Multimedia Transport in Multihop Dynamic Packet Radio Networks†

Chunhung Richard Lin and *Mario Gerla*
Computer Science Department
University of California, Los Angeles
Los Angeles, CA 90095-1432
{chunhung, gerla}@cs.ucla.edu

Abstract

The paper presents a distributed architecture which relies on code division access scheme for multimedia transport in a wireless mobile radio network without a fixed infrastructure. The proposed multicluster architecture has the capability of rapid deployment and dynamic reconfiguration. Without the need of base stations, this architecture can operate in areas without a wired backbone infrastructure.

The presented architecture has two main advantages. First, this architecture can provide spatial reuse of the bandwidth due to node clustering. Second, computer simulation reveals that our cluster structure is robust in the face of topological changes due to node motion, node failure, and/or new node addition. In all, this architecture provides a stable infrastructure for the integration of different types of traffic in a dynamic radio network.

1. INTRODUCTION

Current wireless systems, such as cellular systems, have fixed network configuration and fixed base stations or severs that are linked by a wired backbone infrastructure. In some cases, such as emergency disaster relief, when the backbone is not available, this type of architecture is infeasible. The goal of WAMIS (Wireless Adaptive Mobile Information Systems) is to overcome these constraints, and to develop techniques for the design of wireless networks that are adaptable to a variety of transmission environments, network configurations, and user services (including data, voice and image) [4, 7-9]. We propose a novel architecture which enables rapid deployment and dynamic reconfiguration of a network of wireless stations. More specifically, we propose a wireless mobile instant infrastructure concept for multimedia communications. If an infrastructure exists, the wireless network will also provide access to it through multiple hops.

The WAMIS infrastructure consists of a multihop packet radio network interconnecting a population of geographically distributed nodes. The presence of both mobility and multimedia requirements is the main challenge of the research. Since it is difficult to coordinate nodes in a dynamic packet radio network, contention-based channel access protocols, like ALOHA and CSMA, are easier to implement than controlled protocols, like TDMA [1, 5]. In fact, in a TDMA system some form of local or global control is needed to schedule transmissions so that they are nonoverlapped in time.

Another potential candidate is code division multiple access (CDMA) which provides flexibility and graceful degradation, and allows users to transmit several spread-spectrum waveforms simultaneously in the same bandwidth by using orthogonal (low cross correlation) spread-spectrum waveforms. CDMA features asynchronous operation, ability to add new users, multipath rejection, and antijamming properties [3, 6, 10].

The paper is organized as follows. Section 2 describes the assumptions and definitions used in the paper. Section 3 presents a distributed clustering algorithm and discusses its properties. Section 4 describes the channel access scheme. Section 5 analyzes the system performance. Section 6 concludes the paper.

2. ASSUMPTIONS AND DEFINITIONS

Each node contains an identical transceiver which can either transmit or receive at any given time. In addition, each node uses an omni-directional antenna for transmission. In the spread-spectrum code-division system, the receiver should be set to the same code as the designated transmitter. We assume that there is a particular set of spread-spectrum codes with low cross-code interference. Since the number of codes we can use is very limited, spatial reuse of codes will be important. Finally, all radio nodes use the same power for packet transmission.

There are two types of conflicts which can occur in the system: type I conflict occurs when two or more transmissions in the *same code* arrive at a receiver simultaneously; type II conflict occurs when a transmission arrives at a receiver which is tuned to a *different code*. Our study employs *no capture* model for type I collision. That means the receiver is unable to detect any packet when there are two or more simultaneous transmissions. All colliding packets must be retransmitted. Furthermore, the interference from different codes is not considered in type II conflict.

---

† This work was supported by the Advanced Research Projects Agency, ARPA/CSTO, under Contract J-FBI-93-112 "Computer Aided Design of High Performance Wireless Networked Systems".

0-8186-7216-1/95 $04.00

The following definitions as well as notations will be frequently used in the sequel.

Definition 1: (System Topology)

The *system topology* is a graph $G = (X, U)$, where X is the set of nodes, and U the set of logical edges. It is used to represent a packet radio network. There is only one transceiver in each node and the network operates in a half-duplex mode. A logical edge (x, y) means that node y is node x's one-hop neighbor under the current transmitting power, and vice versa. Fig. 1 shows a topology of a multihop packet radio network.

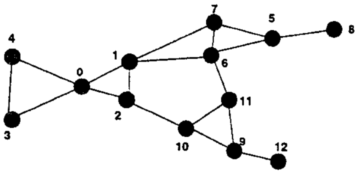


Fig. 1: The system topology

Definition 2: (Cluster)

Let $d(x, y)$ be the shortest hop distance between x and y. A *cluster* $C_i \subset X$ is a set of nodes, where for any two nodes $x, y \in C_i$, $d(x,y) \leq 2$. Namely, any two nodes in a cluster are at most two hops away. We define a *cluster coverage* $\{C_i\}$ of V, such that $X = \bigcup_i C_i$ and $C_i \cap C_j = \emptyset$, if $i \neq j$.

