Probabilistic Protocols for Node Discovery in Ad Hoc Multi-channel Broadcast Networks

G. Alonso[1], E. Kranakis[2], C. Sawchuk[2], R. Wattenhofer[1], and P. Widmayer[1]

[1] Department of Computer Science, Swiss Federal Institute of Technology, ETH Zurich, Switzerland
[2] School of Computer Science, Carleton University, Ottawa, ON, K1S 5B6, Canada

Abstract. Ad hoc networks consist of wireless, self-organizing nodes that can communicate with each other in order to establish decentralized and dynamically changing network topologies. Node discovery is a fundamental procedure in the establishment of an ad hoc network, as a given node needs to discover what other nodes are in its communication range. Existing multi-channel node discovery protocols are typically constrained by the network configuration that will be imposed on the nodes once they are discovered. We present a communication model that is independent of the network configuration that will be established after node discovery. We present a pair of node discovery protocols for $k \geq 2$ nodes in a multi-channel system and analyze them using the given communication model.

1 Introduction

Ad hoc networks consist of wireless, self-organizing nodes that can communicate with each other in order to establish decentralized and dynamically changing network topologies. Since these networks are an integral part of the new wireless solutions sought for home or personal area networks, sensor networks, and various other commercial and educational networks, eliminating the shortcomings of ad hoc networks is an important goal in network research [12].

Before a node can communicate with the other nodes in its communication range, it must be aware of those nodes and thus node discovery is an essential part of the rendezvous layer for any node that engages in ad hoc network formation [16]. Efficient network formation requires that the rendezvous layer be able to find all nodes in communication range in the shortest time and with the smallest energy expenditure possible. Obviously, the complexity of node discovery is a function of both the number of nodes present and the number of communication channels available to these nodes. Until recently, nearly all ad hoc networks were formed by nodes that used single channel technology such as 802.11 or IR LANs and thus most of the research about node discovery in ad hoc networks assumes there is a single broadcast channel [15]. The introduction of Bluetooth [8], however, has boosted interest in node discovery in multi-channel systems with frequency-hopping. Such research is especially important since the

S. Pierre, M. Barbeau, and E. Kranakis (Eds.): ADHOC-NOW 2003, LNCS 2865, pp. 104–115, 2003.

node discovery protocol in the Bluetooth standard [8] does not scale well and is both time and energy intensive [16].

The node discovery protocol in the Bluetooth standard [8] is asymmetric in that it assigns different roles and different frequency-hopping speeds to various nodes. Salonidis et al [14] point out that when two or more Bluetooth users want to form an ad hoc network, they cannot explicitly assign roles. They need a symmetric protocol for node discovery, i.e., one that does not depend on pre-assigned roles for the nodes. Salonidis et al [14][15], Law et al [10], and Siegemund and Rohs [16] have subsequently developed symmetric node discovery protocols for Bluetooth.

Naturally, these protocols are constrained by the configuration requirements of Bluetooth, e.g., scatternets are comprised of connected piconets where the latter contains one master and seven slave nodes. There exist few multi-channel node discovery protocols that are independent of any network configuration. Since the performance of existing multi-channel node discovery protocols is inextricably linked to the resulting network configuration, it is difficult to compare the performance of protocols that execute in different network configurations. In this paper, however, we present a communication model that is independent of any network configuration that may be imposed on nodes once they are discovered. The model is an extension of the work by Alonso et al [1] to the multi-channel case for $k \geq 2$ nodes. We present a pair of node discovery protocols for $k \geq 2$ nodes and analyze them using the multi-channel communication model.

1.1 Multi-channel Communication Model

Consider a collection of $k \geq 2$ nodes and $f \geq 2$ broadcast channels or frequencies. At each point in time, a given node must either talk (T) or listen (L) on one of the f channels. A node cannot talk and listen at the same time. The state of a node is denoted by (S, i) where $S = T$ or $S = L$ and i represents the chosen frequency, $i = 1, \ldots, f$.

A node a hears the broadcast of another node b if, at the given time, nodes a and b choose the same frequency i, node a listens (L) and node b talks (T), and no other node talks on frequency i. If a node other than node b also talks on frequency i, then collision occurs on frequency i and no node listening on that frequency hears a broadcast. (In this model, spatial frequency reuse, like that used in cellular phones, is not possible.) The nodes are unable to distinguish between collision and noise when listening to a given frequency. *Node discovery* occurs when node a hears the broadcast of node b and, in the next step, node b hears the broadcast of node a.

