Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics
San Antonio, TX, USA - October 2009

Novel Reconfigurable Randomized Broadcast Algorithm for Channel-Aware Wireless Networks

Scott C.-H. Huang, Shih Yu Chang, *Member, IEEE,* Hsiao-Chun Wu, *Senior Member, IEEE,* and Peng-Jun Wan

***Abstract*— In this paper, we study the channel-aware minimum-latency broadcast scheduling problem using the probabilistic model. We establish an explicit relationship between the tolerated transmission-failure probability and the latency of the corresponding broadcast schedule. Such a tolerated transmission-failure probability is calculated in the strict sense that the failure to receive the message at any single node will lead to the entire broadcast failure and only if all nodes have successfully received the message, do we consider it a successful broadcast. We design a novel reconfigurable broadcast scheduling algorithm such that the latency is evaluated under such a strict definition of failure. Our derived latency bound associated with this new randomized algorithm is substantial to guarantee the low broadcast latency for the complete broadcasting success thereby.**

***Index Terms*— Wireless networks, scheduling, randomized algorithm.**

I. Introduction

Broadcast is a classical problem that arises in many applications of communications. For multi-hop wireless networks, in particular, broadcast is a very time-consuming operation because it involves tedious contention, collision, and congestion. How to reduce the broadcast latency can be deemed quite challenging. There exist many different approaches to reduce the latency [1], [2]. Scheduling is one of the most effective approaches. By carefully scheduling each node's message transmission, we can often avoid both interference and collision. To achieve the minimum-latency broadcast schedule with nearly optimal latency, we will focus on the randomized scheduling algorithm in this paper.

The conventional network scheduling approach is often based on the *deterministic* model, in which all message transmissions are assumed successful in the absence of interference. For example, in the deterministic model, each receiver operates on an *imaginary transmission* associated with an *interference range*. That is, a receiver is guaranteed to receive a message successfully at a certain time, if, at this time, exactly one node (or transmitter) within its transmission range is transmitting and no other node within its interference range is interfering (i.e. transmitting another message). As a matter of fact, this assumption does not reflect the probabilistic nature of realistic wireless communications. However, the practical interference channel environment was not extensively addressed in the existing literature. Hence, we would like to dedicate this paper to the studies of the wireless broadcasting schedules in the interference channels.

Henceforth, we would like to consider the broadcast schedule problem in the interference channel characterized by the *signal-to-interference-plus-noise ratio* (SINR). Moschibroda and Wattenhofer considered the problem of scheduling in a given network topology using the SINR model [3]. In practice, we may not be able to often acquire the network topology information in real time, if the interference is considered. In other words, we need to do scheduling to broadcast the messages throughout the entire network even in the dynamical interference environment without constant network information updates, but the previous work in [3] focused only on the latency issue. We consider a more practical situation here. In the wireless broadcast, a non-source node cannot transmit a message unless it has already received from another node beforehand. This assumption makes our work fundamentally different from [3]. We impose a probabilistic restriction on the successful message transmission arising from the interference channels using the theoretical results in [4] where it was proved that the data aggregation rates $\Theta((\log n)/n)$ and $\Theta(1)$ are optimal for the communication systems with the path-loss exponents $2 < \alpha < 4$ and $\alpha > 4$, respectively.

In this paper, we also consider the minimum-latency broadcast scheduling problem, in which the message transmission failure is quantified in a probabilistic manner. Our studies here can be deemed as a new attempt to design the effective scheduling schemes for practical wireless networks in the interference environment. We adopt the SINR model to tackle this problem and establish an explicit relationship between the tolerated transmission-failure probability and the latency of the corresponding broadcast schedule. We calculate the tolerated transmission-failure probability in the strict sense that a single message-transmission failure will result in the failure of the whole broadcast task. Only if all nodes have received the message successfully, do we call it a successful broadcast task. Our novel scheduling algorithm is designed in a very careful way that, even under such a strict definition of failure, our algorithm can still achieve a low broadcast latency to solve the minimum-latency broadcast scheduling problem.

