International Multidisciplinary Journal of Emerging Technologies and Applications (IMJETA)
Vol. 1, No. 2, pp. 70-94, May 2026
Received May 15, 2026; Revised May 20, 2026; Accepted May 29, 2026
Published May 31, 2026

IEEE 802.16 MAC Protocol for the Management of Connections and Dynamic Service Flows

Ana Morales Bezeira [0000-0001-5482-0512]
Centro de Investigación de Comunicaciones y Redes (CICORE), Universidad Central de Venezuela, Facultad de Ciencias, Escuela de Computación
Caracas, Venezuela, 1050
ana.v.morales@ucv.ve

Abstract. The IEEE 802.16 standard specifies and describes the air interface of the Broadband Wireless Access Systems point to multipoint fixed and mobile networks, and it is limited to the description of the MAC (Medium Access Control) layer and physical (PHY) layer. The standard defines informally an unmanageable set of procedures associated with the service flows (SF) at the MAC layer, also related to a connection element called dynamic service flows (DSF). This research defines the management of these connections as "IEEE 802.16 MAC Protocol for Connection Management and DSF", and studies the protocol to facilitate the reuse of some the concepts in the specification of emerging wireless networks. This research modelled and analysed the protocol using Coloured Petri Nets and the protocol verification methodology proposed by J. Billington. The results are a detailed and precise description of the protocol, and the service specification; the functional validation of such models through the determination of regular expressions of the language; the generation of the protocol language and the service specification; different drawbacks in the specification; and the PROSEGA/CPN Software to simplify the analysis of the models.

Keywords: Service Flow, Dynamic Service Flows, Protocol for Management of Connections, Coloured Petri Nets, PROSEGA/CPN.

1. Introduction

IEEE 802.16, specifies the air interface of point-multipoint Broadband Wireless Access Systems for fixed and mobile networks, and is limited to the description of the MAC and physical layer (PHY), in which all data flow is oriented to connections, to meet the requirements of emerging mobile broadband networks based on OFDMA (Orthogonal Frequency Division Multiple Access) such as 5G and 6G networks, which, according to Ahmadi (2009) "exploit, improve and expand fundamental concepts that were originally used in mobile WiMax". Likewise, in the

https://imjeta.org/index.php/imjeta

works presented by (Jeong et al., 2025; Abubakar Ibrahim et al., 2022; Ravishankar & Patil, 2017) they reuse the basic concepts of the IEEE 802.16 MAC in current wireless networks, especially in the evolution towards 5G and 6G networks and the Next Generation of Air Transportation Systems, (Kamali, 2018; Noh et al., 2020).

However, the standard defines procedures associated with service flows (SF) at the MAC layer and which are also related to a connection element called dynamic service flow (DSF) (Marks et al., 2004). Likewise, it introduces the notion of transactions for DSF management activities. These attempt to facilitate the understanding of the protocol specification, as well as the hierarchical definition of its procedures. Therefore, this research defines the management of these connections as the IEEE 802.16 MAC Protocol for Connection Management and DSF. It also studies and analyses the protocol to facilitate the reuse of some of its concepts in the specification of emerging wireless networks (Ahmadi, 2009).

The procedures associated with the protocol are complex, described by combining narrative, state machines, state transition diagrams, flow charts, and character sequence diagrams. Likewise, the notation used to describe the state transition diagrams of the DSF and the transactions associated with them is not a standard notation, ignoring relevant aspects (Morales et al., 2017; Morales & Villapol, 2013). This results in its implementation being confusing and imprecise, which could impact inadequate and correct implementations in the manufacturing of the equipment associated with the technology, thereby causing inconveniences in managing links and connections between entities that are difficult to detect.

On the other hand, studies carried out on the versions of the standard have made it possible to verify the lack of a formal and detailed description of certain events that should occur in the procedures carried out at the MAC layer for the management of connections and DSFs (Morales et al., 2017; Morales & Villapol, 2013; 2010) such as correct execution sequence of service primitives; the relationship between DSFs and the transactions that support them; lack of clarity in how the MAC CS sublayer is informed of a rejection of any connection management request from the internal layers of the requesting entity or the network; Likewise, it is not specified what it means to terminate the protocol at the service specification level, nor in what state the service interfaces should be left after a protocol termination occurs.

In this sense, designing a logically coherent protocol and testing its correctness is often a challenging task. Holzmann (1992), presents best practices that should be considered for protocol design based on validation models. These express the essential characteristics of the protocol, without taking into account implementation details. Likewise, Sarkar & Gutiérrez (2014), highlights that automated tools are required that allow interpretation of validation models to find design flaws with a high level of precision.

Thus, the complexity exhibited in the standard evidenced the importance of studying the IEEE 802.16 MAC Protocol for the management of connections and

DSFs, thereby improving the description of the existing procedures, and facilitating the reuse of these elements in the emerging networks.

In previous works the author functionally validated some of these aspects (Morales et al., 2017; Morales & Villapol, 2013; 2010) using Coloured Petri Nets (CPN) (Billington et al., 2004; Jensen & Kristensen, 2009), and CPN Tools. Also, Billington's protocol specification and verification methodology (Billington, 1991). Was used to verify properties of correctness, structure, logical consistency, and completeness of the protocol design (Kristensen & Fagerland, 2013).

Since CPNs provide the mathematical formalism of modelling and analysis that is required, they provide techniques to support the correct design and maintenance of communication protocols. Likewise, they allow the use of a convenient graphical visualization of system models, and help system designers and developers of standards and protocols, to ensure and analyses their correctness and performance properties (Kristensen & Fagerland, 2013).

It should be noted that, except for two works (Meligy et al., 2014; Morales et al., 2014), the publications associated with the IEEE 802.16 standard have not focused on formally studying and validating the protocols associated with the procedures for managing connections and DSFs. The reviewed works on IEEE 802.16 and mobile WiMax focus on aspects such as QoS planning, mobility improvements, bandwidth management, and better use of spectrum, among others.

1.1. Related Works

The literature on the IEEE 802.16 standard shows that formal techniques have rarely been applied to study it, to verify and validate the protocols and procedures described in the core of the specification. Thus, Meligy (2014), they developed an IEEE 802.16 MAC protocol model focused on a new Petri Net which they called Higher Order Stochastic Reward Net (HOSRN). In this proposal, the HOSRN modelling formalism is used to capture the QoS requirements of the different traffic classes defined in the standard. Likewise, the model was analysed using accessibility graphs.

In this same order of ideas, Morales et al., (2017), they presented an analysis of the IEEE 802.16 MAC protocol for the management of Connections and their transactions, placing special emphasis on the management of SFs and DSFs, using CPNs. They present a detailed description of the protocol, show weaknesses observed in the standard, and analyse the potential of using the transaction-oriented approach in the specification of protocols aimed at resource reservation.

In Morales et al., (2014) used CPNs to validate the behaviour of the IEEE802.16 MAC ARQ mechanism. The validation of the ARQ mechanism provided a clearer description of the mechanism and allowed the detection of a set of drawbacks, with which the researchers presented a proposal for improvements to such drawbacks, which included the proposal of a new machine state.

Previously, Narayana et al., (2006) used TLA+ as a formal method to automatically check for denial of service (DoS) vulnerabilities of network protocols, using the IEEE 802.16 standard as a case study. For their part, Wang et al., (2005) and Saddoud et al., (2014), using continuous-time Markov chain analytical models proposed schemes for the CAC (Call Admission Control) procedure. Both works modelled the behaviour of the SS (Station Subscriber). Morales and Villapol in (2010; 2013) obtain with the use and analysis of the CPNs the language of the service specification and the state machine through which the IEEE 802.16 MAC entities transit from the perspective of the occurrence of the service primitives that support the protocol.

The rest of the work is structured as follows: in the second section the methodology used will be briefly described, and the IEEE 802.16 MAC protocol for managing connections and DSFs will be described. In section three a description of the developed models and the analysis of the models will be presented. Finally, in the fourth section, the conclusions are presented.

2. Methodology: specification and verification of protocols

The steps of Billington's methodology (Billington, 1991) applied in this work were: description of the architecture; service specification and protocol specification; modelling of both the service specification and the protocol; analysis of the generated models; language generation of both the service specification and the protocol; comparison of languages.

The development of the models was carried out using CPN Tools and they were analysed using various strategies, as well as the Protocol Engineering Techniques proposed by Ming T. Liu (1989). Initially, an analysis of the simulation results was carried out at the level of behaviour and functionality. Subsequently, the state space or Occurrence Graph (OG) and its corresponding Strongly Connected Component Graph (SCC) were generated and inspected from CPN Tools. The general properties of the CPNs for the developed models were also examined. The analysis stage also required the implementation of OG minimization strategies, and its corresponding conversion to a FSA (Finite State Automaton), to facilitate and make manageable the analysis, the generation of languages and their comparison. For this, the PROSEGA/CPN software was developed (Carrasquel et al., 2018).