An event E describes the states of the k nodes at a given point in time:

$$E = \begin{pmatrix} S_1 & i_1 \\ S_2 & i_2 \\ \vdots & \vdots \\ S_k & i_k \end{pmatrix} \tag{1}$$

where (S_m, i_m) is the state of the m-th node.

A node discovery protocol dictates how a node should choose its state at each point in time. A run of a given protocol is the sequence of events generated by the node's choices. Let $E \to E'$ denote that event E is immediately followed by event E' in a given run of the protocol. A run terminates when the k nodes have discovered each other. In the two node case, the last two events of the run, $E \to E'$, are 1) in event E, the first node hears the second node talk, and 2) in the last event, E', the second node hears the first node talk. Thus node discovery with two nodes occurs under the events

$$\begin{pmatrix} T\ i \\ L\ i \end{pmatrix} \to \begin{pmatrix} L\ j \\ T\ j \end{pmatrix} \text{ or } \begin{pmatrix} L\ i \\ T\ i \end{pmatrix} \to \begin{pmatrix} T\ j \\ L\ j \end{pmatrix} \tag{2}$$

The relationship between frequencies i and j depends on the protocol's frequency allocation method and is discussed below.

Let a node be represented by the random variable X that assumes the values of (S, i), the possible states of the node. When a node must randomly choose whether to talk or listen, let p denote the probability that the node will talk (T) and let $q = 1 - p$ denote the probability that the node will listen (L). When a node must randomly choose a frequency, let F_i denote the probability that the node will choose frequency i. Thus p_i, the probability that a given node will talk on frequency i, equals $\Pr[X = (T, i)] = pF_i$ and q_i, the probability that a given node will listen on frequency i, equals $\Pr[X = (L, i)] = qF_i$. Since $p + q = 1$ and $\sum_{i=1}^{f} F_i = 1$, then $\sum_{i=1}^{f} p_i + \sum_{i=1}^{f} q_i = 1$.

After certain events, e.g., one node hears the broadcast of another node, a node may have to decide whether to stay with the same frequency i in the next step, i.e., **static** frequency allocation, or to again randomly choose a frequency, i.e., **dynamic** frequency allocation. If the initial contact occurred on a given frequency i, one might argue that frequency i is a natural choice for further communication and thus static frequency allocation should occur. One can also argue, however, that chances for continued contact may be just as good if the next frequency is again randomly chosen and thus dynamic frequency allocation can be used.

We assume that nodes are synchronized so that they start an algorithm at the same time, choose their respective states at the same time, and maintain those states for the same amount of time. While it is unlikely that all the nodes that want to participate in a given session of node discovery will start the node discovery protocol at the same time, Salonidis et al [15] demonstrate that node synchronization can be accomplished in a reasonable amount of time. They show that if the times at which the respective nodes start the protocol are modelled as a carefully chosen Poisson process then, after a first node has started the node discovery protocol, the remaining nodes have start times that are identically and independently distributed according to a truncated exponential distribution. Given this distribution, a timeout value can be estimated and incorporated into the beginning of the node discovery protocol so that node synchronization occurs before the nodes engage in discovery. The size of the timeout is usually small relative to the time required for node discovery.

We also assume that the nodes know the value of k, the number of nodes in the system. If the number of nodes k was unknown, then a node might need to estimate k in the course of a node discovery protocol, but we leave the study of such cases to a later date.

1.2 Our Contribution

As mentioned earlier, the multi-channel communication model just described for $k \geq 2$ nodes is **independent** of any network configuration that might be imposed on the nodes once they are discovered. We present two node discovery protocols for $k \geq 2$ nodes and analyze them using the multi-channel communication model.

In the random protocol **RP**, each node randomly chooses whether to talk or listen and also randomly chooses a channel or frequency. The nodes' respective choices of actions (talk or listen) and frequencies over time can be represented by a string of symbols. If the nodes' respective choices of actions and frequencies in a given time t are such that one node can hear the other node's broadcast, then the subsequence of symbols representing that event is called a success pattern. By analyzing the occurence of these success patterns, we determine that the expected run time of the random protocol **RP** for two nodes is

$$\frac{1 + \sum_{j=1}^{f} p_j q_j}{2\left(\sum_{j=1}^{f} p_j q_j\right)^2}$$

We also analyze another node discovery protocol for the two node case. In the conditional protocol **CP**, a node randomly chooses to talk or listen until 1) the node talks or 2) the node listens *and* hears the other node's broadcast.

