Analysis of the Properties of the Bluetooth Baseband Connection Establishment Using Colored Petri Nets

María Elena Villapol

Laboratory of Mobile and Wireless Networks, School of Computer Science,
Universidad Central de Venezuela,
Venezuela

maria.villapol@ciens.ucv.ve

Abstract. Bluetooth provides communication between devices via radio frequency with a range of around 10 meters. The Bluetooth specification includes a set of adopted and fundamental protocols. Baseband is a fundamental protocol that is responsible for the connection establishment among a master and up to seven slave devices. This paper describes a model of the Baseband connection establishment protocol and presents the analysis approach and results. The protocol is modeled using *Colored Petri Nets*. The model provides a clear, unambiguous and precise definition of the considered features of the baseband protocol, which is missing in the current protocol specification. The model is analyzed for a set of general properties, such as correct termination, and a set of Baseband protocol's specific properties defined in this paper. Some of the properties are checked by querying the occurrence graph, and the others are verified using a CTL-like temporal logic. The results show that the Baseband model satisfies the defined properties.

Keywords. Bluetooth, Colored Petri Nets, temporal logic, verification, occurrence graph.

Análisis de las propiedades del establecimiento de la conexión Bluetooth Bandabase usando Redes de Petri Coloreadas

Resumen. Bluetooth permite la comunicación entre dispositivos a través de frecuencias de radio en un área de alrededor de 10 metros. La especificación Bluetooth incluye un conjunto de protocolos fundamentales y adoptados. El protocolo Bandabase es un protocolo fundamental que es responsable del establecimiento de la conexión entre un maestro y hasta siete dispositivos esclavos. Este artículo describe un modelo del protocolo de establecimiento de conexión y presenta el enfoque del análisis y los resultados. El protocolo es modelado utilizando las *Redes de Petri Coloreadas*. El modelo proporciona una definición clara, inequívoca y precisa de las acciones consideradas del protocolo de Bandabase, que faltan en la especificación del protocolo actual. El modelo es analizado en función de un conjunto de propiedades generales, tales como la terminación correcta, y un conjunto de propiedades específicas del protocolo Bandabase que se definen en este documento. Algunas de las propiedades se comprueban examinando el grafo de ocurrencia, y las otras se verifican mediante una lógica temporal basada en CTL. Los resultados muestran que el modelo de Bandabase satisface las propiedades definidas.

Palabras claves. Bluetooth, Redes de Petri Coloreadas, lógica temporal, verificación, grafo de ocurrencias.

1 Introduction

Bluetooth is a standard technology which provides wireless communication among computer devices located at distances of about 10 meters. The specification, currently version 4.0 [3], has been developed by the *Bluetooth Special Interest Group* (SIG), and includes a description of the specific protocols developed exclusively for the Bluetooth wireless technology and an explanation of the profiles, which define how the technology can be used to support different applications [1, 2]. Baseband is a specific transport protocol which supports the establishment of a connection among a master device and up to seven slave devices.

Formal methods provide techniques to support the design and maintenance of communication protocols. We have found little work related to a detailed study of the Bluetooth Baseband protocol using formal methods except for our initial work [16, 17]. In a similar paper, Feldmann *et al.* [8] developed a model to study the performance of a Bluetooth *scatternet* network using *Colored Petri Nets* (CPNs). The work done by Feldmann *et al*. [8] is concentrated on Bluetooth data exchange once a connection has been established, while our work is focused on a formal study of Bluetooth establishment connection. In another related paper, Duflot M. *et al.* [6] present some performance results of the Bluetooth Baseband inquiry process using probabilistic model checking with the aid of the PRISM tool. Unlike Duflot M. *et al.* [6], we study the functional properties of the Baseband protocol which specifies how a *piconet* is established including the inquiry and page processes using CPNs and model checking. In addition, all of the related work mentioned above is based on either version 1.0 or 1.1 of the specification. In this paper, we focus on version 2.1.

CPNs [7] are a formal technique used for modeling many kinds of systems, particularly communication protocols [7, 9]. In this paper, the Bluetooth Baseband connection establishment protocol as described in the Bluetooth specification version 2.1 [2] is modeled using CPNs and analyzed with a software tool called CPN Tools [13]. Although the current version of the specification is version 4.0 released after this research work was finished, it does not change the connection establishment protocol [3]. The Bluetooth specification provides a narrative description of how a connection is established and defines several states of operation of a device to support these functions. Although a state diagram illustrating these states is included, the actions taken to move from one state to another are not defined clearly. So, one of the contributions of this work is development of a clear, unambiguous and precise definition of the Baseband connection establishment protocol, which is missing in the current specification. We also define a set of specific properties for the connection establishment based on the standard [2] and our own experience of how the technology should work. The properties are checked by querying the occurrence graph using ML [11] functions suited to the CPN Tools environment and the CTL-like temporal logic supported by the tool [3, 12].

Our first attempt of modeling and analysis of the Bluetooth Baseband protocol was presented in [16]. Since then, the model has been significantly revised, restructured and refined as demonstrated in [16]. For example, in [16], we only considered the Bluetooth establishment connection in one device. The current version of the model [16] specifies how a connection is established between two devices. The new model has been developed using an incremental modeling and validation methodology [17] to increase our confidence in its validity. Thus, following the methodology described in [17], in this paper, and for first time, we formally define a set of Bluetooth Baseband’s specific properties using set theory, standard CPN notation, and a CTL-like temporal logic, and verify the model against these properties.

The paper is organized as follows. Section 2 presents an overview of Bluetooth, which includes a description of the Baseband connection establishment protocol. Starting with a discussion of scope, Section 3 describes the Baseband connection establishment CPN model. Specific properties of the protocol are defined in Section 4 and verified in Section 5. Section 6 concludes the paper and presents future research directions.