2.1. IEEE 802.16 MAC Protocol and Service Specification for Connection Management and DSFs

The IEEE 802.16 standard was developed to be a broadband wireless access technology for an area in which an IEEE 802.16 cell (BS, Base Station) is capable of serving fixed and mobile users (SS or MS, Mobile Station) that is within a radius of several kilometres.

The standard focuses on specifying the MAC and PHY layers, and the functions performed in these layers have been divided into two planes: the control and data plane, and the network management plane. Likewise, the MAC layer is subdivided into three sublayers, the MAC Convergence Sublayer (MAC CS), the MAC Common Part Sublayer (MAC CPS), and the Privacy Sublayer (Marks et al., 2004). The MAC CPS sublayer represents the operating core of the MAC protocol for managing connections; this sublayer provides the basic functionalities of the access system of an IEEE 802.16 network. Thus, this research is developed around the functionalities supported by this sublayer.

Connection management activities are carried out through transactions for the establishment, maintenance, and termination of connections for the transport of SF between two IEEE 802.16 entities that require communication. The establishment of connections can be initiated by either an SS/MS or the BS. This sublayer is also responsible for determining the QoS requirements used in the planning and data transmission activities over the physical layer.

The protocol consists of the exchange of management messages and activation of procedures related to the transactions that accompany them (Morales et. al., 2017), as well as the exchange of service primitives between the MAC CS sublayer and the MAC CPS sublayer in the peer entities (Morales & Villapol, 2013). These primitives represent protocol support between peer entities at the MAC layer level. Both service primitives and protocol messages are exchanged between the entities of the MAC SC sublayer and the MAC CPS sublayer, as well as between the SS/MS and the BS.

2.1.1. Service Flows (SF) and Dynamic Service Flows (DSF)

The MAC layer provides QoS support through two fundamental elements: connections (identified Connection ID) and SFs. Thus, an SF is defined as a service element for transporting information from higher layers at the MAC layer level, which provides unidirectional transport of packets flowing through a connection and includes the required QoS parameters. The SFs are provided through a set of connection management transactions, to which an element called DSF is associated, through the exchange of management messages of the DSx type. There are three basic types of SF, Provided SF, Admitted SF and Active SF, these are described by Morales et. al., (2017).

The specification of the provisioning mechanism, of the SFs as pointed out by Loutfi (2007), is outside the scope of the standard and this procedure is left in the hands of the service provider's network management system. In this sense, the IEEE 802.16 MAC protocol for the management of SF and DSF is supported by a set of service primitives, and the exchange of MAC layer management messages that are exchanged between the MAC SC sublayer entities. and the MAC CPS sublayer, between the SS/MS and the BS.

2.1.2. Service Flows and their relationship with transactions and management messages

The SS/MS may request the BS to admit an SF, using the sending of a management message called Dynamic Service Addition Request (DSA-REQ), including identification parameters (SFID) and QoS requirements. Likewise, the parameters of an admitted and not yet activated SF can be dynamically modified by the BS, through the sending of a Dynamic Service Change Request (DSC-REQ) type message. If any of the entities requires closing a connection, it will initiate the procedure using a message of the type Dynamic Service Deletion Request (DSD-REQ) (Morales et. al., 2017).

2.1.3. IEEE 802.16 Service Primitives

Services are the facilities provided by one layer (known as the service provider) to its upper layer (e.g., service user) through a Service Access Point (SAP) (Villapol, 2003). The service provides an abstraction to the upper layers, of the complexity and underlying details in the lower layers (protocols, rules, formats, and procedures), for two or more entities communicating on different devices within a network. Communication between layers of two or more entities that implement a communication protocol is defined using a set of transactions called service primitives (Villapol, 2003). which are: request, indication, response, and confirmation. Thus, the service specification is often given as a sequence of events that are possible in an abstract interface between the user (an application or other protocol) and the communication protocol.

In this sense, the IEEE 802.16 standard in (Marks et al., 2004) specifies a set of service primitives that provide support for the MAC protocol for connection management, in which an abstract way of describing the interaction between the sublayer is provided. MAC CS (service user) and the MAC CPS sublayer (the service provider), define the service as a set of transactions.

Fig. 1 shows that the occurrence of a request-type service primitive (denoted as step 1) in the MAC CS sublayer triggers the occurrence of a sequence of service primitives, which may include some or all of them (Morales & Villapol, 2013). A sequence of service primitives that includes a confirmation (service confirmed), will include all the primitives described above (i.e. request→ indication → response→ confirmation) and will be expressed explicitly by the service provider (in this case, the MAC CPS sublayer) to the service user (MAC CS sublayer) that issued the request.

An unconfirmed sequence of service primitives will include only a request, an indication, and probably a response (that is, request → indication → response). The case of unconfirmed services is not contemplated in the standard, since service requests must always be confirmed (Morales & Villapol, 2013).

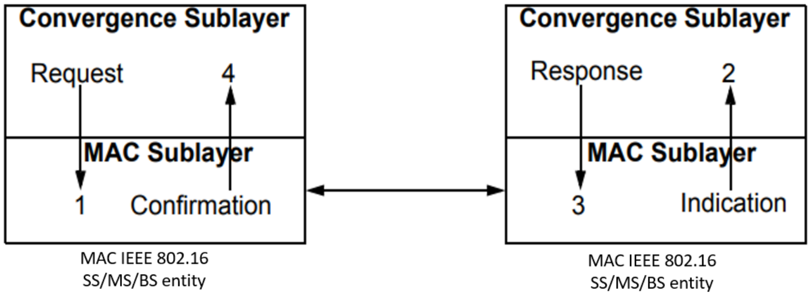


Figure 1: Communication between peer entities, and between MAC CS and MAC CPS. Adapted from (Marks, 2004)

In the IEEE 802.16 MAC protocol, under certain circumstances, a management message is not sent to the receiving peer station, so the "confirm" primitive is issued by the MAC CPS sublayer at the requesting entity (Morales & Villapol, 2013). Examples of this situation are when a connection management request is rejected by the MAC CPS sublayer at the requesting entity, or the connection management request is rejected by the network service provider. These aspects of rejections are not clearly described in the specification documents (Morales et. al., 2017; Morales & Villapol, 2013). The MAC service primitives defined in IEEE 802.16, the sequence of their occurrence for the connection management processes at the MAC layer level, as well as the messages exchanged, are shown below in Table 1.

Table 1: Service primitives and their corresponding management messages

Transaction	Primitive Service	Management Messages
Creating a connection	MAC_CREATE_CONNECTION.Request MAC_CREATE_CONNECTION.Indication MAC_CREATE_CONNECTION.Response MAC_CREATE_CONNECTION.Confirmation	DSA-REQ DSx_RVD DSA-RSP DSA-ACK
Changes to a connection	MAC_CHANGE_CONNECTION.Confirmation MAC_CHANGE_CONNECTION.Indication MAC_CHANGE_CONNECTION.Response MAC_CHANGE_CONNECTION.Confirmation	DSC-REQ DSx_RVD DSC-RSP DSC-ACK
Terminating a connection	MAC_TERMINATE_CONNECTION.Confirmation MAC_TERMINATE_CONNECTION.Indication MAC_TERMINATE_CONNECTION.Response MAC_TERMINATE_CONNECTION.Confirmation	DSD-REQ DSD-RSP

The activities related to the management of connections and DSFs can be initiated (or stimulated) by both an SS/MS or a BS. Fig. 2 shows an example of interaction between peer entities in the negotiation of a connection. This transaction interaction model is the object of modelling and analysis in this work.

2.1.4. MAC protocol for the management of DSF and its transactions

This protocol is described from a hierarchical view in which its behaviour is defined at the different levels of abstraction in each of the participating entities. The operation of the protocol is presented using a set of state transition diagrams. At the highest level of abstraction, the SFs in communicating entities adhere to only two states (Morales Bezeira, 2019). The Null state expresses the non-establishment of connections for a DSF, and the Operational state, is enabled once a connection is established (Marks et al., 2004).

The entities in communication will remain in this last state, during the occurrence of DSC type transactions. An SF will only return to the Null state if a transaction of type DSD is triggered. Each management message sequence represents a single transaction (creation, change, or termination). Transactions DSA (creation) and DSC (changes) consist of a sequence of the type REQ→RSP→ACK. While the DSD transaction consists only of a sequence of type REQ→RSP. If the DSA and DSC transactions are initiated by an SS/MS, a DSx-RVD type message will also be sent after receiving the REQ. A more detailed description of these processes is developed by the authors in (Morales Bezeira, 2019).

