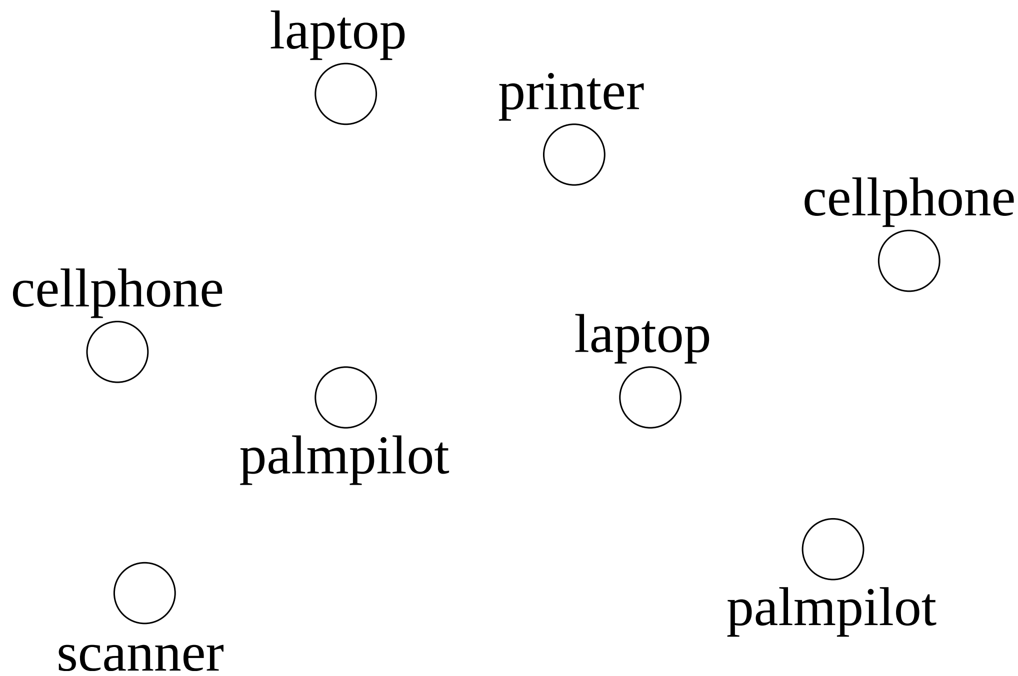
Randomized Protocols for Node Discovery in Ad-Hoc Multichannel Broadcast Networks

By
Evangelos Kranakis, Cindy Sawchuk
(Carleton U, Ottawa)
and
Gustavo Alonzo, Roger Wattenhofer, Peter Widmayer
(ETH, Zürich)

ADHOC-NOW, Montreal, 2003

How to Establish a Link

A set of nodes is to form a connected network.



What protocol should they follow to get connected?

How to Establish a Link

Nodes follow the protocol below:

