DESIGN AND PERFORMANCE EVALUATION OF WIRELESS MULTI-PROTOCOL LABEL SWITCHING (WMPLS)

By

MAURICIO ANTONIO SUBIETA BENITO

Bachelor of Science in Systems Engineering
Universidad Católica Boliviana
La Paz, Bolivia
1996

Master of Science in Telecommunications Management
Oklahoma State University
Stillwater, Oklahoma
2002

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
DOCTOR OF PHILOSOPHY
July, 2006

DESIGN AND PERFORMANCE EVALUATION

OF WIRELESS MULTI-PROTOCOL

LABEL SWITCHING

(WMPLS)

Dissertation Approved:

Dr. Jong-Moon Chung

Dissertation Adviser

Dr. Keith Teague

Dr. Charles Bunting

Dr. Nophill Park

Dr. R. G. Ramakumar

Dr. A. Gordon Emslie

Dean of the Graduate College

ACKNOWLDEGMENTS

First and foremost I would like to thank God for letting me get here. I want to thank my wife Naneida and my son Mateo for their patience, support, sacrifice and encouragement. You both are my inspiration, my will to live and my reason to exist. I love you both so much. Thank you.

Next, I would like to thank my mother Graciela, my father Gunnar and my brother, also Gunnar, for always being there and encouraging me through the years and from the distance. You will always be in my heart.

I would also like give my warmest and dearest thank you to Elia, my mother-in-law. Thank you for support, care, advice and understanding. I have learned much about life from you.

I would like to also thank my advisor, Dr. Jong-Moon Chung, for letting me through the doors of the unknown and for supporting me through my journey of learning and experimenting. Thanks for helping me realize the dream of reaching to this point in my career.

Finally, I would like to thank my friends from the ACSEL and OCLNB laboratories, from whom I've learned much and with whom I've lived good and bad times. Thanks Raj, Antonio and Hooi-Miin. Life is always much easier with friends like you.

TABLE OF CONTENTS

Chapter Page

1. INTRODUCTION 1

2. SIMULATION, MODELING AND MULTITHREADING 6

2.1 Modeling for Simulation using UML 7
2.1.1 Statechart Diagrams 8
2.1.2 Interaction and Collaboration: Sequence Diagrams 8
2.2 Finite State Machine 10
2.3 Multithreaded Simulation 11

3. PERFORMANCE EVALUATION 13

3.1 Delay and Loss Performance 15
3.1.1 Buffer Sharing and Partitioning 17
3.1.2 Policing and Marking of Packets 17
3.1.3 Packet Discard Mechanisms 20
3.1.4 Traffic Shaping 21
3.2 Queuing Theory and Queuing Disciplines 22
3.2.1 Call Admission Control 24
3.2.2 Buffer Overflow and Loss Probability 27
3.3 Complexity Analysis 30
3.3.1 Asymptotic Notation 31

4. MPLS TECHNOLOGY 34

4.1 Label Switching 35
4.2 Fundamental Concepts of MPLS 36
4.2.1 MPLS Labels 36
4.2.2 Label Switched Path (LSPs) and Label Swapping 37
4.2.3 Mapping Data to an LSP 38
4.3 MPLS Signaling Protocols 39
4.4 Traffic Engineering 40
4.5 Explicit Routes and Abstract Nodes 42
4.6 Constraint-Based Routing and Resource Reservation 43
4.7 Extensions to RSVP for LSP Tunnels (RSVP-TE) 44
4.7.1 Reuse of the RSVP Functionalities 44
4.7.2 Distributing Labels with RSVP-TE 45
4.7.3 Resource Requests and Reservation Process 48
4.7.4 Summary of RSVP-TE Messages and Objects 49

5. WIRELESS MPLS TECHNOLOGY 53

5.1 Applicability of WMPLS 54
5.2 Technical Overview of WMPLS 55
5.3 WMPLS Reference Model 56
5.4 WMPLS Architecture 57
5.5 Design Considerations 60
5.5.1 WMPLS Label Format 61
5.5.2 Extensions to RSVP-TE for WMPLS Operations 63
5.6 WMPLS State Machine 66
5.7 WMPLS Methodology and Implementation 76
5.7.1 WMPLS Initial Path Setup 78
5.7.2 LSP Establishment during Handover 80
5.7.3 WMPLS Over IMT-2000 82
5.8 WMPLS Mobile Ad Hoc Networking Support 85
5.8.1 Ad-Hoc and Mobile Ad Hoc Networks 86
5.8.2 WMPLS Ad-hoc Networking Performance Considerations 88
5.8.3 WMPLS Ad-Hoc Networking (WMPLS-AHN) Mechanisms 89

6. SCENARIO ANALYSIS AND SIMULATION 99

6.1 Complexity Analysis Metrics 99
6.1.1 Message Complexity 99
6.1.2 Computation Complexity 100
6.1.3 Storage Complexity 101
6.2 Scenario Definition and Performance Evaluation Criterion 102
6.2.1 Scenario 1: LSP Tunnel Preemption and Establishment 102
6.2.2 Scenario 2: Re-routing of LSPs 102
6.2.3 Scenario 3: Route Traversal and Identification of LSP Tunnels 103
6.3 Complexity Analysis of WMPLS 103
6.3.1 Scenario 1 – LSP Tunnel Preemption and Establishment 106
6.3.2 Scenario 2 – LSP Re-Route 112
6.3.3 Scenario 3 – Route Traversal and LSP Identification 115
6.4 Simulation and Performance Evaluation 116
6.4.1 Buffer Sharing and Packet Policing and Discarding Analysis 118
6.4.2 Buffer Overflow and Delay Bound Violation Probability Analysis 119

7. SUMMARY, FUTURE RESEARCH AND CONCLUSIONS 122

7.1 Summary 122
7.2 Future Research 123
7.3 Conclusions 124

8. REFERENCES 127

LIST OF TABLES

Table Page
Table 2.1 – State transition table 10
Table 3.1 – Traffic distribution models 14
Table 3.2 – Buffer partitioning and buffer sharing mechanisms 16
Table 3.3 – Properties of the $\Theta(n)$ and the $O(n)$ functions 33
Table 5.1 – WMPLS header flag bits 63
Table 5.2 – WMPLS header flow control and error control acknowledgement 64
Table 5.3 – Field description of the WIRELES_LABEL_REQUEST object 65
Table 5.4 – Field description of the WIRELES_LABEL_REQUEST object 66
Table 5.5 – WMPLS Finite State Machine Description Table 70
Table 5.6 – Transition Description Table for WMPLS 73
Table 6.1 – Upper Bound Values for the Neighbor Discovery Process Message Complexity Analysis of Scenario 1. 109
Table 6.2 – Upper Bound Values for the Neighbor Discovery Process Message Complexity Analysis of Scenario 1. 111
Table 6.3 – Upper bound values for the message complexity analysis of the LSP Re-Route process 113

LIST OF FIGURES

Figure Page
Figure 2.1 – Finite State Machine 10
Figure 2.2 – Message Sequence Chart 11
Figure 3.1 – The double leaky-bucket policing mechanism 18
Figure 3.2 – Leaky buckets and token buckets 21
Figure 3.3 – Overview of a queuing system 23
Figure 3.4 - Call admission control operation diagram. Packets received on the ports of the switch are analyzed based on the scheduling discipline and based on the scheduler decision the packets are accepted or dropped 25
Figure 4.1 – Position of the shim (MPLS) header 37
Figure 4.2 – Format of the MPLS shim header 37
Figure 4.3 – The RSVP-TE Label Request Objects 46
Figure 4.4 – RSVP-TE Label Object format. 46
Figure 4.5 – RSVP-TE Session Object format. 47
Figure 4.6 – RSVP-TE Sender Template Object format 47
Figure 4.7 – RSVP-TE Explicit Route Object format. 47
Figure 4.8 – RSVP-TE Record Route Object format. 48
Figure 4.9 – Formal definition of the RSVP-TE Path message 49
Figure 4.10 – Formal definition of the RSVP-TE Resv message 50
Figure 4.11 – Definition of the Flow Descriptor Object 50
Figure 4.12 – Definition of the FF Flow Descriptor Object and that contains FlowSpec subobjects 50
Figure 4.13 – Definition of the SE Flow Descriptor Object and that contains filter specs 51
Figure 4.14 – Definition of the RSVP-TE Hello message 52

Figure 5.1 – Example of the applicability of WMPLS.....54
Figure 5.2 – WMPLS reference model, with Native and Overlay models defined.....56
Figure 5.3 – WMPLS architectural model..... 58
Figure 5.4 – WMPLS header format without the Control or CRC bits 61
Figure 5.5 – WMPLS header format 3-bit sequence number control field and CRC field..... 62
Figure 5.6 – WMPLS header format 7-bit sequence number control field and CRC field..... 62
Figure 5.7 – Wireless Label Request object format. Class = 19, C_Type = WIRELESS_LABEL_REQUEST 64
Figure 5.8 – Hop Count and Sequence Number Object format. Class=HCSN, C_Type=HOP_COUNT_SEQUENCE_NUMBER.....65
Figure 5.9 – Finite State Machine definition for WMPLS 67
Figure 5.10 – Message Sequence Chart showing the node activation and path setup for the best-case scenario.....74
Figure 5.11 – Message Sequence Chart for the Connection Phase with authentication error 75
Figure 5.12 – Message Sequence Chart for the Data Transfer Phase with failure to provide required QoS, TE, Addressing and Security 76
Figure 5.13 – An example of a WMPLS network.....77
Figure 5.14 - WMPLS Initial Path Setup message exchange 80
Figure 5.15 – Path establishment during hand over 82
Figure 5.16 – Message exchange and information flow during and after hand over..... 82
Figure 5.17 – WMPLS over IMT-2000 83
Figure 5.18 – Neighbor Discovery process 91
Figure 5.19 – Initial Route Calculation..... 94
Figure 5.20 – Route recalculation procedure..... 96
Figure 5.21 – A hierarchical structure 98
Figure 6.1 – Network layout for complexity analysis 105
Figure 6.2 – Neighbor Discovery Process flowchart for Scenario 1 108

Figure 6.3 – Neighbor Discovery and Path Setup flowchart for Scenario 1.... 111
Figure 6.4 – Flowchart for the LSP Re-Route process 114
Figure 6.5 – Buffer sharing and partitioning comparison results between WMPLS and WATM 118
Figure 6.6 – Deadline violation probability comparison between WMPLS and WATM 120
Figure 6.7 – Buffer Overflow Probability comparison between WMPLS and WATM 120

1. INTRODUCTION

The developments provided by communication engineers in the wired and wireless networking areas has achieved great accomplishments over the last few years and many new services have spawned and been improved thanks to faster speeds, greater bandwidth and quality-of-service (QoS) provisions. The improvements in wired and wireless technologies, however, do not encompass a synchronized evolution, and while the advances achieved on the both these two domains have been tremendous, there is still no simple solution to the problem of interconnecting both domains in a standard and homogeneous way.

On the wired domain, fiber optic advances have led to achieve tremendous transfers speeds (up to tens of Tbps), which have also pushed the development and acceptance of new communication protocols capable of supporting these speeds such as the Multi-Protocol Label Switching (MPLS). On the wireless side, the advances have led to improvements on medium access control mechanisms and also transfer speeds especially for wireless Local Area Network (LAN) technologies (such as 802.11x), with emphasis on secure and encrypted data communications. Despite the tremendous research efforts focused on wireless LAN technologies, a major paradigm change has not been yet accomplished in order to transform current one-hop WLANs into multi-hop WLANs that can either be set up in infrastructure, ad-hoc, or mobile ad-hoc modes, which is clearly corroborated in [42] where it is stated that all existing protocols from the application layer to transport, network MAC, and physical layers need to be enhanced or reinvented, because current wireless protocols lack scalability, suffer from throughput performance issues, and most importantly, are entirely heterogeneous to protocols of wired networks.

MPLS is the culmination of a continuous development process led by several companies in the networking field that overcomes IP networking problems such as scalability, network hotspots, limited functionality, and most of all lack of interoperability. Toshiba derived one of the earliest IP-controlled switching technologies named Cell Switching Router (CSR) that used a proprietary protocol to forward cells. Ipsilon, which was acquired by Nokia, invented IP Switching in which ATM hardware was programmed with switching instruction based on a proprietary protocol to forward the packets. Tag switching, proposed by Cisco Systems, was not limited to ATM switches, but was a more generalized approach to label switching in which label information was distributed using the Tag Distribution Protocol (TDP), which is still being used for MPLS label distribution. Aggregate Route-Based IP Switching (ARIS), supported by IBM, was a control-based switching technology like TDP, although it was also general in its applicability, it was mostly focused on ATM hardware. Lucent Technologies also provided their own development initiative based on the IP Navigator platform that was acquired from Ascend (formerly Cascade).

However, even the latest industrial efforts in order to deploy fully standardized MPLS, carried by major network equipment designers, such as Cisco, still rely on a paradigm that provides overlay models for currently deployed layer-2 technologies such as Frame Relay and ATM. One of this efforts backed by Cisco is known as Any Transport over MPLS (AToM) [6].

Recent service demands in wireless communications have significantly changed, where the traditional focus was commonly limited to voice channels over wireless point-to-point connections between the base station and the wireless terminal/phone, thus not requiring any complex routing or switching networking topologies. A large number of the current wireless service applications require broadband data communications as well as advanced wireless networking. None of the current wireless protocols are suitable to interface and support the service level agreement (SLA) requirements of MPLS, or Generalized MPLS (GMPLS)

networks regarding differentiated services (DiffServ), QoS and traffic engineering (TE) networking features.

A major milestone achieved from this research work is the design of the Wireless MPLS (WMPLS) protocol, which is a novel wireless networking topology that can used to provide DiffServ, QoS, and TE in wireless networks. WMPLS technology was developed as a homogeneous protocol with MPLS and GMPLS regarding the protocol architecture and networking topology including the applied signaling protocols, which are the Label Distribution Protocol (LDP) [5] and Resource Reservation Protocol with Traffic Engineering enhancements (RSVP-TE) [11]. Additionally, interoperability issues with the current mobile wireless technologies and applications to mobile ad-hoc networking (MANET) were considered in the modeling of WMPLS [10]. WMPLS is applicable to all existing ad hoc and mobile ad hoc networks. In addition, since WMPLS is a layer 2 protocol it can be applied in Mobile IP and related networking topologies as the underlying switching mechanism.

The main objective of the research presented in the upcoming chapters is focused on providing definitions, extensions, and recommendations to the WMPLS initiative as first devised in [7], designed to provide a single and homogeneous infrastructure that will close the gap between wired and wireless MPLS domains. By designing and enabling a seamless and transparent protocol and methodology based on current MPLS technologies, and with a minimum amount of overhead and processing time, network connectivity will be allowed to expand without boundaries, allowing traffic transport over hybrid networks with QoS and traffic engineering guarantees to be achieved.

The specific objectives and contributions that this research work presents include:

- Fully integrated QoS and TE parameters that can support a hybrid wired

and wireless communication network that does not require manual intervention for setup or management, including one-hop and multi-hop wireless networks.

- Full support of MPLS and GMPLS capabilities in wireless networks, in either infrastructure or ad-hoc modes, involving one-hop or multi-hop routing capabilities, with full support for synchronous or asynchronous data connections, including flow/error control mechanisms to enhance hop-by-hop reliability.
- Transparent link setup and signaling message exchange for upper layers (TCP/IP layer 4 and above), with full compatibility and interaction with layer 3 protocols (IP, ICMP, and IGMP).
- Standardized platform for specific types of transport networks that will enable transparent interaction between third and fourth generation wireless communication systems (i.e., W-CDMA, CDMA2000, DMB), and current high-speed optical networking (i.e., 10GE (IEEE 802.3z), SONET).
- Extensions for multicast real-time traffic, with QoS guarantees and proper content delivery control [7].
- Mobility support for ad-hoc networks, with resilient location and connection management procedures in order to maintain QoS, Grade-of-Service (GoS) and Class-of-Service (CoS) parameters.

The motives for extending MPLS into the wireless domain, thus giving origin to WMPLS, are based on four circumstances. The first circumstance is focused on the need of high-speed and high-quality broadband wireless data communications, and to satisfy this growing demand for diverse services and quality of data, wireless networks need to fully and homogenously integrate with wired networks, and then the QoS, GoS, and CoS features requested of the wide area network or Internet can be supported through the wireless network as well. The second circumstance is based on the fact that MPLS and GMPLS technologies have achieved a level of almost mature development, and are about ready to be massively deployed throughout carrier networks, and intuitively the

wireless version of these technologies resulted in the motive to develop WMPLS. The third circumstance is focused on the need of a protocol that can support end-to-end and hop-by-hop connection-oriented or connectionless mobile communications, supporting infrastructure and ad hoc modes. The fourth and final circumstance relies on the need and application of wireless differentiated services in order to provide homogeneous traffic services for hybrid wired/wireless networks.

As mentioned above, one of the main efforts provided in industry has been lead by Cisco with their AToM infrastructure. AToM works by encapsulating Layer 2 frames and transporting them across a MPLS network, supporting Layer 2 services such as ATM virtual private networks (VPNs), while aggregating and integrating transport technologies and taking advantage of proven MPLS QoS and scalability. Using AToM, a Layer 2 frame is encapsulated at the provider edge router. The frame is then transported across the IP/MPLS backbone. Upon reaching the provider edge router on the other side of the backbone, it is un-encapsulated and sent to its destination as a Layer 2 frame. However, there are no provisions to include the wireless domain in any of the initiatives being conducted.

This document is organized as follows. Chapter 2 covers the main characteristics of the MPLS technology. Chapter 3 presents a brief introduction to modeling and simulation theory based on multithreaded models. Chapter 4 presents the MPLS technology. Chapter 5 covers the structural design of WMPLS, which is the main component of this research work. Chapter 6 focuses on the performance evaluation analysis of the presented design through a simulation. Chapter 7 presents a summary of the current work, provides an insight of future work and the conclusions.

2. SIMULATION, MODELING AND MULTITHREADING

Simulating any model or real-life event involves generating proper statistical and stochastic mechanisms of the model and then observing the resultant flow of the model over time. Depending on the purpose of the simulation, there are certain parameters that need to be specified and determined in such a manner that they represent the characteristics that can be found on the actual event to be replicated. However, because the evolution of the model over time involves a complex logical structure and interaction of its elements, it is not always apparent how to keep track of this evolution so as to determine these quantities of interest [20]. Modern research on system modeling and control is mainly focused on either continuous variable systems (CVS) or discrete event systems (DES) [21]. The former are modeled and analyzed by differential equations in order to capture the physical dynamic behavior, while the latter have been described on several different frameworks to capture the logical and sequential behavior, such as finite state automata and others. Both the CVS and DES are developed to reduce the complexity for modeling real systems, however, most real-life systems show a behavior that combines both the time-driven and event-driven dynamics together as a hybrid dynamic system (HDS), also known as hybrid systems.

Two categories for modeling frameworks have been proposed for HDS. The first one relies on the event-driven dynamics that are included in traditional CVS models; while the other one relies on the time-driven dynamics that are included in the DES models. Given the fact that the state of the DES is usually artificial and not real, DES is not the most appropriate model to be described by traditional CVS models. Merging the discrete event behavior with the continuous model creates an even more complex model that is very difficult to realize. Trying to extend the event-driven model to include time-driven characteristics would also

yield a very complex model that might lead to a state-space explosion problem [23]. Therefore, when discrete logic modeling and continuous behavior characteristics are somehow interrelated in a hybrid system, both modeling techniques (CVS and DES) evolve into an unnatural and complex model. Additionally DES modeling with complex time dynamics significantly affects the performance of the simulation.

Recently multi-threading and multi-tasking capabilities have been extended from a software-only environment into the design of the microprocessors that can fully take advantage of this methodology in order to speed up calculations, increase the number of floating-point and numerical operations, and improve the performance of the operating system by providing native support for various tasks to be running simultaneously.

2.1 Modeling for Simulation using UML

Simulating a real world phenomenon requires the definition of a model that will include all the necessary components, and in which the parameters can be manipulated in order to provide different results that will help in the final analysis. Specialized tools have been developed to make this process simpler, while capturing as much detailed information as possible, and one of the most commonly used is the Unified Modeling Language (UML).

UML is an object-oriented methodology used for software engineering that preserves the properties of convergence and clarity in the design of models, allowing them to be portable and compatible. UML is inherently a discrete language and it emphasizes the discrete representation of the dynamic behavior that real systems present. UML uses a state machine representation as the primary means of capturing complex dynamic behavior based on statecharts [22]. UML in general consists of around nine diagrams that correspond to the standard static and dynamic aspects. The design typically requires a static

diagram, such as the class diagram, and a dynamic diagram such as the statechart.

2.1.1 Statechart Diagrams

Statecharts are a type of behavioral model, sometimes are referred as timing models, that are used to represent the logical order and conditions in which the events occur. These diagrams model the behavior of an object or an entity (known in UML as a *class* [56]). State charts are made up of two basic elements: states and transitions. These states and transitions describe the behavior of an instance of a class over logical time (that for this case represents real time, although UML is not a tool to represent real time events). States may show when an activity is taking place, or show that an event is waiting to happen. A transition shows how to change from one state to another, or sometimes to the same state but under a different condition.

There are three types of states: Normal, Start, and End. Each transition may have zero or more conditions which explain how a transition may be crossed. A condition is a Boolean condition that will usually relate to the value of an attribute. A transition may also have zero or more actions associated. An action is defined as an activity that takes very little time to be executed. This time should be negligible when being analyzed in the context of the whole algorithm. Finally, a Transition may have zero or more Events associated. An even is the passing of a message usually from one object to another. There are two types of events: Send and Receive.

2.1.2 Interaction and Collaboration: Sequence Diagrams

Interaction and collaboration diagrams model interactions between instances of classes or objects. The main focus of a collaboration diagram is to show the operational structure of a system, which allows the definition of scenarios that highlight pertinent objects in a particular situation and ignore all others.

Effectively the collaboration diagram shows message flows between objects which are displayed in terms of their physical organization.

Message sequence diagrams or charts (MSC) show the high level behavior of a system from the logical timing point of view. These diagrams are composed of three basic elements: Objects, links and messages. The objects have the characteristic of showing a life line, i.e. the order of the events in a logical manner. This time line is present whenever the object is active, and it is graphically represented as a vertical line with logical time traveling down the line. The objects for the sequence diagram are shown going horizontally across the page. Message sequence charts have a rigid structure for the layout of the model, putting emphasis on the order of the objects and their logical timing sequence. They must be shown staggered down the diagram dependent on when they are created, however for the purpose of this research, they will all be presented across the top of the diagram.

In order to model the simulation of the WMPLS protocol, the extensions for hybrid dynamic systems proposed in [21] will be used. In this work the proposed modeling approach results in a natural representation for hybrid dynamic systems and provides compact and clear specifications of complex interdependencies between discrete and continuous behaviors. This methodology has been chosen because it relies mainly on the definition of a state machine representation of the problem, as explained in Section 2.2. Typically, engineering protocols rely solely in the proper definition of a state-transition diagram that can be generalized as a state machine. Even though UML is a modeling language that does not have any direct relationship with any programming language, several tools allow code generation for C, C++, Java and other programming languages, simplifying the process of translating the model structure into working code. Several new extensions allow the interaction between specialized programming environments, such as MATLAB and Simulink.

2.2 Finite State Machine

A finite state machine (FSM) or state machine is a model of a system interaction that is composed of states, transitions and actions. A state stores information about the past and it reflects the input changes from the system start to the present moment. A transition indicates a change in a given state and is described by a condition that needs to be fulfilled in order to enable the transition. An action is a description of an activity that is to be performed at a given moment.

A FSM can be represented using a the UMLS representation of a statechart (or state transition diagram) as shown in Figure 2.1. Additionally, several state transition table types are used. The most common representation is shown in Table 2.1, in which the combination of the current state (B) and condition (Y) shows the next state (C).

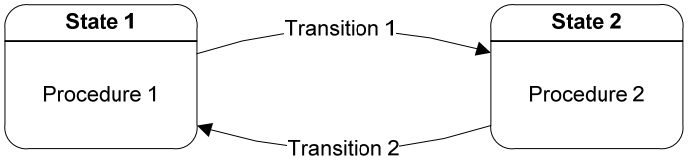


Figure 2.1 – Finite State Machine

Current State / Condition	State A	State B	State C
Condition X	...	...	...
Condition Y	...	State C	...
Condition Z	...	...	...

Table 2.1 – State transition table

The specification of a system based on a finite state machine diagram and the state transition table gives rise to the Message Sequence Diagrams or Charts (MSCs). An MSC is a graphical and textual language for the description and specification of the interactions between system components. The main area of application for MSCs is as an overview specification of the communication

behavior of real-time systems, in particular telecommunication switching systems and protocol engineering [55]. Message Sequence Charts may be used for requirement specification, simulation and validation, test-case specification and documentation of real-time systems. An example of a MSC is provided in Figure 2.2.

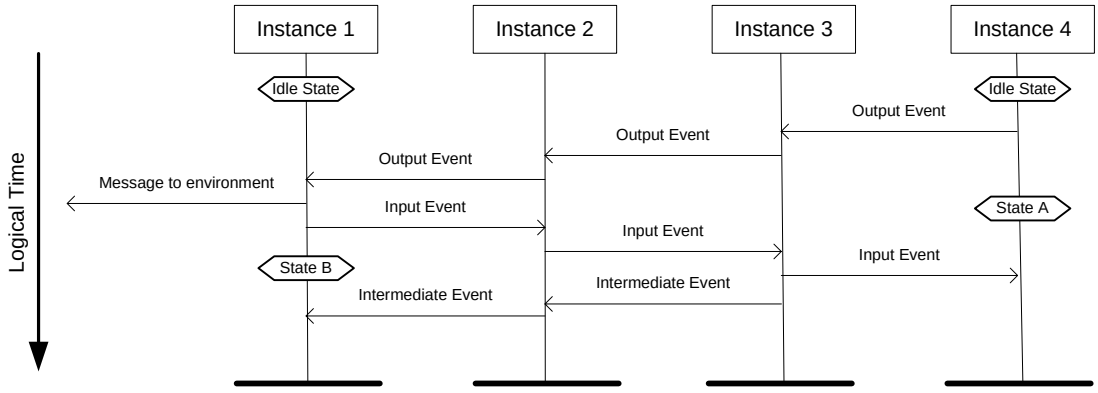


Figure 2.2 – Message Sequence Chart

2.3 Multithreaded Simulation

A real-time system is comprised of many events that occur simultaneously, and thus they need to be processed concurrently, and the interactions between these processes adhere to certain timing constraints [23]. Simulating the behavior of the real-time system then requires to mimic the concurrent interactions as closely as possible in order to obtain results that include the interactions between process that are simultaneously being executed.

Multitasking is the ability to have more than one program working at what seems like the same time. Multitasking can be done in two ways, depending on whether the processes are interrupted without consulting with them first or whether processes are only interrupted when they are willing to yield control. The former is called preemptive multitasking; the latter is called cooperative (or non-preemptive) multitasking. Although harder to implement, preemptive multitasking is much more effective. With cooperative multitasking, a process

that gets corrupted can end up using all the resources and disable the rest of the processes indefinitely.

Multithreading extends the concept of multitasking by abstracting the process in a more detailed manner: individual processes will appear to do multiple tasks at the same time. Each task is usually called a thread. Processes that can run more than one thread at once are said to be multithreaded. The essential difference between threads and processes relies on that while each process has a complete set of its own variables; threads share the same memory space that contains the data, which constitutes a potential point of failure if not properly managed. However, shared variables make communication between threads more efficient and easier to program than inter-process communication (IPC). Moreover, on some operating systems, threads are simpler to execute than processes because it takes less overhead to create and destroy individual threads than it does to launch new processes.

Multithreaded simulation provides the mechanism to successfully achieve hybrid dynamic system simulation because it enables the simulation process to perform concurrent tasks that are executed independently of each other. Specifically, a true representation of a state machine, based on a state-transition model, can be achieved by means of a multithreaded program.

3. PERFORMANCE EVALUATION

For the design and implementation of the Wireless Multiprotocol Label Switching (WMPLS) protocol, it is necessary to find out whether the protocol design will be used to the best effect, and the analysis of its performance is needed for this purpose. The methods for performance evaluation fall under two types of categories: measurement and predictive techniques. The latter technique involves mathematical analysis and simulations, which results in a model that can be later compared to other techniques used as reference, and point out the benefits and drawbacks of the proposed design.

It is important to realize that there are certain traffic issues that are the key components in obtaining the proper performance evaluation of a system, and these relate specifically to the protocol definitions themselves, including the lack of certain features within the protocol itself and its design.

Measurement methods need the availability of real network deployments for experimentation. The advantage of this approach is the capability of performing direct measurement of the performance of any protocol without losing any details of its operation. Unfortunately, there are some limitations of this approach, primarily because an experimental environment cannot include real-world events, and the availability of resources as well as the dimensions of the experiment is limited by budget and time.

The predictive technique involves performing analysis of scenarios that require simulations to be run in order to obtain data for further and more detailed analysis. The main factors to consider when comparing analysis and simulation are the accuracy of the results, the time to produce these results, and the cost

involved in using this approach. An advantage of this method is that analytical solutions can be used relatively quickly; however, detailed and specific characteristics that belong to the traffic can be overseen and not included, thus providing limited or flawed results. For this reason, analysis is often used to produce an approximation of a real-world system, with fast and cost-effective results.

Model Name	***Model Description***	***Model Parameters***
Negative Exponential Distribution	Used for inter-arrival time definition, service and packet behavior [36]	t – time λ – rate of arrival, or rate of service
Geometric Distribution	Used for inter-arrival time definition, service and packet behavior [36]	k – time slots p – probability of arrival or end of service in a time slot
Poisson Distribution	Used to determine number of arrivals or the amount of calls received [36]	T – time k – number of arrivals λ – rate of arrival
Binomial Distribution	Used to determine number of arrivals or the amount of calls received [36]	k – number of arrivals p – arrival probability N – number of timeslots or number of inputs
Batch Distribution	Used to determine the number of arrival [36]	k – number of arrivals p – probability of existence of a batch of arrivals in a time slot $b(k)$ – probability of existence of k arrivals in a batch M – maximum number of arrivals in a batch
ON-OFF two-state	Used to determine rate of arrivals [36]	R – rate of arrivals $E[on]$ - mean number of arrivals in the ON state C – service rate $E[off]$ – mean number of time units in the OFF state
Pareto Distribution	Used to determine the number of arrivals and the amount of calls receive [36]	δ – minimum amount of received calls x – number of arrivals, or amount of received calls α – power law decay

Table 3.1 – Traffic distribution models

There are several key components that need be analyzed in order to provide a proper framework for either the analytical or experimental approaches. In the following subsections, the most important parameters used in evaluating the performance of WMPLS compared to other protocols are defined. However, and

given that WMPLS is devised from a newer technology, it is necessary to clarify that only the comparable results are explained and analyzed in detail. The newer technological characteristics and advantages of WMPLS are mentioned and briefly explained, as a thorough analysis and evaluation of these capabilities is beyond the scope of this research.

3.1 Delay and Loss Performance

For packet-based (and also for cell-based) networks, the fundamental component that affects its performance is the queuing delay experienced by packets traversing the buffers within the devices that comprise the network. The queuing behavior implies that the packets experience variations in the delay through a buffer and also, if this delay becomes too large, it experiences packet loss.

In its simplest form a buffer has a fixed service rate, a finite capacity for temporary storage, and a first-in-first-out (FIFO) discipline of service; the queuing behavior depends on the type of traffic being multiplexed through that device. There are pre-defined models that help in the description of this behavior, shown in Table 3.1.

However, these disciplines do not allow different performance requirements to be guaranteed by the network. For example, for the best-effort IP model, all traffic suffers similar delay and loss, and in ATM the most stringent requirement limits the admissible load [36]. For the analysis of MPLS and WMPLS, a more robust and complete model that copes with multi-service requirements and provides differentiated performance measures is needed [37].

