Performance Evaluation for a Quasi-Synchronous Packet Radio Network (QSPNET)

Ayan Banerjee, *Member, IEEE*, Ronald A. Iltis, *Senior Member, IEEE*, and Emmanouel A. Varvarigos, *Member, IEEE*

***Abstract*—We propose a new media-access and connection-establishment protocol for an *ad-hoc* quasi-synchronous packet radio network (QSPNET). In the QSPNET, the bandwidth is partitioned into a data channel, used to transmit packets, and a control channel, used to make reservations. Transmitted waveforms in the QSPNET are made quasi-synchronous by using a local GPS clock. The QSPNET uses a novel linear decorrelator receiver for multiuser detection, which permits the reception of quasi-synchronous code division multiple access (QS-CDMA) waveforms. We initially describe the QSPNET and its connection and flow control protocols, giving the rules of transmission and reception followed by all mobiles. We also provide performance results for the case where connection requests are generated at each node of the QSPNET according to a random process over an infinite time horizon. In particular, we obtain results on the achievable throughput and the average delay as a function of the transmission radius, the quasi-synchronous uncertainty interval, the duration of the connections, and the buffer size per node.**

***Index Terms*—*Ad hoc* packet radio networks, quasi-synchronous CDMA, tell-and-go protocol.**

I. Introduction

THE MAJORITY of current and proposed wireless multiple-access networks employ a cellular structure. In the case of the first-generation Analog Mobile Phone System (AMPS) and the second-generation digital TDMA systems, the cellular structure provides increased capacity through frequency reuse. A cellular network model requires dedicated base stations and cannot accommodate widely geographically dispersed users. Packet radio networks (PRNs), on the other hand, consist of a number of distributed packet radio units (PRUs) that communicate with each other by using other mobile users as relays. PRNs are more flexible than cellular networks since they do not require any fixed infrastructure.

Most of the PRN systems that have been investigated in the literature (see [8], [33], [19] and [14]) assume datagram service. Of the algorithms that provide dedicated time slots for data transmission (see [30], [5], [7], and [31]), either an autonomous self-configuring implementation is not a consideration in the design, or the issue of changing topology and connectivity is not sufficiently addressed. The reservation-based DTSAP protocol [21], [22] sets up virtual circuits (VCs) between users and considers changes in the topology, but it requires a roundtrip delay for connection setup before the transmission of any data can take place.

In this paper, we develop a media-access and connection-establishment protocol for an *ad-hoc* quasi-synchronous packet radio network (QSPNET), which operates without any central facility (base station). QSPNET combines the benefits of reservation-ALOHA [29], [16] with quasi-synchronous code division multiple access (QS-CDMA) (see [25], [6], [11], and [9]) modulation to provide high throughput and resistance to fading multipath channels. The channel coding model is different from other recent work being done in [24] and [15]. The multiaccess protocols can be combined with a connection and flow control protocol, which we also describe, to achieve lossless communication. Flow control is exercised by coupling capacity with buffer space, so that when a buffer at a receiving node starts to fill up, a proportional fraction of the capacity is frozen through the transmission of throttle packets. Connection establishment is of the tell-and-go type, and is therefore appropriate for delay-sensitive sessions that cannot tolerate the end-to-end roundtrip delay required by wait-for-reservation protocols.

A key obstacle to the implementation of noncellular CDMA systems is the need for power control. Multiuser detection can lessen the requirements for power control but requires the knowledge or estimates of code delays (see [18] and [32]). In the self-configuring multihop QSPNET, we use a linear decorrelator receiver [11], [9] for multiuser detection that permits reception of QS-CDMA waveforms without the need for power control. This is achieved by using local GPS clocks at the PRUs to make transmissions quasi-synchronous.

We use simulations to obtain results on the performance of the QSPNET with respect to several system parameters. We are particularly interested in two performance criteria: 1) the average delay, which is the time (including connection and queueing delays) that elapses from the time a connection request is generated at a node to the time the last packet of that connection is absorbed at the destination node; and 2) the throughput, which is the average number of sessions absorbed per node and (slot, code) pair in the network. We obtain results on the way delay and throughput change as a function of the transmission radius of the mobiles, the quasi-synchronous interval of received waveforms, the average length of the sessions, and the buffer size at a node.

The remainder of the paper is organized as follows. In Section II, we give an overview of the quasi-synchronous packet

Manuscript received May 10, 2000; revised December 8, 2000 and March 18, 2001; recommended by IEEE/ACM TRANSACTIONS ON NETWORKING Editor T. Todd. This research was supported by the California Department of Transportation through the Center for Interoperability.

A. Banerjee is with Calient Networks, San Jose, CA 95138 USA (e-mail: abanerjee@calient.net).

R. A. Iltis is with the Department of Electrical and Computer Engineering, University of California, Santa Barbara, CA 93106-9560 USA (e-mail: iltis@xanadu.ece.ucsb.edu).

E. A. Varvarigos is with the Department of Computer Engineering and Informatics, University of Patras, Patras, Greece (e-mail: manos@ceid.upatras.gr).

Publisher Item Identifier S 1063-6692(01)08995-6.

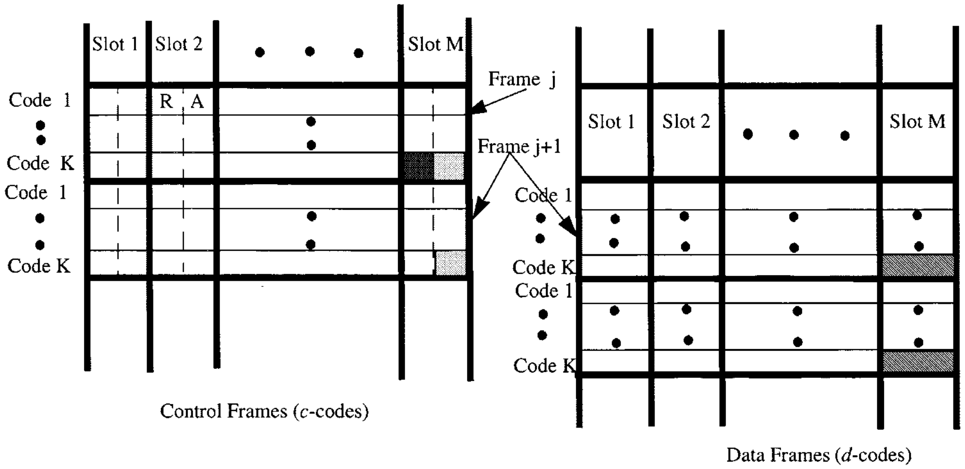

Fig. 1. Illustrates the underlying channel structure in the system. The channel is divided into control and data frames, with M slots in each frame. The control slots are further divided into a forward minislot and a reverse minislot. There are K codes in each frame and there is a one-to-one correspondence between the control codes and the data codes. Requests in the current (slot, c-code) pair of the control frame are made for data slots of the (slot, d-code) pair of the following frame. For example, transmissions in the Mth slot of control frame j of c-code K reserves the Mth slot of data frame $j+1$ of d-code K.

radio network and discuss issues related to synchronization and quasi-synchronous reception. In Section III, we describe the connection and flow control protocol, and give the rules for packet transmission and reception. In Section IV, we provide results on the throughput and delay of the QSPNET. Finally, in Section V, we conclude the paper.

II. Quasi-Synchronous Packet Radio Network (QSPNET)

The proposed QSPNET is an *ad-hoc* packet radio network that departs from the cellular concept and the idea of central coordination of the packet radio units (PRUs). QSPNET is self-organizing, easy to deploy and able to accommodate dynamically changing topology and connectivity. Each PRU (mobile) in the QSPNET is autonomous, and is responsible for performing in a fully distributed way the network management functions that relate to it. In addition to generating its own messages, a mobile is also responsible for forwarding messages of other users; a multihop capability thus being obtained by using other users as relays.

The channel in QSPNET is divided into two parts: the *data channel* used for data transmission, and the *control channel* used for making reservations and providing other control information (see Fig. 1). The underlying time axis in both channels is divided into frames, each containing M slots. In the same slot, up to K codes are available for use, thus allowing multiple users to share the same slot. The data and control channel are implemented using two different types of codes, referred to as the d-codes and the c-codes. The c-codes reserve (slot, d-code) pairs and provide feedback and GPS information related to the location of the transmitter. The actual user data is modulated onto a direct sequence (DS) d-code for transmission. There exists a one-to-one correspondence between a (slot, c-code) pair of the control frame with a data (slot, d-code) pair of the following frame. In the QSPNET, slot and frame boundaries are known to the mobile PRU through a local GPS clock. The effects of the possible inaccuracies of the GPS clock is one of the issues examined in this paper.