2 Bluetooth Overview

Bluetooth [1, 2, 10] is a radio frequency (RF) technology that provides short range connectivity to personal computers, portables, PDAs, among others. Bluetooth is designed to replace traditional interfaces, such as RS-232 connectors, to provide a uniform interface to access voice and data services, to provide access to a wide area network using a personal gateway such as a cell phone, and to provide a communication infrastructure that may be used to support collaborative groups (meetings, conferences).

Bluetooth devices operate in 2.4 GHz frequencies (more specifically, the frequency band in most countries is 2.4 - 2.4835 GHz) which are also known as the *Industrial, Scientific and Medical* (ISM) bands. The radio transmission uses a technique called frequency hopping: the signal to be sent is divided in pieces and sent over a channel which is chosen randomly among 79 physical channels of 1 MHz. The transmission rate is 1 Mbps [2].

2.1 Protocol Stack

The Bluetooth protocol stack is shown in Fig.1. The protocols are classified into the following groups:

- **Fundamental protocols (core protocols):** specific protocols developed by the Bluetooth SIG.
- **Cable replacement protocol:** supplies control signaling that emulates the type of signaling usually associated with cable links.
- **Telephony control protocols:** define the call control signaling to establish voice calls and data using Bluetooth devices. They also include a protocol (AT commands) which specifies how a modem can be controlled.
- **Adopted protocols:** existing protocols used for various purposes in the upper layers.

The *Host Controller Interface* (HCI) is a standard interface for host devices to access the lowest layers of the Bluetooth protocol stack. Through the HCI, a device may direct its Baseband entity to create a link to a specific device, execute inquiries, request authentication, etc. [2, 10]. Also, data traffic passes through the HCI as it is received or transmitted by the host.

2.2 Baseband Connection Establishment

In this section, we briefly describe the procedure for establishing a Baseband connection which is modeled as part of this work. A detailed description of either the Baseband or any other Bluetooth protocol is outside the scope of this work and can be found in [2,10].

In Bluetooth, a *piconet* is a collection of devices able to communicate. A piconet contains a master device and at most 7 slave devices which are connected in an ad hoc manner as shown in Fig. 2. Additionally, a device in a piconet can be part of another piconet (as a master or slave). This kind of overlap is known as *scatternet* (see Fig.2.).

The Bluetooth baseband protocol specifies how a piconet is established and how devices are added to or removed from the piconet. Fig. 3 shows a state diagram illustrating the different states used for piconet management; this diagram has been borrowed from the Bluetooth specification [2]. There are three major states: STANDBY, CONNECTION and PARK, and seven substates: *page*, *page scan*, *inquiry*, *inquiry scan*,

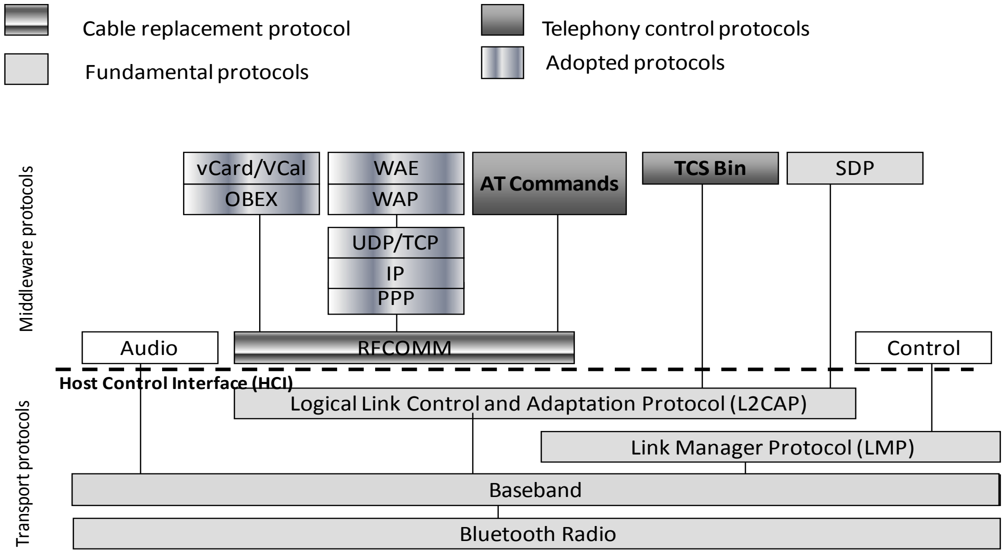

Fig. 1. The Bluetooth protocol stack

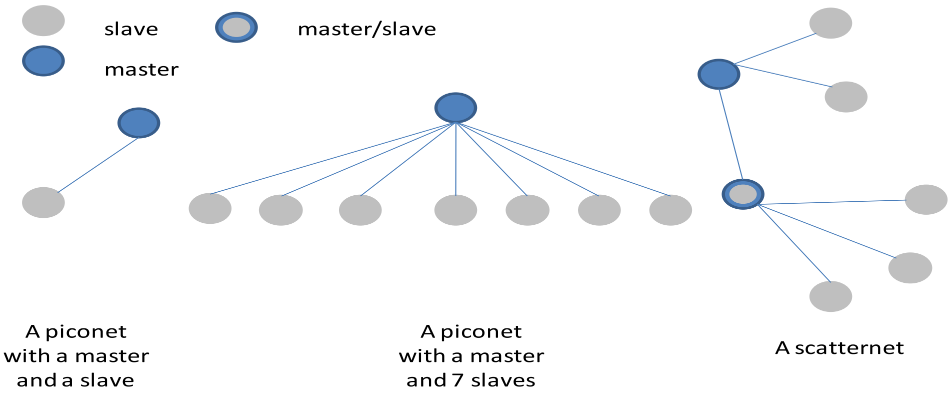
slave
master/slave
master
A piconet with a master and a slave
A piconet with a master and 7 slaves
A scatternet

Fig. 2. Examples of Bluetooth communication topologies

master response, *slave response*, and *inquiry response*. In the initial state, i.e., STANDBY, a device is not connected to any device and is not part of any piconet. During the inquiry operation, a device collects information, such as the Bluetooth device address and clock values, about other nearby devices. There are three device discovery (inquiry) substates which are described in the following paragraph.

