Reviewing the Service Specification of the IEEE 802.16 MAC Layer Connection Management: A Formal Approach

Ana Morales
Facultad de Ciencias, Universidad Central de Venezuela
Caracas, Venezuela, 1040
ana.morales@ciens.ucv.ve

and

María Villapol
Facultad de Ciencias, Universidad Central de Venezuela
Caracas, Venezuela, 1040
maria.villapol@ciens.ucv.ve

ABSTRACT

In most of the communication protocol specification documents, there is little, if any, use of more formal techniques for specifying the protocols, such as state and service primitive tables. Thus, these documents are sometimes ambiguous, difficult to understand, and imprecise. The IEEE 802.16 standard document is responsible for specifying and describing the air interface of the BWA systems (Broadband Wireless Access Systems) point to multipoint fixed and mobile networks, and is limited to the description of the MAC(Medium Access Control) layer and physical (PHY). Since the MAC layer is connection-oriented, the standard defines how the connection management service is provided. The service is specified as the occurrence of a set of well-defined service primitives. However, the description of the service specification is somehow informal and presents some ambiguities and inconsistences. So in this paper, we describe the omissions, uncertainties and discrepancies found in the standard documents and propose some solutions to fix these problems. We also provide a formal description of the connection management service specification using Finite State Automata (FSA).

Keywords*:* IEEE 802.16, MAC Layer, Service Primitives, Service Specification, Finite State Automata (FSA)

1 Introduction

WiMax is a wireless technology for metropolitan area networks that was developed by the IEEE Working Group 802.16 [1]. This technology provides an alternative to cable access networks such as fiber optic links and digital subscriber lines (DSLs). Compared to its competitor cable technologies, WiMax is easier to deploy and is poised for more ubiquitous broadband access in the future. WiMax users (subscriber stations, SS) access the network through exterior networks communicating with central radio base stations (BS). The IEEE 802.16 protocol stack includes two layers: the physical and MAC layers. The MAC layer is connection-oriented, so all the services including connectionless services are mapped to a connection. Thus, the standard defines the connection management procedures and how the service is provided to the upper layer protocol [1] [2].

The *service specification* describes the service that is provided to the user. This is often given as a sequence of events that are possible at an abstract interface between the user (an application or another protocol) and the protocol. The communication between adjacent layers can be described at an abstract level through the occurrence of the *service primitives*, which provide an abstract way to describe the interaction between the service user and the service provider. The service specification is defined at a higher level of abstraction than the protocol specification and can be described in the protocol specification document or in a separate one [3]. The IEEE 802.16 MAC layer connection management service specification is described in the standard document [1]. The specification explains the service by using narrative and some event sequence diagrams. However, does not impose message formats or state machines for the use of these primitives [1]. Thus, this lack of formalism in describing the service specification may lead to ambiguity and inconsistency which may cause possible errors in the implementation of the MAC layer

protocol. Our first attempt at modeling and analysis of the MAC CPS connection management service specification is presented in [14], where a formal and detailed description of the service specification given. The analysis results of the model of the service specification show some inconsistencies and information gaps in the standard document. Thus, this paper describes the omissions, ambiguities and inconsistencies observed in the standard, and propose some solutions to fix these problems. We also provide a formal description of the connection management service specification using *Finite State Automata* (*FSA*).

Coloured Petri Nets (CPNs) are a formal technique used for modelling many systems, particularly communication protocols [4][5][6][7][8]. Billington et al. [5] present a methodology for specifying and validating communication protocols and show some example of how the methodology can be applied using CPNs. In the methodology, the specification process is divided into: the service specification and protocol specification. We use the formal approach proposed by Billington [5] to describe the IEEE 802.16 MAC layer connection management service specification.

The paper is organised as follows. Section 2 presents a description of the IEEE 802.16 standard. Section 3 explains briefly the protocol verification methodology proposed by Billington. Section 4 describes the IEEE 802.16 MAC layer connection management service specification. Section 5 presents some relevant aspects of the service specification based on the application of the methodology. And finally in Section 6 concludes the papers and show some future directions of the work.

2 An Overview of IEEE 802.16

The IEEE 802.16 standard is responsible for specifying and describing the air interface for *Broadband Wireless Access* (*BWA*) systems for fixed and mobile users and is limited to the description of the MAC and PHY layers. The initial version of the IEEE 802.16 standard was completed in October 2001 [1] and its main goal was to provide broadband wireless access. It was also developed to provide support for mesh topologies and provide handoff (handover) of the signal between base stations [9][10].

The standard was designed to accommodate a set of air interfaces based on a common MAC protocol but with different physical layer specifications depending on the use of radio spectrum and the regulations associated with them [1] [2].