A control slot is further subdivided into two minislots, called the *forward minislot* and the *reverse minislot*. In the forward minislot, a transmitter PRU sends out a request packet using a particular c-code, and if successful, it reserves the corresponding data (slot, d-code) pair of the following frame. If two users transmit simultaneously using the same c-code, a collision occurs and a request packet has to be retransmitted after a random time interval. On the successful reception of a request packet, the receiver PRU transmits an ACK using the same c-code. Each *subsequent* slot of the corresponding d-code channel is then reserved for contention-free transmission between the transmitter and the receiver PRU. As discussed in Section III, ACK transmissions are used to acknowledge data packets in order to deal with the hidden terminal problem, that occurs when packets collide without the transmitting nodes hearing the collisions. The reservation period ends when the (slot, d-code) pair is empty for the first time, and the given pair is then open for contention. The mechanism by which PRUs set up virtual circuits is given in Section III.

A. Network Synchronization and Quasi-Synchronous Reception

The c-code transmissions have a low processing gain to reduce the effect of propagation delay on timing uncertainty. The control information is sent out of band with respect to data and the bandwidth required for it is only a small fraction of the total bandwidth. The actual user data is transmitted using pseudonoise sequences (d-codes) at a much higher processing gain (equivalently, data packets contain many more bits than control packets). Due to the difference in processing gain, it is assumed that $T_{cc}/T_{dd} \gg 1$, where T_{cc} and T_{dd} represent the c-code and d-code chip durations, respectively (see Fig. 2). The bit epochs of both the d-code and the c-code transmissions are synchronized to a common GPS received clock. Hence, if the GPS clocks were accurate and all transmitting nodes were equidistant to a given receiver, the received waveform would be

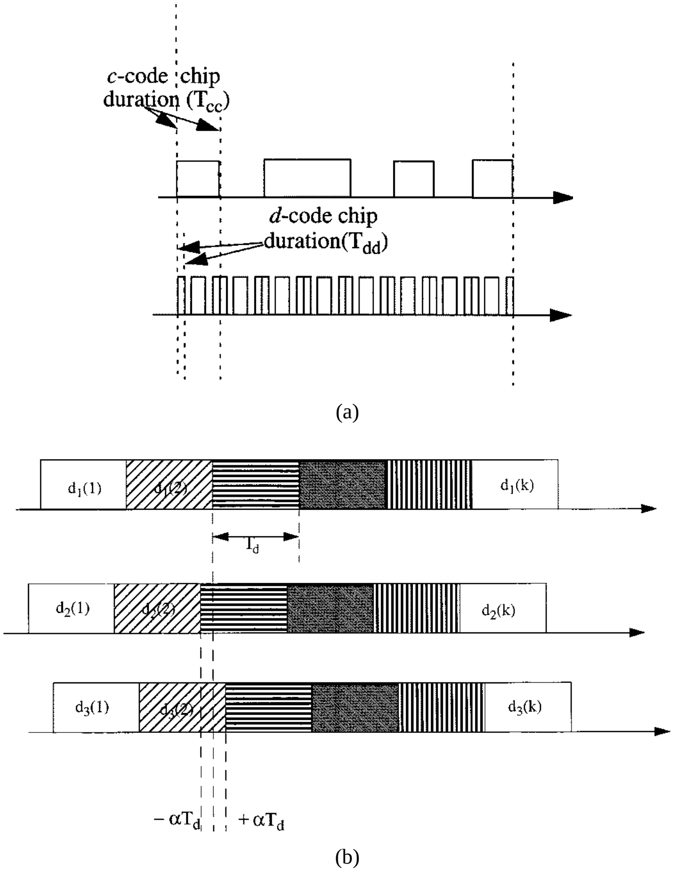

Fig. 2. (a) Illustrates the relationship between the d-code chip duration T_{dd} and the c-code chip duration T_{cc}. (b) Illustrates the timing uncertainty between received signals from different users.

synchronous. However, due to oscillator drifts, GPS timing errors, and unequal propagation delays, the actual received waveform is quasi-synchronous [11].

Specifically, we assume that the user delays in the absence of propagation delay satisfy $T_n \in [-\alpha T_d, \alpha T_d]$, where T_d is the bit duration of the d-code, $\alpha \ll 1$ and the subscript n specifies the delay in transmission from node n to a given receiving node. The total delay, including that due to propagation is then given by $T_n + r_n/c$, where r_n is the distance from node n to the receiver, and c is the velocity of propagation. If we further assume that $T_n + r_n/c \in [-\alpha T_c, \alpha T_c]$, where T_c is the bit duration of the c-code, then the received c-code waveform appears quasi-synchronous (that is, with the uncertainty of the order of a few control chips), since the additional propagation delay is still only a fraction of the c-code bit duration.

In order to reserve slots in a series of frames, a mobile monitors the control code correlator output (see Fig. 3) to determine if a (slot, d-code) pair has already been reserved. If not, the mobile uses the corresponding c-code to send a request packet specifying a user ID, relative x–y position, and the number of successive frames to be reserved, if known. Receiving nodes then use the relative x–y position data to determine the transmission delays which transmitters should use and send this information back in ACK packets. The data transmission at the receiver is also realigned, such that the received d-code waveforms are quasi-synchronous, satisfying $T_n \in [-\alpha T_d, \alpha T_d]$.

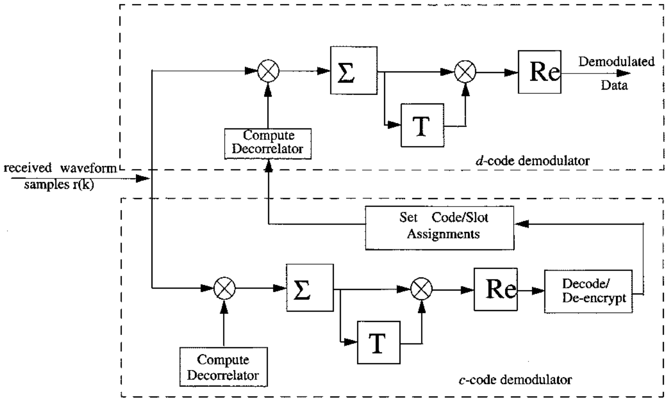

Fig. 3. Decorrelators used in the PRU receivers for quasi-synchronous reception of signals.

B. Decorrelator Structure

The multiuser detector proposed for QSPNET is based on the linear decorrelator design developed in [9] and [11]. The overall demodulators for the c-code and d-code have the same complexity as that of a matched filter and are shown in Fig. 3. The basic method (which is valid for control codes, since propagation delays can be neglected) is first reviewed here, and then modified for varying propagation delays (as is needed for data codes). For simplicity, assume constant propagation delay, with $r_n = r$ corresponding to the distance from users $n = 1, 2, \ldots, N$ to the receiving node. Then, the received signal in continuous-time is

$$r(t) = \sum_{n=1}^{N} a_n d_n(m) s_n(t - T_n) + n(t) \tag{1}$$

for $t \in [mT_c, (m+1)T_c)$, where $a_n \in \mathcal{C}$ is the complex amplitude and phase of user n, and $s_n(t)$ is the particular c-code reserved by user n. The information symbols (c-code bits) are denoted by $d_n(m)$, for user n and data bit m. The additive noise $n(t)$ is circular white Gaussian,[1] with spectral density $2N_o$. Note that since T_n is confined to the quasi-synchronous uncertainty interval $[-\alpha T_c, \alpha T_c]$, intersymbol interference from adjacent bits $\{d_n(m+1), d_n(m-1)\}$ can be neglected. The signal $r(t)$ is low-pass filtered and sampled at the chip rate, yielding the received signal vector (see [9], and [11])

$$\mathbf{r}(m) = \sum_{n=1}^{N} a_n d_n(m) \mathbf{s}_n(T_n) + \mathbf{n}(m) \tag{2}$$

during bit m. It is assumed that the Doppler spread is sufficiently small so a_n is constant. Assume that user 1, or equivalently, data bit $d_1(m)$ is to be detected. Then the decorrelator is defined by the vector

$$\mathbf{h} = [\mathbf{I} - \mathbf{P}_{\mathbf{S}'}]\mathbf{s}_1(T_1). \tag{3}$$

The detected data bit for user 1 is then given by

$$\hat{d}_1(m) = \operatorname{sgn}\left(\operatorname{Re}\left\{\mathbf{h}^H \mathbf{r}(m)\right\}\right). \tag{4}$$

[1] Circular white Gaussian noise is a complex-valued Gaussian noise process, $z(t) = x(t) + jy(t)$, where $x(t)$ and $y(t)$ are independent, zero-mean Gaussian processes, with the same power spectral density.