In the *inquiry* substate, a device (potential master) transmits inquiry packets which are received by devices (potential slaves) in the *inquiry scan* substate. After receiving an inquiry packet, the slave enters the *inquiry response* substate and returns an inquiry response message. An inquiry packet is referred to as the inquiry ID packet and notifies slaves that a master is looking for other Bluetooth devices in proximity. An inquiry response packet is a FHS (*Frequency Hop Synchronization*) packet. Both the ID and FHS packets contain appropriate *access codes* (ACs), which are used for several purposes including identifying transmissions in different piconets (see also [2]). The *general inquiry access code* (GIAC) is usually used as the *inquiry access code* (IAC) in the inquiry packets.

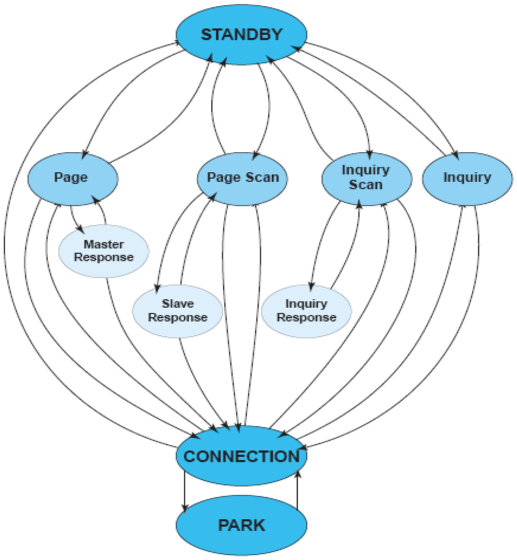

Fig. 3. State diagram of the Baseband connection establishment [2]

During the page operation, one device invites another to join its piconet. Thus, in the *page* substate, a device (potential master) may connect to any device (potential slave), which is in the *page scan* substate. In the *page scan* substate, a slave listens for page packets (slave ID packets) from the paging device (potential master). When it receives a page packet, the slave enters the *slave response* substate. In this substate, the slave waits to receive a response packet (a FHS packet) from the paging device. After receiving the packet, the slave responds with another packet (slave ID packet) and enters the CONNECTION state (i.e., it is now connected to the master). On the other hand, the paging device enters the *master response* substate after receiving a page response packet (a slave ID packet) from the paged device (a potential slave). The master transmits a packet (a FHS packet) containing the information necessary for the device to enter the CONNECTION state. Once it receives a response from the paged device (a slave ID packet), the master enters the CONNECTION state. The master sends its first traffic packet (a POLL packet) with the new connection (master) parameters. The slave that is already in the CONNECTION state may respond with any type of packet, such as a POLL packet. The slave ID and response FHS packets contain the *device access code* (DAC) of the slave, while the POLL packets contain the *channel access code* (CAC) of the master.

Once connected, a device can operate in any of the following modes: in the Active mode, it participates in a piconet and can listen, transmit and receive packets. In the Sniff mode, it can only listen on specified slots. The Hold mode is a low power mode where a device can still participate in the exchange of synchronous packets. Finally, the device can be in the PARK state where it cannot participate in the piconet but is retained as part of it. The device can be disconnected at any time returning to the default state of STANDBY. The operation modes are described in detail in [2].

3 CPN Model of the Bluetooth Baseband Connection Establishment Protocol

The Bluetooth specification does not explain clearly how the state transitions shown in Fig. 3 occur. In this section, we describe a model of the Baseband connection establishment protocol based on the specification [2] and the description given in [10]. The model is created using the CPNs with the aid of CPN Tools version 2.2.0 [13].

3.1 Assumptions and Scope

Since the Bluetooth Baseband is a complex protocol [2], the scope of the model is limited to make analysis tractable. The model was developed incrementally by adding protocol features concerning the connection establishment. Thus, this paper presents an improved version of the model presented in [16], which specifies how a connection is established between two devices (an example of the Bluetooth topology is shown in Fig.4), which are initially in the STANDBY state. In this model, only the establishment of a Baseband connection between a master and a slave is considered. This limitation is found in a number of mobile devices with restricted capabilities such as mobile phones.

According to the Bluetooth specification [2], any device can initiate the device discovery (inquiry) procedure; however, the device that initiates the paging is the master. In this work, for simplicity, we assume that the master initiates both the discovery and paging procedures. Since any device can become either a master or a slave in a piconet, this assumption affects neither the model nor the analysis results presented in this work.

Additionally, CPNs support modeling of time constraints have been omitted in this version of the model. This is because initially we are only interested in the functional specification of all the transitions shown in Fig.3.

