A Petri-Net-Based Automated Distributed Dynamic Channel Assignment for Cellular Network

Shin-Yeu Lin and Ting-Yu Chan

***Abstract*—In this paper, we propose a Petri-net (PN)-based automated distributed dynamic channel assignment (DDCA) method for the cellular network. We view DDCA as a discrete event system (DES) and propose a PN to model the automated DDCA. The spontaneous handshaking mechanism in the proposed PN can resolve the hard constraint of the cellular network so that no two cells within the channel reuse distance can use the same channel. Instead of the commonly adopted packing and resonance conditions for channel assignment and reassignment, we propose available channel-based channel-selection and channel-reassignment schemes. We rigorously prove the adequacy of the proposed PN and the satisfaction of hard constraint. We also test the proposed PN-based automated DDCA method using numerous test cases. The test results show that our method outperforms the comparing methods in terms of blocking probability.**

***Index Terms*—Blocking probability, cellular network, dynamic channel assignment (DCA), handshaking, Petri net (PN).**

I. Introduction

MOBILE communication systems are among the fastest growing areas in telecommunications. To effectively utilize the limited resources of the radio spectrum, the geographical coverage area is divided into hexagonal cells to form a cellular network, and the radio spectrum can be reused in noninterfering cells. Due to the considerable increase in the number of mobile users, maximization of spectrum utilization has been one of the most important research issues in mobile communications, and channel assignment is a key to achieving this goal.

In general, the channel-assignment strategies, which are a popular subject in this journal, can be classified into four categories: 1) *fixed channel-assignment* (FCA) strategy [1], [2]; 2) *dynamic channel-assignment* (DCA) strategy [3]–[6]; 3) *borrowing channel-assignment* (BCA) strategy [7], [8]; and 4) *hybrid channel-assignment* (HCA) strategy [9], [10]. Among these four strategies, FCA is the easiest to implement; however, it is least efficient in most of the cases. Having been demonstrated to be superior to some BCA [3], DCA is most flexible and is an active channel-assignment strategy in some other types of networks, such as wireless mesh networks [11]–[13] and multihop cellular networks [14], [15]. Furthermore, the techniques developed for DCA can also be used in the HCA strategy [10]. Thus, we will focus on DCA in this paper.

In a conventional approach, the DCA is centrally implemented in the *mobile switching center* (MSC) [3]–[6], in the pool of a cluster [16], or in a cell with a central controller [10], which collects the information of all cells in the network. However, determining the best available channel for a new call arising in a cell in a centralized manner is a *combinatorial optimization problem*, which is computationally intractable, and several heuristic methods were proposed [4], [10], [17]. The disadvantages of such centralized schemes are as follows: 1) The MSC may be overloaded and cause possible failure [18], and 2) it is time consuming to determine the best available channel, even if heuristic methods are used [4], [10], [18], and it causes a new call blocked for just one unsuccessful assignment, as indicated in [10]. To overcome these disadvantages, distributed DCA (DDCA) was proposed to determine the channel acquisition and release in the base station (BS) of a cell [20]–[22]. However, the DDCA faces the following challenges: 1) How to cope with the *hard constraint* that any two cells within interfering distance cannot use the same channel under the situation that each BS will determine its own available channel for the new call; 2) how to select an adequate channel for the new call based on limited information from the surrounding cells only; and 3) how to perform the channel reassignment.

The DDCA methods proposed in [20]–[22] were only dealing with the issue of challenge 1. Because of the shortage of an explicit and precise model to describe the system, they resolve the conflict between neighboring cells' intention in acquiring the same channel using a cumbersome procedure by sending message to neighboring cells to *request for channel utilization permission*. A similar situation appears in [18] and other distributed channel assignment methods [23]. By viewing the DDCA as a discrete event system (DES), one of the contributions of this paper is proposing a Petri net (PN) to model the *automated* DDCA for a cellular network. A *spontaneous handshaking* mechanism will be imbedded in the proposed PN to clear out the conflict in acquiring the same channel between neighboring cells and resolve challenge 1. Consequently, there is no need to request channel utilization permission from neighboring cells as the aforementioned DDCA methods did. Therefore, the proposed PN for automated DDCA will completely be different from the PN for centralized DCA proposed in [16]. To resolve challenges 2 and 3, differing from the conventional criteria employed in centralized DCA, we will propose novel channel

Manuscript received November 3, 2008; revised January 19, 2009 and March 26, 2009. First published May 15, 2009; current version published October 2, 2009. This work was supported in part by the National Science Council in Taiwan under Grant NSC97-2221-E-009-088. The review of this paper was coordinated by Dr. C. Lin.

S.-Y. Lin is with the Department of Electrical Engineering, Chang Gung University, Tao-Yuan 333, Taiwan (e-mail: sylin@cc.nctu.edu.tw).

T.-Y. Chan is with the Department of Electrical and Control Engineering, National Chiao Tung University, Hsinchu 300, Taiwan, and also with Elan Microelectronics Corporation, Hsinchu 308, Taiwan (e-mail: tingyu.chan@emc.com.tw).

Digital Object Identifier 10.1109/TVT.2009.2022968

selection and reassignment criteria that are most suitable for the proposed automated DDCA, which is another contribution of this paper.

We organize this paper in the following manner: In Section II, we model the automated DDCA using PN and present our channel-selection and channel-reassignment schemes. In Section III, we present the properties and prove the adequacy of the proposed PN and show the satisfaction of hard constraint. In Section IV, we test our PN-based automated DDCA method on a typical cellular network with nonuniform traffic patterns and compare the test results with those obtained by the existing methods. Finally, we draw a conclusion in Section V.

II. Petri Net Model for Automated Distributed Dynamic Channel Assignment

A. Motivation

The methods to model a DES are not limited to PN. There are other methods, such as finite-state automata, Markov chain, and queuing-type models. The cellular network consists of a group of cells, for example, there are 49 cells in a 7×7 parallelogram-shaped cellular network. Supposing that X_1 represents the state space of a channel in cell 1, then the state space of a channel in a cellular network consisting of N cells can possibly be as large as $X_1 \times X_2 \times \cdots \times X_N$ if modeled by finite-state automata. This means that combining multiple cells rapidly increases the complexity of the finite-state automata model. A similar situation occurs to the Markov chain model. However, the PN model possesses an ability to decompose or modularize a complex interacting system, such that the interactions between neighboring cells can be resolved by adding a few places and transitions, which will be indicated in Remark 2. Then, from a PN model, one can conveniently see the dynamics of individual cells, discern the level of their interaction, and ultimately decompose a cellular network into logical distinct cell modules, as will be seen in this section. Such a decomposition capability is hardly achieved by a queuing-type model. Thus, PN is most suitable for modeling an automated DDCA.

B. Basic Terminologies of PN

The PN model consists of two parts: 1) the PN graph and 2) the PN dynamics. The PN graph considered here is a bipartite graph (P, T, A), as shown in Fig. 1(a), where P denotes the finite set of places marked by circles, T denotes the finite set of *timed* and *untimed* transitions marked by bars, and A denotes the set of directed arcs from places to transitions and from transitions to places in the graph, i.e.,

$$P = \{p_1, p_2, p_3, p_4, p_5\}$$
$$T = \{t_1, t_2\}$$
$$A = \big\{(p_1, t_1), (p_2, t_1), (t_1, p_3), (t_1, p_4) \quad (t_1, p_5), (p_4, t_2), (p_5, t_2), (t_2, p_2)\big\}.$$

The weight of each directed arc considered in this paper is 1.

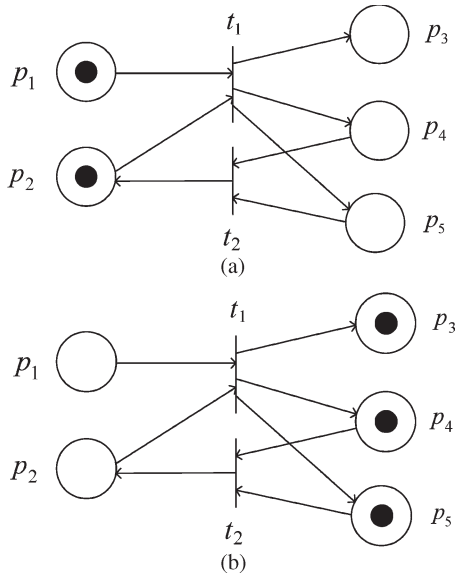

Fig. 1. (a) Example PN graph. (b) Resulting state of (a) after t_1 fires.

In PNs, events driving a DES are associated with transitions. A transition is enabled to occur (or fire), provided that the conditions related to the places input to this transition are satisfied. An untimed transition immediately fires once enabled, whereas a timed transition will fire after a time duration once enabled, unless specifically specified. The mechanism indicating whether a condition related to a place is met or not is provided by the presence of a token in that place. For example, in Fig. 1(a), both p_1 and p_2, which are the conditions input to t_1, are satisfied; thus, t_1 is enabled and can fire. The state of a PN graph is defined as its marking, which is represented by the number of tokens in all places. For example, we let $[p_1, p_2, p_3, p_4, p_5]$ denote the vector of all places, and the state of the PN graph in Fig. 1(a) is [11000], which represents that there is one token in p_1, one token in p_2, and no token in p_3, p_4, and p_5. The PN dynamics is to describe the transition of the state in a PN, and the state transition mechanism is provided by moving tokens through the net when any transition fires, hence changing the state of the PN. In other words, the state of the PN graph changes when an event occurs, i.e., a transition fires. The tokens in the places are moving through the net in the following manner. If a transition fires, then the number of tokens in the places input to the fired transitions is decreased by one and increased by one in the places output from the fired transition. For example, the marking in Fig. 1(b) is the resulting state of the marking in Fig. 1(a) when t_1 fires. The foregoing description about the PN is a general terminology. Some modifications are needed for the PN that models the automated DDCA, as described in the following two sections.