If a given node talked at time t, it will listen at time $t+1$ in an attempt to determine if the other node heard its broadcast, while if the given node listened at time t and heard another node's broadcast, it will talk at time $t+1$ in an attempt to answer the other node. The nodes will choose their respective frequencies according to either static or dynamic frequency allocation.

The **CP** protocol has two phases. The first phase ends for a given node when that node either talks or hears the other node talk. The second phase consists of one step and the node's behaviour in that step is determined by whether it talked or listened at the end of phase 1. A single execution of the two phases is called a subrun. If, at the end of a subrun, node discovery has not occurred, then another subrun is executed. The length of a subrun of the **CP** protocol is an identically distributed random variable with a finite mean and the number of subruns in the **CP** protocol is a random variable with non-negative integer values and a finite mean. Since the length of a subrun is independent of the number of subruns for the **CP** protocol, Wald's identity implies that the expected run time of the **CP** protocol for two nodes is the product of the expected length of a subrun and the expected number of subruns.

A node's choice of frequency is random for each step in phase 1, but the frequency choice in the phase 2 (one step) depends on whether static or dynamic frequency allocation is used. With static frequency allocation, the frequency used in phase 2 is the same frequency used in the final step of phase 1, while with dynamic frequency allocation, the frequency for phase 2 is randomly chosen.

The expected run time for the **CP** protocol with two nodes is

$$\frac{2p(1-p)+1}{(2p(1-p))^2(\sum_{i=1}^{f} F_i^2)}$$

with static frequency allocation and

$$\frac{2p(1-p)+1}{(2p(1-p))^2(\sum_{i=1}^{f} F_i^2)^2}$$

with dynamic frequency allocation. The expected run time for the **CP** protocol with two nodes is longer under dynamic frequency allocation, as opposed to static frequency allocation, by a factor of $\phi = 1/\sum_{i=1}^{f} F_i^2$. For example, if there are f equally likely frequencies such that $F_i = F_j$ for all i, j, then the expected run time of the **CP** protocol with two nodes is f times greater under dynamic frequency allocation than under static frequency allocation.

Having analyzed the **RP** and **CP** protocols for the two node case, we turn to the $k \geq 2$ node case. In the random protocol **RP** for $k \geq 2$ nodes, each node again decides at random whether to talk or listen and also randomly chooses a frequency. The expected run time for the **RP** protocol with $k \geq 2$ nodes is

$$\frac{1+\sum_{j=1}^{f} p_j q_j (1-p_j)^{k-2}}{2\binom{k}{2}\left(\sum_{j=1}^{f} p_j q_j (1-p_j)^{k-2}\right)^2}$$

Unfortunately, calculating the expected run time of the **CP** protocol for $k > 2$ nodes is not as straightforward. At any time $t > 0$ in the **CP** protocol, a given node can be both in phase 1 relative to one subset of nodes and in phase 2 relative to another subset of nodes. Tracking the potential overlap of phases across the nodes becomes more complicated as the number of nodes increases. Our analysis of the expected run time for the **CP** protocol with $k \geq 2$ nodes, therefore, relies on simulation methods rather than a closed-form solution.

1.3 Outline of the Paper

In section 2, we present and analyze the random protocol **RP** and the conditional protocol **CP** for the two node case. In section 3, we present and analyze the $k \geq 2$ nodes case for the **RP** and **CP** protocols. The paper ends in section 4 with some summary remarks and a brief description of open problems. Due to space limitations, only outlines of the proofs are given.

2 Random Protocol for Two Node Multi-channel System

In the random protocol **RP**, each node decides at random whether to talk (T) or listen (L). The two nodes thus generate an event at each time t

$$E = \begin{pmatrix} S & i \\ S' & i' \end{pmatrix}$$

such that S and S' are either T or L, and i and i' are the frequencies chosen.

