks, are bottlenecks, and need to be sized according to the service that is wanted to provide.

Impact of the server resources. From the test of the Large Network size with 20 GW at point 5.4.3.1, due to the different clients, and the availability of plenty wireless and wired network resources, this tests highlight the limitations of the video server node either by its processing capacity as for its links. Video server is another bottleneck, and needs to be sized according to the service required.

Impact of the number of request. In the results, we can see that, in average, the number of wireless hops increases with the number of requests served. This is coherent if one takes in mind that usually the more hops a path has, the higher interference it has. Besides, it can be also observed that the configuration than serves less request per time is the scenario where all the requests have the same origin, statistically we could affirm that with a confidence of 95% serving 6 requests each time, the proportion of serving a concrete request, evaluated in more than 30 different networks with similar features, is equal or more than 50%. Next configuration is the Medium Network with 2 GW test at point 5.4.3.2. Due to the scarce number of GW, this tests aims to highlight the limitations of a scarce number of GW in the network design. Statistically, we can state that with a confidence of 95%, serving 30 requests each time, the proportion of serving a concrete request is equal or more than 50%. Finally, with the scenario Large Network with 20 GW at test 5.4.3.1, we could state that with a confidence of 95%, serving 100 requests each time, the proportion of serving a concrete request, evaluated in more than 30 different networks with similar features, is equal or more than 50%.

From our point of view, the clients usually don't ask for several videos at the same time, and it is easy to provide the video server with resources. The critical point is the planning of the necessary number of GWs, according to the dimension of the network and the number of simultaneous requests.

Impact of Load Balancing functionality. In the wireless network, the weight increases steadily as requests are served due to the fact that the wireless weight increases moderately after the use of a link. In wireless networks the load balancing is an important functionality due to the fact that as a link is used its interference increases.

6 Conclusions and Future Work

The main objective of this project is to design and implement a simulator of a global embedding algorithm for VN requests that integrates a multi-channel wireless mesh network with a wired network. We propose to integrate Software Defined Networking (SDN) and network functions virtualization (NFV) principles in heterogeneous networks.

Our scenario is based on a video on demand service located in a wired network (Internet), where requests are received and served at a specific time. Clients are located in a wireless mesh network in a neighboring community. This project is specifically focused on the algorithm of the SDN controller that dynamically provision and manage virtual resources. The controller's orchestrator function interworks with the NFV server to manage network resources for the coordinated control of the end-to-end infrastructure. We propose to balance traffic among the gateways of a Wireless Mesh Network, implementing the gateway balancing logic in the controller.

The key challenge of this project is the embedding problem in wireless mesh networks, due to the complex interference among links and its severe coupling with network topology. This challenge requires the design of embedding algorithms adapted to WMNs particular features. To perform this, we have designed a global virtual access network embedding algorithm that can ensure the coexistence of different requests on the same substrate and provide the most suitable network resources according to the requirements of the requests, networks constraints (CPU, BW), and network load balancing. We have used Dijkstra's algorithm to find the minimum path weight and to provide a global wired and wireless network path. In WMNs we have used a metric that takes into account the intraflow and interflow interferences. Due to the non-isotonic nature of the intraflow interference value and its incompatibility with the Dijkstra's algorithm, we have implemented a mechanism to convert non-isotonic metrics into isotonic ones.

To provide a tool to evaluate the proposed algorithm, we have implemented a framework simulator in java. This simulator has allowed us to test the performance of the proposed global embedding algorithm on several scenarios of random heterogeneous wireless and wired networks combined with random requests. The data obtained from the simulator has been processed and presented in EXCEL graphs.

From the results obtained, we state that the proposed global embedding algorithm efficiently uses the resources of the physical substrate network to find an effective global path for an integrated wireless and wired network. The global embedding algorithm prioritizes the WMNs, and finds the most efficient path in terms of less interference. The load balance functionality, balances the requests along the network, and makes a more equilibrated use of the resources. This is an important functionality in WMNs, because the main problems in wireless networks are due to the interferences, if the links are used in a distributed way, the interferences are also distributed.

As design guidelines, from our point of view the critical points in the network design of an integrated wired and wireless mesh network that want to provide Internet services, are the following ones. First, the number of GW. It is necessary to find a balance between the number of GW, the network size and client requirements. Second, the capacity of client nodes. They must be able to meet the requirements of its own requests. Third, the video server capacity, its redundancy and its connection degree to the wired network. Finally, the channel assignment and the radios features, which are out of scope of this project.

