Computer-Aided Modeling of Spread Spectrum Packet Radio Networks

Elvino S. Sousa, Member, IEEE, John Silvester, Senior Member, IEEE, and Thomas D. Papavassiliou, Student Member, IEEE

Abstract—Closed-form expressions for the performance of radio networks are generally not available. For spread spectrum networks, it is not even clear how to represent the underlying network, since the existence of links is heavily dependent on activity in other parts of the network. We have developed a set of computer-based modeling tools to help understand the performance and behavior of these networks. These tools allow for numerical evaluation of important network performance measures and experimentation with network parameters. A connectivity model for a direct sequence spread spectrum packet radio network has been developed. This model allows the computation of network throughput and can be used to study routing techniques. Our current model places direct sequence spread spectrum networks on an equal footing with narrowband networks which have been modeled using the collision channel and transmission range concepts. We describe the mathematical models and algorithms used to evaluate network performance and present several examples to demonstrate the impact of various parameters.

I. INTRODUCTION

SUCCESSFUL design of a packet radio network [1], [2], as in other communication networks, will depend on the good use of divide and conquer principles as manifested in the layering approach. However, the division or layering step is more difficult for packet radio networks than that for hardwired networks, such as the ARPANET, due to the nature of the radio channel.

The complete system design incorporates aspects of: 1) communication theory, such as modulation, and forward error correction coding; 2) multiple access, to achieve statistical multiplexing on the radio channel; 3) transmission power assignment, to enhance network connectivity and minimize network self-interference; 4) routing, to make effective use of the network, bypass high traffic or jammed links, and choose optimum source-destination paths in the case of multihop; and 5) flow control, to maintain network stability in the case of excessive network demand which may degrade performance. The nature of the radio channel is such that it binds all of the above system aspects together so as to make the design of one affect the others to some extent.

The use of spread spectrum on a single link (nonnetwork context) is fairly well understood [3]. Also, its use in a multiple access environment (performance given a fixed set of simultaneous transmissions) has been extensively analyzed [4]. A number of papers have dealt with the use of spread spectrum in a multiple access packet switching environment [5]-[11], where single-hop networks have been mostly considered.

Research in multihop networks was presented in [12]-[18] for narrowband transmission. However, these models do not generalize easily to include spread spectrum signaling. For spread spectrum signaling, initial results have been obtained in [19]-[21] where a "soft" interference model for spread spectrum is used.

The ultimate goal in multihop network modeling is to study the effect of various parameters (processing gain, transmission power, propagation model, transmission probabilities, routing strategies, traffic patterns, etc.) on network level performance measures such as throughput and delay. In addressing these issues, analytical results are difficult to obtain. Direct simulation ([22], [23], for example) is useful for analysis of specific detailed designs, but is limited since the number of parameters involved is so large. What is necessary is a set of computer tools that can be used to gain intuition as to the approximate effect of basic system parameters on the overall system performance. Towards this end, we have developed a set of modeling tools that make use of graphics to experiment with various definitions of network topology, and to study the effect of various parameters on topology. When possible, we also compute network performance measures. The program allows for the creation of random networks and then models the interference characteristics that the nodes encounter as a function of the other nodes' location and transmission activity. By moving a probe around the network, we are able to investigate the interference at different locations. Several simple heuristic routing algorithms have been implemented and are compared. Additional features of the program are the ability to move nodes, add new nodes, or delete existing nodes in an attempt to adapt network topology to improve performance.

In [24], Ruston and Sen introduce an approach for designing packet radio networks with the goal of improving network capacity by altering connectivity values that can be used on a wide range of topologies, traffic requirements, and protocols. The approach uses artificial intelligence techniques to formulate heuristic design rules that are incorporated into a rule-based computer optimization program. Starting from an initial network topology, the program moves along feasible directions of improvement and produces on average 15-20% capacity improvement on both random and regular network inputs. However, the approach is more applicable to narrowband networks since a "hard" interference model is used to define connectivity.
In this paper, we present the current state of the network model (which we have arrived at with the help of the program). In Section II, we abstract the basic results for direct sequence spread spectrum that we use. In Section III, we describe some network topology models. In Section IV, we give the current state of our mathematical model including the computation of throughput and transmission delay. In Section V, we give a qualitative description of the effect of various parameters on network topology which can be observed by running the program. We also present some results obtained for the performance of two routing schemes.

II. SPREAD SPECTRUM MODELING

In this paper, we assume that direct sequence spread spectrum is used. The modeling approach presented in this paper can, however, be generalized to other types of spread spectrum. The key parameter in a direct sequence scheme is the processing gain $G$, or the number of code chips per transmitted symbol. We assume that the receiver is a standard correlator receiver. The noise at the output of the correlator consists of a set of components (one from each interferer) plus a background noise component due to a white noise random process with power spectral density $R_{0}/2$ at the receiver input. For moderate values of $G$, the total noise at the correlator output has a compound Gaussian distribution [25]. Within a symbol, the noise has a Gaussian distribution, but the variance varies from symbol to symbol. For the same total interference power, the variance has the largest variation in the case of one strong interferer, where the noise variance is dependent on the relative carrier and spreading code chip phases of the signal and interferer. In the case of many weak interferers, the noise distribution reduces to the Gaussian distribution where the variance has zero variation.

For network modeling purposes, it is convenient to employ a threshold model on the total interference where, if the total interference is less than the threshold, then the transmitted signal is receivable in the sense that, if the receiver is not captured by another terminal, it can receive the designated signal. This threshold model is dependent on the use of good error correcting codes and on a small variation of the noise variance. The above variation of the SNR at the correlator output, in the case of one strong interferer, occurs if the carrier and spreading code clocks of the signal and interferer have the same frequency. In real networks, there will be frequency offsets between the various clocks which tend to cause a reduction in the variation of the noise variance [26], hence making the threshold model more applicable.