2.1 Components of the IEEE 802.16 Architecture

The IEEE 802.16 standard specifies the basic entities in the architecture (see Figure 1). They are described as follows:

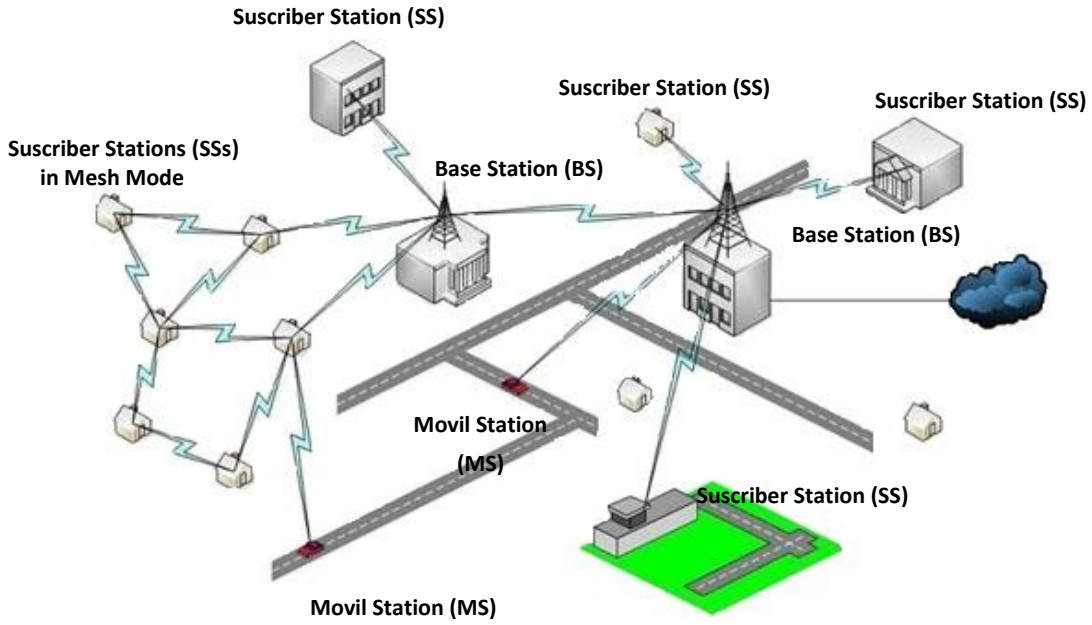

Figure 1: Basic Components of the IEEE 802.16 Architecture.

Base Station (BS): is a device that provides general control service, management functions and connectivity to the subscriber station and includes some entity instances of the MAC and PHY layers.

Subscriber Station (SS): is a generalized device which provides connectivity between a suscriber equipment and a base station and includes some entity instances of the MAC and PHY layers.

Mobile Station (MS): is an optional entity defined in the later versions of the IEEE 802.16 standard document [9] [10] and includes some additional capabilities to support vehicular mobility. It also provides additional support for specific functions such as SS handoff management and energy saving. Thus the MS delivers the functionalities for supporting both wireless broadband fixed and mobile users.

2.2 Reference Model

The IEEE 802.16 Reference Model includes two major components: the data/control plane and the management plane [1][9][10]. The data/control plane includes two layers: the physical and MAC layers. The MAC layer is further divided in three sub-layers: *Service Specific Converge Sublayer* (CS), *MAC Common Part Sublayer* (MAC CPS), and *Security Sublayer* (see Figure 2). The control and data planes define how the information is encapsulated or desencapsulated at the MAC level, and modulated or demodulated at the physical level. The functions of the management plane are: classification, security, application QoS and connection settings among others. The layers are described as follows:

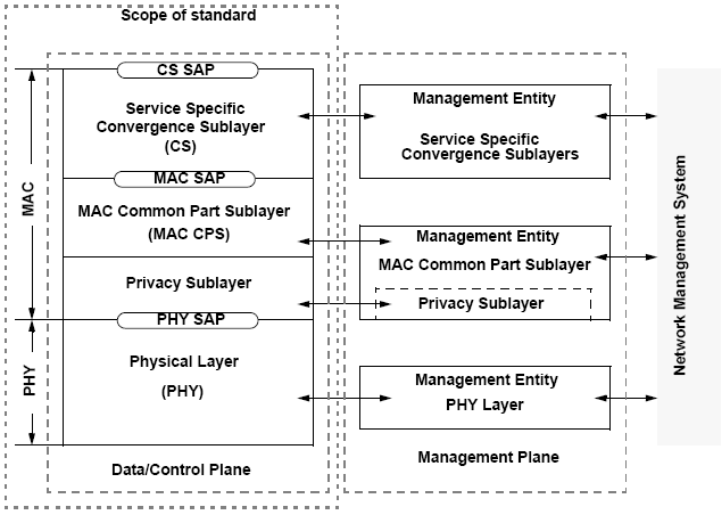

Figure 2: IEEE 802.16 Reference Model.

- ***Service Specific Convergence Sublayer (CS*)**: the *Service Data Units* (*SDUs*) from the external networks are received through the MAC CS SAPs. Thus, the functions of the CS include the classification of external SDUs and their associations with the appropriate *Service Flow* (*SFID*) and *Connection ID* (*CID*).

