INTERNATIONAL JOURNAL OF COMMUNICATION SYSTEMS
Int. J. Commun. Syst. (2010)
DOI: 10.1002/dac.1183

Analytical modeling of bidirectional multi-channel IEEE 802.11 MAC protocols

Vandana Gupta[1], Mingwei Gong[2], S. Dharmaraja[1,*,†] and Carey Williamson[3]

[1]*Department of Mathematics, Indian Institute of Technology Delhi, New Delhi, India*
[2]*Department of Computer Science and Information Systems, Mount Royal University, Calgary, AB, Canada*
[3]*Department of Computer Science, University of Calgary, Calgary, AB, Canada*

SUMMARY

This paper presents an analytical approach to model the bi-directional multi-channel IEEE 802.11 MAC protocols (Bi-MCMAC) for *ad hoc* networks. Extensive simulation work has been done for the performance evaluation of IEEE 802.11 MAC protocols. Since simulation has several limitations, this work is primarily based on the analytical approach. The objective of this paper is to show analytically the performance advantages of Bi-MCMAC protocol over the classical IEEE 802.11 MAC protocol. The distributed coordination function (DCF) mode of medium access control (MAC) is considered in the modeling. Two different channel scheduling strategies, namely, *random channel selection* and *fastest channel first selection* strategy are also presented in the presence of multiple channels with different transmission rates. M/G/1 queue is used to model the protocols, and stochastic reward nets (SRNs) are employed as a modeling technique as it readily captures the synchronization between events in the DCF mode of access. The average system throughput, mean delay, and server utilization of each MAC protocol are evaluated using the SRN formalism. We also validate our analytical model by comparison with simulation results. The results obtained through the analytical modeling approach illustrate the performance advantages of Bi-MCMAC protocols with the fastest channel first scheduling strategy over the classical IEEE 802.11 protocol for TCP traffic in wireless *ad hoc* networks. Copyright © 2010 John Wiley & Sons, Ltd.

Received 15 November 2009; Revised 23 May 2010; Accepted 11 August 2010

KEY WORDS: IEEE 802.11 MAC protocols; wireless *ad hoc* networks; stochastic reward net; M/G/1 queue; throughput; delay

---

[*] Correspondence to: S. Dharmaraja, Department of Mathematics, Indian Institute of Technology Delhi, New Delhi, India.
[†] E-mail: dharmar@maths.iitd.ernet.in

Contract/grant sponsor: Publishing Arts Research Council; contract/grant number: 98-1846389
Contract/grant sponsor: Department of Science and Technology, India; contract/grant number: RP 01907
Contract/grant sponsor: iCORE (Informatics Circle of Research Excellence)

1. INTRODUCTION

IEEE 802.11 [1] is a set of standards for carrying out wireless local area network (WLAN) computer communication in the 2.4 and 5 GHz frequency bands. The standard defines the physical (PHY) layer specifications for wireless transmission, as well as the Medium Access Control (MAC) protocols applied to regulate access to the shared wireless channel. The IEEE 802.11 standard is widely exploited by a broad range of wireless communication devices. In the recent years, IEEE 802.11 WLAN has been widely deployed in universities, offices, airports and other public places. The 802.11 specification supports two fundamentally different MAC schemes, namely the distributed coordination function, DCF, and the point coordination function, PCF. DCF is designed to support asynchronous data transport, where all users have an equal chance of accessing the network. PCF is designed for the transmission of delay-sensitive data and uses a central access point. In this paper, we will concentrate on DCF and the *ad hoc* mode of operation with no access point. Many studies have been conducted for the performance evaluation of IEEE 802.11 MAC protocols in DCF mode [2–5].

The classical IEEE 802.11 MAC protocol considers a single shared channel. Nevertheless, multiple channels can be used. For example, in the widely deployed IEEE 802.11b standard, three non-overlapping channels can be used simultaneously. Multi-channel MAC protocols are receiving greater attention recently, particularly for wireless *ad hoc* networks as using multiple channels at the physical layer allows multiple nodes to transmit and receive data concurrently, without interfering with each other. Several multi-channel MAC (MCMAC) protocols have been proposed to improve the overall *ad hoc* network performance [6–8] and to meet the ever increasing throughput demands of applications. These approaches can be classified into different categories, depending on the channel assignment strategy and the availability of multiple wireless radios. Most approaches dedicate one channel for control packets, and use the remaining channels for data packets, whereas some approaches treat all channels identically. So and Vaidya [9] studied a multi-channel MAC protocol with a dedicated control channel. Ramachandran *et al.* [10] considered a centralized channel assignment algorithm.

Extensive research has been carried out on the performance evaluation of IEEE 802.11 protocol based on simulation [11–13]. However, simulation techniques have their own drawbacks. Simulation models can become intractable, computationally intensive, specific to parameters chosen, and difficult to be re-used by other researchers. This facilitates the need to develop an analytical model which is mathematically tractable, less time consuming and can be re-used by other researchers. Few analytical approaches are also available in the literature that models the IEEE 802.11 protocol using the discrete time Markov chain (DTMC). Bianchi in [2] have presented a simple analytical model based on the DTMC to compute the saturation throughput performance of the 802.11 DCF. In [4], a 3-D Markov chain model is proposed to derive the saturation throughput of IEEE 802.11 DCF in multi-hop networks. Cao *et al.* in [14] created a 2-D Markov chain model to characterize the 802.11 DCF system instability. But continuous time Markov chain (CTMC) is not used so far for the same to the best of our knowledge. Hence, this motivates us to develop an analytical model based on CTMC to model the multi-channel bidirectional IEEE 802.11 MAC protocol. A stochastic reward net (SRN)-based analytical model is presented in [5] for the performance analysis of IEEE 802.11 DCF, and the non-Markovian M/G/1 concept is used for the first time. Our paper extends this prior work and uses the M/G/1 concept to model the IEEE 802.11 Bidirectional Multi-Channel MAC (Bi-MCMAC) protocol proposed by Kuang *et al.* [7, 15], and study the performance advantages of its two features, namely bidirectional

DOI: 10.1002/dac

channel reservation for TCP traffic, and multi-channel operation, over the classical IEEE 802.11 protocol.

In this paper, we present an analytical framework for evaluating multi-channel MAC protocols using M/G/1 queue. To model the dynamics of the protocol and to obtain the performance measures, we applied SRN modeling technique that is an extension of stochastic Petri nets (SPN) [16]. This technique has proven to be a popular tool for modeling and performance analysis of complex discrete-event stochastic systems. And due to the availability of user-friendly software packages with graphical interfaces the development, modification, and quantitative evaluation of these SRNs is easier and less error-prone than, e.g. using a simulation language. We use SRN modeling technique as it allows for the concise specification of the system, and provide a convenient framework for generating the underlying stochastic process that governs the system's behavior. This stochastic process is then further analyzed using known techniques such as Markov chain because of the equivalence between SRN and CTMC models [17]. However, depending on the number of places in the SRN model, the state space of the underlying CTMC can become very large. Solving the CTMC would therefore be very complex and tedious. Nevertheless, solving the underlying CTMC of an SRN model can be automated using several software tools such as SHARPE [18], TimeNET [19], SPNica [20], etc. We, therefore, have used SHARPE software tool for simplifying the model analysis.

The main objective of this paper is to show how the use of multiple channels along with the bidirectional channel reservation policy enhances the performance of *ad hoc* networks based on IEEE 802.11 MAC protocol using analytical modeling. We model each station as an M/G/1 queue to get the average system throughput and mean delay. The DCF operation is applied to the first packet at head of line (HoL) at every station, and rest of the buffer is then modeled as M/G/1 queue. The performance measures are then obtained by applying the SRN modeling. We also present two different channel scheduling policies [21] that can be applied in the presence of heterogenous multiple channels, which further improves the performance of *ad hoc* networks. Consequently, there are four different SRN models presented in this paper. The first model is based on the classical IEEE 802.11 MAC protocol. This serves as a baseline for performance comparison. The second model considers multiple channels with the bidirectional channel reservation concept, while the third and fourth models combine both these features along with different channel scheduling policies. The model analysis indicates the feasibility of our analytical approach.

The remainder of the paper is organized as follows. Section 2 provides the background information on the existing IEEE 802.11 MAC protocols and the Bi-MCMAC protocol. Section 3 describes the analytical modeling for the protocols, presenting four SRN models. The performance measures are described in Section 4. The numerical results illustrating the practical applicability of the analytical models and model validation via simulation are presented in Section 5. Finally, concluding remarks and future work are presented in Section 6.

2. IEEE 802.11 MAC PROTOCOLS

2.1. Classical IEEE 802.11 MAC protocol

The default IEEE 802.11b MAC protocol is DCF. It is the fundamental MAC technique of the IEEE 802.11 wireless LAN standard. This protocol allows multiple stations to contend for a wireless channel, sharing it in a fair fashion with at most one station transmitting a (successful) frame on the channel at a time. It is a random access scheme based on the Carrier Sense Multiple Access with Collision Avoidance protocol (CSMA/CA). Countdown timers and exponential back-off are used in DCF mode to minimize the number of collisions occurring on the network from concurrent frame transmissions. A well-known problem related to the limited carrier sensing range is the hidden node problem. Nodes beyond the carrier sensing range of each other may try to send a frame to the same destination at the same time, causing excessive collisions at the receiver's end. The classical IEEE 802.11 protocol uses 'virtual carrier sensing' to overcome this problem.