Model:	Buffer-space Partitioning
Description:	Used to allocate space to virtual buffer based on performance requirements and decay rate
Parameters:	X – total buffer space available X_i – buffer space for virtual queue i S_i – scaling factor for the overflow probability for virtual queue i dr_i – decay rate for virtual queue i (obtained using queuing analysis)
Equations:	$S_1 \cdot dr_1^{X_1} = S_2 \cdot dr_2^{X_2} = \cdots = S_j \cdot dr_j^{X_j} = \cdots = S_V \cdot dr_V^{X_V}$ $X = \sum_{j=1}^{V} X_j$ *(3.1)* $X_i = \frac{X + \sum_{j=1}^{V} \left(\frac{\log(S_j)}{\log(dr_j)} \right)}{\log(dr_i) \cdot \sum_{j=1}^{V} \left(\frac{1}{\log(dr_j)} \right)} - \frac{\log(S_i)}{\log(dr_i)}$
Model:	Shared-buffer
Description:	Assesses the performance improvement when sharing buffer space across various output buffers using the decay rate
Parameters:	d_r – decay rate per individual buffer $p(k)$ – probability of individual buffer contains k packets $P_N(k)$ – probability that the shared space has k packets $Q_N(k)$ – shared buffer overflow probability
Equations:	$p(k) = (1 - d_r) \cdot (d_r)^k$ $P_N(k) = \sum_{j=0}^{k} P_{N-1}(j) \cdot P_1(k-j)$ *(3.2)* $Q_N(k) = 1 - \sum_{j=0}^{k} P_N(j)$

Table 3.2 – Buffer partitioning and buffer sharing mechanisms

The solution to this requirement is to manage the buffer, both on the entry and exit points, involving policies for partitioning (classifying) and sharing the buffer space and server capacity (e.g. per flow queuing), policing, packet discard mechanisms, and queue scheduling (such as precedence queuing, weighted fair queuing, earliest deadline first, etc).

3.1.1 Buffer Sharing and Partitioning

With queuing disciplines such as per-flow queuing, weighted fair queuing, earliest deadline first, each virtual buffer can be modeled as having its own server capacity and buffer space, and any of the analysis methods for FIFO queues can be applied, as appropriate to the multiplexing scenario and traffic sources [36]. Typically, these models provide a decay rate for each virtual queue, which in turn will be used along with the performance requirement to assess the partitioning of the buffer space [36].

The primary benefit of partitioning is to maintain different performance guarantees for a variety of service types sharing a transmission line. The disadvantage of this procedure is that it is not optimal when considering the overall loss situation at the end of a line: the loss of a packet from a full virtual queue may not be necessary if the buffer space is shared, especially when the buffer space can be shared across multiple lines [36]. Table 3.2 shows the models for the partitioning and sharing techniques.

3.1.2 Policing and Marking of Packets

Any traffic flow that traverses through a network device can be configured to expect data to arrive at a certain maximum rate defined by a service agreement. The process of verifying that the arriving data stream complies with the predefined rate is called *policing* or metering. If a packet violates the committed rate, it can be either demoted (lowering its output priority) or discarded. The packet can also be marked so that downstream devices recognize the marked packets, which will receive a higher discard probability than the rest of the traffic [44].

Several algorithms are used for policing. The generic cell-rate algorithm used in ATM marks cells as either compliant or noncompliant. Noncompliant cells have a cell-loss priority bit set to 1 so that they have higher discard probability downstream. Cell-loss prioritization involves two leaky buckets, as shown in

Figure 3.1. The first measures the cell's peak rate and marks those cells that exceed the threshold value. The succeeding bucket measures the average rate, marks those exceeding it and passes unmodified cells below the average rate [44].

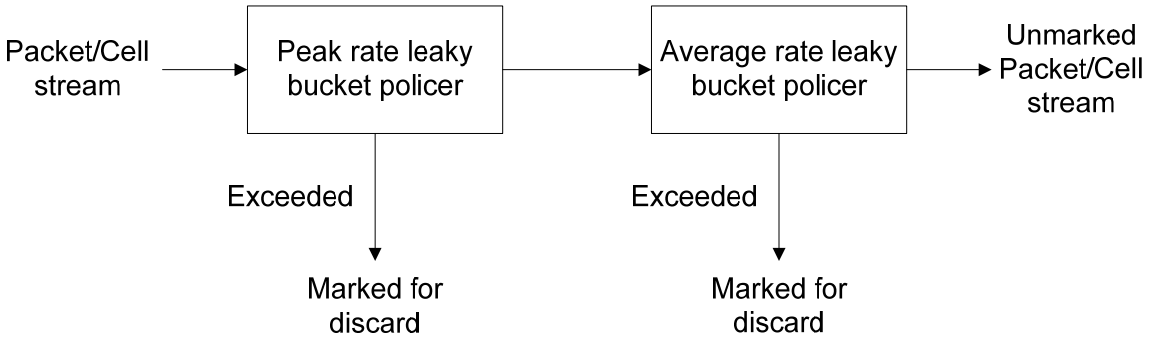


Figure 3.1 – The double leaky-bucket policing mechanism

In the case of the IP protocol, policing has been achieved through the Integrated Services (IntServ) and Differentiated Services (DiffServ) mechanisms, the single-rate and two-rate three-color markers-known as srTCM and trTCM, respectively, have been designed as the policing mechanism [44]. Each of those schemes uses two token buckets to differentiate normal and excess traffic. For trTCM, four parameters are specified: a peak information rate, a peak burst size, the committed information rate (CIR), and the committed burst size. The two token buckets are used to measure the peak and committed rates and burst sizes. Packets exceeding peak information rate and peak burst size are marked red, those exceeding CIR and committed burst size are marked yellow; otherwise packets are marked green. For srTCM, three parameters are specified for each connection: the CIR, the committed burst size and the excess burst size. SrTCM's implementation is similar to that of trTCM because it also uses two token buckets, except that the rate difference between peak and committed traffic is monitored for violation rather than for the absolute value.

WMPLS uses the policy and marking of traffic defined by DiffServ, in which the traffic characteristics are extended by establishing a SLA between an upstream network and a downstream domain. The SLA specifies the packet classification

and marking rules and may also specify traffic profiles and actions to traffic streams. The packet classification policy identifies the subset of traffic that may receive a differentiated service by being conditioned and/or mapped to one or more behavior aggregates within the network. Traffic conditioning performs metering, shaping, policing and marking to ensure that the traffic entering the network conforms to the rules terms in the SLA and in accordance with the network's service provisioning policy.

3.1.2.1 Classifiers

Traffic classifiers select packets in a traffic stream based on the content of some portion of the packet header. Classifiers are used to redirect packets matching some criteria to an element of a traffic conditioner for further processing.
There are two types of classifiers: the Behavior Aggregate (BA) classifier, which classifies packets based on the DiffServ code point only; and the Multi-field (MF) classifier that selects packets based on the value of a combination of one or more header fields, such as source address, destination address, DiffServ field, protocol ID, source port and destination port numbers, and other information such as incoming interface.

3.1.2.2 Traffic Profiles

Traffic profiles specify the temporal properties of a traffic flow selected by a classifier and provide mechanisms for determining if a particular packet belongs or not to a given profile. Different conditioning actions may be applied to the packets that fit different profiles, including different accounting actions

3.1.2.3 Traffic Conditioners

A traffic conditioner may contain various elements such as a meter, a marker, a shaper, and dropper. A traffic stream is selected by a classifier, which redirects the packets to a logical instance of a traffic conditioner where a meter is used to

measure the traffic stream against a given traffic profile. The state of the meter with respect to a particular packet is used to take a marking, dropping, or shaping action.

Traffic meters measure the temporal properties of the stream of packets selected by a classifier against a traffic profile, and it passes state information to other conditioning functions to trigger a particular action for each packet. Packet markers set the DS field of a packet to a particular codepoint, adding the marked packet to a particular DS behavior aggregate. Traffic shapers delay some or all of the packets in a traffic stream in order to bring the stream into compliance with a traffic profile. A shaper usually has a finite-size buffer, and packets may be discarded if there is not sufficient buffer space to hold the delayed packets. Packet droppers discard some or all of the packets in the traffic stream in order to bring the stream into compliance with a traffic profile.

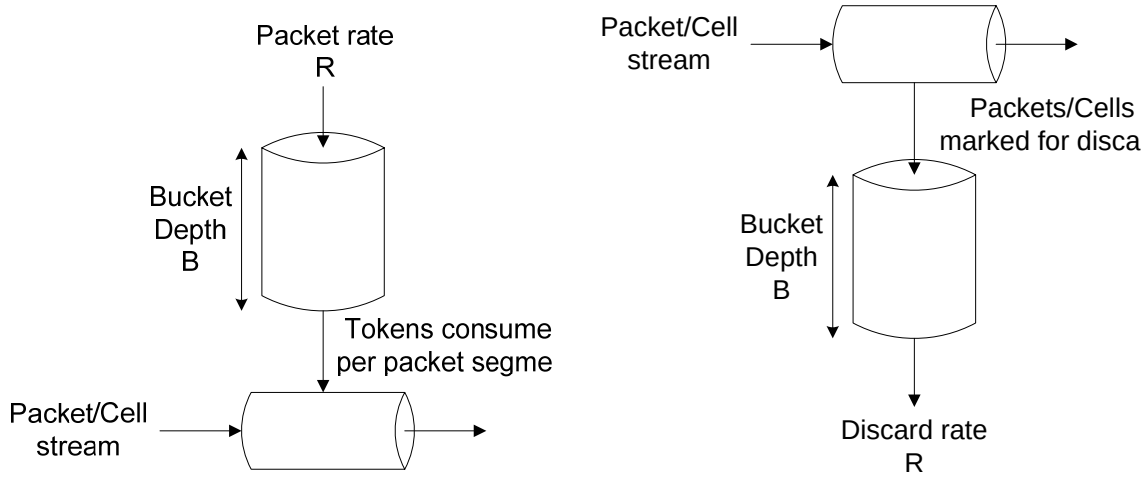
3.1.3 Packet Discard Mechanisms

Any packet that violates the bandwidth requirements defined in the policing mechanism must be discarded and the primary reason for discarding any packet is to avoid congestion at any node in the network. Once a packet flow has undergone the policing mechanism and was found to be in violation, it can be marked for immediate or deferred discarding.

A common practice that network devices employ is random packet discard that occurs when the input queues fill up. A popular packet-discard mechanism is random early detection (RED) and its variation, weighted RED (WRED). In RED, whenever the queue begins to fill up, packets are discarded based on the calculation of a probability of average queue occupancies. In the case of WRED, the packets are randomly discarded, but they are priority weighted so that higher-priority packets are less likely to be discarded [36].

3.1.4 Traffic Shaping

Traffic shaping is the mechanism that regulates outgoing traffic in order to comply with SLAs and also reduces congestion in downstream nodes. Traffic shaping can be used with the queuing disciplines discussed previously, for example, a rate shaper located after a WFQ provides one of the most effective mechanisms for sharing pre-allocated bandwidth among several traffic flows [44]. There are two popular traffic-shaping mechanisms, which are based on token buckets and leaky buckets, as shown in (Figure 3.2). These two techniques are similar in that both use buckets of tokens to control the flow of traffic through a queue. The primary difference lies in the fact that with the token-bucket scheme, tokens are accumulated at a constant rate R into a bucket with depth B, while in the leaky-bucket scheme tokens are leaked at a constant rate R, from a bucket with depth B.



(a) Token Bucket (b) Leaky Bucket

Figure 3.2 – Leaky buckets and token buckets

Leaky buckets work well with cell-switching technologies, while token buckets work well with packet-switching technologies. Since cells are fixed in size, their transmission can proceed without having to wait for all cells in the PDU. On the other hand, packets are sent when enough tokens are accumulated to account for all segments of a packet; before enough tokens are available for all packet

segments, the packet is kept in the queue. A leaky-bucket shaper provides a rather constant and smooth flow with a rate *R*. Since all cells are of the same size, the average rate equals the peak rate R, and burst size is constant. A token-bucket shaper, however, rate-limits the outgoing flow down to an average rate *R*, and it permits bursty traffic to proceed while limiting the maximum burst size to the depth of the token bucket. Token and leaky buckets can also be used for traffic policing by passively observing and marking out-of-profile packets. A cell stream is out-of-profile if the leaky bucket overflows, and a packet stream is out of profile if the token buckets underflow.

ATM currently supports only a number of class-of-service queue priorities, but MPLS can be deployed together with DiffServ in such a way that it can provide the mechanisms upon which DiffServ per-hop behavior, such as guaranteed traffic forwarding, can be enforced by the local nodes belonging to the paths.

The scalability of a technology is inversely proportional to the amount of state information that packets must maintain and the control plane of the network devices must keep track of. The IP protocol is highly scalable because of this reason. However, this protocol does not guarantee bandwidth and delay bounds, and the only means available to provide traffic management mechanisms are provided by higher-level protocols, such as congestion control through TCP feedback. MPLS provides stateless forwarding of packets once the signaling protocols have provided the LSP setup, which makes the operations of the protocol more efficient than ATM (which also proceeds in a similar way, but is less efficient due to the fixed and small size of the cells).

3.2 Queuing Theory and Queuing Disciplines

A fundamental part of the performance evaluation process for network and protocol design is the queuing process, due to the fact that every network device and component uses buffers due to the contention for limited networking resources (bandwidth or processing capabilities). The concept of a queue that will

be used in this document is then presented as a mathematical expression of the idea of resource contention. As shown in Figure 3.3, arrivals coming into a system need a certain amount of service for which they can wait in a storage area called buffer or queue, and after a predefined maximum amount of time, the arrivals are serviced and leave the system.

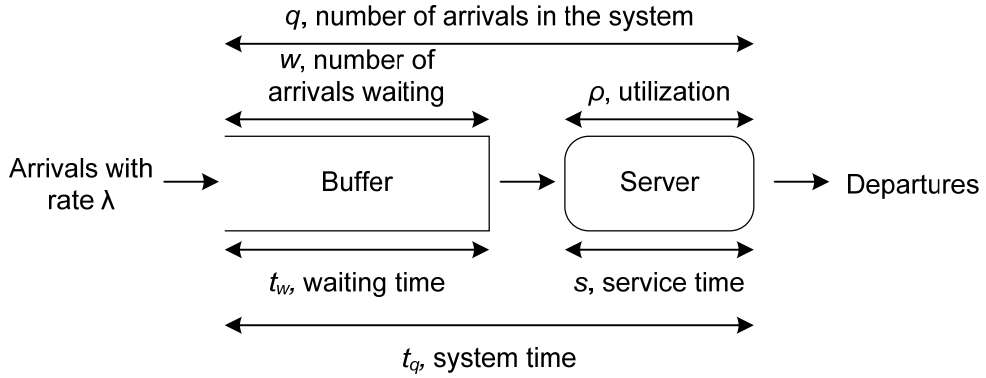


Figure 3.3 – Overview of a queuing system

Any queuing system is described by the characteristics of the arrival process, the service pattern, the number of service channels, and the system capacity. The arrival characteristics are defined as the input of a system and it is usually referred as the average number of arrivals per unit of time, or by the average time between arrivals (also known as mean inter-arrival time). Older analysis techniques assume a deterministic arrival input process in which every *t* time units an arrival occurs, whereas newer analysis techniques assume stochastic characteristics of the arrival process in which the probability distribution function is used to clearly characterize the arrivals for the analysis.

Queuing also covers the mechanisms for transmitting outgoing packets from the same queues according to certain queuing disciplines. There are various mechanisms and algorithms that are used for selecting packets from queues for transmission: FIFO queuing, priority queuing, weighted round-robin queuing, weighted fair queuing (WFQ) also known as generalized processor sharing (GPS), and earliest deadline first (EDF) are some of the current and better approaches.

Priority queuing uses a set of FIFO queues, each for a different priority level, and packets with higher priority are always chosen before packets with lower priority. This mechanism is easy to implement, but it can cause starvation of low-priority connections if there is a large number of connections of higher priorities, thus more sophisticated queuing schemes should be used.

The weighted round robin mechanism assigns each incoming packet to a different queue that is in turn associated with a different weight. A round-robin scheduler iterates through the different queues and sends the packets. The frequency of the packet transmission is determined by the weight of the queue. This technique is efficient with fixed-size cells like ATM but not with variable-size packets because the amount of bandwidth required transmitting a cell is the same. In the WFQ discipline each packet transmission may take a different amount of bandwidth, depending on the size of the packet. It has been demonstrated that traffic sources that apply WFQ that are rate limited provide end-to-end delay bounds, a key parameter for quality-of-service (QoS). In WFQ every traffic flow receives its fair amount of bandwidth and any excess unused bandwidth is distributed to each flow according to its weight to provide for peak bandwidth.

3.2.1 Call Admission Control

To provide end-to-end performance guarantees, the traffic SLA must be established, and resources must be reserved. Traffic flows then need to be monitored to ensure compliance with the contract. At the core of these functions lays the assessment of performance resulting from traffic flowing over the network resources, which rely on various forms of queuing analysis. The admission control function assesses whether a new connection or traffic flow can or cannot be admitted. This mechanism takes into consideration the contracts that the network is currently supporting and the available resources for any new traffic flows. In order to do this the admission controller must be aware of the structure and configuration of the buffers, which includes the scheduling

discipline, packet discard mechanism, traffic shaping, and partitioning mechanisms, and based on them it should provide an appropriate assessment of whether the requested resources and performance requirements can be met.

The main function of a packet scheduler is to classify traffic and provide enough bandwidth sharing capabilities such that heterogeneous traffic types, with different services classes, are provided with the resources that meet the required QoS constraints, as shown in Figure 3.4. It is for this reason that the packet scheduler is a critical network component..

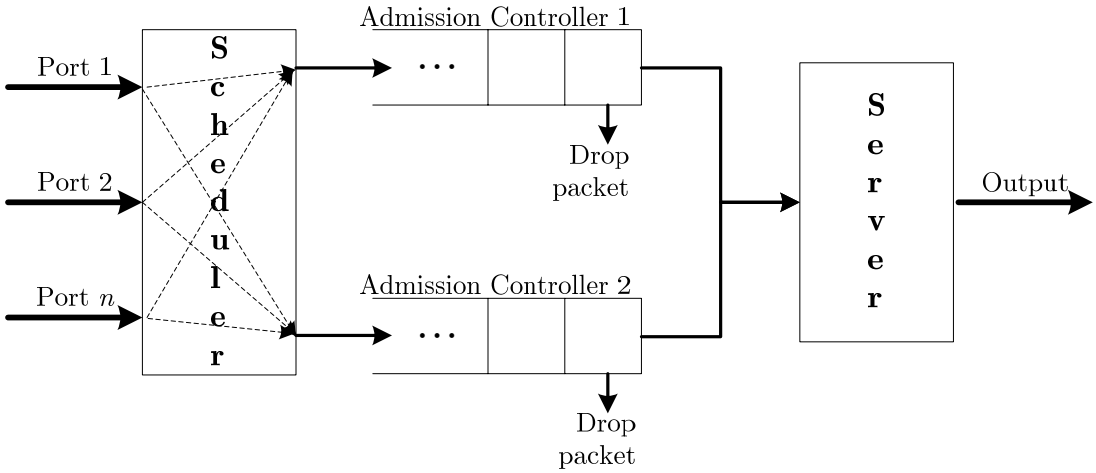


Figure 3.4 - Call admission control operation diagram. Packets received on the ports of the switch are analyzed based on the scheduling discipline and based on the scheduler decision the packets are accepted or dropped

The primary objective of the call admission control mechanism (CAC) is to decide whether or not to accept a connection in order to prevent network degradation or congestion. There are two types of CAC mechanisms: parameter-based and measurement-based. The former CAC mechanism requires explicit knowledge of traffic parameters, while the latter employs real-time measurements in order to find traffic characteristics. Typically the parameter-based algorithm is known to be inaccurate and ineffective due to the difficult task to fully characterize the network traffic based on a number of finite parameters, especially for self-similar

traffic. These limitations make the measurement-based approach the choice for the network performance analysis.

The CAC mechanism is defined in the ITU Recommendation I.371 [36] [45], and it specifies the actions taken by the network during the call set-up phase (or during call renegotiation) to establish whether a connection can be accepted or should be rejected, which is defined in a very similar form in the ATM Forum Traffic Management Specification [46].

The CAC algorithm needs to know the source traffic characteristics and the required performance in order to determine whether the connection can be accepted or not, and if accepted, the network resources to allocate. However, neither of the recommendations mentioned above specify any particular CAC algorithm, they only define the possible CAC policies, and the implementation is left to the network operator to choose or even implement. Nevertheless, the ITU recommendation mentions three operating principles, which serve as a starting point to define a proper CAC discipline:

a) Multiplexing of constant bit rate (CBR) streams
b) Rate-envelope multiplexing
c) Rate-sharing statistical multiplexing

The first operating principle corresponds to allocation of resources under peak rate traffic circumstances which is the deterministic bit rate transfer capability and it deals with packet and cell scale queuing behavior, which arises only as a result of source multiplexing [36], which occurs in networking devices continually (either by different traffic sources of different traffic types coming from the same source). The second and third operating principles allow for the statistical multiplexing of variable bit-rate streams and are two approaches that provide statistical bit rate transfer capability.

The rate-envelope multiplexing is also known as the burst-scale loss factor, in which traffic is handled in a buffer-less fashion, which is useful when the

objective is to keep the total input rate within the parameters of the service rate, that is avoiding queuing altogether, in which excess traffic is simply discarded.

Finally, the rate-sharing statistical multiplexing assumes there is a large buffer space available to cope with the excess traffic, and allows higher admissible loads, but there is also greater delay associated. The objective changes to share the service capacity by providing sufficient buffer space to absorb the excess packet rate.

These three different operating principles require different traffic parameters to describe the source traffic characteristics. The first one requires only the peak packet rate of the source (as it deals mainly with constant bit rate traffic). The second principle requires the peak packet rate and the mean packet rate, whereas the third one needs the previous two parameters and some sort of measure of the burst length (to determine the necessary and optimal buffer space). The actual parameters depend on the CAC policy and the information it needs for the implementation of the algorithm.

WMPLS, based on MPLS, involves these three types of mechanisms, as well as ATM, but the admission control mechanism involves additional complexity due to the inclusion of DiffServ and TE parameters. ATM, and WATM, defined their traffic shaping and CAC policies based on IntServ and the differentiation of CBR, VBR and ABR (UBR), including the real-time variations. These parameters are compared in a later section, showing that WMPLS is better, or at least, similar in the performance measurement.

3.2.2 Buffer Overflow and Loss Probability

As mentioned in the third operating discipline mentioned above, buffer space is reserved for data to be stored while the process of aggregation or statistical multiplexing takes place. If the packet arrival rate exceeds the service rate, the queuing delay increases and packet loss occurs if the buffer is overfilled. The

drawbacks of queuing management are the primary reason for QoS degradation, and for this reason the packet loss or overflow probability becomes a very important metric that defines the performance of the overall system.

***Definition 1**: Consider a queue with a fixed service rate C, and define the amount of arriving traffic to the queue during the time interval $(s,t]$ by $A(s,t)$. The queue length at time t is given by $Q(t)$, and assuming the queue size is infinite, the queue length $Q(t)$ satisfies the following equation [47]:*

$$Q(t) = \max\{Q(s) + A(s,t) - C(t,s), 0\}, \text{ for any } s \le t. \tag{3.3}$$

Assuming that $A(s,t)$ has stationary increments the expected arrival rate needs to be less than the service rate in order to maintain the stability of the system, i.e. $\mathbf{E}\{A(t,s)\}/t := \lambda < C$. From [47], the queue length at time t is given by:

$$Q(t) = \sup_{s \le t}[A(s,t) - C(t-s)] \tag{3.4}$$

The overflow probability $\mathbf{P}\{Q > x\}$ has been extensively studied in order to characterize the behavior of the queue, primarily because the distribution of the queue length is a key performance metric for the packet loss probability and the delay distribution [47]. However, due to the supremum operator defined in equation (3.4), which is a non-linear operator, the overflow probability $\mathbf{P}\{Q > x\}$ is inaccurate even when proper arrival statistics are completely characterized.

Buffering and, consequently, overflow probability, as mentioned in the previous section, arises from the fact that various traffic sources are statistically multiplexed. This leads to an in depth analysis of the queue length behavior, which can be viewed as follows: Let X_i be the number of packets generated by the $i-\text{th}$ user at any point in time over a link with capacity C (packets/second), and the transmission time for X_i packets is defined by $T_i = X_i / C$. Assuming there are K users that require an specific amount of bandwidth C, the network has to

provide KC units of bandwidth to maintain the users serviced, which is not necessarily feasible due to the fact that bandwidth limitations cannot be easily expanded based upon users requests. In the case of statistical multiplexing, for all the K users, $S_K = X_1 + X_2 + \cdots + X_K$ packets will share the same link with static bandwidth, and the overall packet transmission time is calculated as $T_K = S_K / KC$. If the probability density functions (p.d.f.) of X_i are independently and identically distributed (i.i.d.), then the first and second order statistics of the aggregate traffic are similar to those of the individual sources (i.e. $\mathbf{E}\{T_K\} = \mathbf{E}\{T_1\}$ and $\mathbf{Var}\{T_K\} = \mathbf{Var}\{T_1\}/N$), which implies that the total transmission time converges to its mean value ($\mathbf{E}\{X_i\}/C$) by the law of the large numbers [47]. The values tend to stabilize (smooth out) as more and more sources are statistically multiplexed, which corresponds to the theory explained by the Central Limit Theorem (CLT).

The previous analysis can be extended so that the queue length behavior can be studied in more detail. For example, a queue (with a length of Q^K) with a predefined capacity of $C(K) = \sum_{i=1}^{K} C_i$ can serve K traffic flows, $A_i(s,t)$, $i = 1,2,\ldots,K$, and (3.2) we obtain:

$$Q^K(t) = \sup_{s \le t} \sum_{i=1}^{K} \left[A_i(s,t) - C_i(t-s) \right] \tag{3.5}$$

The definition of the supremum operator states that $\sup(a+b) \le \sup(a) + \sup(b)$, thus $Q^K(t) \le \sum_{i=1}^{K} Q_i(t)$, where $Q_i(t) = \sup_{s \le t} \left[A_i(s,t) - C_i(t-s) \right]$, which shows that from the total queue-length prospective it is better to aggregate traffic flows with aggregate capacity [47]. In the case that the queue lengths are not equally distributed, the actual total queue-length can be much smaller than the sum of all the individual $Q_i(t)$ queues, this is something that occurs normally in a network because of the heterogeneous nature of traffic.

3.3 Complexity Analysis

Complexity analysis is used as a performance measurement technique in which an algorithm is studied and measured in terms of the number of steps needed to complete the algorithm's goal. This performance measurement has been thoroughly used in several research initiatives, such as the work presented in [48] in which is used in conjunction with synchronization delay to measure the performance of a mutual exclusion algorithm used to effectively share resources in distributed systems. In [49], the authors use this performance metric to statistically measure the performance of the Cluster-based Topology Control (CLTC) protocol. The authors in [40] calculate the storage complexity and communication complexity to analyze the scalability of various ad hoc network routing protocols and introduce the routing overhead of periodically updated link state (LS) messages, which follow the order of $O(N^2)$ where N indicates the number of nodes in the network. The detailed investigation that shows the derivation of the upper bound of the message complexity of the IP address auto configuration protocols mobile ad hoc networks (MANET) is presented in [52].

Message complexity analysis can be divided into separate components in order to be better analyzed. The general methodology of analysis is based on [50], in which a flowchart is used to analyze the time complexity of an image segmentation algorithm based on the recursive shortest spanning tree (RSST). The authors of [51] point out that time complexity is one of the most important factors to measure or compare the performance of different algorithms, and therefore, should be considered when an algorithm is being developed. Based on the complexity analysis method of [50], the message complexity of WMPLS, and other protocols, is analyzed. The method of adding the upper bounds of the time complexity measured at each step are used in the study of the performance of WMPLS because the protocol structure is composed of a sequence of discrete distinctive procedures with its own message complexity. Thus, by adding the

message complexity measured at each step, the message complexity of the whole protocol can be calculated.

3.3.1 Asymptotic Notation

When the analysis of any event, specifically the protocol behavior that has an algorithmic nature, involves a large number of iterations and components, the execution time, and the performance metrics are dominated by the effects of the input size itself. The notation to describe the asymptotic behavior and execution time of an algorithm are defined in terms of functions whose domains are the set of natural number, and those notations are convenient for describing the worst case running-time function $T(n)$ [54], which are usually defined only on integer input sizes, however, the asymptotic notation can be extended into the realms of the real numbers, or it can be restricted even further into the natural number range.

3.3.1.1 Θ-Notation

For any given function $g(n)$, $\Theta(g(n))$ denotes the set of functions $f(n)$ for which there exist positive constants c_1 and c_2, and n_0 such that $0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n)$ for all $n \geq n_0$.

The previous definition shows that $f(n)$ belongs to the set $\Theta(g(n))$, if it can be found in between $c_1 g(n)$ and $c_2 g(n)$ for a sufficiently large n. This also implies that $f(n) \in \Theta(g(n))$, but for reasons that will become apparent later, the following notation is used to convey that $f(n)$ is a member of $\Theta(g(n))$, $f(n) = \Theta(g(n))$. For this definition, $g(n)$ is an asymptotically tight bound [54] for $f(n)$.

The definition of $\Theta(g(n))$ requires that $f(n)$ be nonnegative for the case of n being sufficiently large, that is $f(n)$ is asymptotically nonnegative, and consequently $g(n)$ is also asymptotically negative, or else $\Theta(g(n))$ would be an empty set. Intuitively the lower order terms of an asymptotically nonnegative (which could be a positive) function can be disregarded in order to determine the asymptotical tight bounds because they are very small for large values of n. A very small fraction of the highest-order term is good enough to dominate the lower-order terms

3.3.1.2 O-Notation (Big O)

The notation explained in the previous sub-section bounds a function asymptotically from two sides, above and below. The O-Notation is used when only the asymptotic upper bound is available. For any given function $g(n)$, $O(g(n))$ is defined such as there exists a function $f(n)$ for which there exists positive constants c and n_0 such that $0 \leq f(n) \leq cg(n)$ for all $n \geq n_0$.

This notation, as mentioned before, is used to give an upper bound on a function, to within a constant factor, and it is clear that $f(n) = \Theta(g(n))$ implies $f(n) = O(g(n))$, which can also be shown as $O(g(n)) \subseteq \Theta(g(n))$. This notation allows describing the running time and characteristics of an algorithm merely by inspecting the algorithm's overall structure. Since this notation describes an upper bound, when used to analyze the worst-case scenario of an algorithm, the bound is for every input of the algorithm.

3.3.1.3 Comparison of Functions

Many of the properties of real numbers apply to asymptotic comparisons. Table 3.3 shows the properties assuming that $f(n)$ and $g(n)$ are asymptotically positive.

Transitivity:	$f(n) = \Theta(g(n))$ and $g(n) = \Theta(h(n))$ then $f(n) = \Theta(h(n))$ $f(n) = O(g(n))$ and $g(n) = O(h(n))$ then $f(n) = O(h(n))$
Reflexivity:	$f(n) = \Theta(f(n))$ $f(n) = O(f(n))$
Symmetry:	$f(n) = \Theta(f(n))$ iff $g(n) = \Theta(g(n))$
Transpose Symmetry:	$f(n) = O(f(n))$ iff $f(n) = \Omega(f(n))$

Table 3.3 – Properties of the *Θ(n)* and the *O(n)* functions

4. MPLS TECHNOLOGY

MPLS has its origins in several packet switching technologies used and developed in the 1990s. It was not until 1997 in which the Internet Engineering Task Force (IETF) created an MPLS Working Group in order to define and standardize the protocols and approaches for MPLS.

The main driver for the design and implementation of MPLS emerged from the need to solve the problems that IP networks had, coupled with overlay network topology complications, such as IP over ATM (IPoATM). IP networks have inherent limitations due to its specification, which, for example, results in hotspots due to routing protocols like OSPF [35], and lack of flexibility of services such as DiffServ. The problems that IP suffered indirectly from the overlay model were, for example, the lack of interoperable ATM implementation due to the complexity of the signaling protocol (Q.2931), and the scalability limitations that this implied.

Packet switching originally began with the X.25 protocol definition. Data networks did not have a encompassing technology behind them until X.25 defined a major change in the paradigm of network services, using a best-effort approach, allowing the user to requests certain levels of service. This became the first standardized platform that different vendors could rely on in order to establish connectivity to a unified public network. X.25 used an identification mechanism for each packet on the same physical network based on a logical channel number (LCN), and from this methodology the term *virtual circuit* (VC) emerged.

Successor technologies to X.25 are Frame Relay and ATM, which also used the virtual circuit concept. In the case of Frame Relay VCs are identified as Data Link Connection Identifiers (DLCIs), and ATM identifies them as Virtual Path Identifiers (VPI) or Virtual Circuit Identifiers (VCIs), but regardless of their names, they are considered virtual circuit identifiers, and most importantly label values.

MPLS networks interact with legacy networks, and thus, the labeling scheme used for its proper functioning needs to correlate easily with ATM and Frame Relay labels, which is why most of the RFCs include explicit references to these architectures.