Definition 3: (Center and Radius of a Cluster)

The *center* of C_i is defined to be the node x_0 such that $\min_x \max_y d(x,y)$, $x, y \in C_i$. $\max_y d(x_0,y)$ is called the *radius* of a cluster.

Definition 4: (Degree of a Node)

The *degree of a node* x is the number of its one-hop neighbors. It is denoted by $d_G(x)$.

Definition 5: (Order of a Repeater)

For an edge $u = (x, y)$, x and y belong to different clusters. x and y is called *repeaters*. The number of clusters which a repeater can reach in one hop is called *the order of the repeater.* Note that the order of a repeater includes the cluster which it belongs to. Thus, the minimal order of a repeater is 2.

Definition 6: (Edge Partially in a Cluster $A \subset X$)

If one of the two end nodes of an edge u belongs to a cluster A, and the other does not belong to A, then u is said to be an *edge partially in* A, and we write $u \in \omega^+(A)$. Similarly, we define an *edge totally in* A, if both of the end nodes of u belong to A, and the set $\omega^-(A)$. Finally, the set of *edges incident to* A is denoted by $\omega(A) = \omega^+(A) \cup \omega^-(A)$.

Notations:

$\Gamma_1(x)$ $\equiv$ the set of all one-hop *neighbors* of x

$\Gamma_2(x)$ $\equiv$ the set of all two-hop *neighbors* of x

Fig. 2 shows a possible clustering of the topology in Fig. 1. In particular, $C_1 = \{0, 1, 2, 3, 4\}$, $C_2 = \{5, 6, 7, 8\}$, and $C_3 = \{9, 10, 11, 12\}$. The centers of C_1, C_2, and C_3 are node 0, 5, and 9 respectively. The radii of the three clusters are all 1. Node 1, 2, 6, 7, 10, 11 are repeaters. The orders of all repeaters are 2. $\omega^+(C_1) = \{(1,6)\ (1, 7)\ (2, 10)\}$, $\omega^+(C_2) = \{(1,6)\ (1, 7)\ (6, 11)\}$, $\omega^+(C_3) = \{(2, 10)\ (6, 11)\}$.

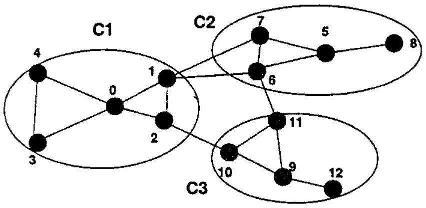


Fig. 2: Clustering nodes

3. THE MULTICLUSTER ARCHITECTURE

Most hierarchical clustering architectures for mobile radio networks are based on the concept of *clusterhead* [3, 4]. The clusterhead acts as a local coordinator of transmissions within the cluster. It differs from the base station concept in current cellular systems in that it does not have special hardware and in fact it is dynamically selected among the set of stations. However, it does extra work with respect to ordinary stations, and therefore it may become the bottleneck of the cluster. To overcome these difficulties, in our approach we abandon the clusterhead approach altogether and adopt a fully distributed algorithm [8, 9].

The objective of the proposed clustering algorithm is to find an interconnected set of clusters covering the entire node population. Namely, the system topology $G(X, U)$ is divided into small partitions (*clusters*) with independent control. A good clustering scheme will tend to preserve its structure when a few nodes are moving and the topology is slowly changing. Otherwise, there will be a lot of overheads to reconstruct clusters. Within a cluster, it should be easy to schedule packet transmissions and to allocate the bandwidth to real time traffic. Across clusters, the spatial reuse of codes can be facilitated. Since there is no notion of clusterhead, each node within a cluster is treated equally. This permits us to avoid vulnerable centers and hot spots of packet traffic flow.

3.1. The Centralized Clustering Algorithm

We are based on node connectivity to construct clusters in which any two nodes are two hops away at most. Our centralized clustering algorithm is shown below.

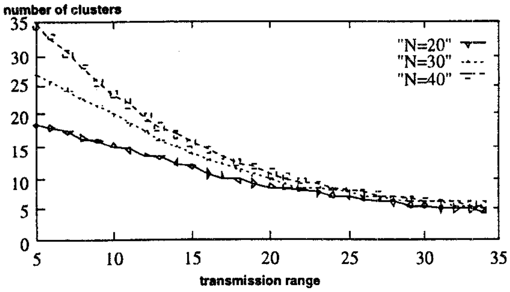


Fig. 3

Centralized Clustering Algorithm:

```
0.  $i = 0$.
1.  Set $x = \min\{y \mid d_G(y) = \max_{z \in X} d_G(z)\}$.
2.  $C_i = \{x\} \cup \Gamma_1(x)$;
    $X = X - C_i$;
    $U = U - \omega(C_i)$.
3.  If $X \neq \emptyset$, then $i = i + 1$ and goto 1.
```

After step 1, node x is the node with the largest degree. In case of a tie, select the node with the lowest ID. After step 2, node x and its one-hop neighbors construct a cluster. Remove the nodes and edges incident to this cluster from the graph, and then restart the algorithm, which continues until every node is assigned a cluster. Obviously, the algorithm construct clusters with *radius* at most 1. Fig. 2, for example, shows the result of applying this algorithm to the set of nodes in Fig. 1.

