Multicluster, mobile, multimedia radio network *

Mario Gerla and Jack Tzu-Chieh Tsai

Computer Science Department, University of California, Los Angeles, Los Angeles, CA 90024, USA

Abstract. A multi-cluster, multi-hop packet radio network architecture for wireless adaptive mobile information systems is presented. The proposed network supports multimedia traffic and relies on both time division and code division access schemes. This radio network is not supported by a wired infrastructure as conventional cellular systems are. Thus, it can be instantly deployed in areas with no infrastructure at all. By using a distributed clustering algorithm, nodes are organized into clusters. The clusterheads act as local coordinators to resolve channel scheduling, perform power measurement/control, maintain time division frame synchronization, and enhance the spatial reuse of time slots and codes. Moreover, to guarantee bandwidth for real time traffic, the architecture supports virtual circuits and allocates bandwidth to circuits at call setup time. The network is scalable to large numbers of nodes, and can handle mobility. Simulation experiments evaluate the performance of the proposed scheme in static and mobile environments.

1. Introduction

Current wireless systems are constrained by fixed bandwidth allocation (on a per connection basis), fixed network configuration such as cellular systems, and by a reliance on a tethered infrastructure of fixed base stations or servers that are linked by a wireline network. In some cases, such as emergency disaster relief or battlefield communications, when the wireline network is not available, this type of architecture is infeasible. In addition, current systems support only a narrow range of services representing applications that are generally narrowband in nature. To overcome these constraints and advance the state of the art in Wireless Adaptive Mobile Information Systems (WAMIS), we propose a novel architecture which enables rapid deployment and dynamic reconfiguration of a network of wireless stations. We use SS-CDMA (Spread Spectrum Code Division Multiple Access) to allow flexible spectrum sharing, and to combat interference and multipath fading [1,2]. In consideration of SS-CDMA and efficient spatial reuse of communication bandwidth, power control and multi-hop support become critical and essential [1]. Thus, we develop distributed algorithms for adaptive power adjustment and multihop routing.

A key requirement in WAMIS is the support of multimedia applications which combine real-time traffic (voice, video) and bursty traffic (data, image). Real-time sessions require bandwidth and delay guarantee, and are generally carried on virtual circuits which are established at call setup time. Bursty traffic, on the other hand, is carried in a connectionless, (i.e. datagram) mode. For connection oriented traffic (voice, video), admission control must be exercised before the call is accepted in the network. After call establishment, the virtual circuit (VC) connection must guarantee bandwidth and quality of service (QoS). Datagram traffic is not subject to call acceptance control as virtual circuit traffic is. Thus, it may create congestion in the network. To avoid congestion, link-by-link and end-to-end flow and congestion control schemes must be introduced. For efficient and reliable datagram transport, explicit acknowledgments are used. Note that the passive acknowledgment scheme (i.e., the transmission on the next hop acts as an ACK on the previous hop) first introduced in the ARPA packet radio network [3] will not be suitable here since CDMA is used, and therefore the forwarding node might not use the same code as the originating node.

In this paper, we present a multi-cluster, multi-hop packet radio network architecture that addresses the above challenges and implements all required features. In a mobile, multi-channel (code), and multi-hop environment, the topology is dynamically reconfigured to handle mobility. Routing and bandwidth assignment are designed so as to meet the various types of traffic requirements. Node clustering, VC setup and channel access control are the underlying features which support this architecture, and will thus be the main focus of this paper.

The rest of the paper is organized as follows: in section 2, the multi-cluster network architecture is presented. Channel access scheme and VC establishment are presented in section 3. Performance results are shown in section 4. Section 5 will discuss the extensions of the basic algorithm. Finally, conclusions are given in section 6.

* This work was supported by the U.S. Department of Justice/Federal Bureau of Investigation, ARPA/CSTO under Contract J-FBI-93-112 Computer Aided Design of High Performance Wireless Networked Systems.

2. Multi-cluster network architecture and clustering algorithm

Several network alternatives can be considered for WAMIS. In our case, because of the requirements of efficient network resource control, multimedia traffic support and suitability to CDMA, we have selected a distributed cluster approach [4]. In fact, clustering provides a convenient framework for the development of important features such as code separation (among clusters), channel access, routing, power control, virtual circuit support and bandwidth allocation.

Following the clustering approach, the entire population of nodes is grouped into clusters. A cluster is a subset of nodes which can (two-way) communicate with a clusterhead and (possibly) with each other. In Fig. 1, nodes A, B and C are clusterheads for their coverage area respectively. Each of them serves as a regional broadcast node, and as a local coordinator to enhance channel throughput. Within a cluster, we can easily enforce time-division scheduling. Across clusters, we can facilitate spatial reuse of time slots and codes.

The objective of the clustering algorithm is to find a feasible interconnected set of clusters covering the entire node population. A good clustering algorithm should be stable to radio motion, i.e. it should not change the cluster configuration too drastically when a few nodes are moving and the topology is slowly changing. Otherwise, the clusterheads will not control their clusters efficiently and thus lose their role as local coordinators.

To this end, two distributed clustering algorithms are considered. One is the lowest-ID algorithm [5]: the lowest-ID node in a neighborhood is elected as the clusterhead. The other is the highest-connectivity (degree) algorithm, which is a modified version of [6]. In this case, the highest degree node in a neighborhood becomes the clusterhead. The algorithms are described below:

I. LOWEST-ID CLUSTER ALGORITHM:

Each node is assigned a distinct ID. Periodically, the node broadcasts the list of nodes that it can hear (including itself).

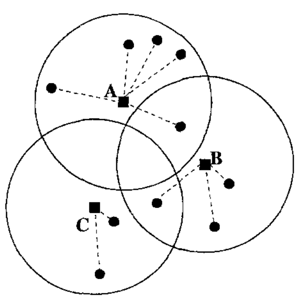


Fig. 1. Example of clustering.

- A node which only hears nodes with ID higher than itself is a *"clusterhead"* (CH).
- The lowest-ID node that a node hears is its clusterhead, unless the lowest-ID specifically gives up its role as a clusterhead (deferring to a yet lower ID node).
- A node which can hear two or more clusterheads is a *"gateway"*.
- Otherwise, a node is an ordinary node.

Fig. 2 shows a 10 node example, where nodes 1, 2 and 4 are clusterheads; nodes 8, 9 are gateway nodes.

II. HIGHEST-CONNECTIVITY CLUSTER ALGORITHM:

Each node broadcasts the list of nodes that it can hear (including itself).