The rest of this paper is organized as follows. In Section II, we present the network model, the crucial parameters and

Scott C.-H. Huang is with the Department of Computer Science, City University of Hong Kong, Kowloon, Hong Kong (e-mail: shuang@cs.cityu.edu.hk). Shih Yu Chang is with the Department of Computer Science, National Tsing Hua University, Taiwan (e-mail: shihyuch@cs.nthu.edu.tw). Hsiao-Chun Wu is with the Department of Electrical and Computer Engineering, Louisiana State University, Baton Rouge, LA 70803, USA (e-mail: wu@ece.lsu.edu). Peng-Jun Wan is with the Department of Computer Science, Illinois Institute of Technology, Chicago, IL, USA (e-mail: wan@cs.iit.edu).

978-1-4244-2794-9/09/$25.00 ©2009 IEEE

the assumptions to be used in later sections, particularly the tessellation and the coloring techniques. We present our novel randomized broadcast scheduling algorithm in Section III. A concrete example is given in Section IV. In Section V, we focus on a very important parameter γ, defined in Section II, and discuss how to appropriately select it to make a fully-connected network regime. Numerical results are given in Section VI to evaluate our proposed method. Concluding remarks will be drawn in Section VII. All the proofs for the underlying theorems in this paper will be omitted due to the page limit.

II. Network Model and Problem Statement

Let V be the set of nodes within the network of interest in a two-dimensional Euclidean space, and each node is associated with an identical transmission power P. According to physics, we know that if a node $u \in V$ (transmitter) is transmitting with power P, the theoretical received signal strength P_v at another node $v \in V$ (receiver) is given by

$$P_v = \frac{P}{d(u,v)^{\alpha}}, \tag{1}$$

where $d(u,v)$ is the distance between u, v and α is a constant called the *path-loss exponent*. A commonly assumed path-loss exponent α is greater than **two** [5]. Hence, the theoretical interference strength I_v is

$$I_v = \sum_{w \in T-\{u\}} \frac{P}{d(v,w)^{\alpha}}. \tag{2}$$

In Eq. (2), $T \subset V$ is the set of the nodes scheduled to transmit in the current time slot, in which only u alone is the transmitter and all other nodes are interferers. The SINR at v is thus given by

$$SINR_v = \frac{P_v}{N + I_v}, \tag{3}$$

where N is the background noise power. The probability $Pr[v]$ characterizes that a node v receives a message successfully in a time slot such that

$$Pr[v] = 1 - Ae^{-B \cdot SINR_v}, \tag{4}$$

where A, B are both positive constants dependent on the real environment. Also, if $Pr[v]$ is too small (i.e. smaller than a threshold value p_κ), we regard it as transmission failure. Here p_κ is called the *transmission failure threshold probability*, which manifests the probabilistic nature of the successful broadcast task.

Network Model: Given a set of nodes V and the system parameters A, B, P, N, α, p_κ, we define the *relaxed threshold radius* r_κ as

$$r_\kappa = \sqrt[\alpha]{\frac{PB}{(1+\gamma)N \ln \frac{A}{1-p_\kappa}}}, \tag{5}$$

where $\gamma > 0$ is a constant called the *relaxation factor*. We define the *transmission graph* G_T as $G_T = \big(V, E_T(r_\kappa)\big)$ where $E_T(r_\kappa) = \{(u,v) | \overline{uv} < r_\kappa\}$. Note that the relaxed threshold radius r_κ as well as the edge set $E_T(r_\kappa)$ depend on the relaxation factor γ. We assume that G_T is fully connected by carefully choosing γ. Justifications for this assumption as well as how to choose γ are given in Section V.

A. Problem Formulation

Given a set of nodes V, a source $s \in V$, and system parameters A, B, P, N, α, p_κ, we suppose that the graph G_T (which is only related to the system parameters A, B, P, N, α, p_κ) is fully connected by properly selecting γ and every node knows its own location. An *admissible broadcast schedule* can be represented as a collection of the subsets $\{U_1, U_2, \ldots\}$ satisfying the following requirements: (1) for all i, $U_i \subset V$ represents the set of nodes scheduled to transmit in time slot i; (2) a node cannot be scheduled to transmit unless it has already received successfully from a neighboring node in G_T in an earlier time slot; (3) at the end, all nodes in V receive the broadcasted message successfully at least once. The *latency* of an admissible broadcast schedule is the first time slot for (3) holds. Obviously, there will be different latencies when different admissible broadcast schedules are employed. The objective of the *minimum-latency broadcast scheduling* (MLBS) problem is *to find an admissible broadcast schedule minimizing its latency.*