- ***Common Part Sublayer (MAC CPS*)**: is responsible for controlling the access to the medium. The other basic functions of the MAC CPS layer are data encapsulation/de-encapsulation, and data packing and fragmentation. It is also responsible for data control error (error detection and retransmission strategy). Additionally, this provides mechanisms for traffic control and QoS provision. The MAC layer is connection-oriented, so all the services including connectionless services are mapped to a connection.

- ***Security Sublayer*:** is responsible for delivering privacy to subscribers in the wireless network. It provides authentication and secure key exchange and encryption on the connections established between the either a SS or a MS and the BS.

- **Physical Layer (PHY):** defined to work on the 10-66 GHz band. It supports adaptive burst profiling in which some transmission parameters, such as modulation and coding schemes, may be changed on either a per-connection or per-subscriber basis to adapt to channel changing conditions and to provide varying levels of service. *Time Division Duplexing* (TDD) and *Frequency Division Duplexing* (FDD) techniques are supported. The data encapsulated in MPDUs are carried in TDD or FDD PHY frames

2.3 Connection Management

The MAC CPS is connection-oriented, so a connection must be established between peer convergence processes. Once a connection is established, it may require to be maintened. Maintenance activities are needed, for example, to deal with dynamic bandwidth requirements. Finally, a connection may be terminated by either the BS or SS.

Management activities of connections are supported through a set of service primitives between the entities of the MAC CS and CPS implemented in both the SSs and the BS. Some aspects of the connections are described as follows [1] [11]:

- **Connection establishment:** after a SS register with the BS, the transport connections are established. Each connection is associated with a service flow and defines the relationship between the CS peer entities that use this MAC service flow. Additionally, new connections can be established in order to change the established characteristics of a client service.

- **Changes of the connection characteristics:** once a connection is established it may need to be actively maintained. Maintenance requirements vary according to the type of service. The modification of the connections may require maintenance stimulated by the SSs or the network connection (BS).

- **Connection termination:** connections can be terminated. The termination of a connection can be initiated by either the BS or the SS.

The mentioned management activities are supported through the use of static configurations and, dynamic additions, modifications and terminations of the connection.

2.4 Definition of the MAC Services

More communication technologies use layers structured hierarchically to divide the communication functions. Each layer performs a set of functions intended to provide some services to the higher layer. A service is a capability of a layer (service provider) that is provided to the layer above it (i.e. service user) through the *service access point* (SAP) (see Figure 3). A SAP is an address, which identifies the boundary between two adjacent layers. The communication between adjacent layers can be described at an abstract level through the occurrence of the service primitives (see Figure 3). The service primitives provide an abstract way to describe the interaction between the service user and the service provider. Each primitive can be a request, indication, response or confirmation. A *request* is issued by the service user for requesting some service from the service provider. An *indication* is used by the service provider to notify the service user that the other peer has invoked a request primitive or the provider itself has generated an event. A *response* is used by the service user to acknowledge receipt of the indication primitive from the service provider. Finally a *confirmation* is used by the service provider to notify the requesting service user that the activity initiated by the request has been successfully completed [12].

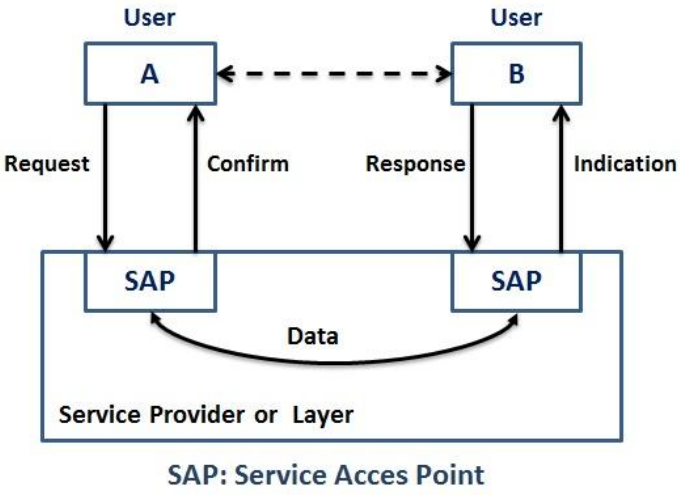

Figure 3: Communication between peer entities and between CS and MAC layers CPS.

The IEEE 802.16 standard [1] defines the service specification for the connection establishment, change and termination. It includes a set of service primitives which provide an abstract way to describe the interaction between the CS (service user) and the MAC CPS (service provider) (see Figure 4). These primitives are shown in Table 1 and described as follows:

MAC_CREATE_CONNECTION (Request/Indication/Response/Confirmation): a CS entity uses this primitive to establish a connection.

MAC_CHANGE_CONNECTION (Request/Indication/Response/Confirmation): a CS entity uses this primitive to change the characteristics of an established connection.

MAC_TERMINATION_CONNECTION (Request/Indication/Response/Confirmation): a CS entity uses this primitive to terminate a connection.