The projection matrix $P_{S'}$ is given by

$$\mathbf{P}_{\mathbf{S}'} = \mathbf{S}'\left[\mathbf{S}'^{T}\mathbf{S}'\right]^{-1}\mathbf{S}'^{T} \tag{5}$$

where

$$\begin{aligned}\mathbf{S}' = &[\mathbf{s_2}(-U_{\text{int}}T_{cc}),\,\ldots,\,\mathbf{s_2}(U_{\text{int}}T_{cc}),\,\ldots,\\ &\mathbf{s_N}(-U_{\text{int}}T_{cc}),\,\ldots,\,\mathbf{s_N}(U_{\text{int}}T_{cc})].\end{aligned} \tag{6}$$

For the given chip duration, integration, and sampling, $\mathbf{S}' \in R^{P\times(2U_{\text{int}}+1)}$, where P is the number of chips per bit. The matrix S' then corresponds to the undesired signal vectors, at discrete delay values $U_{\text{int}}T_{cc}$ where T_{cc} is the control chip duration and U_{int} (a positive integer) satisfies

$$U_{\text{int}}T_{cc} = \left\lceil \frac{\alpha T_c}{T_{cc}} \right\rceil T_{cc}. \tag{7}$$

When a chip duration integrate-and-dump is used for sampling, it is shown in [9] that

$$\mathbf{s}_n(T_n)^T\mathbf{h} = 0 \tag{8}$$

for any $T_n \in [-U_{\text{int}}T_c, U_{\text{int}}T_c]$, and for $n = 2, 3, \ldots, N$. Thus the undesired users are completely rejected by the decorrelator, for *any* delay value falling within the quasi-synchronous interval. As a result, the detector is resistant to the near–far effect, lessening or even eliminating the need for power control.

The data code decorrelator is now modified for the QSPNET system to include the effects of the propagation delay r_n/c as follows.[2] Recall that the c-code waveform sent out by a receiver specifies the d-code time slot and the location of the user n. In order to detect user 1, let us assume that the receiver is realigned to user 1, whose propagation delay is r_1/c. The receiver sends this information to all other sources that are transmitting to it. All other transmitters add a relative delay $\tau_n = (r_1 - r_n)/c$ before transmitting their data. Since the receiver is realigned to user 1, the received vector can now be written as

$$\mathbf{r}(m) = \sum_{n=1}^{N} a_n d_n(m)\mathbf{s}_n(T_n + r_n/c + \tau_n - r_1/c) + \mathbf{n}(m) \tag{9}$$

where T_n is the error due to oscillator drifts and GPS timing errors and is bounded by $T_n \in [-\alpha T_d, \alpha T_d]$. Note that with the choice of τ_n the decorrelator is given by $\mathbf{h}$ as in (3), with $\mathbf{S}' \in R^{P\times(2U_{\text{int}}+1)}$ redefined as

$$\begin{aligned}\mathbf{S}' = &[\mathrm{s}_2(-U_{\text{int}}T_{dd}),\,\ldots,\,\ldots,\,\mathrm{s}_2(U_{\text{int}}T_{dd}),\,\ldots,\\ &\mathrm{s}_N(-U_{\text{int}}T_{dd}),\,\ldots,\,\mathrm{s}_N(U_{\text{int}}T_{dd})]\end{aligned} \tag{10}$$

where $U_{\text{int}} = \lceil \alpha T_d/T_{dd} \rceil$. The resulting decorrelator is $\mathbf{h} = [\mathbf{I} - \mathbf{P}_{\mathbf{S}'}]\mathrm{s}_1(T_1)$, and again satisfies (8) [9] for any T_n falling in the QS uncertainty interval, with $n = 2, 3, \ldots, N$. Thus, the undesired user d-codes are completely rejected. With the processing gain of a PN code, the interference from neighboring transmitters can thus be mitigated in the absence of tighter chip level synchronization.

C. Bit Error and Packet Error Probabilities

In this section, we obtain the data packet (d-code) error probability. The c-code error rate, which is much lower than the d-code error rate, is ignored. The probability P_b of a received data bit in error for QPSK, BPSK, or MSK modulation using the linear decorrelators described in Section II-B is given by [9]

$$\begin{aligned}P_b &= \frac{1}{2}\,\text{erfc}\left(\sqrt{\frac{E_b}{N_o}\,\mathbf{s}_1^H[\mathbf{I} - \mathbf{P}_{S'}]\mathbf{s}_1}\right)\\ &= \frac{1}{2}\,\text{erfc}\left(\sqrt{\frac{\eta E_b}{N_o}}\right)\end{aligned} \tag{11}$$

where E_b is the received bit energy, $2N_o$ is the spectral density of the circular white Gaussian noise, $\mathbf{s}_1$ is the desired signal vector [detection is assumed without loss of generality for user 1, see (3)] such that $\mathbf{s}_1^H\mathbf{s}_1 = 1$, and $\mathbf{P}_{S'}$ is the projection matrix formed by the undesired users. The factor η giving the loss in signal-to-noise ratio (SNR) is approximated by the following lower bound [11]

$$\begin{aligned}\eta &= \mathbf{s}_1^H[\mathbf{I} - \mathbf{P}_{S'}]\mathbf{s}_1\\ &\geq 1 - \frac{(2U_{\text{int}}+1)(N-1)t_{\max}^2}{1 - [(2U_{\text{int}}+1)(N-1) - 1]t_{\max}}\end{aligned} \tag{12}$$

where U_{int} denotes the number of chips in the QS uncertainty interval, N is the number of users that are within the range of the receiver, and $t_{\max}$ is the maximum correlation between any two codes of the system.

In the results to be presented in Section IV, we assume that power loss between two mobiles varies as the fourth power of the distance [12]. Using (11) and (12), and assuming that the received SNR from a distance R_o km is 5 dB, an upper bound on the bit error probability P_b^d from a distance R_d km, is given by

$$P_b^d = \frac{1}{2}\,\text{erfc}\left(\sqrt{\eta_{\text{lower bound}}10^{0.5}\left(\frac{R_d}{R_o}\right)^{-4}}\right). \tag{13}$$

In our results, we further assume that each data packet of 282 bits is encoded by a Reed–Solomon (63, 47) code resulting in an encoded packet of 378 bits, with an encoding scheme of 6 bits/symbol.[3] Thus, a maximum of $(63-47)/2 = 8$ symbols can be corrected [27]. The probability of a symbol error P_{es}, assuming that transmitted bit errors occur independently, is

$$\begin{aligned}P_{\text{es}} &= P\ \text{(code symbol in error)}\\ &= 1 - P\ \text{(all 6 bits are received correctly)}\\ &= 1 - (1 - P_b^d)^6.\end{aligned} \tag{14}$$

[2] We recognize that the proposed decorrelator structure is only suitable for short-range communication due to frequency-selective multipath. It is possible to develop adaptive QS-CDMA receivers that can jointly estimate the channel and delays [10] and [13]. However, these models are more complex and beyond the scope of the paper.

[3] The choice of the size of the data packets is consistent with the cellular digital packet data (CDPD) standards.

The probability of a code data symbol error P_{ecs} is approximated by [23]

$$P_{\text{ecs}} = \frac{1}{63} \sum_{j=9}^{63} j \binom{63}{j} P_{\text{es}}^{j} (1 - P_{\text{es}})^{63-j}. \tag{15}$$

Finally, the probability that a data packet is received in error, assuming that symbol errors occur independently, is given by

$$\begin{aligned} P\,(\text{packet error}) &= P\,(\text{any data symbol in error}) \\ &= 1 - (1 - P_{\text{ecs}})^{47}. \end{aligned} \tag{16}$$

III. Connection Establishment and the Tell-and-Go Protocol

We make the following assumptions for the PRUs: 1) each PRU is equipped with a GPS clock and a direct sequence spread-spectrum modem; 2) the connectivity between any pair of PRUs is determined by a physical layer protocol (see [1] and [20]) and may be unidirectional or bidirectional; 3) PRUs cannot transmit and receive simultaneously using the same code; 4) each PRU has a unique identification number associated with it and it is known to all other PRUs; and 5) each PRU maintains a topology table for selecting a route.

In most protocols for connection-oriented traffic, a session has to wait for a pretransmission end-to-end roundtrip delay for connection setup (see [21], [2], [4], and [28]) before any data packets can be transmitted. In this section, we develop a connection establishment and flow control protocol, called the tell-and-go (TG) protocol, that uses virtual circuit switching and is appropriate for sessions that cannot tolerate the end-to-end roundtrip delay required for call setup.