1. Start
2. Synchronization
3. Discovery
4. Paging
5. Connection established.

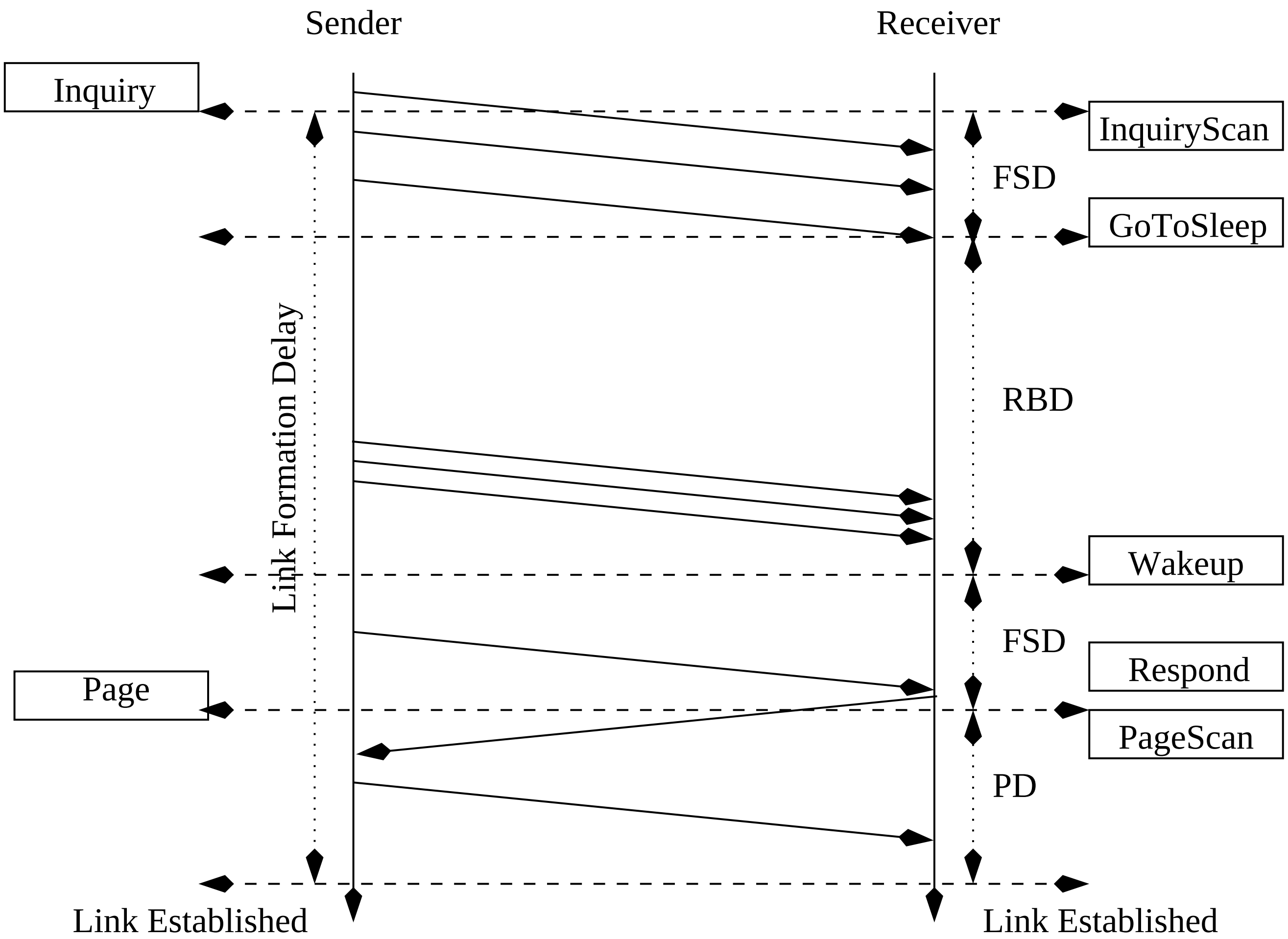
Example: Bluetooth Asymmetric Protocol

This has three main phases.

1. Nodes can use the same frequency but may be out of phase due to internal clock differences. To overcome this problem sender and receiver hop at frequencies of different speeds. This causes some Frequency Synchronization Delay.
2. Next follows the discovery protocol in which nodes enter the Inquiry and/or InquiryScan states.
3. Paging procedure in which the sender uses the receiver's page hopping sequence to initiate a Device Access Code packet that can be heard only by the receiver device.

We are interested in phase 2.

Example: The Bluetooth Asymmetric Protocol



FSD = Frequency Synchronization Delay PD = Paging Delay
IAC = Inquiry Access Code RBD = Random Backed Delay

LF (Link Formation) Delay

This is determined by three random variables.

1. **FS (Frequency Synchronization) Delay:** About 10 ms.
2. **RB (Random Backed) Delay:** About 639.75 ms.
3. **P (Paging) Delay:** Considered negligible since it follows the inquiry state (about 625 μs, i.e., negligible.)

The Bluetooth asymmetric protocol implies the following delay equation

$$\mathrm{LF} = 2 \cdot \mathrm{FS} + \mathrm{RB} + \mathrm{P} \approx 659.75\ ms.$$

It is important to note that item 2 (also called Discovery Delay) is the main cause of delay in the Bluetooth asymmetric protocol.

Discovery Delay Procedure

Bluetooth supports the paradigm of **spontaneous connectivity**. The procedure used for node discovery is called **Inquiry** and connections are established based on information exchange.

1. Bluetooth node is set into **Inquiry** mode by the application.
2. Then sends **Inquiry** messages to probe for other nodes.
3. Other Bluetooth nodes (within the range) only listen.
4. They reply to **Inquiry** messages only when they have been set explicitely to **InquiryScan** mode.

To prevent "collisions" and since **Inquiry** needs to be initiated periodically then some type of randomness must be employed in order to determine the time interval between two **Inquiries**. This technique is called **Collision Avoidance**.

Connection Establishment

- Once a unit has discoverd another unit, connection establishment is very fast.
- In an ideal scenario, the expected delay for link formation (Discovery plus Connection) is about 1 sec when both nodes follow the uniform distribution between the **Inquiry** and **InquiryScan**.
- In practice this takes several seconds.