DCF has two operating modes: the basic channel access mode and the RTS/CTS (Request-to-Send/Clear-To-Send) mode. In this paper, we consider the RTS/CTS mode of operation. The RTS/CTS mechanism works as follows. A source node that wants to transmit a frame reserves the channel by exchanging RTS/CTS messages with the target destination node. When a node wants to send packets, it first sends a short RTS request to the receiver node. If the channel is available for use, the receiver replies by sending a short CTS response, which allows the frame transmission to begin. If the channel is busy, no CTS is sent, and the data frame transmission is deferred, thus avoiding a collision. The RTS can be retransmitted, if needed, to elicit a CTS response when the channel becomes available. The RTS and CTS packets include the expected time duration for which the channel will be in use for data transmission. Other nodes that overhear these packets must defer their transmission for the duration specified in the RTS/CTS packets. For this reason, each node maintains a variable called the Network Allocation Vector (NAV) that records the busy time duration for the channel. This NAV effectively reserves the spatial area around the sender and receiver for frame transmission.

2.2. *Bidirectional multi-channel MAC protocols*

An MCMAC protocol extends the IEEE 802.11 MAC to use multiple physical-layer channels. With more than one channel, throughput gains are possible by allowing multiple transmissions to occur simultaneously. Because these simultaneous transmissions occur on different wireless channels, frame collisions are reduced. In an MCMAC protocol, two steps are needed prior to data transmission. First, a *channel negotiation procedure* must determine which channel is to be used for a transmission between two nodes. Second, a *channel reservation procedure* must notify other nodes regarding how long the chosen channel is reserved for this transmission episode.

The Bi-MCMAC protocol [15] has been proposed as an extension of the IEEE 802.11 MAC protocol. This protocol improves the TCP performance in multi-hop wireless *ad hoc* networks and also reduces the link-layer contention using two key ideas:

- The protocol extends the RTS/CTS handshake to do bidirectional channel reservations. By scheduling a bidirectional data transfer with a single RTS/CTS handshake, the contention between TCP data and ACK packets is reduced.
- The protocol uses multiple transmission channels, say K *channels*, at the physical layer. One of the K channels ($K>1$) is a control channel, while the other $K-1$ are data channels. By using multiple channels, the contention distance of the protocol is decreased, resulting in an improved TCP throughput in multi-hop networks.

The Bi-MCMAC protocol uses a single control channel, and multiple data channels. Each data transfer has a *control phase* and a *data exchange phase*. In the control phase, control frames are exchanged between the communicating nodes on the control channel to negotiate the data channel to be used, as well as to indicate the channel reservation time. Upon hearing the control frames,

DOI: 10.1002/dac

other nodes employ virtual carrier sensing to determine data channel reservation information. After the control phase, both the sender and the receiver tune to the chosen data channel, and the data exchange phase begins. In the data exchange phase, the sender sends a TCP packet to the receiver using a wireless frame. If the frame is correctly received, the receiver sends a MAC-layer acknowledgment back to the sender, piggybacked on a TCP packet (data or ACK) destined to that node (if any). Both nodes return to the control channel after the data exchange phase, to start another round, if desired.

In the following section, we develop separate analytical models for the classical RTS/CTS mode of operation of the IEEE 802.11 MAC protocol, and the proposed Bi-MCMAC protocol using SRN. These models capture the various aspects of both the protocols, which are discussed above. Furthermore, we also propose two different channel scheduling policies that can be applied in the presence of multiple channels with heterogenous rates, and also develop SRN models for them.

3. SRN MODELS

We model each station as an M/G/1 queue. The DCF operation is applied to the first packet at the HoL of the buffer, and the subsequent packets in the buffer are modeled using M/G/1 queue. The performance measures are obtained by applying the M/G/1 theory in the top level, and the numerical results are obtained from the SRN models at the low level. In this section, we present the four SRN models that are explained below in detail. For a detailed study on modeling with SRN, the readers are referred to [16–18]. The SRN models are developed following the decomposition approach [5], which says that the characteristics of the system can be captured by studying the behavior of the reference station in detail, and the cumulative behavior of all the other stations, as all the stations are independent and behave identically.

3.1. Model 1: IEEE 802.11b MAC

In this subsection, we develop the SRN model for the original IEEE 802.11 MAC protocol with RTS/CTS mechanism. This model assumes a single shared channel for operation. The SRN model for this protocol is illustrated in Figure 1. There are two main parts of this model. The upper part of the model (the small dashed rectangle) represents the background traffic activity generated by N ambient nodes in the *ad hoc* network, i.e. it captures the cumulative behavior of all the other stations. The lower part of the model (the large dashed rectangle) represents the foreground traffic activity (i.e. the movement of TCP data packets and TCP acknowledgements) between two representative nodes i and j in the network, i.e. it captures the behavior of the reference station in detail. The three circles in the gap between the two rectangles represent the TCP sender i (on the left), the TCP receiver j (on the right), and the shared wireless channel (in the middle). The places, timed transitions, and immediate transitions associated with all the four SRN models are listed in Tables I–III, respectively.

Note that the SRN model assumes exponentially distributed firing times for all the timed transitions, although some events in the SRN model might be deterministic (e.g. RTS, CTS, MAC−ACK) rather than random (e.g. packet inter-arrival time, background traffic). This assumption is made to facilitate SRN modeling. We first discuss the foreground traffic activity, by considering the transmission of a TCP data packet from node i to node j. The firing of the timed transition $T_{\text{TCP\_data}}$ represents the generation of a new TCP data packet at node i. The firing time of $T_{\text{TCP\_data}}$ is

DOI: 10.1002/dac

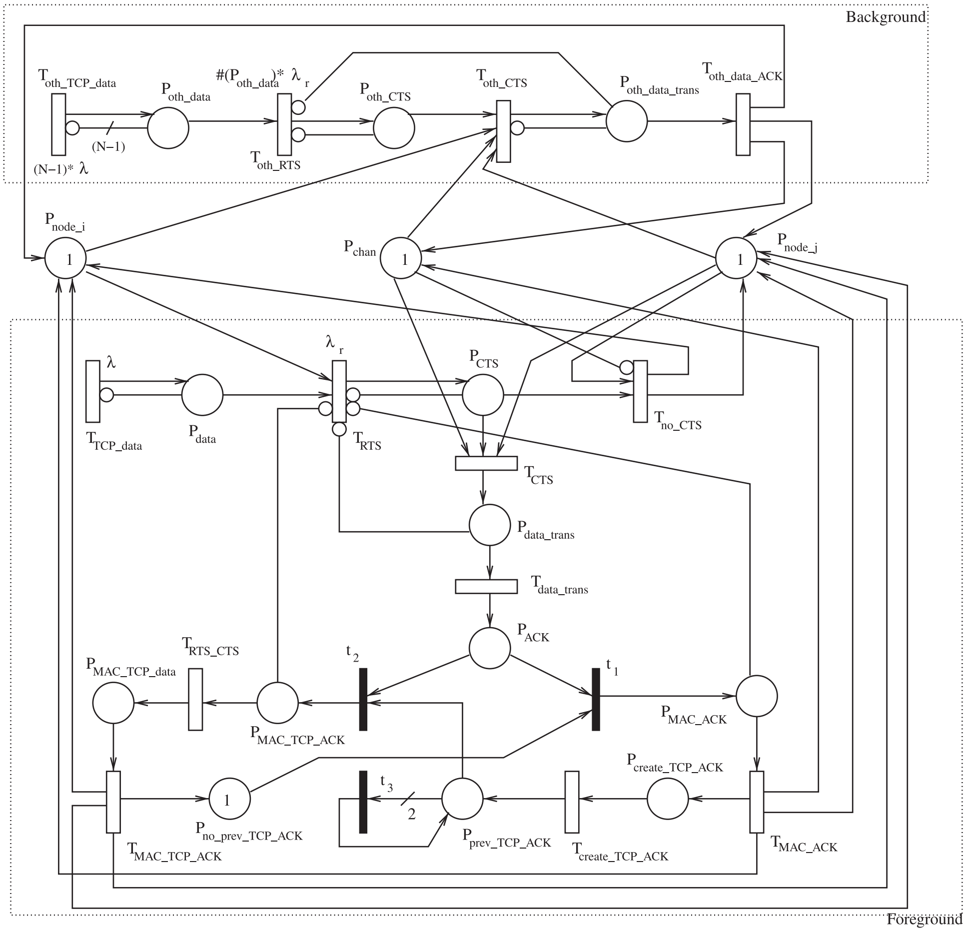

Figure 1. SRN Model for IEEE 802.11 MAC.

exponentially distributed with parameter λ. A token in the place P_{data} represents the presence of a data packet. It is observed that the DCF operations apply only to packets that are at the HoL of the buffer at each station. Hence, it is sufficient to model the DCF operations of the HoL packets at all stations. Hence, an inhibitor arc with multiplicity 1 is added between place P_{data} and transition $T_{\text{TCP\_data}}$. A token in the place $P_{\text{node\_}i}$ represents that node i is ready to send an RTS frame when there is at least one token in the place P_{data}. Before the actual transmission of the data packet, node i and node j must perform an RTS/CTS handshake. For this purpose, node i sends an RTS frame. This is represented by the firing of the timed transition T_{RTS}. The firing time of T_{RTS} is exponentially distributed with parameter λ_r. Node i cannot send an RTS frame for the next data packet until the transmission of the previous data packet is completed. That is, node i must receive the CTS frame from node j in response to the RTS frame before sending the next RTS frame. Also, it must receive MAC−ACK or MAC−TCP−ACK from node j indicating that the transmission of the data packet is complete. These are ensured by the inhibitor arcs from the places P_{CTS}, $P_{\text{MAC\_ACK}}$, $P_{\text{MAC\_TCP\_ACK}}$, and $P_{\text{data\_trans}}$ to the timed transition T_{RTS}.