With the threshold model, the noise at the output of the correlator is assumed to have a constant variance given by the expected value of the variance in the case of compound Gaussian modeling. The average signal-to-noise ratio was initially computed in [27]. For our purposes, we use the following form obtained in [25]:

$$\text{SNR} = \frac{Y_0}{2G \sum_{i=1}^{M} Y_i + R R_{0}}$$

where $Y_0$ is the signal power at the receiver, $Y_i$ is the power of the $i$th interferer, $M$ is the number of interferers, and $K$ is the symbol transmission rate. The symbol error probability is $\frac{1}{2}\text{erfc}(\sqrt{\text{SNR}})$. Actually, it is more convenient to work with the interference to signal ratio and, since our results are in terms of thresholds, we define the interference to signal ratio as follows:

$$\text{ISR} = \sum_{i=1}^{K} Y_i + GN_0$$

where $N_0 = \frac{3}{2}R^2R_{0}$. Let us denote by $\eta$ the SNR in (1) required for an acceptable error rate at the decoder input (a typical value is less than 10 dB). We define the threshold parameter

$$K = \frac{3G}{2\eta} \equiv \frac{G}{D}$$

where $D = \frac{1}{\eta}$. We will assume that a packet at the receiver can be successfully received if and only if $\text{SNR} > \eta$, or more conveniently $\text{ISR} < K$.

III. NETWORK MODELING

Three basic topology models have been employed in past research in multihop packet radio networks. We refer to these as the random, connectivity matrix, and regular models. In the random model, we do not assume that the terminal positions are known; rather, the network is assumed to occupy some region of the plane and the terminals are assumed to be uniformly distributed in this region. In some cases, we do not want to deal with network boundary effects; thus, an infinite network is assumed and the appropriate distribution of terminals is the Poisson distribution. This model was initially used by Silvester [28], [12] to study the transmission range versus interference tradeoff in a multihop narrowband network with a slotted ALOHA transmission protocol. It was then generalized to the CSMA protocol, to a nonuniform distribution of terminals, to slotted ALOHA with capture in [13], [29], and to spread spectrum networks in [19].

In the connectivity matrix model, the network is represented as a directed graph where vertices represent terminals and an edge is present from $A$ to $B$ if $B$ can hear $A$. This model was used in [30], [31] to obtain theoretical results for the throughput of a multihop CSMA network.

The third model is the regular model where it is assumed that the above connectivity graph is "regular," i.e., it has some structure. Examples of these are the multiconnected rings and grid networks studied in [32].

The above graph models are not applicable to a direct sequence spread spectrum network. A possible modification is to generalize the collision channel so as to allow more than one transmission on the channel. However, such models are not accurate for distributed ground radio networks since the number of transmissions is typically not a good indication of the total interference at a terminal. A better model is the model presented in the preceding section where packet success is determined by the ratio of the signal to interference power. We refer to this as a "soft" interference model. To our knowledge, except for [21], such a model has not been employed elsewhere.

To study routing issues, it is convenient to represent a communication network as a graph. For systems like the ARPA-NET, there is an obvious representation of the network as a
graph, where vertices represent terminals and edges represent transmission links. With radio transmission, the multiuser communication system is also called a network; however, the identification of the system with a graph is nontrivial.

In narrowband packet radio networks, a multihop network can be modeled using the connectivity matrix described above. Although the resulting graph is usually called a network, it is not a network in the graph theoretical sense since the flows in different links interfere with each other. With spread spectrum signaling, it is even less clear as to what a graphical representation of the network should be, especially if the processing gain is to be included. If we assume the interference model discussed in the previous section, then it is possible to transmit successfully to a terminal at a distance that is limited only by $2\xi$. In practice, this distance may be much greater than the optimum transmission range and the probability of success may be very small. Thus, if we represent the network in the same way that narrowband networks are represented, we will have a graph which contains many links that are mostly unreliable due to high ISR, hence, the representation will not be useful. Instead, we would like a graph where the included links are those which satisfy the relation

$$P[I\text{SR} < K] > u$$

(4)

where $u$ is a parameter which we call the up probability. If we fix a link in the network, then the probability of success is a decreasing function of the link distance. To obtain a graphical representation for the network, we will compute $P[I\text{SR} < K]$ for every pair of nodes and consider as links only those pairs that satisfy (4). The parameter $u$ can then be varied to see links of different quality. An easy case to consider is that where $u = 1$. In this case, the processing gain is large enough so that for the network links shown the only cause of failed transmissions is contention at the receiver (either due to other transmissions to that receiver or the node not being in the receive mode) and not a high ISR. In Fig. 1, we show some resulting network topology graphs for different values of the parameter $u$. These network diagrams are described further in Section V.

Of course, the link quality is affected by the transmission probabilities that the nodes use since they impact $P[I\text{SR} < K]$. Determination of the node activity levels (transmission probabilities) is a hard problem in general, since we must know traffic levels, solve the routing problem, and then solve a fairly complex set of nonlinear equations by iteration (discussed in more detail below). For some environments (to avoid traffic analysis, for example) or for initial assumptions, we may assume that all nodes use the same transmission probabilities (transmitting even though they have nothing to send on occasion). In Fig. 2(c), we have used $p = 1$; this corresponds to the worst possible interference scenario.