- A node is *elected* as a clusterhead if it is the most highly connected node of all its *"uncovered"* neighbor nodes (in case of a tie, lowest ID prevails).
- A node which has not *elected* its clusterhead yet is an *"uncovered"* node, otherwise it is a *"covered"* node.
- A node which has already elected another node as its clusterhead gives up its role as a clusterhead.

Fig. 3 shows the same 10 node example, where nodes 5,7 and 8 are clusterheads; nodes 2, 3, 9, 10 are gateway nodes.

III. PROPERTIES OF THE TWO CLUSTER ALGORITHMS:

(a) No cluster heads are directly linked.

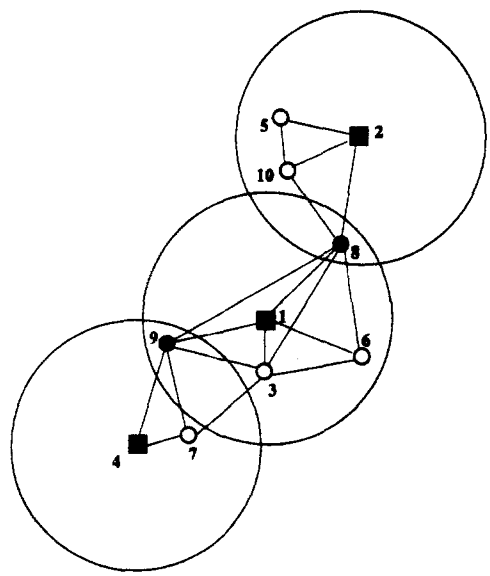


Fig. 2. Example of cluster formation (lowest-ID).

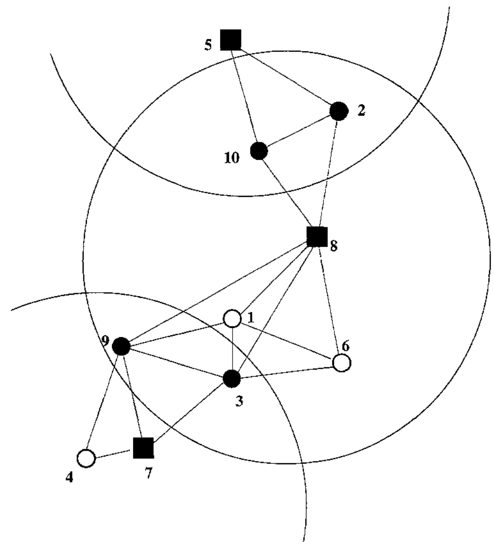


Fig. 3. Example of cluster formation (highest-connectivity).

(b) In a cluster, any two nodes are at most two-hops away, since the clusterhead is directly linked to every other node in the cluster.

Following these properties of clustering algorithms, each node is either a clusterhead itself or is directly linked to one or more clusterheads. Note that property (a) allows only one clusterhead per cluster. The clustering algorithm must be performed as rapidly as possible, so that each clusterhead can take and maintain control of its members efficiently.

We have simulated these two clustering formation algorithms to see which one is more suitable for our networking purpose. When the nodes are moving, and the connectivity changes, we observed that nodes re-elect their clusterheads. Two measures were monitored during the movement of nodes, namely: (a) the number of nodes which change their roles as clusterheads, and; (b) the number of nodes which switch clusters. At the beginning of simulation, we randomly (uniformly) generate N nodes inside a 100×100 square as shown in Fig. 4. We assume two nodes can hear each other if their distance is within the transmission range. In our simulation, we ran two types of experiments corresponding to different types of movements: (a) random movements: each node moves randomly by one unit in each direction (or stays where it is) between time ticks; (b) predefined trajectories: each node moves along straight trajectories, and bounces against the walls of the 100×100 square. The speeds are the same (1 unit/time tick) in both experiments. In our experiments, we monitor cluster changes per time tick. The results are reported in Figs. 5(a) and (b), and are almost identical for the two types of movements. We notice that the lowest-ID clustering algorithm yields the fewest changes in either measure (Fig. 5). That means, it provides a more stable cluster formation than the highest-connectivity one. This is because in the

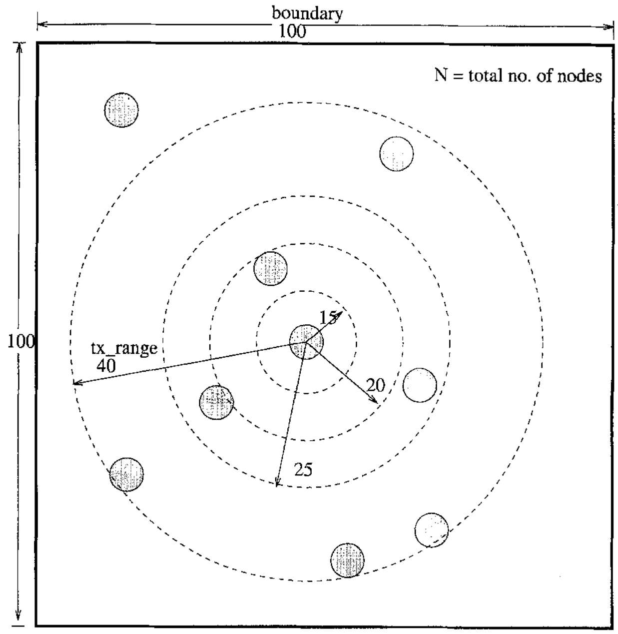


Fig. 4. Simulation boundary and tx_range.

latter scheme when the highest-connectivity node drops even one link due to node movement, it may fail to be re-elected as a clusterhead. While the lowest-ID node can still be a clusterhead. For these reasons, we will base our algorithm on the lowest-ID clustering formation.

We note that, as radio connectivity changes, different radios are chosen as clusterheads and gateways. Thus, each radio must be capable of taking on both of these roles. Performing these roles, however, does not require additional resources (e.g. buffers, processing power etc) since protocol support functions are well distributed among all nodes in the cluster, as discussed in section 3. For example, routing within a cluster is fully distributed. The route does not need to go through a clusterhead. For example, in Fig. 2, a packet can be directly routed from node 3 to node 6 without going through 1. An inter-cluster route must, however, go through gateways (in Fig. 2, the path from 7 to 10 is {7,9,8,10}), since clusters use different codes and only gateways can listen to both codes alternatively as discussed in section 3.