An outgoing (slot, d-code) pair at a node is used to implement one hop of a (multihop, in general) virtual circuit (VC) connection, with the corresponding (slot, c-code) pair being used for the setup, control, and tearing down of the VC. In particular, each node monitors the control code correlator output to determine which (slot, d-code) pairs are being used. In the TG protocol, the source node sends a setup packet to the destination to reserve capacity and set the routing tables, followed immediately by the data packets. Therefore, the TG protocol can be viewed as a reservation protocol, where the reservation phase and the data transmission phase overlap. Upon receiving the setup packet, a node can try to reserve an available (slot, d-code) pair by transmitting a request packet during the forward minislot of the corresponding (slot, c-code) pair. If no other node within the transmitter's reception radius is using that pair or is trying to capture it at the same time, and if the request packet is received correctly, the receiver sends an ACK packet during the subsequent reverse minislot. Upon the correct reception of the ACK, the transmitter can start forwarding data packets (which immediately follow the setup packet) using the (slot, d-code) it has captured.

In a packet radio network, it is possible for collisions to occur, without the knowledge of the users generating the collisions. This *hidden terminal* problem arises when two users, A and B, attempt to communicate to a third user C using the same (slot, d-code) pair; where A and B are outside their mutual range, but C is in the range of both. An efficient solution to the hidden terminal problem is to acknowledge each data packet transmission that uses a particular (slot, d-code) pair through the transmission of an ACK packet on the corresponding (slot, c-code) pair. All nodes that are within the radius of reception of node C and therefore have the potential of causing interference at the receiver C are then forbidden from using that (slot, d-code) pair. In other words, it is through the transmission of ACK packets by the receivers that nodes learn which (slot, d-code) pairs are reserved. In this way, a reserved (slot, d-code) pair cannot be reused by transmitters that lie within a sphere of radius equal to the transmission radius $R_{\max}$ from the intended receiver; this results in an increase of the capacity by a factor of up to four over systems that use "busy tones" [26]. (In this latter type of system, all nodes that hear a transmission send a busy tone, and a channel cannot be reused within a radius of roughly $2R_{\max}$ around the intended receiver.)

If the capacity that can be captured by the setup packet at an intermediate node is not sufficient to accommodate the session, or if the rate of the session increases after the connection is established, packets may have to be buffered, and back-pressure is exercised to the previous nodes, and finally to the source node, to control the transmission rate. In particular, if the buffer space available for a session at an intermediate node fills up, a THROTTLE packet is sent on the reverse control minislot of the corresponding (slot, c-code) pair, and it can be distinguished from an ACK packet by using a bit in its header.

Following the transmission of a THROTTLE packet, the corresponding (slot, code) pair is temporarily released, and it is open for contention by users that want to transmit to other nodes. The channel is returned to the original session after B frames, where B is the buffer size per (slot, code) pair at the receiving node. If after B frames the buffers of the receiving node are still full, another THROTTLE packet is sent. It can be seen that in the event a receiver is able to forward the queued packets after sending a THROTTLE packet, it needs at least B frames to empty its buffer. As a result, if the downstream node gains access to the channel within B frames, it has enough packets to transmit before starting to receive packets from its upstream node, and capacity does not unnecessarily stay idle. The state diagrams and the detailed rules followed by transmitting and receiving nodes are given in [3].

A. Fairness and Priority

In this section, we will sometimes refer to the (slot, c-code/d-code) pair in a frame as a channel (consisting, as described earlier, of a forward and a reverse control minislot and a data slot). In order to provide fair network access to all users so that heavy users do not monopolize the available capacity, while at the same time providing different priorities and transmission rate guarantees to applications that require them, we can use a variation of the multiaccess protocol described in the previous section, where each node "owns" a particular channel (or set of channels). When an owner is not using its channel, other nodes can capture it by contention, but when the owner wants it back, it simply transmits a request on the corresponding (slot, c-code) pair, and all sources hearing an ACK or collision in the subsequent reverse minislot are forbidden to use that channel in the next frame, letting the owner capture it. In this way, each node is guaranteed a minimum transmission rate even in the presence of congestion.

The problem of deleting and adding new owners, which arises from the mobility of the users, can be addressed by having a node periodically affirm its ownership of a channel through the transmission of (dummy) request packets, possibly forcing a collision. When two mobiles that own the same channel approach each other, one of them will have to abandon the channel (this is similar to the handover problem in cellular networks). A user who loses, due to this slow drifting of the mobiles, a channel that it owns, can capture a new one by transmitting on a (slot, c-code/d-code) channel that is not owned by other users, forcing any temporary user to cease transmission. When a node is unable to capture new capacity (either as an owner or as a temporary user) during the duration of a call, "soft hand-over" can be implemented by letting nodes share the same channel until one of them can capture another appropriate channel.

IV. Performance Evaluation of QSPNET

In our simulations, mobiles are distributed over a two-dimensional (2-D) region according to a Poisson distribution with density λ per unit area, so that the probability of finding k mobiles in a given area A is given by

$$\Pr(k \text{ mobiles in an area } A) = \frac{(\lambda A)^k e^{-\lambda A}}{k!}.$$

Each mobile can be the source of at most one session at any given time and session destinations are uniformly distributed over all the nodes of the network except for the source. We consider unicast communication in this paper. Each PRU maintains a topology table and selects a route according to the most forward routing strategy [8]. Note that any routing algorithm can be used with the media access and connection control protocols that have been developed. When more than one mobile located within the reception radius of a PRU transmit simultaneously using the same code, all packets are destroyed.[4]

A source node can be in the ON or the OFF state. When a node is in the OFF state it does not generate any packets. A source moves from the OFF state to the ON state during a slot (that is, a new session is generated) with probability γ. When a node is in the ON state, it generates a packet every frame (that is, every M slots) according to a Bernoulli process with probability δ. Thus, the average number of packets in a session is $L = 1/(1-\delta)$. The offered load p_o per node, which is the average number of packets generated at a node per slot, is given (see [3] for derivation) by

$$p_o = \frac{1}{M + \dfrac{1-\delta}{1-\gamma}}. \tag{17}$$

In our simulation results, we assume that the data channel access rate D_p is 20 packets per second, which results in a frame duration of $T_f = 5.0 \times 10^{-2}$s. Each data packet has 282 information bits and is encoded by a Reed–Solomon (63, 47) code resulting in an encoded packet of 378 bits, assuming an encoding scheme of 6 bits/symbol. There are $K = 3$ d-codes (or c-codes) in the system. The number of data slots in a frame is taken to be $M = 8$, which results in an aggregate data bit rate D_b of 60.48 kb/s for each code of the data channel. Essentially, M and K define the granularity of the rates [they have to be multiples of $C/(MK)$, unless users are allowed to share a (slot, d-code) pair using, for example, technologies such as TDM, etc.], this gives a mechanism to choose the product MK. A large value of MK gives better granularity (more and more efficient sharing since users reserve the smallest multiple of $C/(MK)$ that is larger than their desired rate) but also has disadvantages (complexity and buffer increase and also light load delay increases). In this simulation, we assume that each user needs only one (slot, d-code) pair.

TABLE I
Simulation Parameters Used

Data Channel Access Rate D_p (per code)	20 packets/sec
Frame Duration T_f	50.0 ms
Data Bit Rate D_b (per code)	60.48 Kbits/sec
Data Packet Size	378 bits
Data Slots per Frame M	8
Data Slot Duration	6.25 ms
Control Packet Size	93 bits
Control MiniSlot Duration	3.125 ms
Area under Simulation	3 km ×3 km
Density of PRUs	10.8 users/km^2
Number of Codes K in the system	3
Max Correlation t_{max} between any two codes	0.06

The data bit duration is $T_d = 1.6534 \times 10^{-5}$ s, and since there are 511 chips per data bit, the data chip duration is $T_{dd} = 3.3257 \times 10^{-8}$ s. Each control packet has 45 information bits and is encoded using a Reed–Solomon (31, 15) code resulting in an encoded packet of 93 bits, assuming an encoding scheme of 3 bits/symbol. The control bit duration is $T_c = 3.3602 \times 10^{-5}$ s, and since there are 127 control chips per bit, the control chip duration is $T_{cc} = 2.645 \times 10^{-7}$ s. We performed our simulations for a 3 km × 3 km area with $\lambda = 10.8$ mobiles/km^2. We summarize the parameters in Table I.

A mobile can transmit to other mobiles that are within a distance of $R_{\max}$ km from it. Fig. 4 illustrates the average number of hops $\overline{h}$ that a session needs to make to move from a source node to its destination node, for several values of the transmission radius $R_{\max}$, for a particular instance of the network. Note that the network under simulation partitions into two or more subsets if the transmission radius $R_{\max}$ is set to values less than 0.51 km. In Fig. 4, we also illustrate the maximum number of hops that a session needs to make for varying values of the transmission radius $R_{\max}$.