Fig. 4. Network topology used in the CPN model

3.2 Model Hierarchy

We deal with Baseband protocol’s complexity by using the hierarchical constructs of CPNs [7]. Hierarchies are built using the notion of a *substitution transition* which may be considered a

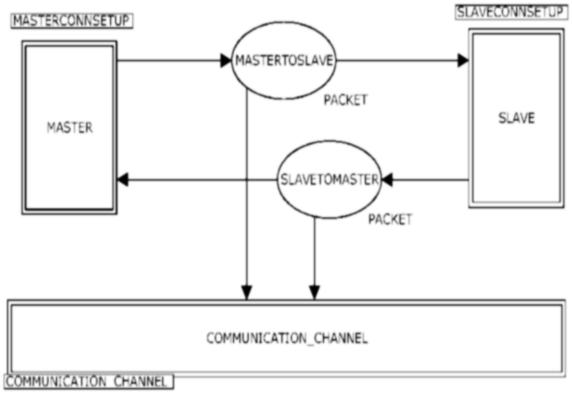

Fig. 5. Top-level page of the CPN model

macro expansion. The model starts with a top-level CPN diagram which provides an overview of the system being modeled and its environment. In the hierarchical CPNs, this top-level diagram will contain a number of substitution transitions. Each of these substitution transitions is then refined by another CPN diagram which may also contain substitution transitions. The top-level diagram and each of the substitution transitions is defined by a module called a *page*.

The top-level page is shown in Fig.5, which describes the network topology. This page includes substitution transitions (drawn as rectangles) for the Baseband connection establishment protocol implemented in the master and slave devices, and they are defined by their own pages (MASTERCONNSETUP and SLAVECONNSETUP pages, respectively). The COMMUNICATION_CHANNEL transition models an unreliable shared channel medium, so Bluetooth packets may be lost.

Another main component of a CPN is a *place* which is drawn as a circle or ellipse and may represent a condition or a state. Each place has a type or an associated set of colors (*color set*), which determines the type of data the place may contain. In Fig.5 the places MASTERTOSLAVE and SLAVETOMASTER are typed by PACKET (which is described in Section 3.3) and represent the Baseband packets traveling to either the slave or the master device, respectively. A *marking* of a place comprises a (multi-) set of values (known as *tokens*) taken from the place's type. For example, in Fig.8, in the initial marking, the place MASTERSTATE has a token with the value STANDBY. Places and transitions are connected by arcs which indicate the type of data required or produced by the substitution transitions.

The CPN subpages and pages interface through *port* and *socket places*. Subpages have port places which allow them to receive or deliver, or to receive and deliver tokens from the higher level pages. For example, the substitution transition MASTER in Fig.5 has an incoming and an outgoing places (SLAVETOMASTER and MASTERTOSLAVE, respectively), which are called sockets. Sockets are related to port places on the corresponding subpages by providing port assignments.

3.3 Global Declaration

Fig.6 shows the color sets (types), variables, and functions from the *global declaration*. The color sets BOOL, STRING and INT represent the types: Boolean, string and integer, respectively, found in other programming languages. STATE is an enumeration type representing the states and substates of Bluetooth devices (see Fig.3.). TYPE is an enumeration type and represents the type of packets which can be exchanged between the master and the slave devices during the connection establishment phase. The type AC is also an enumeration color set and represents the

```
colset E = with e;
colset INT = int;
colset BOOL = bool;
colset STRING = string;
colset STATE = with STANDBY|
        INQUIRY|INQUIRYSCAN|
        INQUIRYRESPONSE|
        PAGE|PAGESCAN|
        MASTERRESPONSE|
        SLAVERESPONSE|
        CONNECTION;
colset TYPE = with ID|FHS|POLL|SLOT;
colset AC = with IAC|GIAC|DAC|NLL|CAC;
colset PACKET = product TYPE * AC;
colset TIND =with  s1|s2;
var state: STATE;
var prevstate: STATE;
var anypacket: TYPE;
var state2: STATE;
var par: AC;
var packet: PACKET;
```

Fig. 6. Global declaration of the CPN model

access codes (AC) used during the connection establishment. PACKET is the product of the types TYPE and AC and represents a Baseband packet. TIND is an enumeration type used to control some of the actions of the Baseband protocol. The rest of the declarations define the variables (var) used in the model.

3.4 Master Connection Establishment Page

The hierarchical model of the Baseband connection establishment consists of eight pages. Not all pages of the model are described in this paper; however, a more detailed description of the model can be found in [16]. Thus, only the MASTER and INQUIRY pages, which are related to the connection establishment protocol implemented in the master device, are described. They are representative of the operation of the CPN model.

Fig. 7 shows the MASTERCONNSETUP page. It includes two substitution transitions—INQUIRY and PAGE—which model the two major connection establishment operations, i.e., Inquiry and Page, respectively (see Section 2.2). They are expanded in their respective pages. The place MASTERSTATE represents the states and substates that a master device can be in according to the explanation given in Section 2.2. The other places are port places described in Section 3.2.

Fig. 8 shows the page that models the Inquiry operation performed by a master device. It includes seven *transitions*. In CPNs, the transitions are drawn as rectangles and represent actions of the system. They are connected to places by arcs. In Fig.8, the transitions model the actions associated to the incoming and outgoing arcs of the Inquiry node shown in Fig.3. An inquiry procedure may be either initiated automatically in a periodic manner or initiated manually using an HCI Inquiry command [2]. The transitions Inquiry_Period-Ends and HCI_Inquiry model these actions, respectively. The occurrence of any of these transitions updates the substate of the master device to *inquiry* and sends an inquiry packet (ID packet) to the network.