Requiring routes to go through gateways eliminates some routing options. In Fig. 2, for example, the path from 7 to 3 is {7,9,3}. These restrictions can be removed by introducing the notion of "distributed gateway" (DG). Namely, a distributed gateway is a pair of nodes, neither of which is a gateway, within hearing range and residing in different clusters. For example, the pair (7,3) is a DG. One can easily verify that, by allowing DGs along intercluster routes, unrestricted routing is achieved. An important benefit of DGs is that of maintaining connectivity in pathological situations where the basic clustering algorithm would lead to disconnections. Consider the example in Fig. 6. Lowest ID clustering as well as highest-connectivity clustering lead to two clus-

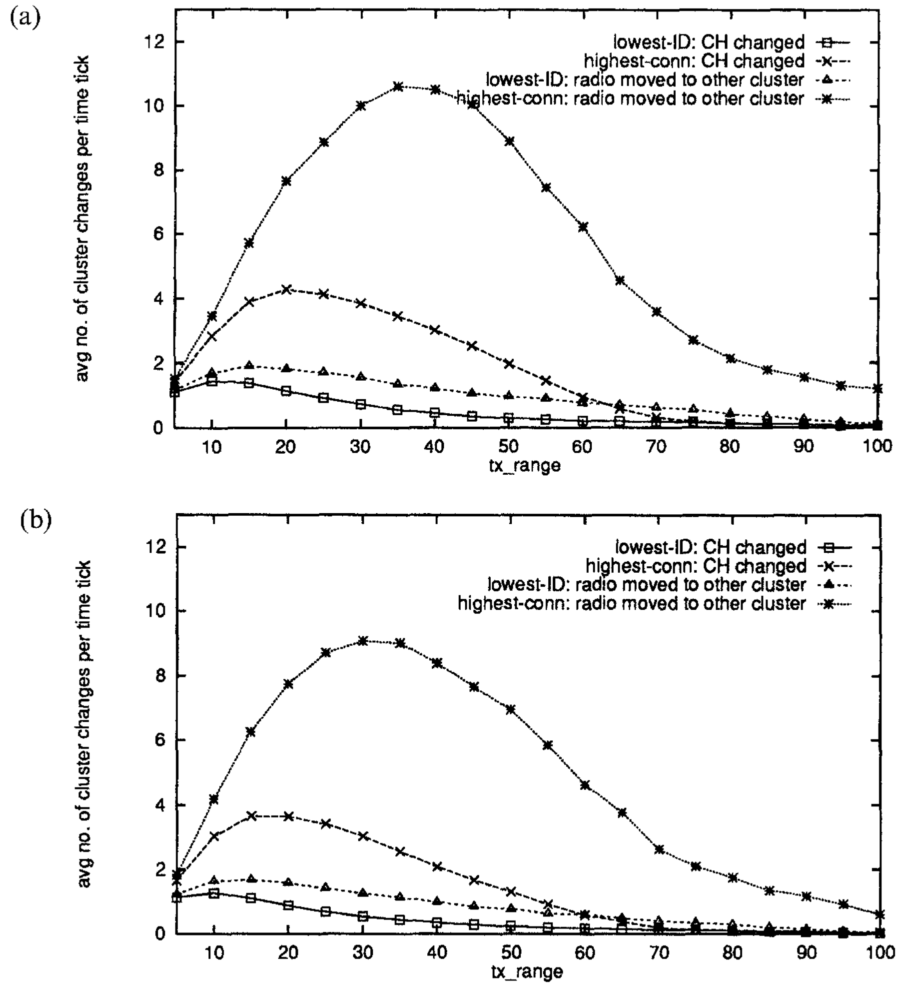


Fig. 5. Comparisons of clustering ($N = 30$). (a) Random movements; (b) predefined trajectories.

ters (with node 1 and 2 as CHs, respectively) which are disconnected, since neither 3 nor 4 qualify as gateways. With the DG option, the pair (3,4) becomes a distributed gateway, and can thus support routes across the two clusters. The implementation of distributed gateways introduces some extra overhead, since two radios must periodically rendezvous on a common code to communicate with each other. Thus, it may be advisable to enable this option only when the two clusters are otherwise disconnected.

We have run several experiments to evaluate the effectiveness of the DG technique. The first set of experiments was based on 20 nodes, with transmission range $R = 50$. For this case, we generated 10,000 sample node distributions in the 100×100 square. On average, 2 DG pairs were found in each sample. Connectivity (defined as fraction of connected node pairs) without exploiting the DGs was 99.4%. With the DGs, connectivity increased to 1. In a second set of experiments, the range was reduced to $R = 40$. In this case, connectivity without/with DGs was 90% and 98.3% respectively. Again, 2 DG pairs were found on average in each sample. These results show that when connectivity is weak (as in the $R = 40$ case), the DG strategy can be extremely effective.

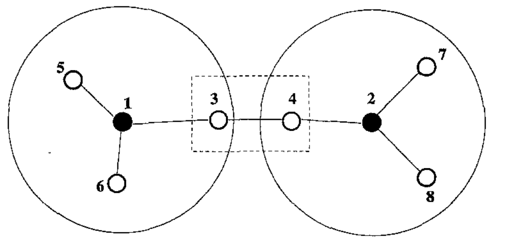


Fig. 6. Example for distributed gateways.

3. Channel access scheme

Following the clustering algorithm, the entire node population is organized and coordinated by clusters and clusterheads. Now the problem is how to run the algorithm in real-time and how to take advantage of clusterheads to perform networking functions efficiently. More specifically, how do we select the appropriate channel access method within each cluster to enhance the throughput and utilization of the limited channel resources.

In view of the real time traffic component (which requires dedicated bandwidth), we have chosen Time-Division Multiple Access (TDMA) within a cluster. Code Division Multiple Access (CDMA) can also be overlayed on top of the TDMA infrastructure; namely, multiple sessions can share the same TDM slot via CDMA. In this case, the near-far problem and related power control algorithm become critical to the efficiency of the channel access [7]. In addition, separate codes are assigned to different clusters in order to reduce the effect of intercluster interference. Specifically, we allow only one transmission (per code) within each cluster, and use different sets of codes in neighboring clusters.The cluster solution with code separation reduces power interference and maintains spatial frequency reuse. As the network grows large, the clusters may outnumber the available codes. To overcome this problem, codes can be reused in non-adjacent clusters. Assuming a 4-cluster code reuse, and allocating two separate codes to each cluster (for overlayed CDMA/TDMA), plus one control code (which is common to all clusters), the total number of orthogonal codes required by this scheme is 9.

Within the framework of the basic architecture, we can implement various protocols and functions. The following will describe the implementation of the clustering algorithm and of the channel access scheme.