IV. MATHEMATICAL MODEL

We assume that we have a network consisting of $n$ terminals. The terminals transmit signals whose power attenuates according to the function

$$g(r) = \frac{1}{r^\alpha}$$

(5)

where $3 < \alpha < 5$ for ground propagation. A commonly used value, which is based on a two-ray propagation model, is $\alpha = 4$ [33]. We index the terminals by the integer $i$, $i = 1, \cdots, n$. Let the distances between the terminals be $\{r_{ij}\}$, then we can form the following interference matrix $\{Y_{ij}\}$, where

$$Y_{ij} = \frac{1}{r_{ij}}$$

(6)

The system is slotted and, during each slot, terminal $i$ transmits a packet to terminal $j$ with probability $p_{ij}$. The whole transmission activity is therefore specified by the matrix $P = \{p_{ij}\}$. The transmission probabilities for the terminals are denoted as $p_i$.

$$p_i = \sum_{j=1}^{n} p_{ij}$$

(7)

To obtain the parameter $u$, we need to compute the probability that the interference is less than a given threshold for a link $(s, d)$. A direct method is to compute the probability density of the total interference as the convolution $f_{sd} = f_{s} * f_{d} * \cdots * f_{sd}$, where $w(x) = \delta(x - GN_0)$.

$$f_{sd}(x) = (1 - p_{i})\delta(x) + p_{i}\delta(x - Y_{sd}) \quad i \notin \{s, d\}$$

(8)

and $\delta(\cdot)$ is the Dirac delta function, and then integrate the tail of $f_{sd}$. The exact computation of these convolutions is not feasible for large $n$ since the result consists of a sum of $2^n$ delta functions. However, we may approximate the computation by first discretizing the interference powers $Y_{sd}$, so that the loca-
Fig. 2. Effect of different transmission probabilities: \( n = 40, K = 40, \alpha = 4, a = 0.8 \). (a) \( p = 0.1 \), (b) \( p = 0.4 \), (c) \( p = 1 \).

The convolution may then be computed efficiently by first sorting the numbers \( Z_d \) so that we start by convolving the smallest two densities (smallest \( Z_d \)), and proceed by choosing the next smallest density. The shape of the resulting density actually gives us more information on the link interference than the probability \( P[ISR < K] \).

An alternative approach, which obtains the probability \( P[ISR < K] \) directly, is to use the following algorithm.

**Algorithm 1:**

(For notational convenience, we assume receiver node \( d \) and use \( Y_i \) to represent \( Y_{d,i} \), etc.)

Assume \( k \) interferers with transmission probabilities \( p_i \) and interference powers \( Y_i \), \( i = 0, \cdots, k - 1 \). Let \( S \) be the signal power.

Re-index (sort) the interfering terminals so that they are in order of increasing power, \( Y_i \).

Defining \( C_o = 0, C_i = \sum_{j=0}^{i-1} Y_j, i = 1, \cdots, k \).

Set \( th = KS - GN_0 \).

The probability \( P[ISR < K] \) is returned by the function \( Ps(k - 1, th) \) where \( Ps(\cdot, \cdot) \) is computed recursively as follows:

\[
Ps(v, th) = \begin{cases} 
0 & \text{if } (th < 0) \\
Ps(v + 1, th) & \text{if } (C_{v+1} \leq th) \\
1 & \text{else}
\end{cases}
\]

In Fig. 3, we show a network and to the right (smaller plot) the probability density function (pdf) at two points of the network. In (a), the probe is close to an interferer, hence the bimodal pdf; in (b), there are terminal that is similar to the probe, hence, the unimodal pdf. Taking thermal noise into account and the fact that a propagating signal attenuates with distance, there is a transmission range limited by \( N_0 \) only.

**A. Link Throughput**

We denote the link throughputs by the matrix \( \{s_{ij}\} \). The value \( s_{ij} \) is the probability that terminal \( i \) successfully transmits a
packet to terminal \(j\) in a given slot. Let us fix a transmitter \(S\), whose index is \(s\), a receiver \(D\) whose index is \(d\), and compute the link throughput \(s_{sd}\). For a packet to be successfully transmitted on the link \((s, d)\), a number of events must occur. First, terminal \(S\) must transmit a packet addressed to terminal \(D\); this event has probability \(p_{s,d}\). Second, since we are assuming that each terminal has only one receiver, terminal \(D\) must not be transmitting; the probability of this event is \(1 - p_{d,d}\). Third, the total interference at the receiver must be such that we have

\[
I_{sd} = GN_0 + \sum_{i \in \{s, d\}} Y_i < KY_{sd}
\]

where we have defined \(I_{sd}\) as the total interference affecting the link \((s, d)\), and \(K\) is given by (3). The parameter \(N_0\) is assumed to be constant throughout the network. Four events, given the above three events, it must be the case that terminal \(D\) captures the packet from \(S\) and not some other transmission. To compute the probability of the last event, we define a packet, transmitted from terminal \(k \neq d\), to be receivable if

\[
I_{sd} = GN_0 + \sum_{i \in \{s, d\}} Y_i < KY_{sd}
\]

that is, a packet is receivable if it has a sufficiently low ISR. We assume that all equal-packet sizes have an equal probability of being captured. Given that there are \(k\) receivable packets (including the packet from \(S\)), we assume that each one has a probability of being captured equal to \(1/k\). We may therefore write the throughput computation as follows:

\[
s_{sd} = p_{s,d}(1 - p_{d,d}) \left[ I_{sd} < K & S \text{ is captured by } D \right].
\]