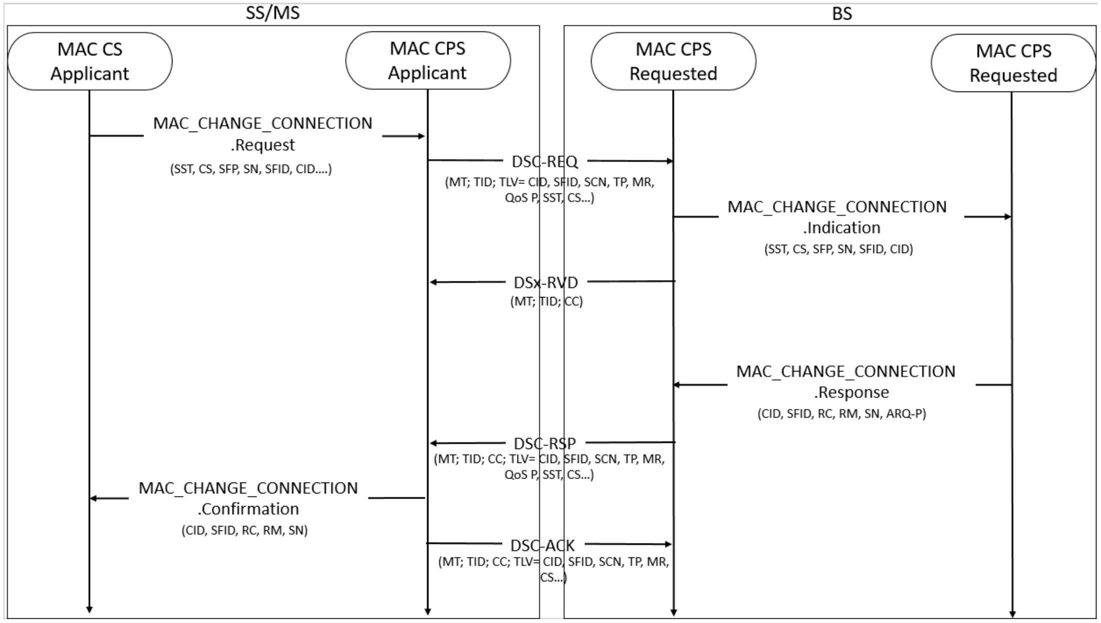


Figure 2. Logical sequence of service primitives and exchange of management messages between the MAC CS and the MAC CPS in Connection Change, initiated by an SS.

Similarly, once the transactions for the establishment of a connection associated with a DSF have been initiated, internally each DSF will go through a set of states as the internal events of the protocol occur. Fig. 3 represents the state transitions of the DSFs at a high level of abstraction and generally controls their state. The protocol operation model defines eight possible substates through which a DSF can go (Morales Bezeira, 2019). Once the DSF is stable in the *Operational* state, it can remain iterating in the *Nominal* substate, transcending the *ChangingLocal* and *ChangingRemote* substates, both in the initiating entity (local entity) and in the non-initiating entity of the protocol (entity remote). It is noteworthy that a DSF will not return to the *Null* state until it has been deleted and all ongoing transactions (within the nominal state) have been completed.

Morales et. al., (2017) present and describe the valid inputs to the state transition diagram of DSFs from entities in higher layers (e.g., user applications); valid entries to the state transition diagram of a DSx transaction, from the state transition diagram of a DSF; and valid inputs to the state transition diagram of a DSF, from the state transition diagram of a DSx transaction. This diagram can be consulted in (Morales Bezeira, 2019; Morales et. al., 2017).

Multiple causes trigger the change of state of a DSF from a transaction, such as the receipt of DSx-REQ messages, initiation of management activities from the upper layers, for example, the request for an SF Add or an SF Change. Detailed examples of the occurrence of all these events and the correlation between both state transition diagrams can be consulted in Morales Bezeira (2019).

2.2. Modelling and Analysis of Service Specification and Protocol Specification

The modelling activities were executed in several iterative phases using CPN Tools. Modelling of the service specification was carried out in two stages. Regarding the protocol, this activity was executed in three stages. The objective of the developed models was to verify and validate the functional correctness of the specification.

2.2.1. Modelling Assumptions and Decisions

To facilitate the modelling and analysis tasks, it is not specified what type of component has initiated the transaction, that is, a requesting IEEE 802.16 MAC entity and a requested IEEE 802.16 MAC entity are modelled, without specifying who represents the BS or the SS/MS; Only a point-to-point topology is considered to ensure that both the service specification and the protocol specification related to connection management behave correctly for point-to-point environments; The models were developed assuming that the service provider is reliable and that errors are resolved by it. (Morales Bezeira, 2019)

These assumptions have allowed the focus to initially be focused on validating the occurrence of the service primitives, and the internal procedures and transactions of the protocol, ignoring possible disturbances caused by errors produced by the nature of the physical medium.

To examine the correct occurrence of the service primitives and the DSF management procedures, the parameters accompanying each primitive and management message have not been considered, since the correct functioning of these two mechanisms does not depend on the information contained therein. Likewise, the service provider represents a passive entity within the model, this is because the main objective of the model is to describe the activity that occurs between the service user and the provider, at a level of abstraction above the protocol.

Regarding the sending and receiving of ACK messages, their reception is modelled through transitions in the model of the side that represents the requested entity (or non-initiator of the request). However, the reception of this message does not generate the occurrence of a service primitive in the requested entity, but it does affect at the protocol level the final state in which the requested MAC entity must remain, after receiving the ACK.

2.2.2. Service Specification Modelling

Phase 1 consisted of the development of the initial basic model of the service specification (Morales & Villapol, 2010). With this, an approximation of the behaviour of the service primitives that support the protocol was achieved. Likewise, this model allowed us to determine a set of information gaps and ambiguous aspects present in the standard.

Phase 2 consisted of developing the service specification model in greater detail. The hierarchical constructors provided by the CPN were used, to obtain a more complete model, but at the same time easy to handle, understand and analyses. In this second model, improvements were incorporated into the specification regarding the drawbacks found in phase 1 (Morales & Villapol, 2013). Likewise, in this phase, the model analysis was carried out by applying OG inspection techniques, reviewing the behavioural properties of the CPNs, validating the behaviour of the service specification, minimizing the OG to an FSA, and once the FSA was obtained, the language corresponding to the service specification was generated. In this section we will limit ourselves to outlining certain aspects derived from phase 2 of modelling; greater details of both phases can be consulted in (Morales & Villapol, 2013; 2010).

2.2.3. Hierarchical Structuring of the Model

A structure of four modules was developed, in which the "Top" module is observed in Fig.3, this provides an overview of the protocol at the level of the three procedures for managing connections (creation, changes, and termination), made up of the transitions of substitution (represented in double rectangles) that model each procedure, they are: *CreatConnection, ChangeConnection*, and *TerminateConnection*.

In the model, four (4) places are defined, which are present in all internal levels (or modules) of the hierarchy, they are: ReqMAC, which represents the requesting MAC CPS sublayer; RspMAC representing the requested CPS MAC sublayer; and the RqToRp and RpToRq places that represent the air interface through which network management messages travel. In addition to these places, a set of own places have been used in the internal modules that allow the state of the service entities of the MAC CPS sublayer to be preserved before a state change. This, is to be able to restore the state of the service entities in situations of unexpected rejections that cause the abrupt termination of the protocol. These rejections of management requests can be caused internally by the MAC CPS interface or the network provider (that is, by the air interface). In Fig. 4, you can see the module corresponding to the *ChangeConnection* procedure.

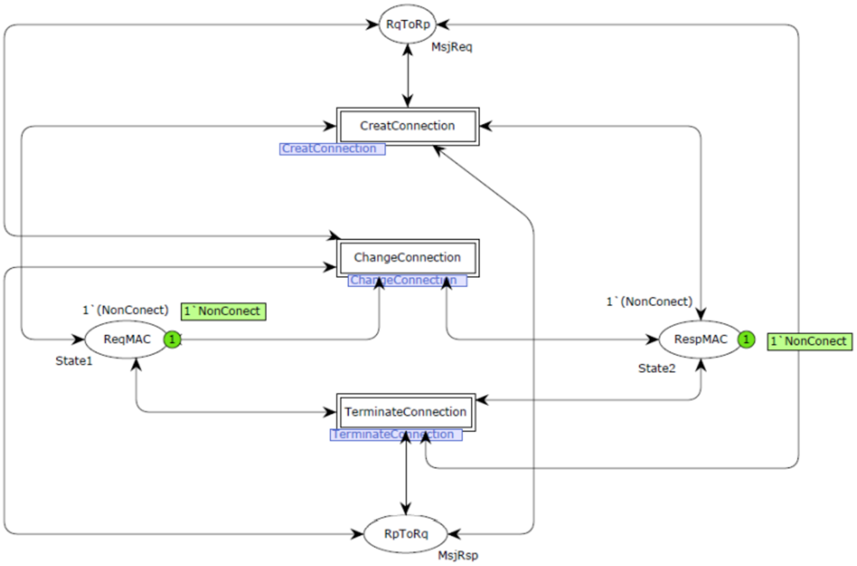


Figure 3. Top Level: IEEE 802.16 MAC Service Specification

The *ChangeConnection* module models the process of changes to the characteristics of an already established connection. This module is made up of ten transitions, five port places (ReqMAC, Control, RqToRp, RespMAC, RpToRq), as well as places specific to this module, they are: StateReqt on the requesting entity and StateResp on the side of the non-requesting entity. Table 2 lists the transitions that model the occurrence of service primitives associated with a connection change procedure. During the modelling process, a set of relevant aspects arose and some ambiguities were observed in the standard documents, related to the occurrence of the service primitives, which led the authors to propose the correct execution sequence of the same that must be executed between the MAC CS and CPS sublayers in both the requesting entity and the requested entity (Morales & Villapol, 2013).