3.1. Synchronous time division frame

The transmission time scale is organized in frames, each containing a fixed number of time slots. The entire network is synchronized on a frame and slot basis. The frame/slot synchronization mechanism is not described here, but can be implemented with techniques similar to those employed in the wired networks, e.g. "follow the slowest clock" [8], properly modified to operate in a wireless mobile environment. Propagation delays will cause imprecisions in slot synchronization. However, slot guard times (fractions of millisecond) will amply absorb propagation delay effects (in the order of microseconds). A frame is divided into two phases, namely, control phase and info phase as shown in Fig. 7. The following subsections will explain how these two phases are used to perform the desired functions.

3.2. Control phase

The control phase is used to perform all the control functions, such as slot and frame synchronization, clustering, routing, power measurement, code assignment, VC set-up, etc. By exchanging the connectivity information among the neighbors, each node selects its clusterhead and updates its routing tables. Clusterheads assign the slot(s) and code to each VC request within their covering area. The number of slots per frame assigned to a VC is determined by the bandwidth required by the VC.

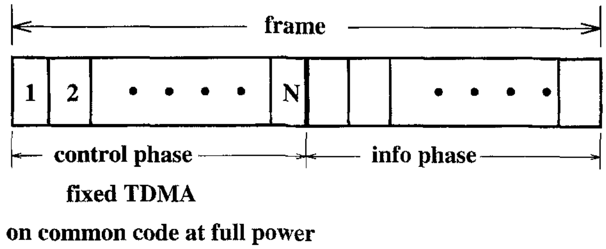


Fig. 7. Channel access frame.

Another function of the control phase is the exchange of power gain lists among neighbors. Power gain is actually defined as the power propagation loss from transmitter to receiver. As described below, the clusterhead (CH) can gather all power gain lists from its members, and maintain the power gain matrix. This matrix is useful to control power adjustment and code division inside the cluster.

As depicted in Fig. 7, the control phase uses fixed TDMA with full power transmission on a common code. That is, each node takes turns to broadcast its information to all of its neighbors in a predefined slot, such that the network control functions can be performed distributedly. The following functions are performed by each node during the control phase.

Each station broadcasts:

- its up to date routing list defining the next node on the route to each destination and the number of hops (Bellman-Ford Routing Alg.),
- ID of its CH,
- its power gain list with respect to its neighbors (derived from received power measurements),
- slot reservation status of its information subframe, indicating also which code it will listen to in the free slots,
- ACKs for last information subframe.

Each station performs the following routing, and power control functions:

- updating the routing table according to the routing lists that it has received from the neighbors (Bellman-Ford Routing Alg.),
- measuring power gains for all neighbors; updating power gain matrix according to power gain lists received from all its neighbors,
- measuring SIR (Signal to Interference Ratio) for the incoming links,
- selecting its clusterhead,
- recording the slot reservation status of its neighbors,
- obtaining ACK and reserving the slot.

At the end of the control phase, each node has learned (from the information broadcast by the clusterhead) the channel reservation status of the information phase, and the power measurement matrix to all its neighbors. This information will help it schedule free slots, verify failure of reserved slots, and drop expired real time packets.

Since in a mobile network clusters are dynamically reconfigured (and therefore CHs change), each node must be able to take on the role of clusterhead. The proposed scheme allows, however, the exclusion from CH duty of predefined sets of nodes, for example nodes which are moving too fast (airplanes), or which lack the required resources (sensors).

From Fig. 7, we note that the size of the control frame (which uses a common code) grows linearly with the number of radios in the network. If the network grows to several hundred radios, the control O/H becomes prohibitive. The overhead can be dramatically reduced by reusing the same control slot for radios in non-adjacent clusters (in the same way as codes were reused). Assuming 4-cluster slot reuse and an average of six nodes per cluster, a 24 slot control frame is required. This number can be further reduced by assigning to each cluster, during normal operation, only two control slots: one for the CH, and another to be used by the other radios in the cluster in a round robin fashion. Full channel access functionality is maintained, but performance is degraded because of slower route updating, VC reservations, etc. Regarding ACKs, one should note that the CH can ACK all successful data transmission in the cluster, except for its own (recall that voice/video packets are not ACKed). The radio transmitting in the "other" control slot will ACK the CH transmissions.

Finally, one should note that the proposed control structure can be generalized to support any distributed routing scheme (either link state or distance vector type). In particular, loop free, distance vector schemes can be employed (although, they have not yet been implemented in our simulator) [11]. Flooding can, of course, also be used.

3.3. Info (voice/video/data) phase

The info phase must support both virtual circuit and datagram traffic. Since real time traffic (which is carried on a VC) needs guaranteed bandwidth during its active period, bandwidth must be preallocated to the VC in the info phase before actual data transmission, as depicted in Fig. 8. That is, some slots in the info subframe are reserved for VCs at call set-up time. The remaining slots (free slots) of each cluster can be accessed by datagram traffic using an S-ALOHA scheme. Furthermore, CSMA probing of VC slots by datagram stations can be considered for reuse of silent VC slots. Two or more VCs can share the same time slot using CDMA. In this case, transmit power must be adjusted in order to yield acceptable signal to interference ratio (SIR). A power control algorithm is carried out which is based on the computation, exchanging and updating of SIR values at the nodes involved. These ratios, in turn, can be computed from the power gain matrix obtained from the control phase, or can be obtained via a separate distributed procedure [7,9].

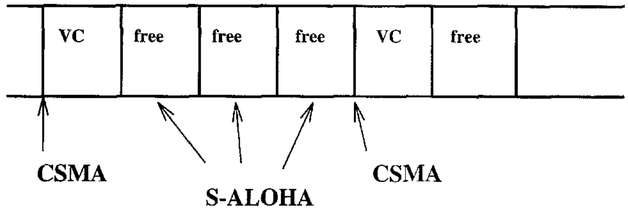


Fig. 8. Channel access of info phase subframe.

3.4. Example of VC set-up

Let us consider the topology in Fig. 2, and give an example to illustrate how a VC is set up. Suppose that node 7 wants to establish a connection to node 5. First, node 7 broadcasts a VC request through the control phase. The clusterhead, node 4, is responsible for servicing this request. Since there is no direct link between node 7 and 5, the clusterhead selects the first leg of the path through node 9 (by inspection of its routing table), and assign slot(s) and code to this link.