We use the technique described in [3,11] to analyze the **RP** protocol. The random protocol **RP** succeeds when, for some $i, j = 1, 2, \ldots, f$, either

$$\begin{pmatrix} T & i \\ L & i \end{pmatrix} \rightarrow \begin{pmatrix} L & j \\ T & j \end{pmatrix} \text{ or } \begin{pmatrix} L & i \\ T & i \end{pmatrix} \rightarrow \begin{pmatrix} T & j \\ L & j \end{pmatrix} \tag{3}$$

Define the events A_i and B_i as follows:

$$A_i = \begin{pmatrix} T & i \\ L & i \end{pmatrix},\ B_i = \begin{pmatrix} L & i \\ T & i \end{pmatrix} \tag{4}$$

A success pattern is a pair of events such that the two nodes discover each other, e.g., A_iB_j, $i, j \in 1, \ldots, f$, and thus there are $2f^2$ success patterns:

$$A_1B_1, \ldots, A_1B_f, \ldots, A_fB_1, \ldots, A_fB_f, B_1A_1, \ldots, B_1A_f, \ldots, B_fA_1, \ldots, B_fA_f.$$

For each $i, j \in \{1, 2, \ldots, f\}$, the pattern A_iB_j (respectively B_iA_j) may either overlap itself, or the last event of A_iB_j (respectively B_iA_j) may overlap with the first event of B_jA_k (respectively A_jB_k) for $k \in \{1, 2, \ldots, f\}$. The former case occurs with probability $\frac{1}{p_iq_ip_jq_j}$ while the latter case occurs with probability $\frac{1}{p_jq_j}$.

The resulting system of $2f^2$ linear equations is

$$\left[\begin{array}{c|c} D & U \\ \hline U & D \end{array}\right] \begin{bmatrix} \Pi \\ \Pi \end{bmatrix} = \begin{bmatrix} E[N] \\ E[N] \\ \vdots \\ E[N] \\ E[N] \\ E[N] \\ \vdots \\ E[N] \end{bmatrix} \tag{5}$$

where

$$\Pi' = [\pi_{1,1}, \ldots, \pi_{1,f}, \pi_{2,1}, \ldots, \pi_{f,1}, \ldots, \pi_{f,f}]$$

and $\pi_{i,j}$ (respectively $\pi_{i+f,j}$) is the probability that A_iB_j (respectively B_i, A_j) occurs before any other pattern. N is the run time for exactly one subrun of RP. However, because RP always executes exactly one subrun, N is also the run time for the entire protocol.

The $2f^2$ x $2f^2$ matrix $\left[\begin{array}{c|c} D & U \\ \hline U & D \end{array}\right]$ is defined as follows:

- D is a diagonal f^2 x f^2 matrix with the $((i,j),(i,j))$-th entry equal to $\frac{1}{p_i^2 q_i^2}$.
- U is an f^2 x f^2 matrix formed by a column of f matrices, i.e., $U = [V, V, \ldots, V]^T$ where V is defined as

$$V = \begin{bmatrix} \frac{1}{p_1q_1} & \frac{1}{p_2q_2} & \cdots & \frac{1}{p_fq_f} & 0 & 0 & \cdots & 0 & \cdots & 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & \frac{1}{p_1q_1} & \frac{1}{p_2q_2} & \cdots & \frac{1}{p_fq_f} & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & \cdots & \frac{1}{p_1q_1} & \frac{1}{p_2q_2} & \cdots & \frac{1}{p_fq_f} \end{bmatrix}$$

With the condition

$$\sum_{i=1}^{i=f}\sum_{j=1}^{j=f} \pi_{ij} = 1 \tag{6}$$

the resulting system of linear equations has $2f^2+1$ unknowns and $2f^2+1$ equations. Solving this system of equations gives us $E[N]$, the expected runtime of **RP**.

Theorem 1 (RP). *The expected run time for the* **RP** *protocol is:*

$$\frac{1+\sum_{j=1}^{f} p_jq_j}{2\left(\sum_{j=1}^{f} p_jq_j\right)^2}. \tag{7}$$

3 Conditional Protocol for Two Node Multi-channel Systems

The conditional protocol **CP** is implemented as a series of two-phase subruns where phase 1 consists of a finite number of random steps and phase 2 consists of a single step.

In phase 1 of a subrun of the **CP** protocol, a node follows the random protocol **RP** until 1) the node talks (T) or 2) the node listens (L) *and* hears the other node's broadcast.

Phase 2 of a subrun of the **CP** protocol consists of a single step. The behaviour of a node in phase 2 is conditional on the way in which phase 1 ended. If a node talked (T) at the end of phase 1, then it will listen (L) in the phase 2 in an attempt to determine if the other node heard its broadcast. If a node listened (L) and heard the other node's broadcast at the end of phase 1, then it will talk (T) in phase 2 in an attempt to answer the other node's broadcast.