Table 2. Transitions that model the occurrence of service primitives in the *ChangeConnection* module

Service Primitive	Transition Name
MAC_CHANGE_CONNECTION.Request	MACChgConnReq
MAC_CHANGE_CONNECTION.Request	MACChgConnReq2
MAC_CHANGE_CONNECTION.Indication	MACChgConnInd
MAC_CHANGE_CONNECTION.Response	MACChgConnRsp
MAC_CHANGE_CONNECTION.Response	MACChgConnRsp2
MAC_CHANGE_CONNECTION.Confirmation	MACChgConnCf

MAC_CHANGE_CONNECTION.Confirmation	MACChgConnCf2
MAC_CHANGE_CONNECTION.Confirmation	ConnfrRejected
MAC_CHANGE_CONNECTION.Confirmation	MACChgConnCfRech

2.2.4. Modelling of the MAC protocol for the management of DSFs and their transactions

This stage was developed in three phases. Phase 1 models the behaviour of the protocol from the perspective of the DSF state transition diagram presented in (Marks et al., 2004); Phase 2 models the behaviour of the protocol from the perspective of the transaction state transition diagrams, to contrast the behaviour of individual transactions (due to their level of complexity in the specification) with the corresponding behaviour of the DSF; in phase 3, so that a clear view of the occurrence of the service primitives could be had simultaneously with the protocol actions. (Morales Bezeira, 2019)

2.2.5. Model for the management of DSF

In this model, an abstract representation of the messages exchanged between the entities participating in the protocol was developed, as well as the procedures specified for the operation of the protocol and the internal states of the entities that maintain the processing of the messages, and the environment in which the messages are processed that protocol is executed. For this model, a hierarchical structuring of all levels of the network was also followed, in which three levels of hierarchy made up of five modules were defined.

The model defines the DSFSTT module, which is used to represent an abstract and global vision of the protocol from the point of view of the transactions that support the management of the DSF. The global state of the transactions and therefore of the DSF can be NULL or NOMINAL. The possible substates that the MAC CPS service interfaces go through can be: *AddingLocal, AddFailed, AddingRemote, ChangingLocal, ChangingRemote, Deleting, Deleted, Null, or Nominal.* These are represented as a pair (global state, substate), (Morales Bezeira, 2019). The DSFChg module (which models changes in the characteristics of a connection) is briefly described below. It is shown in Fig. 5.

2.2.6. Model for transaction management

To deepen the functional validation process of the MAC protocol for the management of connections and DSFs, below, the model developed in the internal processing of the DSA transaction will be described as a case study of the transactions, and its close linking and communicating with the DSF state transition diagram. In this model, the previously presented model was used, and a set of modifications were made to represent the internal behaviour of the transaction, (Morales Bezeira, 2019). As part of the adjustments made, new type declarations were added, the hierarchy of the modules in the model was restructured and a greater number of places and transitions to the base model were added.

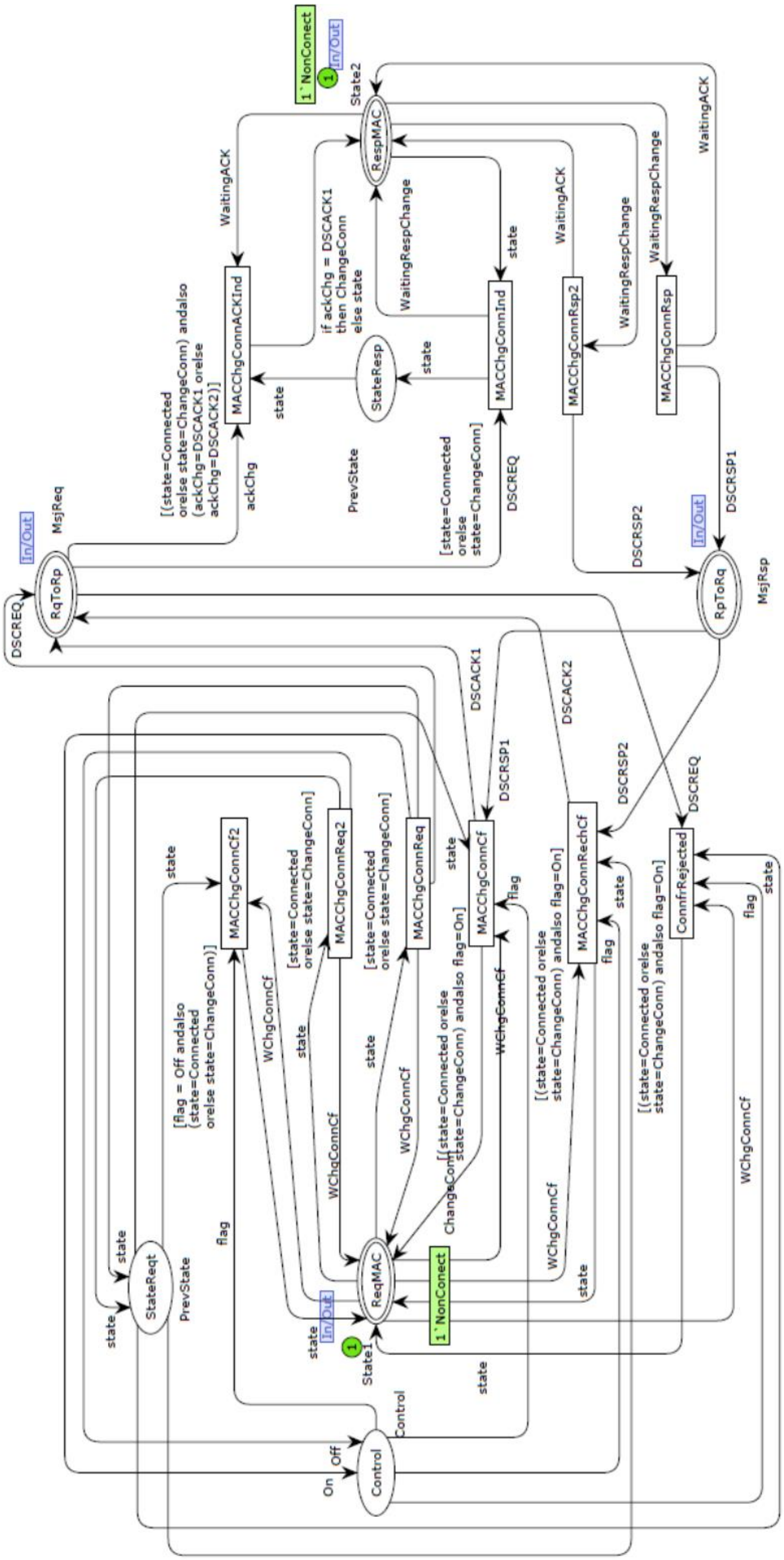


Figure 4. Participation comparative of the two groups

The resulting transaction management model is hierarchically structured into seven modules. In Fig. 6, the first three modules are presented. At the top, you can see the highest level of abstraction with two substitution transitions, on the left side of the image you can see the module corresponding to the local entity that initiates a transaction, and on the right side, you can see the module corresponding to the remote entity. Similarly, the *AddingLocal* module is presented in Fig. 7. (Morales Bezeira, 2019)

The module in Fig. 7 models the procedures that occur from the DSF level, and how from this level the details of the processing of the DSA transaction are entered. The module is made up of a substitution transition: *AddDSALocal* (highlighted in blue); and five transitions: *DSAFailed, DSASucceeded, DSAEnded, DSAErred*, and *Delete*. These last transitions model the events that cause a state change in the DSF from the moment the processing of a transaction begins. The substitution transition models in detail all the events that occur during the processing of the DSA transaction from the moment an SFAdd is received at the DSF level. (Morales Bezeira, 2019)

2.3. Analysis of the models

The analysis of the different models was conducted in different stages that involved: simulation, calculation and generation of the OG for functional analysis; analysis of the basic properties of CPNs; minimization of the OG to an FSA, which allowed obtaining a directed graph more manageable than the resulting OG; language generation of both the service specification and the protocol specification. To do this, a set of regular expressions was defined that express the sequences of events in the models, and finally, a comparison was made between said languages.

To carry out these activities, CPN Tools and PROSEGA/CPN (Carrasquel et al., 2018) were used. The PROSEGA/CPN Software was developed in this work to automate and systematize the conversion and analysis process of the OG proposed by Villapol (2003). The software integrates seamlessly into CPN Tools. In Fig. 8, the FSA resulting from the Service Specification model is presented, and Fig. 9 shows the FSA resulting from the MAC Protocol model for DSF management.