Next, after getting the slot assignment from clusterhead 4, node 9 inspects its routing table and discovers that it has to make a VC request again to get to next node 8. It thus places a request to clusterhead 1, which is common to both 9 and 8. Clusterhead 1 will assign a slot to link 9-8. Node 8 forwards the request and so on, until the path is completely traced to destination node 5 as depicted in Fig. 9. The connection may be characterized by a given QoS, i.e., a certain bandwidth requirement, which is translated into number of slots per frame, or a

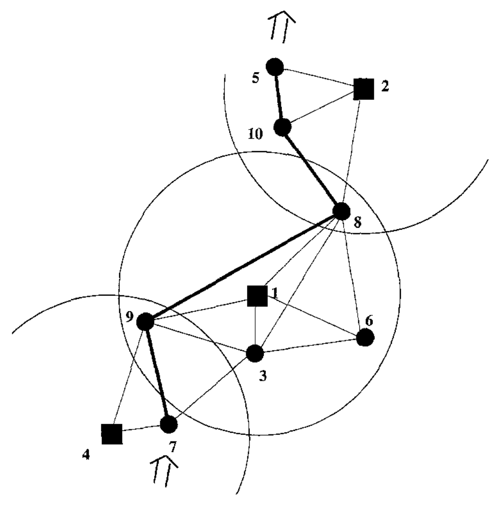


Fig. 9. Example of VC set-up.

given channel quality, related to SIR value or error rate on a link. These QoS requirements are checked and enforced during the VC set-up phase.

3.5. Fast Reservation VC

The VC set-up scheme described in the previous section is suitable for a network with slow mobility. In a highly mobile environment it may happen that the time required to set up a new VC is comparable to the interval between path changes. Thus, the conventional VC reconfiguration scheme cannot catch up with station movements. For highly mobile environments we propose an alternate scheme based on *fast reservations*. We already showed that the clustering algorithm can adequately adjust to station movements. We further assume that the dynamic routing algorithm can efficiently track the changes in topology. In our experiments we have used the basic Bellman-Ford algorithm which leads to occasional loops during transitions. In the future, we plan to implement a more robust, loop free routing algorithm such as described in [11].

In the Fast Reservation algorithm each packet in the VC stream is routed individually, based on the destination address, very much like a datapacket. As a difference, however, the first packet in the VC stream, upon successfully capturing a slot in the info subframe, will *reserve* it for all subsequent frames. If the slot remains unused for a certain number of frames, it is declared free by the cluster controller and it is returned to the free slot pool. This scheme, which was inspired by the PRMA protocol [12], allows the VC stream to dynamically select a new path to destination when the old path fails. Of course, each path change will cause some disruption (i.e. possible out of sequencing; delay to acquire a slot on the new path; possible looping if the routing algorithm is not loop free, etc). However, the disruption is much less severe than if a new path had to be set up from source to destination using the conventional scheme. In the following section, experimental results show that the Fast Reservation scheme can efficiently handle a high degree of mobility.

Several refinements of the basic Fast Reservation scheme are currently explored. One interesting feature, made possible by CSMA probing, is the assignment of priority to VC packets in capturing free slots. Namely, the VC packet is transmitted immediately, while the datagram packet must "probe" the slot before transmission. Probing also allows the reuse of unused VC slots by datagram traffic, thus alleviating the overhead caused by obsolete reservations left over after a path change. Another option is the dropping of "least significant" packets within a hierarchically encoded video or voice stream following a rerouting. The dropping of least significant bits has been applied in the past in several communication systems either to increase channel capacity (at the expense of signal equality) or to avoid congestion [13]. In our case, the dropping of low priority packets during the rerouting phase will alleviate congestion and reduce reservation delay, at the expense of temporary signal quality degradation. Quality is restored automatically after rerouting, to the degree that there is sufficient bandwidth on the new path.

Having defined both a static and a "fast" VC set-up procedure, we are now faced with the dilemma of deciding which should be used in what case. One way to avoid this problem is to combine the two schemes into a single, fast reservation based scheme. Essentially, the call request packet of the conventional scheme now becomes a fast reservation packet. This packet, however, has lower priority than the packet of an already established connection, so that the new call is blocked first if there is insufficient bandwidth in the network to accommodate it.

4. Performance evaluation

4.1. Link throughput

Link throughput is defined as the sum of the throughputs on the links which are simultaneously active in the multicluster network. Since channel rate is assumed uniform over the entire network, link throughput is proportional to the number of simultaneously active links. Link throughput is a simple figure of merit. It allows to derive preliminary tradeoffs between system parameters. In order to compute link throughput, recall that power interference across clusters is reduced by code separation. Since power drops very rapidly with distance, in each time slot at least one link per cluster can be active with tolerable intercluster interference. Thus, the average number of simultaneous active links is approximately equal to the average number of clusters. As long as a cluster has one or more links (i.e., it is not an isolated, single node cluster), the cluster has link throughput equal to 1. Here, we make the optimistic assumption that there *always* is *one* successful transmission per cluster per slot. This assumption is acceptable when comparing different strategies. However, it does not take into account data slot collisions, silence gaps, control O/H etc. For a more accurate evaluation, simulation will be used.

Following the same idea as in Fig. 4, we generate different topologies by placing N nodes in the square with variable transmission power range. More precisely, we generate about 100 random placements of the N nodes in the 100×100 square. We then step throughput various values of the transmission range and for each value we obtain different topologies corresponding to the different (random) placements. We average our performance measures over the random placements. The average aggregate link throughput versus transmission range is shown in Fig. 10. Also, the average number of clusters

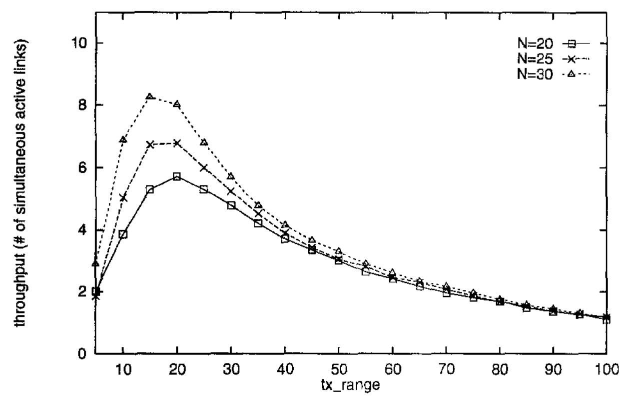


Fig. 10. Link throughput.

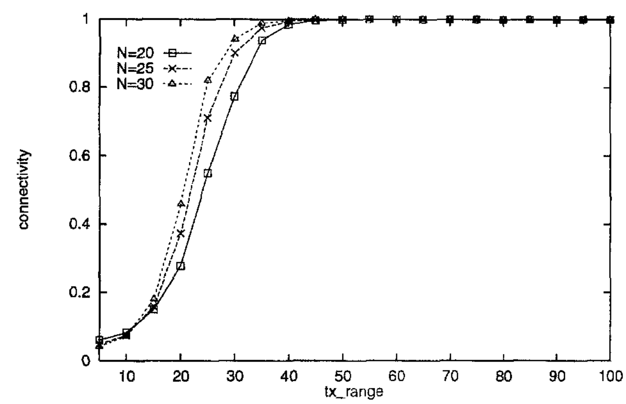


Fig. 12. Connectivity.

and connectivity are monitored while generating those topology (Fig. 11, Fig. 12). The *"connectivity"* here is defined as the fraction of node pairs which can communicate (through single or multiple hops).