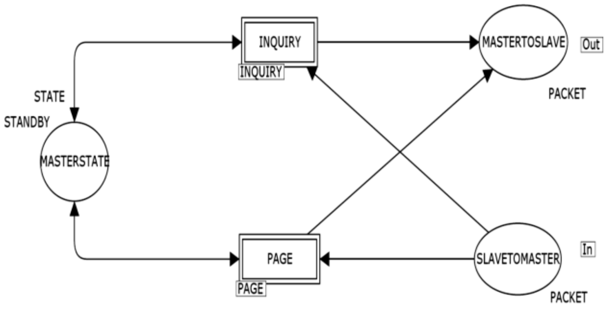

Fig. 7. Master Baseband connection establishment page

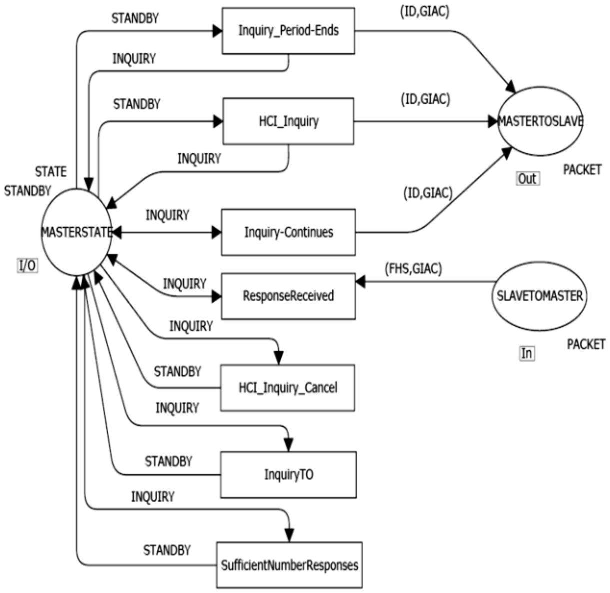

Fig. 8. Inquiry page

Once in the *inquiry* substate, the master device sends ID packets continuously, which is modeled by the transition Inquiry-Continues. Between transmissions of ID packets, the device may listen for responses (i.e., FHS packets) from a slave; this is modeled by the transition ResponseReceived. The inquiry period ends when a command is generated to cancel the inquiry operation, when a timeout expires (InquiryTO), or when the operation is stopped by the Baseband Resource Management [5] because the master has received enough responses. These actions are modeled by the transitions HCI_Inquiry_Cancel, InquiryTO and SufficientNumberResponses, respectively. If any of these transitions occurs, the device enters the STANDBY state.

4 Properties of the Baseband Connection Establishment Protocol

We wish to verify the Bluetooth Basedband protocol against a set of properties. They include Baseband protocol specific properties related to the establishment of a connection. In this section, the properties are defined informally and formally.

Firstly, we summarize the standard CPN notation [7] which we need to define the properties formally:

- M(p) denotes the marking of place *p*.
- [M > denotes the set of markings that are reachable from marking M, with [M0 > denoting the set of markings reachable from the initial marking M0.

Some properties are verified using ASK-CTL [3], which is based on the *computational tree logic* (CTL) [5] and has been adapted for expressing properties of the occurrence graph of CPNs. A detailed description of ASK-CTL is out of the scope of this paper, so we only describe the formulae relevant to our research.

The syntax of ASK-CTL has two mutually recursive categories of formulae, namely, state and transition formulae which are expressed over domains of either states or transitions of the occurrence graph. Path quantified state formulae and transitions formulae are interpreted over paths. A path is a finite or infinite sequence of states and transition occurrences. Some operators have been derived from the path quantification operators (see [3,12]). They are intended to increase the readability of the formulae in some cases. Ev ($\mathcal{A}$) and Along ($\mathcal{A}$) are derived path quantification state formulae, and $\mathcal{A}$ denotes state formulae. Ev ($\mathcal{A}$) means that $\mathcal{A}$ holds within a finite number of steps for all paths. Along ($\mathcal{A}$) means that there exists a path which either is infinite or ends in a dead marking, along which $\mathcal{A}$ holds in every state. They can be expressed as follows [3]:

- Ev $\mathcal{A} \equiv$ AU(tt,$\mathcal{A}$)
- Along $\mathcal{A} \equiv \neg$Ev $\neg\, \mathcal{A}$

where $\neg$ is the standard Boolean negation operator and *tt* is understood as the constant value true. The AU(A_1,A_2) operator expresses that A_1 holds for all paths until a marking where A_2 holds is reached.

4.1 Connection Establishment Property

The aim of the page procedure is to establish a connection between a master and a slave device. Thus, the property is defined as follows. Once a master device enters the *page* substate, it is possible to establish a connection with the slave device.

Let $\boldsymbol{M}_{PAGEMASDEV}$ be the set of markings where the substate of the master is *page*:

$$\boldsymbol{M}_{PAGEMASDEV} = \{M \in [M_0>|\ \ M(MASTERSTATE) = PAGE\}.$$

Let $\boldsymbol{M}_{CONNECTEDDEV}$ be the set of markings where the state of both the master and slave is CONNECTION:

$$\boldsymbol{M}_{CONNECTEDDEV} = \{M \in [M_0>|\ \ M(MASTERSTATE) = CONNECTION \wedge M(SLAVESTATE) = CONNECTION\}.$$

Definition 1: The CPN model satisfies the connection establishment property iff:

$$\forall M \in \boldsymbol{M}_{PAGEMASDEV},\ \exists M' \in \boldsymbol{M}_{CONNECTEDDEV}: M' \in [M>$$

4.2 Inquiry Property

The aim of the inquiry procedure is to discover devices in the vicinity. So the property is defined as follows. Once the master device enters the *inquiry* substate, it is always possible for it to discover a slave device in proximity.

Let $\mathbf{M}_{INQUIRYMASDEV}$ be the set of markings where the substate of the master device is *inquiry*:

$$\boldsymbol{M}_{INQUIRYMASDEV} = \{M \in [M_0>|\ \ M(MASTERSTATE) = INQUIRY\}.$$

Let $\mathbb{M}_{INQUIRYRESPONSESLADEV}$ be the set of markings where the substate of the slave device is *inquiry response*:

$\boldsymbol{M}_{INQUIRYRESPONSESLADEV} =\{M\in[M_0>|\ M(SLAVESTATE) = INQUIRYRESPONSE\}$.