The OG of the Service Specification is composed of 11 nodes and 117 arcs; Likewise, the OG of the Protocol Specification is composed of 11 nodes and 42 arcs. The final models did not return dead transitions, dead markings, nor the presence of Deadlocks. These results indicate that all transitions in the model can occur at least once in the sequence of occurrence for each reachable state of the network from the initial marking (initial state), both from the point of view of service specification and of the Protocol Specification.

The FSA of the service specification, shown in Fig. 8, is composed of two terminal states S0 and S4, they correspond to the only two valid states for the entities in communication, these are: Not Connected and Connected. Likewise, it is made up of a total of eleven states that specify the state of the entities through the occurrence of service primitives in the negotiation and management of a connection.

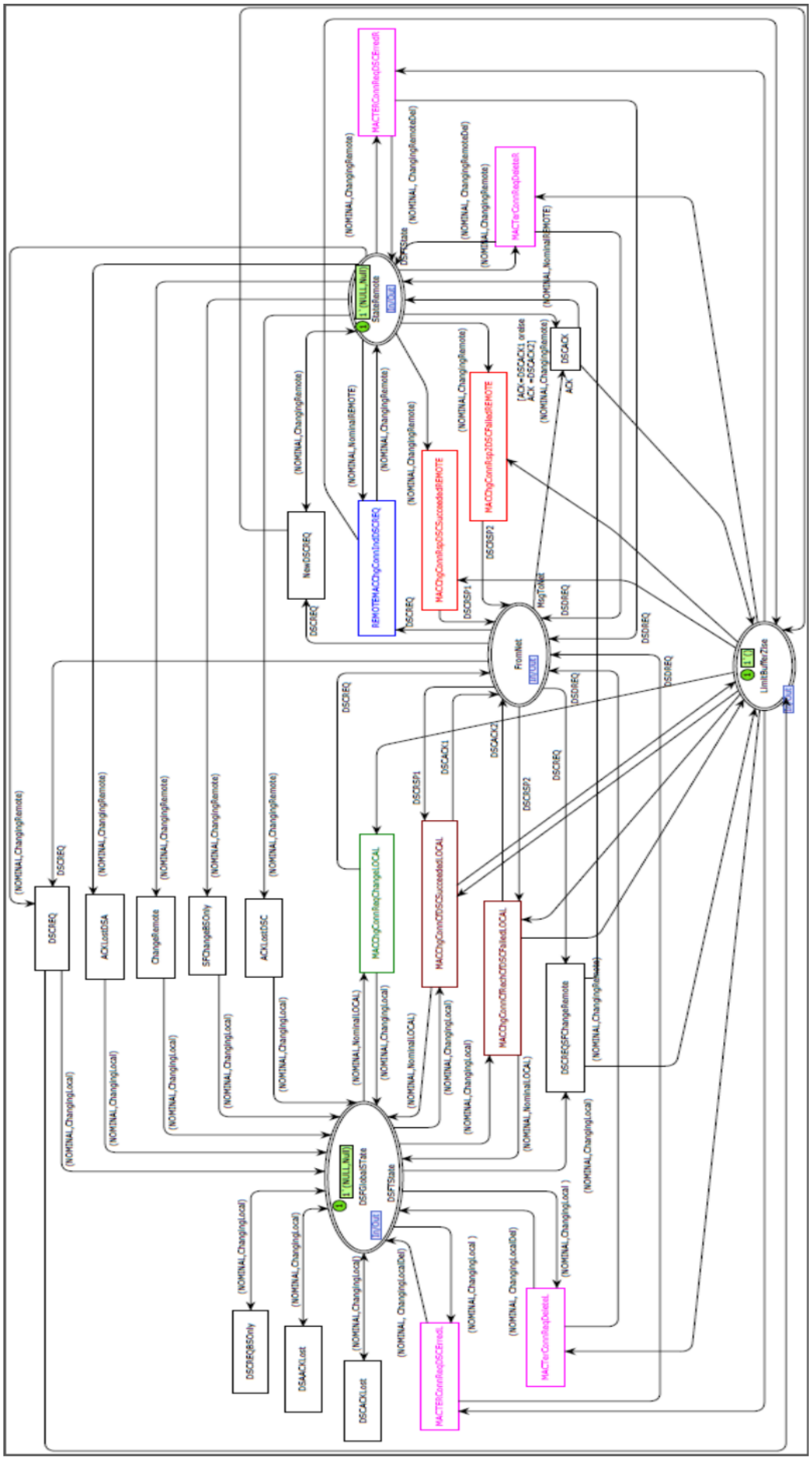


Figure 5. DSFChg module of the CPN model of DSF management.

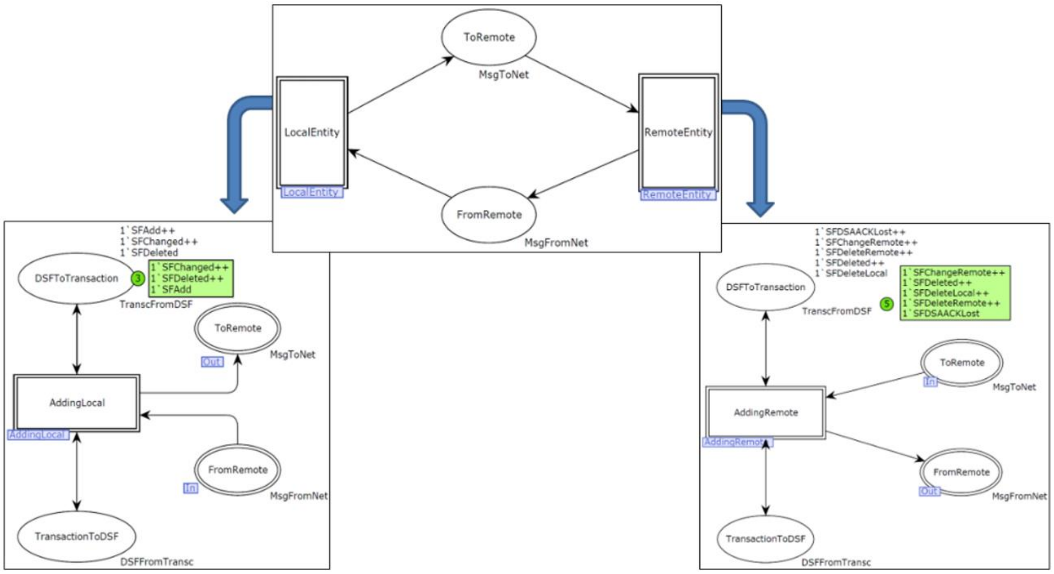


Figure 6. DSA vs DSF Transaction Model Module Hierarchy.

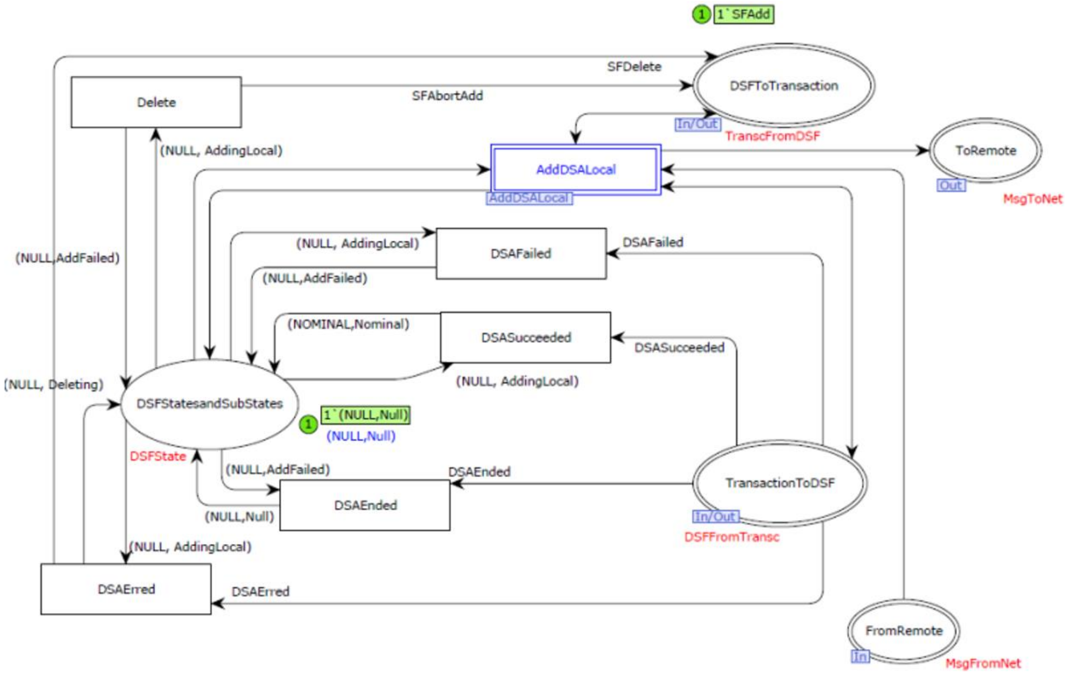


Figure 7. CPN model *AddingLocal* Module for DSA Transaction.