We now address the issue of the computation of the last factor of (11). The difficulty of this computation depends on the set of terminals that may transmit to terminal \(D\). First, we consider the case where the transmission from terminal \(S\) is always receivable regardless of the activity of the other terminals. We also assume that the set of terminals \(\{T_i\}\) for which \(p_{s,d} \neq 0\) are precisely those terminals that are always receivable. This set of terminals may be obtained from the network graph, with the terminal transmission probability \(p = 1\), as those terminals from which there is an incoming link to terminal \(D\). Denote the above set of terminals, not including \(S\), by \(C = \{T_i, T_j, \cdots, T_n\}\). In this case, the outcome of the third event in (11) is solely dependent on the activity of the terminals in \(C\). Let \(i_j\) be the indicator function for the event terminal \(T_i\), transmits to terminal \(D\). Equation (11) then becomes

\[
s_{sd} = p_{s,d}(1 - p_{d,d}) \left( \sum_{i \in C} i_j \right) \left( \sum_{k=0}^{\infty} P[\mathcal{N}_C = k] \cdot \frac{1}{k + 1} \right) \]  

where \(\mathcal{N}_C = \sum_{i=1}^{n} i_j\) and \(i_j\) is a Bernoulli random variable with \(P[i_j = 1] = p_{s,d}\). For the case that the terminals in \(C\) have equal transmission probabilities to terminal \(D\), the random variable \(\mathcal{N}_C\) has a binomial distribution. For the case of unequal transmission probabilities, we may use the following recursive computation technique to obtain \(P[\mathcal{N}_C = k]\). Let \(n_0 = |C|\). The procedure \(\text{Prob}(n_0 - 1, k)\) computes \(P[\mathcal{N}_C = k]\), where \(\text{Prob}(n, k)\) is defined as follows.

**Algorithm II**

Set \(q = p_{s,d}\), for \(j = 0, \cdots, n_0 - 1\); \n
\(\text{Prob}(n, k)\) \{

- if \((k < 0)\) return \((0)\); 
- if \((k = -1)\) return \((1)\);

\(\text{else} \{

- \(P \leftarrow 1\);
- \(v \leftarrow 0\);
- \(p \leftarrow p_{s,d}\);
- \(Y_{sd} \leftarrow Y_{sd} - Y_i\);
- \(I_{sd} \leftarrow I_{sd} - I_i\);
- \(I_{sd} \leftarrow I_{sd} + Y_i\);
- \(Y_{sd} \leftarrow Y_{sd} + Y_i\);
- \(I_{sd} \leftarrow I_{sd} - I_i\);
- \(Y_{sd} \leftarrow Y_{sd} - Y_i\);
- \(I_{sd} \leftarrow I_{sd} + Y_i\);
- \(Y_{sd} \leftarrow Y_{sd} + Y_i\);
- \(I_{sd} \leftarrow I_{sd} - I_i\);
- \(Y_{sd} \leftarrow Y_{sd} - Y_i\);
- \(I_{sd} \leftarrow I_{sd} + Y_i\);
- \(Y_{sd} \leftarrow Y_{sd} + Y_i\);
- \(I_{sd} \leftarrow I_{sd} - I_i\);
- \(Y_{sd} \leftarrow Y_{sd} - Y_i\);
- \(I_{sd} \leftarrow I_{sd} + Y_i\);
- \(Y_{sd} \leftarrow Y_{sd} + Y_i\);
- \(I_{sd} \leftarrow I_{sd} - I_i\);
- \(Y_{sd} \leftarrow Y_{sd} - Y_i\);
- \(I_{sd} \leftarrow I_{sd} + Y_i\);
- \(Y_{sd} \leftarrow Y_{sd} + Y_i\);
- \(I_{sd} \leftarrow I_{sd} - I_i\);
- \(Y_{sd} \leftarrow Y_{sd} - Y_i\);
- \(I_{sd} \leftarrow I_{sd} + Y_i\);
- \(Y_{sd} \leftarrow Y_{sd} + Y_i\);
- \(I_{sd} \leftarrow I_{sd} - I_i\);
- \(Y_{sd} \leftarrow Y_{sd} - Y_i\);
- \(I_{sd} \leftarrow I_{sd} + Y_i\);
- \(Y_{sd} \leftarrow Y_{sd} + Y_i\);
- \(I_{sd} \leftarrow I_{sd} - I_i\);
- \(Y_{sd} \leftarrow Y_{sd} - Y_i\);
- \(I_{sd} \leftarrow I_{sd} + Y_i\);
- \(Y_{sd} \leftarrow Y_{sd} + Y_i\);
- \(I_{sd} \leftarrow I_{sd} - I_i\);
- \(Y_{sd} \leftarrow Y_{sd} - Y_i\);
- \(I_{sd} \leftarrow I_{sd} + Y_i\);
- \(Y_{sd} \leftarrow Y_{sd} + Y_i\);
- \(I_{sd} \leftarrow I_{sd} - I_i\);
- \(Y_{sd} \left\} 

For large values of \(n_0\), the above algorithm has a large running time (on the order of \(2^n\)). In such a case, we may approximate the distribution of the random variable \(\mathcal{N}_C\) by a Poisson distribution with mean \(\lambda\), where

\[
\lambda = \sum_{j=1}^{n} p_{s,d}. \]  

Even though the approximation of the distributions may not be very good, the computation of the expectation of the random variable \(1/(\mathcal{N}_C + 1)\) as in (12) is surprisingly well approximated for a wide range of sets of \(p_{s,d}\)'s. The approximation of the summation in (12) is then

\[
\sum_{k=0}^{\infty} \frac{\lambda^k e^{-\lambda}}{k!} \cdot \frac{1}{k + 1} = 1 - e^{-\lambda}. \]  

As an example, we compare the exact evaluation of the sum in (12) to the approximation in (14) for different cases of \(n\), assuming that all the transmission probabilities to \(D\) are equal. The worst case is that where \(n\) is small and \(\lambda\) is large. We give the following pairs, where the first number is the exact value and the second is the approximation: for \(n = 1\), the three \(\lambda\)'s the corresponding pairs are \((0.951, 0.951), (0.819, 0.824), \) and \((0.610, 0.632); and for \(n = 8\), we have \((0.951, 0.951), (0.821, 0.824), \) and \((0.626, 0.632), \) respectively.