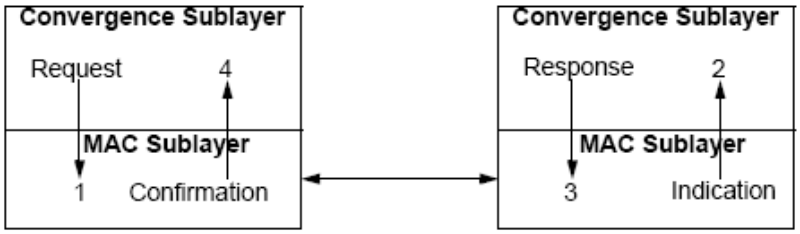

Figure 4: Communication between peer entities and between CS and MAC layers CPS.

Table 1: Service primitives for connection management

Connection establishment	MAC_CREATE_CONNECTION.Request
	MAC_CREATE_CONNECTION.Indication
	MAC_CREATE_CONNECTION.Response
	MAC_CREATE_CONNECTION.Confirmation
Change of the connection characteristics	MAC_CHANGE_CONNECTION.Request
	MAC_CHANGE_CONNECTION.Indication
	MAC_CHANGE_CONNECTION.Response
	MAC_CHANGE_CONNECTION.Confirmation
Connection termination	MAC_TERMINATE_CONNECTION.Request
	MAC_TERMINATE_CONNECTION.Indication
	MAC_TERMINATE_CONNECTION.Response
	MAC_TERMINATE_CONNECTION.Confirmation

The messages DSA-REQ, DSC-REQ and DSD-REQ are generated as a consequence of the occurrence of the MAC_CREATE, MAC_CHANGE, and MAC_TERMINATION service primitive requests respectively. The messages DSA-RSP, DSC-RSP and DSD-RSP are generated as a consequence of the occurrence of the MAC_CREATE, MAC_CHANGE, and MAC_TERMINATION service primitive responses respectively. The messages DSA-ACK or DSC-ACK are generated by the requesting MAC CS entity as an acknowledgement of a dynamic service addition or dynamic service change, respectively [1]. They are not generated as the occurrence of any service primitive.

2.5 Sequence of Service Primitives

A service primitive sequence may include some or all of the above types of service primitives. The MAC CPS connection establishment, change and termination service primitive sequences allowed for each service user are defined and shown in Tables 2 and 3. A primitive listed in a column header may only be followed by the primitives listed in the row headers that are marked with an “X”. The shaded rows indicate the sequences that can occur at the requesting side while the others represent the sequences that can occur at the requested side. These service primitive sequences were derived from IEEE 802.16 standard document [1].

A service is confirmed (confirmed service) when the service provider, which may be either the network or the requested service user, gives an explicit confirmation to the requesting service user. Thus a confirmed service primitive sequence includes all of the service primitives defined above. If a service is not confirmed (non-confirmed service), an explicit confirmation from the service provider is not required. Thus a non-confirmed service primitive sequence only includes a request and an indication. As we can see in Tables 2 and 3, all the MAC CPS connection management services are confirmed.

Table 2: Service primitives sequences Part1

	MAC_CREAT_CONNECTION				MAC_CHANGE_CONNECTION			
	Req	Ind	Rsp	Conf	Req	Ind	Rsp	Conf
MAC_CREAT_CONNECTION.Request	X			X				
MAC_CREAT_CONNECTION.Indication			X					
MAC_CREAT_CONNECTION.Response		X	X					
MAC_CREAT_CONNECTION.Confirmation	X			X				
MAC_CHANGE_CONNECTION.Request	X			X	X			X
MAC_CHANGE_CONNECTION.Indication		X	X			X	X	
MAC_CHANGE_CONNECTION.Response		X	X			X	X	
MAC_CHANGE_CONNECTION.Confirmation	X			X	X			X
MAC_TERMINATE_CONNECTION.Request	X			X	X			X
MAC_ TERMINATE_CONNECTION.Indication		X	X			X	X	
MAC_ TERMINATE_CONNECTION.Response		X	X			X	X	
MAC_ TERMINATE_CONNECTION.Confirmation	X			X	X			X

Table 3: Service primitives sequences Part2

	MAC_TERMINATE_CONNECTION			
	Req	Ind	Rsp	Conf
MAC_CREAT_CONNECTION.Request	X			X
MAC_CREAT_CONNECTION.Indication		X	X	
MAC_CREAT_CONNECTION.Response		X	X	
MAC_CREAT_CONNECTION.Confirmation	X			X
MAC_CHANGE_CONNECTION.Request	X			X
MAC_CHANGE_CONNECTION.Indication		X	X	
MAC_CHANGE_CONNECTION.Response		X	X	
MAC_CHANGE_CONNECTION.Confirmation	X			X
MAC_TERMINATE_CONNECTION.Request	X			X
MAC_ TERMINATE_CONNECTION.Indication		X	X	
MAC_ TERMINATE_CONNECTION.Response		X	X	
MAC_ TERMINATE_CONNECTION.Confirmation	X			X

3 Protocol Verification Methodology

Formal protocol specification and verification involves a set of activities, which range from the description of the protocol architecture to the verification of the proposed model [3] [5]. Billington et al. [5] present these activities as a set of systematic steps and call them a protocol verification methodology. The aim of this section is to describe the main steps involved in the protocol verification methodology as follows:

