# Formation And Maintenance of Self-Organizing Wireless Networks

Keith Scott
Keith.Scott@jpl.nasa.gov
Jet Propulsion Laboratory
4800 Oak Grove Drive
Pasadena, CA 91109-8099

Nicholas Bambos
bambos@leland.stanford.edu
Department of EES&OR
Stanford University
Stanford, CA

ABSTRACT: *There are numerous military, commercial, and scientific applications for mobile wireless networks which are able to self-organize without recourse to any pre-existing infrastructure. We present the Self-Organizing Wireless Adaptive Network (SWAN) protocol, a distributed networking protocol capable of managing such networks. The SWAN approach is based on* **dynamic topology management with power control**, *allowing it to adapt gradually to the changing enviornment instead of periodically discarding the network topology information and rebuilding the network from scratch. In addition, under SWAN* **control information is distributed** *instead of being concentrated in a "control phase". This provides significant savings when the acquisition times of the modems are high.*

## 1 Introduction

Given a set of mobile users equipped with radio transceivers, we would like to provide a means for them to "self-organize" into a wireless network. The problem is to find an efficient distributed algorithm for sharing the communications resources in an way that provides as much bandwidth as possible for user communications while maintaining a measure of fairness.

In a stationless network, any pair of nodes may be able to communicate, but due to physical restrictions, certain sets of communications may be incompatible. Given a particular traffic matrix and node arrangement, it is, of course, possible to determine if the traffic matrix can be supported and, if so, how. Thus we could envision an approach where each new transmission requires a total rearrangement of the existing network in order to be accommodated. While this is interesting in principle, the overhead required to find a new compatible resource allocation and the latency of switching to it make this method unusable.

Instead, a common approach is to first convert the stationless network into a more manageable form by imposing a restricted topology that defines which nodes are allowed to communicate. This allows us to bring in all of our knowledge and experience with wired networks. For example, once a restricted topology has been imposed, it is no longer the case that all communications can take place in a single hop, and the nodes may exchange routing tables so that packets can be forwarded across the multiple hops that may be needed to reach their destinations.

Thus the first problem faced by a node in such a network is to determine its set of possible communications partners. This is itself a non-trivial task. Some mechanism must exist for two nodes which have no prior knowledge of each others' existence to become neighbors in the network topology (consider an isolated, two-node network just powering on). This method must also protect ongoing transmissions from being unduly degraded, as described below.

While imposing a topology vastly simplifies the task of finding compatible resource allocations, some work still remains to be done. If a pair of nodes is using a particular transmission resource, whether it is a narrowband frequency slot, a time slot, or a spread-spectrum code, that resource will be unavailable to other nearby users. For example, if the nodes in a wireless network are using direct sequence spread spectrum radios to communicate, there is the danger that one transmitter may completely dominate several receivers in his vicinity, preventing them from receiving from any other node in the network, regardless of the spreading code they choose (the near-far problem).

This paper presents the Self-Organizing Wireless Network or SWAN algorithm for managing such a network where the users employ direct sequence spread-spectrum radios for communication. SWAN is responsible for forming and maintaining the network topology described above, and relies on a distributed power control algorithm to ensure that the transmission schedules of the various nodes remain compatible.

The rest of this paper is organized as follows. Section 2 outlines, in broad terms, some existing self-organizing network protocols, and describes the power control algorithm that is key to SWAN. Section 3 describes the SWAN protocol, which can be broken down into two layers. Section 3.1

presents the mechanism by which new links can be added to the network topology, while section 3.2 covers the maintenance of existing links. Section 4 presents the results of simulating the SWAN protocol. Finally, we conclude and present some ideas for future work in section 5.

# 2 Background

## 2.1 Other ad-hoc network methods

Several authors have examined protocols for self-organizing networks [FSM89, BE81, PKS85, BR90, RS86, GT95]. A common approach to maintaining the network topology in the face of node mobility and changing radio conditions is to periodically tear down the entire network structure and to regenerate the network topology from scratch. These methods continually cycle through alternating phases of network operation. In the first phase, the protocols gather information about the network topology. This information is then used to form a compatible transmission schedule for the exchange of user data during the second phase.