We conclude that the global virtual access network embedding framework and its developed software simulator work efficiently and effectively, and detect weaknesses that should be taken into account in the network design. Our embedding algorithm may leave some rooms for further improvement, but our findings on the key features are expected to provide useful implications to the embedding algorithm research in wireless mesh networks. Additionally, the development of the framework simulator can motivate researchers to contribute to the maturity of the mesh network domain from a simulation perspective.

During the development of this project, we had got some ideas to understand better and improve the obtained results, but due to the limited duration of this project we didn't implement all of them. We propose them as a future research lines, namely: extension of the algorithm to include the channel assignment functionality; the configuration of different requests patterns; and the inclusion of different models of metrics evaluating the performance of the global embedding algorithm in all of these cases.

7 Bibliography

[1] I. F. Akyildiz, X. Wang, and W. Wang, "Wireless mesh networks: a survey," *Computer Networks. ISDN Syst.*, vol. 47, no. 4, pp. 445–487, 2005.

[2] Parth H. Pathak and Rudra Dutta, "A Survey of Network Design Problems and Joint Design Approaches in Wireless Mesh Networks," *IEEE COMMUNICATIONS SURVEYS & TUTORIALS*, vol. 13, no. 3, 2011.

[3] E. C. Efstathiou, P. A. Frangoudis, and G. C. Polyzos, "Stimulating participation in wireless community networks," *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, pp. 1–13, April 2006.

[4] P. Antoniadis, B. Le Grand, A. Satsiou, L. Tassiulas, R. Aguiar, J. Barraca, and S. Sargento, "Community building over neighborhood wireless mesh networks," *IEEE Technol. Soc. Mag.*, vol. 27, no. 1, pp. 48–56, Spring 2008.

[5] Yang Yalin, Wang Jun, Kravets Robin, "Designing Routing Metrics for Mesh Networks," *Proceedings of the IEEE Workshop on Wireless Mesh Networks (WiMesh). IEEE Press*, 2005.

[6] Yaling Yang, Jun Wang, and Robin Kravets, "Interference-Aware Load Balancing for Multihop Wireless Mesh Networks," *ACM Mobihoc, 2005.*

[7] Yaling Yang and Robin Kravets, "Contention-Aware Admission Control for Ad Hoc Networks," *IEEE Transactions on Mobile Computing*, vol. 4, no. 4, pp. 363–377, 2005.

[8] R. Draves, J. Padhye, and B. Zill, "Comparison of routing metrics for static multi-hop wireless networks," *ACM SIGCOMM*, 2004.

[9] Richard Draves, Jitendra Padhye, and Brian Zill, "Routing in Multi-Radio, Multi-Hop Wireless Mesh Networks," *ACM Mobicom*, 2004.

[10] Yaling Yang, Jun Wang, and Robin Kravets, "Interference-aware Load Balancing for Multihop Wireless Networks," *Department of Computer Science, University of Illinois at Urbana-Champaign, 2005.*

[11] David B Johnson and David A Maltz, "Dynamic Source Routing in AdHoc Wireless Networks," *Mobile Computing. Kluwer Academic Publishers*, vol. 353, 1996.

[12] Charles Perkins. "Ad-hoc on-demand distance vector routing," *In MILCOM panel on Ad Hoc Networks*. 1997.

[13] Yaling Yang, Jun Wang, Robin Kravets "Designing Routing Metrics for Mesh Networks," *IEEE Workshop on Wireless Mesh Networks, 2005.*

[14] Pradeep Kyasanur and Nitin Vaidya. "Multi-Channel Wireless Networks: Capacity and Protocols," *Tech. Rep., University of Illinois at Urbana-Champaign*, 2005.

[15] Hossam Hassanein and Audrey Zhou. "Routing with Load Balancing in Wireless Ad hoc Networks," *ACM MSWiM, **2001.***

[16] Sung-Ju Lee and Mario Gerla."Dynamic Load-Aware Routing in Ad hoc Networks," *IEEE ICC*, 2001.

[17] Atul Khanna and John Zinky. "The Revised ARPANET Routing Metric,"*ACM SIGCOMM*, 1989.

[18] E. Anderson and T. Anderson."On the Stability of Adaptive Routing in the Presence of Congestion Control," *IEEE INFOCOM*, 2003.