4.1 Label Switching

Switching is the process of forwarding data packets within the network based on a label associated with each packet. Traditional IP routing is a form of packet switching, where and IP address is used to determine the next node in the path to the destination, but this process has many limitations, which have been addressed by label switching. Some of the advantages of label switching compared to normal routing and packet switching are defined as follows [8]:

- Routing might create hot-spots, which are nodes in the network that are overloaded because they are part of the shortest or the best route.
- The need of maintaining a large amount of information for routing creates a need for faster processing, which is not the case when maintaining switching information, that can be executed much faster than a routing decision
- Extending the capabilities of LANs by means of virtual private networks (VPN) can be automatically performed by MPLS rather than a normal IPSec connectivity. Additionally all the broadcasting and multicasting capabilities can be extended using MPLS VPNs, rather than the traditional IPSec methodology that limits this activity.

- The process of labeling data flows has been used in newer technologies, which offers the possibility of provisioning overlay models such as TDM or SONET in a dynamic fashion [8].

The combination of traffic engineering provisioning and the simplicity of providing services such as VPNs under varied types of infrastructures are very attractive for carrier companies and service providers. Since most of the switching network elements or components are capable of forwarding IP packets without having to attach a label to them, allows a hybrid network deployment (including switching and routing mechanisms) that increases the efficiency of the network, and provides a large amount of flexibility in its design.

4.2 Fundamental Concepts of MPLS

Label switching needs a small, fixed format label that will be attached to each data packet so it can be injected in the network, which means that each packet will be carrying an identifying entity that will inform the equipment the way in which it needs to be forwarded. In this section, the key components about assigning labels, creating paths to forward traffic, managing the identifiers when traversing a mixed network, and the signaling and control plane information exchange that will occur in order to maintain the sanity and integrity of the network are discussed.

4.2.1 MPLS Labels

In an MPLS network the packets are labeled by inserting and additional shim header that fits between the network header and the IP header as shown in Figure 4.1. This header carries a label of 20 bits in length, and the following fields:

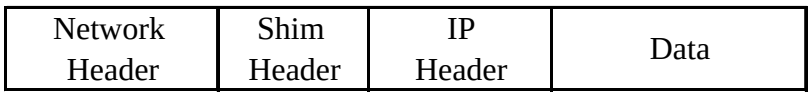


Figure 4.1 – Position of the shim (MPLS) header

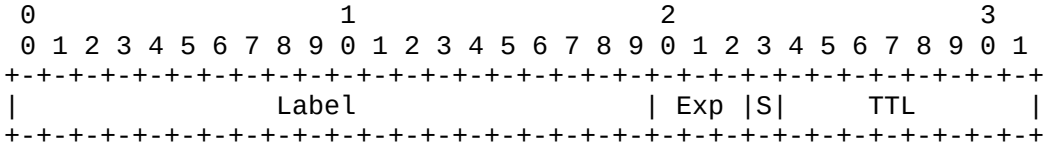


Figure 4.2 – Format of the MPLS shim header

- Three experimental bits that are used for grading services.
- One bit that indicates if this label is the last in a tack of labels.
- Eight bits that carry the Time-To-Live (TTL) information, which will be copied from the IP header, to be used in the same manner as the IP layer does.

4.2.2 Label Switched Path (LSPs) and Label Swapping

The path that is formed for a data packet to flow through the network is defined by the transition in the label values. The mapping for each node is constant, and then the path is determined by the label value at the first node. This path is called a Label Switched Path (LSP). Each node in the MPLS keeps a table that is used to determine the next hop in the LSP. The table is referred to as the Label Forwarding Information Base (LFIB). Once a packet is received from an upstream Label Switch Router (LSR), the LFIB will be consulted to see what interface the packets needs to use to continue its path. In certain cases the label that needs to be used to continue to the downstream router needs to be changed for a new value, and this process is known as Label Swapping.

4.2.3 Mapping Data to an LSP

One of the main functions performed by the first LSR, also known as Label Edge Router (LER), is to determine with which LSP to associate the data packets. In the simplest case the decision is based on the destination, and can be obtained via a look-up mechanism which is similar as those used in IP, which results in the definition of a label value.

The experimental bits in the MPLS header allow the definition of eight classes of service to be defined for an LSP. These values can be used to define prioritized processing of the information and the assignation of LSR resources. The set of all packets that are processes with the same priority is called a Forwarding Equivalence Class (FEC). A FEC is described by the parameters that are used to identify the packets that constitute it. In its simplest form, a FEC can be defined as a single IP address.

The main function of the ingress LSR is to identify the FEC to which each packet belongs to, and to use this to determine the LSP and first label so that it can forward the packet through the MPLS network. Many LSR implementations include a FEC to a Next Hop Label Forwarding Entry (NHLFE) mapping table to help with this resolution.

In the same way that IP packets converge toward their final destination, so the paths taken by MPLS packets may converge. It is possible to maintain distinct LSPs from ingress to egress, but this is not necessary and might waste resources. Instead, LSPs can merge at transit nodes so that packets from the same FEC can be grouped together to the same LSP.

MPLS also allows for LSPs to be tunneled, which is a useful mechanism that allows many LSPs to be treated in the same way in the core of the networks while being still individually treated at the edge. This enhances the scalability of the LSRs in the core of the network and significantly improves the manageability of

connections across the network. Tunneling services are provided by means of label stacking. When a packet reaches the ingress LSR a normal label is attached to it, and when it reaches the entry point to the tunnel, another label is superimposed on the packet, which is also known as pushing a label into the stack. The top-most label is used to forward the packet through the tunnel. At the exit point of the tunnel, the superimposed label is removed, or popped from the stack, revealing the label of the tunneled LSP.

4.3 MPLS Signaling Protocols

All the efforts in the development of label switching rely on a control mechanism to distribute label information between the nodes in the network. These mechanisms are called signaling protocols.

The main objective of the signaling protocols is to setup LSPs, and manage all the information related to this process. Dynamic label distribution manages the creation and continuous management of the LFIBs, being this the key function of the signaling protocol.

The management of the MPLS network is carried by the Control Plane, which use the forwarding characteristics of the IP protocol in order to carry its basic identifying and routing capabilities. The motivation to use IP comes partly from the fact that the predominant deployed data and routing networks use and carry the IP protocol. This dependency leads to Routing-Based Label Distribution, in which labels are allocated and advertised to match the routes in the local routing table. This mechanism is also known as downstream unsolicited label distribution. In this label distribution scheme, each LSR examines its routing table and, for each onwards route, it advertises a FEC and a label on the other interface [8]. Downstream unsolicited label distribution is a good solution to ensure that all data can be forwarded immediately using an LSP. However, it uses a large amount of network resources because labels are distributed to all the routers regardless of the existence of any traffic flowing through the network.

Furthermore, each LSR will advertise a label for each route to each of its neighbors, even when that is not the optimal way to set up an LSP.

An alternative label distribution technique is called On-Demand Label Distribution. This mechanism addresses the issues of the unsolicited label distribution, with the main difference that the LSPs are not necessarily pre-established. In this case, the upstream LSR makes an explicit request to a downstream node for a label to use for a particular FEC. The downstream LSR may answer immediately or may forward the request further downstream. This mechanism can be triggered, or initiated, depending on the network conditions, and the request for a label is usually in the direction that is given by the local routing table. It is possible to have both mechanisms running on a given network at the same time.

Signaling protocols reduce the number of interactions between management and the network when a new LSP is required. Instead of sending a message to each of the nodes on the LSP, a single management request can be sent to the ingress of the LSP and the signaling protocol can set it up automatically.

4.4 Traffic Engineering

Traffic Engineering (TE) is the process in which data is routed through the network according to a management view of the availability of resources, and the current and expected traffic loads of the network. The class of service and the required quality of service can be factored into this process [8][9][11].

Traffic engineering can be performed as an offline process in which data is collected from the whole network and it is then processed to obtain the most efficient and optimal routes for traffic to be delivered to the end nodes. Alternatively, it may be a dynamic operation in which each new traffic flow is routed according to the current network usage. Traffic engineering can also be performed on a flow-by-flow basis, allowing the redirection of traffic around hot-

spots or network. In this way, traffic engineering helps the network provider make the best use of the available resources to meet service level agreements.

A key characteristic of MPLS is to facilitate improved traffic engineering in service provider networks, where mostly on-demand label distribution is used. However, in order to provide the traffic engineering more complex constraints are presented to the signaling protocol. For example, if a path is computed offline or at the initial node, it can be signaled as an explicit route in order to meet certain bandwidth requirements. This requires the signaling protocol to not only know what the route from source to destination, but also the bandwidth resources in each of the intermediate nodes, so that the proper route can be established for the provision of the TE, even though this would not be the shortest-path according to normal routing protocols.

MPLS offers the ability to implement traffic engineering at low cost in equipment and operational expenditure. Some of the components that make it attractive to be used in a traffic engineered network include:

- MPLS has the ability to establish explicitly routed LSPs
- Resources within the network can be dynamically reserved as LSPs are being established, and can also be updated in a similar fashion, guaranteeing that traffic flows will have the required QoS levels.
- Traffic can be split in parallel LSPs, that is, multiple LSPs can be set between the source and destination and traffic can be distributed among them.
- LSPs can be preempted so that network resources can be managed automatically when LSPs of higher priority are being set up.
- Recovery procedures can be defined describing how traffic can be transferred to alternate LSPs in the event of a failure.

An important characteristic that separates MPLS from any other implementation of traffic engineered protocols is the load-sharing capability decisions which need

to be made only once, at the ingress LER, rather than at each node within the network. This makes traffic propagation considerably more efficient for the overall network.

4.5 Explicit Routes and Abstract Nodes

As mentioned above, MPLS can be set up using the existing routing information provided by upper layer protocols, which implies that preferred routes (shortest-path routes) tend to converge, which places much traffic onto a few links, while other links remain underutilized.

The fundamental advantage that MPLS provides for traffic engineered networks is the ability to set up a virtual circuit switched overlay to the IP routing model. MPLS signaling protocols will allow the path or explicit route of the LSP to be provided by the ingress LSR or LER. Explicit routes are specified as a well-ordered series of hops expressed as IP addresses, IP prefixes, or identifiers of autonomous systems. The LSP must traverse de hops in order.

Because each hop can be an expression of multiple hosts (an IP prefix or an autonomous system), the elements of an explicit route are referred to as *abstract nodes*. The LSP must traverse the abstract nodes in the order that they are specified in the explicit route, and where the abstract node implies more than one actual node, the LSP must traverse at least one LSR that is a member of the abstract node.

The abstract node hops in the explicit route may each be defined as *strict* or *loose*. In a strict hop, no LSR may be inserted in the actual path of the LSP between the LSRs that are members of the previous abstract node and those that are members of the current hop. If the hop is loose, the local routing decisions may fill in additional nodes necessary to reach an LSR that is a member of the abstract node.

4.6 Constraint-Based Routing and Resource Reservation

A virtual circuit-switched overlay is established using MPLS, and if specific reservation requirements are associated with each LSP, it becomes possible to reserve precisely enough resources for each flow and to guarantee SLAs based on a much more precise allocation of network capabilities. MPLS requires the following information in order to fully provide constraint-based services:

- The routing protocol must advertise the capabilities and available resources on each of the links.
- The application that requires establishing a traffic flow, and the set up of an LSP, must indicate the characteristics of the flow in order to map the proper resources of the network.
- The computation of the paths must take into consideration the requirements of the LSP and the availability of network resources by performing constraint-based routing.
- The signaling protocol must support and provide explicitly routed LSPs.
- The MPLS signaling protocol must also be able to signal the LSP resource requirements so that the appropriate reservations can be made at each LSR along the path.

It is still possible to set up a new LSP by appropriate the resources used by other LSPs. This process of LSP *preemption* needs a controlling mechanism that will ensure the integrity of the entire network management process. In MPLS, preemption is achieved by using two priority values associated with each LSP. The *holding priority* indicates how hard an existing LSP will hold on to resources once they have been assigned to it. The *setup priority* establishes how important the setup of the new LSP is. An LSP with a greater setup priority may preempt an LSP with a lower holding priority. Obviously, network thrash will be avoided only if the LSPs have holding priorities greater than or equal to their own setup priorities.

4.7 Extensions to RSVP for LSP Tunnels (RSVP-TE)

The Resource ReserVation Protocol (RSVP) is suitable to be extended to provide label distribution in traffic engineered MPLS networks because it deals with end-to-end reservation of resources for traffic flows, a concept similar to traffic engineered MPLS. The MPLS Working Group within the IETF decided to focus its efforts on this protocol as the MPLS signaling protocol for traffic engineering applications, and to undertake no new efforts relating to CR-LDP [12].

4.7.1 Reuse of the RSVP Functionalities

RSVP-TE manages to reuse RSVP fairly comprehensively. All seven of the RSVP messages find a use in RSVP-TE, even though the ResvConf is less significant than when it is used in RSVP [8][13][14]. RSVP is essentially a request/response protocol [8][14] with Path messages being used to navigate a path and request resources for traffic flows, and Resv messages returning along the path to indicate what resources should be reserved.

This flow of messages matches the requirements of Downstream On-Demand label distribution [25] and can be extended easily adding information to the messages. Since RSVP messages are built from objects [15][16], which are basically Length-Type-Variables (LTV) structures, this is easily achieved [17][18].

Although RSVP contains proper mechanisms for describing traffic and for specifying the reservation requirements, it does not have the facilities for other aspects that are required for a traffic engineered MPLS protocol. The following extensions were made to cover the needs of MPLS:

- Label management
- Requesting and controlling route definition
- Preemption of resources
- Maintaining connectivity between LSRs

RSVP has a drawback which is the processing overhead associated with the soft state nature of this protocol. Traffic engineered MPLS LSPs do not always need to fluctuate according to changes in the routing database, especially when the route is explicitly defined, and for this reason the extensions were made to comply with [19].

4.7.2 Distributing Labels with RSVP-TE

LSP setup is requested in RSVP-TE using the downstream on-demand label distribution by the inclusion of the Label Request Object on a Path message. This object is an extension that traditional RSVP does not include.

The initiator determines the type of message in sending by means of the C-Type of the object. The request included in the Path message takes three forms as shown in Figure 4.3, which defines the type of underlying network that will be carrying the traffic, and they all carry a layer 3 protocol identifier to indicate the egress of the LSP what kind of traffic it should expect to discover when it removes the shim header from the MPLS packets. The format of the RSVP-TE label object is shown below.

The RSVP are identified using the Session Object which is present on all the RSVP messages, and RSVP-TE provisions two new C-types to distinguish between the routing protocol version (IPv4 or IPv6). If LSP merging is provided, all the ingress LSRs must assign the same session identifier, which means that the Tunnel ID field in the session object must be known to all sources (which can be set using a management protocol or set by the application itself) and that each source must use the same Extended Tunnel ID object. The format of the session object is shown in Figure 4.5.

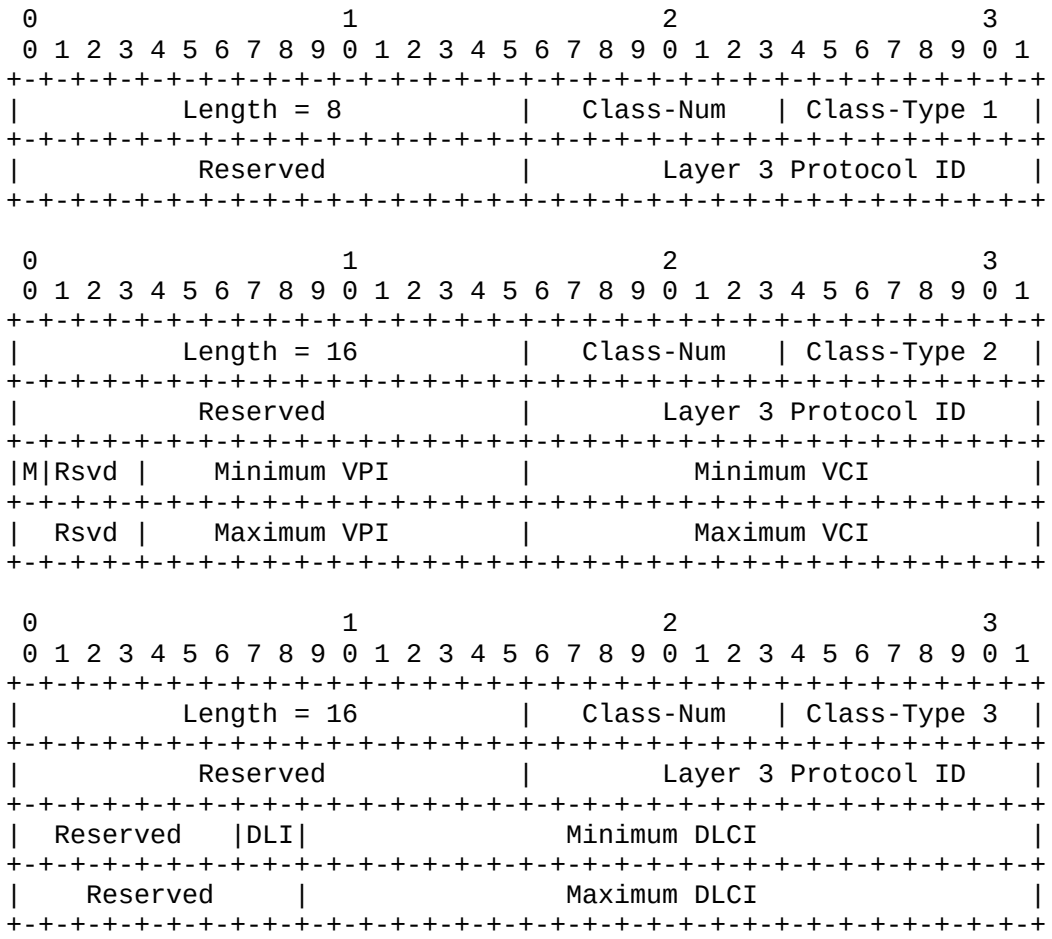


Figure 4.3 – The RSVP-TE Label Request Objects

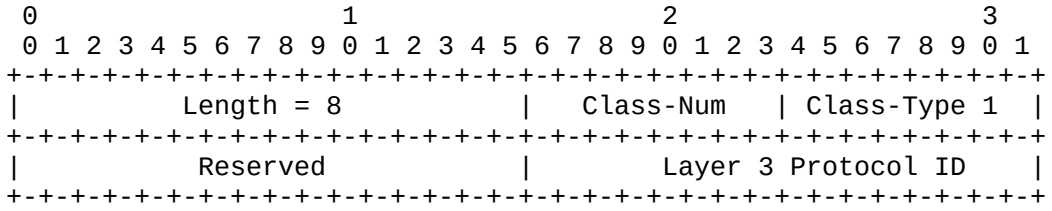


Figure 4.4 – RSVP-TE Label Object format.

The Sender Template object is used to identify traffic flows in RSVP within the context of a session, and its format is show in Figure 4.6. In RSVP-TE there may often be little difference between a traffic flow and a tunnel, and the RSVP-TE form of the Sender Template Object includes a source address as in RSVP and a LSP ID to replace the source port field information. The LSP ID is unique within

the context of the source address and it represents an instance of the tunnel identified by the Tunnel ID of the Session Object.

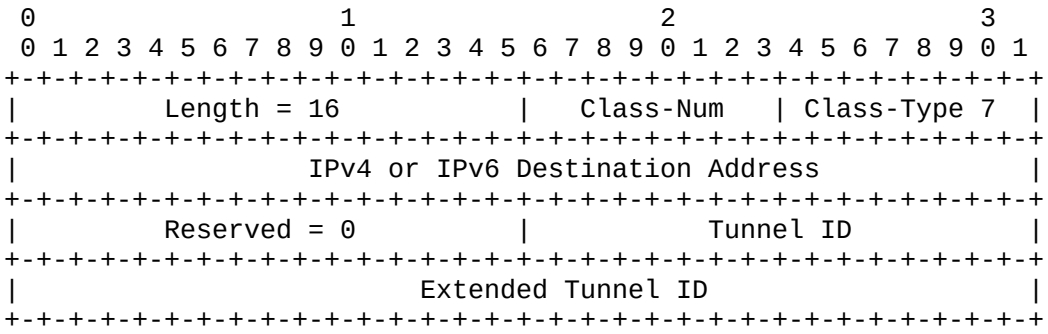


Figure 4.5 – RSVP-TE Session Object format.

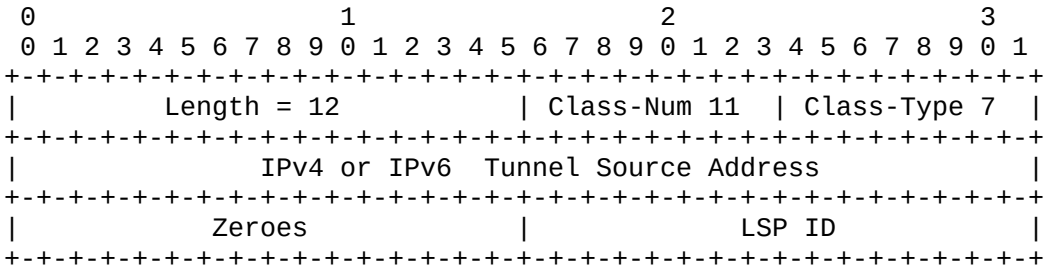


Figure 4.6 – RSVP-TE Sender Template Object format.

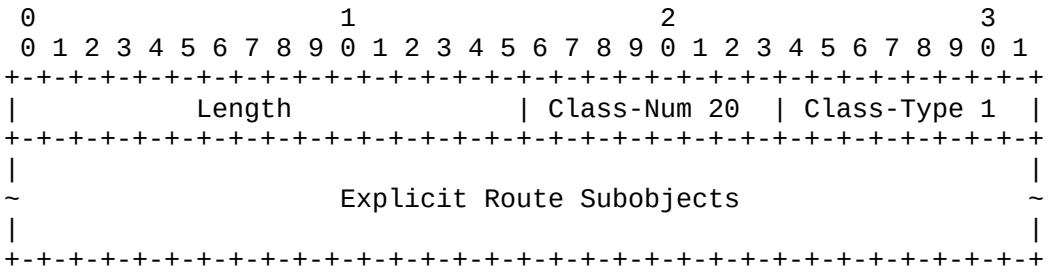


Figure 4.7 – RSVP-TE Explicit Route Object format.

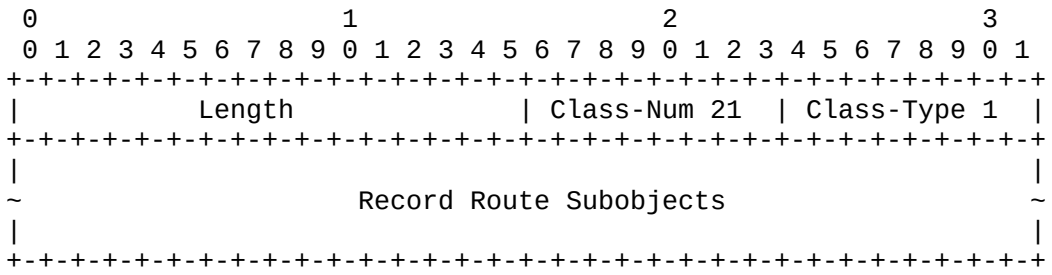


Figure 4.8 – RSVP-TE Record Route Object format.

Explicit routes are signaled by the inclusion of an Explicit Route Object (ERO). This is a single header followed by series of subojects, and the value of these subobjects are encoded a Type-Length-Variable triplets. The ERO and its subobjects are subject to the processing rules. RSVP-TE also includes a mechanism known as *route record*, which is handled through the Route Record Object (RRO) that can be present either on a Path or Resv message. It records the hops through which the message has been routed and when received at the egress or ingress of the LSP provides information of all the nodes the messages traversed in the path. The format for the ERO and the RRO objects are shown in Figure 4.7 and Figure 4.8 respectively.

4.7.3 Resource Requests and Reservation Process

The FlowSpec objects are inherited from RSVP without any change. RSVP defines three reservation styles that allow for different modes of resource sharing. Two of these, the Fixed Filter (FF) and the Shared Explicit (SE) are supported in RSVP-TE, and the Wildcard Filter (WF) is not considered appropriate for traffic engineering since its real application is used for multipoint-to-point flows in which only one sender sends at any time.

In RSVP-TE the choice of style is made by the egress node, but should be influenced by the setting of the SE style. If the FF is used, a unique label and unique resource reservations are assigned to each sender, which means that there

are no resource sharing and no merging of the LSPs. On the other hand, the SE style allows sharing and merging of LSPs, which is particularly useful in rerouting techniques such as make-before-break.

The choice between the FF and the SE styles is therefore governed by the function within the network. If resource sharing and LSP merging are not supported, FF must be used. Manu existing MPLS implementations do not support SE style and will clear the SE style desired bit in the Session Attribute object as they forward the path message. This act in itself does not guarantee that the egress will not select the SE style, but may help to prevent it.

4.7.4 Summary of RSVP-TE Messages and Objects

The definitions of the messages in RSVP-TE are found in [11] and the objects that are inherited from RSVP without change (such as the FlowSpec objects) are defined in [13], and the notation used is called *Backus-Naur Form* (BNF). The ordering of objects within a message is strongly recommended, though it is not mandatory (except the members of composite objects must be kept together) and an implementation should be prepared to receive objects in any order while generating them in the order listed in the figures below.

```
<Path Message> ::=      <Common Header> [ <INTEGRITY> ]
                        <SESSION> <RSVP_HOP>
                        <TIME_VALUES>
                        [ <EXPLICIT_ROUTE> ]
                        <LABEL_REQUEST>
                        [ <SESSION_ATTRIBUTE> ]
                        [ <POLICY_DATA> ... ]
                        <sender descriptor>

<sender descriptor> ::= <SENDER_TEMPLATE> <SENDER_TSPEC>
                        [ <ADSPEC> ]
                        [ <RECORD_ROUTE> ]
```

Figure 4.9 – Formal definition of the RSVP-TE Path message

```
<Resv Message> ::=      <Common Header> [ <INTEGRITY> ]
                        <SESSION>  <RSVP_HOP>
                        <TIME_VALUES>
                        [ <RESV_CONFIRM> ]  [ <SCOPE> ]
                        [ <POLICY_DATA> ... ]
                        <STYLE> <flow descriptor list>
                        [ <RECORD_ROUTE> ]
```

Figure 4.10 – Formal definition of the RSVP-TE Resv message

Figure 4.9 shows the formal definition of the Path message. The sequence of objects, Sender Template, Sender TSpec, Adspec, and Record Route are often referred to as the Sender Descriptor. This becomes relevant in the context of the Resv message, which will carry information relevant to one or more Sender Descriptor. Figure 4.10 shows the definition of the Resv message format, where the Flow Descriptor List is a composite sequence of objects that allows a single Resv message to describe reservations for multiple Sender Descriptors requested on Path messages. There are two methods of listing Flow Descriptors within RSVP-TE, depending on the Sender Selection Control field in the Style object. Only Fixed Filter and Shared Explicit styles are supported in RSVP-TE (See Figure 4.11). The FF Flow Descriptor (Figure 4.12) is a composite object that contains the FilterSpec and Label objects, which optionally includes the Record Route object. The last element of the FF Flow Descriptor is recursive, allowing a list of sub-lists where each sub-list starts with a FlowSpec.

```
<flow descriptor list> ::=    <FF flow descriptor list>
                              | <SE flow descriptor>
```

Figure 4.11 – Definition of the Flow Descriptor Object

```
<FF flow descriptor list> ::= <FLOWSPEC> <FILTER_SPEC>
                              <LABEL> [ <RECORD_ROUTE> ]
                              | <FF flow descriptor list>
                              <FF flow descriptor>

<FF flow descriptor> ::= [ <FLOWSPEC> ] <FILTER_SPEC> <LABEL>
                         [ <RECORD_ROUTE> ]
```

Figure 4.12 – Definition of the FF Flow Descriptor Object and that contains FlowSpec subobjects

```
<SE flow descriptor> ::=      <FLOWSPEC> <SE filter spec list>

<SE filter spec list> ::=     <SE filter spec>
                              | <SE filter spec list>
                              <SE filter spec>

<SE filter spec> ::=           <FILTER_SPEC> <LABEL>
                              [ <RECORD_ROUTE> ]
```

Figure 4.13 – Definition of the SE Flow Descriptor Object and that contains filter specs

The SE style also uses compound objects, as shown in Figure 4.13, but there is only one FlowSpec object that may be present; the subsequent list of SE Filter Specifications matches Sender Descriptors and all use the one FlowSpec. The rest of the RSVP-TE messages are kept unchanged.

The RSVP Hello extension enables RSVP nodes to detect when a neighboring node is not reachable. The mechanism provides node-to-node failure detection. When such a failure is detected it is handled much the same as a link layer communication failure. This mechanism is intended to be used when notification of link layer failures is not available and unnumbered links are not used, or when the failure detection mechanisms provided by the link layer are not sufficient for timely node failure detection. It should be noted that node failure detection is not the same as a link failure detection mechanism, particularly in the case of multiple parallel unnumbered links.

The Hello extension is specifically designed so that one side can use the mechanism while the other side does not. Neighbor failure detection may be initiated at any time, which includes when neighbors first learn about each other or just when neighbors are sharing Resv or Path state. The Hello extension is composed of a Hello message (shown in Figure 4.14), and the Hello message processing between two neighbors supports independent selection of, typically configured, failure detection intervals. Hello Messages also contain enough information so that one neighbor can suppress issuing hello requests and still perform neighbor failure detection. A Hello message may be included as a sub-message within a bundle message.

```
<Hello Message> ::=     <Common Header> [ <INTEGRITY> ]
                        <HELLO>
```

Figure 4.14 – Definition of the RSVP-TE Hello message

Neighbor failure detection is accomplished by collecting and storing a neighbor's "instance" value. If a change in value is seen or if the neighbor is not properly reporting the locally advertised value, then the neighbor is presumed to have reset. When a neighbor's value is seen to change or when communication is lost with a neighbor, then the instance value advertised to that neighbor is also changed. The HELLO objects provide a mechanism for polling for and providing an instance value. A poll request also includes the sender's instance value. This allows the receiver of a poll to optionally treat the poll as an implicit poll response. This optional handling is an optimization that can reduce the total number of polls and responses processed by a pair of neighbors. In all cases, when both sides support the optimization the result will be only one set of polls and responses per failure detection interval. Depending on selected intervals, the same benefit can occur even when only one neighbor supports the optimization.

5. WIRELESS MPLS TECHNOLOGY

The development of the WMPLS protocol as a homogeneous technology to MPLS and GMPLS has been focused on providing traffic engineering (TE), Quality-of-Service (QoS), and Differentiated Services (DiffServ), while supporting integrated services of real-time and non-real-time data. As mentioned above, the main driver for the design of this new protocol has its roots on four main circumstances, most of which come from the realization of another initiative related to wireless ATM. As MPLS is the obvious evolution from the problems and limitations of ATM, the same can be said about WMPLS.

Some of the limitations and problems that were present in the design and operation of ATM and WATM networks were that initially they were not design to support service differentiation of prioritized data (i.e., differentiated services). Being a high-speed switching technique that relied on many services from upper layers, ATM and WATM did not include the mechanisms to provide a sufficiently robust error and flow control in order to enhance hop-by-hop reliability, and it expected the upper or lower layer protocols to control ARQ reliability services. The lack of proper data integrity management made WATM very unreliable for connectionless or connection-oriented mobile and ad-hoc wireless networking that intrinsically require efficient data relay functions. Finally, ATM and WATM are designed to switch fixed-size cells of 48 bytes where the type of application specifies the payload ATM adaptation layer (AAL) segmentation and reassembly (SAR) format. This puts a limit to the flexibility of applications (payload coding for error control or encryption) to various wireless systems, as the performance of a wireless system relies on effective transmission of large data packets when the conditions are given, and they're should be able to fallback to lower speeds, thus smaller packet sizes, when the wireless medium is congested.

5.1 Applicability of WMPLS

The provision of guaranteed QoS, the negotiation of TE parameters, soft-handover capabilities embedded in WMPLS, and provision for local participation in network management make it a suitable protocol for commercial wireless high-speed data services and distribution of multimedia content to portable wireless devices like cellular phones, PDAs, and laptop computers with high reliability. Applying multicasting service applications in MPLS can also be effectively achieved by means of the signaling protocols, which in this case will be RSVP-TE [12], and provide a wide range of applications, including the delivery of music, video, games, as well as other applications to any subscriber in the wireless network.