Definition 2: The CPN model satisfies the inquiry property iff:

$\forall M \in \boldsymbol{M}_{INQUIRYMASDEV},\ \exists M' \in \boldsymbol{M}_{INQUIRYRESPONSESLADEV}: M' \in [M>$

4.3 Connection Establishment Delay Property

Bluetooth devices may experience a long connection establishment delay [14], so the property is defined as follows. The master device may remain in the *page* substate for a very long time.

Let IsPageNode(M´)= M´(MASTERSTATE) = PAGE.

Definition 3: The CPN model satisfies the connection establishment delay property iff:

$\forall M\in \boldsymbol{M}_{PAGEMASDEV,} ALONG\ (IsPageNode)$

4.4 Inquiry Delay Property

Bluetooth devices may experience a long inquiry procedure delay [14], so the property is defined as follows. The master device may remain in the *inquiry* substate for very long time.

Let IsInquiryNode(M´)= M´(MASTERSTATE) = INQUIRY

Definition 4: The CPN model satisfies the inquiry delay property iff:

$\forall M\in \boldsymbol{M}_{INQUIRYEMASDEV,} ALONG\ (IsInquiryNode)$

4.5 Disconnection Property

A device in the connected state may remain in the state of connected even though the other device is disconnected. Thus, the property is defined as follows. Once either a master or slave device enters the STANDBY state (i.e., the device is disconnected), the other device may remain in the CONNECTION state.

Definition 5: The CPN model satisfies the disconnection property iff:

$\exists M \in [M_0 >: (M\ (MASTERSTATE) = STANDBY \wedge M(SLAVESTATE) = CONNECTION) \wedge \exists M' \in [M_0 >: (M'(MASTERSTATE) = CONNECTION \wedge M'(SLAVESTATE) = STANDBY)$

5 Analysis and Verification

CPN Tools [13] is used to simulate and analyze the CPN model. The model is analyzed by generating the occurrence graph, which is also called an *Occurrence Graph (OG)* [7], and its corresponding *Strongly Connected Component* (*SCC*) graph. The OG includes all possible markings that can be reached from the initial marking. It is represented by a directed graph where the nodes represent the markings and the arcs represent the occurring binding elements. A SCC of the OG is a maximal sub-graph, whose nodes are mutually reachable from each other. A SCC graph has a node for each SCC and arcs which connect each SCC node with other SCCs.

The CPN model presented in Section 3 can generate an infinite OG due to periodic inquiry and page packets sent by the master device and unbounded communication places (i.e., the places MASTERTOSLAVE and SLAVETOMASTER). We thus modify the model so that the communication places have finite capacity using a standard approach [9]. The modified model is described in [16].

The properties are checked by implementing the definitions in Section 4 as OG query functions in ML [11] and by using the ASK-CTL library [12]. To illustrate the approach, this section describes the ML query for the connection establishment property and the inquiry delay property. All properties, except for the connection establishment delay and inquiry delay properties, are verified by examining the nodes of the OG. The delay properties are checked using the model-checking library embedded into CPN Tools.

5.1 Initialization

The CPN model is initialized by distributing tokens to one or more places of the model to create the *initial marking*. Each of the places MASTERSTATE and SLAVESTATE contains a token for the STANDBY state. Each of the communication places MASTERTOSLAVE and SLAVETOMASTER contains five empty slots indicating that there is space for 5 packets. The rest of the places do not contain any token.

Table 1. OG results

Statistical Information	OG	SCC graph
Number of Nodes	200.364	1
Number of Arcs	1.622.989	0
Calculation Time (hh:mm:ss)	(02:16:01)	(00:06:02)

5.2 Occurrence Graph and SCC Graph Statistics

In the course of our investigation [16], a series of OGs was generated for different initial markings. This was to gain further confidence in the model as it was developed. For example, in the simplest case, we just considered a channel of capacity two. Here we only present the results for the fully featured model, for the initial marking given above.

The size of the OG and its SCC graph, and the corresponding generation times using CPN Tools are shown in Table 1. CPN Tools was run on a 2.13 GHz Windows Intel Core 2 PC with 3 GB RAM. Since the size of the SCC is one, either the master or slave device can enter the STANDBY state (initial state) from any reachable state. This agrees with the state diagram shown in Fig. 3.

5.3 General Properties

Boundedness, home, and liveness properties [7] were investigated to validate and debug the model and to provide insight into the Baseband connection establishment protocol’s behavior (see Table 2 and Table 3). This information is included in the standard report for the OG generated by CPN Tools [13].

Table 2. Upper bounds for the communication places

Place	Integer Bounds	Multi-set Bounds
MASTER STATE	1	1`STANDBY++ 1`INQUIRY++ 1`PAGE++ 1`MASTERRESPONSE ++1`CONNECTION
SLAVE STATE	1	1`STANDBY++ 1`INQUIRYSCAN++ 1`INQUIRYRESPONSE ++1`PAGESCAN++ 1`SLAVERESPONSE++ 1`CONNECTION
MASTER TO SLAVE	5	5`(ID,GIAC)++ 5`(ID,DAC)++ 5`(FHS,DAC)++ 5`(POLL,CAC)++ 5`(SLOT,NLL)
SLAVE TO MASTER	5	5`(ID,DAC)++ 5`(FHS,GIAC)++ 5`(POLL,CAC)++ 5`(SLOT,NLL)

Boundedness. Integer and multi-set bounds were analyzed for the places of the model. For example, Table 2 lists the upper integer and multi-set bounds for the CPN model places. Upper Integer bounds describe the maximum number of tokens that can occur in a place, while multi-set bounds indicate which tokens can occur in a place [7] and their maximal multiplicity.