This cyclic approach to network management has two main disadvantages. First, forcing all of the nodes to periodically participate in the topology gathering phase induces a large and possibly unnecessary amount of overhead. This becomes particularly difficult as the acquisition time (the time required for a receiver to "lock on" to a new transmitter's signal) increases. Second, generating a compatible transmission schedule is a complex task, especially as the network size grows.

## 2.2 Interference and distributed power control

We will be assuming spread-spectrum radios at each of the nodes. A simple approximation for the received signal-to-noise ratio (SNR) at a particular node $j$ when receiving from node $i$ is:

$$\mathrm{SNR_j} = \frac{P_i G_{i,j} C_{i,j}}{\sum_{k \neq i} P_k G_{k,j} C_{k,j} + \eta_j} \tag{1}$$

where

- $P_i$ is the transmission power of node $i$.
- $G_{i,j}$ is the path gain between transmitter $i$ and receiver $j$, including any coding gain.
- $C_{k,j}$ is the inter-channel interference between the channels used by transmitter $k$ and receiver $j$.
- $\eta_j$ is the thermal noise power at receiver $j$.

From the form of equation 1 it is easy to see why some care must be taken when transmitting. If a particular node $j$ begins transmitting with a very high power $P_j$, it could easily cause the SNRs of a whole host of ongoing transmissions to drop precipitously.

We will make use of the Distributed Power Control with Active Link Protection (DPC-ALP) power control algorithm [CBP94], summarized briefly here, to ensure that ongoing transmissions are not accidentally destroyed by new ones. Under DPC-ALP, transmitter powers are updated in a series of steps, and every transmission is in one of two states, active or inactive. A transmitter's power at step $i+1$ is a function of its power in step $i$, its state (active or inactive) at step $i$, the desired signal to noise ratio $\gamma$, and the received SNR at step $i$ (fed back from the receiver to the transmitter). Thus if there are $N$ transmissions, with the power of the $i^{th}$ transmission during the $k^{th}$ step given by $P_i^k$ then:

$$P_i^{k+1} = \begin{cases} \delta P_i^k & \text{Inactive Transmissions} \\ \delta P_i^k \frac{\gamma}{\mathrm{SNR}_i^k} & \text{Active Transmissions} \end{cases} \quad 1 \leq i \leq N \tag{2}$$

All transmissions begin in the inactive state at a very low power (possibly commensurate with the noise power as seen by a typical receiver). A transmission becomes active once its received SNR crosses the threshold $\gamma$. A consequence of DPC-ALP is that, in a static network, if a transmission becomes active at step $T$, it will remain active for all $t > T$. This holds because active transmissions are more aggressive in updating their powers, and may in fact prevent other inactive transmissions from becoming active.

## 2.3 Assumptions

The SWAN protocol makes certain assumptions about the underlying network services. In particular we assume that the nodes are equipped with direct-sequence spread spectrum radios capable of supporting the DPC-ALP power control algorithm just mentioned. This means that the radios can provide a measure of a received packet's SNR and allow the transmission power to be set on a per-packet basis. We neglect the finite number of spreading codes available to a particular radio, so that the received SNR is indeed given by equation 1.

For the moment we have assumed that all nodes in the network share a common clock. We are currently investigating the effects of imperfect node synchronization.

For the simulation results of section 4 we further assume that each radio is also capable of using one of several different frequency bands. It was found that this greatly increased the efficiency of new link formation.

## 3 The SWAN protocol

As with many other ad-hoc networking methods, SWAN divides time into a repeating series of frames, which are further subdivided into slots, as shown in figure 3. SWAN employs the DPC-ALP power control algorithm during most of these slots to ensure that ongoing transmissions are not interrupted by transmissions being set up. In addition, there is a single "slot" (possibly of a different size) at the beginning of each frame that is not subject to any power control restrictions. This random access period is used to allow nodes to form new connections within the network. Once a connection has been established between two nodes, they set up periodic control calls to exchange the data needed to maintain link between them and transmit user data.