C. Definitions and Assumptions

Preliminaries: We let $C = \{c_1, \ldots, c_{N_c}\}$ denote the set of all channels in the cellular network, where c_i is the *identifier* of the ith channel, such that a smaller index represents a lower frequency, and N_c denotes the total number of channels in the system. For the sake of presentation, we assume that the

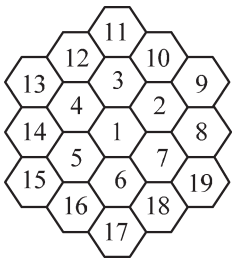

Fig. 2. Portion of the cellular network.

considered cellular network is a *three-cell cluster* system, such that the channel reuse distance is two cells [25]. Thus, the *hard constraint* for the cellular network considered here is that no cell can use the same channel used by its neighboring cells. This implies that the channel used in the considered cell can be used in any other cell, except for the neighboring cells. For example, in the portion of a cellular network shown in Fig. 2, cells 2, 3, 4, 5, 6, and 7 cannot use the same channel used in cell 1. However, the proposed approach can be extended to a N-cell cluster system.

We define the following six *real* channel conditions for each channel: 1) *available for assignment* denoted by AV; 2) *selected for assignment* denoted by S; 3) *in use* denoted by IU; 4) *used by one neighboring cell* denoted by U1N; and 5) and 6) *used by two and three neighboring cells* denoted by U2N and U3N, respectively.

Remark 1: For a three-cell cluster system, each channel in a cell can be used by three neighboring cells at most based on the structure of the cellular network and the hard constraint.

We *assume* that each cell will send the updated real channel condition of each channel to all neighboring cells whenever it changes, and the *message propagation delay* denoted by Δt_d between any two neighboring cells is assumed to be equal. In general, Δt_d is on the order of microseconds.

We define the six cells neighboring the considered cell as the first-tier cells and define a cell that is neither the considered cell nor a first-tier cell but is neighboring to any first-tier cell as a second-tier cell. For example, cells 2, 3, 4, 5, 6, and 7 in Fig. 2 are first-tier cells of cell 1, and cells 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, and 19 are the second-tier cells. Without loss of generality, we index the first-tier cells based on their relative positions, such that cell 2 is denoted as first-tier cell 1, cell 3 is denoted as first-tier cell 2, etc. We define two types of tokens: 1) *marked* and 2) *unmarked*. The token marked with a channel identifier denotes that the token is associated with the marked channel. The unmarked token is associated with a new call.

Places: We define three sets of places ($P_{\rm nc}$, $P_{\rm ch}$, and $P_{\rm auxi}$, $i = 1, \ldots, 6$) for the new calls, real channel conditions, and auxiliary channel conditions that involve the six first-tier cells, respectively, such that $P_{\rm nc} = \{Q, B\}$, $P_{\rm ch} = \{\text{IU, AV, U1N, U2N, U3N, S}\}$, and $P_{\rm auxi} = \{\bar{S}_i, S_i, A_i, E_i, CIU_i, CRL_i,\ i = 1, \ldots, 6\}$. Place Q represents the condition that the new call is in the processing queue; place B represents the condition that the new call is blocked; places AV, IU, U1N, U2N, U3N, and S represent the real channel conditions, which have previously been described; and place S_i represents that the condition of the channel in first-tier cell i is S. We let the complement of S denoted by $\bar{S}$ represent one of the following conditions: AV, U1N, or U2N; then, $\bar{S}_i$ denotes that the condition of the channel in first-tier cell i is $\bar{S}$, and the conditions represented by A_i, E_i, CIU_i, and CRL_i, $i = 1, \ldots, 6$ will be described later in the PN dynamics (see Section II-D). The places in $P_{\rm nc}$ are only related to new calls; thus, only unmarked tokens can appear in them. On the other hand, only marked tokens can appear in the places in $P_{\rm ch}$ and $P_{\rm auxi}$, $i = 1, \ldots, 6$. To distinguish the real and auxiliary channel conditions, we employ dotted circles to represent all the places in $P_{\rm auxi}$.

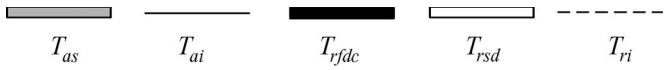

Fig. 3. Graphical representation of the five types of transitions.

Transition Types: We define five types of transitions: 1) intracell stochastic timed denoted by $T_{\rm as}$; 2) intracell instantaneous denoted by $T_{\rm ai}$; 3) intercell fixed timed with double check denoted by $T_{\rm rfdc}$; 4) intercell specially designed timed denoted by $T_{\rm rsd}$; and 5) intercell instantaneous denoted by $T_{\rm ri}$; and they are graphically represented by the gray thick bar, thin bar, black thick bar, blank thick bar, and dashed thin bar, respectively, as shown in Fig. 3. Note that $T_{\rm ai}$- and $T_{\rm ri}$-type transitions represent untimed transitions, whereas $T_{\rm as}$, $T_{\rm rfdc}$, and $T_{\rm rsd}$ represent timed transitions.

Transition Enabling and Firing: Since we have two types of tokens (marked and unmarked) and the marked tokens with different channel identifiers are tokens of different marks, the enabling of a transition employed in our PN to model the automated DDCA will differ from the case described in Section II-B. We say that a transition defined in our PN can be *enabled* only if its input places from $P_{\rm ch}$ or $P_{\rm auxi}$, $i = 1, \ldots, 6$ consist of tokens of the same marks, and the input places from $P_{\rm nc}$ consist of unmarked tokens. The tokens shot from the fired transition will be the same marked tokens if the output places are in $P_{\rm ch}$ or $P_{\rm auxi}$, $i = 1, \ldots, 6$ and will be the unmarked tokens if the output places are in $P_{\rm nc}$. None of the intracell types of transitions have to do with the places associated with the channel conditions of neighboring cells, such as S_i, $\bar{S}_i$, CIU_i, and CRL_i, $i = 1, \ldots, 6$. However, a $T_{\rm as}$-type transition will fire after an *indefinite* time duration once enabled, and a $T_{\rm ai}$-type transition will immediately fire once enabled. On the contrary, all the intercell-type transitions involve the channel conditions of the neighboring cells. Once enabled, a $T_{\rm rfdc}$-type transition will fire after a fixed time duration if the enabling conditions still hold or will be disabled otherwise. The enabling and firing of $T_{\rm rsd}$ is more complicated and will be explained later in the PN dynamics (see Section II-D). A $T_{\rm ri}$-type transition will immediately fire once enabled. It should be noted that the two types of transitions ($T_{\rm rfdc}$ and $T_{\rm rsd}$) defined here differ from the typical transitions described in [24], because their firing mechanisms are specially designed for the channel-selection scheme and the spontaneous handshaking mechanism to resolve the interactions between neighboring cells.

Transitions: To distinguish from the places, we employ a *small letter* to represent a transition. Based on the nature of the events and the previously defined transition types, we put all

the events in our PN into the set of proper transition type, such that $T_{\rm as} = \{$call completes (cc) & channel reassigned $(cra)\}$, $T_{\rm ai} = \{$new call arrives (nca), call blocked (cb), update the number of calls in the cell $(uncc)$, channel reselection $(crs_j,\ j = 1, \ldots, 2^6 - 1)$, assign channel $(ac)\}$, $T_{\rm rfdc} = \{$channel selection $(cs)\}$, $T_{\rm rsd} = \{$channel yet acquired by the ith first-tier cell (cya_i), channel has been selected by the ith first-tier cell $(chs_i),\ i = 1, \ldots, 6\}$, and $T_{\rm ri} = \{$force to change $(frt_{ij},\ j = 1, 2, 3,\ i = 1, \ldots, 6)$, force to recover $(frr_{ij},\ j = 1, 2, 3,\ i = 1, \ldots, 6)\}$. We will describe the purpose of the events that cannot be told directly from their meaning in the following section.

Remark 2: The set of places $P_{\rm auxi} = \{\bar{S}_i, S_i, A_i, E_i, CIU_i, CRL_i,\ i = 1, \ldots, 6\}$, the sets of transitions $T_{\rm rfdc} = \{cs\}$ and $T_{\rm rsd} = \{cya_i, chs_i,\ \ i = 1, \ldots, 6\}$, as well as their firing mechanism, and the set of transition $T_{\rm ri} = \{frt_{ij}, frr_{ij},\ j = 1, 2, 3,\ i = 1, \ldots, 6)\}$ are designed to resolve the interactions between neighboring cells.

D. PN Graph and Its Dynamics for Automated DDCA

The PN graph for modeling the automated DDCA is shown in Fig. 4. Due to the page space limitations, we only show the PN graph for one cell, which is the part enclosed by the dashed line and marked on top as the considered cell. To explain the interaction between neighboring cells, we show minimum representation of the channel conditions in first-tier cell 1, which is also enclosed by the dashed line and is at the right-hand side of the considered cell. For the sake of easier reference, we rewrite the meaning of each transition type and each transition in Fig. 4. In the following, we describe the dynamics of the proposed PN step by step.

New Call Arrives: When a new call arrives, the transition nca will first check whether the number of calls in the cell, including the assigned and the yet assigned calls, exceeds N_c. If it exceeds N_c, then the call will be rejected; otherwise, nca will fire, and an unmarked token will be output to place Q, as can be observed from Fig. 4. In the meantime, we will add one to the number of calls in the cell. Since checking the number of calls in the cell almost consumes no time, we categorize nca in $T_{\rm ai}$, as previously described. The unmarked token shot to place Q will be appended with a *lifetime*, which is equal to the *call-request* response time. A switch denoted by a slant line segment over the arc is used at the output of place Q, such that if the *remaining* lifetime (RL) of the unmarked token is greater than 0, then Q is connected to transition cs and connected to cb otherwise, as indicated in Fig. 4. Note that the RL will decrease as time passes.

Channel Selection: In the case that Q is connected to a $T_{\rm rfdc}$-type transition cs at time, for example, t, suppose that the enabling conditions for cs hold, i.e., if there are marked tokens in place AV, and if the same marked tokens also appear in places $\bar{S}_i$ for all $i = 1, \ldots, 6$, as can be observed from Fig. 4, then the *channel-selection scheme* will be carried out to select a channel to assign for the new call with the *least* RL. The channel-selection scheme will be presented in the following section.

Remark 3: 1) The reason why the channel conditions of the first-tier cells have to be in $\bar{S}_i$, for all $i = 1, \ldots, 6$, can be stated in the following. If the condition of the channel in the first-tier cell i is not in $\bar{S}_i$, it implies that this channel has been selected earlier or in use by first-tier cell i, then the considered cell will not consider this channel as a candidate for assignment. 2) If the condition of the channel in the considered cell is AV, then the condition of this channel in any first-tier cells has no chance to be U3N due to the hard constraint. This is the reason why $\bar{S}$ does not include the condition U3N. 3) Since a new call may not successfully be assigned with the selected channel, the unmarked tokens in place Q may be more than one and with different RLs. To choose the unmarked token with the least RL to assign is a reasonable choice based on the *first-come first-serve* principle. 4) In Fig. 4, we only show the place $\bar{S}_1$, as well as S_1, from first-tier cell 1 in dotted circles and use the dots to represent similar places from the other first-tier cells.