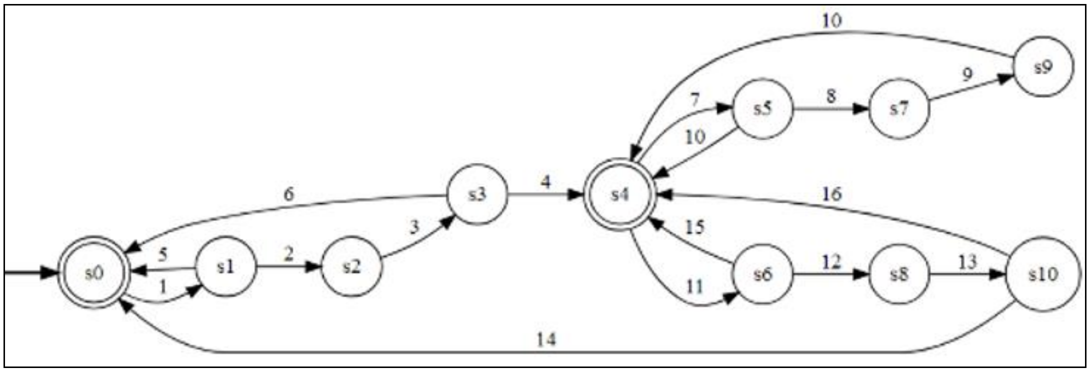


Figure 8. FSA resulting from the minimization of the OG corresponding to the Service Specification.

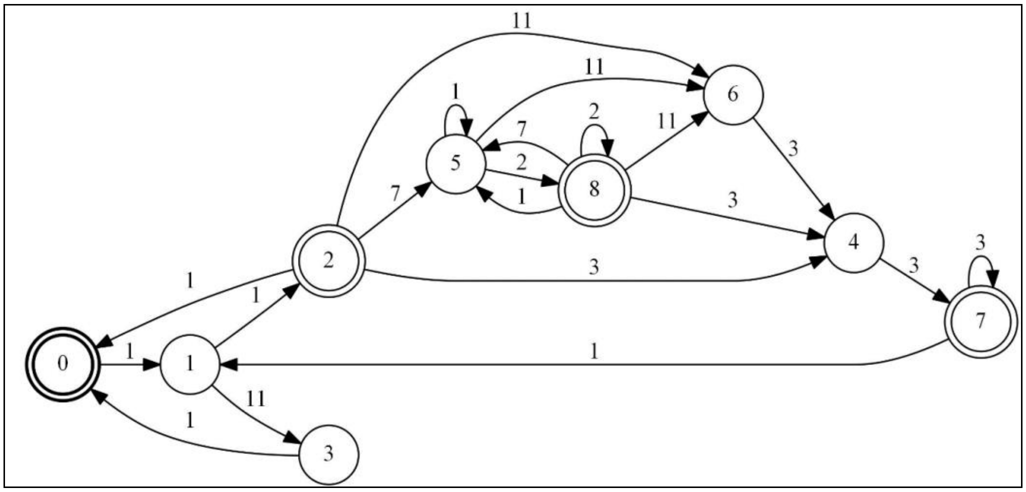


Figure 9. FSA resulting from the minimization of the OG corresponding to the MAC protocol.

The state S0 represents the case in which a service primitive of type MAC_CREAT_CONNECTION.Request, represented by the arc whose label is 1, occurs at the initiating entity and is immediately followed by a service primitive of type MAC_CREAT_CONNECTION.Confirmation, represented with the arc whose label is 5, that is, confirmation of rejection of the request (either due to the internal MAC, or due to lack of resources on the network).

In the same way, state S4 represents the cases in which a sequence of service primitives is completed, as illustrated in Fig. 1, for some of the processes involved in the management of connections (creation, changes, or termination). The establishment of a connection culminates with an affirmative MAC_CREATE_CONNECTION.Confirmation primitive, leaving the service interfaces in the *Connected* state.

Once the connection is established, multiple changes to its characteristics can be requested with a primitive of the type MAC_CHANGE_CONNECTION.Request, and such transaction culminates with a primitive of the type MAC_CHANGE_CONNECTION.Confirmation, reaching a terminal state (state S4). Likewise, this terminal state also represents the case in which termination of the connection is requested, and such request is rejected for multiple reasons (internal to the MAC, by the network, or by the requested entity RespMAC), leaving the MAC service entities IEEE 802.16 in either the *Connected* or *ChageConn* state in the S4 terminal state.

Likewise, the FSA produced for the DSF management protocol, Fig. 9, is made up of four terminal states, which specify the valid states in which the entities that establish, maintain and release connections associated with the DSF could be found and your transactions. These are (NULL, Null) node 0, (NOMINAL, Nominal) node 8, (DELETING, Deleting) node 7, (NOMINAL, Deleting) node 2. These states define the completion of a valid sequence of execution of a management transaction or service primitives.

The inscriptions in the arcs of the FSA identify the events that trigger the state transition of the entities, in the case of Fig. 8, the occurrence of a service primitive; in the case of Fig. 9, management messages of some DSF transaction.

The FSA analysis of the Service Specification not only corroborates the adherence of its description to what is defined in (Marks et al., 2004), but it also revealed that the sequence described in the standard is incomplete, so the authors propose the complete and correct sequence execution of service primitives in (Morales & Villapol, 2013). When comparing the minimized FSAs of the Protocol and the Transactions, it is possible to observe that the state transition diagram for the transactions, and the state transition diagram of the DSFs of the protocol specification and the standard are not equivalent. To validate this result, the languages corresponding to such FSAs were generated and compared.

On the other hand, because the service interfaces can always return to the initial state, this aspect is reflected in the FSAs of Figures 8 and 9, in which the arcs denoted with the inscriptions 5, 6, and 14, for the service specification, and the arc with the inscription 1 for the management of the DSF, cause the network to cyclically initiate connection establishment, change and termination procedures once the service interfaces are found in the initial state *NonConnect* for the FSA case of Fig. 8, and (NULL, Null) for the case of Fig. 9.

2.4. Service Specification Vs the IEEE 802.16 MAC Protocol for the management of DSFs and their transactions

Starting from the FSAs calculated and previously analyzed, a comparison was made between the languages resulting from both FSAs. To do this, it was necessary to define a set of regular expressions that define the language of both specifications. This was done in three stages that involved changes in the base models: stage 1, service specification model vs. protocol specification without modifications; stage 2, service specification vs protocol specification with minor modifications in the latter; and stage 3, service specification vs. protocol specification with major modifications to the protocol that involved the addition of two new places, for the state and control models corresponding to the protocol specification. (Morales Bezeira, 2019)

The service specification language has been defined in this work as $\mathbf{L_{ServSpec}}$, and the protocol language has been defined as $\mathbf{L_{MACProtDSF}}$. In this context, the alphabet "$\sum$" for the FSA in Fig. 8 is: $\Sigma = \{1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16\}$, and, the alphabet for the FSA in Fig. 9, is: $\Sigma = \{1,2,3,7,11\}$. From these alphabets, the chains on the alphabet (also called words) were determined, which are nothing more than the finite sequences of symbols of said alphabet.

Thus, for example, the set Σ^* includes all possible strings that can be formed from the alphabet symbols for the FSA of Fig. 8. Therefore, $\mathbf{L_{ServSpec}}$ recognized by the automaton of Fig. 8 is a subset of strings (or symbols) formed from the previously defined alphabet Σ following the FSA transitions, whereby $\mathbf{L_{ServSpec} \subset \Sigma^*}$.