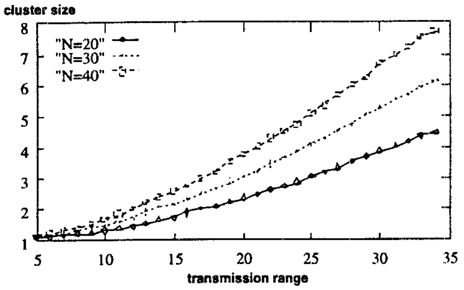


Fig. 4

We simulate this algorithm by placing N nodes randomly in a 100×100 area. We assume two nodes can hear each other if their distance is within the transmission range. From the simulation, we find the average size of clusters (i.e. the average number of nodes in a cluster) and the number of clusters. They are controlled by the transmitting power, which decides the transmission range. In Fig. 3 and Fig. 4, the larger the transmitting power, the more nodes a cluster contains, and the fewer clusters the system has. *Repeaters* relay packets from one cluster to another. Every repeater is time-sharing among a set of clusters, since only one transceiver is available and operates in a half-duplex mode. So the *order* of a repeater (see Definition 5) should be as small as possible for getting higher throughput (the minimal order of a repeater is 2). Fig. 5 shows the average order of repeaters versus the transmission range. From this figure, we can find the orders of most repeaters are 2 based on the clustering algorithm.

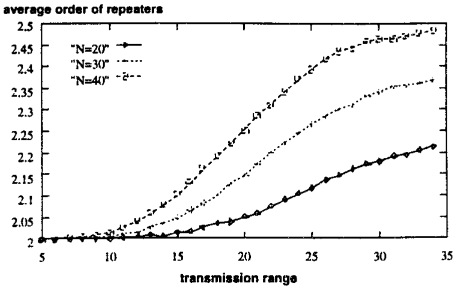


Fig. 5

Because the topology is dynamically changed in WAMIS, the reliability of packet routing is important to guarantee the integrity of network services. Thus, the existence of at least one path between a pair of nodes is required. The number of repeaters will affect the number of paths. In Fig. 6, we can find more than 50% of nodes are repeaters if the transmission range is over the interval (30, 35).

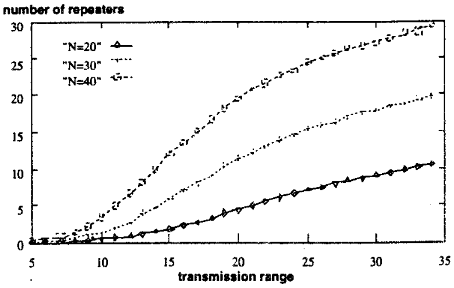


Fig. 6

In Fig. 7, we find that adjusting transmission range will increase the *connectivity* of the system. The connectivity is defined as the fraction of node pairs which can communicate through single or multiple hops. From the point of connectivity, the reasonable transmission range should be more than 35.

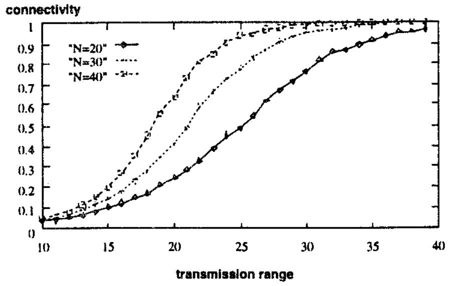
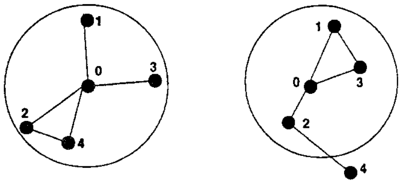


Fig. 7

WAMIS must provide continuous support of mobile nodes within the area of coverage [4, 9]. Our cluster structure provides a very robust infrastructure, which is not easily disrupted by mobility. We measure the stability of the multicluster architecture by assuming that nodes move and by counting how many nodes switch from one cluster to another (or construct a new cluster) within a time tick. For example, Fig. 8(a) is the original cluster. After one time tick, Fig. 8(b), the original cluster structure is destroyed, since $d(1, 4) = d(3, 4) = 3 > 2$. We assume every node knows the current cluster topology. In the situation of Fig. 8(b), the node which has the highest connectivity (in case of a tie, the lowest ID) (node 0 in this example), and its neighbors $\Gamma_1(0)$ (node 1, 2, 3) do not switch cluster. The other nodes (node 4) must find other clusters. Fig. 9 shows the stability of this cluster updating procedure. The mobility model we employ is that the direction of node motion is uniformly distributed over the interval $(0, 2\pi)$, and the distance over $(0, e)$ at each time tick. In Fig. 9, e is set to be 3 units. From Fig. 9, we note that the average number of nodes which switch clusters per time tick is relatively small. So, our cluster maintenance scheme based on node connectivity is stable in the mobile environment. It leads to better stability than the scheme reported in [4]. It also provides easier adaptation to topological changes.