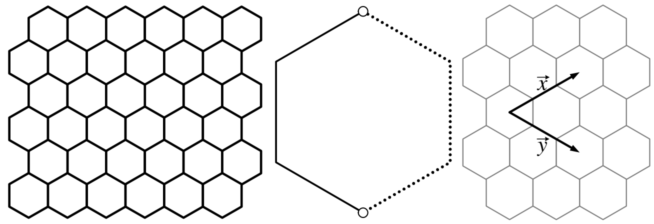
In order to facilitate the problem clearly, now we introduce the important terms, concepts, and methods that will be used extensively throughout this paper.

B. Underlying Terms, Concepts, and Methods

Maximal Independent Sets (MIS): A subset $S \subset V$ is an *independent set* of G if the nodes in S are pairwise non-adjacent, and a *maximal independent set* (MIS) S of G is an independent set of G while no proper superset containing S is an independent set of G. Any node ordering $v_1, v_2, \cdots, v_n$ of V induces an MIS S in the following first-fit manner. Initially, $S = \{v_1\}$. For $i = 2$ up to $i = n$, add v_i to S if v_i is not adjacent to any node in U. Details of MIS can be found in [6].

Hexagonal Tessellation and Colorings: A tessellation of the plane is a way of partitioning it into identical (or similar) pieces. A hexagonal tessellation is partitioning the entire plane into hexagons, as shown in Figure 1 (a). Each hexagon is half open, half closed, without both the topmost and the bottommost points, as shown in Figure 1 (b). We can color this tessellation in various ways. Without loss of generality, in this paper, we will choose the following coloring method.

Method 1: (**Coloring Method**) We introduce a new coloring method here for the future use in broadcast scheduling. Given a hexagonal tessellation and a natural number k, let r denote the radius of a hexagon. Define the vectors $\vec{x} = (3\sqrt{3}r/2, 3r/2)$ and $\vec{y} = (3\sqrt{3}r/2, -3r/2)$ as shown in Figure 1 (c). The lengths of $\vec{x}$ and $\vec{y}$ are both $3r$. Repeat the following process for all $1 \le i \le 3k^2$. Randomly pick an uncolored hexagon whose center is located at $\vec{h}$. Color all the hexagons with color i whose centers are located at $\vec{h} + ak\vec{x} + bk\vec{y}$ for some $a, b \in \mathbb{Z}$.



(a) Hexagonal tessellation (b) One hexagon (c) $\vec{x}$ and $\vec{y}$

Fig. 1. Illustration of hexagonal tessellation and coloring using $\vec{x}$ and $\vec{y}$.

An Example of Method 1: Take $k = 3$ for example. Suppose that we randomly pick up a hexagon H_0 and color it as $i = 1$. According to our coloring method, we should color the hexagons whose centers are located at $\vec{h} + 3a\vec{x} + 3b\vec{y}$ for all $a, b \in \mathbb{Z}$. We repeat this coloring task until $i = 3k^2 = 27$, by which we can color all hexagons.

Lemma 2.1: Method 1 results in a $3k^2$-coloring. Hexagons attributed by the same color are separated by at least $(3k-2)r$.

Note that the procedure of Method 1 is not unique. There are still many different ways to color these hexagons, and we may just consider one of them without loss of generality. For more details, see [7].

III. Novel Randomized Broadcast Scheduling Algorithm

In order to combat the minimum-latency broadcast scheduling problem and to provide a low-latency solution, we propose a novel reconfigurable and channel-aware randomized broadcast scheduling algorithm here. Our proposed algorithm involves two phases, namely (1) *virtual backbone tree construction* and (2) *broadcast scheduling*. Phase (1) is described as follows. Given the network parameters stated in the previous section, we look at G_T's Breadth First Search (BFS) tree (refer to [8] for details), and then divide V into layers $L_0, L_1, L_2, \dots, L_R$ where R is the radius of G_T associated with source s. We sort all nodes in V according to their layers in an ascending order. Let $BLACK$ denote the MIS of G_T induced by such a node ordering. The nodes in $BLACK$ are referred to as the *black nodes*, or the *dominators*, since $BLACK$ is also a dominating set of G_T. The nodes not belonged to $BLACK$ are called *white nodes*.