It is possible that the conditions of all the available channels that enable cs change by the end of executing the channel-selection scheme and disable cs. If such a situation occurs, no channel is selected, and the new call will stay in Q waiting for further assignment before RL $= 0$. This is the reason why we categorize cs in $T_{\rm rfdc}$, and the details of such a situation will be presented later in the channel-selection scheme (see Section II-E). However, if a channel is successfully selected, then a marked token corresponding to the selected channel in AV and in $\bar{S}_i$, $i = 1, \ldots, 6$, and the unmarked token with the smallest RL in Q will be shot, and one token with the same mark will be output to place S at time $t + \Delta t_{cs}$, as can be observed from Fig. 4, where Δt_{cs} denotes the time duration for the timed transition cs as the time needed to execute the proposed channel-selection scheme. Δt_{cs} is on the order of 10^{-5} s obtained from our simulations, as presented in Section IV. In the meantime, a token with the same mark will also output to $\bar{S}_i$ for $i = 1, \ldots, 6$ because $\bar{S}_i$ and S_i are designed for recording the channel condition of first-tier cell i, as indicated in the previous section. That means that the marked tokens in $\bar{S}_i$ or S_i will be neither gained nor lost. Therefore, whenever $\bar{S}_i$ or S_i inputs to any transitions, there will be an arc output from that transition to $\bar{S}_i$ or S_i, as observed from Fig. 4. Furthermore, it should be noted that the token in S_i (or $\bar{S}_i$) may arbitrarily be moved to $\bar{S}_i$ (or S_i) whenever the condition of the corresponding channel in first-tier cell i changes.

Spontaneous Handshaking: Place S inputs to both $T_{\rm rsd}$-type transitions cya_i and chs_i, whereas $\bar{S}_i$ and S_i input to cya_i and chs_i, respectively, for $i = 1, \ldots, 6$, as shown in Fig. 4. Apparently, the marked token associated with the selected channel being present at $\bar{S}_i$ (or S_i) implies that the channel has yet to be acquired (or has been selected) by first-tier cell i. As previously indicated, the enabling and firing of the $T_{\rm rsd}$-type transition is complicated and will be explained here. Following from the previously mentioned channel-selection, suppose that the marked token associated with the selected channel is present at place S at time $t + \Delta t_{cs}$. Then, for each i, the timed transitions cya_i and chs_i will start to count the time, and either cya_i or chs_i will fire at $t + \Delta t_{cs} + \Delta t_d$, depending on which is enabled at the time instance right before $t + \Delta t_{cs} + \Delta t_d$, because the corresponding marked token may move from $\bar{S}_i$ to S_i during $(t + \Delta t_{cs}, t + \Delta t_{cs} + \Delta t_d)$. Details of this situation are subsequently described. At time $t + \Delta t_{cs}$, the marked token shown in $\bar{S}_i$ represents the condition of this channel

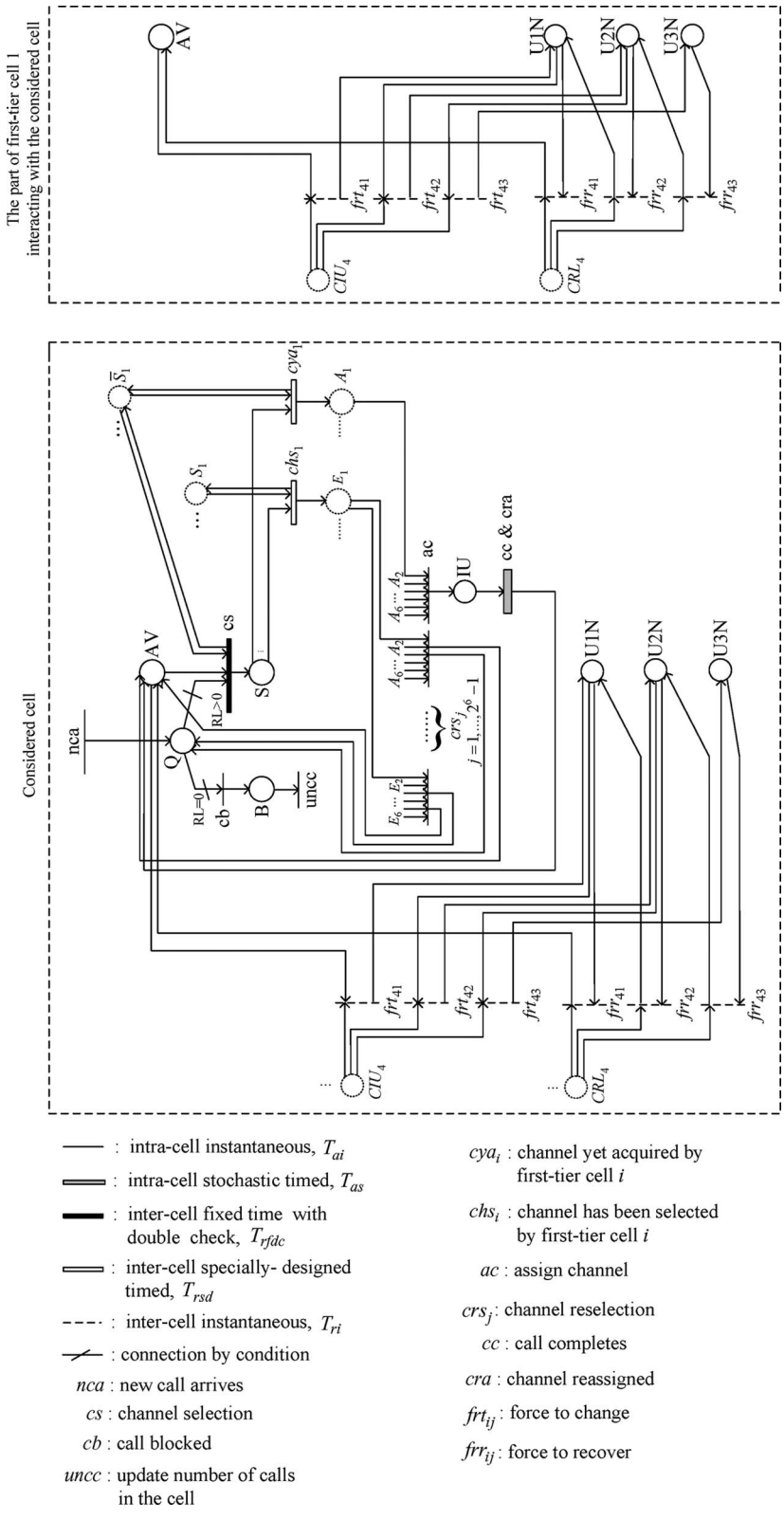

Fig. 4. Proposed PN for modeling the automated DDCA.

in first-tier cell i at time $t + \Delta t_{cs} - \Delta t_d$ due to message propagation delay. However, it is possible that the condition of this channel in first-tier cell i changes from $\bar{S}$ to S during $(t + \Delta t_{cs} - \Delta t_d, t + \Delta t_{cs})$, whose updated information will appear at place S_i during $(t + \Delta t_{cs}, t + \Delta t_{cs} + \Delta t_d)$ due to message propagation delay. This is the reason why we define the enabling time instance of the transitions cya_i and chs_i to be the instance right before $t + \Delta t_{cs} + \Delta t_d$. Therefore, according to Fig. 4, if the marked token stay in place $\bar{S}_i$ throughout the entire duration $(t + \Delta t_{cs}, t + \Delta t_{cs} + \Delta t_d)$, transition cya_i will fire at $t + \Delta t_{cs} + \Delta t_d$ and output the token with the same mark to place A_i, the condition of which represents that

first-tier cell i *allows* the considered cell to use this channel. However, if the considered marked token changes from $\bar{S}_i$ to S_i during $(t+\Delta t_{cs}, t+\Delta t_{cs}+\Delta t_d)$ and stays in S_i at the time instance right before $t+\Delta t_{cs}+\Delta t_d$, transition chs_i will fire at $t+\Delta t_{cs}+\Delta t_d$, and a token with the same mark will be output to place E_i, the condition of which represents that first-tier cell i has selected this channel *earlier* than the considered cell. Based on the assumption that each cell will send the updated condition of each channel to all the neighboring cells whenever it changes, the above handshaking process is *spontaneous*, which differs from the response by request in the previously mentioned requesting channel utilization permission from neighboring cells in some existing DDCA methods.

Remark 4: The time durations for transitions cya_i and chs_i, $i=1,\ldots,6$ are all the same as Δt_d, which counts from the moment that the considered marked token is present at S.

Channel Assignment: All places A_i, $i=1,\ldots,6$ having the same marked tokens implies that this channel has yet to be selected by any first-tier cell. Then, the transition ac is enabled and instantaneously fires; subsequently, a token with the same mark is output to place IU, as indicated in Fig. 4. In the meantime, a message indicating the considered channel is now *in use* will be sent to all the first-tier cells. When this message is received at first-tier cell i, a marked token associated with the channel being in use will appear in the place CIU_i, which is represented by dotted circles in Fig. 4 regarding the part of first-tier cell i interacting with the considered cell. CIU_i will input to the $T_{\rm ri}$-type transition frt_{ij}, $j=1,2,3$. The purpose of frt_{ij} is to force first-tier cell i to change the condition of the channel from AV (available) to U1N (used by one neighboring cell) or from U1N to U2N (used by two neighboring cells) or from U2N to U3N (used by three neighboring cells) to reflect the fact that the channel is now in use in the considered cell. Therefore, there are also input places AV for frt_{i1}, U1N for frt_{i2}, and U2N for frt_{i3} from first-tier cell i, as shown in Fig. 4, and one of frt_{ij}, $j=1,2,3$ will immediately fire once enabled. It should be noted that from the viewpoint of first-tier cell 1, the considered cell is first-tier cell 4, and this is why we put CIU_4 and the corresponding transitions frt_{41}, frt_{42}, frt_{43} in Fig. 4.