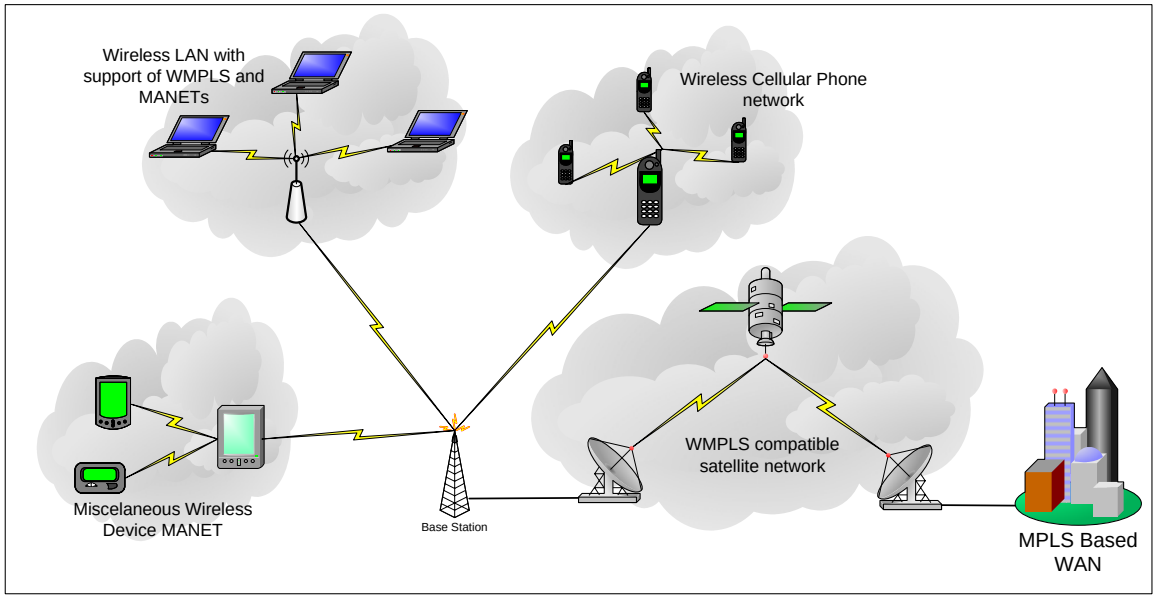


Figure 5.1 – Example of the applicability of WMPLS

WMPLS can also applied for satellite communication applications, like direct broadcast or direct multicast of high-volume traffic content, due to the MPLS robust design for backbone network data transmission (see Figure 5.1). WMPLS is also very suitable for dynamic high-speed MANET [10] applications, mostly

because of its ability to provide high-speed reliable soft-handover procedures with guaranteed QoS negotiation. WMPLS also supports aggregation of traffic and node relaying mechanisms which enhances the performance and robustness of a MANET.

5.2 Technical Overview of WMPLS

The basic operations of WMPLS are the same as that of standardized MPLS, with additional features to ensure reliability and efficiency by providing flow and error control coding mechanisms, which can be applied hop-by-hop or end-to-end. Due to the flexibility of the window size, packet transmission and retransmission can be controlled based on the changing channel conditions, improving the data throughput, maintaining the negotiated traffic parameters, and reducing the BER and packet loss parameters. With the use the RSVP-TE signaling protocol, including some wireless extensions, dynamic path setup and control of soft handover procedures, traffic aggregation and node relaying mechanisms are possible. RSVP-TE also ensures flexible QoS and TE negotiations to be made in order to support DiffServ, minimizing the number of dropped or lost calls between the nodes. Additional functionality of WMPLS includes the flow control of traffic to control and limit the rate in which an application transmits information, and also provides congestion control for data transfer over the different types of traffic that share several transport connections (based on Forward Equivalence Classes (FECs)).

WMPLS does not require any modifications to existing routers and switches in the current MPLS backbone network. In the boundary of the wireless mobile communication network and the backbone network, however, a translator is needed in order to remove WMPLS additional headers and control information, and present the packets to the backbone network as standard MPLS headers and packets. Since WMPLS is homogeneous with MPLS, the translation procedure is straightforward and implies a minimal processing. Once the conversion is done,

the packets can traverse the network as if they originated from the any node within backbone network.

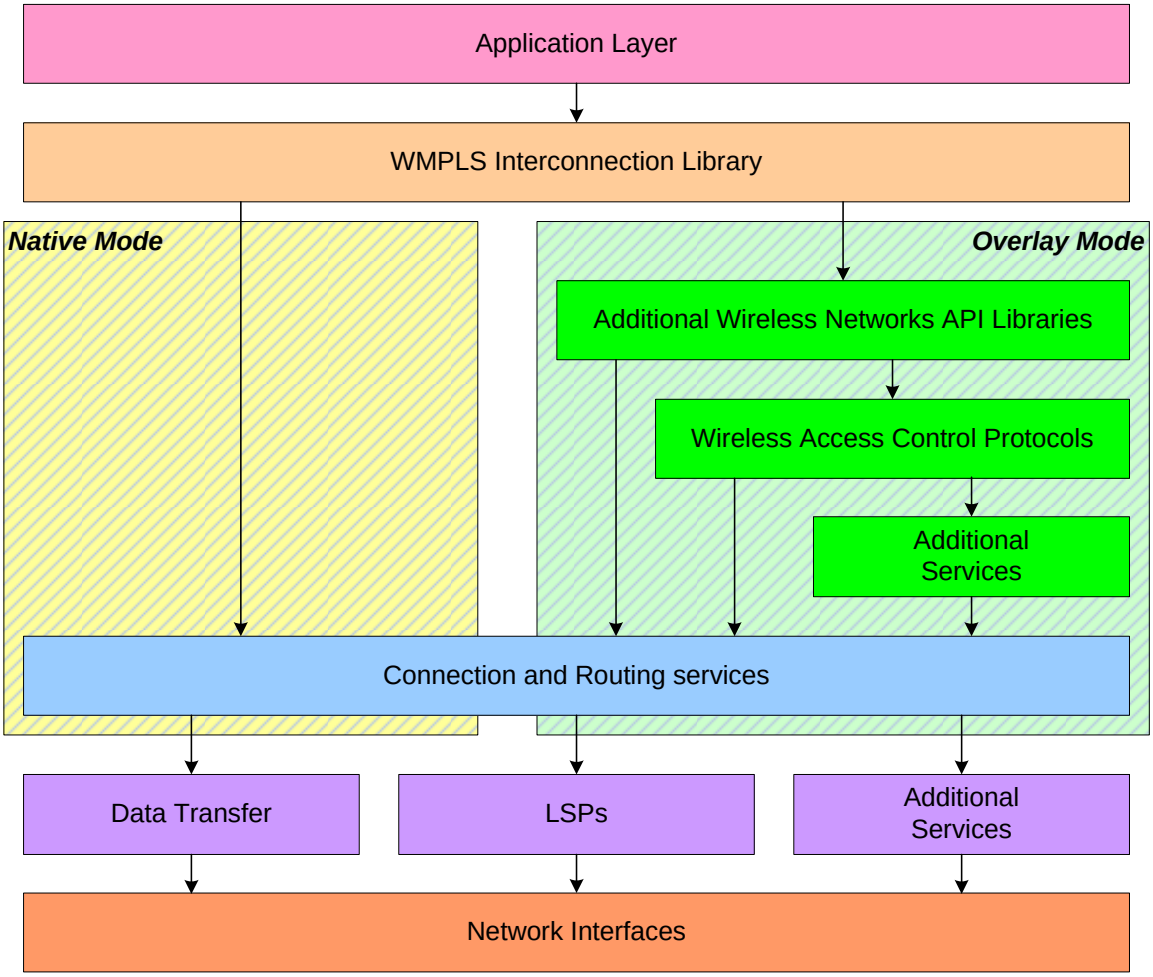


Figure 5.2 – WMPLS reference model, with Native and Overlay models defined

5.3 WMPLS Reference Model

The model used in the design of WMPLS is based on two different modes of operation. The Native Mode is the most efficient and homogeneous mode, which is the direct extension of MPLS into the wireless domain. The network in this case is purely controlled by WMPLS signaling and all the operations of data transport and routing decisions are based on the RSVP-TE protocol alone. The second mode, the Overlay Mode, relies on services provided by other intermediate protocols already available for wireless networks, which provide

routing information and/or transport mechanisms over various types of infrastructures. For example, WMPLS can be implemented over WLAN protocols (802.11x) and can work based on Ethernet frames managed by a RTS/CTS medium access control scheme using CDMA/CA for network access policing.

Figure 5.2 shows a graphical representation of the reference model used for the design of WMPLS, in which the both the native and overlay modes have a common infrastructure that is transparent for the application layer, allowing the services currently available to be transparently deployed throughout the network. However, the interaction between the actual transport services is left to the proper mechanisms in the case of the overlay model, as the intrinsic details of the transport network should be transparent to WMPLS as long as the QoS and TE parameter negotiation is manageable. For both cases the signaling is in charge of establishing and releasing networking resources and LSPs to properly map them to QoS profiles.

5.4 WMPLS Architecture

The architectural model of WMPLS is show in Figure 5.3, in which the WMPLS Service Access Point (SAP) serves as an API for higher services protocols, as the transport protocols, by either directly accessing WMPLS capabilities through its SAP, or by using other routing protocols interfaces through it, and it also provides access to the Logical Link Layer (LLC) operations either in native or overlay mode, using additional Medium Access Control (MAC) procedures. Additional internal interfaces are used for sending signaling information regarding the WMPLS node, for sending encapsulated data in between upper and lower layer protocols, and management information for local or remote monitoring and or management (by means of SNMP or similar frameworks).

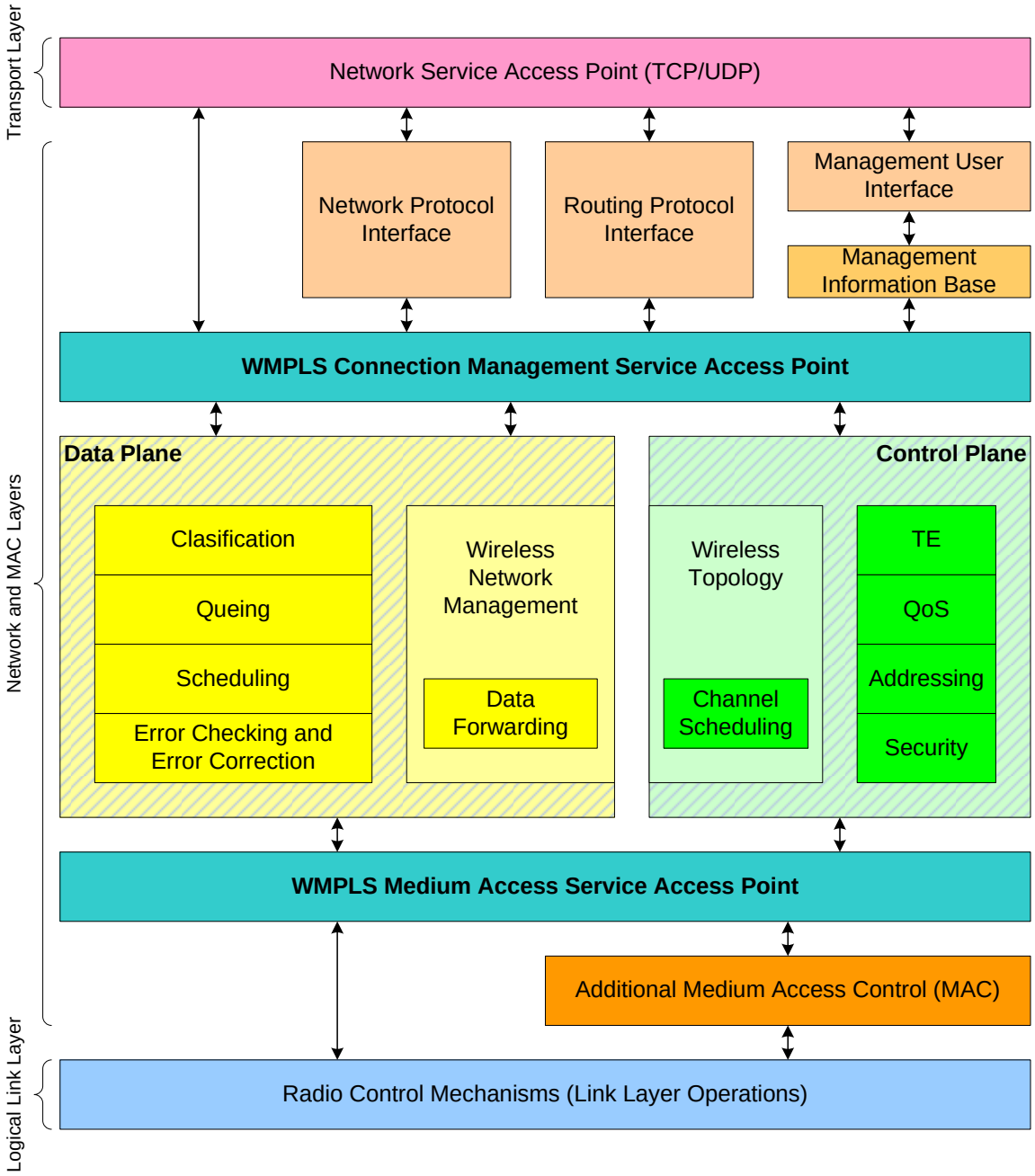


Figure 5.3 – WMPLS architectural model

The coordinating component of the entire infrastructure is the WMPLS Connection Management SAP. In the context of WMPLS a *connection* is referred to as any kind of end-to-end communication service, including transmission of datagrams, best-effort data channels or multimedia transfer covering audio,

video and data connections. The establishment of any communication process (or connection) is initiated by upper layer applications that can provide a set of parameters that conform to a typified QoS or TE traffic profile, or it can be best effort traffic exchange. All interaction between the WMPLS stack (including the other network protocols stacks) is performed through the WMPLS Call Management SAP. All signaling information, including traffic management and path set-up is performed by the Control and Data Planes based on information provided by the upper layers through the WMPLS.

Each connection is individually mapped to a WMPLS connection entry in order to maintain specific state information and connection identification (for a context look-up mechanism) which will be used to map it to an LSP and or a FEC. All the actual mappings and connection information is managed through the Connection Manager, but is actually stored in the Control Plane of the protocol.

The Data Plane performs two major tasks for the traffic in order to provide the QoS and TE functionalities: The first task is in charge of the classification, queuing and scheduling of traffic, and in case the incoming traffic was corrupted during the transfer over the wireless medium, it provides error checking and basic error correction capabilities. The first task is performed on traffic that is either traversing the network and arrives at any given node, or it is traffic coming from upper layers through the WMPLS Connection Management SAP. The second task is focused on traffic ready to be transmitted to the next available node, which is also known as the *Data Forwarding* task. For this secondary task to be successfully completed, a very close interaction and cooperation between the *Wireless Network Management* component of the Data Plane and the *Wireless Topology* component of the Control Plane. The interaction between these two components allows for a proper handling and management of traffic that is sent over the wireless medium due to its time-varying nature. When operating in the overlay mode, the data forwarding mechanisms are simplified to those employed in native mode MPLS.

The primary task of the Control Plane is to manage the TE, QoS, addressing and security components of the communication channel. The channel scheduling operates jointly with the data forwarding component, and relies on the information provided by the addressing and security components. The TE and QoS components interact with the applications through the WMPLS connection management SAP, and store required information per connection and govern the scheduling of traffic so that the data plane can properly schedule it. These parameters are also primarily used for the path establishment phase, in which the proper links between nodes need to be established and continually monitored in order to allow traffic flow based on the predetermined parameters. All the signaling message exchange and procedures related to RSVP-TE are managed by the control plane operations.

Both the data and control plane share a Management Information Base (MIB) that contains the information regarding the topology of the network, the table of neighboring nodes and the current connections each node has. This information is also used for routing mechanisms in order to determine the available paths between end-points of a connection.

The WMPLS Medium Access SAP provides the unified interface for interconnecting to the wireless medium access control (MAC) mechanisms that will finally transport the data between end-points of the connection. This unified interface provides homogeneous services for native WMPLS interconnections, or overlay model mechanisms without any difference. The simplification on the transport of the data when operating in an overlay model is provided by the control plane.

5.5 Design Considerations

This section includes the proposed protocol definitions, changes and extensions that will allow WMPLS to be homogeneous with the MPLS and GMPLS

definitions. Initially all the architectural design components are discussed, and then the algorithmic and procedural design are discussed.

5.5.1 WMPLS Label Format

The basic building block of WMPLS, the label format, involves three fundamental protocol header formats, which are discussed below. The modifications are made to the format shown in Figure 4.2. In all of the three header formats, the first 2 bits of the 20-bit Label field are used as a Flag (F) field. The flag field will indicate if a Control field and a cyclic redundancy check (CRC) field are applied or not when they are set, and it will also indicate the length of the applied Control field either being 1 or 2 bytes, corresponding to the number of sequence bits used, either 3 or 7 bits, respectively. The WMPLS header format with no Control field and no CRC field, shown in Figure 5.4, are be used in the case it is being operating in an overlay model, where the lower of upper layer protocol provide error and flow control mechanisms. In order to identify this label format, the first two bits of the label will be set to zero, which will imply that no control field and no CRC field are being used.

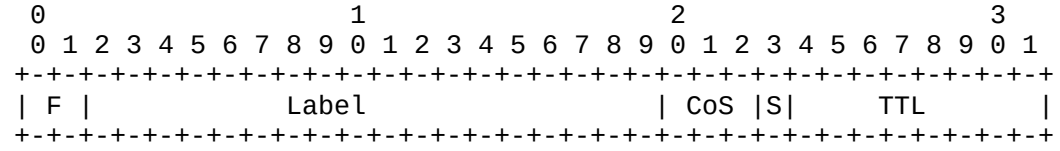


Figure 5.4 – WMPLS header format without the Control or CRC bits

In the Control field, as shown in Figure 5.5, N(S) is the sending sequence packet number and N(R) is the automatic retransmission request (ARQ) or flow control acknowledging frame sequence number.

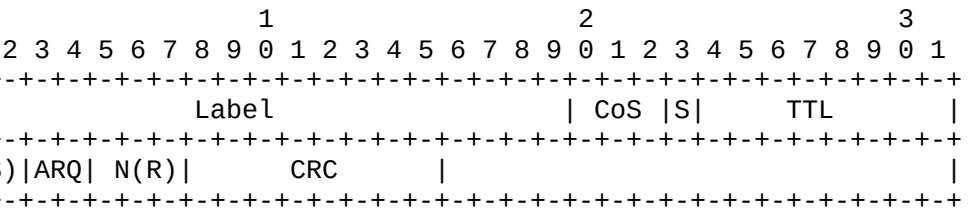


Figure 5.5 – WMPLS header format 3-bit sequence number control field and CRC field

Using more sequence numbering bits, as show in Figure 5.6 will allow larger flow control window to be established, which will in turn allow high-speed sequential frame transmission. This option can be used to enable end-to-end or hop-by-hop error and flow control. In applications of mobile ad-hoc networking, it is necessary to have the option of hop-by-hop error and flow control, as it will be discussed later.

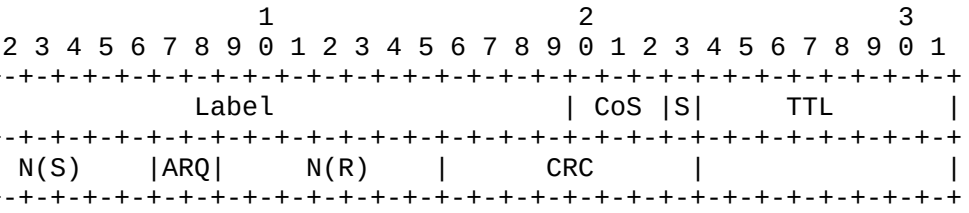


Figure 5.6 – WMPLS header format 7-bit sequence number control field and CRC field

The flag bits determine which of the above header formats will be used. The detailed bit combinations are shown in Table 5.1. The values for the ARQ bits are defined in **Table 5.2**.

The Resource Reservation Protocol with Traffic Engineering extensions (RSVP-TE) allows strict explicit routing and loose explicit routing, and the focus of the research is on the applications of the loosely explicitly routing topology in WMPLS in order to enable simple and reliable soft handover procedures for mobile communications. The section of the wireless mobile network that may change due to handover procedures is defined as a group of abstract nodes. Grouping the mobile hosts will enable the wireless network to perform handover

from one base station to another within a mobile cellular environment without breaking the LSP connection.

5.5.2 Extensions to RSVP-TE for WMPLS Operations

The RSVP-TE extensions were developed in order to support LSP control in MPLS networks so that both strictly and loosely explicitly routed LSPs (ER-LSP) [25] could be provided. For the loose segment in the explicitly routed (ER) LSP, the hop-by-hop routing can be used in the Path message forwarding. For the setup of a WMPLS LSP, a Path message will be transmitted from the source LSR. In the Path message, the LABEL_REQUEST object will request the desired label types for WMPLS setup operations, informing the nodes of the desired LSP to reserve the requested traffic parameters. The extension necessary to trigger the setup of a WMPLS LSP through RSVP-TE needs to have a new C-Type assignment within the LABEL_REQUEST object, such that proper wireless traffic parameters and connection types can be recognized in the Path message.

Flag Bits		***Control Field Sequence Numbers N(R) and N(S) and 2-bit FEC & ARQ***
0	0	No Control and CRC bits
0	1	3-bit N(R) and 3-bit N(S)
1	0	7-bit N(R) and 7-bit N(S)
1	1	Reserved for future applications

Table 5.1 – WMPLS header flag bits control bits

The format of the Path and the Resv messages supporting wireless applications follows the format defined in [11] with extensions to the Label Request Object.

ARQ and Flow control bits		*Flow Control and Error Control Acknowledgement of frames*	*Control Symbol*
0	0	Accumulative acknowledgement of N(R-1)	RR
0	1	Receiver Not Ready flow control and accumulative acknowledgement of N(R-1)	RNR
1	0	Go-Back_N ARQ REJECT N(R) signal and accumulative acknowledgement of N(R-1)	REJ
1	1	Selective Reject/Repeat N(R) signal	SREJ

Table 5.2 – WMPLS header flow control and error control acknowledgement

5.5.2.1 Extensions to the Label Request Object for LSP setup

Among the objects of the Path message, the Label Request Object requires extensions to be made in order to work with the wireless application labels. Based on the protocol definitions of [11], there are three possible C_Types specified under the Label Request Class 19, which is the Label Request Object class number.

The first type, Type 1, is a Label Request without a label range defined. Type 2 is a label request with an ATM label range, and Type 3 is a label request with a Frame Relay label range. In order to request a wireless application specified label an additional wireless label C_Type (Wireless Label) is defined as the WIRELESS_LABEL_REQUEST object. The fields on the wireless label request, shown in Figure 5.7 are defined

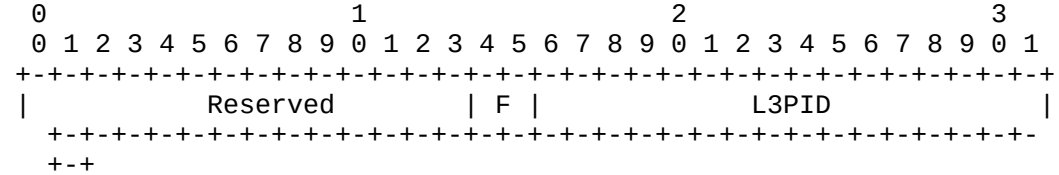


Figure 5.7 – Wireless Label Request object format. Class = 19, C_Type = WIRELESS_LABEL_REQUEST

Field Name	*Description*
Reserved	This field is reserved. It MUST be set to zero on transmission and MUST be ignored on receipt, as defined in [11].
F	Flag ID used to identify the WMPLS label type, as defined in Table 5.1.
L3PID	Identifier of the layer-3 protocol using this path. The standard Ethertype values used in [11] are also used in this field.

Table 5.3 – Field description of the WIRELES_LABEL_REQUEST object

5.5.2.2 Extensions to the Label Object

The Label Object needs to be extended in order to operate with the wireless labels contained in it. In the case where the labels are carried by the Resv messages, the wireless application labels must be distinguished from the generic 32-bit labels so that no conflict will occur while trying to read the label. The format of the LABEL object is defined by the 2-bits of the Flag (F) field, based on the values of Table 5.1. The Label object class is defined as 16, and the C_Type is defined as WIRELESS_LABEL. The label for a sender must immediately follow the FILTER_SPEC field for that sender in the Resv message, as defined in [11].

5.5.2.3 Hop Count and Sequence Number (HCSN) Object

The HCSN object is optional, and its function is to provide the necessary hop count and message sequence numbering required in the setup of LSPs in mobile ad-hoc networks (MANET). The information contained within this object is used to distinguish multiple receptions (copies) of the same frame in MANET applications. Figure X and Table X show the details of the definition of this new object.

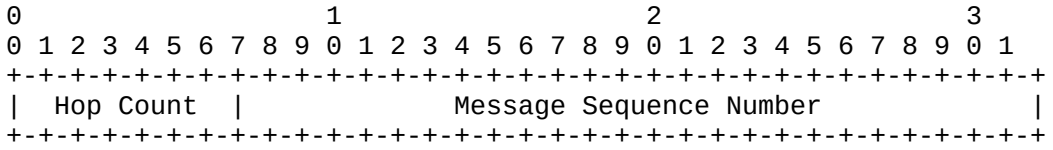


Figure 5.8 – Hop Count and Sequence Number Object format. Class=HCSN, C_Type=HOP_COUNT_SEQUENCE_NUMBER

Field Name	*Description*
Hop Count	Records the current hop count of the message. It must be set to zero on transmission and must be increased by 1 on reception.
Message Sequence Number	Contains the current message sequence number. Each source may begin from a random number and for the next message transmitted it must be increased by 1.
L3PID	Identifier of the layer-3 protocol using this path. The standard Ethertype values used in [11] are also used in this field.

Table 5.4 – Field description of the WIRELES_LABEL_REQUEST object

5.6 WMPLS State Machine

The design of WMPLS begins with the definition of the state machine that represents the entities and relations that will interact in the algorithmic of the protocol. The state machine shown in Figure 5.9 determines the main steps and interactions between the components that govern the functioning of WMPLS. The state machine abstracts the details of the implementation, and provides a complete view of the design and interaction between the system components. Because WMPLS is based on MPLS, the state machine includes only the extensions proposed for the new protocol.

The state machine diagram does not include the signaling protocol mechanisms because the extensions proposed do not modify the signaling algorithms. The proposed extensions modify only the information contained in the label number, and all the relevant steps for using and interpreting this information are shown in detail by the WMPLS state diagram.

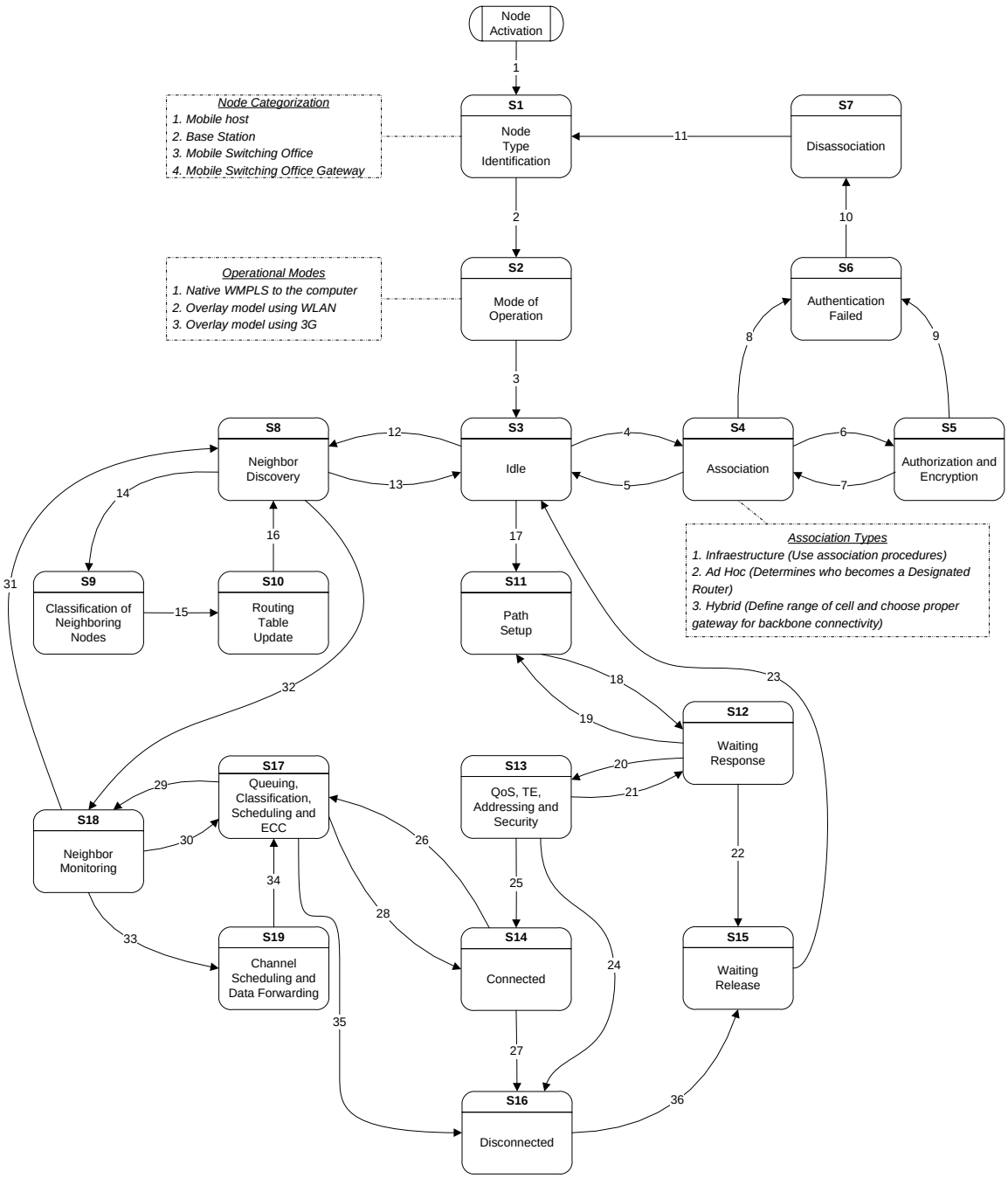


Figure 5.9 – Finite State Machine definition for WMPLS

Table 5.5 shows the summary of the states and their main descriptions. Table 5.6 shows the summary of the transitions and the main functions that control the message exchange between the states.

WMPLS FSM State Description Table	
State Name	***Description***
1. Node Type Identification	Determines the type or role of node the device will be for the entire time that it is part of the network. Nodes are not allowed to switch role types during the time they are part of the network. Any node can take one of the following four personalities: 1. Mobile Host (MH) 2. Base Station (BS) 3. Mobile Switching Office 4. Switching Office Gateway (MSO-GW)
2. Mode of Operation	Determines the type of network that the node will belong to. There are three types of network types that are defined for WMPLS: 1. Native: Provides native WMPLS/MPLS services to the end-user. This implies the user needs to have an stack implementation of WMPLS/MPLS implemented as part of the operating system. 2. Overlay model using WLAN technologies. WMPLS exists on top of the layer that provides the transport mechanisms defined by IEEE 802.11 WLAN standards. The end-user does not need a WMPLS stack implemented on their device. 3. Overlay model using 3G technologies. WMPLS exists over a pre-defined 3G or higher standard.
3. Idle	Any node not participating in any data transport scenario remains in this state. The node constantly listens to see if new LSPs are being created and if it should participate. It also constantly exchanges information with the surrounding nodes for proper maintenance of its routing table.
4. Association	A node enters this state initially when joining a new wireless network domain. The procedures for this process follow the IEEE 802.11 association mechanisms for native and WLAN overlay model. In case of the 3G overlay model, it will follow the procedures defined per standard.
5. Authorization and Encryption	Sub-state in charge of performing authorization and encryption steps for the Association state.
6. Authentication Failed	Sub-state that performs security keys and data structure re-initialization after failure of authenticating with the network.
7. Disassociation	Sub-state that reinitializes the RF associations and resets the configuration parameters of the node to its default values.

8. Neighbor Discovery	This state makes use of known reactive and proactive wireless routing protocols, such as DSDV, AODV, DSR, SSA or WAR [57] (check from Sang-chul's Thesis) to discover the wireless. It relies on wired routing protocols like OSPF to discover the nodes on the wired network.
9. Classification of Neighboring Nodes	Classifies the surrounding nodes and enters and updates the Neighbor Table. This table allows the proper functioning of states 11 and above.
10. Routing Table Update	Updates the Routing Table and allows the node to calculate the position and reachability of other nodes. Additionally provides the calculation of the metrics for proper wireless and wired routing protocol functioning.
11. Path Setup	This state initiates the path setup for a data transfer. The path setup initiates the calculation of the best route (LSP) to initiate the unidirectional link to the requested destination. The calculation is performed based on the information stored in both the Neighbor Table and Routing Table. This state initiates the RSVP-TE signaling to the rest of the nodes.
12. Waiting Response	The node enters this state waiting for the response from the nodes the previous state contacted for setting up the path. When a positive response is received, it will transition to State 13. In case it does not receive a positive response after a Timeout period, it will transition to State 15.
13. Quality-of-Service, Traffic Engineering, Addressing and Security	If a node successfully replied to the request initiated by State 11, the data structures (buffers and queues) are allocated in memory and the scheduler is informed about the traffic prioritization. The remaining Control Plane data structures are also initialized. Additionally, the encryption mechanisms are initialized if the communication needs to be transferred securely. If the host does not have enough computing resources it initiates the tear down of the LSP and moves to State 15. If the data structures and scheduler are successfully updated, it transitions to State 14.
14. Connected	Informs upper layer protocols about the successful data connection and provides the service access point (SAP) handler and identifier in order to start the traffic flow and allocates resources for the Data Plane operations (see Figure 5.3). If the SAP is not accepted or the traffic negotiations have changed, the state transitions to State 16.
15. Waiting Release	Resets the data structures and parameters defined by the Path Setup state and removes any references to any LSP, buffer or queue that were allocated. It also clears any timers that were triggered by State 12. After cleaning it transitions to the Idle State.