(a) (b)

Fig. 8

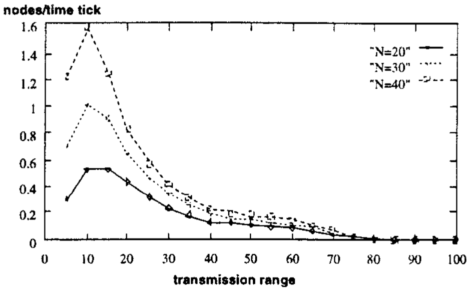


Fig. 9 ($e = 3$)

3.2. Distributed Clustering Algorithm

Before implementing this centralized algorithm in a distributed manner, we consider some operational assumptions underlying the construction of the algorithm in a radio network. These assumptions are common to most other radio data link protocols [2, 3].

A1: Every node has an unique ID and knows the IDs of its 1-hop neighbors. This can be provided by a physical layer for mutual location and identification of radio nodes.

A2: A message sent by a node is received correctly within a finite time by all its 1-hop neighbors.

A3: Network topology does not change during the algorithm execution (relaxed in the next section).

The algorithm is message driven and there are four types of messages used: NEIGHBOR, CENTER, JOIN, and UPDATE_DEGREE. Any node can initiate the algorithm by broadcasting NEIGHBOR, which contains the IDs of its 1-hop neighbors. The following symbols and functions are used in the algorithm:

- x: the ID of the current node which is running the algorithm;
- $\Gamma_{1,2}(x)$: the set of node x and its 1-hop and 2-hop neighbors;
- $d_G^e(x)$: the effective-degree of node x with initial value $d_G(x)$;
- *Cluster*(x): the set of nodes in x's cluster;
- *HighestPriorityNode*(A): the function which returns the node ID with the largest $d_G^e(x)$, $x \in A$; in case of a tie, return the node with the lowest ID.

Distributed Clustering Algorithm:

```
1.  On receiving NEIGHBOR($\Gamma_1(y)$),
    1.1 add elements to $\Gamma_{1,2}(x)$.
    1.2 if having not broadcast NEIGHBOR($\Gamma_1(x)$), broadcast it.
    1.3 if $\Gamma_{1,2}(x)$ is already completed and $x = HighestPriorityNode(\Gamma_{1,2}(x))$,
        1.3.1 broadcast CENTER(x).
        1.3.2 Cluster(x) = $\{x\} \cup \Gamma_1(x)$.
        1.3.3 $\Gamma_{1,2}(x) = \bigcup_{z \in \Gamma_1(x)} \Gamma_1(z)$.
        1.3.4 $d_G^e(x) = |\Gamma_1(x)|$.
        1.3.5 reset $\Gamma_1(x)$.
        1.3.6 exit.
2.  On receiving CENTER($y$),
    2.1 broadcast JOIN($x, y$).
    2.2 Cluster(x) = $\{y\} \cup \Gamma_1(y)$.
    2.3 $\Gamma_{1,2}(x) = \bigcup_{z \in \Gamma_1(x)} \Gamma_1(z)$.
    2.4 $d_G^e(x) = |Cluster(x) \cap \Gamma_1(x)|$.
    2.5 reset $\Gamma_1(x)$.
    2.6 exit.
3.  On receiving JOIN($y, z$),
    3.1 $\Gamma_{1,2}(x) = \Gamma_{1,2}(x) - \{y\}$.
    3.2 if $y \in \Gamma_1(x)$,
        3.2.1 let $A = \{v \mid v \in \Gamma_1(x) \text{ and } z \in \Gamma_1(v)\}$.
        3.2.2 $\Gamma_{1,2}(x) = \Gamma_{1,2}(x) - A$.
        3.2.3 $\Gamma_1(x) = \Gamma_1(x) - A$.
        3.2.4 $d_G^e(x) = d_G^e(x) - |A|$.
        3.2.5 broadcast JOIN($y, z$) and UPDATE_DEGREE($x, d_G^e(x), \Gamma_1(x)$).
    3.3 if $x = HighestPriorityNode(\Gamma_{1,2}(x))$,
```

```
        3.3.1 broadcast CENTER(x).
        3.3.2 Cluster(x) = $\{x\} \cup \Gamma_1(x)$.
        3.3.3 $\Gamma_{1,2}(x) = \bigcup_{z \in \Gamma_1(x)} \Gamma_1(z)$.
        3.3.4 $d_G^e(x) = |\Gamma_1(x)|$.
        3.3.5 reset $\Gamma_1(x)$.
        3.3.6 exit.
4.  On receiving UPDATE_DEGREE($y$, $d_G^e(y)$, $\Gamma_1(y)$),
    4.1 update the effective degree of node $y$ and $\Gamma_1(y)$;
    4.2 if $x$ = HighestPriorityNode($\Gamma_{1,2}(x)$),
        4.2.1 broadcast CENTER(x).
        4.2.2 Cluster(x) = $\{x\} \cup \Gamma_1(x)$.
        4.2.3 $\Gamma_{1,2}(x) = \bigcup_{z \in \Gamma_1(x)} \Gamma_1(z)$.
        4.2.4 $d_G^e(x) = |\Gamma_1(x)|$.
        4.2.5 reset $\Gamma_1(x)$.
        4.2.6 exit.
    4.3 if $y \in \Gamma_1(x)$, broadcast UPDATE_DEGREE($y$, $d_G^e(y)$, $\Gamma_1(y)$).
```