**Figure 1: The TDMA frame, divided into a short random access period and a data subframe.**

The rest of this section describes these two layers, link formation and control calls, in more detail.

### 3.1 Link formation and topology maintenance

To form new links in the network, nodes handshake during the random access periods at the beginning of each TDMA frame as shown in figure 3.1. Each stage of the handshaking takes place during a *different* frame, so that if node $i$ probes for neighbors during frame $T$, a node may respond to him during the RA period in frame $T + 1$. The handshaking procedure is distributed across several TDMA frames to reduce the total network overhead; by using only one transmission during each RA period, we require only one radio acquisition time and hence can shrink the length of each RA period.

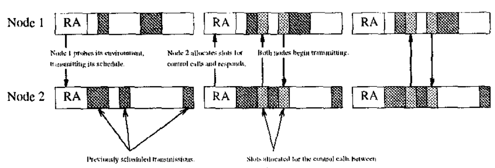


**Figure 2: Handshaking procedure during random access periods.**

In the first handshaking stage, a node probes its surrounding by broadcasting its willingness to form new network connections. To lower the chances of collisions, nodes generate these probe messages probabilistically by calculating, just before each random access period, a probing probability based the node's current view of the network: the number of neighbors it currently has; information gained by the node from listening to previous random access periods; and possibly information from higher network layers as well. If it decides to probe for new neighbors, the node chooses a transmission power and broadcasts a packet containing its ID and a list of the free slots in its TDMA schedule. Again, all handshaking transmissions take place during the random access periods and are not subject to power control and hence are unreliable.

Nodes which receive probing messages may choose to respond during the *next* RA period. As in the first stage, the decision to respond should be a probabilistic function of the responding node's view of the network. When a node responds to a probe packet, it dictates the slots to be used for a pair of control calls which will be responsible for maintaining and managing the new link. Since these control calls are allocated from slots in the data portion of the TDMA frame, they must follow DPC-ALP and hence begin with a low power in the inactive state. If and when both of the control calls become active, then a new link which can be used to transport user data has been added to the network.

### 3.2 Control calls

This section describes SWAN's second layer: the control calls used to manage the links between nodes which have agreed to become neighbors in the network graph. These control calls carry acknowledgments of packets received over the previous frame, link control information (requests to allocate slots for user data transmissions, notifications that particular calls should be torn down, etc.).

To support DPC-ALP, acknowledgments are in the form of the received SNR of a particular transmission. For multislot calls, the receiver can either report the received SNR of each slot individually, allowing much finer power control at the transmitter, or may elect to return the minimum received SNR over the entire group of slots. This second method saves bandwidth in the acknowledgment path at the expense of network capacity. Network capacity is reduced in this case because some slots will be using more transmission power than they need in order that the worst slot can still maintain its required level, $\gamma$.

In addition, the control calls provide a natural means for higher protocol layer to exchange information such as routing tables and network-wide control messages.

A simple timeout mechanism is used to destroy links that become unusable. If a control call goes unanswered for a specified timeout period, the endpoints declare the link down and deallocate the slots that were used for all communications on that link. In such cases it is the responsibility of the higher protocol layers to reroute user calls as appropriate. Since the control calls occur during the data portion of each frame, they are subject to power control. Thus the control calls, and hence the link, will fail only under conditions not covered by the DPC-ALP algorithm. Such conditions include times when the nodes move so far apart that they are no longer able to communicate or when existing active control calls become incompatible due to node mobility.