Channel Reselection: Note that for each i, the marked token can appear in one and only one of A_i and E_i, because only one of cya_i and chs_i can fire. Therefore, if the marked token is present at least one of the six E_i, $i=1,\ldots,6$, then one and only one of the transitions crs_j, $j=1,\ldots,2^6-1$ is enabled and instantaneously fires, as indicated in Fig. 4. The above situation implies that at least one of the six first-tier cells has selected this channel for assigning their new call earlier than the considered cell, which then has to reselect a channel. Subsequently, a token with the same mark will be output to place AV, and an unmarked token will be output to place Q with updated RL, as can be observed from Fig. 4. We will then check whether the RL of this unmarked token equals 0. If yes, transition cb is enabled as indicated in Fig. 4, and the unmarked token is shot to place B, which indicates the new call is blocked and enable the transition $uncc$ to decrease the number of calls in the cell by one; otherwise, a channel-reselection process is carried out by following previously mentioned procedures.

Remark 5: Differing from the centralized DCA and some DDCA [10], any new call with unsuccessful channel assignment in our PN-based automated DDCA method can be retried as long as its RL >0. Thus, we have resolved the second disadvantage of centralized DCA indicated in Section I.

Call Completes and Channel Reassigned: Once the marked token appears at place IU, the transition cc of type $T_{\rm as}$ is enabled and will be fired once the call completes. The time duration of cc is stochastic, which is determined based on an exponential probability distribution. When cc really occurs, the transition cra will simultaneously occur to carry out the *channel reassignment scheme*, which will be presented later, to select the released channel. Then, a marked token associated with the released channel will be output to place AV, as indicated in Fig. 4, and a message indicating the released channel is sent to all the first-tier cells. When this message is received at first-tier cell i, a marked token associated with the released channel will appear in places CRL_i, $i=1,\ldots,6$, represented by dotted circles in Fig. 4 inside the part of first-tier cell i interacting with the considered cell. The place CRL_i will input to the transitions frr_{ij}, $j=1,\ldots,3$. Similar to frt_{ij}, the purpose of frr_{ij} is to force first-tier cell i to recover the condition of the channel from U1N to AV or from U2N to U1N or from U3N to U2N to reflect the fact that the channel is released in the considered cell. Therefore, there are also input places U1N for frr_{i1}, U2N for frr_{i2}, and U3N for frr_{i3} from first-tier cell i. Similar to frt_{ij}, one of frr_{ij}, $j=1,2,3$ will immediately fire once enabled. In Fig. 4, we also put CRL_4 and frr_{41}, frr_{42}, and frr_{43} to be the graphical expression that describes the relationship between the considered cell and first-tier cell 1 regarding channel release. This completes the dynamics of channel assignment and channel release for one new call.

Remark 6: From the foregoing description, the proposed PN-based automated DDCA method has achieved the event-driven automation based on the assumption that each cell will send the updated real channel condition of each channel to all neighboring cells whenever it changes. That means that when the event of a new call arrival occurs, the process controlled by the proposed PN that leads the new call to one of the two results, i.e., blocked or assigned a channel until call completion, is spontaneous.

E. Channel-Selection Scheme

To maximize channel utilization, the *packing condition* and the *resonance condition* are taken into account in most of the centralized DCA methods in selecting a channel to be assigned for a new call. In centralized DCA, these two conditions are either formulated as terms in the objective function over all cells governed by the MSC (or a central controller) [3], [4], [10] or put into typical compact patterns [5] for assignment reference. Such approaches cannot be implemented in DDCA, because each cell can only have the information from neighboring cells. However, in the real world, the call behavior is of *random* nature, and where and when a new call will arrive is unpredictable. Therefore, a compact pattern like resonance condition is too ideal to occur, and the assignment based on the packing condition can hardly reach the resonance condition either. Thus,

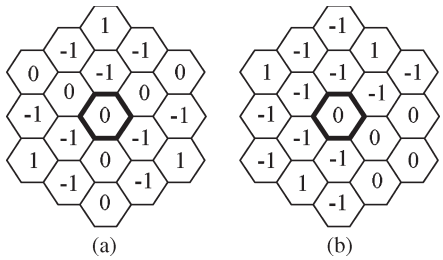

Fig. 5. (a) Condition number of channel c_i for different cells. (b) Condition number of channel c_j for different cells.

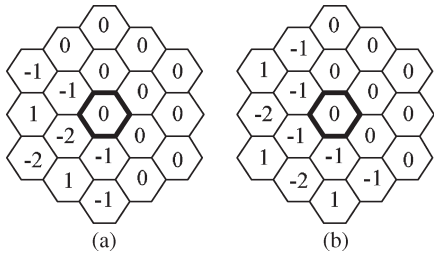

Fig. 6. (a) Condition number of channel c_i for different cells. (b) Condition number of channel c_j for different cells.

the effect of channel utilization resulting from the assignment based on these two conditions is not as good as it seems.

From the call blocking point of view, a call being blocked is simply because there is no *available* channel for assignment. Therefore, the *call blocking probability* will be decreased provided that there are more available channels in the system. This suggests that we should assign the channel, which causes fewer available channels to be unavailable. Fortunately, this idea is particularly suitable for the PN-based automated DDCA, in which whenever a channel is being assigned, the number of available channels in the system becoming unavailable can easily be calculated.

First of all, we will put all the available channels that can enable transition cs in a candidate list. As previously indicated, cs is a T_{rfdc}-type transition, and the condition of some channels may change during the execution of the channel-selection scheme and lose their conditions to enable cs. Such channels cannot be selected for assignment. Therefore, instead of selecting a single channel, we will set the selection priority of the channels in the list by ranking them based on the following channel-selection criteria. However, before presenting the criteria of our scheme, we will first define a function α, which maps the real channel condition to condition numbers, such that $\alpha(\text{AV}) = 0,\ \alpha(\text{IU}) = 1,\ \alpha(\text{U1N}) = -1,\ \alpha(\text{U2N}) = -2$, and $\alpha(\text{U3N}) = -3$. Based on the defined condition numbers, we see that any channel that is available for assignment in the considered cell, the condition number of which should be 0, can only be 0, -1, or -2 (i.e., $\bar{S}$) in first-tier cells due to the enabling conditions of transition cs. Thus, if a channel is assigned for a new call in the considered cell, then the condition number of this channel in first-tier cells will be forced to change from 0 to -1, or from -1 to -2, or from -2 to -3 due to the firing of one frt_{ij}, $j = 1, 2, 3$ for $i = 1, \ldots, 6$. Thus, the first criteria of our channel-selection scheme is to rank the channels in the list, such that the rank of the channel is higher if its condition numbers in the first-tier cells are of a smaller number of 0 s, which will make fewer available channels unavailable if the channel is selected for assignment. For example, Fig. 5(a) and (b) shows the condition numbers of two channels c_i and c_j, where the cell enclosed by a bold boundary represents the considered cell. Then, there will be three (or one) available channels becoming unavailable if channel c_i (or c_j) is selected to assign for the new call in the considered cell. Thus, based on the first criteria, channel c_j has a higher rank than c_i.

In the case that there are more than one channel having the same rank based on the first criteria, we will further compare them using the second criteria. We denote the sum of condition numbers of a channel in all the first-tier cells as the indicator of *regional channel utilization*; thus, this indicator for an available channel is bounded above by 0 and below by -12. In the case of two channels having the same number of available channels in the first tier cells, the channel with the larger indicator implies that there *may be* more cells in the second tier using this channel, the selection of which will lead to a more compact pattern in channel utilization. Therefore, our second criteria is to rank the channels having the same rank resulting from the first criteria in the way that the rank of the channel is higher if its indicator of channel utilization is larger. For example, both channels c_i and c_j shown in Fig. 6(a) and (b) have the same number of 0 in the first-tier cells; however, their indicators of regional channel utilization are -4 and -3, respectively, and we can observe that there are two cells in the second tier using channel c_i in Fig. 6(a) and three cells using channel c_j in Fig. 6(b). Thus, our second criteria is to rank channel c_j higher than channel c_i. Now, we can present our channel-selection scheme as follows.

1) For all the channels in the candidate list for assigning the new call, the channel whose condition numbers in the first-tier cells are of smaller number of 0 s is less than 0 s is ranked higher.
2) In the case that more than one channel is ranked the same from 1), the channel with the larger indicator of regional channel utilization will be ranked higher.
3) If there is still more than one channel ranked the same from 2), then the channel whose index of channel identifier is closer to the channels in use is ranked higher, because operating on channels with largely dispersed identifiers may take longer [18].
4) Once ranking the channels in the candidate list is complete, starting from the top-ranked channel, check whether the associated marked token is still in AV and $\bar{S}_i$, $i = 1, \ldots, 6$. If yes, assign that channel to the new call; otherwise, proceed with the next channel in the list. Supposing that the associated marked tokens of all channels in the list are not in AV or any one of $\bar{S}_i$, $i = 1, \ldots, 6$, then cs is disabled and cannot fire, and the unmarked token associated with the new call will stay in Q, waiting for further assignment.

F. Channel-Reassignment Scheme

When a call is completed, i.e., transition cc occurs, the used channel will be released. In fact, this is an excellent time to *reassign* the channels for existing calls in the considered cell.

In other words, we will let transition cra occur to determine which of the channels that are in use, including the one just completing the call, is most beneficial to release and reassign the corresponding call to the channel just completing the call. However, differing from cs, cc & cra is an intracell-type transition, the condition of whose sole input place IU will not change until we release it. Therefore, channel reassignment will be simpler than channel-selection.

Channel release is the reverse procedure of channel-selection. Therefore, we prefer to release the channel that makes more unavailable channels available as the first criteria. Consequently, our first criteria for the channel-reassignment scheme is to select the channel whose condition numbers in the first-tier cells have the largest number of -1 s.

In the case where there is more than one channel selected based on the first criteria, we will further compare them using the second criteria, which is also a reverse procedure of the second criteria of the channel-selection scheme. That is to say, we will select the channel with the *smallest* indicator of regional channel utilization to release, because the smaller indicator implies that there are fewer cells in the second tier using this channel, and we prefer to keep those with a larger indicator to reach a more compact pattern.

Hence, our channel-reassignment scheme can be stated as follows.