A node starts the algorithm either on receiving a WAKE message from an upper layer or by the first reception of some message sent by another node. In the first step, node x entering the algorithm exchanges the 1-hop neighbor ID list with its neighbors (step 1.2). Only after the list of its 1-hop and 2-hop neighbors has been completed, step 3 and 4 can then be executed. At this point, a node which is the highest priority node in its locality could become the *center* of a cluster (step 1.3). It sends CENTER message to its 1-hop neighbors. The center's 1-hop neighbors which have not been assigned a cluster join the center's cluster and broadcast JOIN messages (step 2). The JOIN message is forwarded to nodes which are two hops away from the node initiating the JOIN message. On receiving the JOIN message, the nodes which have not been assigned a cluster update their effective degrees and broadcast the update information (step 3.2). Similarly, the information of the new effective degrees are forwarded to nodes two hops away. It is notable that the information propagation is always limited to a 2-hop radius. At this point the algorithm proceeds among nodes that have not joined to a cluster. The algorithm terminates when every node has joined a cluster.

We illustrate the node clustering algorithm by applying it to the sample network given in Fig. 1. When node 6 is waked up, it starts exchanging NEIGHBOR message which contains the set {1, 5, 7, 11} with its one-hop neighbors. After the message is received by its neighbors 1, 5, 7, and 11, each of them thus adds {1, 5, 7, 11} to its local set $\Gamma_{1,2}$, and broadcasts the set of its neighbor's IDs.

The message sent by 11 is received by 6 and also wakes up 9 and 10 (2-hop away from 6) which send their neighbors {10, 11, 12} and {2, 9, 11} respectively. In a similar way, the message from 1 wakes up node 0, and 2 which send {1, 2, 3, 4} and {0, 1, 10} respectively. The message from 5 wakes up node 8 which sends {5}.

We shall concentrate on the execution of the algorithm at node 6. After receiving the messages from 1, 5, 7, and 11 at node 6, $\Gamma_{1,2}(6)$ contains the entries {0, 1, 2, 5, 6, 7, 8, 9, 10, 11}. At this point the set $\Gamma_{1,2}(6)$ is already completed. Now *HighestPriorityNode*($\Gamma_{1,2}(6)$) returns 0. Thus, node 6 must wait for 0 to determine its cluster first.

Node 0 receives the set of neighbors from 1, 2, 3, and 4. $\Gamma_{1,2}(0)$ contains the entries {0, 1, 2, 3, 4, 6, 7, 10}. Since $\Gamma_{1,2}(0)$ is ready and *HighestPriorityNode*($\Gamma_{1,2}(0)$) returns 0, it broadcasts CENTER(0) to $\Gamma_1(0)$. After receiving the message from 0, node 1, 2, 3, and 4 broadcast JOIN(1, 0), JOIN(2, 0), JOIN(3, 0), and JOIN(4, 0) respectively. At this time, a cluster {0, 1, 2, 3, 4} is constructed. On receiving JOIN(1, 0), node 6 updates its own effective degree $d_G^e(6) = 3$, $\Gamma_{1,2}(6) = \{5, 6, 7, 8, 9, 10, 11\}$ and $\Gamma_1(6) = \{5, 7, 11\}$, relays JOIN(1, 0) and broadcasts UPDATE_DEGREE(6, 3, Γ_1 (6)). Similarly, node 7 and 10 do the same operations, since they are adjacent to the new cluster. Successively, node 5 and 9 broadcast CENTER messages. Eventually, every node will determine its cluster. The final clusters obtained by the algorithm are: {0, 1, 2, 3, 4}, {5, 6, 7, 8} and {9, 10, 11, 12}.

3.3. Adapting to Topological Changes in a Dynamic Radio Network

WAMIS are a *dynamic radio network* in which: 1) nodes can change location; 2) nodes can be removed; and 3) nodes can be added. A topological change occurs when a node disconnects and connects from/to all or part of its neighbors. Thus, the cluster structure may be destroyed (i.e., there exist two nodes in a cluster more than two hops away, as shown in Fig. 8(b)). Our algorithm must adapt to topology changes and maintain the cluster structure. The extension of our algorithm is described below.