We now consider the case where the incoming links to the receiver are soft links. The transmission from \(S\) may or may not be receivable, depending on the activity of the other users. We partition the set of interfering terminals \((n - 2)\) into three subsets: a subset \(S_0\) that does not transmit, a subset \(S_0\) that transmits but not to \(D\), and a subset \(S_0\) that transmits to \(D\). For the packet of interest to be received, the sum of the interference power from the terminals in sets \(S_0\) and \(S_0\) plus the background noise \(GN_0\) must be less than the threshold \((KY_{sd})\); given such an event, we determine the number \(k\) of transmissions in \(S_0\) that are receivable. The conditional probability of reception is then \(1/(k + 1)\). Since each terminal has three states, the set of interferers has \(3^{n-2}\) transmitting states. Thus, a direct computation of the last factor of (11) by conditioning on each state is not feasible for \(n > 20\). One way to handle this problem is to assume that the events \{interference is less than threshold\} and \{the packet from \(S\) is captured\} are independent. The probability of the former may be computed by either the convolution method, or by using Algorithm I, and the probability of the latter is computed as in (14) or by using Algorithm II. We are currently developing an algorithm to perform the exact computation, i.e., without making the independence assumption.
At this point, we are able to compute link success probabilities for a given set of node locations, routes, and transmission probabilities. The problem is that the transmission probabilities and interference patterns. In the next section, we describe a procedure to compute the transmission probabilities \( \{ p_{ij} \} \) that are needed to achieve a particular throughput and a search procedure to find the maximum supportable throughput for the given set of routes.

**B. Network Throughput**

In packet radio networks, the problem of determining a set of optimum node-to-node transmission probabilities \( \{ P = \{ p_{ij} \} \) to use, in order to optimize some measure of network performance (e.g., maximize throughput or minimize delay), is hard in general. Since, in a multihop network, the routing scheme dictates how traffic is forwarded between nodes, it affects the throughput directly. In addition, these probabilities influence the traffic and congestion levels in the network and, therefore, (circularly) affect the choice of a “good” routing algorithm. Some recent promising work on this problem for narrowband networks can be found in [34].

We can simplify the problem by assuming a known static topology (i.e., number of nodes and their positions) and the routing to be fixed. For a given end-to-end traffic requirement matrix \( J = \{ t_{ij} \} \) (assumed to be normalized, i.e., \( \Sigma t_{ij} = 1 \)) and a routing scheme \( \Omega \), we can assign flows to the links on the paths determined by \( \Omega \) and compute a set of desired link flows \( \Phi = \{ \phi_{ij} \} \). In general, for some links, \( \phi_{ij} \) will be zero and we refer to these links as unused links and for the remaining links \( \phi_{ij} \) will be nonzero and we refer to them as used links. Let \( \Sigma \) be the set of links that are used. For the purpose of maximizing throughputs, we adopt the heavy traffic assumption. Under this assumption, nodes always have packets to be transmitted and the throughput for link \((i, j)\) is expressed as

\[
x_{ij} = p_{ij} A_i(P), \quad (i, j) \in \Sigma
\]

as in (11), with \( s, d \) replaced by \( i, j \). The \( A_i(P) \) are conditional success probabilities that depend on the transmission activity of the nodes and possibly the topology.

The problem of maximizing the network throughput can be formulated as follows. We define the normalized end-to-end traffic matrix \( J \) by \( y_{ij} \), i.e., \( y_{ij} = t_{ij} \). Thus, \( F = \{ y_{ij} \} \) represents the true network end-to-end traffic requirement matrix. The link traffic pattern \( \Phi \) will also be scaled to \( F = \gamma \Phi = \{ \gamma \phi_{ij} \} \).

Let \( f_{ij} = \gamma \phi_{ij} \). We want to solve for \( \gamma \) where

\[
\gamma^* = \sup \{ \gamma \}
\]

such that

\[
f_{ij} = \gamma \phi_{ij} = p_{ij} A_i(P), \quad (i, j) \in \Sigma
\]

is feasible for some \( \{ p_{ij} \} \) with \( 0 \leq p_{ij} \leq 1 \). \( \gamma^* \) represents the maximum value of the sum of end-to-end traffics that the network can support for the given traffic requirement \( J \) and routing \( \Omega \).

We decompose (16) and solve the following two problems.

a) Given a set of required link throughputs \( F = \{ f_{ij} \} \), find if they are feasible, i.e., if they satisfy

\[
f_{ij} = p_{ij} A_i(P), \quad (i, j) \in \Sigma
\]

and, if so, compute the transmission probabilities \( P = \{ p_{ij} \} \) that will yield these throughputs.

b) Given a link traffic pattern \( \Phi = \{ \phi_{ij} \} \), find \( \gamma^* \), the maximum value of \( \gamma \) such that \( \gamma \Phi \) is a feasible set of link traffic requirements.

Similar problems of solving nonlinear simultaneous equations have been encountered in [33], [36]. To solve a), we rewrite (17) as

\[
p_{ij} = \frac{f_{ij}}{A_i(P)}.
\]

We can use a fixed point type of iteration as in the following algorithm.