The results confirm our intuition that there exists a tradeoff between transmission range and throughput. For relatively smaller transmission (tx) range, the graph consists of several isolated subgraphs, with good spatial reuse but poor connectivity. Too small a tx range, however, leads to a decrease in throughput since most of the clusters contain only one node, (i.e. the CH), and no links. As the tx range grows, we have better connectivity but less efficient spatial reuse, and thus lower throughput. Adaptive power adjustment can be used to optimize the tradeoff between connectivity and spatial reuse. Techniques similar to those reported in hierarchical routing optimization [14,15] can be employed for finding optimal cluster size, although in this case novel criteria (e.g. spatial reuse) and constraints (e.g. distributed implementation) must be accounted for.

In the previous experiment, we only have considered TDMA. If we adopt the power adjustment algorithm [9,10], then we can combine TDMA with CDMA. As discussed in section 3, CH maintains the power gain matrix for pairwise members. If a call request comes in, and there are no free slots, then the CH will search for a feasible reserved VC slot so that two VCs can share the slot in CDMA mode. "Feasible" here means that their pairwise power gains satisfy a set of conditions such that they can adjust their tx powers and coexist with acceptable signal quality.The CDMA sharing scheme leads to approximately a 20% throughput improvement, as shown in Fig. 13.

4.2. End-to-end throughput

In the previous experiments the throughput was defined as the sum of single link throughputs. This performance measure, albeit simple to compute, is however biased towards small transmission range, i.e. a large number of small clusters, many of which may actually be disconnected. In practice, we are more interested in end to end (rather than single link) throughput, and would like to maintain connectivity among as many node pairs as possible in the system. In order to reflect these requirements, we introduce a new throughput measure, which we call the weighted end-to-end throughput, defined as follows:

$$Th = \sum_{i=1}^{CS} f_i \frac{C_i}{L_i},$$

where:

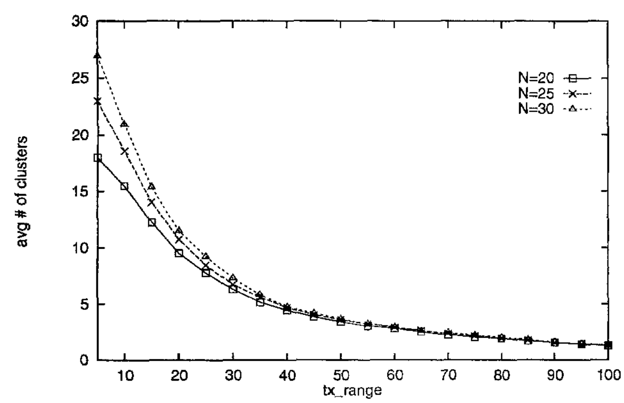


Fig. 11. Average number of clusters.

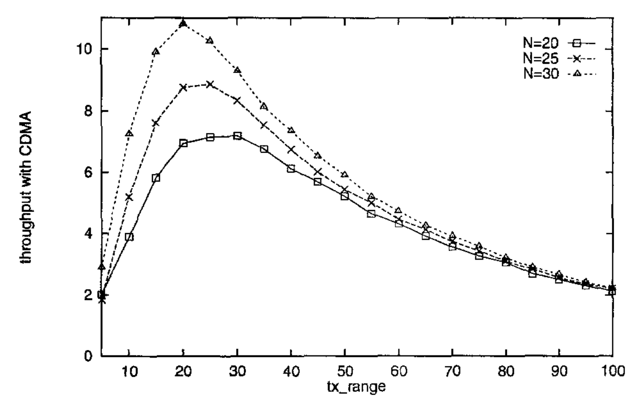


Fig. 13. Link througout with power controlled CDMA in a cluster.

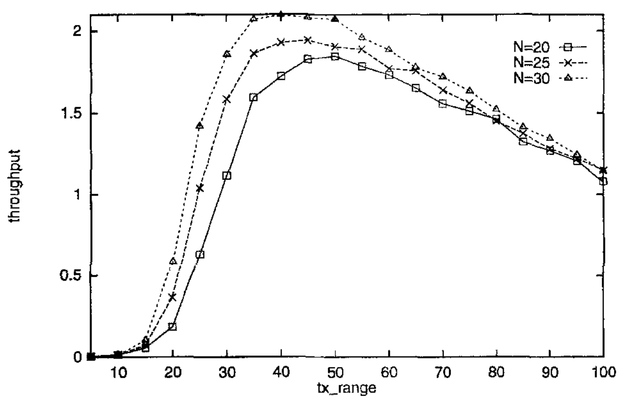


Fig. 14. Weighted end to end throughput.

CS is the number of connected subgraphs in the topology,

C_i is the link throughput of connected subgraph i, $1 \leq i \leq CS$,

L_i is the average hop length in C_i,

f_i is the fraction of the node pairs interconnected by C_i.

For example, if the topology is connected, $CS = 1$ and $f_1 = 1$. We expect that this new throughput measure will lead to larger values of optimal transmission range. In fact, an increase in transmission range will decrease average hop length and increase the fractions of connected node pairs. The results, reported in Fig. 14 and Fig. 15, show that the optimal tx range is between 40 and 50, both for pure TDMA and combined TDMA and CDMA. From Fig. 12 we also note that for that range of values the graph is connected with very high probability. The weighted throughput is lower than the link throughput (as expected) by a factor of about 4. It is also remarkable to note that, under this new measure, the combined TDMA and CDMA scheme yields an 80% improvement over the simple TDMA scheme, as compared to 20% improvement observed with the link throughput measure. In part, this is because the higher transmit range operation creates larger clusters (see Fig. 11) and offers more CDMA sharing opportunities (we need at least 4 nodes to perform CDMA sharing).

4.3. Conventional VC performance with mobile radios

For VC connections, which are supported essentially in a circuit switched mode, it is important to evaluate not only the throughput, but also the ability to withstand radio movements and dynamic cluster reconfigurations. In this section we consider conventional (as opposed to "fast") VC set-up. Typically, a new connection will be established if breakage occurs. Some voice loss, however, will be experienced during the transition (which may last seconds). It is of interest, therefore, to evaluate the probability of breakage as a function of radio mobility and call duration.