Figure 3.2 shows an example SWAN network and schedule. Idle slots are blank. Slots containing communications

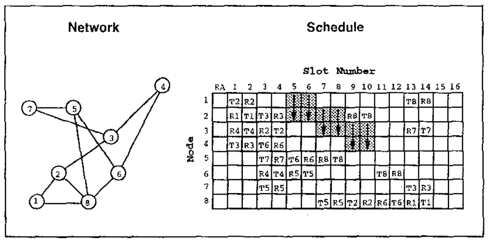

**Figure 3: SWAN Network And Schedule Showing Distributed Control Information and User Call (Gray).**

are marked $XN$ where $X$ is either $T$ or $R$ depending on if the node is transmitting or receiving and $N$ is the node's partner in the communication. The gray slots represent a user call from node 1 to node 4, with the arrows representing packet transmissions.

# 4 Simulation results

In this section we report the results of a computer simulation written to evaluate the SWAN protocol's performance. As described above, SWAN deals only with the formation and maintenance of individual links in the network; no node has any notion of the network topology beyond its immediate neighbors. To quantify notions such as network connectivity which deal with the existence or absence of multi-hop paths through the network, the simulations employ a simple flooding protocol which distributes network topology information to every node in the network.

We say that a path exists in the network between nodes $i$ and $j$ if under node $i$'s view of the network topology, there is a communications path from $i$ to $j$. In a network of $N$ nodes, we define node $C_i^t$, node $i$'s connectivity at time $t$, as the fraction of other nodes in the network to which $i$ has a path (each node always has a path to itself). That is, if node $i$ can form paths to $M$ nodes in the network at time $t$, we say that $i$'s connectivity at $t$ is $C_i^t = M/N$, and the network connectivity is defined as:

$$C_N^t = \sum_{i=1}^{N} C_i^t / N \tag{3}$$

The following parameters and assumptions were used in the simulations:

- All nodes were restricted to a 100x100 unit square. At the end of each frame, each node moved $m$ units in a random direction. Nodes reflected off the boundaries of the square.
- There were 40 slots per frame and 5 non-interfering frequency bands.
- The required SNR was $\gamma = 10$ and we assumed a coding gain of 20.
- Nodes were allowed to transmit at most 60 units, and at most 55 units when forming connections.
- Control calls were assumed to occupy one slot, and the contention period occupied one slot during each frame.
- The timeout for unacknowledged transmissions was 25 frames for regular transmissions. Transmissions were allowed 35 frames when first established.
- Each node was allowed to maintain at most 6 neighbors.
- Each node used information from the topology layer to try to maintain three link-disjoint paths to every other node in the network.

Results of the simulation were used to evaluate the following performance measures of the SWAN protocol.

- Time to construct the network, defined as the minimum $t$ such that $C_N^t = 1$.
- Time required to add a new node to an existing network.
- SWAN's ability to maintain network connectivity in the face of node mobility.

Note that the time required to connect a network of 20 nodes is far less than 20 times that required to add a node to an existing network due to frequency reuse during link formation.

# 5 Conclusion and future work

This paper presented the SWAN protocol, a novel approach to self-organizing network management. Under the SWAN protocol, new connections are formed by handshaking during short contention periods at the beginnings of each TDMA frame, where nodes "probe" their surroundings looking for new neighbors. Once a link between two nodes has been established, periodic control calls maintain and manage the link. Using the DPC-ALP power control algorithm during the data portions of the TDMA frames ensures that new transmissions being powered up do not destroy ongoing communications. The ability of the SWAN