Now we can construct the virtual backbone tree. We do the followings for all i from 1 to $R-1$. For all black nodes v at layer $i+1$, find its parent $p(v)$ in G_T's BFS tree. Color $p(v)$ as blue. Find $p(v)$'s dominator $d_{p(v)}$ at either layer i or layer $i-1$. Finally we connect $p(v)$, v and $d_{p(v)}$, $p(v)$.

When the above loop finishes, we do the followings for all the remaining white nodes u. Find u's dominator d_u and connect u, d_u. Finally the virtual backbone tree is therefore constructed. Note that the layers of the BFS tree and those of the virtual backbone tree may be different.

Define

$$\begin{cases} r_1 = \sqrt[\alpha]{\frac{8P}{\gamma N}\left(\frac{2}{\alpha-2} + \frac{1}{\alpha-1} + 3\right)} \\ r_2 = \max\left(2r_\kappa,\ \sqrt[\alpha]{\frac{24P}{\gamma N}\left(\frac{2}{\alpha-2} + \frac{1}{\alpha-1} + 3\right)}\right) \\ \Pi_1 = 3\left\lceil \frac{2}{3}\left(\frac{r_1}{r_\kappa} + 2\right)\right\rceil^2, \quad \Pi_2 = 3\left\lceil \frac{2}{3}\left(\frac{r_2}{r_\kappa} + 2\right)\right\rceil^2 \\ \Pi = \Pi_1 + \Pi_2 \end{cases} \tag{6}$$

We can tessellate the plane into half-open half-closed hexagons of radius $\frac{r_\kappa}{2}$, and apply Method 1 to carry out a Π_1-coloring with $k = \left\lceil \frac{2}{3}\left(\frac{r_1}{r_\kappa} + 2\right)\right\rceil$. We use C_1 to denote this coloring ($C_1 : R^2 \to N$, where the coloring C_1 transforms a hexagon into an integer color index). Then, we apply Method 1 again to carry out another Π_2-coloring with $k = \left\lceil \frac{2}{3}\left(\frac{r_2}{r_\kappa} + 2\right)\right\rceil$. We use C_2 to denote this coloring.

Consequently, we can undertake Phase (2) in our proposed scheme now. The broadcast scheduling algorithm based on the constructed virtual backbone tree is described in Algorithm 1.

Algorithm 1 Randomized Broadcast Scheduling

- **repeat** the following $\left\lceil \frac{\ln(n/\epsilon)}{\ln(1/(1-p_\kappa^2))} \right\rceil$ times in parallel for each node v that either is the source or has just successfully received the message.
  - **if** v is black **then**
    - Wait until $Time \mod \Pi \equiv 0$.
    - Schedule v to transmit to all of its child(ren) at $Time \mod \Pi \equiv C_1(v)$.
  - **end if**
  - **if** v is blue **then**
    - Wait until $Time \mod \Pi \equiv \Pi_1$.
    - For each black child w of v, v transmits to w in the virtual backbone tree at $Time \mod \Pi \equiv \Pi_1 + C_2(w)$.
  - **end if**
- **until** done

The latency associated with our proposed algorithm can be evaluated using the following new theorem.

Theorem 3.1: In Algorithm 1 (we refer as Alg. 1 in brief), the probability that all nodes have successfully received the message by time

$$\frac{\Pi}{p_\kappa^2}\left[R + \ln(n/\epsilon) + \sqrt{2R\ln(n/\epsilon) + \ln^2(n/\epsilon)}\right] \tag{7}$$

is at least $1 - 2\epsilon$.

Theorem 3.1 establishes an explicit relationship between the tolerated transmission-failure probability 2ϵ and the latency of the corresponding broadcast schedule we introduce in this section. The tolerated transmission-failure probability is calculated in the strict sense that even a message transmission-failure at any single node will cause the whole broadcast failure. Only if all nodes have successfully received the message, do we call it a success. Theorem 3.1 tells us that this probability is at least $1 - 2\epsilon$. The complete proof of this new theorem is omitted and shown in our forthcoming paper instead.