The comparison made between languages produces a machine that is capable of generating all the strings that the automaton of Fig. 8 accepts ($\mathbf{L_{ServSpec}}$ language), and that are not accepted by the automaton of Fig. 9 ($\mathbf{L_{MACProtDSF}}$ language). i.e. ($\mathbf{L_{ServSpec}}$–$\mathbf{L_{MACProtDSF}}$). Likewise, the reverse process was carried out ($\mathbf{L_{MACProtDSF}}$–$\mathbf{L_{ServSpec}}$), which would result in a machine with all the chains accepted by the automaton in Fig. 9, and that are not accepted by the automaton in Fig. 8. Thus, the resulting language ($\mathbf{L_{res}}$) of the $\mathbf{L_{ServSpec}}$–$\mathbf{L_{MACProtDSF}}$ machine is defined as follows, (Morales Bezeira, 2019):

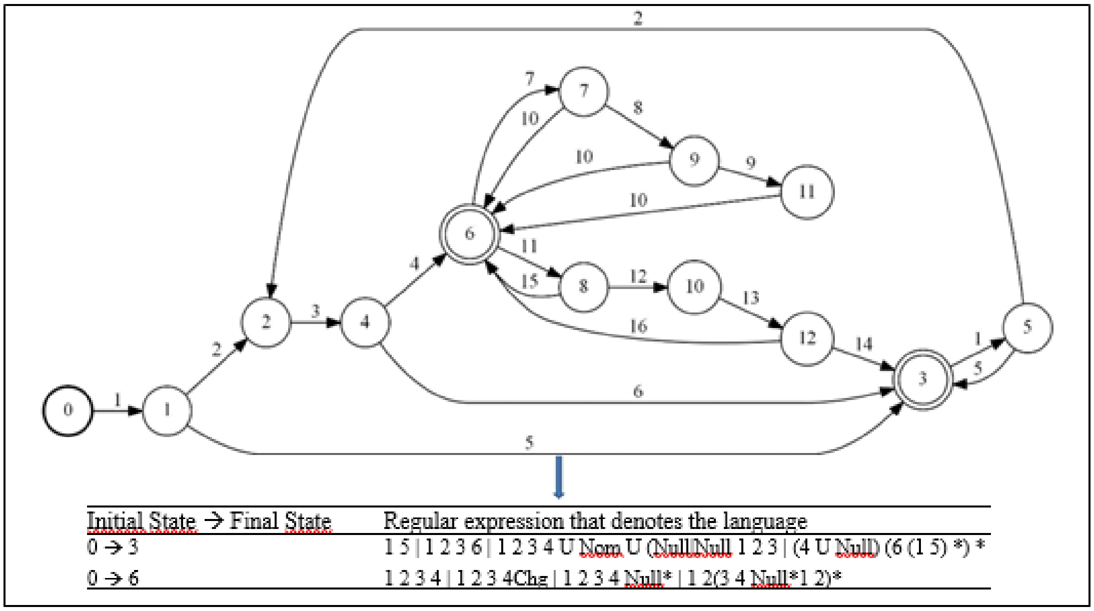
$$\mathbf{L_{res}} = L_{ServSpec} - L_{MACProtDSF} = \{p \mid p \in L_{ServSpec} \text{ and } p \notin L_{MACProtDSF}\} \qquad (1)$$

Similarly, the resulting language ($\mathbf{L_{res'}}$) of the $\mathbf{L_{MACProtDSF}}$–$\mathbf{L_{ServSpec}}$ machine is defined as follows:

$$\mathbf{L_{res'}} = L_{MACProtDSF} - L_{ServSpec} = \{p \mid p \in L_{MACProtDSF} \text{ and } p \notin L_{ServSpec}\} \qquad (2)$$

The result of applying (1), in the first stage of analysis generated a non-empty FSA, this implies that there are differences between $\mathbf{L_{ServSpec}}$–$\mathbf{L_{MACProtDSF}}$, therefore, there is a set of alphabet strings from the protocol language that does not exist in the language of the service specification.

The FSA resulting from applying (1), $\mathbf{L_{res}}$, and the regular expressions that denote the protocol language for DSF management is shown in Fig. 10, as well as the regular expressions that describe such language. While the FSA resulting from applying (2), $\mathbf{L_{res'}}$, produced an empty machine, since the FSA of the protocol specification shown in Fig. 9 is capable of generating all the chains that are possible to obtain in the FSA of the service specification presented in Fig. 8.



Initial State → Final State	Regular expression that denotes the language
0 → 3	1 5 | 1 2 3 6 | 1 2 3 4 U Nom U (NullNull 1 2 3 | (4 U Null) (6 (1 5) *) *
0 → 6	1 2 3 4 | 1 2 3 4Chg | 1 2 3 4 Null* | 1 2(3 4 Null*1 2)*

Figure 10. FSA resulting from calculating $L_{res}= L_{ServSpec}–L_{MACProtDSF}$

As a result of applying stage two, an infinite language was obtained, this is because it is always possible to return the entities in connection to the initial state (*Null* or *Nonconnect*), likewise, the appearance of multiple cycles in the FSA denotes the occurrence of infinite sequences of chains accepted by the FSA. Examples of this are the strings {1,2,3,6,(1,5)* and {1,2,3,4,(7,10)*}.

Since this second approach to the IEEE 802.16 MAC Protocol model for the management of DSFs and connections does not provide a clear correspondence between the occurrence of service primitives, both on the side of the requesting entity and the requested entity, a reengineering was carried out on the *SFAdd*, *SFChg*, and *SFDelete* modules so that there was a clear identification of the protocol events corresponding to the service specification that occur in both entities, (Morales Bezeira, 2019).

In the third approach, significant changes were made to the models, especially, a place called *LimitBufferzise* was included which would allow controlling the amount of tokens that the places that model the provider's network can earn, and thereby avoid the classic explosion problem of states that are presented in CPN networks, thereby allowing us to obtain a finite number of achievable markings in the model, and thus facilitate the analysis task. This change can be seen in the model in Fig.5. This artifice was originally proposed by Jensen and Kristesen (2009).

Table 3 shows the execution of eight different test scenarios, and the results obtained from the calculations of the state space of the modified model with the management of tokens of different sizes in the *LimitBufferSize* place.

Table 3. Calculated state space sizes for different *Limitbuffersize* place configurations

	No Limit	Limit Buffer=1	Limit Buffer=2	Limit Buffer=3	Limit Buffer=4	Limit Buffer=5	Limit Buffer=6	Limit Buffer=7	Limit Buffer=8
State Space (OG)									
# states	∞	24	702	4029	14257	39242	92031	192736	286038
# arcs	∞	38	1536	9824	37108	106861	259217	557401	843554
# marked dead	indeterminate	3	198	1137	3980	10844	25214	52427	79269
Calculation time (hh:mm:ss)	∞	0:00:06	0:00:06	0:00:06	0:00:14	0:00:47	0:04:13	0:17:20	1:00:35
Status	indeterminate	Full	Full	Full	Full	Full	Full	Full	Partial
SS Graph									
# states	∞	10	435	2502	8811	24118	56262	117250	176895
# arcs	∞	16	747	4784	18081	52104	126463	272045	424157
Calculation time (hh:mm:ss)	∞	0:00:00	0:00:00	0:00:00	0:00:04	0:00:31	0:04:00	0:15:00	0:40:00

Based on these results, the OG was partially graphed, and the configuration in which *LimitBufferSize* allows having a maximum of 2 tokens simultaneously in the FromNet place was analyzed since it was observed that the OG product of the configuration with 1 token in the *LimitBufferSize* place provides an incomplete execution sequence of protocol events. The resulting OG with *LimitBufferSize*=2, returned 702 nodes, 1536 arcs and 198 dead markings.

The dead markings were analyzed using sentences in CPN ML language that allow the information of the dead markings to be displayed in detail. The ML function ListDeadMarkings () allowed us to obtain a list of all the dead markings of the model with a configuration of 2 tokens as a limit. The dead markings obtained were inspected using the ML print (NodeDescriptor, Node) function, which shows a representation of the information associated with a specified node in the state space.

The analysis carried out on the information of the dead markings yields the expected results, all the dead markings obtained are a reflection of the inconsistent and ambiguous aspects exhibited by the IEEE 802.16 MAC protocol specification for the management of DSFs and connections in the documents of the standard, (Morales Bezeira, 2019). An example of such aspects is the scenario in which an event of type Delete may be triggered arbitrarily in the local entity, being in the *AddlingLocal* state. While this procedure is being processed, the remote entity is processing the creation of the initially requested DSF, without the remote entity being notified of the abrupt Delete procedure. Thus, a point will be reached where the local and remote entities reach an inconsistent state.

Likewise, the results of comparing the languages applying (1) and (2) are shown in Table 4. The presence of non-empty FSAs implies that in both languages there is a set of strings that are not accepted by one of the FSAs in comparison. Likewise, it was observed that the protocol specification present in the standard involves a whole set of events not contemplated in the service specification, which explains having obtained a non-empty FSA. For example, the initiation of termination procedures or connection changes in any of the entities involved, during the processing of a previous management transaction, without adequate signaling to all MAC layers in the communicating entities. This reveals the set of inaccuracies observed in the protocol specification.

Table 4. Application of $L_{res}=L_{ServSpec}-L_{MACProtDSF}$ and $L_{res'}= L_{MACProtDSF}-L_{ServSpec}$

	L_{res}	$L_{res'}$
# states	26	169
# arcs	42	324
# final states	4	28
# initial states	"0"	"0"
# accessible states	26	169
# coaccessible states	26	169
# connected states	26	169
Language	∞	∞

3. Conclusions

The establishment and management of connections by reserving resources with variable traffic characteristics in mobile and wireless environments result in extremely complex protocols, which makes their understanding and implementation difficult. In this work, the IEEE 802.16 MAC Protocol for Connection Management and DSFs was modeled and analyzed to present a more precise and detailed description of the specification, so that such concepts can be reliably reused in the future development of wireless access networks based on OFDMA.