Questions

- Can we improve on the discovery process?
- Can we propose alternative protocols that have better performance?
- Can we compare the proposed discovery protocols?
- What are the limits of the collision avoidance process in wireless systems?

Notation and Definitions

1. Assume a system with K nodes communicating by broadcasting messages.
2. At any given time a node can be at a given frequency $i = 1, 2, \ldots, f$ either T (Talking) or L (Listening).
3. The state of a node is denoted by the pair (S, i) where $S = T$ or $S = L$ and $i = 1, 2, \ldots, f$.
4. The nodes are synchronized: they change states at the same time and remain in a given state for a period of time that is identical for all nodes.
5. An event E describes the state of the K nodes of the system.

Events

An event E describes the state of the K nodes of the system:

$$E = \begin{pmatrix} S_1 & i_1 \\ S_2 & i_2 \\ \vdots & \vdots \\ S_K & i_K \end{pmatrix},$$

where (S_k, i_k) is the state of the k-th node.

For each event E we denote by k^E the state of the kth node in event E.

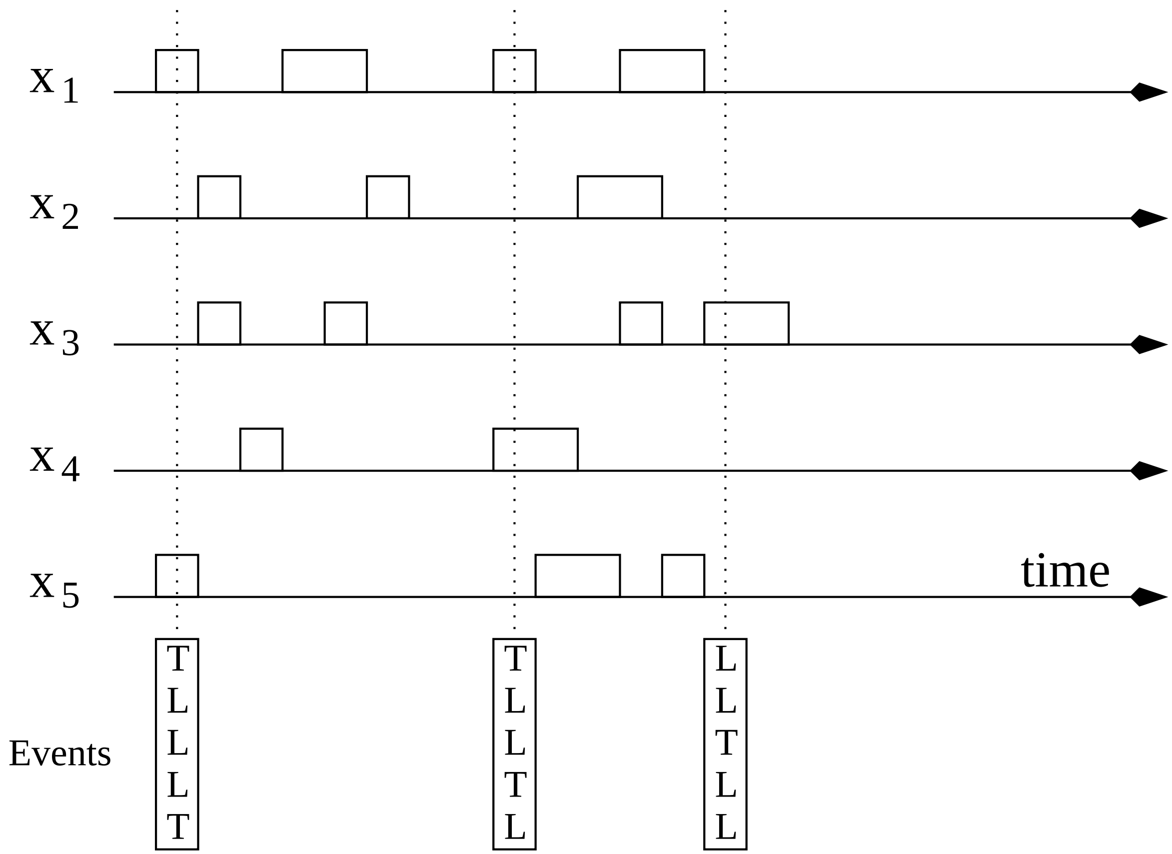
Events

The conditions for a node k to receive a message from another node l are defined as follows.

A node k receives a message from the other node l if, at any given time,

1. nodes k, l are at the same frequency, say i,
2. node k is listening, while node l is talking at frequency i, and
3. all other nodes in the system are either in another frequency or else they are *listening* in frequency i.

A Sequence of Events



A Run of the Protocol

A run of a protocol is a sequence of events.