We define the *single-hop throughput* H as the average number of successful transmissions per node and (slot, code) pair. We define the *session throughput* S as the average number of successful transmissions per node and (slot, code) pair of packets that reach their destination and are the last packet of

[4] Note that, in practice, a PRU may be able to "capture" a packet even in the presence of other simultaneous transmissions that use the same code; in that case the network throughput increases, but no changes are needed in our protocol.

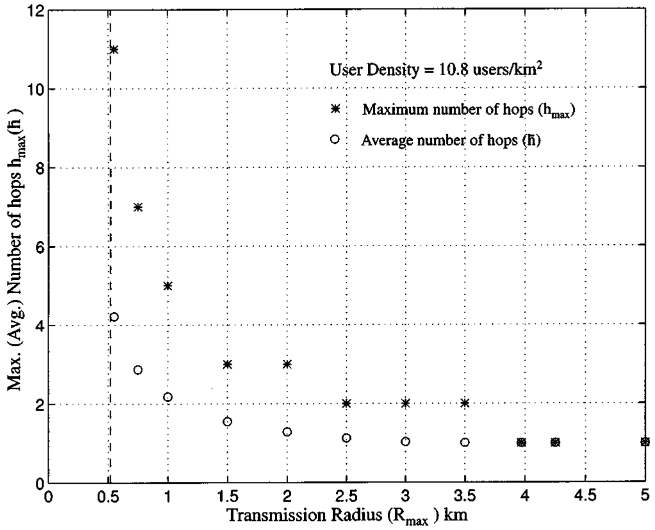

Fig. 4. Illustrates the average and maximum number of hops for the network under consideration. The network becomes disconnected for transmission radius $R_{\max} < 0.51$ km (dotted line).

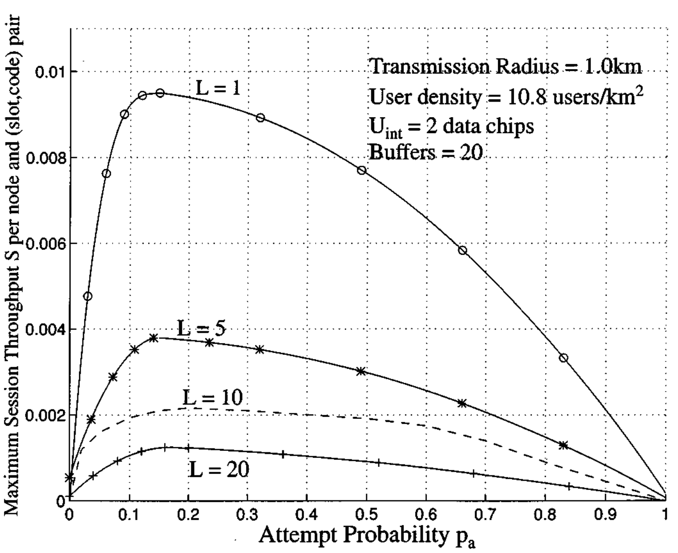

Fig. 5. Illustrates the variation in the maximum session throughput S as the attempt probability p_a is varied for different values of the average session length L. Simulation parameters that have been fixed are transmission radius $R_{\max} = 1.0$ km, SNR of 5 db at a distance $R_o = 0.95$ km, uncertainty interval of duration $U_{\rm int} = 2$ data chips, number of slots $M = 8$ in a frame and number of codes $K = 3$ in the system.

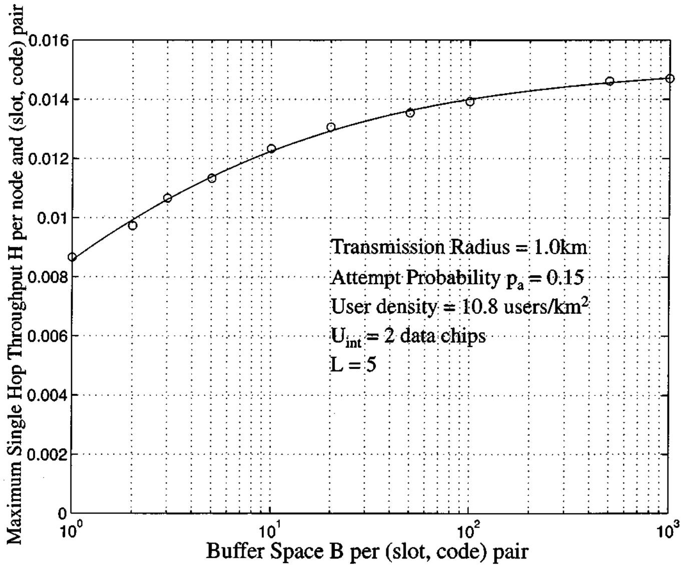

Fig. 6. Illustrates the maximum single-hop throughput H for varying buffer sizes B. Simulation parameters that have been fixed are transmission radius $R_{\max} = 1.0$ km, SNR of 5 db at a distance $R_o = 0.95$ km, probability of attempt in an empty slot $p_a = 0.15$, average length of a generated session $L = 5$ packets, uncertainty interval of duration $U_{\rm int} = 2$ data chips, number of slots $M = 8$ in a frame and number of codes $K = 3$ in the system.

their sessions (that is, session completions). We also define the *average total delay* D of a session as the time (measured in frames) that elapses between the time a session is generated at a node and the time the last packet of the session reaches its destination (including the connection delay and all queueing delays at intermediate nodes).

An important parameter that controls the session throughput S is the *attempt probability* p_a, which is the probability with which an active node (that is, a node that wants to capture an outgoing channel for an originating or transit session) transmits a request packet in an empty (slot, code) pair. Fig. 5 illustrates the maximum session throughput S as a function of the attempt probability p_a and the average session length L. We observe that the throughput increases and then decreases with increasing values of p_a. This is because for small values of p_a the channel remains mostly idle, whereas for high values of p_a, increased collision in the channel decreases the throughput. We further observe that the value of p_a that maximizes the session throughput S changes only slightly with the average session length L. Of course, the optimal value of the attempt probability does depend on the transmission radius $R_{\max}$ and the user density λ, because these determine the average number of neighbors of each mobile. For example, for $R_{\max} = 1$ km, and $\lambda = 10.8$ users/km^2, the optimal value for p_a is 0.18. In all simulation results to follow, the value of p_a has been optimized to obtain the maximum session throughput (for example, for $R_{\max} = 0.55$, 0.75, 1.5, and 2.0 km, the values of the attempt probability that maximize the throughput are $p_a = 0.26$, 0.22, 0.16, and 0.10, respectively, for user density $\lambda = 10.8$ users/km^2). In a practical implementation, an algorithm to optimize on the attempt probability may be run; however, we do not consider it in this paper.

We let B be the number of buffer spaces for each (slot, d-code) pair, that is, there are a total of MBK buffer spaces at a node. Figs. 6 and 7 illustrates the session throughput S and the single-hop throughput H for different values of the buffer size B. A high value of H (which is local in nature) does not imply that S is large, since S depends also on the number of hops in the sessions path (and therefore on the maximum transmission radius $R_{\max}$). The relation between S and H is given as $H = S\bar{h}L$. With decreasing buffer size B, both the maximum single-hop throughput H and the maximum session throughput S decreases. The decrease in throughput is due to the effect of the THROTTLE packets, which reduce the accepted load into the system in order to prevent packet loss. However, note that small buffer sizes are sufficient in providing relatively high throughput, and buffer sizes larger than 100 packets per (slot, code) pair increase the throughput only slightly.

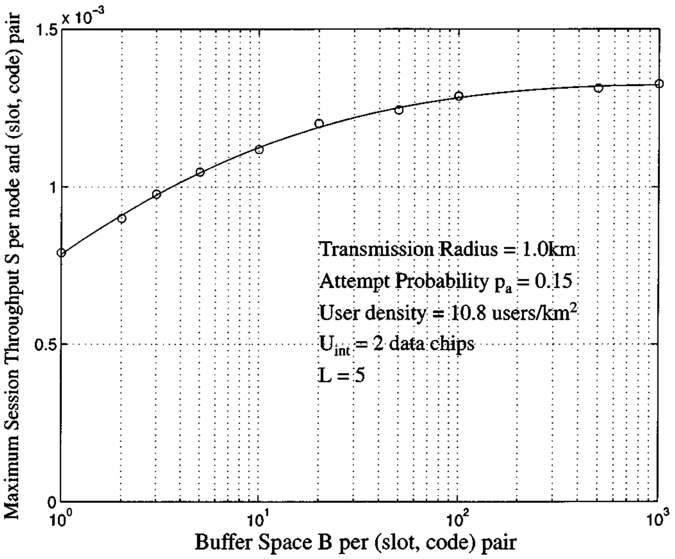

Fig. 7. Illustrates the maximum session absorption throughput S for varying buffer sizes B. Simulation parameters that have been fixed are transmission radius $R_{\max} = 1.0$ km, SNR of 5 db at a distance $R_o = 0.95$ km, probability of attempt in an empty slot $p_a = 0.15$, average length of a generated session $L = 5$ packets, uncertainty interval of duration $U_{\text{int}} = 2$ data chips, number of slots $M = 8$ in a frame and number of codes $K = 3$ in the system.