DOI: 10.1002/dac

Table I. List of places in SRN models.

Place	Meaning
P_{node_i}	State of node *i*
P_{node_j}	State of node *j*
P_{chan}	State of the channel
$P_{\mathrm{con\_chan}}$	State of control channel
$P_{\mathrm{data\_chan}}$	State of data channel
P_{data}	Packet arrive at node *i*
P_{data_1}	State of data channel 1
P_{data_2}	State of data channel 2
P_{data_3}	State of data channel 3
P_{CTS}	Node *j* is ready to send CTS
$P_{\mathrm{CTS\_CRN}}$	Node *j* is ready to send CTS and CRN
$P_{\mathrm{data\_trans}}$	Packet is ready to be transmitted
P_{ACK}	Node *j* is ready to send ACK
$P_{\mathrm{MAC\_ACK}}$	Node *j* has only MAC ACK
$P_{\mathrm{non\_bidirec}}$	Node *j* has only MAC ACK
$P_{\mathrm{create\_TCP\_ACK}}$	Start creating the TCP ACK in higher layer
$P_{\mathrm{prev\_TCP\_ACK}}$	Node *j* has some previous TCP ACKs
$P_{\mathrm{MAC\_TCP\_ACK}}$	Node *j* has both MAC and TCP ACK
P_{bidirec}	Node *j* has both MAC and TCP ACKs
$P_{\mathrm{MAC\_TCP\_data}}$	MAC and TCP ACKs are ready to be send as data packet
$P_{\mathrm{no\_prev\_TCP\_ACK}}$	No previous TCP ACK at node *j*
P_{counter}	Keeps track of the data channel being used
P_{return}	Releases data channel after completion of packet transmission
$P_{\mathrm{oth\_data}}$	Packets arrive at other nodes
$P_{\mathrm{oth\_CTS}}$	One of the other nodes is ready to send CTS
$P_{\mathrm{oth\_CTS\_CRN}}$	One of the other nodes is ready to send CTS and CRN
$P_{\mathrm{oth\_data\_trans}}$	Packet is ready to be transmitted

A token in the place P_{channel} represents that the (single) shared wireless channel is available for frame transmission. Once the channel is available, node j replies to the RTS frame by sending a CTS frame. A token in the place P_{CTS} represents that a CTS frame is ready to be sent. The firing of the timed transition T_{CTS} represents node j sending a CTS frame. However, if the channel is not available, then node j does not reply with a CTS. The firing of the timed transition $T_{\mathrm{no\_CTS}}$ represents this case. The inhibitor arc from P_{channel} to $T_{\mathrm{no\_CTS}}$ ensures that node j cannot send a CTS frame if the channel is unavailable. Note that we do not consider back-off in our model. The firing of T_{CTS} deposits a token in the place $P_{\mathrm{data\_trans}}$, which represents that node i is now ready to send a data packet. The firing of the timed transition $T_{\mathrm{data\_trans}}$ represents the transmission of a data packet from node i to node j. After receiving the data packet, node j has to reply with an ACK frame. A token in the place P_{ACK} represents that node j is ready to send an ACK frame. Node j can send two types of ACKs to node i, namely MAC$-$ACK and TCP$-$ACK. First, the MAC$-$ACK is created, then the TCP$-$ACK. If node j has only the MAC$-$ACK and no pending TCP$-$ACK, then it sends only the MAC$-$ACK. However, if it has both a MAC$-$ACK and a pending TCP$-$ACK, then it sends both these ACKs together to node i. When there is a token in each of the places P_{ACK} and $P_{\mathrm{no\_prev\_TCP\_ACK}}$, the immediate transition t_1 is enabled. A token in the place $P_{\mathrm{no\_prev\_TCP\_ACK}}$ represents that no previous TCP$-$ACK is available at node j. The firing of t_1 deposits a token in the place $P_{\mathrm{MAC\_ACK}}$. A token in $P_{\mathrm{MAC\_ACK}}$ represents that node

Table II. List of timed transitions in SRN models.

Transition	Meaning (time duration)	Average time (rate^{-1})(μs)
$T_{\text{TCP\_data}}$	Packet arrival at node i	$1/\lambda$
T_{RTS}	Sending RTS by node i	362
T_{CTS}	Sending CTS by node j	314
$T_{\text{CTS\_CRN}}$	Sending CTS and CRN by node j	654
$T_{\text{no\_CTS}}$	No channel is available	50×10^3
$T_{\text{data\_trans}}$	Transmitting a packet	1303
$T_{\text{data\_trans}_1}$	Transmitting a packet via data channel 1	1034
$T_{\text{data\_trans}_2}$	Transmitting a packet via data channel 2	1303
$T_{\text{data\_trans}_3}$	Transmitting a packet via data channel 3	1763
$T_{\text{MAC\_ACK}}$	Sending MAC ACK	202
$T_{\text{create\_TCP\_ACK}}$	Creating the TCP ACK in higher layer	0.1
$T_{\text{RTS\_CTS}}$	Sending RTS by node i and CTS by node j	686
$T_{\text{MAC\_TCP\_ACK}}$	Time period to send MAC and TCP ACK	1515
$T_{\text{oth\_TCP\_data}}$	Packet arrival at other nodes	$(N-1)*(1/\lambda)$
$T_{\text{oth\_RTS}}$	Sending RTS by any one of the other nodes	Refer Equation (1)
$T_{\text{oth\_CTS}}$	Sending CTS by other nodes	314
$T_{\text{oth\_CTS\_CRN}}$	Sending CTS and CRN by other nodes	654
$T_{\text{oth\_data\_ACK}}$	Transmitting a packet and to receive ACKs ACK by other nodes	2818
$T_{\text{oth\_data\_ACK}_1}$	Transmitting a packet and to receive ACK by other nodes through data channel 1	1801
$T_{\text{oth\_data\_ACK}_2}$	Transmitting a packet and to receive ACK by other nodes through data channel 2	2817
$T_{\text{oth\_data\_ACK}_3}$	Transmitting a packet and to receive ACK by Other nodes through data channel 3	6451

Table III. List of immediate transitions in SRN models.

Transition	Meaning
t_1	Node j has only MAC ACK
t_2	Node j has both MAC and TCP ACK
t_3	Arrival of second TCP ACK at Node j
t_4	Data channel 1 is released
t_5	Data channel 2 is released
t_6	Data channel 3 is released

j is ready to send a MAC−ACK. The firing of the timed transition $T_{\text{MAC\_ACK}}$ represents node j sending a MAC−ACK to node i. Node j sends only the MAC−ACK since there is no previous TCP−ACK. The firing of $T_{\text{MAC−ACK}}$ also deposits tokens in the places $P_{\text{node\_}i}$, $P_{\text{node\_}j}$, and P_{channel}, indicating that the data packet transmission from node i to node j is complete.

TCP−ACKs are created at node j. The firing of the timed transition $T_{\text{create\_TCP\_ACK}}$ creates a new TCP−ACK in the TCP layer and deposits a token in the place $P_{\text{prev\_TCP\_ACK}}$. A token in the place $P_{\text{prev\_TCP\_ACK}}$ represents a TCP−ACK available at node j, but not yet sent to node i. Note that if the number of tokens in the place $P_{\text{prev\_TCP\_ACK}}$ reaches 2, then the immediate transition t_3 is enabled, removing two tokens from $P_{\text{prev\_TCP\_ACK}}$ and then depositing one token back in the same place. This ensures that only one token at a time is present at $P_{\text{prev\_TCP\_ACK}}$, representing the latest TCP−ACK. This models TCP's 'delayed-ACK' mechanism, wherein one cumulative TCP−ACK is typically sent for every two data packets received. When there is a token in each of the places $P_{\text{prev\_TCP\_ACK}}$ and P_{ACK}, this enables the immediate transition t_2. The firing of t_2 deposits a token in the place $P_{\text{MAC\_TCP\_ACK}}$, which represents both a MAC−ACK and TCP−ACK. When node j has a TCP−ACK, then it sends both MAC−ACK and TCP−ACK to node i as a new data frame. For this, node i and node j again perform the RTS/CTS handshake. The firing of the transition $T_{\text{RTS\_CTS}}$ represents this handshake. A token in the place $P_{\text{MAC\_TCP\_data}}$ represents this MAC−TCP−ACK as a data frame. The firing of the timed transition $T_{\text{MAC\_TCP\_ACK}}$ represents node i receiving the MAC−TCP−ACK, and sending a MAC-layer ACK frame to node j. The firing of this transition deposits tokens in the places $P_{\text{node\_}i}$, $P_{\text{node\_}j}$, and P_{channel}, indicating that a data packet exchange between node i and node j is complete.