16. Disconnected	Resets all the data structures on the Data Plane and the Control Plane. The resources allocated to buffers, queues, QoS and TE parameters, encryption and addressing are reinitialized or released.
17. Queuing, Classification, Scheduling and Error Checking and Correcting	Categorizes the information coming from the upper layer protocols through the SAP provided by State 14, and it starts the classification buffering and scheduling of the traffic before they are transmitted over the wireless medium. In case there is data corruption or inconsistency due to changes in the Neighbor Table or the Routing Table significantly changes, this state transitions to the Disconnected State.
18. Neighbor Monitoring	Verifies that the Neighbor Table and Routing Table have not changed for this LSP before sending the traffic in order to guarantee traffic delivery. If the changes significantly alter the established path, it transitions to State 16, otherwise it transition to state 18.
19. Channel Scheduling and Data Forwarding	Schedules the channel for proper access control and it forwards the packets to the next hop router or node in the LSP. All the contention mechanisms to avoid or detect collisions are provided by additional lower layer MAC protocols.

Table 5.5 – WMPLS Finite State Machine Description Table

The definition of the WMPLS FSM is limited in the fact that it does not allow for nodes to change their types dynamically. The node type definition is done by configuration; however, further work on the protocol definition can extend the capabilities in order to allow dynamic node type definition which can increase network performance and reliability.

Additionally, in an effort to provide network reliability and resilience when delivering traffic, State 11 has the provision to preemptively setup two LSPs, based on the first and second best options of traffic delivery to the end-node. The first option is always the best route that provides the QoS and TE parameters for traffic delivery and it is the one that gets established. The second option connection preempts the resources to be rapidly established in case the first connection is dropped. The second connection might be provisioned by using the Shared Explicit filter FlowSpec defined by RSVP, while the primary connection will most likely will always be provided by the Fixed Filter FlowSpec.

WMPLS FSM Transition Description Table	
Transition	***Description***
1. Init_Node()	Initiates the operation of the node after being powered on and initializes the node to the initial values stored in the configuration.
2. Oper_Mode()	Based on the configuration parameters, this transition determines the operational mode of the node.`
3. Idle_Node()	Once the Control and Data Plane data structures have been initialized, this transition clears the flag on the node indicating its ability to provide communication services.
4. Init_Assoc()	Transitions from the Idle to the Association state when the Idle state receives a request for data transfer.
5. Init_Assoc_Ack()	Transitions from the Association back to the Idle State if the association to another node or a Base Station was successful.
6. Auth_Encr_Req()	If the Association state determines access to the network needs to be authorized and requires encryption it trigger this transition.
7. Auth_Encr_Ack()	This message is sent upon successful authorization and negotiation of the encryption scheme.
8. Assoc_Nack()	If the association process could not successfully contact any surrounding nodes, or if the network is too busy to accept new nodes, this message is sent to State 6 indicating the reason for failure.
9. Auth_Enc_Nack()	This message is sent if the node was not able to associate to any network due to authentication or encryption failures.
10. Auth_Fail()	Message sent to the Disassociation state indicating the failure of association to a wireless network. This message triggers actions to turn of the RF modulator and send a re-initialization message.
11. Re-Init_Node()	Message sent to trigger a complete system re-initialization without rebooting.
12. Req_Neigh_Disc()	After the node is associated to a wireless network, the Idle state requests the neighbor discovery process to begin transmitting data.
13. Rep_Neigh_Disc()	This transition indicates either the success or failure on finding and identifying the neighbors.
14. Neigh_Class()	Triggers the classification of the nodes based on the information provided by the routing protocols obtained by the Neighbor Discover state.
15. Upd_Route_Table()	Once the nodes have been classified and updated in the Neighbor Table, this message is sent to update the Routing Table.

16. Neigh_Class_Done()	This transition indicates the update of the Routing and Neighbor tables has been completed successfully.
17. Init_Path()	When the Idle state receives a positive acknowledgement from the Neighbor Discover state, it initiates the Path Setup process.
18. Path_Wait_Req()	Transition that disables the node from starting a new connection until the current LSP setup process has been completed.
19. Path_Wait_Reply()	Informs the Path Setup State of the successful or unsuccessful path establishment. If successful, the number of active path is increased, otherwise the pending request is cancelled.
20. QoS_Param_Req()	In case of positive acknowledgement for the path setup, this transition triggers the QoS, Te, Addressing and Security State to initialize the data structures required for traffic management in both the Data and Control Planes.
21. QoS_Param_Nack()	This transition informs the Waiting Response State about the failure to establish traffic management parameters.
22. Path_Est_Timeout()	If a path is not setup within the predefined timeout, this transition initiates the release of the resources allocated for the path setup.
23. Init_Path_Fail()	Indicates the failure to establish an end-to-end path. The cause maybe network congestion, failure to provide requested traffic management parameters, node too busy or any other type of software or hardware failure.
24. QoS_Param_Fail()	Transitions to the Disconnected State in case the traffic management parameters cannot be provided by the node.
25. QoS_Param_Succ()	Transitions to the Connected State when the traffic management parameters are successfully allocated.
26. Que_Class_Req()	This transition occurs after all the parameters for the Data and Control planes have been allocated for data transmission.
27. Conn_Fail()	This transition occurs when the data structures and parameters for the Data and Control Planes cannot be allocated or have encountered an error.
28. Que_Class_Ack()	Confirms to the Connected State that the data structures were successfully initialized and data forwarding will take place.
29. Neigh_Mon ()	Triggers the final verification of node reachability before data forwarding begins.

30. Neigh_Mon_Reply()	Informs the Queuing, Classification, Scheduling and ECC State about a change in the Neighbor or Routing Tables. If the changes are not significant the data forwarding might begin.
31. Neigh_Upd_Req()	Triggers and update to the Routing and Neighbor Tables.
32. Neigh_Upd_Reply()	Returns the most updated information in the Routing and Neighbor Tables.
33. Start_Forwarding()	Triggers the data forwarding process.
34. Forwarding_Status()	Informs the Queuing, Classification, Scheduling and ECC State if the forwarding process is occurring properly, or if a failure has been detected.
35. Forwarding_Error()	This transition is triggered in case an error occurred while forwarding data (i.e. the end-node has been disabled).
36. Comm_Error()	This transition indicates that a communication error has occurred and that the resources must be released for further data transmissions.

Table 5.6 – Transition Description Table for WMPLS

Figure 5.10 shows an example of the message sequence chart for WMPLS. From this figure two major phases can be identified. The first phase, the Connection Phase performs the network connectivity tasks. The Data Transfer Phase performs the routing and traffic forwarding tasks. This figure exemplifies the best-case scenario in which a node becomes active, determines the type of network that it will be a part of, associates to the wireless network, discovers its neighbors, sets the LSP for traffic exchange and begins forwarding and receiving data.

Figure 5.11 shows an example that includes and error in the Connection Phase. The node in this figure cannot get properly authenticated and is not accepted in the network.

The node in Figure 5.12 shows an error that occurred in the Data Transfer Phase, in which the node cannot provide the required QoS and TE parameters the connection requires for proper

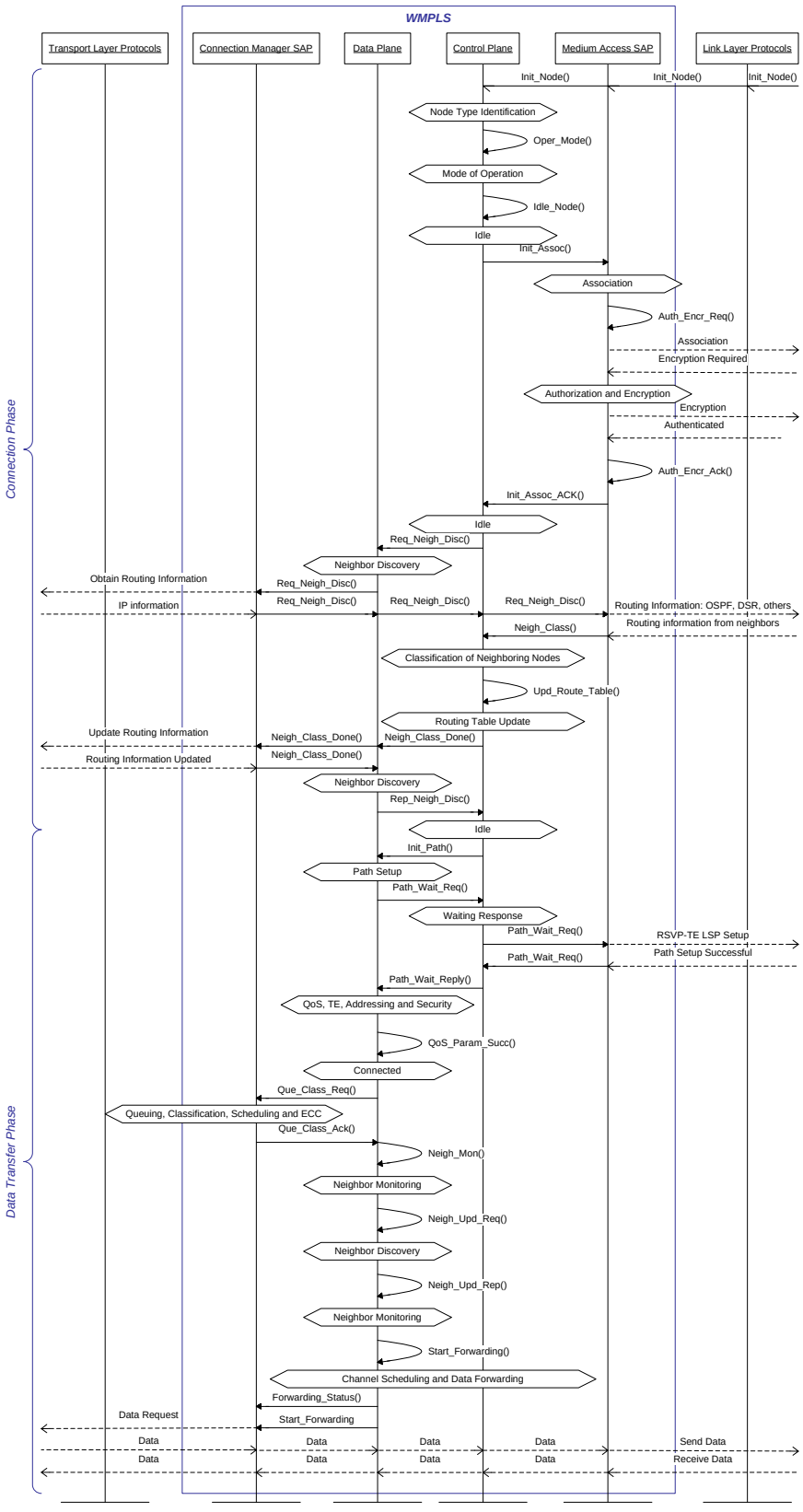


Figure 5.10 – Message Sequence Chart showing the node activation and path setup for the best-case scenario

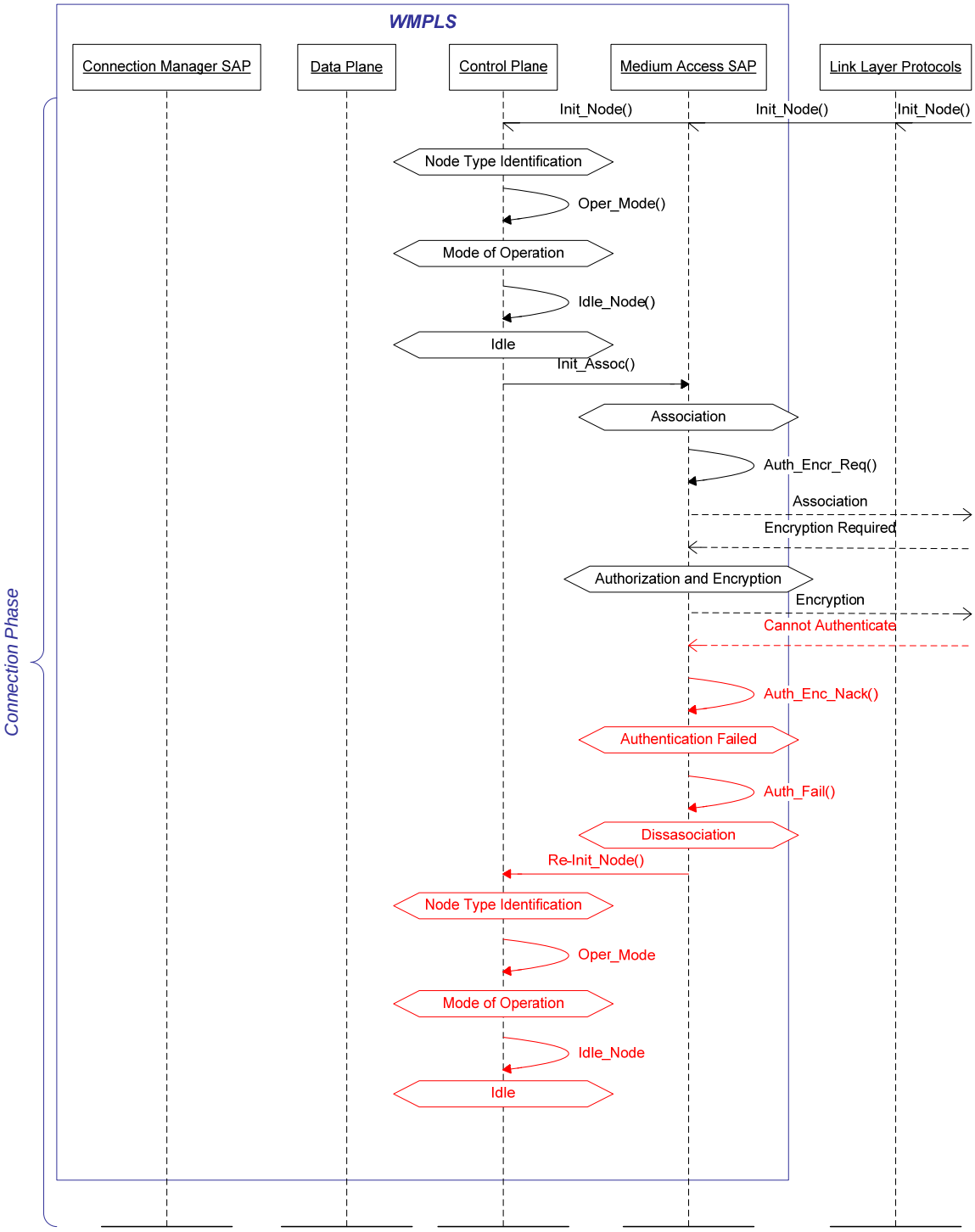


Figure 5.11 – Message Sequence Chart for the Connection Phase with authentication error

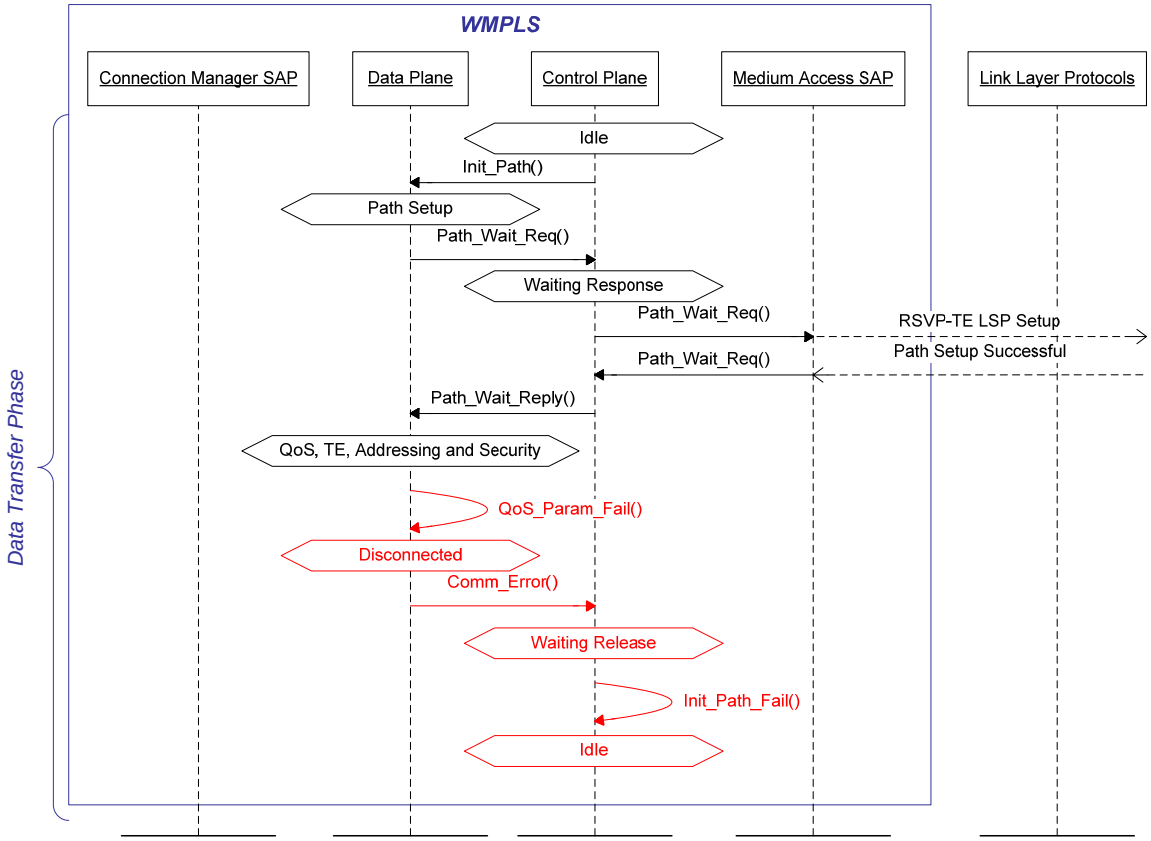


Figure 5.12 – Message Sequence Chart for the Data Transfer Phase with failure to provide required QoS, TE, Addressing and Security

5.7 WMPLS Methodology and Implementation

In an infrastructure-based mobile environment the most important premise of the underlying network is to maintain the communication links for all the nodes connected, especially when they roam between the coverage areas of two base stations or access points. The procedure explained in this section deal with the hand over process in order to rearrange WMPLS communication links between nodes, so that the there is no disruption of the service.

For data communication to occur between two nodes, the forward and the reverse paths can be either symmetric or asymmetric with respect to data rates and/or bandwidth. For example, the bandwidth requirements to download data are commonly higher compared to the bandwidth required to upload requests and control messages. However, this is not true for voice communication, where

the bandwidth required for both forward and reverse paths are the same. For a WMPLS protocol infrastructure to be fully independent of upper layer protocols, the mechanisms in order to provide communication links that can adapt to the needs of the applications needs to ensure that the symmetric or asymmetric characteristics of data exchange can be met, including also the specific QoS requirements that the traffic itself might require. A summary of the terminology used in the upcoming discussion is provided below.

- Mobile Host (MH): A host that is nomadic in nature.
- Base Station (BS): A service provider that governs all the users connected to it.
- Mobile Switching Office (MSO): The routers that provide access points to MHs enabling connection to the network.
- Mobile Switching Office Gateway (MSO-GW): This is an MSO that lies at the border of the mobile communication network and the wired backbone network.

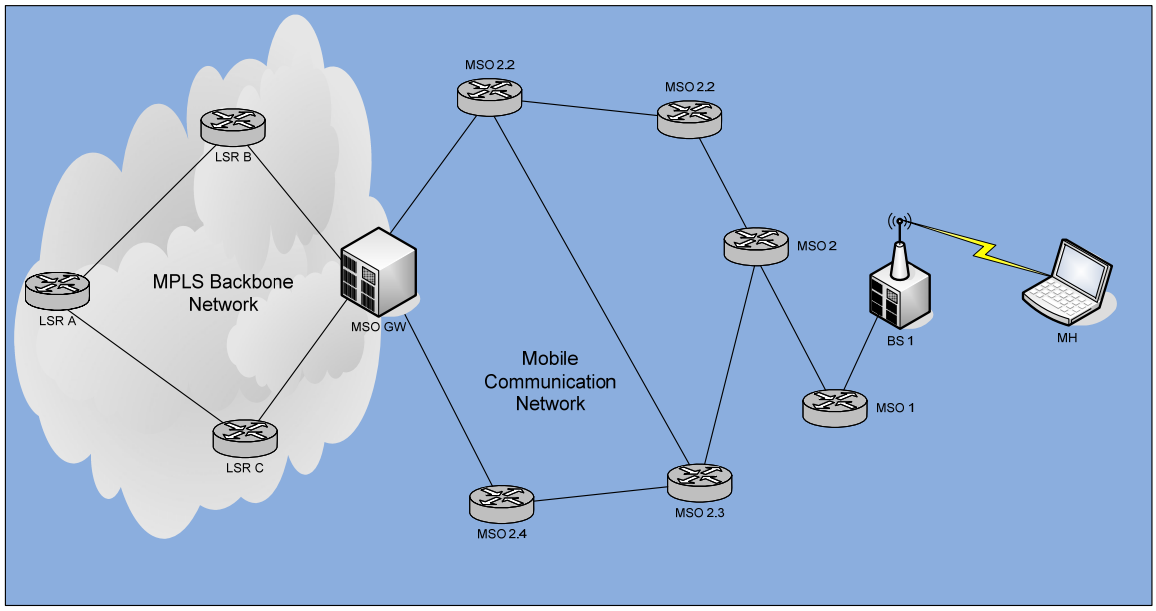


Figure 5.13 – An example of a WMPLS network

WMPLS works with the RSVP-TE signaling protocol, a soft state protocol, which implies that an established LSP has to be refreshed periodically in order to stay alive, which involves control and keep-alive information exchange needed for this purpose. RSVP-TE provides either strictly explicitly routed or loosely explicitly routes [25]. In strictly explicit routing, every intermediate node that involves the end-to-end path is explicitly specified, whereas in loosely explicit routing, not all nodes to be traversed to reach the destination are specified [25]. The nodes that are not specified in the explicit path list in the loose routing mechanism are called abstract nodes. For the design of WMPLS hand over, and other mechanisms, the proposed network configuration applies loosely explicitly routed LSP setup. An example of the topology is illustrated in Figure 5.13.

5.7.1 WMPLS Initial Path Setup

In this section, the initial path setup mechanisms are explained based on Figure 5.13, in which the MH requests a connection to the LSR A. The MSO-GW is an MSO that exists at the border of the mobile communication network and the backbone network, and it provides the bridging capabilities between the wired and wireless domains. Since the MH roams and thus requests connection to different BSs, the path between the MH and the MSO-GW keeps changing. Hence, the path that will exist between the MH and the MSO-GW needs to be defined as the loosely explicitly routed part of the overall LSP that exists between the MH and LSR A. The steps involved in establishing the LSP from the MH to LSR A are discussed in detail in the following paragraphs with reference to Figure 5.14. In the following example it is assumed that the proposed RSVP-TE extensions explained above are being used as the signaling protocol for WMPLS.

1. The MH first identifies and connects to its service-providing base station (BS1). The detailed mechanisms of discovery and connection are left to the lower layer, which provides physical connectivity.
2. The MH requests for a connection to LSR A by sending a Path message to

BS1. Since BS1 is directly connected to MSO 1, this Path message will reach the MSO 1.

3. MSO 1 identifies a path to reach LSR A as MSO 1 → MSO-GW → LSR B → LSR A. Thus the overall path from the MH is MH → BS1 → MSO 1 → MSO-GW → LSR B → LSR A. The path between the MH and the MSO-GW is the loosely explicitly routed part and the path between the MSO-GW and LSR A is the fixed part of the overall LSP from the MH and LSR A, which could be explicitly routed or not.
4. Then, a path between the MSO 1 and the MSO-GW is chosen (see Figure 5.13). In this example, to reach the MSO-GW from MSO 1, there are four possible paths, and it will be assumed that the path selected is MSO 1 → MSO 2 → MSO 2.1 → MSO 2.2 → MSO-GW. Thus the complete overall path ends up being MH → BS1→ MSO 1→ MSO 2 → MSO 2.1 → MSO 2.2 → MSO-GW → LSR B → LSR A.
5. The Path message sent by the MH traverses the selected path through all the nodes until it reaches LSR A.
6. The Resv message is then sent back by LSR A, which traverses the selected path to MH. At all nodes, the reservation and allocation of resources takes place. Labels are also assigned to the individual links that make up the LSP. The path established will support unidirectional traffic flows from the MH to LSR A.
7. Along with the Resv message, LSR A also sends a Path message in order to establish a path from LSR A to the MH.
8. The MH sends back a Resv message as a reply. Then, resource allocation and label assignments for individual links are performed for the path from LSR A to the MH.

As mentioned earlier, the resource requirements for these two paths may be different, thus creating either an asymmetric or a symmetric connection based on the request information.

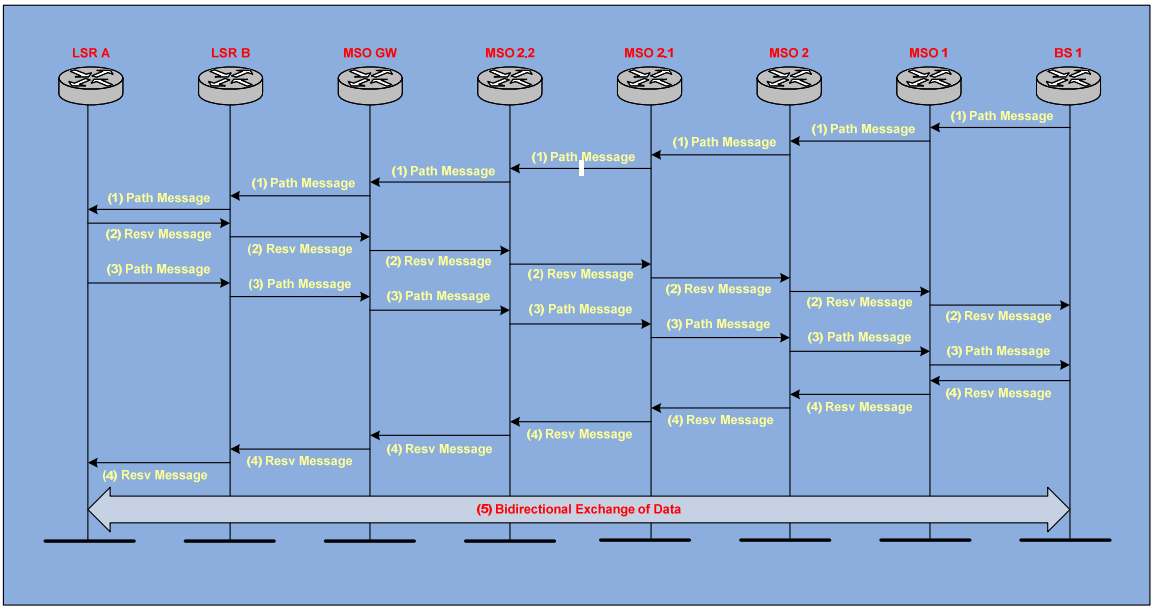


Figure 5.14 - WMPLS Initial Path Setup message exchange

5.7.2 LSP Establishment during Handover

A soft handover procedure is initiated as soon as the need is detected by the MH. When the hand over is imminent, the MH identifies the new BS (BS2) in its reception area, and while the currently established connection through BS1 is still kept alive to receive and transmit packets during handover, the MH tries to establish and alternative path to reach the MSO-GW through BS2 (Figure 5.15 and Figure 5.16). It is assumed that an intermediate router in the loose LSP segment, that can support changes of the handover switching paths, will be identified and used as the connecting point of the changing handover paths. In the operations described below, it is also assumed that the MSOs (which are also LSRs) know about all the network connectivity information obtained by means of the signaling protocol. When the MH decides that a handover is necessary, it will inform both the currently connected BS (BS1) and the new BS (BS2) by sending the handover data and the addresses of BS1 and BS2. This information will arrive at the adjacent MSOs of BS1 and BS2 (i.e., MSO 1 and MSO 1A, respectively). When the request of the MH for handover reaches MSO 1A, it will identify BS1 as being connected to MSO 1 and will also identify the MSO-GW as the router that

the loose path adaptation needs to be requested to. The following paragraphs explain the mechanism of the LSP establishment.

1. The MH sends a Path message to BS2, requesting connection to LSR A. Since the MSO 1A is directly supporting BS2, it will receive the Path message. Since the MSO 1A identifies that the MSO-GW is the common node where the LSPs concur, it selects a path to reach the MSO-GW as MSO 1A → MSO 2A → MSO 2.1A → MSO 2.2A → MSO-GW. The overall path from the MH through BS2 thus being MH → BS2 → MSO 1A → MSO 2A → MSO 2.1A → MSO 2.2A → MSO-GW.
2. A Path message sent by the MH traverses the selected path through the nodes only up to the MSO-GW. The path from the MSO-GW to the LSR A remains fixed.
3. The Resv message is then sent by the MSO-GW, which traverses the selected path in the reverse direction to the MH. At all nodes, the reservation and allocation of resources takes place. Labels are also assigned to individual links in the new LSP.
4. Along with the Resv message, the MSO-GW also sends a Path message (or Label Request message in LDP) in order to establish a path from the MSO-GW to the MH.
5. The MH sends back a Resv message (or Label Mapping message in LDP). Then the resource allocation and the label assignments for individual links are performed for the LSP from the MH to the MSO-GW. (In LDP, the resource allocation would have taken place along with step 4.)
6. Once this path is successfully established, data packets will be forwarded through the newly established path and the former path from the MH through BS1 to the MSO-GW (MH → BS1→ MSO 1→ MSO 2 → MSO 2.1 → MSO 2.2 → MSO-GW) is disconnected.

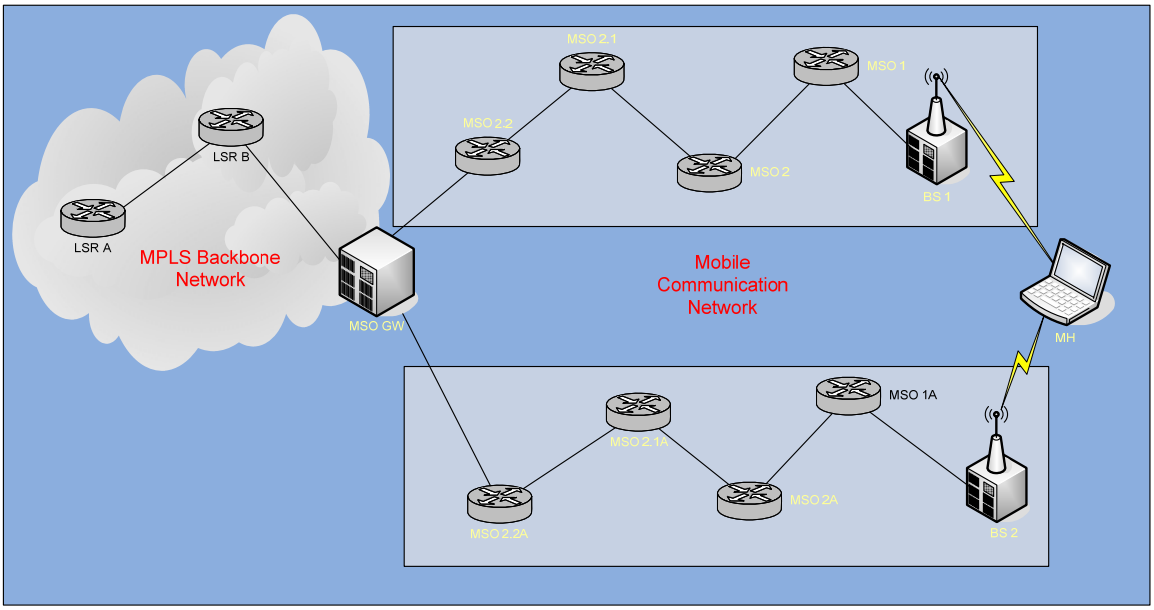


Figure 5.15 – Path establishment during hand over

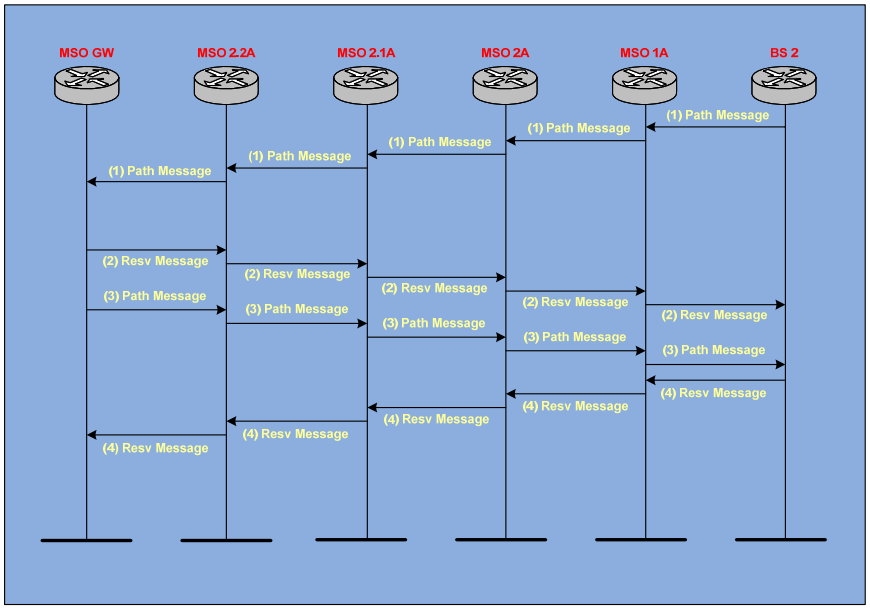


Figure 5.16 – Message exchange and information flow during and after hand over

5.7.3 WMPLS Over IMT-2000

The discussion in this section covers the operations of WMPLS applied to an overlay model having the International Mobile Telecommunications-2000 (IMT-

2000) wireless communication network architecture as the lower layer architecture.