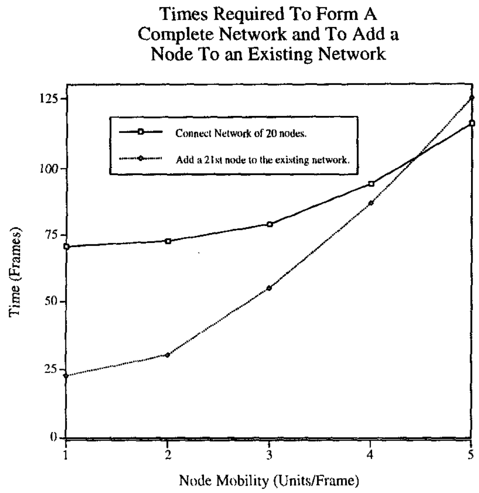


**Figure 4: Time required to initially form a network of 10 nodes using 5 channels.**

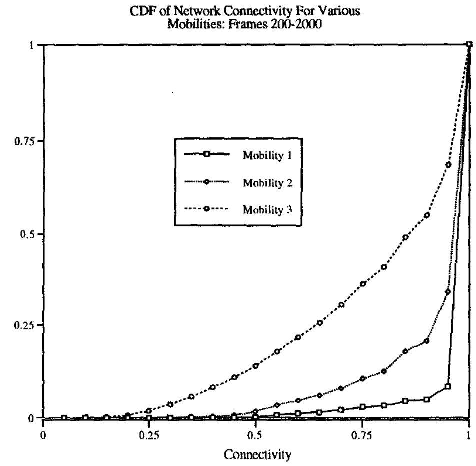


**Figure 5: CDF of the network connectivity during frames 200-2000 for various values of node mobility.**

protocol to cope with node mobility compares favorably to traditional clustering methods, as shown by simulation.

There are several avenues for expanding the current work. First, the link protection characteristic of the DPC-ALP power control algorithm only holds for stationary networks. Indeed, in a mobile network a set of previously compatible transmissions may become incompatible due to node mobility. The effects of changing DPC-ALP parameters on network stability need to be investigated. Also, the impact of imperfect synchronization needs to be addressed. If nodes have different notions of when a frame begins, nodes transmitting in what they believe is the random access period may wipe out other ongoing data communications. Perhaps most interesting is the problem of how nodes should choose their communications partners and so set the network topology. This brings up the possibility that in a stationless network, the topology might actually adapt not only to the changing communications environment, but also to the transmission matrix.

# 6 Acknowledgments

The bulk of this work was performed while the authors were at UCLA under ARPA contract ARPA-CSTO-93-112 through the WAMIS (Wireless Adaptive Mobile Information Systems) project and by the National Young Investigator Award NSF-NCR-9258807. Keith Scott has since joined the Jet propulsion laboratory in Pasadena, and Nicholas Bambos is now an associate professor at Stanford University.

# References

[BE81] Dennis J. Baker and Anthony Ephremides. The architectural organization of a mobile radio network via a distributed algorithm. *IEEE Transactions on Communications*, 29(11):1694–1701, 1981.

[BR90] A. Bhatnagar and T.G. Robertazzi. Layer net: a new self-organizing network protocol. In *MILCOM 90. A New Era. 1990 IEEE Military Communications Conference.*, pages 845–849, 1990.

[CBP94] Shou C. Chen, Nicholas Bambos, and Gregory J. Pottie. On distributed power control for radio networks. In *ICC '94*, 1994.

[FSM89] M. Frankel, N. Shacham, and J.E. Mathis. Self-organizing networks. In *Frontiers in Computing. Proceedings of the International Conference*, pages 165–185, 1989.

[GT95] M. Gerla and J. Tsai. Multicluster, mobile, multimedia radio network. *Wireless Networks*, 1(3):255–265, 1995.

[LNT87] B.M. Leiner, D.L. Nielson, and F.A Tobagi. Issues in packet radio network design. *Proceedings of the IEEE*, 75(1):6–20, January 1987.

[PKS85] M.J. Post, A.S. Kershenbaum, and P.E. Sarachik. A distributed evolutionary algorithm for reorganizing network communications. In *MILCOM 1985*, 1985.

[RS86] T.G. Robertazzi and P.E. Sarachik. Self-organizing communication networks. *IEEE Communications Magazine*, 24(1):28–33, 1986.

[Sco97] Keith Scott. *Control and Routing In Self-Organizing Wireless Networks*. PhD thesis, University of California at Los Angeles, 1997.

[Tob87] F. Tobagi. Modeling and performance analysis of multihop packet radio networks. *Proceedings of the IEEE*, 75(1):135–155, January 1987.