Next we discuss the background traffic part of the SRN model, which contends for the shared wireless channel at random times. The background nodes operate independently from the foreground nodes, although the shared wireless channel is used to serialize their frame transmissions. In this part of the model, we consider the packet transmissions of $(N-1)$ other nodes as if generated from a single aggregate node. The firing of the timed transition $T_{\text{oth\_TCP\_data}}$ represents the generation of a TCP data packet at any one of the $(N-1)$ other nodes. The firing time of $T_{\text{oth\_TCP\_data}}$ is exponentially distributed with parameter $(N-1)*\lambda$. A token in the place $P_{\text{oth\_data}}$ represents a TCP data packet. Since we consider the DCF operations of the HoL packets only, an inhibitor arc with multiplicity $(N-1)$ is added between place $P_{\text{oth\_data}}$ and transition $T_{\text{oth\_TCP\_data}}$, where N is the number of stations. The firing of the timed transition $T_{\text{oth\_RTS}}$ represents one of the $(N-1)$ nodes sending an RTS frame, to perform the RTS/CTS handshake before transmitting the data packet. The firing time of $T_{\text{oth\_RTS}}$ is exponentially distributed. Its rate of firing is marking dependent, and is given by:

$$\lambda_{\text{oth\_}r} = \begin{cases} \#(P_{\text{oth\_data}})*\lambda_r & \#(P_{\text{oth\_data}})<(N-1) \\ (N-1)*\lambda_r & \#(P_{\text{oth\_data}})\geqslant(N-1) \end{cases} \tag{1}$$

The inhibitor arcs from the places $P_{\text{oth\_CTS}}$ and $P_{\text{oth\_data\_trans}}$ to $T_{\text{oth\_RTS}}$ ensure that the sender cannot send the next RTS frame until it receives the CTS frame, and the prior data packet transmission is complete. If the channel is available for transmission, then a receiving node replies to the RTS frame with a CTS frame. The firing of the timed transition $T_{\text{oth\_CTS}}$ represents a receiving node sending a CTS frame to the sender. The input arcs to $T_{\text{oth\_CTS}}$ from the places $P_{\text{node\_}i}$ and $P_{\text{node\_}j}$ ensure that node i and node j refrain from contending for the channel until the data packet transmission by the background nodes is over. The firing of the timed transition $T_{\text{oth\_CTS}}$ deposits a token in the place $P_{\text{oth\_data\_trans}}$, representing a data packet to be transmitted. The firing of the timed transition $T_{\text{oth\_data\_ACK}}$ represents the transmission of the data packet, followed by the ACK. The firing of this transition also deposit tokens in the places $P_{\text{node\_}i}$, $P_{\text{node\_}j}$, and P_{chan}, indicating that a data packet exchange by the background nodes is complete.

3.2. *Model 2*: *Bidirectional multi-channel MAC* (*Bi-MCMAC*)

In this subsection, we develop the SRN model for the Bi-MCMAC protocol [15]. There are two major differences between this model and the previous model, namely the bidirectional channel reservation concept, and the presence of multiple wireless channels. The Bi-MCMAC extends the MCMAC protocol to allow one data frame exchanged in each direction following a single

RTS/CTS/CRN handshake. The extra control frame for Channel Reservation Notification (CRN) is broadcast by the sender to advise neighbors of the channel selection and cumulative NAV duration. Like the MCMAC Protocol, Bi-MCMAC uses K physical channels, with one used for control, and the rest used as data channels. We use four channels ($K=4$) in our model: one is a control channel used for the exchange of control frames, and the other three are homogeneous data channels (i.e. same data rate) used for data packet transmission. Where it differs from the MCMAC, however, is that during the data transmission phase, the two nodes have the option of each transmitting a data frame, therefore providing a bi-directional frame exchange mechanism. This approach improves the efficiency of TCP transfers, which require the movement of TCP data packets and TCP ACK packets in opposite directions through the wireless *ad hoc* network. Next we discuss the multi-channel concept in more detail. As discussed earlier, in this model, the data transmission episode is divided into two phases: the *control phase* and the *data exchange phase*. For these two procedures, control frames (RTS/CTS/CRN) are exchanged between the communicating nodes on the *control channel*. Note that CRN is the new control frame added to extend the RTS/CTS handshake. It contains the information about the data channel to be used for a data packet transmission, and the channel reservation duration. Once the control phase is over, the data exchange phase begins, in which the transmission of the data packet takes place on the chosen *data channel* between the two communicating nodes. After hearing the CRN frame, other nodes know to avoid the specified data channel, but they are free to compete for the control channel. These nodes do not have to refrain from contending for the control channel for the whole duration of data transmission procedure, unlike the previous model.

Figure 2 shows the SRN model for Bi-MCMAC. As in the previous model, there are foreground and background traffic components. The places, timed transitions, and immediate transitions associated with the SRN are summarized in Tables I–III. As discussed earlier, we use exponentially distributed timed transitions to represent not only the exponentially distributed intervals, but also to represent all the deterministic intervals, and the random intervals that are not exponentially distributed.

The primary differences in this model are in the wireless channel representation between node i and node j. In particular, the place for the control channel is separate from that for the data channels. Furthermore, the place for the data channels can hold up to three tokens, representing the multiple physical-layer channels available. The foreground traffic part of the model is similar to that for the existing IEEE 802.11 MAC. When node i has a data packet for transmission, it first sends an RTS frame. When both the control channel and at least one data channel are available, node j sends a CTS frame in response to the RTS frame, which in turn is followed by a CRN frame. This is represented by the enabling of the timed transition $T_{\mathrm{CTS\_CRN}}$. The firing of this transition also deposits a token in the place $P_{\mathrm{con\_chan}}$, indicating that other nodes can compete for the control channel once the RTS/CTS/CRN handshake is over. After the handshake, node i sends the data packet to node j. After node j receives the data packet, it replies with an ACK frame (MAC−ACK or TCP−ACK or both). If no previous TCP−ACK is available, then it sends only the MAC−ACK. However, if it has a pending TCP−ACK packet, then it sends both the MAC−ACK and TCP−ACK to node i. Meanwhile, new TCP−ACKs are created at node j. Note that, in this model, node j sends this MAC−TCP−ACK as an ACK frame and not as a new data frame, hence no additional RTS/CTS handshake is performed before sending the MAC−TCP−ACK. The data exchange phase is over when node i receives the MAC−ACK or TCP−ACK (or both).

The background traffic operation is similar. As in the previous model, we consider the packet transmissions of all the other $N-1$ nodes as if originating from a single aggregate node. If any one

DOI: 10.1002/dac

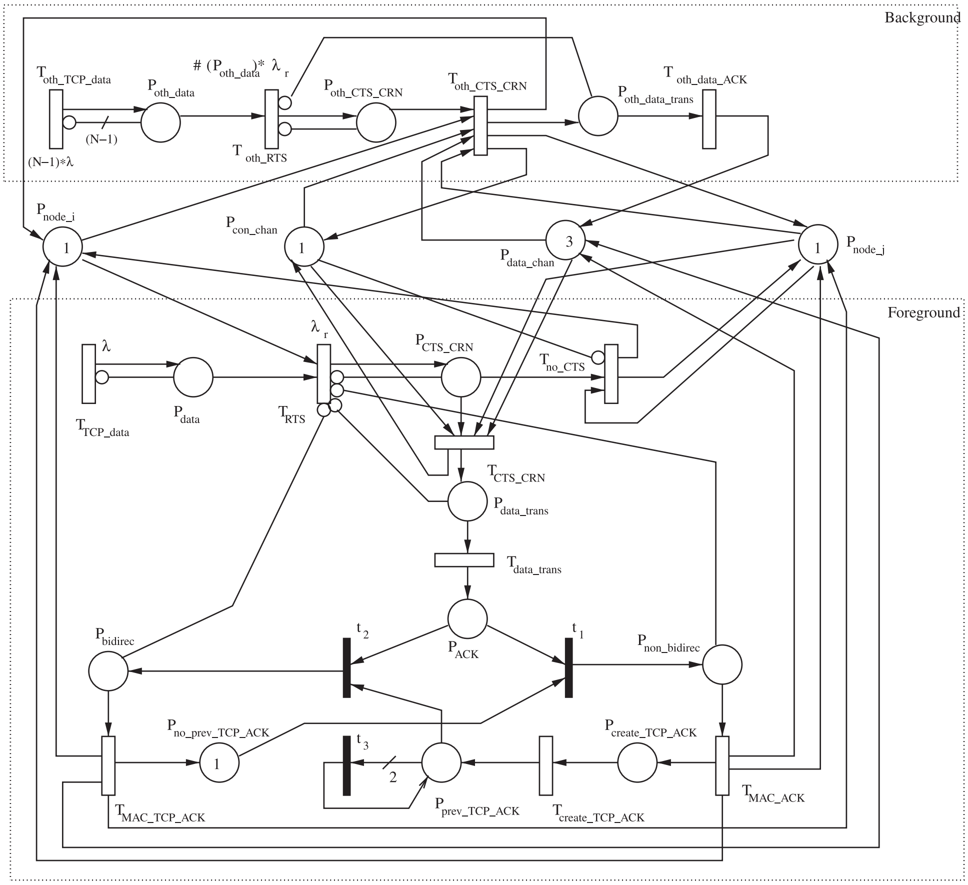

Figure 2. SRN Model for Bi-MCMAC.

of the $N-1$ nodes has a data packet for transmission, it first sends an RTS frame. If both the control channel and a data channel are available, then the receiver node replies with a CTS/CRN frame. Once the exchange of the control frame is over (i.e. the firing of the timed transition $T_{\text{oth\_CTS\_CRN}}$), other nodes can compete for the control channel. After the RTS/CTS/CRN handshake, the sender node sends the data packet and the receiver node replies with an ACK frame. This indicates the completion of the data exchange phase.