We assume every node can be aware of link breakages from the physical layer. Then it broadcasts DISCONNECT message to its remaining neighbors.

```
1.  If link (x, y) fails,
    1.1 $\Gamma_1(x) = \Gamma_1(x) - \{y\}$.
    1.2 $\Gamma_{1,2}(x) = \bigcup_{z \in \Gamma_1(x)} \Gamma_1(z)$.
    1.3 $d_G(x) = d_G(x) - 1$.
    1.4 if y $\in$ Cluster(x), $d_G^e(x) = d_G^e(x) - 1$.
    1.5 broadcast DISCONNECT(x, y) and UPDATE_DEGREE(x, $d_G^e(x)$, $\Gamma_1(x)$).
    1.6 if there exists $v$ in Cluster(x) and $v$ is not in $\Gamma_{1,2}(x)$,
        1.6.1 let $w$ = HighestPriorityNode(Cluster($x$)).
        1.6.2 if $w$ is not in $\Gamma_1(x)$, broadcast LEAVE(x).
2.  On receiving DISCONNECT(y, z),
    2.1 $\Gamma_1(y) = \Gamma_1(y) - \{z\}$.
    2.2 $\Gamma_{1,2}(x) = \bigcup_{v \in \Gamma_1(x)} \Gamma_1(v)$
    2.3 if there exists $v$ in Cluster(x) and $v$ is not in
```

```
    $\Gamma_{1,2}(x)$,
        2.3.1 let $w$ = HighestPriorityNode(Cluster($x$)).
        2.3.2 if $w$ is not in $\Gamma_1(x)$, broadcast LEAVE(x).
3.  On receiving UPDATE_DEGREE(y, $d_{\tilde{G}}(y)$, $\Gamma_1(y)$),
    3.1 if y $\in$ Cluster($x$),
        3.1.1 update the effective degree of $y$.
        3.1.2 if y $\in$ $\Gamma_1(x)$, broadcast UPDATE_DEGREE(y, $d_{\tilde{G}}(y)$, $\Gamma_1(y)$).
4.  On receiving LEAVE(y),
    4.1 if y $\in$ Cluster(x),
        4.1.1 Cluster(x) = Cluster(x) $-$ {y}.
        4.1.2 if y $\in$ $\Gamma_1(x)$ or x = HighestPriorityNode(Cluster($x$)), broadcast LEAVE(y).
```

Step 1.6 and 2.3 are executed when a cluster is destroyed. The example of Fig. 8(a) shows node 4 broadcasts DISCONNECT(4, 0) and UPDATE_DEGREE(4, 1, $\Gamma_1(4)$) (the new $d_{\tilde{G}}(4) = 1$). Node 4 finds 1 and 3 in Cluster(4), but not in $\Gamma_{1,2}(4)$ (step 1.6). So it finds the new *center* of its cluster (step 1.6.1) to be node 0 ($w = 0$). However, 0 is not in $\Gamma_1(4)$ (step 1.6.2). Node 4 has to give up its current cluster by sending LEAVE message and switches to another cluster.

In the distributed clustering algorithm of the previous section, the JOIN and CENTER messages are assumed containing the IDs of all members in the cluster. So the nodes adjacent to the cluster will know which nodes are in the cluster. When a node leaves a cluster, it has to determine which cluster it will switch to. Since it has the information of its adjacent clusters, it can choose one to join by sending a JOIN message. The message JOIN(x, y) means node x requests to join y's cluster.

```
5.  On receiving JOIN(y, z),
    5.1 if z $\in$ Cluster(x),
        5.1.1 Cluster(x) = Cluster(x) $\cup$ {y}.
        5.1.2 if z $\in$ $\Gamma_1(x)$,
            5.1.2.1 $d_{\tilde{G}}(x) = d_{\tilde{G}}(x) + 1$.
            5.1.2.2 broadcast JOIN(y, z).
```

In contrast to link breakages, if node x and y are connected together, then x and y exchange the CONNECT message.

```
6.  On receiving CONNECT($y$, $\Gamma_1(y)$),
    6.1 $\Gamma_1(x) = \Gamma_1(x) \cup \{y\}$.
    6.2 $\Gamma_{1,2}(x) = \Gamma_{1,2}(x) \cup \{y\}$.
    6.3 $d_G(x) = d_G(x) + 1$.
    6.4 if y $\in$ Cluster(x), $d_{\tilde{G}}(x) = d_{\tilde{G}}(x) + 1$.
```

In a dynamic radio network, when a node is deleted, it will send DISCONNECT messages for each link incident to it. When a node is added, it broadcasts CONNECT message. It can ask its neighbors for the information of their 1-hop neighbors. Then it sends JOIN message to join a cluster.

We dedicate two spreading codes for a cluster. One is for data transmission, and the other for immediate acknowledgment. The codes can be reused in another cluster if both clusters are not adjacent. Similarly, we also assign two codes to all edges between two adjacent clusters. The problem of code assignment, which is formulated as the coloring problem, has been studied in [6].

4. TRANSPORT PROTOCOLS

WAMIS have to satisfy heterogeneous traffic requirements over short range radio channels. In the following discussion, we consider the transport of both data and voice traffic in our system. All packets are of fixed length. The channel is assumed slotted. The slot duration, t, is equal to the summation of packet transmission time and channel probing time.

4.1. Collision-free Channel Access within a Cluster

Since our system is distributed and each cluster uses a common channel for packet transmission, we employ a round-robin (RR) scheme, which completely rotates the access priority among the nodes, to control the channel access distributed and conflict-free. The RR scheme gives each node in turn an opportunity to transmit a packet. In addition, the short propagation to transmission time ratio makes CSMA suitable as an access scheme. Thus, we implement RR over CSMA slotted ALOHA (CSMA-RR) for packet transmission within a cluster. In this scheme, if a node, say x, relinquishes its turn to transmit, its 1-hop neighbors contend for this free time slot. The right to transmit in the next time slot passes to the next node of x in logical sequence.