To this end, the experiment, reported in Fig. 16, monitors the probability that a VC connection is broken because of radio mobility. In this experiment, we randomly distribute N nodes inside the square, and generate calls between any node pairwith uniform exponential interarrival time (mean $1/\lambda$). The call duration is exponentially distributed with mean $1/\mu$.We repeat generating N nodes 100 times, and calculate the average call breakage probability. We assume that a radio link can carry up to 4 VC connections (e.g. digitized, compressed speech) in TDMA mode. No CDMA sharing was assumed in this case. We note that for the chosen parameter settings the network is connected with very high probability, therefore, the probability that no route exists between source and destination is negligible.We consider a range of radio speeds from 1 unit/minute to 1 unit/second. Assuming a maximum transmission range of 1000 ft (i.e. 100 units in our 100×100 square), then the unit in our example will correspond to 10 ft. Thus, the lower end may correspond to a slow patrol and the higher end to a slow moving vehicle. We also consider a range of call durations, from 20 seconds (emergency call) to 180 seconds (typical phone call duration). The results show that the broken call probability

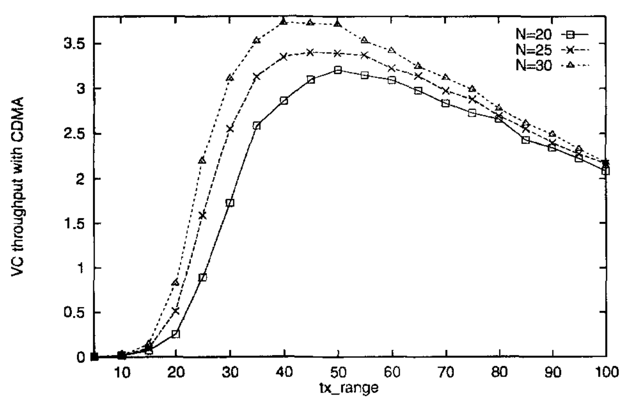


Fig. 15. Weighted end to end throughput with CDMA.

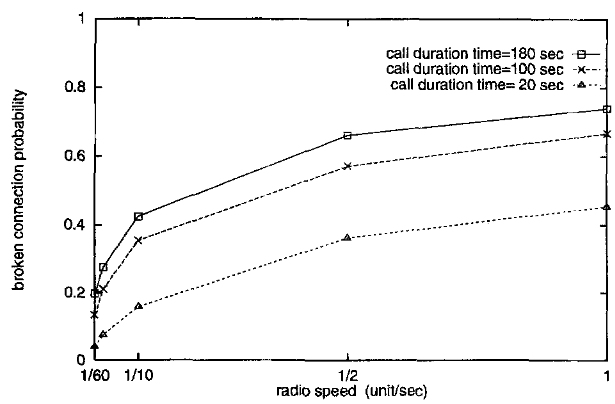


Fig. 16. Call broken probability ($N = 30$, tx range $= 40$, mean arrival rate $= 1/420$ sec).

Table 1
Max movement between route updates =5 units (average speed =54 km/hr).

tx range	35	40	50	60
pkts received	7357	8946	9900	9991
pkts dropped	2640	1051	98	7
out of seq	325	273	184	86
no. of loops	572	365	84	15
mean delay	3.18	2.80	2.04	1.70
avg hops	2.53	2.31	1.84	1.60
mean # of clusters	5.45	4.59	3.50	2.89

increases with radio speed and call duration, as expected. For very low speeds, the probability of call breakage because of mobility is tolerable (indeed, it is in the same order as the call blocking due to no available bandwidth or destination busy). For higher radio speeds, the performance of the conventional VC set-up is inadequate (the user would be forced to redial on a continuing basis). Performance is greatly improved by using the Fast Reservation VC scheme, as shown in the next section.

4.4. Fast Reservation VC performance

The Fast Reservation VC scheme was evaluated via simulation using the parallel simulation package MAISIE, developed at UCLA [16]. The simulated environment was the same as in previous experiments. Namely, 20 nodes are randomly placed in the 100×100 square. Each node is moved after each time interval by X units with probability P (where $P = 0.1$ in all of our experiments). The time interval corresponds to the interval between routing updates in the B-F routing algorithm, which is equal to the time frame period. A realistic reference value for the time frame period in our application is 100 ms.

In the first series of experiments, we assume that only one VC is present in the network, and monitor the performance on this VC as nodes move, for various values of node speed and transmission range. Tables 1 and 2 report the results. In Table 1 the max speed is 5 units/frame with probability 0.1 (i.e. average speed of 50 ft/sec or 54 km/hr). This corresponds to a relatively fast moving vehicle. A total of 10,000 packets were transmitted. Of these, some are dropped if the routing table does not show the next node for the intended destination (i.e. the destination has become temporarily unreachable). Some are delivered to destination out of sequence (in part because of dropping, and in part because of route changes or temporary loops). Number of loops (caused by routing table inconsistencies during transitions), average end to end delay (measured in number of time frame periods), average number of hops covered, and average number of clusters are also reported. We note that performance is quite sensitive to transmission range. A tx range =50 seems to be the minimum acceptable in order to support voice (with < 2% packet loss).

Table 2
tx range = 60 units.

Max movement	1	5	25
pkts received	9995	9991	9767
pkts dropped	3	7	232
out of seq	24	86	474
no. of loops	2	15	137
mean delay	1.91	1.70	1.73
avg hops	1.89	1.60	1.59
mean # of clusters	2.87	2.89	3.23

In Table 2, sensitivity to mobility is explored. We find that decreasing mobility from 5 to 1 units per frame reduces the number of dropped packets. An increase to 25 units/frame (equivalent to 270 km/hr) increases the number of out of sequence packets to 5%, thus rendering voice quality very poor. Clearly, we have reached a limit beyond which a more robust routing algorithm must be used.

5. Extensions to the basic algorithm

The multi-cluster, multi-hop network architecture framework shows great flexibility in supporting both VC and datagram traffic, and in handling dynamic reconfiguration. In addition, it allows many extensions to handle larger and more complex network environments. The following extensions will be the subject of future study:

1. *Cluster size optimization:* So far, we have assumed uniform and static tx power setting over all stations (except for power adjustment for CDMA sharing). We plan to consider in the future also non-uniform, dynamic tx power strategies. Dynamically adjusting the transmit power (rather than operating at full power) allows us to handle non-uniform mobile concentration. Namely, we can limit the number of radios in each cluster, and improve the efficiency of spatial reuse across clusters. Another way of looking at this problem is to find the "optimal" number of neighbors [17].
2. *ACK handling:* In the cluster architecture, clusterheads can act as (imperfect) observers, and can use the control subframe to ACK datagrams. We plan to evaluate the CH-ACK scheme as a complement to explicit hop by hop or end to end ACK schemes.
3. *Scaling to large node population:* The fixed TDMA control phase overhead becomes prohibitive for large node populations.There are several options to solve this problem. One possible option is to spread the control subframe over several info subframes.
4. *More accurate radio propagation models:* In our experiment,we have assumed that attenuation is uniform, and is solely a function of distance. We plan to develop more accurate models, which account for

multipath fading and shadowing, and for the presence of obstacles on the path.