3.3. *Channel scheduling strategies*

In the foregoing multi-channel model, all the data channels are homogeneous (i.e. same data transmission rate). As a result, nodes could use any of these data channels for transmission, and a simple counter was sufficient in the SRN model to keep track of the number of available data channels [7]. In this subsection, we extend the SRN model to consider *heterogeneous* data channels (i.e. channels with different data transmission rates). As in the previous model, we use three data channels at the physical layer, in addition to the control channel. We also consider the effect of channel scheduling strategies in this context. In particular, we consider two simple channel scheduling strategies: a *Random* scheduling strategy that is equally likely to choose any of the available data channels, and a *Fastest-Channel-First* strategy that explicitly chooses the fastest channel that is available. Obviously, the Fastest-Channel-First strategy is expected to outperform Random channel selection.

3.4. Model 3: Bi-MCMAC with random channel selection

Figure 3 shows the SRN model for Random channel selection.

The primary difference from the previous Bi-MCMAC is in the channel representation. In particular, each data channel now has its own place. (We have also moved the control channel place from the left to the right. This cosmetic change reduces some of the line crossings.) Furthermore, the control-flow pathways in both the foreground and background traffic have been replicated threefold. This added complexity is required to keep track of which data channels are in use, so that tokens can be returned to the correct place when data exchanges are complete. The rest of the functioning is logically equivalent to the previous model.

3.5. Model 4: Bi-MCMAC with fastest-channel-first selection

Figure 4 shows the SRN model for Fastest-Channel-First selection. Here, we have three heterogeneous data channels with different transmission rates. We assume that these data channels are numbered based on their transmission rates, with the fastest channel numbered as 1, and the slowest channel numbered as 3.

First, we discuss the foreground part of the model. The model functions in the same way as the earlier models. In this model, we have three heterogeneous data channels numbered as 1, 2, and 3, and represented by the places, P_{data_1}, P_{data_2}, P_{data_3}, respectively. Initially, all these three places have one token each, representing that all the three data channels are available. Channel preferences are achieved with inhibitor arcs from the place P_{data_1} and P_{data_2} to the timed transmissions $T_{\mathrm{CTS\_CRN}_2}$, and $T_{\mathrm{CTS\_CRN}_3}$. Data channel 1 is used for packet transmission if it is available (irrespective of the availability of data channels 2 and 3), since the timed transition $T_{\mathrm{CTS\_CRN}_1}$ is enabled. If data channel 1 is busy, but data channel 2 is available, then data channel 2 is used for packet transmission (regardless of the state of channel 3), since the timed transition $T_{\mathrm{CTS\_CRN}_2}$ is enabled. If both data channels 1 and 2 are busy, then only data channel 3 is used for the packet transmission, with the timed transition $T_{\mathrm{CTS\_CRN}_3}$ enabled.

After receiving a data packet, the receiver has to reply with an ACK frame. It sends MAC_ACK and TCP_ACK to the sender. As soon as the sender receives these ACKs, the data channel used for the packet transmission has to be released. For this, a token is to be deposited to the data channel used for the transmission. Putting the token in the right place requires tracking which data channel was used for the packet transmission. The place P_{counter} is used for this purpose. This place can have either 1, 2 or 3 tokens, to represent the data channel used. If data channel 1 is used, then the firing of timed transition $T_{\mathrm{data\_trans}_1}$ deposits one token in the place P_{counter}. Similarly, the firing of the timed transition $T_{\mathrm{data\_trans}_2}$ deposits two tokens in P_{counter} if data channel 2 is used, and the firing of the timed transition $T_{\mathrm{data\_trans}_3}$ deposits three tokens in P_{counter} if data channel 3 used.

The firing of the immediate transitions t_4 deposits a token to data channel 1. Similarly, the firing of t_5 and t_6 deposits a token to data channels 2 and 3, respectively. The transition to be fired (t_4, t_5, or t_6) depends on whether P_{counter} has one, two or three tokens, respectively. Only one of these

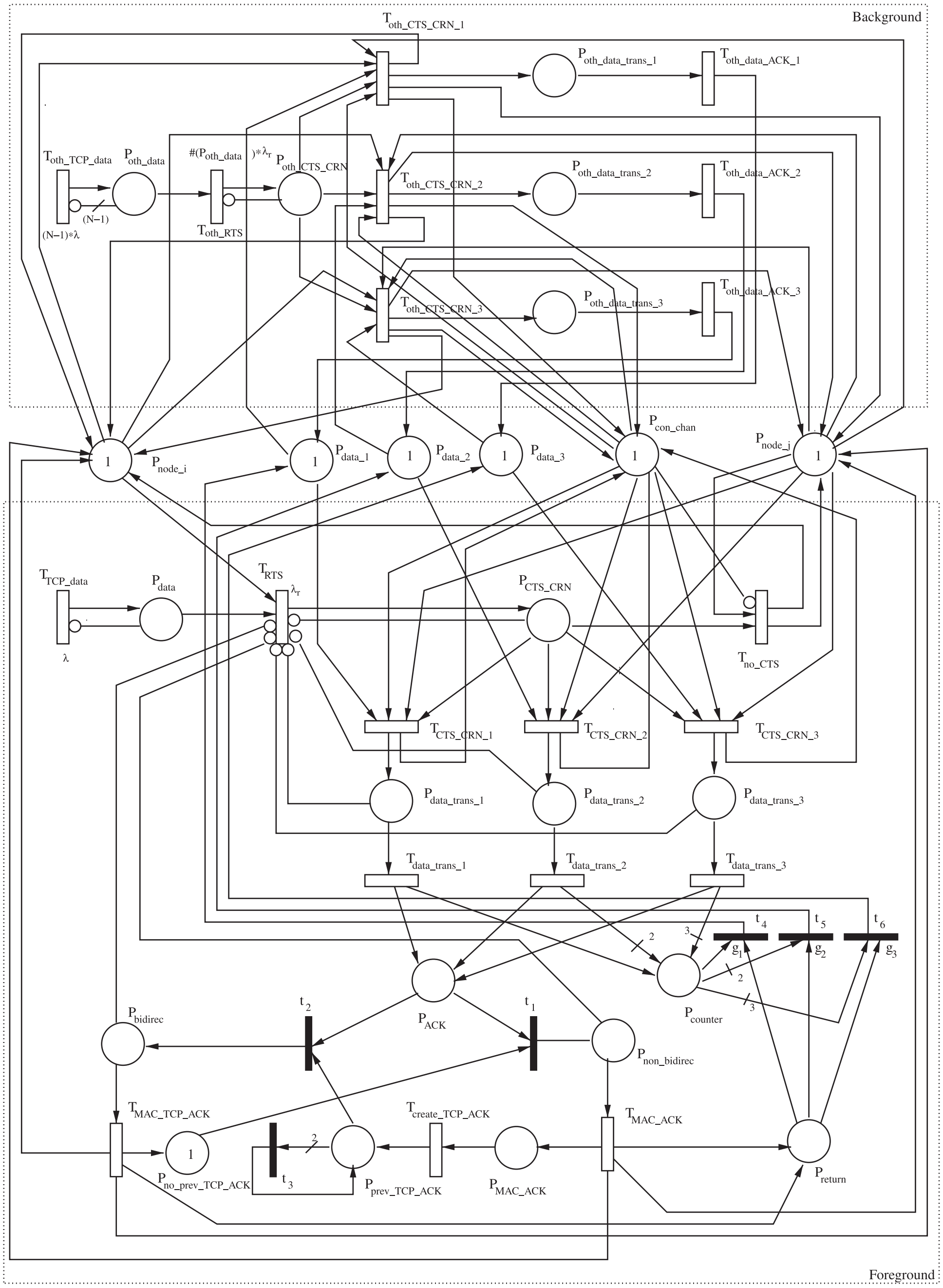

Figure 3. SRN model for heterogeneous Bi-MCMAC with random channel selection.