If a subrun is successful, then node discovery occurs in phase 2 and the **CP** protocol terminates. If a subrun is unsuccessful, however, then another subrun is executed, i.e., phase 1 and phase 2 are repeated, until node discovery occurs.

Let the probability of success in a subrun of the **CP** protocol be denoted by $Pr[\text{success in subrun}]$. Since the subruns of the **CP** protocol are independent trials, the number of subruns of the **CP** protocol is a geometric random variable with parameter Pr[success in subrun]. The expected number of subruns of the **CP** protocol is therefore

$$E[\text{number of subruns}] = \sum_{k=1}^{\infty} Pr[\text{failure in subrun}]^{k-1} Pr[\text{success in subrun}]k \tag{8}$$

3.1 Wald's Identity

If, for the **CP** protocol, the expected number of subruns and the expected length of a subrun are known, then Wald's identity can be used to calculate the expected run time of the protocol.

Wald's identity can be stated as follows [13]. Let $W_i, i \geq 1$ be independent and identically distributed random variables with a finite mean, $E[W] < \infty$. Let N be a stopping time for $W_1, W_2, \ldots$ such that $E[N] < \infty$, i.e., the event $N = n$ is independent of $W_{n+1}, W_{n+2}, \ldots$, for all $n \geq 1$. Then

$$E\left[\sum_{i=1}^{N} W_i\right] = E[W]E[N]. \tag{9}$$

To apply Wald's identity to the present problem, let W_i be the length of a subrun of the **CP** protocol and let N be the number of subruns for the protocol. Defined in this manner, the W_i are identically distributed random variables with a finite mean and N is a random variable with non-negative integer values and a finite mean. The length of a subrun is independent of the number of subruns for the **CP** protocol, so W_i is independent of N. Wald's identity thus implies that the expected run time for the **CP** protocol is the product of the expected length of a subrun and the expected number of subruns, i.e.,

$$E[\text{run time for } \mathbf{CP} \text{ protocol}] = E[\text{length of subrun}]E[\text{number of subruns}]. \tag{10}$$

If we calculate the expected length of a subrun and the expected number of subruns for the **CP** protocol, then equation 10 allows us to calculate the expected run time for the protocol.

3.2 Expected Length of a Subrun of CP Protocol

Phase 1 of the **CP** protocol ends when a node talks or when a node listens and hears the broadcast of the other node. Phase 1 therefore ends when at least one of the nodes talks so the only event that does not bring an end to phase 1 is the event where both nodes listen. Therefore Pr[phase 1 ends] equals

$$1 - Pr[\text{both nodes listen}] = 1 - (1-p)^2 = 2p - p^2$$

This implies that the expected length of phase 1 in the **CP** protocol is

$$E[\text{length of phase 1}] = \sum_{i=1}^{\infty} (2p - p^2)(1 - (2p - p^2))^{i-1} i = \frac{1}{2p - p^2} \tag{11}$$

and therefore, because phase 2 has only one step, the expected length of a subrun is:

$$E[\text{length of subrun}] = \frac{1}{2p - p^2} + 1 \tag{12}$$

As mentioned earlier, the **CP** protocol allows for static or dynamic frequency allocation. With static frequency allocation, a node that uses a frequency i at the end of phase 1 will use the same frequency i in the single step that makes up phase 2. With dynamic frequency allocation, a node randomly chooses a frequency in all steps of either phase.

Substituting the appropriate expressions into equation 10, we obtain the following results.

Theorem 2. *The expected run time for the* **CP** *protocol with static frequency allocation is*

$$\frac{2p(1-p)+1}{(2p(1-p))^2 \sum_{i=1}^{f} F_i^2}. \tag{13}$$

Theorem 3. *The expected run time for the* **CP** *protocol with dynamic frequency allocation is*

$$\frac{2p(1-p)+1}{(2p(1-p)\sum_{i=1}^{f} F_i^2)^2}. \tag{14}$$

The expected run time for the **CP** protocol with two nodes is longer under dynamic frequency allocation, as opposed to static frequency allocation, by a factor of $\phi = 1/\sum_{i=1}^{f} F_i^2$. With a uniform probability distribution for the frequencies, $\phi = f$.

3.3 Comparison of Two Node Protocols

To make a simple comparison of the two node protocols, assume that the probability of talking equals the probability of listening, and that the f frequency choices are uniformly distributed. The **CP** protocol with static frequency yields the best expected run time, $E[\text{run time}] = 6f$, followed by the **CP** protocol with dynamic frequency with an expected run time of $E[\text{run time}] = 6f^2$. The **RP** protocol has the poorest performance, with an expected run time of $E[\text{run time}] = 8f^2 + 2f$.