1) Select the in-use channel, including the channel just completing the call, whose condition numbers in the first-tier cells have the largest number of -1 s. If the selected channel is the channel just completing the call, then release the channel; otherwise, release the selected channel, and reassign the corresponding call to the channel that has just completed the call.
2) In the case where there is more than one channel selected from 1), we will choose the channel with the smallest indicator of regional channel utilization to release and reassign the call as in 1) if necessary.
3) If there is more than one channel resulting from 2), then we will select the channel whose identifier is farther from the channels in use and reassign the call as in 1), if necessary.

Note that 3) is a reverse procedure of the third criteria in the channel-selection scheme. Furthermore, under the proposed channel-reassignment scheme, a *limiting reassignment* process reported in [3], [4], and [10] is automatically carried out, because we use the same channels as before.

III. Properties of the Proposed PN

As indicated in Section I, the proposed PN-based automated DDCA has to deal with three challenges, which can be summarized as follows: 1) satisfying the hard constraints; 2) adequate channel-selection to assign for a new call; and 3) channel reassignment. The latter two challenges had been dealt with by the channel-selection and channel reassignment schemes presented in Sections II-E and F, respectively. In this section, we will show that the proposed PN satisfies the hard constraint. In addition, we have to justify the adequacy of the proposed PN in the following respects: 1) The PN is deadlock free; 2) the resources (i.e., channels) are neither lost nor gained; and 3) the system modeled by the PN is stable, i.e., the number of tokens in any one place does not grow infinitely. To accomplish these tasks, we need a reachability tree for the proposed PN.

A. Reachability Tree

The reachability tree of a PN is a tree that uses states as nodes and transitions as arcs [24]. The construction of this tree starts from the root node, which is represented by the initial state, and the arcs outgoing from the root node are marked by the corresponding enabled transitions. The arc will lead to a new node (state) resulting from the firing of the corresponding transition (arc). The preceding procedures repeat until *duplicate nodes*, which are identical to existing nodes in the tree, or *terminal nodes*, which have no any enabled transitions, are met. In the reachability tree, *dashed lines* will be used to indicate the duplicate nodes.

Due to the similar process of assigning a new call for all channels in all cells, we will present the reachability tree of the proposed PN for just one channel in one cell. For the sake of simplicity in representing the node (state) in the reachability tree, we define the state variable vector in the reachability tree as [Q, B] $\times$ [IU, AV, U1N, U2N, U3N, S], in which the first part corresponds to the conditions of a new call, and the second part corresponds to the real channel conditions. In the sequel, we will denote the second part as the channel state. Since the places $\bar{S}_i$, S_i A_i, E_i, CIU_i, and CRL_i, $i = 1, \ldots, 6$ are auxiliary channel conditions, which help enable transitions, we omit representing them in the state variables; however, we will put the effect of their conditions in the corresponding arc (transition).

We let the initial state be [0 0] $\times$ [0 1 0 0 0 0], and we consider the situation that the new call is accepted; a channel is selected for assignment. Then, the reachability tree of a selected channel in a cell can be shown in Fig. 7.

Illustration of the Reachability Tree: Based on the proposed PN in Fig. 4 and starting from the initial state [00] $\times$ [010000], supposing that transition nca occurs and that the call is accepted, the state moves to [10] $\times$ [010000]. At this state, suppose RL $= 0$, transition cb is enabled and fires; then, the state moves to [01] $\times$ [010000]. Subsequently, transition $uncc$ is enabled and fires, and then, the state moves to [00] $\times$ [010000], which is a duplication of the initial state. Supposing that RL $>$ 0 at node [10] $\times$ [010000], and if all $\bar{S}_i$, $i = 1, \ldots, 6$ have the same marked token denoted by $\bigwedge_{i=1}^{6} \bar{S}_i = 1$, then transition cs is enabled, as shown in the corresponding arc, and we assume that the considered channel is selected. Then, the state moves to [00] $\times$ [000001]. Since we have omitted the representation of $\bar{S}_i$, S_i A_i, and E_i, $i = 1, \ldots, 6$ in the state variables, as previously indicated, we use two arcs (or two sets of concatenated events) to represent the handshaking mechanism. The arc on the left, which is marked by $\bigwedge_{i=1}^{6} cya_i$ and ac, represents the case that S and all $\bar{S}_i$, $i = 1, \ldots, 6$ having the same marked tokens enable all transitions cya_i, $i = 1, \ldots, 6$, which fire the same marked tokens to all places A_i, $i = 1, \ldots, 6$, and then enables ac to assign the channel. Subsequently, the state will move to [00] $\times$ [100000]. The arc on the right, which is marked by

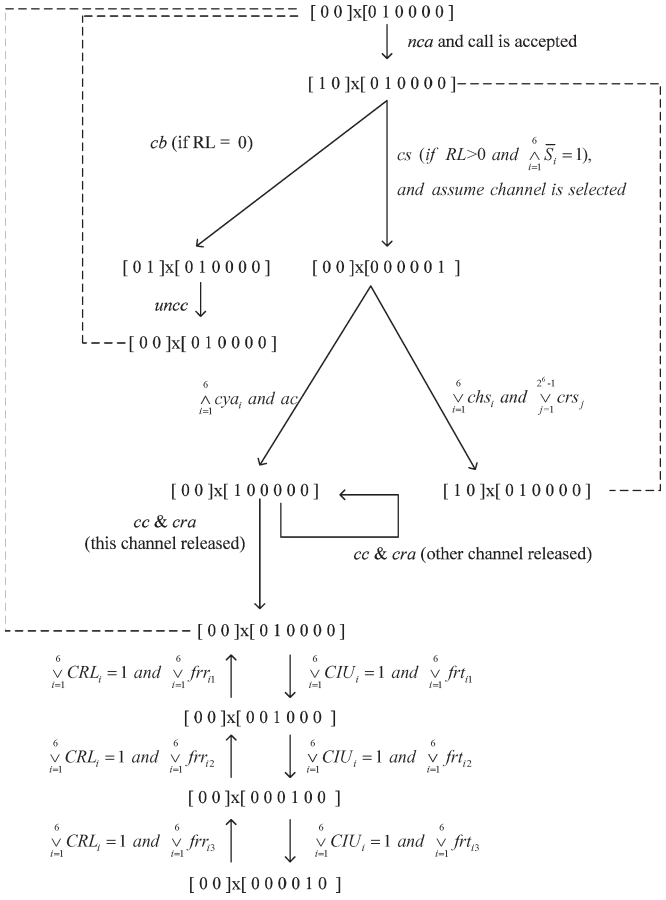

Fig. 7. Reachability tree of the PN for a selected channel in a cell.

$\bigvee_{i=1}^{6} chs_i$ and $\bigvee_{j=1}^{2^6-1} crs_j$, represents the case that S and at least one of S_i, $i = 1, \ldots, 6$ having the same marked tokens enable at least one of the transitions chs_i, $i = 1, \ldots, 6$, which fire the same marked tokens to at least one of the places E_i, $i = 1, \ldots, 6$ and then enables one of the crs_j, $j = 1, \ldots, 2^6 - 1$ to proceed with channel reselection. Subsequently, the state will move to [10] $\times$ [010000], which is a duplicate node, as indicated by the dashed line in Fig. 7. These two sets of concatenated events can be observed from the PN shown in Fig. 4. At state [00] $\times$ [100000], when cc & cra occur, there are two possible results after the execution of the channel reassignment scheme. One result is the considered channel determined to release, and the other result is other channel being released. Supposing that the former occurs as marked on the straight down arc, the state moves to [00] $\times$ [010000], which is a duplication of the initial state, as indicated in Fig. 7. Supposing that the latter occurs as marked on the arc of the self loop, then the state for the considered channel remains unchanged. When the state is at [00] $\times$ [010000] (i.e., channel condition is AV), and supposing that any one of CIU_i, $i = 1, \ldots, 6$ has the same marked token denoted by $\bigvee_{i=1}^{6} CIU_i = 1$ (i.e., the channel condition for one of the six first-tier cells is IU, [100000]), then one of the events frt_{i1}, $i = 1, \ldots, 6$ occurs as indicated by $\bigvee_{i=1}^{6} frt_{i1}$; this event is marked on the arc, and the state moves to [00] $\times$ [001000] (i.e., channel condition is U1N). Similar situations occur to states [00] $\times$ [001000] and [00] $\times$ [000100], as shown in the bottom three states of Fig. 7. When the state moves to [00] $\times$ [000010] (i.e., channel condition is U3N), and supposing that one of CRL_i, $i = 1, \ldots, 6$ has the same marked token denoted by $\bigvee_{i=1}^{6} CRL_i = 1$ (i.e., one of the six first-tier cells just releases the channel), then one of the events frr_{i3}, $i = 1, \ldots, 6$ occurs as indicated by $\bigvee_{i=1}^{6} frr_{i3}$ marked on the arc, and the state moves to [00] $\times$ [000100] (i.e., channel condition is U2N). Similar situations occur to states [00] $\times$ [000100] and [00] $\times$ [001000], as shown in Fig. 7.

B. Deadlock Free

Deadlock is a situation where no further transition can be fired. A complement to this property is the notion of *live* transition. We say that a PN is *live* if there always exists some sample path, such that any transition can eventually fire from any state reached from the initial state [24].

Proposition 1: The proposed PN is live and deadlock free.

Proof: 1) First, we consider the case that the new call is accepted and a channel is selected for assignment, based on which the reachability tree is constructed, as shown in Fig. 7. In this figure, we see that there is no terminal node; thus, there always exists some sample path to reach any state from the initial state. The spontaneous handshaking mechanism formed by the places S, $\bar{S}_i$, S_i, A_i, and E_i, $i = 1, \ldots, 6$, the transitions cya_i and chs_i, $i = 1, \ldots, 6$, and ac and crs_j, $j = 1, \ldots, 2^6 - 1$ has been concatenated by two arcs, as previously described, and the rest of the arcs contain the rest of the transitions in the proposed PN. Since the arc is always outgoing from a node (state), i.e., enabled at that state, then based on the fact that there is no terminal node in Fig. 7, there must exist some sample path, such that any transition can eventually fire from any state reached from the initial state. By the definition of liveness, we conclude that the proposed PN is live under the considered case. 2) For the case that a new call arrives but is rejected, then the transition nca does not fire. However, this cannot be a deadlock, because the occurrence of rejection is due to the number of calls in the PN exceeding N_c. Since any ongoing call will eventually complete, the number of calls in PN must decrease; thus, the transition nca will eventually fire. 3) The case left to be considered is when the new call enters the place Q, as shown in the second node from the top of Fig. 7, and transition cs is either not enabled or enabled at the beginning but disabled by the end of the execution of the channel-selection scheme, as indicated in Section II-E; then, cs will not fire. However, this cannot be a deadlock, because 1) if cs is not enabled during the entire lifetime of an unmarked token, then the transition cb will be enabled and fire, and 2) cs will eventually fire, because the channel selected and used by the neighboring cells will eventually be released due to call completion. Based on the above cases, we conclude that the PN is live and deadlock free. ■

C. Conservation

It is important to ensure that the resource (i.e., channel) is neither gained nor lost. Thus, conservation of resources is

an important property for PN. The places related to the real channels are places IU, AV, U1N, U2N, U3N, and S. Then, by the definition of conservation [24], we say that the proposed PN is conservative with respect to any channel, for example, c, if $\mathrm{c(IU)} + \mathrm{c(AV)} + \mathrm{c(U1N)} + \mathrm{c(U2N)} + \mathrm{c(U3N)} + \mathrm{c(S)} = 1$, where $c(P)$ denotes the number of considered marked tokens associated with channel c in place P.