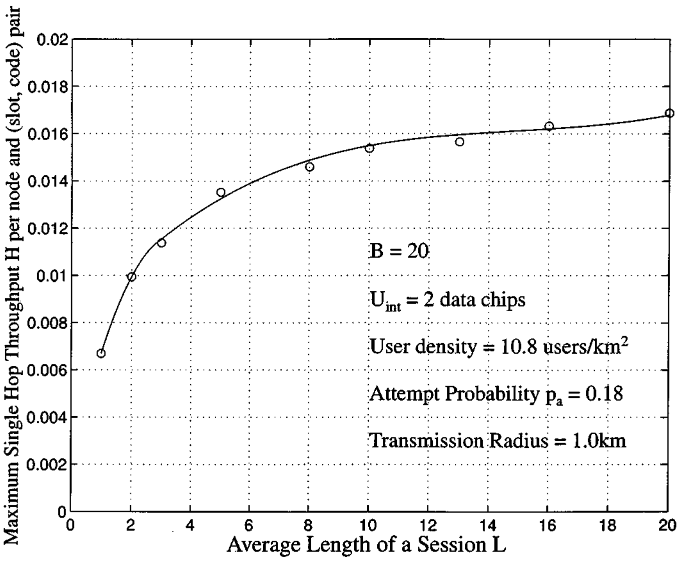

Fig. 9. Illustrates the maximum single-hop throughput H for varying average length L of a session. Simulation parameters that have been fixed are transmission radius $R_{\max} = 1.0$ km, SNR of 5 db at a distance $R_o = 0.95$ km, probability of attempt in an empty slot $p_a = 0.18$, buffer size $B = 20$ packets per (slot, code) pair, uncertainty interval of duration $U_{\text{int}} = 2$ data chips, number of slots $M = 8$ in a frame and number of codes $K = 3$ in the system.

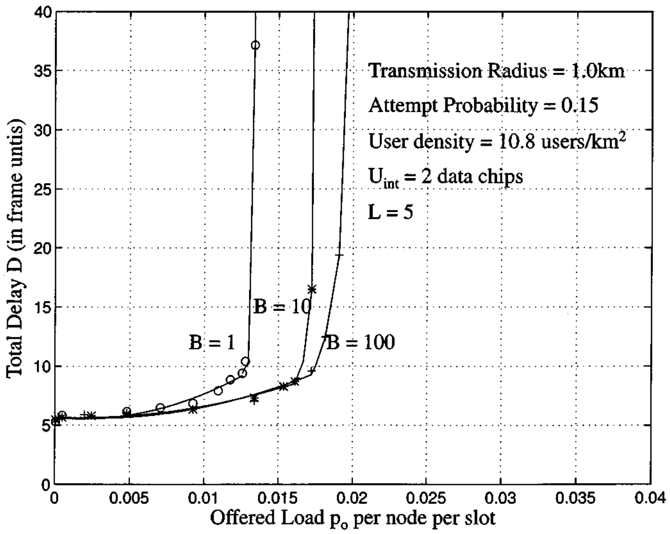

Fig. 8. Illustrates the average connection delay D for varying B buffer sizes. Simulation parameters that have been fixed are transmission radius $R_{\max} = 1.0$ km, SNR of 5 db at a distance $R_o = 0.95$ km, probability of attempt in an empty slot $p_a = 0.15$, average length of a generated session $L = 5$ packets, uncertainty interval of duration $U_{\text{int}} = 2$ data chips, number of slots $M = 8$ in a frame and number of codes $K = 3$ in the system.

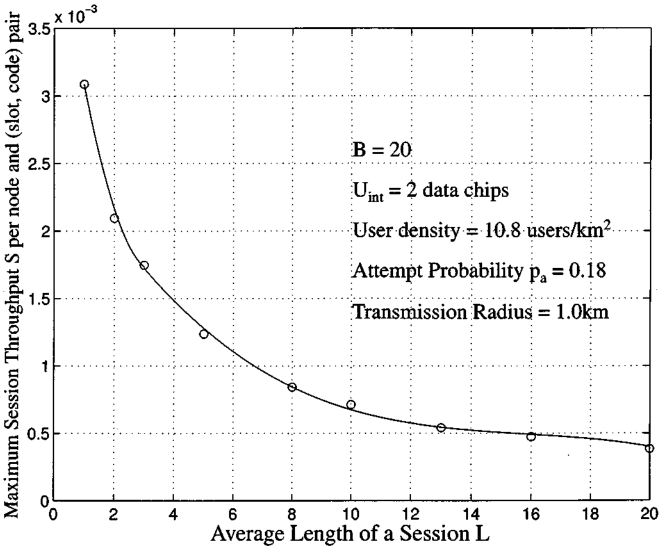

Fig. 10. Illustrates the maximum session absorption throughput S for varying average length L of a session. Simulation parameters that have been fixed are transmission radius $R_{\max} = 1.0$ km, SNR of 5 db at a distance $R_o = 0.95$ km, probability of attempt in an empty slot $p_a = 0.18$, buffer size $B = 20$ packets per (slot, code) pair, uncertainty interval of duration $U_{\text{int}} = 2$ data chips, number of slots $M = 8$ in a frame and number of codes $K = 3$ in the system.

Fig. 8 illustrates the average total delay D for different values of the buffer size B. The average delay becomes unbounded when the offered load to the system is increased. The point at which the asymptote appears (maximum throughput) increases with the buffer size.

Figs. 9 and 10 illustrates the maximum session throughput S and the maximum single-hop throughput H as a function of the offered load for different values of the average length session L and buffer size $B = 20$. As the session length L increases, the maximum single-hop throughput H also increases. The reason for this is the implicit reservation nature of our protocol which limits the number of collisions, and increases the efficiency of the protocol. When the session length L increases, fewer sessions need to compete for the wireless medium to maintain the same level of single-hop throughput, and therefore the session throughput S decreases.

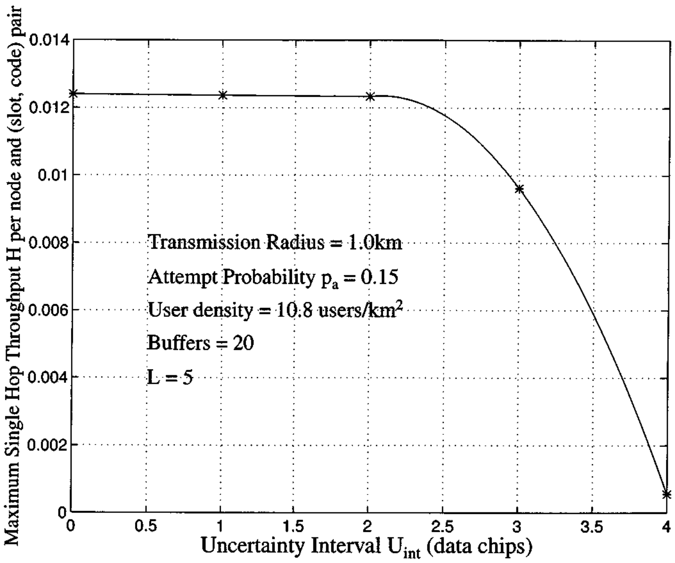

Fig. 11. Illustrates the maximum single-hop throughput H as a function of the uncertainty interval U_{int}. Simulation parameters that have been fixed are transmission radius $R_{\max} = 1.0$ km, probability of attempt in an empty slot $p_a = 0.15$, average length of a generated session $L = 5$ packets, buffer size $B = 20$ packets per (slot, code) pair, number of slots $M = 8$ in a frame, number of codes $K = 3$ in the system, maximum correlation between any two codes of the system $t_{\max} = 0.06$, and SNR of 5 db at a distance $R_o = 0.95$ km.