4 Random Protocol for $k \geq 2$ Node Multi-channel System

In the node discovery problem with $k \geq 2$ nodes, node discovery occurs when **two** of the k nodes discover each other. Consider the event

$$A_i^{ab} := \begin{pmatrix} \vdots & \vdots \\ T & i \\ \vdots & \vdots \\ L & i \\ \vdots & \vdots \end{pmatrix}, \quad \text{respectively, } B_i^{ab} := \begin{pmatrix} \vdots & \vdots \\ L & i \\ \vdots & \vdots \\ T & i \\ \vdots & \vdots \end{pmatrix},$$

such that $a < b$ and

1. the state of the ath node is (T, i), (respectively, (L, i)),
2. the state of the bth node is (L, i), (respectively, (T, i)), and
3. for all $c \neq a, b$, the node c is either talking at a frequency other than i, or listening at any frequency, i.e., no other node talks at frequency i.

Clearly, there are $2f\binom{k}{2}$ such events A_i^{ab}, B_i^{ab}, where $i = 1, 2, \ldots, f$, $a < b$, and $a, b = 1, 2, \ldots, k$. Let the random variable X be defined as the current state of the $k \geq 2$ node system. Suppose that the current state of the system is described by the event A_i^{ab}, i.e., $X = A_i^{ab}$. Given the definition of A_i^{ab}, the event $X = A_i^{ab}$ is the intersection of the three events listed above and thus

$$\Pr[X = A_i^{ab}] = p_i q_i \prod_{c \neq a,b} \left(\sum_{j=1}^{f} q_j + \sum_{j=1, j \neq i}^{f} p_j \right) = p_i q_i (1 - p_i)^{k-2}. \tag{15}$$

Similarly, $\Pr[X = B_i^{ab}] = p_i q_i (1 - p_i)^{k-2}$.

In the random protocol for $k \geq 2$ nodes, node discovery occurs if for some frequencies $i, j \in \{1, 2, \ldots, f\}$ and two nodes $a < b$, event A_i^{ab} (respectively, B_i^{ab}) is followed by event B_j^{ab} (respectively, A_j^{ab}). It follows that the success patterns are $A_i^{ab} B_1^{ab}, A_i^{ab} B_2^{ab}, \ldots, A_i^{ab} B_f^{ab}$, and $B_i^{ab} A_1^{ab}, B_i^{ab} A_2^{ab}, \ldots, B_i^{ab} A_f^{ab}$. As in the two-node case, we calculate the expected runtime of the **RP** protocol by solving a system of linear equations.

Theorem 4. *The expected run time of the* **RP** *protocol for* $k \geq 2$ *nodes is*

$$\frac{1 + \sum_{j=1}^{f} p_j q_j (1 - p_j)^{k-2}}{2\binom{k}{2} \left(\sum_{j=1}^{f} p_j q_j (1 - p_j)^{k-2} \right)^2}. \tag{16}$$

5 Conditional Protocol for $k \geq 2$ Node Multi-channel System

Like the two node case, the conditional protocol **CP** for $k \geq 2$ nodes is implemented as a series of two-phase subruns. In phase 1 of the protocol, a node follows the random protocol **RP** until the node talks (T), or it listens (L) and hears the broadcast of another node.

Phase 2 of a subrun of the **CP** protocol is still a single step and the behaviour of a node in this phase is conditional on the way in which phase 1 ended, i.e., a node that talked (T) at the end of phase 1 will listen (L) in phase 1 while a node that listened at the end of phase 1 will talk (T) in phase 2.

If a subrun of the protocol is successful, node discovery occurs, i.e., **two** of the k nodes discover each other and the protocol terminates. Otherwise, another subrun of the protocol is executed.

While calculating the expected run time of the **CP** protocol for two nodes was straightforward, the corresponding exercise for the $k \geq 2$ node case is quite complicated. For example, a given node can be in phase 1 relative to one set of nodes yet, at the same time, it can be in phase 2 relative to another set of nodes. Our analysis of the expected run time for the **CP** protocol with $k \geq 2$ nodes, therefore, relies on simulation results.