The objectives of IMT-2000 are to provide adaptive communication transfer data rates under roaming conditions, and a peak data rate of 2.048 Mbps under good stationary conditions. To provide QoS, dedicated bandwidth, and differentiated services over this network WMPLS can work in an overlay mode with IMT-2000. This section addresses the interoperability issues between WMPLS and IMT-2000 in order to make the overlay model possible.

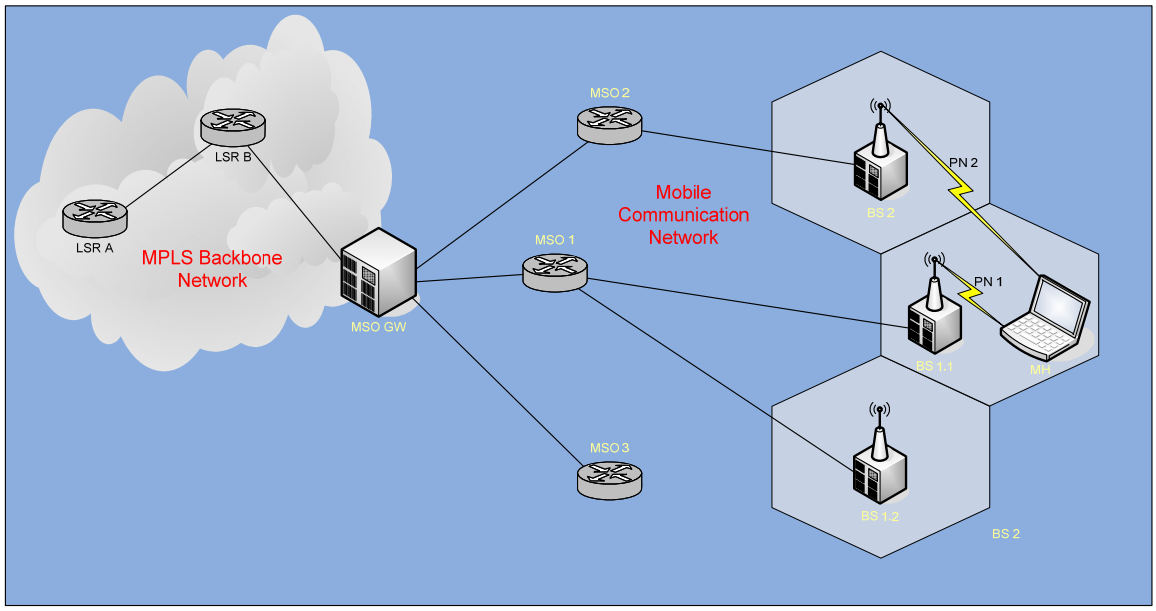


Figure 5.17 – WMPLS over IMT-2000

In the IMT-2000 network, a specific area is split into cells as shown in Figure 5.17. Each cell is identified with a pseudo random noise (PN) sequence being used by the MHs and BS in that cell. Each user is identified uniquely, and valid users are identified using authentication procedures. Since the authentication procedure is conducted at the lower layer (the IMT-2000 layer), there is no need for a separate authentication procedure at the WMPLS layer. After the authentication procedures of the IMT-2000 system have been successfully completed, the mobile system and the BS will be able to send and receive packets.

Through this established channel, the MH sets up a WMPLS path as discussed in section 5.7.1. Any WMPLS packet that is sent is encapsulated within the IMT-2000 protocol payload.

The user authentication will become necessary when WMPLS is used without any underlying wireless protocols; however, the user authentication procedures through extensions of RSVP-TE for WMPLS networks are left for future development.

The following paragraphs provide an example of how an LSP can be set up under the overlay model. This example is based on Figure 5.17. It is assumed that the MH is connected to BS1.1, the PN sequence used for that cell is PN1, and that the MH detects a need for handover and it's expecting the communication link to be handed over to BS2, which belongs to another cell with a different PN sequence, PN2. Assuming overlapping coverage ranges of the BSs, the following steps [24] are carried out:

1. The MH performs the initial cell search, acquires the scrambling code for the BS2, and finds the broadcast control channel (BCCH).
2. The MH then acquires the primary unmodulated synchronous channel (SCH) and obtains the timing information for the secondary SCH.
3. Once the secondary SCH is acquired, the MH achieves the synchronization to BS2.
4. The MH then calculates the timing difference between the two downlinks of BS1 and BS2.
5. MH reports the time difference to BS1.
6. BS1 adjusts the timing of the new downlink soft handover connection to one symbol resolution. BS1 continues to deliver the packets to and from the MH in this adjusted downlink connection.
7. Keeping the BS1 downlink connection alive, the MH establishes the LSP to the MSO-GW through BS2 with the help of MSO 1A. The resource requirements requested of this new path to the MSO-GW through BS2 will

be the same as that of the current path to the MSO-GW through BS1. Since WMPLS uses RSVP-TE as the signaling protocol, such negotiation procedures can be performed during handover in order to get the same TE parameters.

8. Once the new path through BS2 to the MSO-GW with same TE parameters is established, the path through BS1 to MSO-GW is terminated.

It is possible that through the procedures above mentioned soft handover can be achieved in the overlay model of WMPLS over IMT-2000. Similar procedures can be applied when WMPLS is used in conjunction with other wireless technologies in an overlay architecture. One advantage of applying WMPLS procedures over an IMT-2000 infrastructure is that the latter supports the wireless network connection between the BS and the MH, while WMPLS is capable of providing a single or multiple connections over a single wireless link. This capability is especially beneficial when a MH needs to establish a point-to-multipoint connection or when it needs to conduct simultaneous data communication functions of other devices attached to the MH system, whenever the MH is communicating over the network simultaneously with these attached devices. Additional benefits can be obtained when multiple devices may be communicating through a MH using the MH to BS connection as a relay path. Applications of this type are very common in ad-hoc networking, which is explained in further detail in the coming section.

5.8 WMPLS Mobile Ad Hoc Networking Support

This section summarizes part of the research and definitions of the Cambridge Mobile Ad-hoc Routing Protocol as it provided the groundwork for the research presented in this document. References [27], [28], and [29] are related to Mobile Ad hoc Networks (MANETs) and a framework for IP routing in dynamic environments. These concepts are visited in order to present a comparison framework for the new definitions introduced in this document.

5.8.1 Ad-Hoc and Mobile Ad Hoc Networks

A mobile ad-hoc network is characterized by the randomness in its conformance and by the difficulties that imply performing routing tasks in an uncertain, dynamic and fast-changing network environment [26]. An ad-hoc network can be considered as a more generalized concept of a mobile ad-hoc network, in which the mobility of the communicating nodes may or may not be present. The architecture of an ad-hoc network can be primarily described by the lack of a BS that otherwise would serve as a point of contact between the wired and the wireless networks, and that would control and manage its functions. The major disadvantage of an ad-hoc network is that the control and management of the links that make up the network have to be decentralized between all the participating nodes. However, the nonexistence of a base station precludes having to deal with handover issues, bringing down the complexity of the procedures for dynamically changing networks and roaming MHs.

5.8.1.1 Ad-hoc Network Connection Types

The nodes inside an ad-hoc network can be connected in two ways: peer-to-peer or remote-to-remote [26]. The peer-to-peer connection type can be seen between neighboring devices that are not more than one radio hop away. The remote-to-remote connection type relates to the establishment of a route between two nodes that implies using intermediate nodes as part of the path. WMPLS maintains information of the neighboring nodes by means of a peer-to-peer connection, and utilizes signaling protocols (such as RSVP-TE) for permanent, temporary, or momentary connections between source and destination nodes.

5.8.1.2 Node Mobility Types

Depending on the role of the nodes involved and the nature of the mobility (within the ad-hoc network or between ad-hoc networks), the connections will be modified minimally or substantially. For minimal connection impact, which implies a source, destination, and/or intermediate node change in spatial

position, the mechanisms provided by RSVP-TE for route recalculation will be used. Furthermore, since RSVP-TE is a soft-state protocol, a very specific approach will be used in order to maintain the remote-to-remote connections, which is discussed in a later section.

However, for the case in which the mobility implies nodes shifting to and from other ad-hoc networks, a hierarchical addressing methodology that supports dynamic route recalculation between networks is recommended to be used for scalability. Regardless of the node addressing topology applied, the traffic engineering performance and features of WMPLS networking can be supported.

5.8.1.3 Traffic, QoS and Traffic Engineering Parameters

The traffic patterns in an ad-hoc network depend on the type of connection. Additionally, as explained in [26], a hybrid ad-hoc mobile communication pattern that involves fast-moving nodes that create an unstable mobile network can be found. As expected, the QoS and traffic engineering parameters included in a MPLS framework will be highly dynamic and challenging to manage despite the availability of mechanisms provided by the signaling protocols.

An important issue regarding routing also arises. Since any node can become a relay agent for any remote-to-remote connection, traffic aggregation has to be considered. A way to ensure that the communication links will not be disconnected under circumstances in which the network cannot provide the necessary resources to guarantee the QoS and TE parameters, adaptation procedures of the bandwidth, QoS, and GoS must be implemented. In these procedures, reduced QoS and TE values will have to be flexibly and efficiently negotiated. This will reduce the performance of the link but will increase the probability of call acceptance. These concepts then become a key metric for analyzing network performance, that is, the probability of buffer overflow and the delay bound violation probability can be used to determine the appropriateness

of a network design and the validity of the signaling protocols in order to manage dynamic environments.

5.8.1.4 Neighbor Discovery Mechanisms

For any ad-hoc network to function properly, the mechanism for discovering neighboring nodes to establish a communication link must be present. The use of beacon signals [26] for neighbor location purposes, and the use of RSVP-TE Hello messages for exchanging information is a very common way to establish connectivity among peer nodes within a specific transmission range.

5.8.1.5 Size and Bandwidth Utilization

The transmission power for the beacon signal is used to determine the range and diameter of the user discovery area, that is, the potential diameter of the node's connectivity capabilities. The use of the beacon signal enhances the reuse of bandwidth among other nodes inside the ad-hoc network, and most importantly, ensures that a large number of nodes can be part of the ad-hoc network at any time.

5.8.2 WMPLS Ad-hoc Networking Performance Considerations

The performance considerations of a mobile ad-hoc network cannot be compared with the performance considerations of a connection-oriented wired network due to the inherent fading characteristics of the wireless medium, and the existence of noise and interference factors that the wireless channel may experience. Thus, the metrics and parameters considered for performance evaluation are in general considerably different for those already established parameters for wired networks. In [28] there are several performance criteria defined, that are both qualitative and quantitative. Out of this list, several concepts match the design initiatives of WMPLS. For example, WMPLS follows a decentralized and distributed operation scheme based on a neighbor-discovery mechanism.

Additionally, the remote-to-remote links established comprise unidirectional links ensuring loop-free operations. This results from the fact that the LSPs set up using the signaling provided by RSVP-TE are inherently unidirectional. Hence, a route calculation will involve both the downstream and upstream calculations for which the results can be different. Some quantitative metrics used to analyze the network performance are then:

- End-to-end data throughput, delay, and delay violation probability,
- Route acquisition time,
- The percentage of out-of-order delivery of packets [28],
- Buffer overflow probability on a per-node basis.

For this document, however, additional metrics will be involved due to the specific characteristics of a WMPLS mobile ad-hoc network. The metrics considered comprise reliable adaptability to link fluctuations, timely reaction to topology changes, link capacity [26], duration of a remote-to-remote link, and additional load imposed to a node performing relay functions. Some of these metrics have to deal directly with provisioning of QoS and TE guarantees for Differentiated Services (DiffServ), and will be carefully reviewed in the following sections.

5.8.3 WMPLS Ad-Hoc Networking (WMPLS-AHN) Mechanisms

In this section, the specific mechanisms, messages and extensions necessary for establishing ad hoc networks with WMPLS will be explained. As mentioned before, depending on the type of mobility incurred by the nodes (either within or between ad-hoc MANETs) two different link connection and management procedures will be used.

5.8.3.1 WMPLS-AHN within a MANET

The procedures required to establish a working MANET using WMPLS involves four stages:

1. Neighbor discovery
2. Initial route calculation
3. Route re-calculations
4. Route tear-down

5.8.3.1.1 Neighbor Discovery

This procedure allows a node to be aware of adjacent nodes, and does not perform connection setup; it merely collects information of the nodes present and their characteristics. In order to determine if there are any nodes present, the device will use a beacon signal that will be power regulated depending on the density of nodes within the vicinity. If the number of nodes is increased, then the beacon signal transmission power will be decreased in order to minimize the amount of interference among other users. The beacon signal does not carry any information or identification packets and is used only to determine the presence of other hosts.

Once a node is found to be inside the area of transmission, a Neighbor Discovery Message will be issued. This message will contain the information about the issuing node and it will be broadcast with enough power to reach only the adjacent nodes. The Neighbor Discovery messages are not relayed. The receiver of the message will then collect and process the information contained in the message and will store in its Adjacencies Table [28].

The RSVP-TE Hello message enables LSRs to detect its current neighbors [11]. The Hello mechanism enables a LSR to do node failure detection [11]. The RSVP-TE Hello mechanism composes of a Hello message, a HELLO REQUEST object and a HELLO ACK object [11].

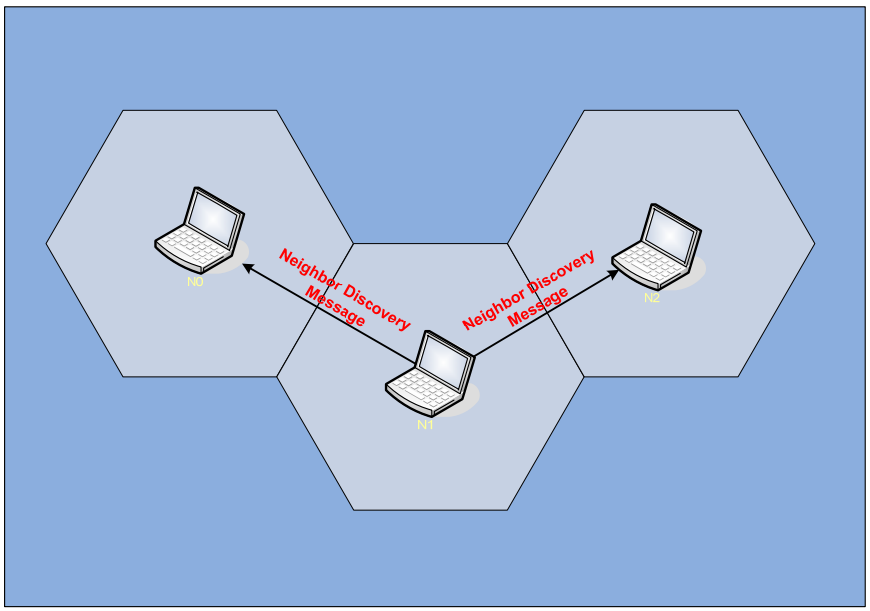


Figure 5.18 – Neighbor Discovery process

Some additional information that may be contained in the Neighbor Discovery Message includes information of the other one-hop-away devices that the issuing node will have at the moment. This allows for devices to maintain a list of devices that are at least one hop distance from neighboring nodes. The procedure for distributing this information resembles the mechanism of the optimized link-state routing protocol [27]; however, the information may include the number of active remote-to-remote links as well as peer-to-peer links, the link direction and characteristics, and a time stamp that will be used when avoiding packet duplication. This information will be used in order to determine the availability of resources for QoS and TE parameter negotiation. Figure 5.18 illustrates the Neighbor Discovery process using the beacon signal coverage area for establishing the adjacencies. Note the Neighbor Discovery Messages are valid only inside a node's coverage area. Upon reception of a Hello Message, the receiving node will evaluate the information and will decide whether to:

- insert the information about a new node entering the MANET
- update the information for a mobile node
- delete the entries associated with a node that has left the MANET

The periodicity of the Hello Message will depend on the number of nodes within the MANET, avoiding flooding of information that can degrade the quality of the transmissions inside the MANET.

5.8.3.1.2 Initial Route Calculation

The initial calculation of a route will be triggered by an upper layer application. The calculation of the route will yield a unidirectional route, as explained before. The source node will issue a Connection Request Message (Path Message for RSVP-TE) in a broadcast fashion for the nodes located within one radio hop. This controlled broadcast is possible due to the information gathered by the Neighbor Discovery process. The initial route calculation message will also include a list of all the nodes that are adjacent to it that should validate the message (Possible Intermediate Node (PIND) list). This PIND list will later be modified and will include only the nodes that can provide the necessary DiffServ requirements, and the nodes excluded from the list will silently drop the message [27].

The Connection Request Message may contain additional information regarding MANET specific details, such as a Request Identifier, a Message Sequence Number [27][28][29], a Hop Count (initially set to 0 since it is being broadcasted by the source node), and the QoS and TE parameters. This information prevents the nodes from duplicating the messages and creating loops, and ensures DiffServ features for the link.

Setting up a LSP can be done in two ways: (i) End-to-End LSP Setup and (ii) Hop-by-Hop LSP Setup.

- End-to-End LSP Setup:
    1. When the intermediate nodes receive a Connection Request Message, they validate the information received in the message by verifying the values. If the values are invalid, the packet will be silently dropped. If the packet is

valid, the receiving node will verify the link operational parameters to see whether the DiffServ request can be supported. If so, the node will update its Adjacency Table information, and will assign a label for the connection being established.

2. The intermediate node then will create a new Connection Request Message and it will send it via broadcast to the adjacent nodes. The message will include updated information about the Hop Count, Message Sequence Number and QoS and TE parameters depending on its own status, but it will keep the Request Identifier for the case in which the source receives the message to silently drop it. For the case in which RSVP-TE is used as the signaling protocol, the Hop Count and Sequence Number Object can be used to provide this functionality.
3. If the DiffServ parameters cannot be met, the node will issue a Connection Response Message (Resv Message for RSVP-TE) with a negative acknowledgment towards the upstream node. This will prevent an upstream node from including this specific host in its PIND list for a random amount of time.
4. The controlled flooding will continue until it reaches the destination node. When the message reaches the destination, the label mappings will be confirmed by a Connection Response Message sent from the destination node towards the source node. This message will be sent directly from the destination node to the source through all the intermediate nodes that the original request message traversed through. In the case that more than one Connection Request Message is received by the destination, the Hop Count and message sequence number value will determine the route to take. If similar routes are obtained, the route will be chosen arbitrarily.
5. The destination node, immediately after issuing the Connection Response Message, will issue a Connection Request message in order to establish another unidirectional path from the destination to the source node. The procedure will be exactly as the one mentioned before, with the difference that the PIND list will include only the intermediate nodes that comprise the remote-to-remote link. If the procedure fails somewhere in the middle,

the decision of choosing an alternative link will be left to the intermediate nodes, and if the process fails, the destination node will initiate a new connection procedure back to the source for reverse path establishment.

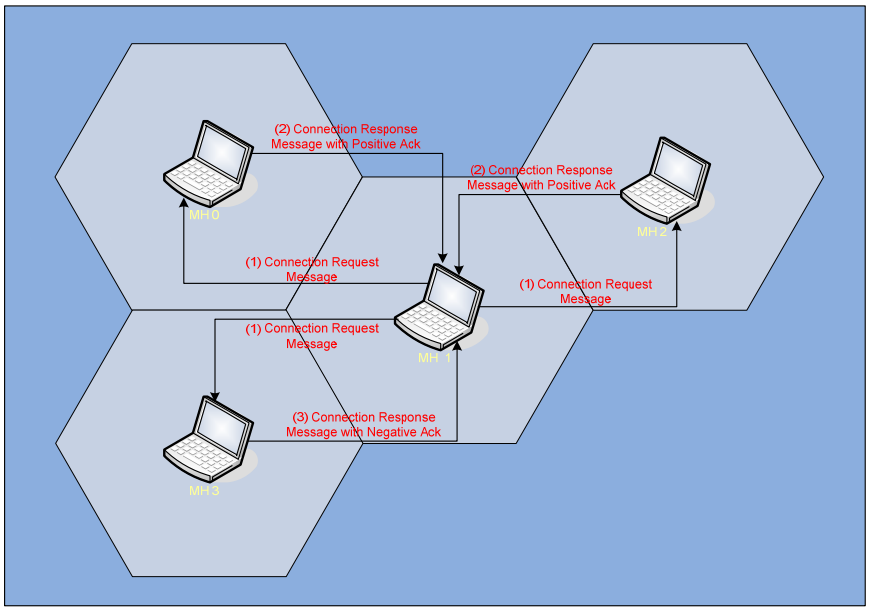


Figure 5.19 – Initial Route Calculation

Figure 5.19 shows the initial route calculation using End-to-End LSP setup procedures. MH0 and MH2 return a positive acknowledgement for the Connection Request Messages, implying that the procedure was valid for the entire route. MH3, however, in this example, returns a negative acknowledgement because it could not find the appropriate requested resources passing through the intermediate nodes to establish a path to MH3.

- Hop-by-Hop LSP Setup:
  1) Step 1 is the same as that for End-to-End LSP setup.
  2) The intermediate node then will send back a Connection Response Message with positive acknowledgement and also simultaneously create a new Connection Request Message with updated Hop Count, Message Sequence Number and QoS and TE parameters. Then the intermediate node may forward this based on a pre-calculated forwarding table

(forwarding case) or may broadcast it to all adjacent nodes (flooding case).

3) Step 3 is same as that for End-to-End LSP setup.
4) Once this path is established all the way to the destination, the destination node will broadcast the Connection Request Message towards the source in order to setup the reverse path to the source.

When the adjacent nodes use flooding, the connection can be established very robustly, although a large amount of unnecessary resources may be wasted. In comparison to this the forwarding topology prevents unnecessary channel resources from being wasted although requires a pre-calculated forwarding table to be prepared. The advantage of using Hop-by-Hop LSP setup procedures is that the LSP setup time is very small compared to that of End-to-End setup procedures. This happens because in the Hop-by-Hop LSP setup procedures the Connection Response Message confirms the reservation and the labels are issued link-by-link as the LSP is being setup. However, in the End-to-End LSP setup procedures, the Connection Response message is sent only after the entire LSP is setup.

5.8.3.1.3 Route recalculations

In the event that one of the nodes starts moving away from its neighbors who it had a peer-to-peer connection with, then the moving node will try to establish an alternative connection between itself and the new neighboring nodes by issuing a Connection Request message. In most cases this may possibly affect the remote-to-remote connection as well. The information about separation between nodes is obtained based on the beacon signal and the Neighbor Discovery messages.

Before establishing the path between the moving node and its peer, the node has to verify that the destination node has not become one of its neighbors, case in which, a direct connection should be established with the destination node.

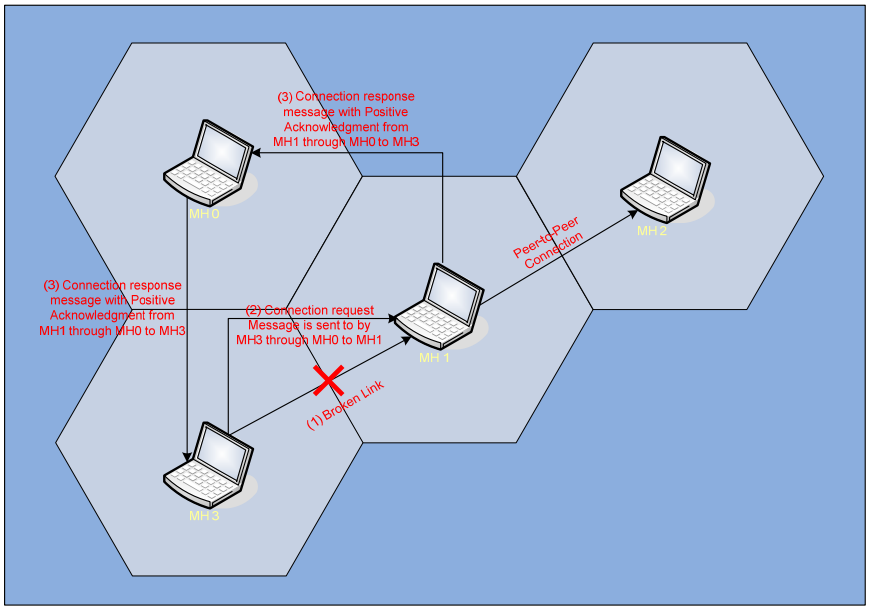


Figure 5.20 – Route recalculation procedure

If a new connection path between the moving node and its peer is found, the traffic is rerouted by means of using RSVP-TE messages. However, if the link is lost, an entirely new connection procedure has to be performed. Figure 5.20 depicts Route recalculation procedures between nodes MH1, MH0 and MH2 using End-to-End and Hop-by-Hop LSP setup procedures respectively. Since MH1 moves away from the coverage area of node MH2, a new intermediate peer-to-peer negotiation has to be done between nodes MH1 and MH0, and similarly between nodes MH0 and MH2.

5.8.3.1.4 Route Tear-Down

A route tear-down is performed when the connection is voluntarily terminated by any of the remote nodes, or when a new route is calculated due to the mobility of any of the participating nodes in the path. This message can also be issued due to an error or inability to provide the DiffServ parameters required for a specific connection.

The Connection Terminate message is used to tear-down the connection between two peer-to-peer nodes, and contains additional information regarding the magnitude of the connection termination. The magnitude can be global or partial. When a voluntary termination or a QoS or TE parameter negotiation failure occurs, a global termination message can be issued, however, when it comes to route recalculation, only a partial route termination message is used.

The Connection Terminate message can be implemented applying several RSVP-TE messages based on the connection status. In RSVP-TE, the Connection Terminate operation can be conducted by the ResvTear message (Msg Type = 6 [11]) that is sent to the upstream LSR. Alternatively a node may terminate its LSP connections using the PathTear message (Msg Type = 5 [11]), which is sent to its downstream LSRs. The PathTear message that is inherent in RSVP removes all the entries in the LSP as well as all reservations.

5.8.3.2 WMPLS between MANETs

Considering the dynamic characteristics of MANETs, cases in which two different MANETs come in close contact with each other and therefore have to merge into one network is possible. However, the assumption of independence between MANETs and any possible interaction between them is defined by the existence of a managing and controlling base station. If there are no base stations defined, the interaction of two MANETs can be seen as aggregation of nodes inside a transmission area in which the procedures mentioned in section 5.7.1 are applicable.

The presence of controlling and managing BSs arise issues relating to handover procedures and hierarchical structures. Based on this reason, WMPLS nodes can achieve remote-to-remote connections among two different MANETs controlled and managed by the base stations. Figure 5.21 shows a hierarchical structure composed of two ad-hoc networks managed and controlled by base stations. Synchronization and handover procedures are provided by the base stations.

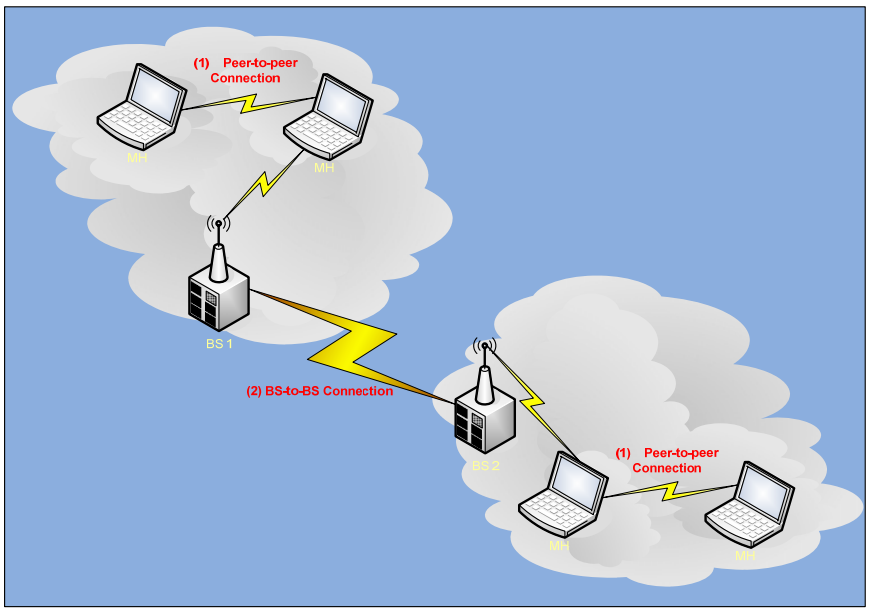


Figure 5.21 – A hierarchical structure

The connection procedures explained in sections 5.7.1and 5.7.1 will be extended in order to support the existence of base stations. Additional mechanisms will also be included that will provide synchronization mechanisms, and are left for future work and research.

6. SCENARIO ANALYSIS AND SIMULATION

Simulation has always been a valuable tool for experimentation and validation of models, architectures and mechanisms in the field of networking. It provides a simple way to test and verify various solutions and scenarios that allow the evaluation of their performance without the need of a real network set up for dedicated experiments [30][31]. In the case of WMPLS, there is not even a real implementation of the protocol, which makes it even more suitable to actually simulate and analyze its behavior and validity in order to provide a refined final design for actual implementation of a prototype.

The simulation will cover several scenarios that will allow the specific analysis of the behavior of the protocol under normal conditions, and also under worst-case-scenario conditions, that will test the robustness and resilience to problems, and will test its capacity to converge optimally.

6.1 Complexity Analysis Metrics

This section includes the definition of the complexity analysis metrics that allow the comparison of WMPLS with other protocols. Most of the analysis in this dissertation is comprised to the message complexity analysis.

6.1.1 Message Complexity

Message complexity, also known as communication complexity [39], is concerned with the number of messages exchanged in order for the algorithm to be completed successfully. In the case of protocol engineering and design, the algorithm is completed successfully in the event in which a communication

channel is established between nodes or peers in a network, or when a message is successfully delivered between two nodes components of the network, or even when an error occurs and communication is lost and the algorithm's mechanism gracefully recover or inform about the errors that are occurring.

In order to fully contemplate the message complexity of WMPLS and use it in comparison to other protocols, the calculations are based on the algorithmic behavior under specific well documented scenarios (usually the best-case scenarios that do not incur in extremely complicated computations and simulations).

6.1.2 Computation Complexity

The computation complexity metrics show the amount of computer or processor cycles that are required to perform the topology analysis of the network and the traffic profiles in order to make an accurate decision regarding the best paths for all the types of traffic, and to maintain the guaranteed parameters for the traffic to traverse the network for a certain amount of time in the future.

One of the biggest challenges when designing WMPLS is to maintain a proper record of the topology and the interconnections of the nodes. The information stored in every node needs to be as accurate as possible in order to guarantee the QoS and TE parameters, while at the same time it needs to keep track of the neighboring nodes location, including an accurate estimate of the capabilities of these nodes so that the signaling of new LSPs, or the update of established LSPs, can be performed as quickly as possible in order to avoid delays in the traffic when nodes change position. For this purpose, the computation of the alternative routes based on variable information regarding the neighboring nodes, needs to be performed as efficiently as possible in order to preempt traffic from being misrouted, or even worse lost because of the failure of an intermediate node, which implies that the algorithms cannot be too complex, which comes at the expense of not providing the most optimal solution for any specific scenario.