IV. Illustration of Our Proposed Algorithm

We present an example here to illustrate the detailed procedure of our proposed algorithm in Section III. Suppose that V, $s \in V$, and system parameters $A, B, P, N, \alpha, p_\kappa$ are all given and we have already chosen γ properly such that G_T is fully connected as depicted in Figure 2 (a). According to our algorithm, we need to construct the virtual backbone tree first. We start by constructing an MIS. In the BFS tree of G_T, the number on each node represents its layer. Then, we sort all nodes according to their layers in an ascending order. Therefore, we start from layer L_0, which contains s only. We add s to $BLACK$ and move on to layer 1. Since all nodes at layer 1 are adjacent to s, none of them can be added and layer 1 is done. Likewise, now we work on layer 2. In a similar manner, we then work on layer 3. The MIS thus contains seven black nodes as depicted in Figure 2 (b). Those nodes which are not labeled black are white. The details cannot be manifested in graphics due to the figure limitation.

Based on the MIS, we may embark on constructing the virtual backbone tree. We start from layer 2 since layer 1 does not have any black node. For each black node at layer 2, we find its parent node at layer 1 in the BFS tree, color it as blue, and connect them. For each blue node, we find a black node in either L_1 or L_0 (in this case L_0) in the BFS tree. We repeat this process at layer 3, find the corresponding blue nodes at layer 2 in the BFS tree, and connect them. We repeat this procedure until all layers in the BFS tree have been visited (only up to layer 3 in this example). Finally we connect the remaining white nodes. Ultimately, the virtual backbone tree is thus constructed, as depicted in Figure 2 (b). The details cannot be manifested in graphics due to the figure limitation.

Now, according to the definitions of Π_1 and Π_2 in Section III, we calculate

$$\left\lceil \frac{2}{3}\left(\frac{r_1}{r_\kappa} + 2\right)\right\rceil = 4, \quad \left\lceil \frac{2}{3}\left(\frac{r_2}{r_\kappa} + 2\right)\right\rceil = 5.$$

Therefore, $\Pi_1 = 48$, $\Pi_2 = 75$, and $\Pi = 123$. The colorings C_1, C_2 are shown in Figure 2 (b) (c) and Figure 3 (a) (b), respectively. In these figures, Figure 2 (b) and Figure 3 (a) show the positions of the nodes while Figure 2 (c) and Figure 3 (b) show the overall colorings. We do not show the nodes in part (b) simply to maintain the legibility. Note that C_1 and C_2 are constructed independently and their colors have nothing to do with each other. Take s for example, $C_1(s) = 30$ while $C_2(s) = 2$. Now, according to Alg. 1, we group 123 time slots altogether as a unit and all black nodes (that have successfully received the message) are scheduled to transmit according to their C_1-colors. In this example, there are 7 black nodes with C_1-colors 14, 15, 24, 29, 30, 35, 36. Therefore, they should transmit in these time slots (colors) repeatedly for every period consisting of 123 time slots. Blue nodes are scheduled to transmit according to their black child(ren)'s C_2-color(s). In this example, there are 5 blue nodes. Take node v for example; v has two black children with C_2-colors 3 and 66, respectively; v should therefore transmit its successfully received message in the 51^{st} and 114^{th} time slots (3+48 and 66+48) repeatedly for every period consisting of 123 time slots. Each black node and blue node should start the transmission once it has successfully received the message, and repeat such transmissions for $\lceil \frac{\ln(n/\epsilon)}{\ln(1/(1-p_\kappa^2))} \rceil$ times.

V. Appropriate Selection of the Relaxation Factor γ

As discussed in Section II, the relaxation factor γ plays an important role in our assumption of conditionally-full connection for any wireless network. We assume that we can always make G_T fully connected by choosing γ appropriately. Here we will explain why this assumption actually makes sense in the MLBS problem. Let us first revisit Eq. (4) in Section II and find the minimum SINR to make the probability of successful reception exceed the threshold probability. In other words, find $SINR_v$ such that $Pr[v] \geq p_\kappa$. According to Eq. (4), it yields