Upon receiving a packet successfully, the receiver transmits an explicit ACK immediately. It can also exploit the ACK packet to piggyback data for the transmitter. On the other hand, upon finishing transmitting a packet, the transmitter listens to the ACK channel to wait for a reception of an ACK. When a time-out occurs, it retransmits the data packet.

We use *piggyback reservations* with packet transmissions to reserve time slots dynamically for packets from active voice sources without conflicts. As a consequence, the nodes with reservation share the channel in a manner closely resembling TDMA. Thus, the throughput is high and the voice packet delay is constrained to meet a specific design limit.

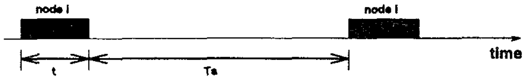


Fig. 10

Let T_s be the maximum interval tolerated between two voice packets. The first packet of a speech session is treated as a data packet and is transmitted by using CSMA-RR, but it has higher priority than data packets to be retrieved from the queue in each node. A voice source schedules its next transmission after a time T_s following a successful transmission, and piggybacks the reservation with the current packet transmission. Fig. 10 shows that node i successfully transmits a voice packet, and it reserves the time for its next transmission. The other nodes are not allowed to contend the reserved slot at that time. So for voice sources, transmission is always collision-free and the maximal delay is guaranteed.

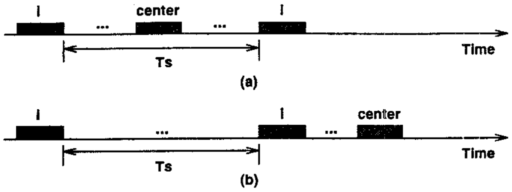


Fig. 11

Because of the two-hop structure of a cluster, the voice source has to make sure whether the other nodes know its reservation before transmitting its packet. To resolve this problem, we have to note that the center node can always keep the new information of reservation, because it can hear any reservation with packet transmission. Furthermore, we assume every node, including the center, places all reservations it knows in the header of its packet. Therefore, once the center transmits its packet, the other nodes can get the new reservation information. The voice source will transmit its packet after center's transmission to guarantee the reservation to be known by the others. There are two cases which would occur. In the first case, Fig. 11(a), the center's transmission before the reserved time slot, node i can follow its schedule to transmit its packet without conflict, since the center forwards the reservation to the other nodes. In the other case, Fig. 11(b) the center's transmission after the reserved time slot, node i can not transmit at the reserved time slot, because the collision is possible now. Only after the center's transmission, the reservation is guaranteed. Slots which are not reserved by voice traffic are accessed according to CSMA-RR protocol.

4.2. Inter-cluster Communications

In addition to the two codes (for information and for ACK) assigned to each cluster, two codes are also assigned to each edge which connects a pair of adjacent clusters for inter-cluster communications. Namely, there are two channels, the transmission (*Xmt*) channel and the acknowledgment (*Ack*) channel, for each pair of adjacent clusters. The access protocol must be able to cope with the limited delay required by voice services.

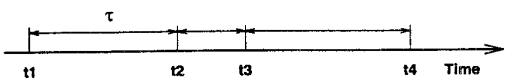


Fig. 12

Each repeater must listen periodically to different codes (in fact, as many codes as its order, as defined in Definition 5). We assume that the repeater, when it is free from voice traffic commitments, shares its time randomly among the various codes.

The access on the *Xmt* channel can be simply carried out in a CSMA unslotted ALOHA fashion. Namely, if the channel is sensed busy, or if the transmission is unsuccessful (no ACK), the packet is regarded as backlogged. As Fig. 12 shows, each backlogged packet repeatedly attempts to retransmit at randomly selected times separated by random delays τ. If the channel is idle at one of these times, the packet is transmitted, which continues until such a transmission is successful. Upon receiving a packet successfully, the receiver uses the *Ack* channel to transmit an explicit ACK packet immediately.

At the first packet transmission, the voice source reports the spreading code which is different from the code used in *Xmt* channel will be used for the following voice packet transmission. Thus, after the first successful transmission, the receiver listens to this spreading code. On the other hand, the voice source schedules its next transmission at a fixed time T_s as intra-cluster communications and uses the piggyback reservation with packet transmissions to reserve the time slot (see Fig. 10). For example of Fig. 13, suppose that a voice stream is transmitting through edge (1, 3). The voice packets are encoded by another spreading code different from the code used in *Xmt* channel. At this time, node 4 can transmit a data packet to node 2 through *Xmt* channel without collision at node 3.

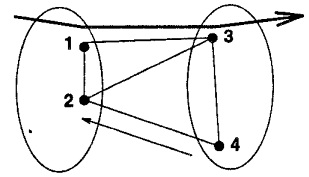


Fig. 13

5. SYSTEM THROUGHPUT

The system throughput is defined as the sum of the throughputs on the links which are simultaneously active in the multicluster network. Uniform traffic matrix is assumed in our simulation. Since the 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. The link throughput is the summation of the throughputs of inter-cluster and intra-cluster. In the simulation, we generate different topologies by populating N nodes in a 100×100 square area randomly. We use various values of the transmission range and for each value we obtain different topologies corresponding to the different placements. For each topology, we can get the throughputs of inter-cluster and intra-cluster. Then we average our throughput measures over the random placements. The average link throughput versus transmission range is shown in Fig. 14.