[19] J. L. Sobrinho."Algebra and algorithms for QoS path computation and hop-by-hop routing in the Iternet," *IEEE INFOCOM*, 2001.

[20] J. L. Sobrinho. "Network Routing with Path Vector Protocols: Theory and Applications," *ACM SIGCOMM*, pp. 49-60, 2003.

[21] Charles Perkins, Pravin Bhagwat. "Highly dynamic destination sequenced distance-vector routing (DSDV) for mobile computers," *ACM SIGCOMM*, 1994.

[22] Tsu-Wei Chen, Mario Gerla. "Global State Routing: A New Routing Schemes for Ad-hoc Wireless Networks," *IEEE ICC*, 1998.

[23] W. Xiang, T. Pratt, X. Wang. "A software radio testbed for two-transmitter two-receiver space time coding wireless LAN," *IEEE Communications Magazine*, vol. 42, no 6, pp. S20–S28, 2004.

[24] K. Sundaresan, R. Sivakumar, M.A. Ingram, T.-Y. Chang. "A fair medium access control protocol for ad hoc networks with MIMO links," *IEEE Annual Conference on Computer Comunications (INFOCOM)*, pp. 2559–2570, 2004.

[25] A. Adya, P. Bahl, J. Padhye, A. Wolman, L. Zhou." A multi-radio unification protocol for IEEE 802.11 wireless networks," *International Conferences on Broadband Networks (BroadNets), 2004.*

[26] M. McHenry."Frequency Agile Spectrum Access Technologies," *FCC Workshop on Cognitive Radios,* May 2003.

[27] B. Lane. " Cognitive radio technologies in the commercial arena," *FCC Workshop on Cognitive Radios*, May 2003.

[28] J. Mitola III."Software Radio Architecture: Object-Oriented Approaches to Wireless System Engineering". Wiley Inter-Science, New York, 2000.

[29] K. Sundaresan, R. Sivakumar, M.A. Ingram, T.-Y. Chang. "A fair medium access control protocol for ad hoc networks with MIMO links," IEEE Annual Conference on Computer Communications (INFOCOM), pp. 2559–2570, 2004.

[30] Yaling Yang, Jun Wang, Robin Kravets. "Designing Routing Metrics for Mesh Networks," *IEEE Workshop on Wireless Mesh Networks*, 2005.

[31] P. Gupta and P. Kumar. "The capacity of wireless networks," *Information Theory, IEEE Transactions on,* vol. 46, no. 2, pp. 388–404, Mar 2000.

[32] P. Santi, R. Maheshwari, G. Resta, S. Das, and D. M. Blough. "Wireless link scheduling under a graded SINR interference model," *FOWANC '09: Proc. 2nd ACM international workshop on Foundations of wireless ad hoc and sensor networking and computing. New York, NY, USA: ACM,* 2009, pp. 3–12.

[33] G. Sharma, R. R. Mazumdar, and N. B. Shroff. "On the complexity of scheduling in wireless networks," *MobiCom '06: Proc. 12th annual international conference on Mobile computing and networking. New York, NY, USA: ACM,* pp. 227–238, 2006.

[34] R. Ramanathan, "On the performance of ad hoc networks with beam forming antennas," *ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC),* pp. 95–105, Oct. 2001.

[35] IEEE 802.11 Standard Group Web Site. Available from: http://www.ieee802.org/11/.

[36] Yaling Yang, Jun Wang. "Interference-aware Load Balancing for Multihop Wireless Networks," *C.S Illinois.* 2005.

[37] MEM Campista, PM Esposito, IM Moraes. "Routing metrics and protocols for wireless mesh networks Network," *IEEE 2008.*

[38] Roger Sedgewick, Kevin Wayne. "Algorithms "*4th. Edition. University of Princeton*. http://algs4.cs.princeton.edu/home/

[39] Cristina Cervelló, Sebastià Sallent. "QoS Routing," Master in Telematics Engineering, 2011.

[40] Roger Sedgewick, Philipe Flajolet. "An introduction to the analysis of algorithms," *University of Princeton.* http://aofa.cs.princeton.edu/home/.

[41] Roger Sedgewick. "Introduction to programming in Java," *University of Princeton.* http://introcs.cs.princeton.edu/java/home/

[42] Rogers Candenhead. "Programación Java 7," Ed. ANAYA, 2012. http://workbench.cadenhead.org/book/java-24-hours/.