$$1 - Ae^{-B \cdot SINR_v} \geq p_\kappa. \tag{8}$$

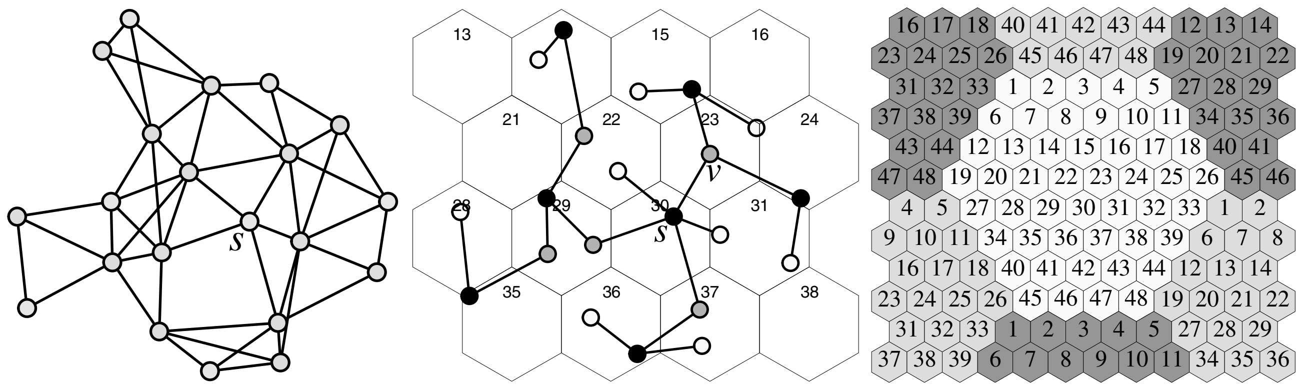
Therefore, $e^{-B \cdot SINR_v} \leq \frac{1-p_\kappa}{A}$, and $SINR_v \geq \frac{1}{B} \ln \frac{1-p_\kappa}{A}$. Since both P and N are assumed to be constants, if there is no interference involved at all, $SINR_v$ only depends on the transmission distance r from the transmitting node to the receiving node. From Eq. (8), we have

$$SINR_v = \frac{P}{r^\alpha N} \geq \frac{1}{B} \ln \frac{1-p_\kappa}{A}, \text{ and } r \geq \sqrt[\alpha]{\frac{PB}{N \ln \frac{A}{1-p_\kappa}}}. \tag{9}$$

Thereby, we define the *threshold radius* $r_{\kappa 0}$ as

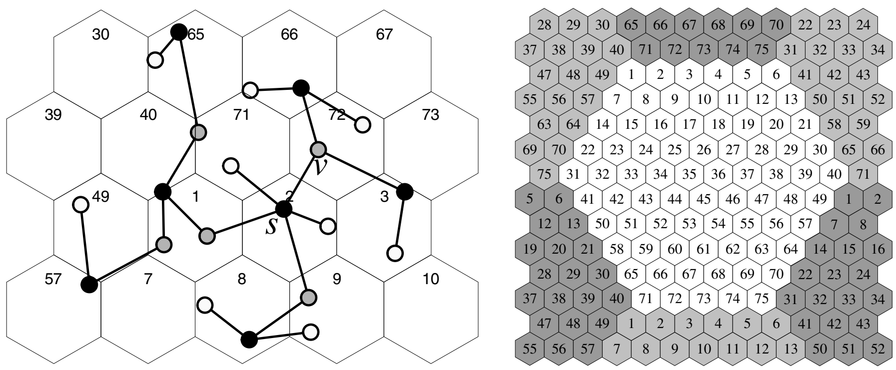
$$r_{\kappa 0} = \sqrt[\alpha]{\frac{PB}{N \ln \frac{A}{1-p_\kappa}}}. \tag{10}$$

According to Eqs. (9) and (10), in order to make the reception successful, the transmission distance must be less than or equal to $r_{\kappa 0}$. Note that Eq. (10) is derived upon when there is no interference. In the MLBS problem, it means that no concurrent transmission is allowed. Therefore, it becomes a trivial problem. In order to make this problem non-trivial, we must accommodate the concurrent transmissions to some extent by *relaxing* the threshold radius a little more. We define the relaxed threshold radius as *the maximum radius that makes $P_r[v]$ greater than the threshold probability, provided that the overall interference is γN*. In other words, we can tolerate up



(a) Topology of G_T (b) C_1: 48-coloring (c) ($\Pi_1 = 48$)

Fig. 2. Illustration of our proposed method for a topology with 48-coloring.