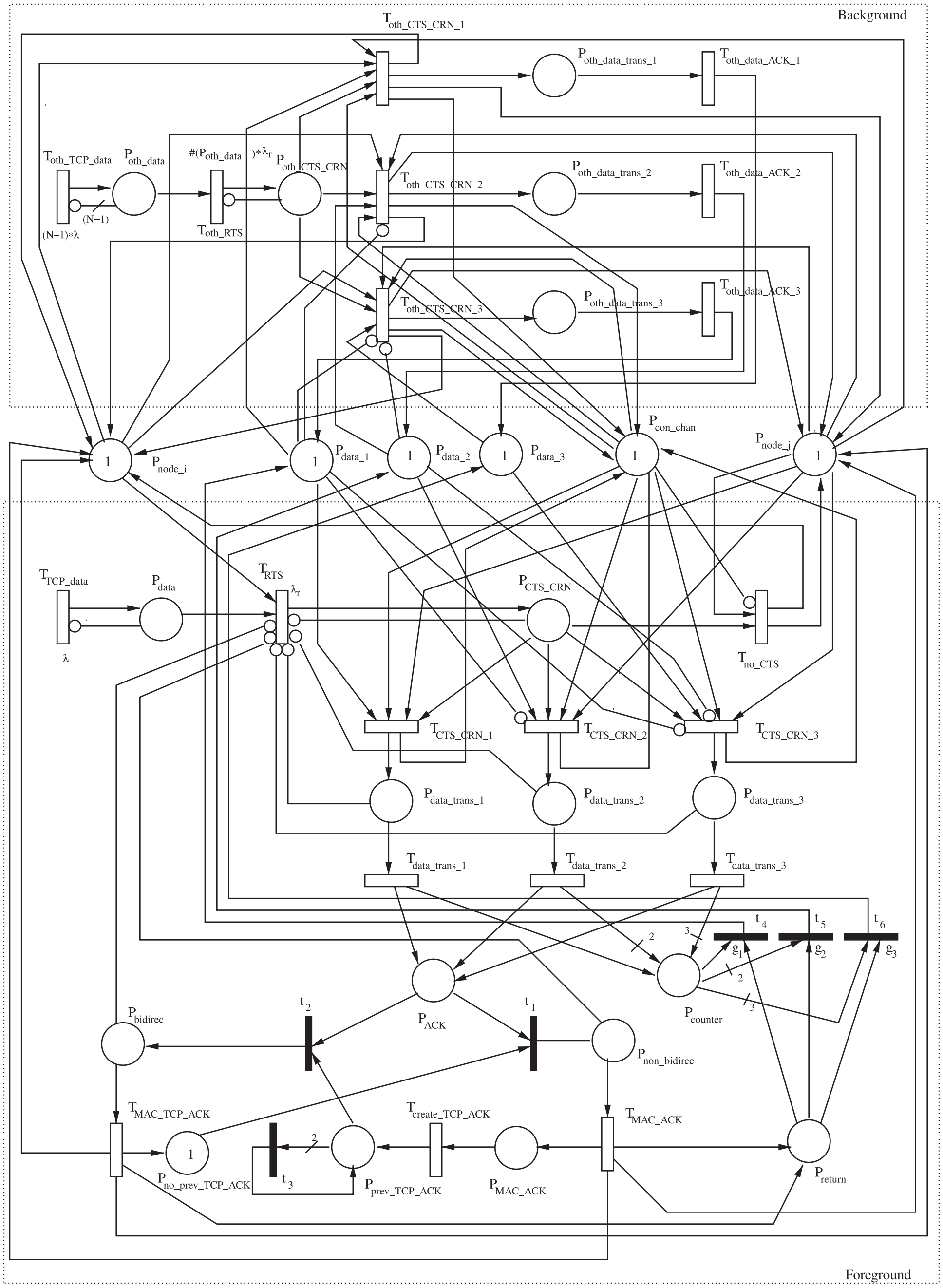

Figure 4. SRN model for heterogeneous Bi-MCMAC with fastest channel selection.

transitions is fired at a time and this is ensured by the guard functions g_1, g_2, and g_3, which are defined as follows:

$$g_1 : \sharp P_{\text{counter}} = 1 \quad \text{and} \quad \sharp P_{\text{return}} = 1$$

$$g_2 : \sharp P_{\text{counter}} = 2 \quad \text{and} \quad \sharp P_{\text{return}} = 1$$

$$g_3 : \sharp P_{\text{counter}} = 3 \quad \text{and} \quad \sharp P_{\text{return}} = 1$$

The rest of the model functions as described previously.

Next, we discuss the background traffic part of the model. This also functions in the same way as the previous models. The only difference is the channel preference. Data channel 1 is used for packet transmission if it is available, irrespective of the availability of data channels 2 and 3. If data channel 1 is busy, but data channel 2 is available, then data channel 2 is used even if data channel 3 is also available at the same time. If both data channels 1 and 2 are busy, then only data channel 3 is used. The inhibitor arc from the place P_{data_1} to the timed transition $T_{\text{oth\_CTS\_CRN}_2}$ ensures that if data channel 1 is busy, then only data channel 2 is used. Similarly, the inhibitor arcs from the places P_{data_1} and P_{data_2} to the timed transition $T_{\text{oth\_CTS\_CRN}_3}$ ensure that data channel 3 is used only when both data channels 1 and 2 are busy. The rest of the functioning is the same as in the previous model.

4. PERFORMANCE METRICS

In this section, we discuss three performance measures, namely, *average system throughput*, *mean delay* from node i to node j, and the server utilization.

(1) *Average system throughput*: We get the average system throughput at the SRN level as follows: The software package SHARPE enables to obtain the average throughput of a transition, which is defined as the average rate at which tokens are deposited by the transition T in its output places. If $\hat{Y}(t)$ is the average number of tokens deposited by the transition T in all its output places up to time t, then the throughput η_T of the transition T is defined as:

$$\eta_T = \lim_{t \to \infty} \frac{\hat{Y}(t)}{t}$$

The average system throughput, η is given by

$$\eta = \eta_{T_{\text{data\_trans}}} + \eta_{T_{\text{oth\_data\_ACK}}}$$

where $\eta_{T_{\text{data\_trans}}}$ gives the throughput of node i and $\eta_{T_{\text{oth\_data\_ACK}}}$ gives the throughput of other nodes.

(2) *Mean delay*: The delay of a packet is defined as the time spent by a packet in the system until it is successfully transmitted (i.e. the packet is received correctly by the destination station, and the corresponding ACK is received correctly by the source station). The SRN formulation provides the mean delay suffered by the packet at the HoL at every station. To compute the mean delay of the subsequent packets at a station, we model each station as an M/G/1 queue, with the mean service time to be the mean delay suffered by the HoL packet. We get the mean delay at the SRN level as follows: The mean delay of the HoL packet of each station, $\bar{D}_{\text{HoL}}$, is the sum of the mean packet holding time (μ) and the sum of the mean

delays undergone by the HoL packet at every stage of the DCF operation. This is obtained by measuring the mean delay of the HoL packet at places P_{data} and P_{CTS} for model 1, and at places P_{data} and $P_{\text{CTS\_CRN}}$ for models 2, 3, and 4 in the SRN. By associating unit reward with all the markings, $\bar{D}_{\text{HoL}}$ can then be obtained by using Little's law as:

$$\bar{D}_{\text{HoL}} = \frac{\sharp(P_{\text{data}})}{\eta_{T_{\text{TCP\_data}}}} + \frac{\sharp(P_{\text{CTS}})}{\eta_{T_{\text{RTS}}}} + \frac{1}{\mu} \quad \text{for model 1}$$

where $\mu =$ Average of firing times of $(T_{\text{CTS}} + T_{\text{data\_trans}} + T_{\text{RTS\_CTS}} + T_{\text{MAC\_ACK}})$ and $(T_{\text{CTS}} + T_{\text{data\_trans}} + T_{\text{RTS\_CTS}} + T_{\text{MAC\_TCP\_ACK}})$

$$\bar{D}_{\text{HoL}} = \frac{\sharp(P_{\text{data}})}{\eta_{T_{\text{TCP\_data}}}} + \frac{\sharp(P_{\text{CTS\_CRN}})}{\eta_{T_{\text{RTS}}}} + \frac{1}{\mu} \quad \text{for models 2, 3 and 4}$$

where $\mu =$ Average of firing times of $(T_{\text{CTS\_CRN}} + T_{\text{data\_trans}} + T_{\text{MAC\_ACK}})$ and $(T_{\text{CTS}} + T_{\text{data\_trans}} + T_{\text{MAC\_TCP\_ACK}})$
The rest of the buffer at the reference station is modeled as an M/G/1 queue with mean service time to be $\bar{D}_{\text{HoL}}$. The mean packet delay, $\bar{D}$ can then be obtained by applying the *Pollackzek–Khinchine* mean value formula [22] as

$$\bar{D} = \bar{D}_{\text{HoL}} \left[1 + \frac{\rho}{2(1-\rho)} (1 + C_D^2) \right]$$

where $\rho = \lambda \bar{D}_{\text{HoL}}$, and if the delay of the HoL packet is represented by the random variable D, then

$$C_D^2 = \frac{E(D^2)}{\bar{D}_{\text{HoL}}^2}$$

where

$$E(D^2) = 2 \left(\frac{\sharp(P_{\text{CTS}})}{\eta_{T_{\text{RTS}}}} \right)^2 \quad \text{for model 1}$$

$$E(D^2) = 2 \left(\frac{\sharp(P_{\text{CTS\_CRN}})}{\eta_{T_{\text{RTS}}}} \right)^2 \quad \text{for models 2, 3, and 4}$$

The mean delay $\bar{D}$ obtained above is the mean delay suffered by any packet in the system because all the stations are independent and behave identically.