- Service specification: describes the service that is provided to the user. This is often given as a sequence of events that are possible at an abstract interface between the user (an application or another protocol) and the protocol. The service specification is defined at a higher level of abstraction than the protocol specification;

- Protocol specification: includes a detailed description of the features of the protocol, which allows it to provide the explicit or implicit services. It consists of a set of rules, formats, and procedures for two or more entities to communicate on different machines (e.g. computer) within a network;

- Underlying service specification: specifies the characteristics of the communication technology by which the different protocol entities communicate and includes the communication service provided by the underlying service in the protocol architecture;

- Composite specification: includes the protocol specification and the underlying service specification. The two specifications form the composite specification of the protocol entities communicating over the underlying service;

- Analysis: requires the proof of the desired properties of the composite specification using reachability analysis and/or theorem proving.

- Comparison: consists of checking if the service specification satisfies the service specification with the composite specification to see if the composite specification is a correct refinement of the service.

Figure 5 shows the steps involved in the protocol verification methodology. Initially, the service specification is modeled using a formal method such as CPNs [4]. Then the composite specification is also modeled. Simulations can be used for initial debugging of the models. While errors are found in the simulation, the model is modified and the simulation activities repeated as shown in Figure 5.

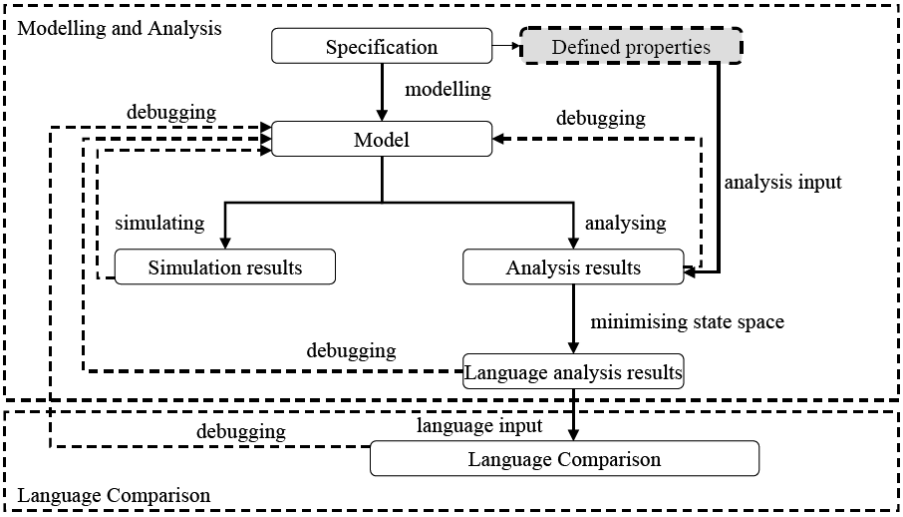

Figure 5: Abstract View of the Protocol Verification Methodology [3]

A *property* refers to a particular characteristic, which should be present in a protocol. For example, Holzmann [13] defines a set of general properties, which can apply to any protocol, such as no deadlocks and no livelocks. Some other particular properties, which apply to the protocol under study, can be also defined. They can be generated from the protocol specification. The desired properties of the protocol must be verified (analysis results) using formal analysis techniques such as state space analysis, system invariants, temporal logic and model checking. The analysis results may show some errors. The errors need to be analyzed in order to determine their causes. They can be a consequence of, for example, a model mistake (e.g. erroneous inscriptions), an inaccuracy of the specification or modeling assumptions. Thus, it may or may not require the model to be modified. If the model does require modification then the simulation and analysis activities must be repeated as shown in Figure 5.

A *service language* consists of all the possible service primitive occurrence sequences, which can occur between the (end-to-end) users of the protocol. The *protocol language* consists of all the possible service primitive occurrence sequences, which can be generated by the protocol entities. The resulting (service or protocol) language can be analyzed. The analysis may consist of checking that all the service primitive sequences are expected. The service and protocol language must be compared to check if they are equivalent. If some of the service primitive sequences are in the protocol language, but not in the service language or vice versa, they need to be analyzed. These sequences can be a result of, for example, a model mistake, an inaccuracy of the specification or modeling/scope assumptions. Thus, it may or may not require the model to be modified.

4 Service Specification of the MAC CPS Connection Management

In this work, we model and analyze the service specification of the MAC CPS connection management using the protocol verification methodology presented above. The service specification is mostly described in the IEEE 802.16 standard document [1]. Since the service primitive sequences are not clearly specified in the document, we have created the Tables 2 and 3. Then we model the service specification using CPNs, with the help of the software tool CPN Tools 2.0 [15]. The model is debugging using the simulation tool integrated into CPN Tools. We validate the model against some behavioral properties of the CPNs such as non-presence of deadlocks [11] [14] using the *state space* (also called *Occurrence Graph* (OG)) method. The model is described and analyzed in [11] [14].