The places MASTERSTATE and SLAVESTATE can have a maximum of one token. This is expected, since they indicate the state of the master and slave, respectively. The state of either the master or the slave varies among all the defined status (see Fig. 3). Each of the communication places may have a maximum number of five tokens. This corresponds with the maximum capacity of the communication channel. The multi-set bounds show that all Bluetooth packets needed for connection establishment can be generated by the protocol. For example, the place MASTERTOSLAVE can contain a token representing an ID packet, which contains a GIAC access code and is used for collecting information about other devices in proximity.

The boundedness results are as expected and further confirm that the model is valid.

Dead Markings. A transition can occur if it is *enabled*. For a transition to be enabled in the current marking, it must be possible to *bind* (assign) data values to the variables appearing on the surrounding arc expressions and in the *guard,* and the following conditions must be met. Firstly, each of the input arc expressions evaluates to tokens that are present on the corresponding input places. Secondly, if there is any guard, it must evaluate to true. A guard is a Boolean expression which may be attached to some transitions.

Thus, a *dead marking* is a marking with no enabled binding elements; so, no transitions are enabled in the marking. In the CPN model, there are no dead markings. This agrees with the state diagram shown in Fig. 3.

Live transitions. A *live transition* is a transition which can always occur again. All the transitions of the model are live, and the system can always reach the initial state where the state of both the master and slave is STANDBY and the communication places are empty.

Home marking. A *home marking* is a marking that can always be reached from all other reachable markings. In the CPN model, all the markings are home markings. Thus, no matter what state the system has reached, it can always reach any other state including the initial state.

Table 3. Results of the home and liveness properties

Property	Result
Dead markings	none
Live transitions	all
Home markings	all
Dead transitions	none

Dead transitions. A *dead transition* is not enabled in any reachable marking. The CPN Tools standard report showed that there are no dead transitions. This is expected as there should be no 'dead code' in the specification.

5.4 The ML Query for the Connection Establishment and Inquiry Delay Properties

This section describes how two of the properties (connection establishment and inquiry delay) are checked using ML queries. The implementations of the properties connection establishment and inquiry are simplified because, in the CPN model, all markings are home markings. Thus the connection establishment property is checked using the following ML functions:

```
1  fun ConnEstProp () = let

2  val pagemasdev = PredAllNodes (fn n
     => (Mark.MASTERCONNSETUP'MASTERSTATE
     1 n = 1`PAGE));
3  val connecteddev = PredAllNodes (fn n
     => (Mark.MASTERCONNSETUP'MASTERSTATE
     1 n = 1`CONNECTION andalso
     Mark.SLAVECONNSETUP'SLAVESTATE 1 n =
     1`CONNECTION));
4  in
5  (pagemasdev <> nil) andalso
    (connecteddev <> nil)
6  end;
```

The function **ConnEstProp** (line 1) checks if a connection between a master, which is in the page substate, and a slave can be established. It returns true if both the list of all markings where the substate of the master device is equal to *page* (line 2) and the list of all markings where the state of both the master and slave devices are equal to CONNECTION (line 3) are not empty. The home marking condition is not checked in the function because it has already been checked during the generation of the OG standard report. The home marking property returns that all the markings are home marking as shown in Table 3.

The inquiry delay property is checked using the following ML functions:

```
1  fun IsInquiryNode n =
    Mark.MASTERCONNSETUP'MASTERSTATE 1 n
    = 1`INQUIRY;
2  fun InquiryLoop (n) = let
3    val ASKCTLformula = ALONG (NF("Is
    Inquiry Node",IsInquiryNode));
4  in
5     eval_node ASKCTLformula n
6  end;
7  fun InquiryDelayProp (a,b) = a andalso
    b;
8  fun InquiryDelayProperty () =
    SearchAllNodes (fn n  =>
    (Mark.MASTERCONNSETUP'MASTERSTATE 1 n
    = 1`INQUIRY), fn n => InquiryLoop
    n,true,InquiryDelayProp);
```

The function **InquiryDelayProperty** (line 8) checks if a master device can remain in the inquiry mode for a long time. It returns true if for all the markings where the substate of the master device is equal to *inquiry*, the function **InquiryLoop** (line 2-6) returns true. The evaluation function **InquiryLoop** (lines 2-6) checks if a master remains in an inquiry state for a very long time. It invokes the ASK-CTL formula, ALONG (line 3), and returns true if there exists a path which is infinite, along which the master device remains in the substate *inquiry* for the marking *n*. Finally, the combination function **InquiryDealyProp** (line 7) combines the current result of the **InquiryLoop** function with the previous result using the Boolean conjunction operator AND.

5.5 Baseband Connection Establishment Properties

Table 4. Analysis of the results

Property	Result
Connection establishment	OK
Inquiry	OK
Connection establishment delay	OK
Inquiry delay	OK
Disconnection	OK

The CPN model is verified against the specific Baseband protocol properties by running all the corresponding queries. Table 4 lists the analysis results which indicate that all the properties are satisfied (OK). The results show that the Baseband connection establishment model works as expected under the assumptions we have made.

6 Conclusions

Colored Petri Nets have been used to model the Bluetooth Baseband connection establishment protocol based on a number of assumptions. We use a simple representative network topology (a piconet with a master and a slave device) to verify that the Baseband protocol will operate correctly under these conditions. This is a first necessary step.

The main problem found during modeling was the lack of a well-defined specification of Bluetooth [2], where mainly a narrative description is present. The resulting model provides a clear, unambiguous and precise definition of the Baseband connection establishment protocol. The model was developed incrementally and checked at each stage to reduce the possibility of modeling errors.