**Algorithm III**

Set \( converge = false \), \( feasible = true \), \( k = 0 \);

\[
p_{ij}(0) \leftarrow f_{ij}, \quad (i, j) \in \Sigma;
\]

\[
p_{ij}(0) \leftarrow \Sigma p_{ij}(0), \quad \text{for } i = 1, \ldots, n;
\]

\[
\text{while } (\text{not converge and feasible}) \{
\]

\[
\begin{align*}
k &\leftarrow k + 1; \\
p_{ij}(k) &\leftarrow f_{ij}/A_i(P(k-1)), \quad (i, j) \in \Sigma; \\
p_{ij}(k) &\leftarrow \Sigma p_{ij}(k), \quad \text{for } i = 1, \ldots, n;
\end{align*}
\]

\[
\text{if } (p_{ij}(k) > 1) \text{ for any } (i, j) \text{ then } feasible \leftarrow false; \\
\text{if } (\{p_{ij}(k) - p_{ij}(k-1)\} < \epsilon) \text{ then convergence } \leftarrow true;
\]

\[
\}
\]

The \( p_{ij}(k) \) are the transmit probabilities from node \( i \) to node \( j \), and \( p_{ij}(k) \) the probability that node \( i \) transmits to any node at step \( k \). Assuming that \( p_{ij} \leq 1 \), we note that \( A_i(P) \leq 1 \) (as conditional success probabilities computed by (11)). Then the \( p_{ij}'s \) are nondecreasing with successive \( k \) and \( A_i(P) \) are nonincreasing. The iteration will stop either if some \( p_{ij}'s \) exceeds 1 (in which case the \( f_{ij}'s \) are declared infeasible) or if all \( p_{ij}'s < 1 \), then \( P \) will approach some limit \( P^* \). The convergence condition can be made very tight, e.g., requiring \( |p_{ij}(k) - p_{ij}(k-1)| < \epsilon \) for all links \((i, j)\) or looser, e.g., \( |p_{ij}(k) - p_{ij}(k-1)| < \epsilon \) for all nodes \( i \). Our experience has been that using the second convergence condition considerably decreases the running time without significant loss of accuracy.

To solve b), we can use a binary search where at each step we test the feasibility of the tentative value of the throughput by Algorithm III.

**Algorithm IV**

Set \( lower = 0 \), \( upper = 1/\max_k \{1,2,\ldots,n\} \Sigma \phi_{ij} \),

\[
\gamma = \frac{(lower + upper)}{2};
\]

\[
\text{while } (\text{upper } - \gamma > \epsilon) \{
\]

\[
F \leftarrow \gamma \Phi;
\]

\[
\text{check feasibility of } F \text{ using Algorithm III};
\]

\[
\text{if (not feasible) upper } \leftarrow \gamma;
\]

\[
\text{else lower } \leftarrow \gamma;
\]

\[
\gamma^* \leftarrow \gamma.
\]

Since we are looking at a packet radio network, the initial value of \( upper \) is determined by the requirement that for each node \( i \) the total successful flow outgoing from \( i \) must not exceed 1, i.e.,

\[
\sum_{j=1}^{n} f_{ij} = \gamma \sum_{j=1}^{n} \phi_{ij} \leq 1, \quad i = 1, \ldots, n.
\]

Together with the maximum throughput \( \gamma^* \), we can obtain the optimum transmission probabilities \( \{ p_{ij}^* \} \). We note that a disadvantage of the binary search method is the number of iterations that may be required by Algorithm III to detect convergence or divergence when \( \gamma \) is close to \( \gamma^* \).
C. End-to-End Delay

When looking at routing optimization problems in conventional point-to-point networks, the optimization criterion typically used to characterize performance is either average network delay (including queueing at intermediate nodes) or maximum throughput. In the multihop packet radio network case, however, exact analytic expressions for the above performance measures do not exist, due to strong dependence between the queues at different nodes. Attempts to formulate a precise mathematical model for this case lead to very complex problems of coupled queues, hence, approximations are used often.

Towards this end, we use a different measure, transmission delay, that corresponds to the time required to transit the network in light load, also called the no-load delay. This measure includes the effects of interference and transmission probability and is useful to compare routing strategies. Looking at a source-destination pair \((s, d)\), suppose that the routing algorithm dictates a path \(\pi_{sd}\) that packets with origin \(s\) and destination \(d\) should follow. Then the path transmission delay is simply the sum of the transmission delays for all links on the path

\[ D_{sd} = \sum_{(i, j) \in \pi_{sd}} D_{ij}, \]

and the network average transmission delay is

\[ D = \sum_{s} \sum_{d} \gamma_{sd} D_{sd}, \]

which depends on the traffic pattern. The parameter \(\gamma_{sd}\) is the traffic requirement for source-destination pair \((s, d)\) and \(\gamma\) is the total traffic requirement.

\[ \gamma = \sum_{s} \sum_{d} \gamma_{sd} \]

\(D_{ij}\) is the average transmission delay on link \((i, j)\) for a packet that is at the head of the queue of \(i\) and is simply evaluated as

\[ D_{ij} = \frac{1}{s_{ij}} \]

where \(s_{ij}\) is the rate of successful transmissions on link \((i, j)\) and is given by (11).

V. ROUTING AND CONNECTIVITY EXPERIMENTS

We now give a qualitative description of the effect of various network parameters on network topology, as determined by running the program. We also give results for some routing experiments. An initial version of this program and network model was described in [20]. The networks are generated in a \(100 \times 100\) square. Each terminal is randomly placed in the square by choosing the \(x\) and \(y\) coordinate from a uniform distribution between 0 and 99. If a terminal already exists at the generated coordinates, then we discard the given point and a new \((x, y)\) pair is chosen; the process is repeated until the desired number of terminals \(n\) are placed.

It might be desirable to exclude the possibility that nodes cluster very close together. Thus, a parameter \(d_{min}\) is introduced, which enforces a minimum distance between terminals. The terminals are placed randomly but such that any pair is at least a distance \(d_{min}\) apart. The parameter \(d_{min}\) can be used to introduce regularity in the network topology. As \(d_{min}\) increases, regularity increases. The case \(d_{min} = 1\) corresponds to completely random networks and higher values of \(d_{min}\) to more regular topologies. If \(d_{min}\) is too large, we may not be able to place all the \(n\) terminals in the square.