Proposition 2: The proposed PN is conservative with respect to any channel.

Proof: From the reachability tree shown in Fig. 7, we can observe that one and only one marked token corresponding to any specific channel, for example, c, can appear in one and only one of the places AV, IU, U1N, U2N, U3N, and S for all the states in the reachability tree. This implies that $\mathrm{c(IU)} + \mathrm{c(AV)} + \mathrm{c(U1N)} + \mathrm{c(U2N)} + \mathrm{c(U3N)} + \mathrm{c(S)} = 1$. Then, based on the definition of conservation, we complete the proof. ■

D. Satisfying the Hard Constraint

The hard constraint of DDCA is any two cells within the channel reuse distance cannot use the same channel. This constraint can be interpreted here as that no two neighboring cells can simultaneously have the channel state [IU, AV, U1N, U2N, U3N, S] for the same channel to be [1 0 0 0 0 0].

Proposition 3: For the same channel, no two neighboring cells in the proposed PN can simultaneously have the channel state [1 0 0 0 0 0].

Proof: Without loss of generality, we can assume that each channel of a cell starts from the initial state [0 0] × [0 1 0 0 0 0]. To prove the proposition, we should show that when the state of the cell moves to [0 0] × [1 0 0 0 0 0], the channel state of this channel in any of the six first-tier cells cannot be [1 0 0 0 0 0]. Starting from the state [0 0] × [0 1 0 0 0 0] supposing that a new call arrives and that the call is accepted, then the state will move to [1 0] × [010000]. Subsequently, the considered channel can be selected for assignment only when $[\bar{S}_i\ S_i] = [1\ 0]$ for all $i = 1, \ldots, 6$. Suppose that the transition cs is enabled at time t and that this channel is selected (i.e., cs fires) at time $t + \Delta t_{cs}$, which implies $[\bar{S}_i\ S_i]$ is still [1 0] at $t + \Delta t_{cs}$ for all $i = 1, \ldots, 6$. We should recall that Δt_{cs} represents the execution time of the channel-selection scheme. Then, the state of the considered cell becomes [0 0] × [0 0 0 0 1]. However, $[\bar{S}_i\ S_i] = [1\ 0]$ at time $t + \Delta t_{cs}$ represents the condition of this channel in first-tier cell i at time $t + \Delta t_{cs} - \Delta t_d$, where Δt_d represents the message propagation delay, as previously defined. During $(t + \Delta t_{cs} - \Delta t_d, t + \Delta t_{cs})$, the condition of this channel in all the first-tier cells may have two possibilities: One is still at $\bar{S}_i$, and the other becomes S. If the former occurs to all the first-tier cells, i.e., the token remains at $\bar{S}_i$ for $i = 1, \ldots, 6$ during $(t + \Delta t_{cs}, t + \Delta t_{cs} + \Delta t_d)$, then based on the proposed PN and Remark 4, the event $\bigwedge_{i=1}^{6} cya_i$ occurs at time $t + \Delta t_{cs} + \Delta t_d$ and is immediately followed by the event ac. Subsequently, the state of the considered cell becomes [0 0] × [1 0 0 0 0 0]; we denote this case as case A. If the latter occurs to any of the first-tier cells, i.e., the token moves from $\bar{S}_i$ to S_i during $(t + \Delta t_{cs}, t + \Delta t_{cs} + \Delta t_d)$ for the corresponding i, then this implies that at least one of the first-tier cells earlier selected this channel. Subsequently, at least one of the transition chs_i, $i = 1, \ldots, 6$ occurs at time $t + \Delta t_{cs} + \Delta t_d$ and is immediately followed by the occurrence of one of the transitions crs_j, $j = 1, \ldots, 2^6 - 1$. Consequently, the state of the considered cell becomes [1 0] × [0 1 0 0 0 0] at time $t + \Delta t_{cs} + \Delta t_d$; we denote this case as case B.

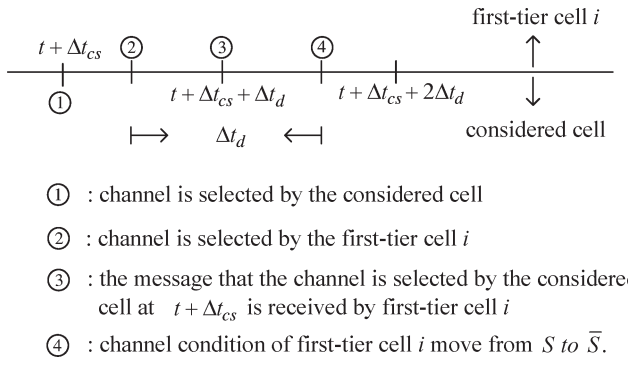

Fig. 8. Second subcase of case A.

In case A, the state is [0 0] × [1 0 0 0 0 0] at $t + \Delta t_{cs} + \Delta t_d$, and a message is sent to all first-tier cells to indicate that this channel is in use, which makes a marked token associated with this channel appear in CIU_i, $i = 1, \ldots, 6$ at time $t + \Delta t_{cs} + 2\Delta t_d$ due to message propagation delay. Now, we consider two subcases of case A. The first subcase is that the condition of this channel in all first-tier cells still remains at $\bar{S}$ during $(t + \Delta t_{cs}, t + \Delta t_{cs} + 2\Delta t_d)$. Then, one of the transitions frt_{ij}, $j = 1, 2, 3$, for all $i = 1, \ldots, 6$ fires at time $t + \Delta t_{cs} + 2\Delta t_d$. Since $\bar{S}$ represents one of AV, U1N, and U2N, therefore, after firing frt_{i1}, or frt_{i2}, or frt_{i3}, the condition of this channel in first-tier cells will move from [0 1 0 0 0 0] to [0 0 1 0 0 0], or from [0 0 1 0 0 0] to [0 0 0 1 0 0], or from [0 0 0 1 0 0] to [0 0 0 0 1 0], respectively. In this subcase, the conclusion of this proposition is proved. The second subcase of case A is that the condition of this channel in any one of the first-tier cells, for example, i, moves from $\bar{S}$ to S during $(t + \Delta t_{cs}, t + \Delta t_{cs} + \Delta t_d)$, as indicated at point ② in Fig. 8. Note that this channel had been selected by the considered cell at $t + \Delta t_{cs}$, as previously described and indicated by the point ① in Fig. 8. This implies that the considered cell selects this channel earlier, the message of which is received by first-tier cell i at time $t + \Delta t_{cs} + \Delta t_d$, as indicated by the point ③ in Fig. 8. Then, based on previous analysis for case B, if the state of this channel in first-tier cell i is S, i.e., [0 0 0 0 0 1], then it will move to [0 1 0 0 0 0], i.e., one of the conditions represented by $\bar{S}$, before $t + \Delta t_{cs} + 2\Delta t_d$, as indicated by point ④ in Fig. 8, because the time duration for chs_i is Δt_d and $crs_j \in T_{\mathrm{ai}}$. Then, in the second subcase, frt_{i1} is enabled and immediately fires at $t + \Delta t_{cs} + 2\Delta t_d$, and the state of this channel in the corresponding first-tier cell i will move from [0 1 0 0 0 0] to [0 0 1 0 0 0]. Thus, in the second subcase of case A, the conclusion of this proposition is also proved.

In case B, the state of the considered cell is [1 0] × [0 1 0 0 0 0] at time $t + \Delta t_{cs} + \Delta t_d$; then, 1) it has no chance to simultaneously have the channel state [1 0 0 0 0 0] with the neighboring cells, and 2) it returns to a state being visited and

previously discussed. Thus, the conclusion of this proposition also holds for case B. In any case, the conclusion of this proposition is proved. ■

E. Boundedness

To ensure the stability of a PN, we have to make sure that the number of tokens in any place will not infinitely grow. In other words, the number of tokens in any place should be bounded above. We say that a PN is k-bounded if the number of tokens in any place does not exceed k [24]. We will verify that the proposed PN is N_c bounded in the following.

Proposition 4: The proposed PN is N_c bounded.