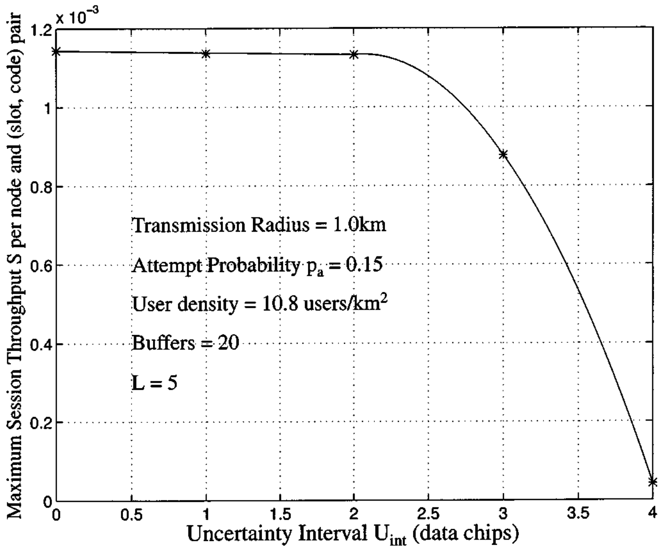

Fig. 12. Illustrates the maximum session throughput S as a function of the uncertainty interval U_{int}. Simulation parameters that have been fixed are transmission radius $R_{\max} = 1.0$ km, probability of attempt in an empty slot $p_a = 0.15$, average length of a generated session $L = 5$ packets, buffer size $B = 20$ packets per (slot, code) pair, number of slots $M = 8$ in a frame, number of codes $K = 3$ in the system, maximum correlation between any two codes of the system $t_{\max} = 0.06$, and SNR of 5 db at a distance $R_o = 0.95$ km.

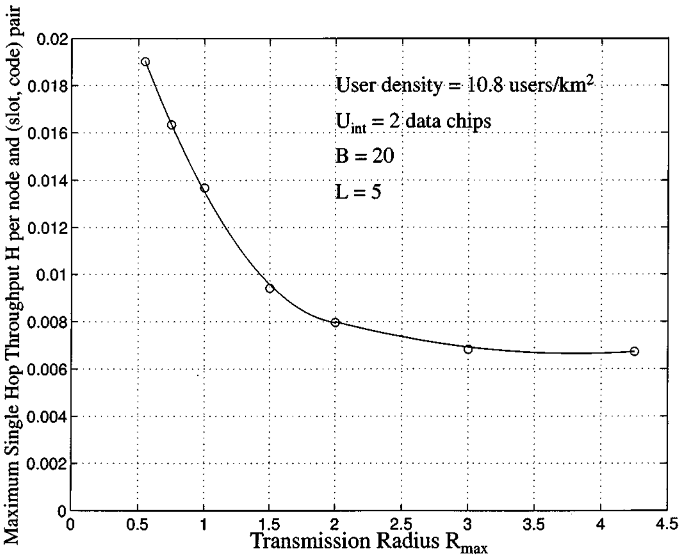

Fig. 13. Illustrates the maximum single-hop throughput H for varying transmission radius $R_{\max}$ and attempt probability. Simulation parameters that have been fixed are average length of generated session of $L = 5$ packets, buffer size $B = 20$ packets per (slot, code) pair, uncertainty interval of quasi-synchronous reception $U_{\mathrm{int}} = 2$, number of slots $M = 8$ in a frame, number of codes $K = 3$ in the system, and SNR of 5 db at distances of $R_o = 0.95, 0.712$, and 0.52 km for transmission radius of 1, 0.75, and 0.55 km.

Figs. 11 and 12 illustrate the session throughput S as a function of the offered load for different values of the uncertainty interval U_{int}, which is due to the inaccuracy of the GPS clocks. As expected, the throughput decreases when the quasi-synchronous uncertainty interval increases. The drop in the throughput when the value of the uncertainty interval increases is due to the increase in the packet reception error probability and the number of retransmissions. For the data chip duration $T_{dd} = 3.3257 \times 10^{-8}$s that we have assumed, the values of $U_{\mathrm{int}} = 0$, 1, 2, 3, and 4 correspond to uncertainty durations of ±0, ±33.3, ±66.6, ±99.9, and ±133.2 ns, respectively. Note that commercial GPS clocks have timing accuracies of the order of 50 ns [17], which for the parameters of our simulation corresponds to $U_{\mathrm{int}} \approx 2$. It is possible to decrease the uncertainty duration fraction α by decreasing the data rate (hence the data symbol duration if the same data chip duration is used); this is a design parameter of the system. Furthermore, it is assumed that the relative delay in signal reception is well within this duration; hence, $R_{\max}$ is design parameter of the system. For the simulation parameters chosen, uncertainty duration yields around 30 km, which is beyond the transmit power of the PRUs.

The transmitter power available at a node determines the transmission radius, which is an important network parameter in minimizing interference, ensuring network connectivity, and dictating the number of hops a session needs to make on its path to its destination. The use of small transmission radius enables the efficient reuse of spectrum and decreases the interference with other transmissions, but it also increases the number of hops a packet has to travel to reach its destination. In Figs. 13 and 14, we provide simulation results that illustrate the variation in the single-hop throughput H and the session throughput S as a function of the transmission radius. As expected, the single-hop throughput H increases when the transmission radius decreases. For values of $R_{\max} \geq 4.25$ km, the network under simulation reduces to a single-hop network and the maximum packet throughput per (slot, code) pair for the

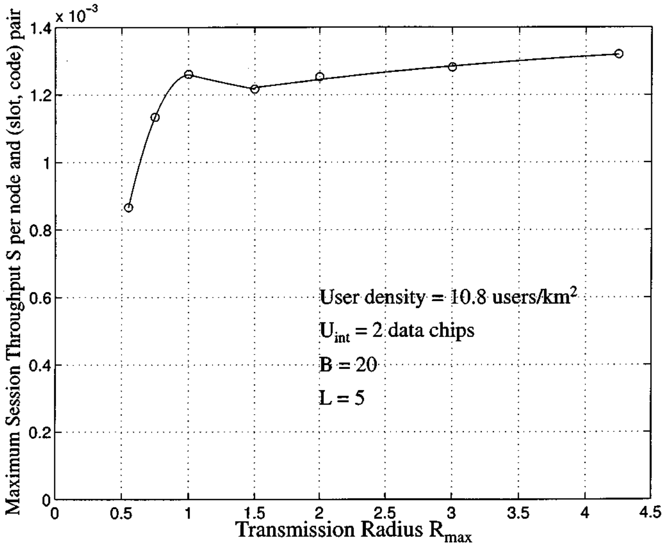

Fig. 14. Illustrates the maximum session absorption throughput S for varying transmission radius $R_{\max}$ and attempt probability. Simulation parameters that have been fixed are average length of generated session of $L = 5$ packets, buffer size $B = 20$ packets per (slot, code) pair, uncertainty interval of quasi-synchronous reception $U_{\rm int} = 2$, number of slots $M = 8$ in a frame, number of codes $K = 3$ in the system, and SNR of 5 db at distances of $R_o = 0.95, 0.712,$ and 0.52 km for transmission radius of 1, 0.75, and 0.55 km.

entire 97-node network is (97×0.0067) 0.65. This is consistent with the analytical results of the single-hop reservation-aloha (P1) protocol derived in [16], where the maximum packet throughput per slot is found to be $(1/(1 + (e/L))$ $(L = 5$ for the results of Fig. 14). The session throughput S, however, decreases (though, not monotonically) when $R_{\max}$ decreases, because the increase in H is more than offset by the increase in the number of hops required for a packet to reach its destination.

V. Conclusion

We proposed a new wireless architecture, called QSPNET, and introduced a media-access and connection-establishment protocol for it. QSPNET is self-organizing, easy to deploy and operates without any fixed infrastructure (base stations). We use a novel linear decorrelator for multiuser detection, which permits the reception of quasi-synchronous CDMA waveforms. The tell-and-go protocol used in QSPNET is a reservation-based connection-oriented communication protocol that achieves lossless communication by coupling buffers with the available capacity, and is suited for sessions that cannot tolerate the end-to-end roundtrip delay.

We also provided performance results for the case where connection requests are randomly generated at each node of the network. Our results indicate that in the stable region of operation, relatively small buffers are sufficient in providing high throughput. Furthermore, the implicit reservation feature of our protocol is useful in maintaining a high session throughput. We have also shown that commercial GPS clocks can be used in our decorrelators without a substantial decrease in the achievable throughput. Our results have also shown that an increase in the transmission radius translates to an increase in throughput at the expense of more transmission power.

References

[1] J. J. Garcia-Luna-Aceves, "A fail-safe routing algorithm for multihop packet radio networks," in *Proc. IEEE INFOCOM*, Miami, FL, Apr. 1986, pp. 434–443.

[2] B. Awerbuch, I. Cidon, I. S. Gopal, and M. Kaplan *et al.*, "Distributed control for PARIS," in *Proc. 9th Annu. ACM Symp. Principles of Distributed Computation*, Quebec, Canada, Aug. 1990, pp. 145–149, 22–24.

[3] A. Banerjee, "Dynamic communication algorithms in wireline and wireless networks and their performance evaluation," Ph.D. dissertation, Dept. of Electrical and Computer Engineering, Univ. of California, Santa Barbara, 1998.