(a) (b)

Fig. 3. C_2: 75-coloring ($\Pi_2 = 75$). Actual nodes are shown in (a) only and the overall coloring is shown in (b).

to γN interference totally and still guarantee our scheduling algorithm's effectiveness. The above reasons lead us to define the relaxed threshold radius in Eq. (5). The assumption that G_T is conditionally-fully connected is actually very reasonable for the following reasons. On one hand, if G_T is not fully connected for any $\gamma > 0$, then *no algorithm that allows concurrent transmissions can have an admissible broadcast schedule.* On the other hand, if G_T is not fully connected, the MLBS problem still makes sense on the *connected subgraph containing the source node* and our algorithm can still work.

Here we present how to choose γ appropriately. We choose γ subject to the following criteria: (1) G_T is fully connected; (2) the overall latency is minimized.

Theorem 3.1 tells us that the latency is of order $O(\Pi R)$, in which $\Pi = O\big((\frac{r_1}{r_\kappa}+2)^2 + (\frac{r_2}{r_\kappa}+2)^2\big)$. Moreover, $r_1 = O(\gamma^{-\frac{1}{\alpha}})$, $r_2 = O(\gamma^{-\frac{1}{\alpha}})$, and $r_\kappa = O\big((1+\gamma)^{-\frac{1}{\alpha}}\big)$. Note that R may be influenced by γ as well. Although there is no explicit relationship between them, generally speaking, R is proportional to $\frac{1}{r_\kappa}$ if nodes are distributed evenly, and the latency is therefore $O\big((1+\gamma)^{\frac{1}{\alpha}}\big)$. Consequently,

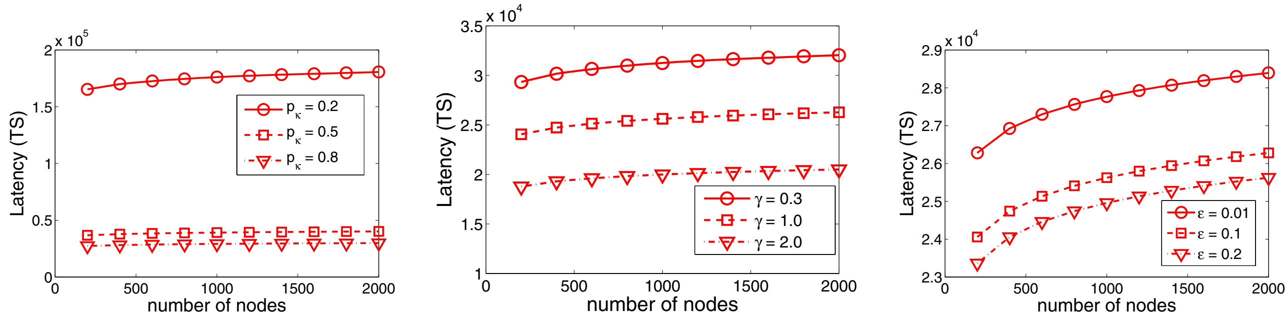
$$\Pi = O\left(\left(\frac{1+\gamma}{\gamma}\right)^{\frac{1}{\alpha}}\right)(1+\gamma)^{\frac{1}{\alpha}} = O\left(\left(1+\frac{1}{\gamma}\right)^{\frac{1}{\alpha}}(1+\gamma)^{\frac{1}{\alpha}}\right). \tag{11}$$

We can see that the latency tends to infinity when γ tends to either 0 or ∞. The minimum latency value can therefore be determined according to elementary calculus as follows. First we determine the range along the real line such that G_T is fully connected in this range. We then express the latency as a function of γ and seek its minimum within this range.

VI. Numerical Evaluation of the Proposed Scheme

In this section, we show the numerical results according to our latency formula presented in Sections III and V. We also demonstrate the relation between the broadcasting latency and the system parameters A, B, p_κ and ϵ. In these numerical evaluations, we fixed $\frac{P}{N} = 5$ and $\gamma = 0.1$ (except for Figure 4 (b)). We listed crucial parameters in Table I.