The OG graph includes not only the transitions representing the connection management service primitives but other transitions, such as error transitions. CPN Tools [15] does not provide explicit support for language generation,

which includes only the service primitive sequences. Since the OG graph can be seen as a *Finite State Automaton* (*FSA*), the service language was generated by using a well-known FSA reduction technique (Barret et al. [16]) with the aid of the FSM Tool [17]. The following steps were followed for the service language generation:

1) The service primitive transitions in the CPN model are assigned different numbers (no zero). The others are marked as zero epsilon (or empty) transitions [16]. The identification numbers used for the connection management service primitives are shown in table 4.

Table 4: Abbreviations and ID numbers used for connection management service primitives

Service Primitive	Transitions	ID Number
MAC_CREAT_CONNECTION.Request	MACCrtConnReq	1
MAC_CREAT_CONNECTION.Request	MACCrtConnReq2	1
MAC_CREAT_CONNECTION.Indication	MACCrtConnInd	2
MAC_CREAT_CONNECTION.Response	MACCrtConnRsp	3
MAC_CREAT_CONNECTION.Response	MACCrtConnRsp2	3
MAC_CREAT_CONNECTION.Confirmation	MACCrtConnCf	4
MAC_CREAT_CONNECTION.Confirmation	MACCrtConnCf2	5
MAC_CREAT_CONNECTION.Confirmation	ConnfrRejected	5
MAC_CREAT_CONNECTION.Confirmation	MACCrtConnCfRech	6
MAC_CHANGE_CONNECTION.Request	MACChgConnReq	7
MAC_CHANGE_CONNECTION.Request	MACChgConnReq2	7
MAC_CHANGE_CONNECTION.Indication	MACChgConnInd	8
MAC_CHANGE_CONNECTION.Response	MACChgConnRsp	9
MAC_CHANGE_CONNECTION.Response	MACChgConnRsp2	9
MAC_CHANGE_CONNECTION.Confirmation	MACChgConnCf	10
MAC_CHANGE_CONNECTION.Confirmation	MACChgConnCf2	10
MAC_CHANGE_CONNECTION.Confirmation	ConnfrRejected	10
MAC_CHANGE_CONNECTION.Confirmation	MACChgConnRechCf	10
MAC_TERMINATE_CONNECTION.Request	MACTerConnReq	11
MAC_TERMINATE_CONNECTION.Request	MACTerConnReq2	11
MAC_TERMINATE_CONNECTION.Indication	MACTerConnInd	12
MAC_TERMINATE_CONNECTION.Response	MACTerConnRsp	13

2) The OG is written to a file in a text format that can be read by the *FSM tool* [17] using an algorithm created for doing this and shown [3]. The *Terminal States*, which are the dead markings and probably other markings in the OG, are generated and written to a text file. Each line of the file corresponds to a *Terminal State*. To simplify and make understandable OG analysis and for the correct generation of the *sequence of service primitives* (e.g. the *language of service*), it is necessary to generate the *Terminal States* for the FSA, which determine the completion of a *sequence service primitives*. Such *Terminal States* are determined by the occurrence of sequences service primitives shown in table 5.

3) A FSM file is generated. It is the result of concatenating the files generated in steps 2. The resulting file, which includes the FSA for the OG is converted to the binary format understood by the FSM tool (using the *fsmcompile* function [17]).

4) The FSA is reduced by following the algorithm described in [16]. The algorithm is based on the following steps: removal of empty move cycles (remove empties), removal of empty moves (remove empties), removal of non-determinism (determinization), removal of inaccessible states (minimization), and reduction by identifying and merging equivalent states (minimization). The FSM tool provides all the programs for supporting these steps. A description of the algorithm and how the FSM tool can be used to implement it is given in [11].

5) The language accepted by the resulting minimal FSA is generated using FSM Tools [17].

Figure 6 shows the FSA that represents the occurrence of the service primitives derived from the OG. The labels on the arcs represent the ID Number which identifies the connection management service primitive as specified in table 4. Each service primitive sequence starts at node "0" (initial state), and ends in a *Terminal State* represented by a double circle (node "0" and "4"). The sequences service primitives shown in table 6 and can be found in the FSA in the figure 6 [11] [14].

We validate the FSA by visual inspection which is possible because of the small size of the automata. Validation procedure consisted to check that all the occurrences of the service primitive sequences are correct. Also, the resulting FSA has cycles. This is expected because of the multiple setup, change and termination requests which can be initiated by an entity.

Table 5: Service Primitives sequences that define the "Terminals States"

Sequences that define "States Terminals"
MAC_CREATE_CONNECTION.Request, immediately followed by a primitive service type MAC_CREATE_CONNECTION.Confirmation
MAC_CREATE_CONNECTION.Request, immediately followed by a primitive service type MAC_CREATE_CONNECTION.Indication, immediately followed by a primitive service type MAC_CREATE_CONNECTION.Response, immediately followed by a primitive service type MAC_CREATE_CONNECTION.Confirmation
MAC_CHANGE_CONNECTION.Request, immediately followed by a primitive service type MAC_CHANGE_CONNECTION.Confirmation
MAC_CHANGE_CONNECTION.Request, immediately followed by a primitive service type MAC_CHANGE_CONNECTION.Indication, immediately followed by a primitive service type MAC_CHANGE_CONNECTION.Response, immediately followed by a primitive service type MAC_CHANGE_CONNECTION.Confirmation
MAC_TERMINATE_CONNECTION.Request, immediately followed by a primitive service type MAC_TERMINATE_CONNECTION.Confirmation
MAC_TERMINATE_CONNECTION.Request, immediately followed by a primitive service type MAC_TERMINATE_CONNECTION.Indication, immediately followed by a primitive service type MAC_TERMINATE_CONNECTION.Response, immediately followed by a primitive service type MAC_TERMINATE_CONNECTION.Confirmation