[4] P. E. Boyer and D. P. Tranchier, "A reservation principle with applications to the ATM traffic control," *Computer Networks and ISDN Systems*, vol. 24, no. 4, pp. 321–324, May 1992.

[5] I. Chlamtac and S. Kutten, "A spatial-reuse TDMA/FDMA for mobile multihop radio networks," in *Proc. IEEE INFOCOM*, 1985, pp. 389–394.

[6] V. M. DaSilva and E. S. Sousa, "Multicarrier orthogonal CDMS signals for quasi-synchronous communication systems," *IEEE J. Select. Areas Commun.*, vol. 12, pp. 842–852, June 1994.

[7] A. Ephremides and T. Truong, "Distributed assignment algorithm for efficient and interference-free broadcasting in radio networks," in *Proc. IEEE INFOCOM*, 1988, pp. IIC.2.1–IIC.2.6.

[8] T.-C. Hou and V. O. K. Li, "Transmission range control in multihop radio networks," *IEEE Trans. Commun.*, vol. COM-34, pp. 38–44, Jan. 1986.

[9] R. A. Iltis, "Demodulation and code acquisition using decorrelator detectors for QS-CDMA," *IEEE Trans. Commun.*, vol. 44, pp. 1553–1560, Nov. 1996.

[10] ——, "Exact and approximate maximum-likelihood parameter estimation for quasi-synchronous CDMA signals," *IEEE Trans. Commun.*, vol. 48, pp. 1208–1216, July 2000.

[11] R. A. Iltis and L. Mailaender, "Multiuser detection of quasi-synchronous CDMA signals using linear decorrelators," *IEEE Trans. Commun.*, vol. 44, pp. 1561–1571, Nov. 1996.

[12] W. C. Jakes, *Microwave Mobile Communications*. New York: Wiley, 1974.

[13] K. Kim and R. Iltis, "Joint detection and channel estimation algorithms for QS-CDMA signals over time-varying channels," *IEEE Trans. Commun.*, submitted for publication.

[14] L. Kleinrock and J. Silvester, "Optimum transmission radii for packet radio networks or why six is a magic number," in *Proc. IEEE Natl. Telecommunications Conf.*, Birmingham, AL, Dec. 1978, pp. 4.3.1–4.3.5.

[15] S. Kumar and S. Nanda, "High data-rate communications for cellular networks using CDMA: Algorithms and performance," *IEEE J. Select. Areas Commun.*, vol. 17, pp. 472–492, Mar. 1999.

[16] S. S. Lam, "Packet broadcast networks—A performance analysis of the R-ALOHA protocol," *IEEE Trans. Comput.*, vol. C-29, pp. 596–603, July 1980.

[17] The GPS Guide. Lowrance Electronics Inc., Tulsa, OK. [Online]. Available: http://www.lowrance.com/lowrance/gps/gpshome.htm

[18] R. Lupas and S. Verdu, "Near–far resistance of multiuser detectors in asynchronous channels," *IEEE Trans. Commun.*, vol. 38, pp. 496–508, Apr. 1990.

[19] H.-J. Perz and B. Walke, "Adjustable transmission power for mobile communications with omnidirectional and directional antennas in an one- and multi-hop environment," in *41st Proc. IEEE Vehicular Technology Conf.*, 1991, pp. 630–635.

[20] C. A. Pmalaza-Raez, "A distributed routing algorithm for multihop packet radio networks with uni- and bi-directional liks," *IEEE Trans. Veh. Technol.*, vol. 44, pp. 579–585, Aug. 1995.

[21] L. C. Pond and V. O. K. Li, "A distributed media access protocol for packet radio networks and performance analysis—Part 1: Network capacity," *Intl. J. Commun. Syst.*, vol. 8, pp. 27–47, Jan.–Feb. 1995.

[22] ——, "A distributed media access protocol for packet radio networks and performance analysis—Part 2: Network setup time and data rate," *Intl. J. Commun. Syst.*, vol. 8, pp. 49–63, Jan.–Feb. 1995.

[23] J. G. Proakis, *Digital Communications*, Third ed. New York: McGraw-Hill, 1996.

[24] S.-W. Kim and A. Goldsmith, "Truncated power control in CDMA communications," *IEEE Trans. Commun.*, vol. 12, pp. 837–841, Apr. 2000.

[25] N. Seuhiro, "A signal design without co-channel interference for approximately synchronized CDMA systems," *IEEE J. Select. Areas Commun.*, vol. 12, pp. 837–841, June 1994.

[26] M. Sidi and A. Segall, "A busy-tone-multiple-access-type scheme for packet-radio-networks," in *Performance of Data Communication Systems and Time Applications*. Amsterdam, The Netherlands: North-Holland, 1981.

[27] B. Sklar, *Digital Communications, Fundamentals and Applications*. New York: McGraw-Hill, 1988.

[28] H. Suzuki and F. Tobagi, "Fast bandwidth reservation scheme with multi-link and multi-path routing," *Comput. Commun. Rev.*, vol. 24, no. 5, pp. 81–106, Oct. 1994.

[29] S. Tasaka, "Stability and performance of the R-ALOHA packet broadcast system," *IEEE Trans. Comput.*, vol. C-32, pp. 717–726, Aug. 1983.

[30] T. D. Todd and E. L. Hahne, "Multiaccess mesh (multimesh) network," *IEEE/ACM Trans. Networking*, vol. 5, pp. 181–189, Apr. 1997.

[31] T. V. Truong, "TDMA in mobile radio network: An assessment of certain approaches," in *IEEE GLOBECOM*, 1984, pp. 16.1.1–16.1.4.

[32] S. Verdu, "Minimum probability of error for asynchronous Gaussian multiple access channels," *IEEE Trans. Inform. Theory*, vol. IT-32, pp. 85–96, Jan. 1986.

[33] V. Wong and C. Leung, "Transmission strategies in multihop packet radio networks," in *Canadian Conf. Electrical and Computer Engineering*, 1993, pp. 1004–1008.

Ayan Banerjee (M'96) received the B.Tech. degree from the Indian Institute of Technology, Kharagpur, India, in 1992 and the M.S. and Ph.D. degrees from the University of California at Santa Barbara in 1994 and 1998, respectively.

He is currently a Systems Engineer with Calient Networks, where he assists in network and switch architecture design and analysis. Prior to joining Calient, he was a network scientist at BBN Technologies where he was involved in the design, analysis, and implementation of networking architectures and protocols. His research interests include switch architectures, constraint-based routing, topology control, dynamic communication algorithms, and software radios.

Dr. Banerjee was awarded the General Affiliates Dissertation Award for his work on the design and performance evaluation of dynamic communication algorithms on wireline and wireless networks.

Ronald A. Iltis (S'83–M'84–SM'91) received the B.A. degree in biophysics from The Johns Hopkins University, Baltimore, MD, in 1978, the M.Sc. degree in engineering from Brown University, Providence, RI, in 1980, and the Ph.D. degree in electrical engineering from the University of California, San Diego, in 1984.

Since 1984, he has been on the faculty of the University of California, Santa Barbara, where he is currently a Professor in the Department of Electrical and Computer Engineering. His current research interests are in spread-spectrum communications, blind equalization, multisensor/multitarget tracking, and neural networks. He has also served as a Consultant to government and private industry in the areas of adaptive arrays, neural networks and spread-spectrum communications.

Dr. Iltis was formerly an Editor for the IEEE TRANSACTIONS ON COMMUNICATIONS. He received the Fred W. Ellersick Award for best paper at the IEEE MILCOM conference in 1990.

Emmanouel A. Varvarigos (M'93) was born in Athens, Greece, in 1965. He received the Diploma in electrical and computer engineering from the National Technical University of Athens in 1988, and the M.S. and Ph.D. degrees in electrical engineering and computer science from the Massachusetts Institute of Technology, Cambridge, MA, in 1990 and 1992, respectively.

In 1990, he was a Researcher at Bell Communications Research, Morristown, NJ. From 1992 to 1999, he was an Assistant and later an Associate Professor with the Department of Electrical and Computer Engineering at the University of California, Santa Barbara. During 1998–1999, he was a Visiting Associate Professor at the Electrical Engineering Department at Delft University of Technology, the Netherlands. In 1999, he became a Professor with the Department of Computer Engineering and Informatics at the University of Patras, Patras, Greece, where he is currently the Director of the Hardware and Computer Architecture Division and the Head of the Communication Networks Lab. His research activities are in the areas of protocols and algorithms for high-speed networks, all-optical networks, high-performance switch architectures, parallel and distributed computing, interconnection networks, VLSI layout design, performance evaluation, and *ad-hoc* networks.

Dr. Varvarigos has received an NSF research initiation award and the First Prize in the national competition in Mathematics.