A run terminates if two nodes have discovered each other. This means that the last two events E, E' satisfy the following property:

1. in event E, node k receives a message from node l, and
2. in event E', node l receives a message from node k.

The decision on whether or not a node will talk or listen as well as on which channel depends on a given deterministic/randomized algorithm and channel allocation depends on a given probability distribution.

Probability distribution of talking/listening in a frequency

A node is represented by a random variable X assuming the values (S, i), where S is either T or L and $i = 1, 2, \ldots, f$.

Associated with this random variable is a probability distribution of the frequencies.

If F_i is the probability that a node is in frequency i, p is the probability that a node is talking, and q is the probability that a node is listening then we have

$$\begin{aligned} p_i &= \Pr[X = (T, i)] = pF_i \\ q_i &= \Pr[X = (L, i)] = qF_i \end{aligned}$$

We assume the frequency allocations are i.i.d. random variables.

Dynamic and static frequency allocation

We analyze two types of node discovery protocols: *dynamic* and *static* frequency allocation.

In either case, the protocol succeeds if two nodes discover each other. The two types of protocols differ in the way they allocate frequencies.

In *static* frequency allocation, the first node talks and the second listens in a given frequency and, in the next step, the second node talks and the first listens in this same frequency.

In *dynamic* frequency allocation, the first node talks and the second listens in a given frequency and, in the next step, the second node talks and the first listens in this same or in a different frequency.

Protocols

By Random Protocol (**RP**), we understand a protocol in which each node decides at random whether to talk or listen.

Thus far, a node has randomly chosen whether to talk or listen, and has also randomly chosen a frequency.

We shall now consider protocols where a node's behaviour is dictated by a simple set of rules.

In the *answering* protocol, **AP**, a node that receives a message will answer, i.e., talk, in the next step.

In the *listening* protocol, **LP**, a node that sent a message, i.e., talked, will listen in the next step.

The *answering* and *listening* protocols can be implemented with either static or dynamic frequency allocation.

Expected Waiting Time in Two Node Systems

	Static	Dynamic
RP:	$\frac{1}{\sum_{i=1}^{f} \frac{2}{\left(\frac{1}{pqF_i^2}+\frac{1}{p^2q^2F_i^4}\right)}}$	$\frac{1+pq\sum_{j=1}^{f} F_j^2}{2p^2q^2\left(\sum_{j=1}^{f} F_j^2\right)^2}$
AP:	$\frac{1+2pq\sum_{i=1}^{f} F_i^2}{2pq^2(\sum_{i=1}^{f} F_i^2)^2}$	$\frac{1+2pq\sum_{i=1}^{f} F_i^2}{2pq^2(\sum_{i=1}^{f} F_i^2)^2}$
LP:	$\frac{1+2pq\sum_{i=1}^{f} F_i^2}{2p^2q(\sum_{i=1}^{f} F_i^2)^2}$	$\frac{1+2pq\sum_{i=1}^{f} F_i^2}{2p^2q(\sum_{i=1}^{f} F_i^2)^2}$

Expected Waiting Time in Multiple Node Systems

Expected waiting time for protocol **RP** with K nodes and static frequency allocation

$$1/\left(2\binom{K}{2}\sum_{i=1}^{f}\frac{1}{\left(\frac{1}{p_i^2q_i^2(1-p_i)^{2K-4}}+\frac{1}{p_iq_i(1-p_i)^{K-2}}\right)}\right)$$

Expected waiting time for protocol **RP** with K nodes and dynamic frequency allocation

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

Random Unicast Protocols in Multiple Node Systems

We analyze the Random Unicast protocol (**RUP**). In this protocol, an attempt is made to select a single node and, if successful, this node then attempts to discover another node in the ad hoc network.

Phase 1: Nodes choose two values in the range $1..K$. **If** no node chooses a repeated value in the first phase, then the **RUP** protocol reverts to **RP**. **Else** it may execute either of the following two protocols.

Listenining Protocol:
Phases 2 and 3: *Selected* node talks for one step. and then, in the next step (the third phase), node listens to hear if contact was made.

Answering Protocol:
Phases 2 and 3: *Selected* node listens until contact is made, and then talks in the next step (the third phase)

RUP Answering/Listening

We have analyzed

- RUP Answering: static/dynamic
- RUP Listening: static/dynamic

Conclusion

- In general, regardless of the accepted standards, it is an interesting problem to explore the limits of the "Collision Avoidance" paradigm,
- It would be interesting to simulate the algorithms proposed and determine optimal performance regions.
- It is possible to give "other" node discovery algorithms but the mathematical analysis is more difficult.