Proof: Starting from place Q in Fig. 4, we will consider place by place in this proof. In the case that cs fires, the unmarked token associated with a new call in place Q will either combine with the marked token associated with the selected channel throughout the channel assignment process until call completion or return to Q for channel reselection if the current channel assignment fails. Combining with the marked token makes the unmarked token disappear. Returning to Q due to failed assignment does not increase the number of unmarked tokens. In the case that cs is neither enabled nor fired, the unmarked token will just stay in Q: neither gained nor lost. Furthermore, if the RL of the new call is equal to 0, then the corresponding unmarked token in Q will be shot to place B. Based on the foregoing analysis, we see that any unmarked token once appearing in Q will eventually disappear. Since we have controlled the number of calls in the cell, including the already assigned and yet assigned calls, to be less than N_c, the number of unmarked tokens in Q at any time instance should be bounded above by N_c. Any unmarked token appearing in B will immediately disappear, because transition $uncc$ is a $T_{\rm ai}$-type transition. Thus, the number of unmarked tokens in B is bounded above by 1. Furthermore, the number of marked tokens in any place associated with real channel conditions, i.e., IU, AV, U1N, U2N, U3N and S, is bounded above by N_c due to the following: 1) There are at most N_c channels and 2) the channel conservation shown in Proposition 2. At the moment, when a marked token appears in CIU_i (i.e., the channel condition of first-tier cell i is IU), the channel condition of the considered cell must be in $\bar{S}$ according to the analysis of case A in the proof of Proposition 3. Therefore, the marked token will immediately shoot once it arrives at CIU_i, because frt_{ij} for every i and j is a $T_{\rm ri}$-type transition. A similar situation occurs to CRL_i, because 1) at the moment, when a marked token appears at CRL_i, the channel condition of the considered cell must be either U1N, U2N, or U3N due to the hard constraint, and 2) frr_{ij} for every i and j is also a $T_{\rm ri}$-type transition. Therefore, the number of marked tokens in CIU_i and CRL_i, $i = 1, \ldots, 6$ is bounded above by 1. Places $\bar{S}_i$, S_i, $i = 1, \ldots, 6$ are used to record the channel conditions of first-tier cell i; thus, the marked tokens in S_i and $\bar{S}_i$ will neither gain nor lose during the operation of PN in the considered cell. Furthermore, for each i, only one of $\bar{S}_i$ and S_i or none can have the marked token for the corresponding channel. Since there are at most N_c channels, then based on the above description, the number of marked tokens in S_i and $\bar{S}_i$ should be bounded above by N_c for any i. For each i, only one of A_i and E_i can have the marked token associated with the selected channel in the handshaking process, and the tokens in A_i or E_i for $i = 1, \ldots, 6$ will immediately shoot once they arrive, because ac and crs_j, $j = 1, \ldots, 2^6 - 1$ are $T_{\rm ai}$-type transitions. Therefore, the number of marked tokens in A_i and E_i is bounded above by 1. Based on the above facts, we conclude that the proposed PN is N_c bounded. ■

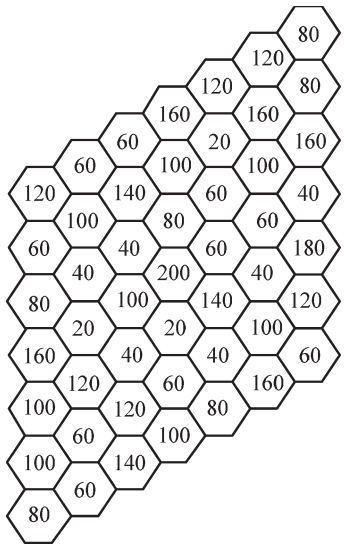

Fig. 9. Cellular network (7 × 7) with traffic pattern #1.

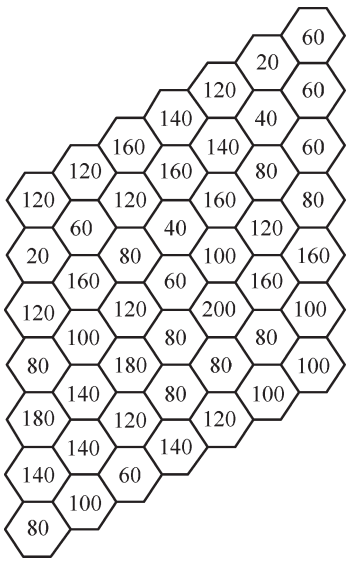

Fig. 10. Cellular network (7 × 7) with traffic pattern #2.

IV. Performance Tests and Comparisons

In our simulations, we use the 7 × 7 parallelogram-shaped cellular network formed by 49 regular hexagonal cells shown in Fig. 9, which is also employed in [3], [4], and [10]. We assume the following for the employed cellular network: 1) The cellular network is a three-cell cluster system; 2) the total number of channels available in the system is 70; 3) each channel can serve only one call; and 4) we consider cochannel

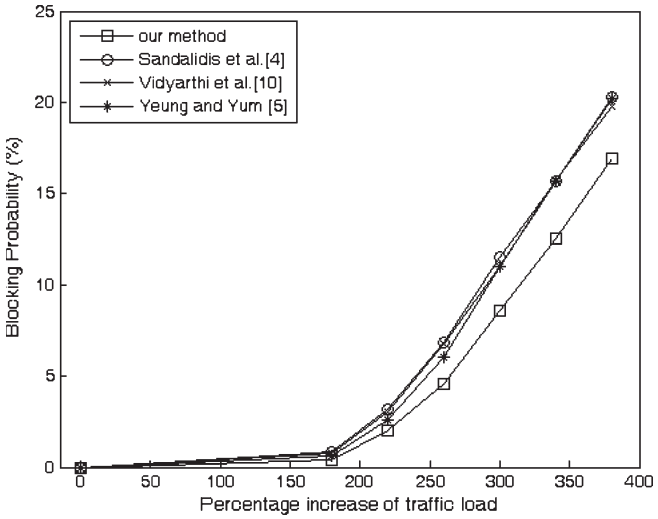

Fig. 11. Blocking probability of all methods under various traffic loads for traffic pattern #1.

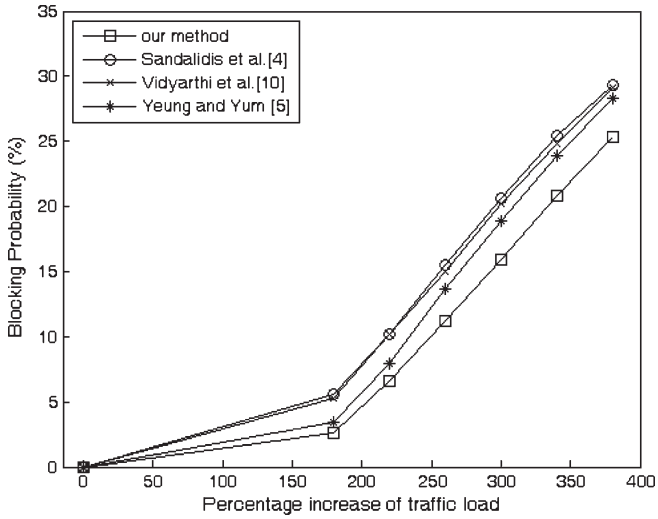

Fig. 12. Blocking probability of all methods under various traffic loads for traffic pattern #2.

interference only, whereas other sources of interference, such as adjacent channel interference, are ignored. We assume that the call arrivals follow a Poisson process with the probability distribution for interarrival time as $e^{-\lambda t}$, where λ represents the mean call-arrival rate. We also assume that the call-holding time follows an exponential probability distribution $e^{-t/\mu}$, where μ represents the mean call-holding time, and we set $\mu = 180$ s throughout our simulations. To adapt our simulations to realistic situations, we consider two nonuniform traffic patterns denoted by traffic patterns #1 and #2, which are employed in [3], [4], and [10], as shown in Figs. 9 and 10, respectively. The numbers with unit calls/hour shown in each cell in Figs. 9 and 10 denote the *initial* mean call-arrival rate. We set the call request response time to be 1 s. The performance of our PN-based automated DDCA method, as well as the other comparing methods, is evaluated by the blocking probability of the incoming calls for the whole system. To test the performance of our method, as well as the other comparing methods, under various traffic loads, we have increased the traffic load by a percent factor ranging from 180 to 380. A percentage increase of traffic load implies that the initial call-arrival rates for all cells are increased by that percentage.

For each traffic pattern and each traffic load, we have simulated our PN-based DDCA method for the simulation length of 20 h, and the resulting blocking probabilities are shown in Figs. 11 and 12 (marked by □). In the meantime, we also record the corresponding *average number of total available channels left in the system* counted at every 100 calls in Figs. 13 and 14 (also marked by □).

Verification of Spontaneous Handshaking Mechanism: For each cell, we set a flag, the initial value of which is set to be 0. According to the dynamics of the spontaneous handshaking, channel assignment, and channel reselection presented in Section II-D, we may perform the following during the simulations to verify the spontaneous handshaking mechanism.

For each cell, if a channel is selected (the corresponding marked token is in place S) and then assigned for a new call (the

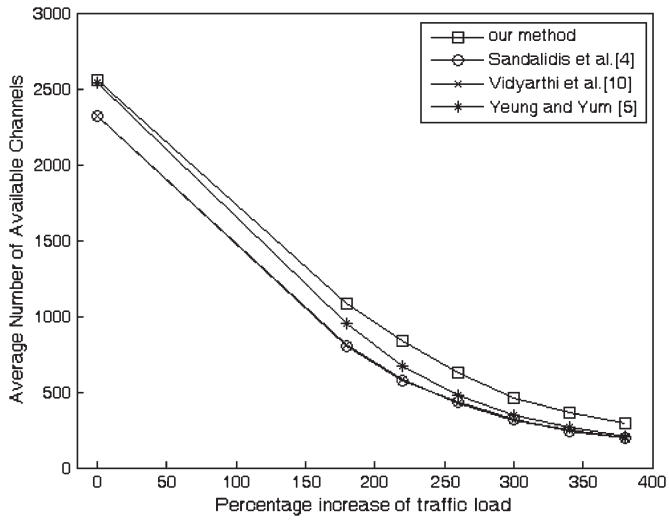

Fig. 13. Average number of total available channels counted at every 100 calls under various traffic loads for traffic pattern #1.

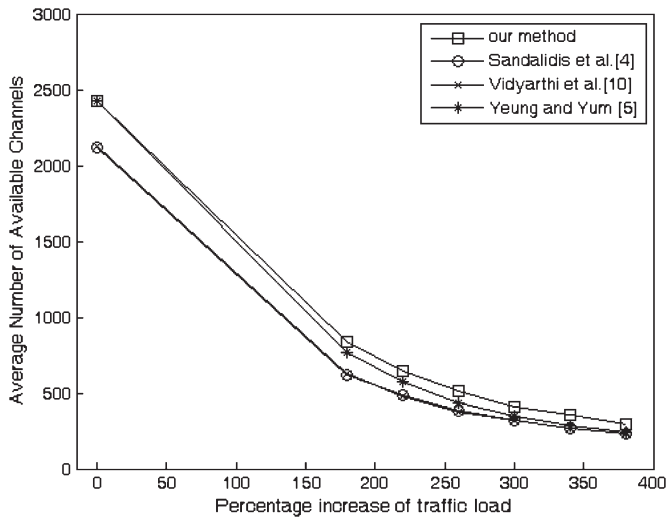

Fig. 14. Average number of total available channels counted at every 100 calls under various traffic loads for traffic pattern #2.

corresponding marked token is in place IU) due to the firing of transition ac (assign channel), then we will check whether the corresponding marked token of this channel appears in all $\bar{S}_i$, $i = 1, \ldots, 6$; if the result is true, then we let the value of the flag unchanged; otherwise, we set flag := flag+1. On the other hand, if a channel is selected and then becomes available (the corresponding marked token is in place AV) due to the firing of any one of the transitions crs_j, $j = 1, \ldots, 2^6 - 1$ (channel reselection), we will check whether the corresponding marked token of this channel appears in at least one S_i, $i = 1, \ldots, 6$; if the result is true, then we let the value of the flag unchanged; otherwise, we set flag := flag + 1. For each simulation case, if the value of flag in every cell remains 0 at the end of simulation, we conclude that the spontaneous handshaking mechanism successfully carries out in the complete simulation. We tested this verification process in all the above simulation cases, and we found that the value of flag of each cell in each case remains 0 at the end of simulation.