The computation complexity analysis of WMPLS is primarily focused to the study of the algorithms in charge of the signaling mechanisms that establish, maintain, and disestablish LSPs. While these tasks are trivial in a wired domain, they become rather complex to manage in a wireless environment because of the constantly changing conditions of the transport medium, and when mobility is considered, the degree of complexity increases exponentially.

6.1.3 Storage Complexity

This measurement is related to the amount of information that is maintained in order to reflect the entire topology of the network, and also the individual characteristics of the links, the traffic types currently supported, and additional management information for control plane mechanisms, and most importantly it contains the label mappings and forwarding information for routing purposes.

The amount of information stored in the memory of the devices affects the overall performance of the system depending on the way it is structured and the time it is needed to access and retrieve it. The number of network nodes and their interconnection will increase the storage complexity exponentially, as more information needs to be stored to maintain the topology of the network. WMPLS needs to keep track of its closest neighbors in order to make correct decision for establishing LSPs, and so it needs to at least have complete information about the nodes one-hop away. Information about nodes further away retrieved during the neighbor discovery process is used to populate the rest of the topology database, which involves that every node will be aware of the entire topology. This involves a very high storage complexity, but it is required to properly guarantee traffic delivery, QoS and TE parameters for traffic that needs it.

6.2 Scenario Definition and Performance Evaluation Criterion

The initial set of scenarios that will be used in order to generate the complexity analysis and the functional simulations are explained in this section. As mentioned above, these scenarios include normal and abnormal operating conditions that can be parameterized by the user in order to make it flexible for all types of analysis. The complexity analysis takes evaluates the worst-case for each of the scenarios defined, which does not imply faulty procedures, but the case in which the message exchange is the most extensive or when the network is very congested.

6.2.1 Scenario 1: LSP Tunnel Preemption and Establishment

This scenario covers the establishment of an LSP using RSVP-TE involving the wired and wireless segments of the network. It includes the negotiation of a single traffic-class LSP using the Fixed Filter (FF) reservation style, and it also covers the setup of an LSP with QoS and TE parameter negotiation. This scenario also provides the capability analysis to see if the proposed extensions can preempt an established LSP tunnel under administrative policy control, using the two types of priority mechanisms (holding and setup priorities). The capabilities are tested on a single traffic-class LSP. This scenario covers a key component of the WMPLS protocol design that allows for a fast recovery mechanism when a network topology change occurs and currently participating nodes of the network move to unreachable positions.

This scenario encompasses the mechanisms that RSVP-TE provides for downstream on-demand label allocation, distribution, and binding of labels.

6.2.2 Scenario 2: Re-routing of LSPs

In this scenario, a link re-route is simulated based on a link or node failure. The establishment of a new LSP based on the FF reservation style is provided as an

alternative, with the inherent analysis of the impact on the traffic resources and convergence time since node failure.

This simulation will also provide the re-routing of the LSP based on the SE filter, in which the LSPID is shared by different links, and the traffic difference is handled by the mechanism provided by RSVP-TE. As with the FF style, the implications on current traffic, QoS and TE parameters, and the convergence time is also evaluated.

6.2.3 Scenario 3: Route Traversal and Identification of LSP Tunnels

RSVP-TE provides the necessary means to keep track of the nodes that comprise an LSP by means of the Record Route Object. This scenario focuses on analyzing the complexity of storing information of all the nodes that are part of an LSP that spans between the wired and wireless networks.

Because the Record Route object is also used for keeping track of the link status information and node failure information, it is used to provide diagnosis information about heavy traffic and congestion that occurs in the network and it is compared to similar mechanisms that are present in other wireless routing algorithms such as AODV and DSR [29]. A key performance evaluator for this scenario is the robustness of the protocol to maintain information flow among all the nodes, and the convergence speed of the network.

6.3 Complexity Analysis of WMPLS

The analysis and framework presented in this dissertation shows the derivation and calculation of the upper bound on the message complexity that characterizes WMPLS compared to other protocols. Since WMPLS is a multi-layer protocol it cannot be easily compared to a specific protocol such as ATM, as is shown in [59], or to other wireless protocols (as shown in [60]), but it needs to encompass a comparison between technologies that specialize in providing services to specific

layer, such as the Logical Link (LL), Medium Access Control (MAC) and Network layers.

For a better understanding of the message complexity analysis, and the upcoming complexity analysis criterion, the definition of a network in mathematical terms is presented next.

Definition 6.1 – A computer network can be represented as a unidirectional graph that can be defined as $G(V,E)$ where V is a set of nodes (which can be represented by $V=\left\{V_1^G,V_2^G,\ldots,V_W^G\right\}$) that contains W elements. E is the collection of pairs of distinct nodes from V that make up a link, which can be presented as $E=\left\{E_1^G,E_2^G,\ldots,E_W^G\right\}$[53]. A connected, acyclic, undirected graph which contains all the nodes is defined as a *tree*. The set of nodes V can be subdivided or partitioned into sub-graphs ($\boldsymbol{V}_1$, $\boldsymbol{V}_2$,..., $\boldsymbol{V}_n$), and each one of them is called a *free tree* $P(V,E)$ and $\left|\boldsymbol{V}_1+\boldsymbol{V}_1+\cdots+\boldsymbol{V}_n\right|=W$. All messages exchanged between peers are unidirectional, corresponding with the type of the graph definition of the network. ∎

Each node in the network strictly follows a *procedure*, which is a *sequence of steps* in the algorithm. Each step leads the node to make a general decision that will either trigger or no a message, whether to take the steps in the procedure or not (which is called *recursion*), whether to fork or branch to a different procedure or not, and whether or not to stop the execution of a step. The method of adding the upper bounds of the time complexity measured at each step can be adapted in the proposed algorithm since the WMPLS algorithmic is composed of a sequence of discrete distinctive procedures where each step has its own message complexity. Therefore, by adding the message complexity measured at each step, the message complexity of the entire procedure can be calculated. Correspondingly, the method of adding the time complexity for each node can be extended in the case of n nodes because WMPLS is composed of recursive

algorithms. Therefore, by adding the message complexity measured at each procedure for each node, the message complexity of WMPLS can be calculated.

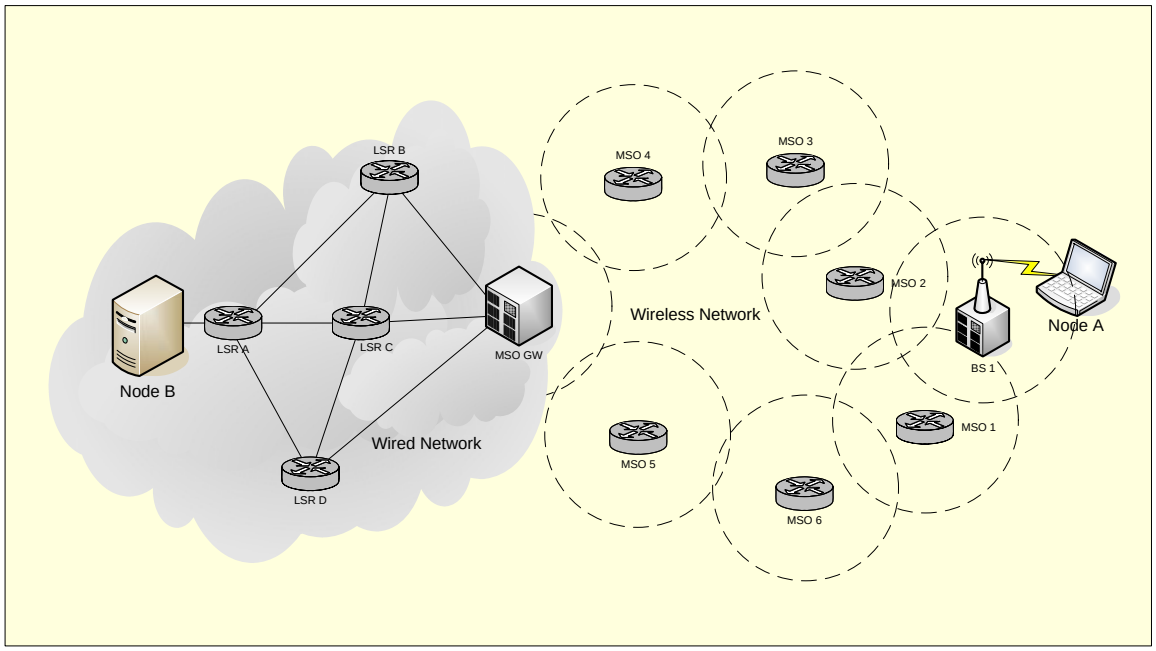


Figure 6.1 – Network layout for complexity analysis

Each of the scenarios mentioned in the previous section includes definitions and some Axioms and their proofs for their operational description. Whenever a network node sends a message based according to the protocol specification, the number of nodes in the free tree that are able to participate of the step are initially calculated. When the nodes that compose the free tree send a message, the nodes identically adapt to the MAC layer protocol to effectively share the wireless medium to communicate with each other, which means that the nodes will conform a homogeneous network. Based on the number of nodes, the maximum number of messages is calculated where the duplicated messages that traverse the network are only considered once. An upper bound calculation of the number of messages is obtained by means of the *big O-notation* [54]. Figure 6.1 shows a network layout that will be used to illustrate the algorithmic for the different scenarios.

6.3.1 Scenario 1 – LSP Tunnel Preemption and Establishment

For the complexity analysis in this scenario, the following definitions are presented:

Definition 6.2 – The process of initiating a connection to the network by the *Neighbor Discovery* (ND) process by means of broadcasting a *Discover Message* (DM) is defined as an *attempt*. A *successful attempt* is defined in the event in which the *Node Found* (NF) message is received from at least from two different nodes within the established time frame before triggering a timeout. An *incomplete attempt* occurs if only one node replies to the DM request and a *failed attempt* when no replies are received before the timeout period. ∎

Definition 6.3 – A *Session* is defined as a sequence of *attempts* that might be of one of the cases mentioned in Definition 6.2. The number of procedures executed in any *session* is defined as n. ∎

Definition 6.4 – When computing the upper bound for the message complexity, the worst case for a *successful session* is composed of $n-1$ consecutive *failed* or *incomplete attempts* and a *successful attempt* at the $n-th$ attempt. The failed session is composed of n *failed* or *incomplete attempts*. ∎

Based on a node that sends a Discover Message in a free tree $P(V,E)$, the identification of the transmitting nodes is rearranged in ascending order such that u nodes, which are $\{V_1,V_2,\ldots,V_u\}$, broadcast or relay an DM message and $n-u$, which are $\{V_{u+1},V_{u+2},\ldots,V_n\}$ only relay the DM message to other nodes.

Since each node in the network or free tree will relay the Discover Message initiated by node V_i, and assuming that no duplicate messages are relayed, the maximum number of network nodes that will rely a DM message is $n-1$.

Therefore, the maximum number of Discover Messages that are sent over the free tree is given by n, which can be represented as $O(n)$, which can be rewritten as:

Axiom 6.1 – For a wireless network with N nodes, the upper bound value for the message complexity for the Discover Neighbor message delivery process is given by $O(N)$. ∎

The DM message will generate a response from the neighboring nodes that are willing to participate in conjunction with it. In the worst case scenario, the node that is the furthest located from the originator of the discovery process might want to respond. For this case, the number of nodes participating of the longest path is defined by t. The maximum number of nodes in the return path of the NF message is composed by the nodes in $\{V_j, V_k, \ldots, V_i\}$ and the number of hops in the path is given by $d(j,i)$. This analysis leads us to the following axiom.

Axiom 6.2 – For a wireless network in which the longest path is composed of t nodes, the upper bound value for the message complexity of the *Node Found* message return process is given by $O(t)$. ∎

Figure 6.2 shows the flowchart for the operations of the *Discover Neighbor* and *Node Found* message exchange, which together form the *Neighbor Discovery* process defined by State 8 defined in the FSM of WMPLS.

From the flowchart we can derive the upper bound value for the complexity analysis in two cases. The first case occurs when no messages are received before the timeout period for a number of attempts that are defined by the *Max_Retry_Threshold* parameter represented by m. Thus, this represents the message complexity analysis for a failure-type case.

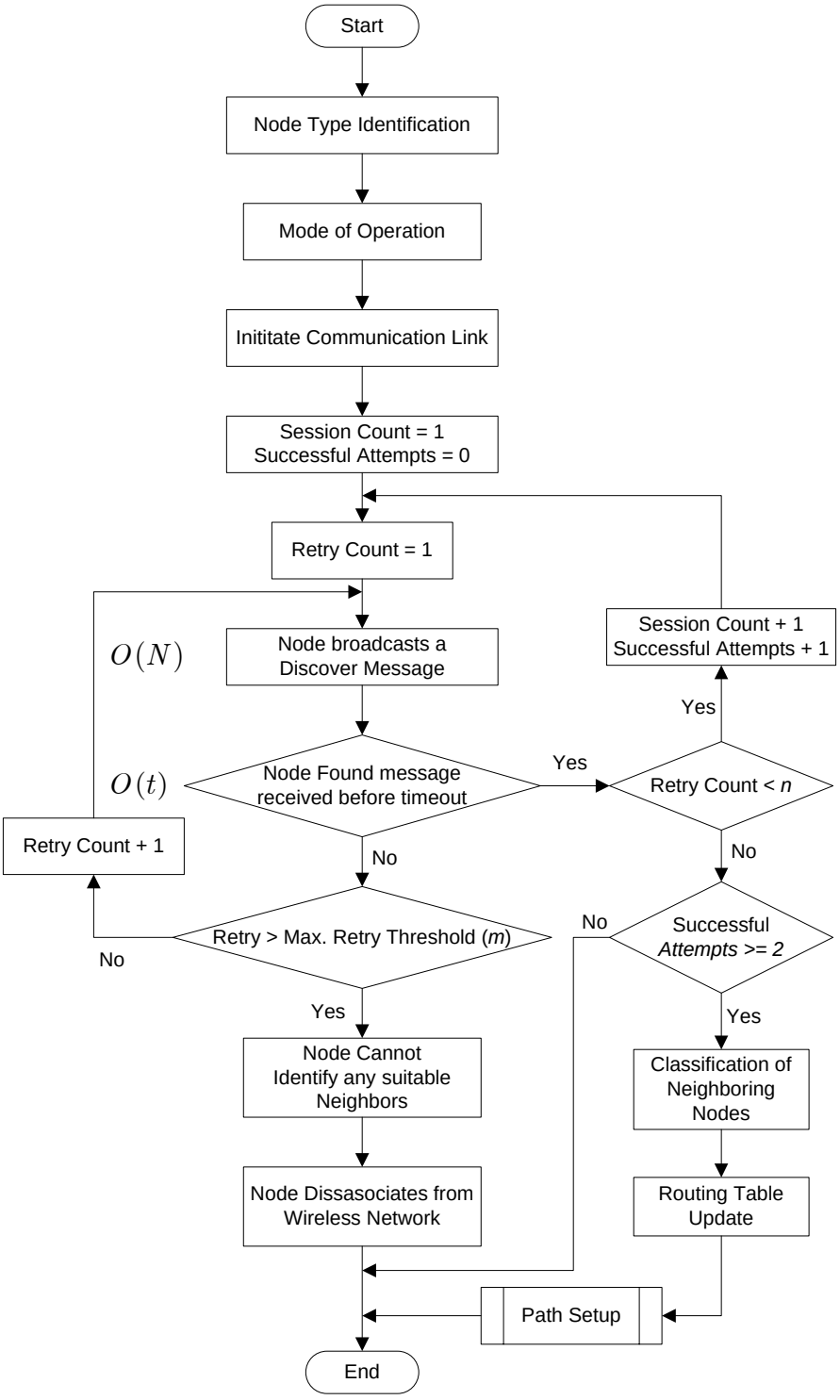


Figure 6.2 – Neighbor Discovery Process flowchart for Scenario 1

The other case presented by the flowchart analyzes the circumstance in which the rest of the nodes respond within a number of sessions that is defined by n. This represents the upper bound value for the complexity analysis of the success-type case. Table 6.1 shows the results obtained.

Neighbor Discovery Failure		Neighbor Discovery Success	
$m \cdot O(N) + O(t)$	(6.1)	$n \cdot [m \cdot O(N) + O(t)]$	(6.2)

Table 6.1 – Upper Bound Values for the Neighbor Discovery Process Message Complexity Analysis of Scenario 1.

Equations (6.1) and (6.2) provide the complexity analysis for the Neighbor Discovery process only. In the case in which there are at least 2 neighboring nodes, the calculation of the upper bound value of the message complexity needs to be extended to reflect the RSVP-TE negotiation for the LSP setup.

Definition 6.5 – The Path Setup process begins with the two *Path Messages* (PM) being sent from the source to the destination, which is defined as an *LSP setup attempt*. A *successful LSP setup attempt* is defined in the event in which the destination node processes these messages successfully along all the intermediate nodes in the path and issues two *Resv Message*. The same process is performed in the reverse order to accomplish a bi-directional communication link. A *failed LSP setup attempt* occurs when neither the initiating LSP setup attempt nor the responding LSP setup attempt can be successfully established. ∎

The Fixed Filter (FF) reservation style used for the one of the Path Messages and the Shared Explicity (SE) style for the second one for the path establishment defined in this scenario. The FF defines that a unique label and unique resource reservations are assigned to each sender. The FF style implies that there is no resource sharing and no merging of LSPs. The SE filter, on the other hand, allows the sharing of resources and merging of LSPs, which is particularly helpful for re-routing techniques (which will be explored in Scenario 2), and provides the LSP preemption mechanism.

Definition 6.6 – An *LSP Session* is defined as a sequence of *LSP setup attempts* as stated in Definition 6.5. The number of procedures executed in any LSP *session* is defined as q. ∎

Definition 6.7 – When computing the upper bound for the message complexity for the Path Setup process, the worst case for a *successful LSP session* is composed of $q-1$ consecutive *failed LSP setup attempts* and a *successful LSP setup attempt* at the $q-th$ trial. ∎

The number of nodes participating of the Path Setup process is defined by u. The maximum number of nodes in Path and Resv message exchange is composed by the nodes in $\{V_j, V_k, \ldots, V_u\}$ and the number of hops in the path is given by $d(u, j)$. This analysis leads us to the following axiom.

Axiom 6.3 – For the Path Setup process that is comprised of u nodes, the upper bound value for the message complexity for the *primary LSP setup attempt* is given by $O(u)$. The Preemption mechanism is provided by the *secondary LSP setup attempt*, which allows fast LSP recovery in case the primary LSP fails. The upper bound value for the message complexity for the *secondary LSP setup attempt* is also given by $O(u)$. The total upper bound value for the message complexity of the *successful LSP session* is thus $2O(u)$. ∎

Figure 6.3 shows the detailed flowchart including both the Neighbor Discovery and the Path Setup process. From the flowchart we can derive the total upper bound value for the complexity analysis for the both the failure and success cases. For the calculations p represents the value of the maximum number of retries per Path Message requests that are not successfully replied, and q represents the maximum number of *LSP Sessions* attempted. Table 6.2 shows the final results for scenario 1.

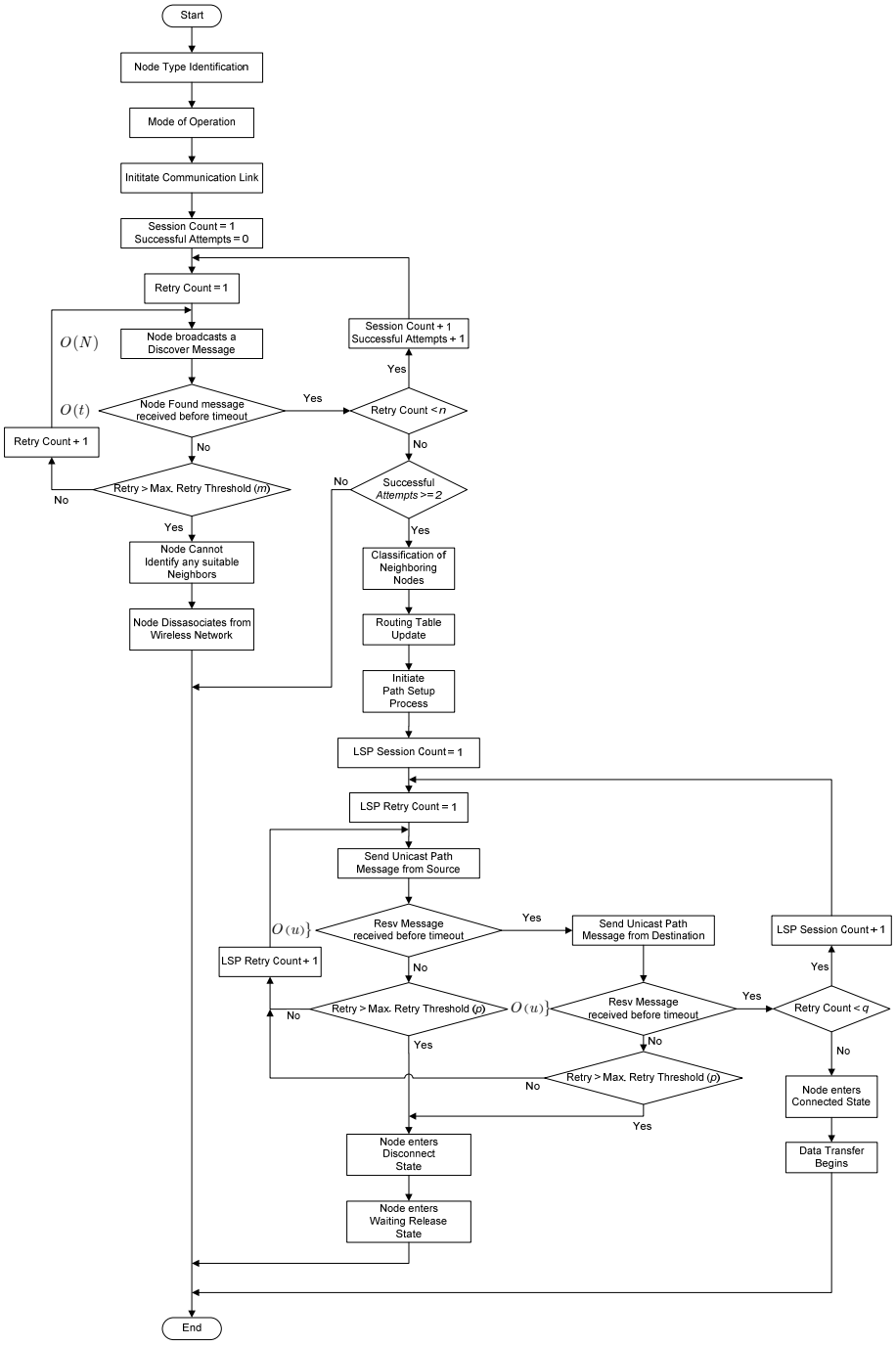


Figure 6.3 – Neighbor Discovery and Path Setup flowchart for Scenario 1

Failed LSP Establishment	Successful LSP Establishment
$m \cdot O(N) + O(t)$ (6.3)	$n \cdot \left[m \cdot O(N) + O(t)\right] + q\left[2p \cdot O(u)\right]$ (6.4)

Table 6.2 – Upper Bound Values for the Neighbor Discovery Process Message Complexity Analysis of Scenario 1.

Equations (6.3) and (6.4) show the total lower bound and total upper bound values respectively for the entire LSP Setup process of WMPLS.

6.3.2 Scenario 2 – LSP Re-Route

The LSP Re-Route process implies that a previously established LSP has either stop working due to failure in one or more intermediate nodes and a new establishment is required. To maintain network sanity and resource allocation efficient, the LSP Re-Route procedure must first release all the resources allocated to the previous LSP and then it must reestablish another LSP. During the LSP teardown process, the traffic is sent over the secondary path created during the Path Setup process, using the SE reservation style. Then, a new Path Setup process is initiated and traffic is re-routed from the SE reservation style to the newly established LSP. Finally, the remaining secondary LSP is also tear down.

Definition 6.8 – The Re-Route process begins with sending a *PathTear Message* (PTM) from the source towards the destination in order to free up the resources for the unidirectional link downstream. A *successful LSP Teardown attempt* occurs when the destination responds this message with a ResvTear (RTM) message. A *failed LSP Teardown* occurs when no message is received from the destination node after the timeout period expires. After a successful LSP Teardown concludes, a new Path Setup process is initiated and when established, the remaining secondary LSP is also subject to the first part of this definition. ∎

Definition 6.9 – An *LSP Re-Route Session* is defined as a sequence of *LSP Teardown attempts* as stated in Definition 6.8. The number of procedures executed in any LSP *session* is defined as n. ∎

Definition 6.10 – When computing the upper bound for the message complexity for the LSP Re-Route process, the worst case scenario for a *successful LSP Teardown session* is composed of $n-1$ consecutive *failed LSP Teardown*

attempts and a *successful LSP teardown attempt* at the $n-th$ trial. ∎

The number of nodes that participate of the LSP Re-Route process is given as two sets, the first one includes the current nodes, and the second one includes the new set of nodes for the new LSP. The current set of nodes is defined by u, and the new set of nodes is defined by v. The maximum number of nodes in this process is composed by the nodes in $\{V_j, V_k, \ldots, V_u, V_v\}$ and the number of hops in the initial path is given by $d(u,j)$, and the number of hops in the new path is given by $d(v,j)$. This analysis leads us to the following axiom.

Axiom 6.4 – For the LSP Re-Route process that is comprised of two sets of u old nodes and v new nodes, the upper bound value for the message complexity of the *successful LSP Re-Route session* attempt is given by $2[O(u)+O(v)]$. ∎

Figure 6.4 shows the detailed flowchart including both the LSP Re-Route and the Path Setup processes. From the flowchart we can derive the total upper bound value for the complexity analysis for the both the failure and success cases.

Failed LSP Re-Route		**Successful LSP Re-Route**	
$2m[O(u)]$	(6.5)	$(n+r)[2m \cdot O(u)] + q[2p \cdot O(v)]$	(6.6)

Table 6.3 – Upper bound values for the message complexity analysis of the LSP Re-Route process

The calculation of the message complexity for the LSP Re-Route does not include any Neighbor Discovery process. In a real wireless network the dynamicity of its conformance make it very difficult to be able to predict how nodes will behave, thus the message complexity analysis of the LSP Re-Route process can be very complex. For this reason, the analysis of the LSP Re-Route process only covers RSVP-TE signaling messages.

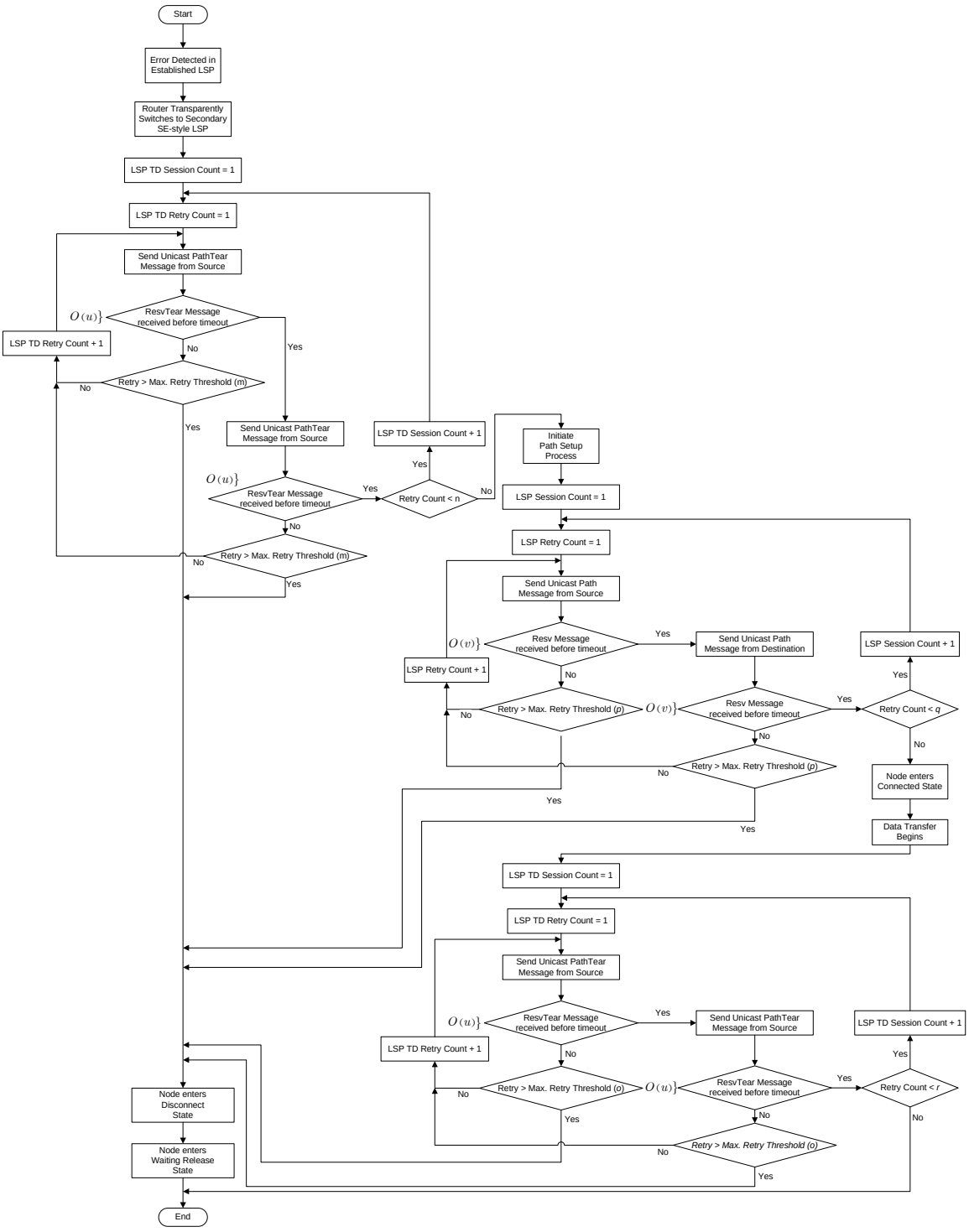


Figure 6.4 – Flowchart for the LSP Re-Route process

6.3.3 Scenario 3 – Route Traversal and LSP Identification

The Record Route object defined by RSVP-TE allows the capability of keeping track of the nodes which the message has been routed through. At the transit nodes, the Record Route object from the Path and Resv messages is combined to see the full route. This object carries the precise sequence of addresses necessary to form a strict explicit route that follows the course of the LSP, and this permits the establishment of a full-duplex path from the point of view of the intermediate nodes in the network.

Since a WMPLS must keep track of all the routes for which is part of, and also the LSPs to which it belongs, the Route Traversal process is a key component that allows each WMPLS node to be aware of not only its neighbor nodes, but of the network as a whole. The following definition provides the starting point for calculating the storage complexity that keeping track of this much information implies compared to other protocols.

Definition 6.11 – The Route Traversal and LSP Tunnel Identification process performs a *Promiscuous Listening* mechanism that obtains and stores information from neighbor propagation messages and the information from the Record Route Object. It identifies the communication pairs that compose the network and stores the information of the whole topology of the network. ∎

The complexity analysis is performed over a network that is composed of N nodes and e communication pairs. To calculate the upper bound computation complexity analysis for the Route Traversal and Identification of LSP Tunnels scenario, the amount of information that each node stores needs to be initially analyzed. The following Axiom provides the

Axiom 6.5 – For the Route Traversal and LSP Tunnel Identification, the storage complexity of a single WMPLS node is given by $2O(N) + 2O\left(e^2\right)$. ∎

Proof – Each node keeps track of its neighbors and stores this information in the Neighbor Table, which at most will contain each node in the network, thus it will be of complexity $O(N)$. Additionally, each node keeps track of the routing information provided by routing protocols, such as DSR, AODV or OSPF, in the Routing Table, and at most this table will keep the information of all the routes in the network. Thus, storing routing information has a complexity of $O(N)$ [40]. Because the node stores the information for each communication pair, which implies every unidirectional LSP, it will store not only the information of the link, but also the QoS and TE parameters. This results in a storage complexity of $O\left(e^2\right)$, which must be doubled to include all communication pairs. ∎

Based on Axiom 6.1, the computation complexity can be defined and calculated for each WMPLS node in the following manner.

Axiom 6.6 – For the Route Traversal and LSP Tunnel Identification, the storage complexity of a single WMPLS node is given by $O\left(N^2\right)$. ∎

The proof of Axiom 6.6 is trivial, because the calculation of the computation complexity is based on the highest-order component of Axiom 6.5.