Terminals are represented by small squares. Unidirectional links are shown as a line with an arrow, and bidirectional links are shown as a line with no arrow. The terminals are assumed to transmit with probabilities \(p_1, \ldots, p_n\). Fix a pair of terminals \(S, D\). A link \(S \rightarrow D\) is drawn if we have \(P(ISR < K) > u\) at \(D\), where terminals \(S, D\) are excluded in computing the total interference. In a network where we have \(u < 1\), we refer to the links as soft links.

The program also has an interference sensor which can be located at different coordinates to look at the probability density of the interference at various points of the network. Terminal positions can also be individually adjusted to modify the network topology. We now describe the effect that key parameters have on network topology.

A. Qualitative Behavior

Let us consider a link and assume that there are no interfering terminals. The maximum link distance \(R_0\) is obtained through (2) and (5) as \(R_0 = (DN_0)^{-1/\alpha}\). This distance may or may not be greater than the maximum distance between terminals. In the case where network self-interference exists, \(R_0\) is the limit to the maximum link distance as the parameter \(u\) is decreased. In the case of zero interference, the parameter \(R_0\) plays the role of a transmission radius in the case of narrowband networks. If we let \(u = \varepsilon\) where \(\varepsilon \ll 1\), then the resulting network graph is the same as a network connectivity graph for narrowband networks with transmission radius equal to \(R_0\). We show two different noise environments in Fig. 4, where the circle around the probe represents \(R_0\). Note that such networks contain many links but the transmissions are subject to collisions on the channel. The other extreme case is when \(u = 1\). In this case, see Fig. 5, the network has fewer links but channel collisions will never occur; the only cause of lost packets is contention for the receiver port.

As previously mentioned, the regularity of the network is determined by the parameter \(d_{min}\). The connectivity is dependent on the regularity; networks with greater regularity have greater connectivity (Fig. 6). This is especially true for large \(\alpha\). The processing gain also determines connectivity. The higher the processing gain, the higher the connectivity (Fig. 7). The increase of connectivity with \(G\) is dependent on the regularity and the power loss exponent \(\alpha\). The processing gain can be increased to the point where the network graph becomes the same as that for \(u = \varepsilon\), as discussed above (maximum connectivity). However, if the network is irregular and \(\alpha\) is large (e.g., 4), then it will take an extremely high value of \(G\) to achieve maximum connectivity. For environments where \(\alpha\) may be smaller, the maximum connectivity may be achieved with a smaller \(G\). In the extreme case \(\alpha = 0\), i.e., no power loss due to distance, \(G\) need only be increased so that \(K = n - 2\) to achieve maximum connectivity.

In the above discussion, for \(u < 1\) we have assumed that the terminal transmission probabilities are approximately equal so that the interference is related to the number of terminals in the vicinity of the receiver and their distances. In cases where the transmission probabilities are different, the connectivity will depend on the transmission probabilities in addition to the regularity, \(\alpha\), and \(G\).

B. Routing Experiments

As we mentioned in Section IV-B, for a given set of end-to-end traffic requirements \(\Delta = \{\tau_{ij}\}\) and a routing matrix \(\Phi\) we can compute a set of desired link flows \(\Phi\) and then solve a sys-
system of nonlinear equations by iteration to obtain the maximum throughput.

To cover the case where we do not know the exact traffic requirements, it is convenient to assume uniform end-to-end traffic; that is, we assign to each pair of nodes a traffic requirement of \( r_{ij} = 1/n(n-1) \).

Next we need to choose a set of paths for routing, compute the desired link throughputs, and optimize with respect to the \( p_{ij} \)'s to maximize network throughput. As noted earlier, the routing problem is very difficult, so we simplify the problem by assuming a single path fixed routing scheme \( R \) based on some heuristic.

1) \( R_1 \): Minimum Hop Using "Dominant" Links: Let us fix the transmission probabilities \( p_{ij} = 1 \) for all terminals. The computer program is then used to draw a network connectivity graph. The resulting links are the ones that cannot be interfered with under any circumstances, we call them dominant links. One heuristic is to select the routing links \( R \), from the set of dominant links by running a minimum hop algorithm. In some cases, the networks resulting from only the dominant links may not be connected, e.g., when the processing gain is such that \( K \) is small (e.g., \( K < 10 \) for \( n = 10 \)), the power loss exponent is large (\( \alpha = 4 \)), and \( d_{min} = 1 \). In this paper, we exclude such cases and look at connected networks. We show an example of the links used in routing in Fig. 8(a) and performance results in Table I.

2) \( R_2 \): Minimum Transmission Delay Routing (min D): In Section IV-B, we gave an expression for the network transmission delay \( D \) given a set of link throughputs \( \{ s_{ij} \} \) and, in Section IV-A, we show how to compute these \( s_{ij} \)'s given the topology and a set of terminal-to-terminal transmission probabilities \( \{ p_{ij} \} \). Initially, however, these \( p_{ij} \)'s are not known. We simplify things by assuming all terminals use the same transmission probability \( p_{ij} = p \). For the uniform traffic assumption,
we evaluate an approximate expression for $s_y$, based on (14):

$$s_y = \frac{p}{n - 1} (1 - p) P \left[ \frac{I_p}{Y_i} < K \right] \frac{1 - e^{-\alpha}}{p}.$$

We now use the average transmission delay (20) as link costs and run a shortest path algorithm to obtain a set of shortest path routes $R_3$. We show an example of the links used in routing in Fig. 8(b) and performance results in Table I for a range of different initial values for the transmission probability $p$.