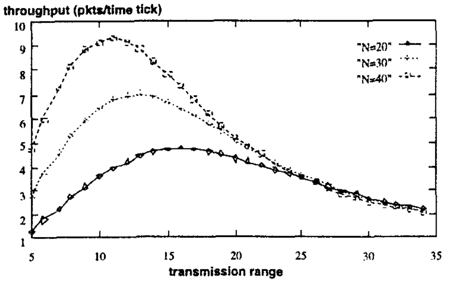


Fig. 14

The results confirm our intuition that there exists a tradeoff between transmission range and throughput. For relatively smaller transmission range, the graph consists of several isolated subgraphs, with good spatial reuse but poor connectivity (see Fig. 7). Too small a transmission range, however, leads to a decrease in throughput since most of the clusters contain only one node, and no link. As the transmission range grows, we have better connectivity but less efficient spatial reuse, and thus lower throughput. In contrast, the behavior of the throughput of the pure slotted ALOHA system is shown in Fig. 15. In this simulation, there are 30 nodes which randomly populate in a 100 × 100 area and use a common spreading code for transmission. Type I (same code) conflict is considered here. Every node uses the same attempt probability which is the probability that a ready node attempts to transmit its packet at each time slot. From the result, we can obviously find the throughput of our system based on CDMA and the multicluster architecture is much better than the slotted ALOHA.

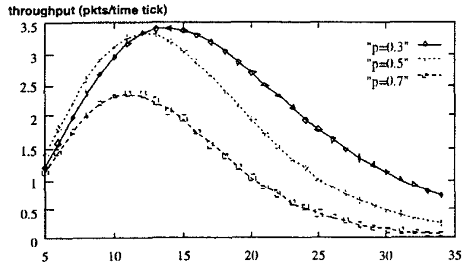


Fig. 15: Throughput of slotted Aloha (one common code and spatial reuse; N=30; p: attempt prob.)

6. CONCLUSIONS

In this paper, we propose a distributed multicluster architecture for transporting multimedia in multihop dynamic packet radio networks. This architecture is not constrained by a fixed infrastructure and, rather, it can be deployed in an environment without infrastructure at all.

In order to reduce the control overhead, we only use a common code within a cluster for packet transmission. Channel accesses by data sources and voice sources are interwoven, with top priority given to voice sources. For voice traffic, the system operates as a TDM system with a cycle time $T_S + t$. Data sources can be regarded as "stealing" the cycle from voice sources. Taking advantage of CDMA, adjacent clusters can transmit at the same time on different codes. The system throughput is much improved over the simpler ALOHA scheme. Finally, simulation experiments have identified key tradeoffs between transmission range and throughput performance, and have shown the advantages of CDMA sharing.

REFERENCES

[1] W. Crowther, R. Rettberg, D. Walden, S. Ornstein, and F. Heart, "A system for broadcast communication: Reservation-ALOHA," *Proc. 6th Hawaii Int. Conf. Syst. Sci.*, pp. 371-374, Jan. 1973.

[2] I. Chlamtac and S. S. Pinter, "Distributed nodes organization algorithm for channel access in a multihop dynamic radio network," *IEEE Transactions on Computers*, pp. 728-737, June 1987.

[3] A. Ephremides, J.E. Wieselthier, and D.J. Baker, "A design concept for reliable mobile radio networks with frequency hopping signaling," *Proceedings of the IEEE*, pp. 56-73, Jan. 1987.

[4] M. Gerla and J.T.-C. Tsai, "Multicluster, mobile, multimedia radio network," to appear in *Wireless Networks Journal.*

[5] D. Goodman and R.A. Valenzuela, K.T. Gayliard and B. Ramamurthi, "Packet reservation multiple access for local wireless communications," *IEEE Transactions on Communications*, pp. 885-890, Aug. 1989.

[6] L. Hu, "Distributed code assignments for CDMA packet radio networks," *IEEE/ACM Transactions on Networking*, pp. 668-677, Dec. 1993.

[7] R. Jain, J. Short, S. Nazareth, L. Kleinrock, and J. Villasenor, "PC-notebook based mobile networking: algorithms, architectures and implementation," *Proceedings of IEEE ICC '95*, pp. 771-777, June, 1995.

[8] C.R. Lin and M. Gerla, "A distributed architecture for multimedia in a multihop dynamic packet radio network," to appear in *Proceedings of IEEE Globecom '95.*

[9] C.R. Lin and M. Gerla, "A distributed control scheme in multi-hop packet radio networks for supporting voice/data traffic," *Proceedings of IEEE ICC'95*, pp. 1238-1242, June. 1995.

[10] R. Pickholtz, D. Schilling, and L. Milstein, "Theory of spread-spectrum communications - a tutorial," *IEEE Transactions on Communications*, pp. 855-884, May 1982.