In Figure 4(a), we compared the transmission latency for different p_κ, in which the number of nodes in the network ranges from 200 to 2000 ($A = 0.6$, $B = 0.5$, $\gamma = 0.1$, $\epsilon = 0.1$). When p_κ is higher, the latency is relatively lower. For example,



(a) Transmission latency for different p_κ (b) Transmission latency for different γ (c) Transmission latency for different ϵ

Fig. 4. Transmission latency versus its system parameters.

parameters	description
A	system parameter defined in Eq. (4)
B	system parameter defined in Eq. (4)
p_κ	transmission failure threshold probability
ϵ	half of the failure probability defined in Theorem 3.1

TABLE I
CRUCIAL SYSTEM PARAMETERS

when p_κ=0.5, the transmission latency is about 3×10^4, and when p_κ=0.8, the transmission latency is about 2×10^4.

Figure 4(b) illustrates the relation between the latency and the relaxation factor γ. The number of nodes still ranges from 200 to 2000 ($A = 0.6$, $B = 0.5$, $p_\kappa = 0.7$, $\epsilon = 0.1$). Note that the latency decreases as γ increases. The reason for this is that the number of nodes covered by a broadcasting action becomes larger if the value of γ increases. Hence, the broadcast latency is getting smaller when the value of γ increases.

Figure 4(c) illustrates the relation between the latency and the maximum tolerable broadcast failure ratio ϵ. The number of nodes ranges from 200 to 2000 ($A = 0.6$, $B = 0.5$, $p_\kappa = 0.5$, $\gamma = 0.1$). Note that ϵ is evaluated in a very strict manner that even if a single node fails to receive the message, the whole broadcast is considered failed. As in Figure 4(c), the lower ϵ is, the larger the transmission latency. The increase of ϵ means the probability of successful reception for each node decreases. If a node fails to receive the broadcast message, it cannot relay the broadcast later. Moreover, none of its children can relay it later either. Obviously, as a result, the transmission latency is surely smaller if ϵ is larger.

VII. CONCLUSION

In this paper, we study the minimum-latency broadcast scheduling problem in the realistic probabilistic model and establish a new explicit relationship between the tolerated transmission-failure probability and the overall latency of the broadcast schedule. We also design a novel algorithm which can dynamically adjust the latency subject to the condition of the interference channel. Our algorithm and analysis can be deemed as the first attempt to combat the low-latency broadcast problem for the scalable cognitive wireless networks in the interference channels.

ACKNOWLEDGMENT

Scott C.-H. Huang was supported in part by the Research Grant Council of Hong Kong under the project GRF CityU 113807.

REFERENCES

[1] Anders Dessmark and Andrzej Pelc, "Tradeoffs between knowledge and time of communication in geometric radio networks," in *SPAA'01: Proc. of the 13th annual ACM symposium on Parallel algorithms and architectures*, New York, NY, USA, 2001, pp. 59–66, ACM Press.
[2] Rajiv Gandhi, Srinivasan Parthasarathy, and Arunesh Mishra, "Minimizing broadcast latency and redundancy in Ad Hoc networks," in *MobiHoc'03: Proc. of the 4th ACM international symposium on Mobile ad hoc networking and computing*, 2003, pp. 222–232.
[3] Thomas Moscibroda, Roger Wattenhofer, and Aaron Zollinger, "Topology control meets SINR: the scheduling complexity of arbitrary topologies," in *MobiHoc'06: Proc. of the 7th ACM int'l symposium on Mobile ad hoc networking and computing*, New York, NY, USA, 2006, pp. 310–321, ACM Press.
[4] Rong Zheng and Richard Barton, "Toward optimal data aggregation in random wireless sensor networks," in *Infocom'07: Proc. of the 26th IEEE Int'l Conference on Computer Communications*. 2007, pp. 249–257, IEEE.
[5] Piyush Gupta and P R Kumar, "Capacity of wireless networks," *IEEE Transactions on Information Theory*, vol. 46, no. 2, pp. 388–404, 2000.
[6] Douglas B West, *Introduction to Graph Theory, 2nd Ed.*, Prentice Hall, Upper Saddle River, NJ 07458, USA, 2001.
[7] Robert A. Leese, "A unified approach to the assignment of radio channels on a regularhexagonal grid," *IEEE Transactions on Vehicular Technology*, vol. 46, no. 4, pp. 968–980, 1997.
[8] Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein, *Introduction to Algorithms, 2nd Ed.*, MIT Press/McGraw Hill, Cambridge, Massachusetts, USA, 2003.