(3) *Utilization*: It is defined as the fraction of time, on average, that the server is busy. We get the steady-state utilization at the SRN level as follows: Consider the place $P_{\text{data\_chan}}$ (equivalently, the places P_{data_i}) in the SRN models. A token in this place indicates the availability of the data channel for packet transmission. If this place is empty, it implies that the data channel is busy in transmitting the packet. Therefore, the steady-state server utilization can be obtained as

Server utilization = Steady-state probability that the place $P_{\text{data\_chan}}$ is empty

5. PRACTICAL INSIGHTS

In this section, we give numerical illustration to demonstrate how the proposed analytical models are useful in studying the performance metrics for IEEE 802.11-based WLAN. For the purpose of numerical illustration, we set the parameter values of several components of the analytical model as follows: there are $N = 10$ stations in the system and all are statistically independent and behave identically. The arrival rate λ is varied from 10 to 100 packets per second. For transition rates of various timed transitions, refer to Table II. We have made comparisons between all the four models in terms of the average system throughput, mean delay, and server utilization. We present these measures as a function of virtual traffic load into the system in Erlangs, V, which is defined as $V \triangleq N\lambda L/B$, where L is the mean packet size and B is the channel bit rate. We also present the average system throughput and mean delay as a function of number of stations, N.

5.1. Numerical results

Figure 5(a) presents the variation of average system throughput with respect to the increasing packet size. This also serves as a validation for our model. Model 4 shows the best performance in terms of the average system throughput, whereas model 1 presents the worst. The average system throughput for Models 3 and 2 is in between. This is as expected. In addition, the average system throughput goes up as the packet size increases.

Figure 5(b) shows the variation of the average system throughput versus virtual load. The figure shows that Models 2, 3, and 4 perform better than Model 1, i.e. full Bi-MCMAC model with multiple channels and bidirectional channel reservations gives better results than the single-channel IEEE 802.11 MAC model. This improvement mainly comes from multiple channels, as data packets can be simultaneously transferred on different data channels. Also, Model 4 with the fastest channel scheduling achieves the best throughput performance, as expected. This comes from two aspects. First, it is due to multiple channels, which effectively reduce packet collisions. Second, always choosing the fastest channel available improves the overall system throughput. Figure 5(c) presents the variation of mean delay versus virtual load. It is observed that mean delay increases with increasing load. Also, delay in Models 2, 3, and 4 is less than that in Model 1 because of more data channels and the bidirectional concept. Overall, Model 4 has the lowest delay

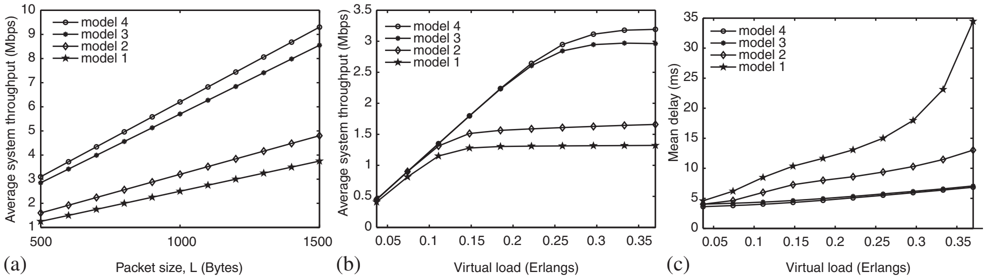

Figure 5. Average system throughput and mean delay versus packet size and virtual load ($N = 10$, $L = 512$ Bytes, $B = 11$ Mbps).

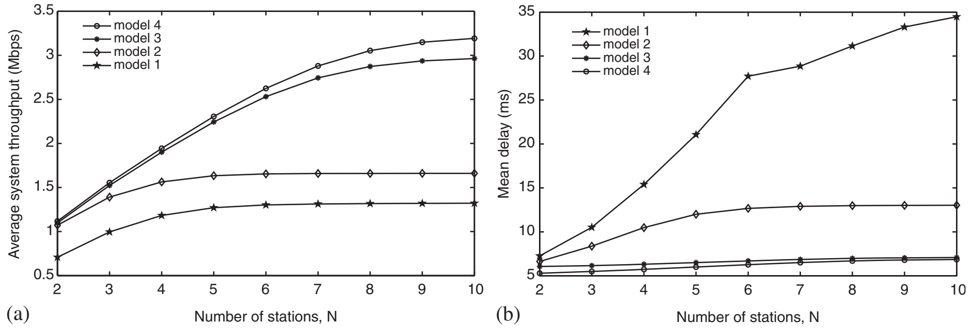

Figure 6. Average system throughput and mean delay versus number of stations ($\lambda = 100$ per sec, $B = 11\,\text{Mbps}$, $L = 512$ Bytes).

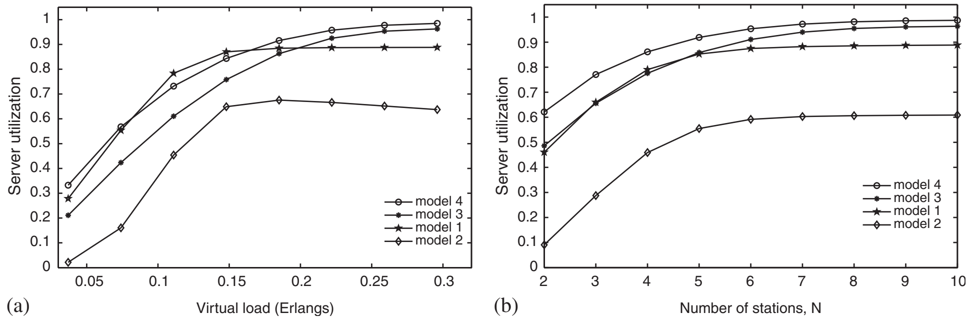

Figure 7. Server utilization versus virtual load and number of stations.

among all the four models. This clearly shows the advantage of the fastest channel scheduling policy as well as the multiple channel concept.

Figure 6 shows the variation of average system throughput and mean delay with respect to the increase in the number of stations. Figure 6(a) shows the variation of average system throughput versus number of stations. It shows that the throughput initially increases on increasing the number of stations, but it reaches a saturation level at some point. Moreover, the graph also shows that Model 4 outperforms the other models. Figure 6(b) presents the variation of mean delay versus number of stations. It is observed that initially mean delay increases with the increase in number of stations, and then reaches a saturation level. In addition, the graph also shows that Model 4 gives the lower delay than the other models. Hence, this graph shows that there is no performance advantage in terms of the average system throughput and mean delay after the number of stations crosses a certain value. Note that the results of Model 3 and Model 4 are quite comparable because of the fact that the two models are structurally similar with the only difference in the channel selection strategy.

Figure 7 shows the variation of server utilization with virtual load and number of stations. Figure 7(a) shows that as virtual load increases, the steady-state utilization of server also increases. Similarly, Figure 7(b) shows that the server utilization increases with the increase in the number

DOI: 10.1002/dac

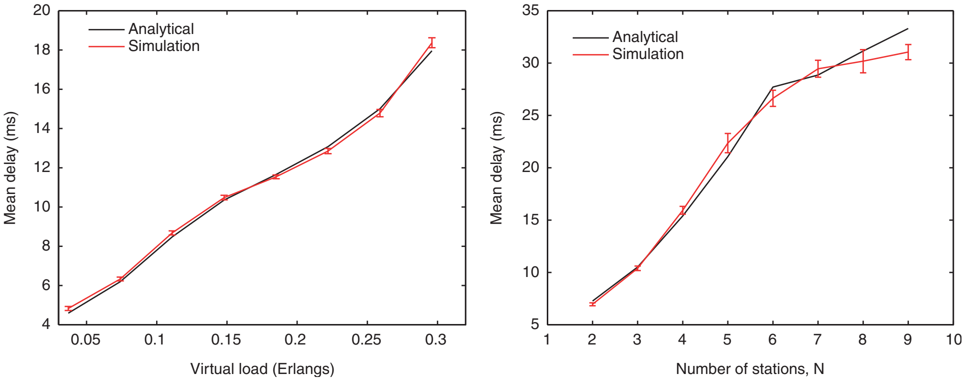

Figure 8. Comparison of analytical versus simulation results for Model 1 ($N=10$, $L=512$ Bytes, $B=11$ Mbps).

of mobile stations. Moreover, it is observed from both the graphs that server utilization of Model 3 and 4 is more than that of Model 1. Again the reason lies in the use of multiple channels with bidirectional concept.