3) $R_3$: One Hop Routing: In spread spectrum networks with no background noise, it is always possible to communicate directly with the destination (albeit with low probability of success). Here, we look at the case of one hop routing between source and destination. This corresponds to a routing graph that is fully connected. We show an example of the links used in routing in Fig. 8(c) and performance results in Table I.

VI. SUMMARY

In this paper, we have presented a modeling approach for spread spectrum radio networks which makes use of computer based tools for investigation of qualitative and quantitative network behavior. We describe a computer implementation of these tools that interfaces to a graphics package, allowing rapid ex-

---

**TABLE I**

<table>
<thead>
<tr>
<th>Routing Strategy</th>
<th>Maximum Throughput, $\gamma^*$</th>
<th>Path Delay, $D$</th>
</tr>
</thead>
<tbody>
<tr>
<td>dominant links</td>
<td>0.66</td>
<td>39</td>
</tr>
<tr>
<td>one hop</td>
<td>0.34</td>
<td>269</td>
</tr>
<tr>
<td>$min {D_i}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$p = 0.05$</td>
<td>1.03</td>
<td>87</td>
</tr>
<tr>
<td>$p = 0.10$</td>
<td>1.03</td>
<td>87</td>
</tr>
<tr>
<td>$p = 0.20$</td>
<td>1.05</td>
<td>79</td>
</tr>
<tr>
<td>$p = 0.30$</td>
<td>0.96</td>
<td>65</td>
</tr>
<tr>
<td>$p = 0.40$</td>
<td>0.89</td>
<td>54</td>
</tr>
<tr>
<td>$p = 0.50$</td>
<td>0.80</td>
<td>55</td>
</tr>
<tr>
<td>$p = 0.60$</td>
<td>0.65</td>
<td>49</td>
</tr>
<tr>
<td>$p = 0.70$</td>
<td>0.69</td>
<td>41</td>
</tr>
<tr>
<td>$p = 0.80$</td>
<td>0.62</td>
<td>45</td>
</tr>
<tr>
<td>$p = 0.90$</td>
<td>0.67</td>
<td>39</td>
</tr>
</tbody>
</table>
perimentation with the large set of parameters without having to resort to extensive simulations. We have found that having the graphics-based implementation has given us insight into the important parameters and has lead to additional experimentation and development.

We are continuing this work, adding additional models to the library so that we can look at other performance measures and algorithms. The work on routing described here is some preliminary work that we expect to lead to improved algorithms.

REFERENCES


Elvino S. Sousa (S'80-M'86) was born in the Azores, Portugal, on December 28, 1956. He received the B.A.Sc. degree in engineering science, and the M.A.Sc. degree in electrical engineering from the University of Toronto, Toronto, Canada, in 1980 and 1982, respectively, and the Ph.D. degree in electrical engineering from the University of Southern California, Los Angeles, in 1985.

He was a Teaching Assistant with the University of Toronto, a Research Assistant with the University of Southern California, and is currently with the Department of Electrical Engineering at the University of Toronto. He has performed research in the areas of data transmission and spread spectrum packet radio networks. He is also interested in mobile and personal communication networks, optical communications, and in the performance of distributed computer systems where he has consulted for the Technology Transfer Institute. At the University of Toronto, he has taught graduate courses in error-correcting codes and mobile communications.
Dr. Sousa is a member of Eta Kappa Nu and a member of the Association of Professional Engineers of Ontario. He has held a Natural Sciences and Engineering Research Council of Canada (NSERC) Postgraduate Scholarship, was a recipient of the IEEE Communications Society Student Scholarship, and is currently an NSERC University Research Fellow.

Dr. Silvester is a member of the ACM and has been active in the Technical Committee on Computer Communications since 1981, acting as Conference Coordinator from 1983 to 1985 and Chairman from 1985 to 1987. He was Program Chairman of the First IEEE Computer Communications Workshop in 1986 and is on the INFOCOM Conference Board, was Vice Chairman of INFOCOM 1989, and General Chair for INFOCOM 1990.

John Silvester (M’79 SM’85) was born in Kent, England, in 1950. He received the B.A. and M.A. degrees in mathematics and operations research from the University of Cambridge, Cambridge, England, in 1971 and 1975, respectively, the M.S. degree in statistics and computer science from West Virginia University, Morgantown, in 1973, and the Ph.D. degree in computer science from the University of California, Los Angeles, in 1980.

Since 1979, he has been with the University of Southern California where he is an Associate Professor in the Department of Electrical Engineering and Director of the Computer Engineering Division. He is the author of over 60 technical papers and has lectured both in the United States and abroad. He has consulted for many of the leading networking companies, the U.S. Army, and was Director of Consulting for Technology Transfer Institute from 1984 to 1986, where he is still a leading consultant. He also teaches short courses on packet radio, network design tools, and data communications fundamentals.

Thomas D. Papavassiliou (S’82) was born in Athens, Greece, on November 29, 1960. He received the Diploma in electrical engineering from the National Technical University of Athens, Athens, Greece, in 1983 and the M.S.E.E. degree from the University of Southern California, Los Angeles, in 1985.

From September 1983 to June 1985, he was a Teaching Assistant and from June 1985 to present, a Research Assistant in the Department of Electrical Engineering, University of Southern California, where he is currently completing his Ph.D. degree. From November 1987 to March 1988, he worked for the LinCom Corporation, Los Angeles, CA, where he was involved in a multiple satellite network project. His research interests are communication networks, queueing theory, computer networks, and spread-spectrum systems.

Mr. Papavassiliou is a member of the Technical Chamber of Greece.