6.4 Simulation and Performance Evaluation

This section contains the results obtained from the WMPLS simulations related to the performance evaluation metrics defined in Chapter 3. The results gathered include the comparison between WMPLS and transport technologies like WATM, and/or routing technologies like DSR and AODV. Because WMPLS is a multi-layer technology, some of its operations can be directly compared to other technologies, but some mechanisms that imply a cross-layer design cannot be easily compared and are analyzed without providing a comparison reference. The Monte Carlo simulation methodology has been used performing 1000 iterations per scenario (in which the number of nodes varies) and provides the results ont

the average to avoid any biasing the algorithm might suffer. The simulation is an implementation of the FSM presented in Figure 5.9 based in C++, the Network Simulator 2 (NS-2) and MatLab. The programming of the message exchange process is performed in a multi-threaded environment using TCP/IP socket programming under a Linux environment.

For the implementation of the simulation the following parameters were defined. The maximum buffer space for any network node is given as 2 Megabytes, primarily because Cisco routers reserve this amount of DRAM memory for buffering tasks [61], and this way the simulation resembles real world characteristics.

The number of different types of traffic was defined to be 4 in order to match the lowest common denominator in order to obtain comparable results. In this case, ATM defines four primary types of traffic: CBR, VBR, VBR-RT and ABR/UBR (which are considered as one as both are best-effort delivery and their difference is not relevant for the present analysis).

For the WMPLS case a four-level priority scheme is implied, and the criteria used for each is based on the committed information rate (CIR) parameter, which is defined by the Bit Error Rate with the following values 10^{-12}, 10^{-9}, 10^{-6} and 10^{-3}, that will be used to match the ATM traffic types.

The simulation is composed of a wireless and wired hybrid network, in which the routing information is provided by OSPF and DSR for the wired and wireless domains respectively. The wireless network spans a total area of 1 km^2. The wireless nodes have an effective radiation range of 100 meters, and the retry parameters are defined as follows:

- Maximum number of errors allowed per attempt = 3
- Maximum number of attempts per session = 5
- Timeout period = 30 seconds

The rest of the parameters vary in order to provide proper results for each of the analysis that can be performed after running the simulation.

6.4.1 Buffer Sharing and Packet Policing and Discarding Analysis

The design of WMPLS requires the use of a Buffer-space partitioning that allows independent buffers for different types of traffic. WATM was designed with a shared buffer mechanism [26], which makes no distinction between buffers of different traffic types, and depends on the ATM Adaptation Layer to provide traffic differentiation.

The simulation implements both these schemes, and the information is collected statistically while the simulation is running. The results from the simulations are then compared with the values obtained from equation (3.1), as shown in Figure 6.5.

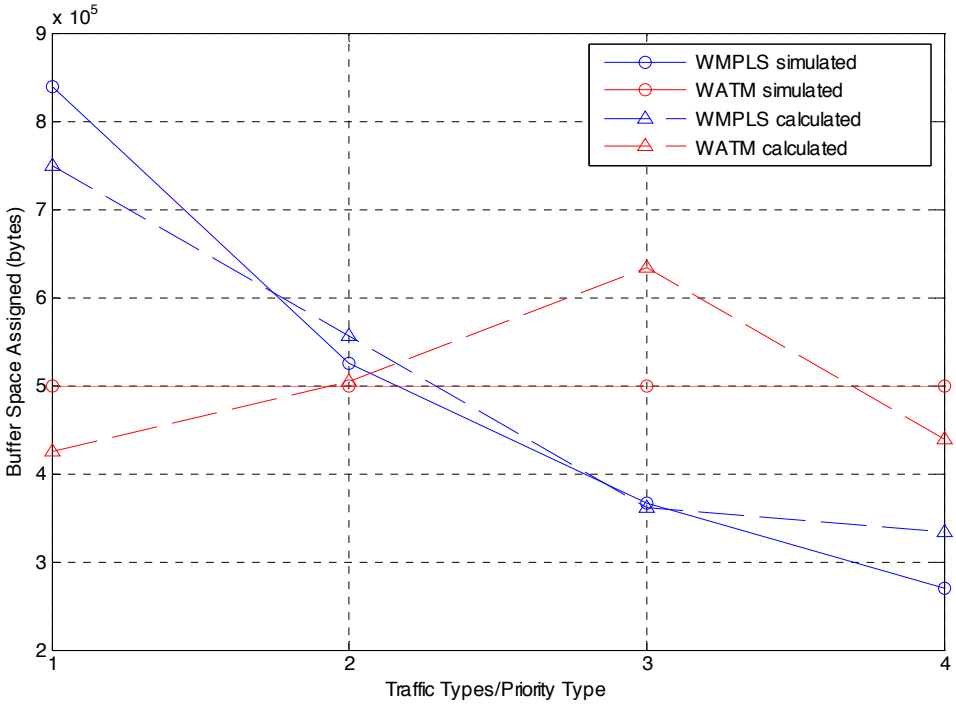


Figure 6.5 – Buffer sharing and partitioning comparison results between WMPLS and WATM

From the obtained results it can be observed that WMPLS provides a better buffer partitioning scheme compared to WATM. It is clear that due to traffic type definitions WATM provides a higher buffer allocation for real time traffic in order to minimize jitter compared to constant bit rate traffic (which is not necessarily affected by delay jitter). WMPLS defines the queuing priorities based on the commited information rate parameters that it has predefined for each traffic class, and the buffer allocation is very straightforward with it. The simulation implies a Random Early Discard (RED) mechanism for both WATM and WMPLS, which determines the assignment of memory to different buffers.

The simulated results vary significantly in the case of WATM because for the results obtained during the calculation of equation (3.1), the inherent traffic characteristics are not incorporated, which does not happens on the simulation.

6.4.2 Buffer Overflow and Delay Bound Violation Probability Analysis

The simulation collects information about the packets that are dropped on average by the buffer-space partition used by WMPLS or by the shared buffer space used by WATM. The delay bound violation probability and the buffer overflow probability profiles from the simulation obtained are shown in Figure 6.6. and Figure 6.7 respectively.

The results show that WMPLS performance is significantly better than the performance of WATM when it comes to provide lower delay violation probability. This is mainly due to the queuing discipline that is used in WMPLS compared to that of WATM. WMPLS provides a Classification, Queuing and Scheduling (CQS) discipline that allows traffic management to be significantly better. Figure 6.7 shows the results for the buffer overflow probability analysis, and the results obtained by the simulation are confirmed by the analysis of the buffer partitioning mechanism, in which WMPLS provided a better buffer partitioning scheme than WATM.

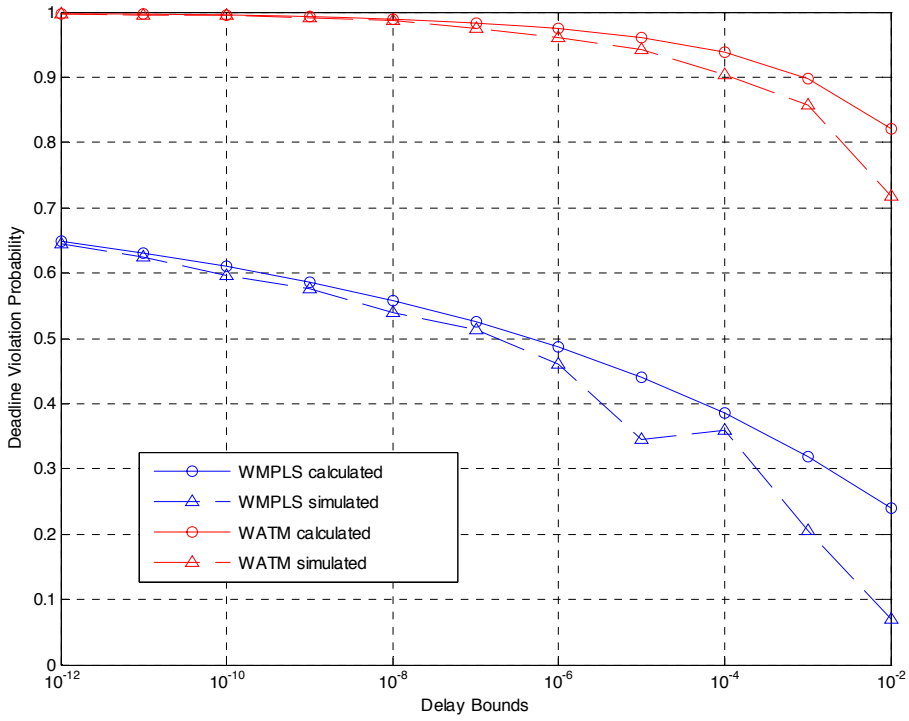


Figure 6.6 – Deadline violation probability comparison between WMPLS and WATM

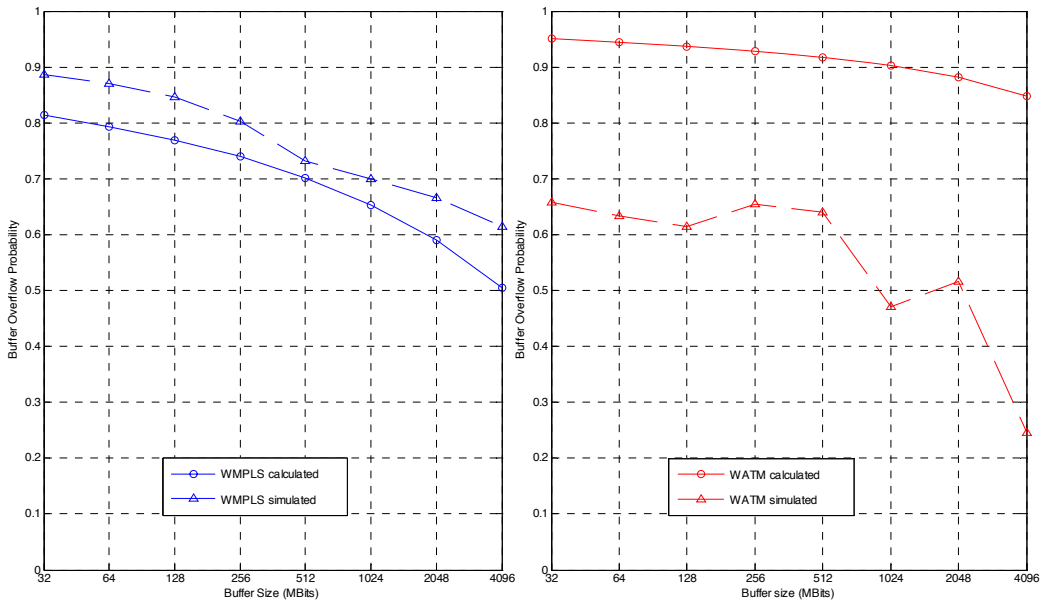


Figure 6.7 – Buffer Overflow Probability comparison between WMPLS and WATM

The efficiency of WMPLS when managing buffers is validated by the overall lower probability of delay violation when compared to WATM. The results obtained from the simulation are higher than the calculated probability based on equations (3.3) through (3.5) because the actual size of the provisioned buffers is smaller than the one calculated through equation (3.1). The same can be said about the significant difference for the WATM case, in which the probability values resulting from the simulation are much lower than the calculations provided by equation (3.1). As we noted earlier, the simulation actually assigns a bigger buffer to the real-time traffic while the values from the equation assign the same buffer space for all the traffic types. Thus, the simulation provides a lower overall buffer overflow probability profile.

7. SUMMARY, FUTURE RESEARCH AND CONCLUSIONS

During the development of this research work several key concepts and factors of wired and wireless networks and their interoperability were studied and analyzed. The following paragraphs summarize the findings that resulted from this research.

7.1 Summary

1. Current wireless technologies are growing extremely fast. Wireless networks are ubiquitous and a myriad of services are being currently deployed that provide access to all types of contents for end-users.
2. Wireless technologies have been developed in isolation and with very specific needs, such as the mobile phone networks and Wireless LANs. Several technologies have been developed with the purpose of bridging the gap between these networks and the wired technologies with unsuccessful results.
3. WATM is a technology that bridges the wireless and wired domains in a homogeneous manner, providing transparent services for upper layer protocols and applications by means of IntServ, however the limitations of scalability and interoperability with other technologies have prevented its deployment or further development.
4. MPLS technology succeeds ATM, and provides support for DiffServ which is a better option compared to IntServ for providing networking services tailored for different types of applications, such as real-time voice transmissions, video and other bandwidth intensive and delay sensitive applications. Additionally MPLS is designed as a cross-layer technology that bridges the gap between technologies transparently. It is capable of

providing services in an overlay fashion such that older transport technologies can be used for providing services that would not be available otherwise.

5. WMPLS, as a new protocol, is a natural extension of the services provided by MPLS that is focused on bridging the gap that exists between the wireless and wired networks. WMPLS overcomes the limitations that WATM presented because of the inherent nature of MPLS to interoperate with other technologies, and most importantly because of the scalability characteristics it possesses.
6. WMPLS relies on the RSVP-TE signaling protocol that provides extensions to support DiffServ parameters. The design of WMPLS includes several changes to the protocol format in order to support the bridging between the wired and the wireless domains.
7. The new extensions to the signaling protocol and the mechanisms that provide homogeneity between the wired and wireless protocol are presented in the form of an architectural design. The architecture of the protocol is based on an UML representation in the form of Finite State Machine diagrams and Message Sequence Charts that illustrate the protocol's message flow.
8. A performance evaluation analysis is then presented in order to show the feasibility of the protocol. An in depth message complexity analysis is presented for the algorithmic of the protocol. Additionally, a performance analysis of WMPLS compared to WATM is presented in order to show the improvements that are comparable under Pareto conditions.

7.2 Future Research

The research presented in this dissertation provides only a starting point from which further work can be extended. Some of the topics that should be covered as a natural continuation of the design process include:

1. The definitions of additional extensions to the RSVP-TE protocol in order

to provide specialized services such as *multicasting services*, which are a very important mechanism of massively distributing information to many receivers [32].

2. The definition of extensions and services to integrate the network with telephony services (such as GSM, EDGE, UMTS, and VoIP) in order to reduce the current complexity of convergence that current networks suffer. The telephony services should be able to operate transparently in order to take advantage of currently deployed technologies, and should also interoperate with emerging wireless technologies such as WiMAX [33].
3. Further investigation will be conducted on the interaction of wireless routing protocols such as AODV and DSR in order to provide better support for mobility in infrastructure mode and ad-hoc networking. The evolution of third and fourth generation wireless communication networks into very high-speed wireless mobile communications needs also to be more closely evaluated in order to provide a robust enough protocol in order to cope with this need.
4. The implementation of a prototype on a hardware platform based on an software platform such as VxWorks, which supports wireless and wired technologies. This prototype implementation will allow the validation and evaluation of the proposed solution, and will allow to make the necessary changes and optimizations in order to have a robust design capable of being developed commercially.
5. An in-depth analysis of the performance evaluation parameters of the network, based on a complete implementation, in order to study the capabilities of the classification, queuing, and scheduling disciplines inherent to MPLS [34], in order to enhance the algorithms for a hybrid network.

7.3 Conclusions

The design of WMPLS, as an extension to MPLS, responds to the necessity to overcome the problems that previous wireless communications protocols have

encountered when trying to provide reliable high-speed data communications. The extensions provided by WMPLS allow the provision of real-time delivery of multimedia content, guaranteed QoS, DiffServ and TE parameters negotiation, and connection-oriented and connectionless communication links in a homogeneous fashion between the wired and wireless domains. These features make WMPLS a suitable protocol to be used as an interoperability layer among current technologies, and also to become a homogeneous platform for future mobile wireless communications.

WMPLS also provides the ability and mechanisms that allows mobile ad hoc networking to be much more flexible, controllable, and reliable. By providing data aggregation and node relaying capabilities, a solid base is presented for military and emergency applications using MANETs.

WMPLS technology is being developed to provide solutions to most of the limitations that WATM-ATM networks have, primarily the limited scalability and the interoperability between ATM implementations and higher level protocols. WMPLS also aims at enabling special features like connectionless ad-hoc and mobile ad hoc networks to be established without the need of complex configurations.

In this research document, the procedures to achieve soft handover in WMPLS have been presented. An overlay model of WMPLS operating over IMT-2000 has been illustrated, and the capabilities of WMPLS to provide support for MANETs in support of QoS and/or TE service features have also been discussed.

WMPLS has been developed as a fully compatible protocol with MPLS for enhanced high-speed translation from the wired network to the wireless portable transceivers. WMPLS is a homogeneous protocol with MPLS and GMPLS by protocol architectural design. It also uses the same control signaling protocols with some extensions, which enables full interoperating features. The basic protocol format of WMPLS closely resembles the original MPLS protocol, but

includes some wireless application specific extensions. Including the mobile wireless port as a part of the overall network connection allows the wireless portable device to be a part of the MPLS path. This enables equivalent features of the MPLS network to exist over the wireless link as well. In addition, general problems such as interoperability with future MPLS networks, and supporting and negotiating differentiated services and traffic engineering features will be solved due to the inherent multiprotocol architecture and additional features that the MPLS networking and signaling protocols provide.

The research shown that fully integrated QoS and TE parameters support and guarantees are provided for a hybrid wired and wireless network without manual intervention for setup or management, including one-hop and multi-hop wireless networks. The mechanisms presented in Sections 5.7 and 5.8 also provide complete support for setting up wireless networks, in either infrastructure or ad hoc mode, involving one-hop or multi-hop routing capabilities, with complete support for connection-oriented and connectionless modes. The mechanisms presented also include the parameters necessary to perform flow and error control tasks independently of higher or lower level protocols.

The performance evaluation analysis presented in Chapter 6 shows the validity of the protocol design and implementation. The graphical results obtained and as shown in Figures Figure 6.5 through Figure 6.7 provide the confirmation that the new protocol performs better than its predecessor in similar tasks. The message complexity analysis provides the basis for future analysis. Current protocols do not provide the mechanisms described by WMPLS and thus are not comparable in this context.

Finally, the platform provided for specific types of transport networks enable transparent interaction between third and fourth generation wireless communication systems and current high-speed optical networking, allowing them to run independently in an overlay fashion.

8. REFERENCES

[1] J.-M. Chung, M. A. Subieta Benito, K. Srinivasan, S.-C. Kim, and Z. Quan, "Performance analysis of WMPLS signaling and control in ad hoc networks," *Proc. of the Midwest Symposium on Circuits and Systems, Aug. 2002*, pp. 627 – 630, Aug. 2002.

[2] J.-M. Chung, M. A. Subieta Benito, K. Srinivasan, "Wireless Multiprotocol Label Switching (WMPLS)," *Internet Draft*, Internet Engineering Task Force (IETF), The Internet Society. [Online] Available: http://www.potaroo.net/ietf/all-ids/draft-chung-mpls-wmpls-00.txt

[3] J.-M. Chung, M. A. Subieta Benito, and K. Srinivasan, (Invited Paper) "Analysis of Wireless Multiprotocol Label Switching (WMPLS) Applications and Performance Features," *MPLS World*, May/June 2002.

[4] M. A. Subieta Benito, J.-M. Chung, "Design and Performance Evaluation of WMPLS," Submitted *IEEE Trans. on Wireless Communications*.

[5] L. Andersson, P. Doolan, N. Feldman, A. Fredette, and B. Thomas, "LDP Specification," RFC 3036, The Internet Society, Jan. 2001.

[6] R. Brown, "Transport to the future: New technologies carry Layer 2 frames across the IP/MPLS backbone," *Packet – Cisco Systems User Magazine*, 1st Quarter, 2002, pp. 21-23.

[7] J. M. Chung, (Invited Paper) "Wireless Multiprotocol Label Switching (WMPLS)," *Proc. of the 35th Asilomar Conference on Circuits, Systems and Computers 2001*, Pacific Grove, CA, USA, Nov. 2001.

[8] A. Farrel, *The Internet and its Protocols*. San Francisco, CA: Morgan Kaufmann, 2004.

[9] D. Awduche, J. Malcolm, J. Agogbua, M. O'Dell, and J. McManus, "Requirements for Traffic Engineering over MPLS," *RFC 2702*, Sept. 1999.

[10] S. Corson, J. Macker, "Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations," *RFC 2501*, The Internet Society, Jan. 1999.

[11] D. Awduche, L. Berger, D. Gan, T. Li, V. Srinivasan, and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels," *RFC 3209*, The Internet Society, Dec. 2001.

[12] L. Anderson and G. Swallow, "The Multiprotocol Label Switching (MPLS) Working Group decision on MPLS signaling protocols," *RFC 3468*, The Internet Society, Feb. 2003.

[13] R. Braden, L. Zhang, S. Berson, S. Herzog and S. Jamin, "Resource ReserVation Protocol (RSVP) - Version 1 Functional Specification," *RFC 2205*, The Internet Society, Sept. 1997.

[14] A. Mankin, F. Baker, B. Branden, S. Bradner, M. O'Dell, A. Romanow, A. Weinrib and L. Zhang, "Resource ReserVation Protocol (RSVP) - Version 1

Applicability Statement - Some Guidelines on Deployment," *RFC 2208*, The Internet Society, Sept. 1997.
[15] R. Braden and L. Zhang, "Resource ReserVation Protocol (RSVP) - Version 1 Message Processing Rules," *RFC 2209*, The Internet Society, Sep. 1997.
[16] J. Wroclawski, "The Use of RSVP with IETF Integrated Services," *RFC 2210*, The Internet Society, Sept. 1997.
[17] A. Terzis, B. Braden, S. Vincent and L. Zhang, "RSVP Diagnostic Messages," *RFC 2745*, The Internet Society, Jan. 2000.
[18] S. Herzog, "RSVP Extensions for Policy Control," *RFC 2750*, The Internet Society, Jan. 2000.
[19] L. Berger, D. Gan, G. Swallow, P. Pan, F. Tommasi, and S. Mondelini, "RSVP Refresh Overhead Reduction Extensions," *RFC 2961*, The Internet Society, Apr. 2001.
[20] S. M. Ross, *A Course in Simulation*. New York, NY: Macmillan, 1990.
[21] J.-S. Lee and P.-L. Hsu, "UML-Based Modeling and Multi-Threaded Simulation for Hybrid Dynamic Systems," *Proc. IEEE Intl. Conference on Control Applications*, Sept. 2002, pp. 1207 – 1212.
[22] D. Harel, "Statecharts: A visual formalism for complex systems," *Sci. Comput. Program.*, vol. 8, pp. 231 – 274, 1987.
[23] S. T. Levi and A. K. Agrawala, *Real-Time System Design*. New York, NY: McGraw-Hill, 1990.
[24] R. Prasad and T. Ojanpera, "An Overview of CDMA Evolution toward Wideband CDMA," *IEEE Commun. Surveys and Tutorials*, vol. 1, no. 1, 4th Quarter, 1998.
[25] E. Rosen and A. Viswanathan, "Multiprotocol Label Switching Architecture," *RFC 3031*, The Internet Society, Jan. 2001.
[26] C.-K. Tok, *Wireless ATM and Ad-Hoc Networks: Protocols and Architectures*. Kluwer, 1997.
[27] T. Clausen, P. Jacket, "Optimized Link State Routing Protocol (OLSR)," *RFC 3626*, The Internet Society, Oct. 2003.
[28] S. Corson, J. Macker, "Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations," *RFC 2501*, The Internet Society, Jan. 1999.
[29] C. E. Perkins, E. M. Belding-Royer, S. Das, "Ad hoc On-Demand Distance Vector (AODV) Routing," *RFC 3561*, The Internet Society, Jul. 2003.
[30] C. Bouras, V. Kapoulas, D. Primpas, "Performance evaluation of the managed bandwidth service with QoS guarantees," *Proc. of Intl. Symposium of Distributed Simulation and Real-Time Applications*, Oct. 2003, pp. 93 – 100.
[31] W. Ryu J.H. You, D.-W. Kim, T.-J. Kim, B.-N Yoon, "Design and performance evaluation of a communication protocol for an information communication processing system," *Conf. Proc. ICCS '94*, vol. 3, Nov. 1994, pp. 1200 – 1203.
[32] C. K. Miller, *Multicast Networking and Applications*, Reading, MA: Addison Wesley Longman, 1999.
[33] C. Smith, J. Meyer, *3G Wireless with WiMAX and Wi-Fi*. New York, NY: McGraw Hill, 2005.

[34] H.-M. Soo, J.-M. Chung, "Analysis of nonpreemptive priority queueing of MPLS networks with bulk arrivals," *Proc. of the Midwest Symposium on Circuits and Systems*, Aug. 2002, pp. 679 – 683.
[35] J. Moy, "OSPF Version 2," *RFC 1583*, The Internet Society, Mar. 1994.
[36] J. M. Pitts, J. A. Schormans, *Introduction to IP and ATM Design and Performance*, West Sussex, England: John Wiley & Sons, 2000.
[37] A. Osbourne, A. Simha, *Traffic Engineering with MPLS*, Indianapolis, IN: Cisco Press, 2004
[38] S. Vegesna, *IP Quality of Service*, Indianapolis, IN: Cisco Press, 2001
[39] J. J. Garcia-Luna-Aceves, J. Behrens, "Distributed, Scalable Routing Based on Vectors of Link States," *IEEE Journal on Selected Areas in Communications*, vol. 13, No. 8, Oct. 1995, pp. 1383 –1395.
[40] X. Hong, K. Xu, M. Gerla, "Scalable Routing Protocols for Mobile Ad Hoc Networks," *IEEE Network*, Jul./Aug. 2002, pp. 11 – 21.
[41] S. H. Kwok, A. G. Constantinides, "A Fast Recursive Shortest Spanning Tree for Image Segmentation and Edge Detection," *IEEE Trans. on Image Processing*, Vol. 6, No. 2, Feb. 1997, pp. 328 – 332.
[42] I. F. Akyildiz, X. Wang, W. Wang, "Wireless Mess Networks: A Survey," *Elsevier Computer Networks*, Vol. 47, No. 4, Mar. 2005, pp. 445 – 487.
[43] U. Black, *MPLS and Label Switching Networks*, Upper Saddle River, NJ: Prentice Hall PTR, 2002.
[44] T. A. Chu. (2002, March 1). Traffic Management Part 2: Pipelining Specifics [Online]. Available: http://www.commsdesign.com/showArticle. jhtml? articleID=16504217
[45] ITU Recommendation I.371, *Traffic Control and Congestion Control in B-ISDN*, Aug. 1996.
[46] ATM Forum AF-TM-0121.000, *Traffic Management Specification*, Version 4.1, Mar. 1999.
[47] D.-Y. Eun, "Analysis, QoS estimation, and decomposition of large networks," Ph.D. dissertation, Dept. Elect. Eng., Purdue Univ., West Lafayette, IN, 2003.
[48] G. Cao and M. Singhai, "A delay-optimal quorum-based mutual execution algorithm for distributed systems," *IEEE Trans. Parallel and Distributed Systems*, Vol. 12, No.12, pp. 1256-1268, Dec. 2001.
[49] C-. C. Shen, and C. Srisathapornphat, and R. L. Z. Huang, and C. Jaikaeo, and E. L. Lloyd, "CLTC: A cluseter-based topology control framework for ad hoc networks," *IEEE Trans. Mobile Computing*, Vol. 3, Vo.1, pp. 18-32, Jan.-Mar. 2004.
[50] S. H. Kwok and A. G. Constantinides, "A fast recursive shortest spanning tree for image segmentation and edge detection," *IEEE Trans. Image Processing*, Vol. 6, No.2, pp. 328-332, Feb. 1997.
[51] A. Boukerche, S. Hong, and T. Jacob, "An efficient synchronization scheme of multimedia streams in wireless and mobile systems," *IEEE Trans. Parallel and Distributed Systems*, Vol. 13, No.9, pp. 911-923, Sep. 2002.
[52] S.-C. Kim, "Message Complexity Analysis of Mobile Ad Hoc Network (MANET) Address Autoconfiguration Protocols," Ph.D. dissertation, Dept. Elect. and Comp. Eng., Oklahoma State Univ., Stillwater, OK, 2005.

[53] J. Gross and J. Yellen, *Graph Theory and Its Applications*, CRC Press, 1998.
[54] T. H. Coirmen, C. E. Leiserson, R. L. Rivest and C. Stein, *Introduction to Algorithms*, Cambridge, MA: MIT Press, 2001.
[55] SDL Forum Society (2005, December 4). What is an MSC? [Online]. Available: http://www.sdl-forum.org/MSC/index.htm
[56] J. Holt, *UML for Systems Engineering*, London, United Kingdom: The Institution of Electrical Engineers, 2001.
[57] I. D. Aron, and S. K. S. Gupta, "Analytical Comparison of Local and End-to-end Error Recovery in Reactive routing Protocols for Mobile Ad Hoc Networks", *Proc. of 3rd ACM Int. on Workshop on Modelling, Analysis and Simulation of Wireless Systems (MSWIN2000)*, pp. 69 – 76 Aug. 2000.
[58] D. Sun, and H. Man, "Performance Comparison of Transport Control Protocols Over Mobile Ad Hoc Networks," *Proc. of the 12th Int. Symposium on Personal, Indoor and Mobile Radio Comm., 2001* , vol. 2, pp. G-83 – G-87, Oct. 2001.
[59] M. Lott, B. Walke, "Performance of a Wireless Ad hoc Network Supporting ATM," *Proc. of the 2nd ACM international Workshop on Wireless Mobile Multimedia*, pp. 18 – 25, 1999.
[60] A. Boukerche, "Performance Evaluation of Routing Protocols for Ad Hoc Wireless Networks," *Mobile Networks and Applications*, Kluwer Academics, Vol. 9, No. 4, pp. 333 – 342, 2004.
[61] Cisco Systems, "Buffer Tunning for all Cisco Routers,"Document ID: 15091, Feb. 2, 2006.

VITA

Mauricio Antonio Subieta Benito

Candidate for the Degree of

Doctor of Philosophy

Dissertation: DESIGN AND PERFORMANCE EVALUATION OF WIRELESS MULTI-PROTOCOL LABEL SWITCHING (MPLS)

Major Field: Electrical Engineering

Biographical:

Personal Data: Born in Potosí, Bolivia, on July 11, 1974, the son of Graciela María Benito Pozzo and Gunnar Enrique Subieta Castro. Married to Naneida Lazarte Alcalá and father to Mateo Sebastian Subieta Lazarte.

Education: Graduated from Hudson High School, Hudson, Ohio in December 1991; received Bachelor of Science in Systems Engineering from the Universidad Católica Boliviana, La Paz, Bolivia in December 1996; received Master of Science degree in Telecommunications Management from Oklahoma State University, Stillwater, Oklahoma in May 2002. Completed the requirements for the Doctorate of Philosophy degree with a major in Electrical Engineering at Oklahoma State University, Stillwater, Oklahoma in July 2006.

Experience: Network Engineer at Oklahoma State University from July 2004 to present; Research Assistant at Oklahoma State University from July 2003 to July 2004; Senior Consultant at PricewaterhouseCoopers from July 2002 to August 2003; Adjunct Professor at the Universidad Católica Boliviana from July 2002 to August 2003; Research Assistant at Oklahoma State University from November 2000 to May 2002; IT Manager at SOBOCE S.A. from February 1997 to July 2000; Adjunct Professor at the Universidad Católica Boliviana from August 1997 to July 2000.

Professional Memberships: Institute of Electrical and Electronics Engineers, Association for Computing Machinery.

Name: Mauricio Antonio Subieta Benito Date of Degree: July, 2006

Institution: Oklahoma State University Location: Stillwater, Oklahoma

Title of Study: DESIGN AND PERFORMANCE EVALUATION OF WIRELESS MULTI-PROTOCOL LABEL SWITCHING (MPLS)

Pages in Study: 130 Candidate for the Degree of Doctor of Philosophy

Major Field: Electrical Engineering

Scope and Method of Study: The research presented in this document focuses on the design of a new protocol for high-speed wireless data communications. The primary goal of this new design is to overcome the limitations of its predecessors, while minimizing the needed resources and maximizing throughput and efficiency in its operations. Another important goal of the study is to provide a homogeneous protocol for wired and wireless networks in order to provide complete interoperability for overlay models and other protocols that can be designed on the basis of this work. The performance evaluation part of this document shows the areas in which improvement has been achieved over previous protocol implementations, and it also shows the areas in which further research is needed in order to improve the performance at least to the levels set by previous protocols.

Findings and Conclusions: This study shows that a native wireless design and implementation of the Multi-Protocol Label Switching (MPLS) protocol provides improvements in the field of wireless data communications, providing a homogeneous platform for voice and data communication networks. The research is open for further improvements and modifications for services not contemplated in this document, and continuous developments should be conducted in order to obtain a working prototype of this proposal.

ADVISER'S APPROVAL: Dr. Keith Teague