5.1 Comparison of Multi-node Protocols

Once again we made a simple comparison of the $k \geq 2$ node protocols by assuming that the probability of talking equals the probability of listening, and that the f frequency choices are uniformly distributed. The expected run time for the random protocol **RP** quickly became astronomical compared to the expected run time for either version of the **CP** protocol. The latter run times were estimated through simulation results. For a given number of nodes k in the **CP** protocol with static frequency allocation, and $f < k$ frequencies, the expected run time fell as the number of frequencies increased and f approached k. The expected run time then increased as f continued to increase. The behaviour of the expected run times for the **CP** protocol with dynamic frequency allocation was less predictable although, when $f < k$, the dynamic version of the protocol often outperformed the static version.

6 Conclusion

We presented a new communication model for node discovery and used it to compare the behaviour of random **RP** and conditional **CP** node behaviour protocols. We found closed form solutions for the expected run times of the protocols with the exception of the conditional protocol when $k > 2$ and were able to compare the performances of the protocols. In the future, it would be useful to explore protocols that use an estimate of the number of nodes within communication range, rather than assume that the number of nodes is known. It is unlikely that the exact number of nodes is known and the effects of poor estimation are likely to be significant.

Acknowledgements

Research of E. Kranakis and C. Sawchuk was supported in part by NSERC (Natural Sciences and Engineering Research Council of Canada) and MITACS (Mathematics of Information Technology and Complex Systems) grants. C. Sawchuk was also supported by an Ontario Graduate Scholarship.

References

1. G. Alonso, E. Kranakis, R. Wattenhofer, and P. Widmayer, Probabilistic Protocols for Node Discovery in Ad-hoc Single Broadcast Channel Networks, Workshop on Mobile AdHoc Networks (WMAN), International Parallel and Distributed Processing Symposium (IPDPS 2003), April 22 - 26, 2003, Nice, France.

2. D. Bertsekas and R. Gallager, Data Networks, Prentice Hall, 1992.
3. G. Blom and D. Thoburn, How Many Random Digits Are Required Until Given Sequences Are Obtained, J. Applied Probability, 19, 518-531, 1982.
4. BlueHoc: An Open-Source Simulator, http://oss.software.ibm.com/developerworks/opensource/bluehoc.
5. Ericsson Microelectronics: ROK 101 007 Bluetooth Module Datasheet Rev. PA5, April 2000.
6. R. Garcés, J.J. Garcia-Luna-Aceves, Collision Avoidance and Resolution Multiple Access for Multichannel Wireless Networks, IEEE Infocom 2000, March 26 - 30, 2000, Tel-Aviv, Israel.
7. W. Feller, An Introduction to Probability Theory and its Applications, Vol. II, Wiley, 1966.
8. J. Haartsen, Bluetooth Baseband Specification v. 1.0, www.Bluetooth.com.
9. O. Kasten, M. Langheinrich, First Experiences with Bluetooth in the Smart-Its Distributed Sensor Network, Workshop on Ubiquitous Computing and Communications, PACT 2001, October 2001.
10. C. Law, A.K. Mehta, K.-Y. Siu, Performance of a Bluetooth Scatternet Formation Protocol, The Second ACM Annual Workshop on Mobile Ad Hoc Networking and Computing (mobiHoc 2001), October 4-5, 2001, Long Beach, California, USA.
11. S. R. Li, A Martingale Approach to the Study of Occurrence of Sequence Patterns in Repeated Experiments, Annals of Probability, 8, 1171- 1176, 1980.
12. C.E. Perkins, editor, Ad Hoc Networking, Addison Wesley, 2001.
13. S. Ross, Stochastic Processes, John Wiley and Sons, 2nd edition, 1996.
14. T. Salonidis, P. Bhagwat, L. Tassiulas, Proximity Awareness and Fast Connection Establishment in Bluetooth, The First ACM Annual Workshop on Mobile Ad Hoc Networking and Computing (MobiHoc 2000), August 11, 2000, Boston, Massachusetts, USA.
15. T. Salonidis, P. Bhagwat, L. Tassiulas, R. LaMaire, Distributed Topology Construction of Bluetooth Personal Area Networks, In Proceedings of the Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2001), April 22 - 26, 2001, Anchorage, Alaska, USA.
16. F. Siegemund and M. Rohs, Rendezvous Layer Protocols for Bluetooth-Enabled Smart Devices, Technical Report, 1st International Conference on the Architecture of Computer Systems, ARCS - Trends in Network and Pervasive Computing, 2002.