In addition, it can be seen from the Figures 5 and 6 that the steady-state condition is reached when number of stations reaches 10 with the chosen parameter values. This is reason for fixing $N=10$ for the numerical analysis.

5.2. *Model validation*

We validate the proposed SRN model with simulation. The dynamics of the protocol is simulated using MATLAB program [23]. The system parameters for simulation are set as shown in Table II. The comparison between the analytical results and the simulation results for mean system delay of Model 1 is presented in Figure 8. Owing to limited space, similar simulation results for other models and other measures are not presented here. To check the accuracy of the simulation results, we use t distribution test and obtained the confidence interval for 99% accuracy of the result. In the figures, simulation results are plotted with 99% confidence interval.

As expected, mean delay increases with increase in virtual load. Moreover, it is observed from the graph that the results from the proposed analytical approach and simulation are in good agreement. Hence, this validates the analytical model.

6. CONCLUSION AND FUTURE WORK

This paper presents SRN-based analytical model for the classical IEEE 802.11 MAC and IEEE 802.11 Bi-MCMAC protocols for illustrating the performance advantages of the later. The protocols are modeled using M/G/1 queue, and the model analysis is done at the SRN level. By making suitable assumptions for the model (e.g. exponentially distributed times of RTS, CTS, MAC ACK, etc.), these protocols can be modeled by a CTMC. Since the construction of an SRN model

DOI: 10.1002/dac

automatically generates the underlying CTMC, we get the performance measures from the SRN at the modeling level, instead of solving the CTMC. This is the first time the M/G/1 concept is used to model and compare the classical IEEE 802.11 protocol with the Bi-MCMAC protocol, to the best of our knowledge. We present the analytical performance results for throughput, delay, and utilization. The numerical results show that the bidirectional and multi-channel concepts of the Bi-MCMAC protocol provide a distinct performance advantage over the classical IEEE 802.11 MAC. We also present two different channel scheduling policies, *random channel selection* and *fastest channel first selection*, and numerical results show that it further enhances the throughput performance of IEEE 802.11 based *ad hoc* networks. We validated our analysis by comparison with simulations.

The proposed analytical model can be reproduced with minor modifications for computing other measures such as fairness and loss probability. This updated model is solvable by using the basic knowledge of probability. Therefore, the proposed analytical model is relevant for network designers and researchers in the area of wireless networks.

ACKNOWLEDGEMENTS

The excellent comments of the anonymous reviewers are greatly acknowledged and have helped a lot in improving the quality of the paper. This research work is supported by the Department of Science and Technology, India. One of the authors (V.G.) thanks CSIR, India for providing her financial support through Senior Research Fellowship. Financial support for this work was also provided by iCORE (Informatics Circle of Research Excellence) in Alberta, Canada and also by Publishing Arts Research Council (98-1846389).

REFERENCES

1. IEEE Standard for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. *P.802.11*, November 1997.
2. Bianchi G. Performance analysis of the IEEE 802.11 distributed coordination function. *IEEE Journal on Selected Areas in Communications* 2000; **18**(3):535–547.
3. Chen J, Sheu S. Distributed multi-channel MAC protocol for IEEE 802.11 *ad hoc* wireless LANs. *Computer Communications* 2005; **28**(9):1000–1013.
4. He J, Pung H. Performance modeling and evaluation of IEEE 802.11 distributed coordination function in multihop wireless networks. *Computer Communications* 2006; **29**(9):1300–1308.
5. Jayaparvathy R, Anand S, Dharmaraja S, Srikanth S. Performance analysis of IEEE 802.11 DCF with stochastic reward nets. *International Journal of Communication Systems* 2007; 273–296.
6. Adya A, Bahl P, Padhye J, Wolman A, Zhou L. A multi-radio unification protocol for IEEE 802.11 wireless networks. *Proceedings of the IEEE International Conference on Broadband Networks* (*Broadnets*), San Jose, CA, U.S.A., 2004; 344–354.
7. Kuang T, Wu Q, Williamson C. MRMC: a multi-rate multi-channel MAC protocol for multi-radio wireless LANs. *Proceedings of the Workshop on Wireless Networks and Communication Systems* (*WiNCS*), Philadelphia, PA, U.S.A., July 2005; 263–272.
8. Raniwala A, Chiueh T. Architecture and algorithms for an IEEE 802.11-based multi-channel wireless mesh networks. *Proceedings of the IEEE INFOCOM*, Miami, FL, U.S.A., 2005; 2223–2234.
9. So J, Vaidya N. Multi-channel MAC for *ad hoc* networks: handling multi-channel hidden terminals using a single transceiver. *Proceedings of the ACM MobiHoc*, Tokyo, Japan, 2004; 222–233.
10. Ramachandran K, Belding E, Almeroth K, Buddhikot M. Interference-aware channel assignment in multi-radio wireless mesh. *Proceedings of the IEEE INFOCOM*, Barcelona, Spain, 2006; 1–12.
11. Hsieh HY, Sivakumar R. IEEE 802.11 over multi-hop wireless networks: problems and new perspectives. *Proceedings of the Vehicular Technology Conference*, Vancouver, Canada, 2002; 748–752.

12. Manshaei MH, Turletti T. Simulation-based performance analysis of 802.11a wireless LAN. *Proceedings of the International Symposium on Telecommunications*, Isfahan, Iran, 2003.
13. Wang G, Turgut D, Bolini L, Ju Y, Marinescu DC. A simulation study of a MAC layer protocol for wireless networks with asymmetric links. *Proceedings of the International Conference on Wireless Communications and Mobile Computing*, Vancouver, Canada, 2006; 929–936.
14. Cao Z, Liu RP, Yang X, Xiao Y. Modeling IEEE 802.11 DCF system dynamics. *Proceedings of IEEE WCNC*, Sydney, Australia, 2010.
15. Kuang T, Williamson C. A bidirectional multichannel MAC protocol for improving tcp performance on multihop wireless *ad hoc* networks. *Proceedings of ACM/IEEE MSWiM*, Venice, Italy, October 2004; 301–310.
16. Marson MA. *Stochastic Petri Nets*: *An Elementary Introduction*. Lecture Notes in Computer Science. Springer: Berlin, 1989; 1–29.
17. Trivedi KS. *Probability and Statistics with Reliability*, *Queueing*, *and Computer Science Applications* (2nd edn). Wiley: New York, 2001.
18. Sahner RA, Trivedi KS, Puliafito A. *Performance and Reliability Analysis of Computer Systems*: *An Example Based Approach Using the SHARPE Software Package*. Kluwer Academic Publishers: MA, U.S.A., 1996.
19. German R, Kelling C, Zimmerman A, Homel G. TimeNET: a toolkit for evaluating non-Markovian stochastic Petri nets. *Performance Evaluation* 1995; **24**:69–87.
20. German R. Markov regenerative stochastic petri nets with general execution policies: supplementary variable analysis and a prototype tool. *Performance Evaluation* 2005; **39**:165–188.
21. Wormsbecker I, Williamson C. On channel selection strategies for multi-channel MAC protocols in wireless ad hoc networks. *Proceedings of the IEEE International Conference on Wireless and Mobile Computing*, *Networking and Communications*, Montreal, Canada, 2006; 212–220.
22. Kleinrock L. *Queuing Systems*: *Volume I*, *Theory*. Kluwer Academic Press: Dordrecht, 1995.
23. Yuhan Moon, Syrotiuk VR. A cooperative CDMA-based multi-channel MAC protocol for mobile *ad hoc* networks. *Computer Communications* 2009; **32**(9):1810–1819.

AUTHORS' BIOGRAPHIES

Vandana Gupta is currently doing PhD in Mathematics from the Indian Institute of Technology Delhi. She is a post graduate in Operations Research from University of Delhi, India. Her current research interests include queuing theory, Markov modeling, and performance issues of wireless networks and dependability analysis of communication systems.

Mingwei Gong is an Assistant Professor in the Department of Computer Science and Information Systems at the Mount Royal University, Calgary, Canada. He received his BE degree in computer engineering from Tianjin University, Tianjin, China, in 2001, and the MSc and PhD degrees in Computer Science from the University of Calgary, Calgary, Canada, in 2003 and 2009, respectively. His research interests are in computer networks and network security.

DOI: 10.1002/dac

S. Dharmaraja is currently an Associate Professor in the Department of Mathematics, Indian Institute of Technology Delhi, New Delhi, India. He received his PhD in Mathematics from the Indian Institute of Technology Madras, India in 1999. Before joining this institute, he was a post-doctoral fellow at the department of electrical and computer engineering, Duke University, USA and then a research associate at the TRLabs, Winnipeg, Canada. His current research interests include queuing theory, Markov modeling, and performance issues of wireless networks and dependability analysis of communication systems.

Carey Williamson is a Professor in the Department of Computer Science at the University of Calgary, where he holds an iCORE Chair in Broadband Wireless Networks, Protocols, Applications, and Performance. He has a BSc (Honours) in Computer Science from the University of Saskatchewan, and a PhD in Computer Science from Stanford University. His research Interests include Internet protocols, wireless networks, network traffic measurement, network simulation, and Web performance.