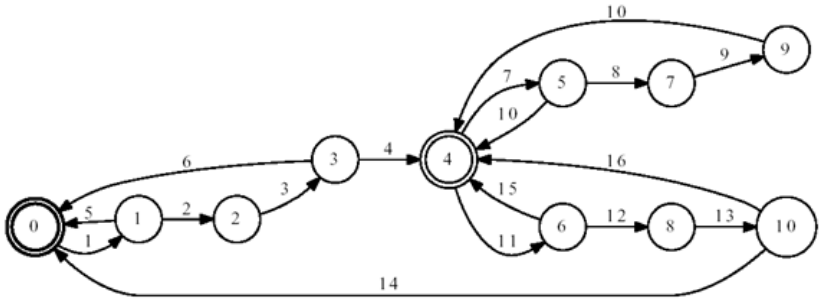

Figure 6: Minimal FSA for the connection management model

5 Highlights of the Service Specification

In this section, we initially describe some problems found during the modeling and analysis of the connection management service specification. Then we explain some proposals to solve the problems. In the service specification step of the protocol verification methodology, we found some information gaps and aspects that are not clearly specified in the document [11] [14]. For example, the standard document does not specify the state of the

interface between the requesting entity and the service provider after the service provider terminates the protocol. Moreover, we could not find the meaning of "terminate the protocol" in the document.

Also, it is important that the standard document specifies the actions taken when a connection management message is received by the protocol entity. Unfortunately, some of these actions are not clearly specified in the IEEE 802.16 standard document. For example, the actions taken by the requested entity when receiving either a DSA-ACK message (i.e. an acknowledgment of a connection creation request) or a DSC-ACK message (i.e. an acknowledgment of a change connection request) sent by the requesting entity are not explained.

A connection setup, change or termination request may be rejected by the service provider. The user service (i.e. the MAC CS layer entity) should know that its request has not been accepted. The standard document does not specify how this situation is reported to the requesting CS entity.

We have proposed some solutions to above questions. They were modeled and analyzed using CPNs as part of the service specification [11] [14] and are described briefly as follows:

- We assume that "terminate the protocol" means that the interface between the service user and the provider returns to the state immediately prior to the generation of a rejected service request that causes the end of the protocol.
- The requested MAC should return to the state it was prior to receive an acknowledgment of a negative response to the request sent by the requesting entity (such as a DSA-REQ or DSC-REQ message).
- The requesting CS entity could be informed of a denial of its request through the occurrence of a service primitive called "Confirmation Rejected" which clearly states that the request was rejected.

Table 6: Examples of service primitive sequences accepted by the FSA

Sequences of service primitives as transitions IDs	Sequences service primitives according to the abbreviations for the names of transitions
1,5	CrtReq2, CrtCf2 ó CrtReq, ConfrRj
1, 2, 3, 6	CrtReq, CrtInd, CrtRsp2, CrtCfRech
1, 2, 3, 4	CrtReq, CrtInd, CrtRsp, CrtCf
1, 2, 3, 4, 7, 10	CrtReq, CrtInd, CrtRsp2, CrtCf, ChgReq2, ChgCf2 ó CrtReq, CrtInd, CrtRsp2, CrtCf, ChgReq, ConfrRj
1, 2, 3, 4, 7, 10, 7, 10	CrtReq, CrtInd, CrtRsp, CrtCf, ChgReq2, ChgCf2, ChgReq2, ChgCf2 ó CrtReq, CrtInd, CrtRsp, CrtCf, ChgReq, ConfrRj, ChgReq, ConfrRj
1, 2, 3, 4, 7, 10, 11, 15	CrtReq, CrtInd, CrtRsp, CrtCf, ChgReq2, ChgCf2, TerReq2, TerCf2 ó CrtReq, CrtInd, CrtRsp, CrtCf, ChgReq2, ChgCf2, TerReq, ConfrRj
1, 2 , 3, 4, 7, 8, 9, 10	CrtReq, CrtInd, CrtRsp, CrtCf, ChgReq, ChgInd, ChgRsp, ChgCf ó CrtReq, CrtInd, CrtRsp, CrtCf, ChgReq, ChgInd, ChgRsp, ChgCfRech

Acknowledgements

This work was developed with financial support from the Council of Scientific and Humanistic (CDCH) Central University of Venezuela (UCV) as part of the project No. PI03-8240-2011/1.