For the purpose of comparisons, we use the methods proposed by Sandalidis *et al.* [4], Yeung and Yum [5], and Vidyarthi *et al.* [10] to solve the same test problems. The setup of these methods for the test is described below. For the evolutionary algorithms proposed in [4] and [10], we consider the combinatorial evolution strategy–DCA model in [4] and the pure DCA mode in [10]. The fitness functions in [4] and [10] employed here are taken for the three-cell cluster system; we use the same parameters and the source code as they used in their simulations. For the method proposed in [5], we modify their two-phase compact pattern-based DCA strategy for the three-cell cluster system according to the guide lines provided in [5].

For each traffic pattern and each traffic load, we have simulated the methods proposed in [4], [5], and [10] with the above setup for the simulation length of 20 h, and the blocking probabilities and the average number of available channels left in the system counted at every 100 calls obtained by these three methods are also shown in Figs. 11–14, as marked by O, $\times$, and $*$, respectively.

From Fig. 13, we see that our method possesses the largest average number of available channels left in the system counted at every 100 calls for traffic pattern #1 under various traffic load. In fact, this provides evidence for our method achieving the smallest blocking probability for traffic pattern #1 under various traffic load shown in Fig. 11 and demonstrates the superiority of our channel-selection and channel-reassignment schemes. Similar conclusions apply to traffic pattern #2 and can be observed from Figs. 12 and 14. Thus, the proposed PN-based automated DDCA method not only works well in a distributed manner but also achieves better performance measured by the blocking probability. Furthermore, it is worth noting that all the methods are coded in C language and implemented in a Pentium-3 20-GHz processor and 1.00-GB random-access-memory personal computer. The central-processing-unit time consumed by our method, excluding the verification of spontaneous handshaking mechanism in each channel assignment, is 4.12551×10^{-5} s on the average, which, for the methods in [4], [10], and [5], are 0.033608, 0.027266, and 4.30038×10^{-5} s on average, respectively.

V. Conclusion

In this paper, we have proposed a PN-based automated DDCA method for a cellular network. The mechanism of the proposed PN is thoroughly described, and we have justified its adequacy. Using the proposed PN, we successfully handle the hard constraint of the cellular network and prove the satisfaction of which in a rigorous manner.

We bring up a new channel-selection and channel-reassignment criteria and demonstrate its superiority by comparing it with the existing methods using numerous simulations. Therefore, we not only propose a clear and efficient automated DDCA method to resolve the overload problem in MSC but also achieve better performance.

Acknowledgment

The authors would like to thank the anonymous reviewers, particularly the reviewer who indirectly suggested to add the word "automated" in the title of the paper.

References

[1] W. C. Y. Lee, Jr., *Microware Cellular Telecommunication Systems*. New York: McGraw-Hill, 1989.
[2] M. Zhang and T. S. P. Yum, "Comparisons of channel-assignment strategies in cellular mobile telephone systems," *IEEE Trans. Veh. Technol.*, vol. 38, no. 4, pp. 211–215, Nov. 1989.
[3] E. Del Re, R. Fantacci, and L. Ronga, "A dynamic channel allocation technique based on Hopfield neural networks," *IEEE Trans. Veh. Technol.*, vol. 45, no. 1, pp. 26–32, Feb. 1996.
[4] H. G. Sandalidis, P. P. Stauroulakis, and J. Rodriguez-Tellez, "An efficient evolutionary algorithm for channel resource management in cellular mobile systems," *IEEE Trans. Evol. Comput.*, vol. 2, no. 4, pp. 125–137, Nov. 1998.
[5] K. L. Yeung and T.-S. P. Yum, "Compact pattern based dynamic channel assignment for cellular mobile systems," *IEEE Trans. Veh. Technol.*, vol. 43, no. 4, pp. 892–896, Nov. 1994.
[6] J. C.-I. Chuang, "Performance issues and algorithms for dynamic channel assignment," *IEEE J. Sel. Areas Commun.*, vol. 11, no. 6, pp. 955–963, Aug. 1993.
[7] K.-N. Chary, J.-T. Kim, C.-S. Yim, and S. Kim, "An efficient borrowing channel assignment scheme for cellar mobile systems," *IEEE Trans. Veh. Technol.*, vol. 47, no. 2, pp. 602–608, May 1998.
[8] H. G. Sandalidis, P. P. Stavroulakis, and J. Rodriguez-Tellez, "Borrowing channel assignment strategies based on heuristic technique for cellular systems," *IEEE Trans. Neural Netw.*, vol. 10, no. 1, pp. 176–181, Jan. 1999.
[9] T. J. Kahwa and N. D. Georgans, "A hybrid channel assignment schemes in large-scale, cellular-structured mobile communication systems," *IEEE Trans. Commun.*, vol. COM-26, no. 4, pp. 432–438, Apr. 1978.
[10] G. Vidyarthi, A. Ngom, and I. Stojmenovic, "A hybrid channel assignment approach using an efficient evolutionary strategy in wireless mobile networks," *IEEE Trans. Veh. Technol.*, vol. 54, no. 5, pp. 1887–1895, Sep. 2005.
[11] K. Ramachandran, E. Belding, K. Almeroth, and M. Buddhikot, "Interference-aware channel assignment in multi-radio wireless mesh networks," in *Proc. INFOCOM*, Barcelona, Spain, Apr. 2006, pp. 1–12.
[12] A. P. Subramanian, H. Gupta, S. R. Das, and J. Cao, "Minimum interference channel assignment in multi-radio wireless network," *IEEE Trans. Mobile Comput.*, vol. 7, no. 12, pp. 1459–1473, Dec. 2008.
[13] R. Akl and A. Arepally, "Dynamic channel assignment in IEEE 802.11 networks," in *Proc. PORTABLE*, Orlando, FL, May 25–29, 2007, pp. 1–5.
[14] X. J. Li and P. H. J. Chong, "A dynamic channel assignment scheme for TDMA-based multihop cellular networks," *IEEE Trans. Wireless Commun.*, vol. 7, no. 6, pp. 1999–2003, Jun. 2008.
[15] X. J. Li and P. H. J. Chong, "Dynamic channel assignment for multihop cellular networks with any reuse factor," *IEEE Commun. Lett.*, vol. 12, no. 5, pp. 346–348, May 2008.

[16] F. James-Romero, D. Munoz-Rodriguez, C. Molina, and H. Tawfik, "Modeling resource management in cellular systems using Petri nets," *IEEE Trans. Veh. Technol.*, vol. 46, no. 2, pp. 298–312, May 1997.
[17] R. Mathar and J. Matlfeldt, "Channel assignment in cellular radio networks," *IEEE Trans. Veh. Technol.*, vol. 42, no. 4, pp. 647–656, Nov. 1993.
[18] Y. Zhany, S. K. Das, and X. Jia, "D-CAT: An efficient algorithm for distributed channel allocation in cellular mobile networks," *Mobile Netw. Appl.*, vol. 9, no. 4, pp. 279–288, Aug. 2004.
[19] H. G. Sandalidis, P. P. Stavroulakis, and J. Rodrignez-Tellez, "Implementation of genetic algorithm to a channel assignment problem in cellular communications," in *Proc. 6th Int. Conf. Advances COMCON*, Jun. 1997, pp. 453–460.
[20] M. Choy and A. K. Singh, "Efficient distributed algorithms for dynamic channel assignment," in *Proc. 7th IEEE Int. Symp. Pers., Indoor, Mobile Radio Commun.*, 1996, pp. 208–212.
[21] A. Boukerche, S. Hong, and T. Jacob, "A distributed algorithm for dynamic channel allocation," *Mobile Netw. Appl.*, vol. 7, no. 2, pp. 115–126, Apr. 2002.
[22] A. Boukerche, S. Hong, and T. Jacob, "A performance study of a distributed algorithm for dynamic channel allocation," in *Proc. ACM MSWiM*, Aug. 2000, pp. 36–43.
[23] J. Yang, Q. Jiang, D. Manivannan, and M. Singhal, "A fault-tolerant distributed channel allocation scheme for cellular network," *IEEE Trans. Comput.*, vol. 54, no. 5, pp. 616–629, May 2005.
[24] C. G. Cassandras and S. Lafortune, *Introduction to Discrete Event Systems*. Boston, MA: Kluwer, 1999.
[25] W. C. Jakes, Jr., *Microwave Mobile Communication*. New York: Wiley, 1974.

Shin-Yeu Lin was born in Taiwan. He received the B.S. degree in electronics engineering from National Chiao Tung University, Hsinchu, Taiwan, in 1975, the M.S. degree in electrical engineering from the University of Texas at El Paso in 1979, and the D.Sc. degree in systems science and mathematics from Washington University, St. Louis, MO, in 1983.

From 1984 to 1985, he was a Research Associate and then a Visiting Assistant Professor with Washington University. From 1985 to 1986, he was a Senior Member of technical staff with GTE Laboratory. From 1987 to July 2009, he was an Associate Professor and then a Professor with the Department of Electrical and Control Engineering, National Chiao Tung University. Since August 2009, he has been a Professor with the Department of Electrical Engineering, Chang Gung University, Tao-Yuan, Taiwan. His major research interests include wireless communication, ordinal optimization theory and applications, optimal power flow, and distributed computations.

Ting-Yu Chan was born in Taiwan. He received the B.S. degree in mechanical engineering and M.S. degree in electrical and control engineering from the National Chiao Tung University, Hsinchu, Taiwan, in 2006 and 2008, respectively.

He is currently an Engineer with Elan Microelectronics Corporation, Hsinchu. He is also with the Department of Electrical and Control Engineering, National Chiao Tung University. His major research interests include optimization theory and applications, wireless communication, and integrated circuit design.