5. *Mix of fast and slow moving stations:* If only a small fraction of the station is moving at fast speed (e.g. helicopters, trucks) while the majority is quasi static, then it is possible to modify clustering and routing algorithms so that the fast moving units once identified as such, are excluded from important roles such as clusterheads or store-and-forwarders (unless they are essential for connectivity). We plan to explore modification to our basic algorithms so as to effectively combine stations of various speeds.

6. Conclusion

A multi-cluster, multi-hop packet radio network architecture for wireless adaptive mobile information systems is presented. This network system is not constrained by a fixed infrastructure, rather, it can be deployed in an environment with no infrastructure at all. It can also handle limited radio mobility. Two clustering algorithms were investigated for suitability to a mobile networking environment. The ID-based algorithm shows good performance in dynamic clustering formation. Clusterheads enhance channel throughput by channel access coordination. Simulation experiments have identified key tradeoffs between transmit power and throughput performance, and have shown the advantages of CDMA sharing. Finally, the performance of the VC establishment scheme in the face of radio mobility was evaluated, showing the efficiency of the Fast Reservation VC scheme for highly mobile applications.

References

[1] K.S. Gilhousen, I.M. Jacobs et al., On the capacity of a cellular CDMA system, IEEE Trans. Veh. Tech. 40 (1991) 303–312.

[2] P. Jung, P.W. Baier and A. Steil, Advantages of CDMA and spread spectrum techniques over FDMA and TDMA in cellular mobile radio applications, IEEE Trans. Veh. Tech. 42 (1993) 357–364.

[3] J. Jubin and J. D. Tornow, The DARPA packet radio network protocols, Proc. IEEE 75 (1987) 21–32.

[4] D.J. Baker, A. Ephremides and J.A. Flynn, The design and simulation of a mobile radio network with distributed control, IEEE JSAC, SAC-2 (1984) 226–237.

[5] A. Ephremides, J.E. Wieselthier and D.J. Baker, A design concept for reliable mobile radio networks with frequency hopping signaling, Proc. IEEE 75 (1987) 56–73.

[6] A.K. Parekh, Selecting routers in ad-hoc wireless networks, *ITS* (1994).

[7] N. Bambos and G.J. Pottie, On power control in high capacity cellular radio networks, IEEE GLOBECOM 2 (1992) 863–867.

[8] Y. Ofek, Generating a fault tolerant global clock using high-speed control signals for the MetaNet architecture, IEEE Trans. Commun. 42 (1994) 2179–88.

[9] S.C. Chen, N. Bambos and G. Pottie, Admission control schemes for wireless communication networks with adjustable transmitter powers, IEEE INFOCOM 1 (1994) 21–28.

[10] N. Bambos, S. Chen and G. Pottie, Radio link admission algorithms for wireless networks with power control and active link quality protection, Technical Report UCLA-ENG-94-25, UCLA, School of Engineering and Applied Science (1994).

[11] M.S. Corson and A. Ephremides, A distributed routing algorithm for mobile wireless networks, Wireless Networks 1 (1995) 61–81.

[12] D.J. Goodman and S.X. Wei, Efficiency of packet reservation multiple access, IEEE Trans. Veh. Tech. 40 (1991) 170–176.

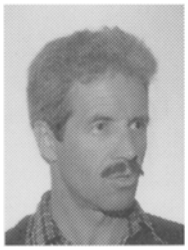
[13] R.W. Muise et al., Digital communication experiments in wide bandwidth packet technology, *1986 International Zurich Seminar on Data Communications*, ed. A. Kundig (1986) pp. 11–13.

[14] F. Kamoun, design considerations for large computer communications networks, Technical Report UCLA-ENG-7642, UCLA, School and Engineering and Applied Science (1976).

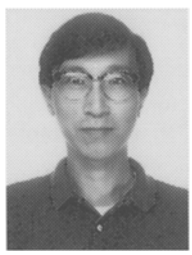
[15] Y. Rekhter, S. Hotz and D. Estrin, Constraints on forming clusters with link-state hop-by-hop routing, Technical Report USC-CS-93-536, University of Southern California (1993).

[16] R.L. Bagrodia and W.-T. Liao, Maisie: a language for the design of efficient discrete-event simulations, IEEE Trans. on Software Eng. 20 (1994) 225–38.

[17] L. Kleinrock and J. Silvester, Spatial reuse in multihop packet radio networks, Proc. IEEE 75 (1987) 156–167.



Mario Gerla received his graduate degree in electrical engineering from Politecnico di Milano, Italy, in 1966 and his M.S. and Ph.D. degrees in Computer science from University of California, Los Angeles, in 1970 and 1973, respectively. From 1973 to 1976, Dr. Gerla was a manager in Network Analysis Corporation, Glen Cove, NY, where he was involved in several computer network design projects for both government and industry, including performance analysis and topological updating of the ARPANET under a contract from DoD. From 1976 to 1977, he was with Tran Telecommunication, Los Angeles, CA, where he participated in the development of an integrated packet and circuit network. Since 1977, he has been on the Faculty of the Computer Science Department of UCLA. His research interests include the design, performance evaluation, and control of distributed computer communication systems and networks.
E-mail: gerla@cs.ucla.edu



Jack Tzu-Chieh Tsai was born in Tainan, Taiwan, R.O.C. He received his BS and MS degrees both in electrical engineering from the National Taiwan University in 1988 and from the University of Southern California in 1991, respectively. After that, he joined the University of California, at Los Angeles in 1991. He is currently a PhD candidate in Computer Science at UCLA. From 1989 to 1990, he was involved in a military joint project with the Industrial Technology Research Institute. It included design of channel access protocols, hierarchical routing for a radio network with tactical units. Since 1993, he has been working on the ARPA WAMIS project, including the network protocol design and implementation for a multimedia radio network. His research interests include network protocol design and implementation for wireless radio networks; performance evaluation and parallel simulation for distributed computer communication network systems.
E-mail: ttsai@cs.ucla.edu