6 Conclusions

In this paper, a service specification of the IEEE 802.16 MAC CPS connection management has been presented and described in detail following the protocol verification methodology proposed by Billington. Because CPN Tool does not support explicit generation of the language service, the FSM tool was used instead. Since, the validation of the service language represented by the minimal FSA was possible because of the small size of the automata.

During our research, we found that the IEEE 802.16 standard document has some information gaps and some aspects are not clearly specified, such as: the state of the interfaces between the service user and the service provider after the termination of the protocol; the meaning of "terminate the protocol"; the actions taken by a requesting MAC CPS entity after receiving a ACK message; and the procedure to report a connection rejection to the MAC CS entity (i.e. requesting service user). We proposed some answers to these questions and change our initial model to incorporate these new aspects [11][14]. The model was validated [11] and the service language is generated and analyzed I this paper. The analysis results are satisfactory providing us confidence on the correctness of the model.

Future work includes to model and analyzing the IEEE 802.16 MAC CPS protocol. After the validation and verification of the protocol specification, and following the protocol verification methodology, we will compare the service specification presented in this paper with the protocol specification.

References

[1] Marks R. B., Kiernan B. G., Bushue C. J. IEEE Sta. 802.16-2001. Local and Metropolitan Area Network, Part 16: Air Interface for Fixed Broadband Wireless Access Systems. October 2002.

[2] Eklud C., Marks R., Stanwood K. and Wang S. IEEE Standard 802.16: A Technical Overview of the Wireless MAN™ Air Interface for Broadband Wireless Access. IEEE Communication Magazine. June 2002.

[3] Villapol M.E. "Modelling and Analysis of the Resource Reservation Protocol Using Coloured Petri Nets". Doctoral Thesis, University of South Australia, November 2003.

[4] Jensen, K. and Kristensen, L. *Coloured Petri Nets: Modelling and Validation of Concurrent Systems.* Springer-Verlag, Berlin Heidelberg. 2009.

[5] Billington J., Gallasch G. and Bing H. A Coloured Petri Net Approach to protocol Verification. *Lectures on Concurrency and Petri Nets. 2004*, Volume 3098/2004, 49-70.

[6] Villapol M.E. and Billington J. "A Coloured Petri Net Approach to Formalising and Analysing the Resource Reservation Protocol". *Special issue of Best Papers presented at CLEI'2002, CLEI (Latinamerican Center for Informatics Studies) Electronic Journal*. ISSN 0717-5000, Vol. 6, No. 1, Paper 1 (25 pages), 2004.

[7] Villapol M.E. "Modelado y Análisis Inicial del Establecimiento de una Conexión Bluetooth Usando las Redes de Petri Coloreadas", *in Proceedings of the Thirty-Second Latin American Computing Conference, CLEI 2006*, Santiago de Chile, Chile, August 21-25(2006).

[8] Villapol M.E, *Modelado del establecimiento de la conexión entre dos dispositivos Bluetooth Usando las Redes de Petri Coloreadas.* Magazine "Avances en Sistemas e Informática", ISSN 1657-7663, Vol. 5, No. 3, Pages 219-231. December 2008.

[9] Eklud C., Marks R., Ponnuswamy S., Stanwood K y Van Waes N. *Wireless MAN: Inside IEEE 802.16 Standard for Wireless Metropolitan Area Network (Paperback).* 1ra Ed., Standard Information Network. IEEE Press. May 2006.

[10] Marks R. B., Stanwood K., Chang D. *IEEE Sta. 802.16e-2005. Local and Metropolitan Area Network, Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems, Amendment 2: Physical and Medium Access Control Layers for Combined Fixed and Mobile Operation in Licensed Bands and Corrigendum 1.* February 2006

[11] Morales, A. "Modelado y Análisis de los Procesos involucrados en la gestión de las conexiones en la capa MAC IEEE 802.16 utilizando Redes de Petri Coloreadas (CPNs)". *Faculty of Science, School of Computing, Central University of Venezuela. Working Up to Category Assistant.* UCV, April 2011.

[12] Black U. OSI: *A Model for Computer Communications Standards.* New Yersey. Prentice-Hall. 1991.

[13] Holzmann G. *Design and Validation of Computer Protocols.* Prentice Hall. 1991

[14] Morales Ana y Villapol Maria E. "Towards Formal Specification of the Service in the IEEE 802.16 MAC Layer for Connection Management". *9th WSEAS International Conference on Computational Intelligence, Man-Machine Systems and Cybernetics (CIMMACS `10).* ISSN: 1792-6998. ISBN: 978-960-474-257-8. Pp. 140-146.

[15] Ratzer, A.V, Wells, L, at el. *CPN Tools for Editing, Simulating, and Analysing Coloured Petri Net.* Lecture Notes in Computer Science, Volume 2679 / 2003, pp. 450 – 462.

[16] Barret W. and Couch J. *Compiler Construction: Theory and Practice.* Science Research Associates. Chicago, 1979

[17] Mohri M., Pereira F., and Riley M. *FSM Library.* AT&T Labs-Research. 2011. http://www.research.att.com/sw/tools/fsm/.