The model is analyzed based on general properties of the protocol and a set of five properties defined and formalized for the first time in this paper. The analysis of three of the properties is carried out by querying the occurrence graph while the rest of the properties are analyzed using the CTL-like temporal logic called ASK-CTL. The analysis of the model shows that the Baseband protocol works as expected under our simplifying assumptions. Some research works, such as the one presented in [14], have described the problem of the Baseband connection establishment delay. In our work, the delay problem is demonstrated by verifying the connection establishment and inquiry delay properties.

Further work will extend the model to include time restrictions in the model, and to relax the assumptions on the piconet topology. However, analyzing this model will be a challenge, due to its inherent complexity. Thus it will be necessary to explore ways of making the model tractable for analysis, including the use of occurrence graph reduction techniques [7, 15].

Acknowledgment

We would like to thank Dr. Euripides Montagne for helpful comments. We would like to thank the Council for Scientific and Humanistic Development (Consejo de Desarrollo Científico y Humanístico, CDCH), UCV, and the Fund of Telecommunication Research and Development (Fondo de Investigación y Desarrollo de las Telecomunicaciones, FIDETEL), Venezuela, for financial support.

References

1. **Bisdikian C. (2001).** *An Overview of the Bluetooth Wireless Technology*. *IEEE Communications Magazine*, 39(12), 86–94.
2. Bluetooth SIG. *Specification of Bluetooth System*. Covered Core Packing version 2.1, July 2007. http://www.bluetooth.org/docman/handlers/downloaddoc.ashx?doc_id=241363.
3. Bluetooth SIG, Inc. *Specification of the Bluetooth System.* Version 4.0, December 2009. http://www.bluetooth.org/docman/handlers/downloaddoc.ashx?doc_id=229737.
4. **Cheng, A., Christensen, S., & Mortensen, K.H. (1996).** Model Checking Coloured Petri Nets Exploiting Strongly Connected Components. *International Workshop on Discrete Event Systems*, Edinburgh, Scotland, UK, 169–177.
5. **Clarke, E.M., Emerson, E.A., & Sistla A.P. (1986).** Automatic Verification of Finite State Concurrent System Using Temporal Logic. *ACM Transactions on Programming Language and Systems*, 8(2), 244–263.
6. **Duflot, M., Kwiatkowska, M., Norman, G., and Parker, D. (2006).** A Formal Analysis of Bluetooth Device Discovery. *International Journal on Software Tools for Technology Transfer (STTT),* 8(6), 621–632.
7. **Jensen, K. & Kristensen, L.M. (2009).** *Coloured Petri Nets: Modeling and Validation of Cocurrent Systems*, New York: Springer.
8. **Feldmann, S., Hartmann, T., & Kyamakya, K. (2003).** Modeling and Evaluation of Scatternets Performance by using Petri Nets. International Conference on Wireless Networks, ICWN '03, Las Vegas, Nevada, USA, 398–404.
9. **Jensen, K., Kristensen, L.M., & Wells, L. (2007).** Coloured Petri Nets and CPN Tools for modelling and validation of concurrent systems. *International Journal on Software Tools for Technology Transfer*, 9(3), 213–254.
10. **Miller, B.A. & Bisdikian, C. (2000).** *Bluetooth Revealed: The Insider's Guide to an Open Specification for Global Wireless Communications.* Upper Saddle River, NJ: Prentice Hall.
11. **Paulson, L.C. (1991).** ML *for the Working Programmer*. Cambridge; New York: Cambridge University.
12. **University of Aarhus – *Computer Science Department. (*1996).** *Design/CPN ASK-CTL Manual*, *Version 0.9,* Aarhus C, Denmark.
13. **Ratzer, A.V., Wells, L., Lassen, H.M., Laursen, M., Qvortrup, J.F., Stissing, M.S., Westergaard, M., Christensen, S., & Jensen, K. *(*2003).** CPN Tools for Editing, Simulating, and Analyzing Colored Petri Net. *ICATPN 2003. Lecture Notes in Computer Science,* 2679, 450–462.
14. **Salonidis, T., Bhagwat, P., & Tassiulas, L. (2000).** Proximity Awareness and Fast Connection Establishment in Bluetooth. *First Annual Workshop on Mobile Ad Hoc Networking and Computing (MobiHoc'00),* 141–142.
15. **Valmari, A. (1998).** The State Explosion Problem*.* *Lectures on Petri Nets I: Basic Models*, *Lecture Notes in Computer Science*, 1491, 429–528.
16. **Villapol, M.E. (2006).** Modelado y análisis inicial del Establecimiento de una conexión Bluetooth Usando las Redes de Petri Coloreadas. Proceedings of the Thirty-Second Latin American Computing Conference, CLEI 2006, Santiago de Chile, Chile.
17. **Villapol, M.E. (2008).** Modelado del establecimiento de la conexión entre dos dispositivos bluetooth usando las redes de Petri coloreadas, *Revista Avances en Sistemas e Informática,* 5(3), 219–231.

Maria Elena Villapol received a Computer Science degree and a Master in Computer Science from the Central University of Venezuela (Universidad Central de Venezuela, UCV), a Master in Digital Communication from Monash University, Australia, and a PhD from University of South Australia. She was a Visiting Scholar for University of Central Florida, USA, and Simon Fraser University, Canada. Currently, she is an Aggregate Professor at the School of Computer Science, UCV, Venezuela, and the coordinator of the Laboratory of Mobile and Wireless Networks. Her research area of interest is wireless networking, in particular, WiMax, WLAN/WPAN technologies, ad-hoc networks, QoS on Wireless Networks, and includes modeling, simulation and analysis of communication protocols using formal methods and other techniques and tools. She has published several technical papers in conference proceedings and journals.

Article received on 08/09/2010; accepted on 07/09/2012.