This research determined that it is extremely difficult to understand the close relationship between transaction processing (DSA, DSC, and DSD) and DSF management since the standard does not have a detailed and descriptive narrative of such a relationship, leaving the reader the freedom of its understanding and consequently of its interpretation and implementation. This situation could directly impact the development or optimal functioning of new and emerging broadband wireless access technologies such as 5G and 6G networks; many of the diagrams present in the specification do not use the appropriate and correct standard notation.

Among the main aspects detected, we can mention: proposing the correct execution sequence of the service primitives that must be executed between the MAC CS and CPS sublayers in both the requesting entity and the requested entity; identification of situations of abrupt cancellations of management procedures, once a transaction has been initiated. This aspect causes the states to be inconsistent in the sending (local entity) and receiving (remote entity) MAC entities; clarify what it means to terminate the protocol under certain special circumstances, not specified in the official documents associated with the standard.

The comparison of languages shows that the language of the protocol does not correspond to the language of the service specification, even though the latter adheres to the specification defined by (Black, 1990). Likewise, the languages obtained contain an infinite set of chains since the entities can always return to the initial state, and the protocol specification involves a whole set of events not contemplated in the service specification, which explains having obtained non-empty FSAs. An incompleteness was observed in the sequence of occurrences of the service primitives and improvements to these are proposed.

4. References

Abubakar Ibrahim, A., Salsabil, S. I., & Lawal, I. A. (2022). Effective Utilization of an Unused Bandwidth in IEEE 802.16 Network. *Journal of Telecommunication, Electronic and Computer Engineering (JTEC),* 14(2), 15–22. https://doi.org/10.54554/jtec.2022.14.02.003

Ahmadi S. (2009). WIMAX UPDATE: An Overview of Next-Generation Mobile WiMAX Technology, *IEEE Communications Magazine*, 47(6), 84–98. https://doi.org/10.1109/MCOM.2009.5116805

Billington J. (1991). Formal Specification of Protocols: Protocol Engineering. *Encyclopedia of Microcomputers*. 7, 299–314.

Billington J., Gallasch G., & Bing H. (2004). A Coloured Petri Net Approach to protocol Verification. In J. Desel, W. Reisig, & G. Rozenberg, (Eds.). *Lectures on Concurrency and Petri Nets: Advances in Petri nets, ACPN 2003. Lecture Notes in Computer Science*, 3098 (pp. 210-290). Springer-Verlag. https://doi.org/10.1007/978-3-540-27755-2_6

Black U. (1990). *OSI: A Model for Computer Communications Standards. 1ra ed*. Prentice-Hall.

Carrasquel J., Morales A., & Villapol M. (2018). PROSEGA/CPN: An Extension of CPN Tools for Automata-based Analysis and System Verification. *In Proc. of the Institute for System Programming*, 30(4). 107-128, https://doi.org/10.15514/ISPRAS-2018-30(4)-7

Holzmann G. (1992). *Design and validation of computer protocols*. *Bell Lab*. Prentice-Hall, 1991. https://www.cs.rug.nl/~grl/ar05/spinbook/contents.pdf

Jensen K., & Kristensen L. (2009). *Coloured Petri Nets: Modelling and Validation of Concurrent Systems*. Springer-Verlag. https://dl.acm.org/doi/10.5555/1618544

Jeong M., Kim K. S., & Na J. (2025). MAC-SCH-Level Traffic Analysis for Energy Saving in 5G/Pre-6G gNBs. *16th International Conference on Information and Communication Technology Convergence (ICTC)* (pp. 506-508). Jeju, Korea. https://doi.org/10.1109/ICTC66702.2025.11388691

Kamali B. (2018). *AeroMACS: An IEEE 802.16 Standard-Based Technology for the Next Generation of Air Transportation Systems*. IEEE PRESS- Wiley.

Kristensen L.M., & Fagerland K.I. (2013). Applications of Coloured Petri Nets for Functional Validation of Protocol Designs. In: K. Jensen, A. Van Der, W.M.P., G. Balbo, M. Koutny, & K. Wolf (Eeds.). *Transaction On Petri Nets and Other Models of Concurrency. Lecture Notes in Computer Science*. 7480 (pp. 56–115). Springer-Verlag. https://doi.org/10.1007/978-3-642-38143-0_3

Loutfi N. (2007). *WiMAX: Technology for Broadband Wireless Access. 1ra ed.* John Wiley & Sons, Ltd. https://doi.org/10.1002/9780470319055

Marks R., Stanwood K., & Chang D. (2004). *IEEE Sta. 802.16TM-2004 (Revision of IEEE Std 802.16-2001), Local and Metropolitan Area Network, Part 16: Air Interface for Fixed Broadband Wireless Access Systems*. The Institute of Electrical and Electronics Engineers, Inc. https://standards.ieee.org/ieee/802.16/3507/

Meligy A., Ibrahim H., & Aqlan A. (2014). Modelling and Verification of 802.16 MAC Protocol using Higher-Order Petri Nets. *I.J. Computer Network and Information Security*. 6(4), (21-28). https://doi.org/10.5815/ijcnis.2014.04.03

Ming T., L. (1989). Protocol Engineering. In Y. Marshall C. *Advances in Computers. Elsevier*, 29. https://doi.org/10.1016/S0065-2458(08)60533-1

Morales Bezeira A. (2019). Modelado y validación funcional del protocolo MAC IEEE 802.16 para la Gestión de las conexiones y flujos de servicio dinámicos entre entidades pares. Doctoral Thesis. Universidad Central de Venezuela. https://doi.org/10.13140/RG.2.2.32931.62244

Morales A., & Villapol M. E. (2013). Reviewing the Service Specification of the IEEE 802.16 MAC Layer Connection Management: A Formal Approach. *IEEE CLEI Electronic Journal*. 16(2). 1-12. https://doi.org/10.19153/cleiej.16.2.2

Morales A., & Villapol M. E. (2010). Towards Formal Specification of the Service in the IEEE 802.16 MAC Layer for Connection Management, in F., Rivas-Echeverria & G., Mousalli-Kayat (Eds.). *Proc. 9th WSEAS International Conference on Computational Intelligence, Man-Machine Systems and Cybernetics (CIMMACS `10),* (pp. 140-146). https://www.wseas.us/e-library/conferences/2010/Merida/CIMMACS/CIMMACS-00.pdf

Morales A., Villapol M.E., & Contreras L. (2014), Improving the Specification of the Wimax ARQ Operation. *In Proceedings XL Conferencia Latinoamericana en*

Informática (CLEI 2014) (pp. 1-8). Montevideo, Uruguay, https://doi.org/10.1109/CLEI.2014.6965138

Morales A., Villapol M., Scalise E., Naranjo C., & Alvárez F. (2017). Analysis of the Dynamic Service Flow Management Transactions Protocol for MAC IEEE 802.16: An Aproach. In *Proc. XLIII Conferencia Latinoamericana de Informática (CLEI 2017)* (1-10). Córdoba, Argentina. https://doi.org/10.1109/CLEI.2017.8226455

Narayana P., Chen R., Zhao Y., Chen Y., Fu Z., & Zhou H. (2007). Automatic Vulnerability Checking of IEEE 802.16 WiMAX Protocols through TLA. *Proceedings of 2nd IEEE Workshop on Secure Network Protocols*, Santa Barbara, CA, USA (pp. 44-49). https://doi.org/10.1109/NPSEC.2006.320346

Noh, G., Chung, H., & Kim, I. (2020). Mobile relay technology for 5G. *IEEE Wireless Communications*, 27(3), 6-7.

Ravishankar H. & Patil K. K. (2017). Throughput optimized using evolutionary computing to guarantee QoS in IEEE 802.16 networks. *International Conference On Smart Technologies For Smart Nation (SmartTechCon). (*pp. 1602-1606*)*, Bengaluru, India, https://doi.org/10.1109/SmartTechCon.2017.8358635

Saddoud A., Chaari Fourati L., & Kamoun L. (2013). A new CAC and packet scheduling scheme for mobile WiMAX networks. *Trans. On Emerging Telecommunications Technologies*, 25(10). 981–992. https://doi.org/10.1002/ett.2689

Sarkar N., & Gutiérrez J. (2014). Revisiting the Issue of the Credibility of Simulation Studies in Telecommunication Networks: Highlighting the results of a comprehensive survey of IEEE publications. *IEEE Communications Magazine*, 52(5). 218–224. https://doi.org/10.1109/MCOM.2014.6815915

Villapol M.E. (2003). *Modelling and Analysis of the Resource Reservation Protocol Using Coloured Petri Nets*. Doctoral Thesis, University of South Australia. https://www.researchgate.net/publication/242393761_Modelling_and_Analysis_of_the_Resource_Reservation_Protocol_Using_Coloured_Petri_Nets

Wang H., Li W., & Agrawal D. (2005). Dynamic admission control and QoS for 802.16 Wireless MAN. *Symposium Wireless Telecommunications*, Clearwater Beach, FL, USA (pp. 60-66). https://doi.org/10.1109/WTS.2005.1524765