A Mobility-Based Framework for Adaptive Clustering in Wireless Ad Hoc Networks

A. Bruce McDonald, *Student Member, IEEE,* and Taieb F. Znati, *Associate Member, IEEE*

***Abstract*—This paper presents a novel framework for dynamically organizing mobile nodes in wireless ad hoc networks into clusters in which the probability of path availability can be bounded. The purpose of the (α, t) cluster is to help minimize the far-reaching effects of topological changes while balancing the need to support more optimal routing. A mobility model for ad hoc networks is developed and is used to derive expressions for the probability of path availability as a function of time. It is shown how this model provides the basis for dynamically grouping nodes into clusters using an efficient distributed clustering algorithm. Since the criteria for cluster organization depends directly upon path availability, the structure of the cluster topology is adaptive with respect to node mobility. Consequently, this framework supports an adaptive hybrid routing architecture that can be more responsive and effective when mobility rates are low and more efficient when mobility rates are high.**

***Index Terms*—Ad hoc networks, dynamic clustering, hierarchical routing, mobile computing, mobility models, routing algorithms, wireless networks.**

I. Introduction

ADVANCES in wireless technology and portable computing along with demands for greater user mobility have provided a major impetus toward development of an emerging class of self-organizing, rapidly deployable network architectures referred to as ad hoc networks [2], [12]. An ad hoc network is comprised of wireless nodes and requires no fixed infrastructure. Any device with a microprocessor, whether highly mobile or stationary, is a potential node in an ad hoc network. This includes mobile telephones, motor vehicles, roadside information stations, satellites, and desktop or hand-held computing devices. Unlike existing commercial wireless systems and fixed infrastructure networks, ad hoc networks cannot rely on specialized routers for path discovery and traffic routing. Consequently, mobile end systems in an ad hoc network are expected to act cooperatively to route traffic and adapt the network to the highly dynamic state of its links and its mobility patterns.

Ad hoc networks evolved largely from the DARPA packet-radio (PR) network program [1], [16], [20]. They are expected to play an important role in future commercial and military settings where mobile access to a wired network is either ineffective or impossible. Potential applications for this class of network include instant network infrastructure to support collaborative computing in temporary or mobile environments, mobile patient monitoring for improved critical care, distributed command and control systems, and mobile access to the global Internet. Furthermore, ad hoc networks have the potential to serve as a ubiquitous wireless infrastructure capable of interconnecting many thousands of devices [31] with a wide range of capabilities and uses. In order to achieve this status, however, ad hoc networks must evolve to support large numbers of heterogeneous systems with a wide range of application requirements [5], [21].

Communication between arbitrary endpoints in an ad hoc network requires routing over multiple-hop wireless paths. The main difficulty arises because without a fixed infrastructure, these paths consist of wireless links whose endpoints are likely to be moving independently of one another. Consequently, node mobility causes the frequent failure and activation of links which leads to increased network congestion, while the network's routing algorithm reacts to the topology changes. Unlike fixed infrastructure networks where link failures are comparatively rare events, the rate of link failure due to node mobility is the primary obstacle to routing in ad hoc networks.

The effectiveness of adaptive routing algorithms depends upon the the timeliness and detail of the topology information available to them. However, minimizing the exchange of information is crucial for efficient operation. In an ad hoc network, significant rates of topological change are expected; consequently, the distribution of up-to-date information can easily saturate the network. Furthermore, information arriving late due to latency can drive network routing into instability. Since the rate of link failure is directly related to node mobility, greater mobility increases both the volume of control traffic required to maintain routes and the congestion due to traffic backlogs. Thus, a crucial algorithm design objective to achieve routing responsiveness and efficiency is the minimization of reaction to mobility [27].

Existing schemes for routing in ad hoc networks can be classified according to four broad categories, namely, proactive routing, flooding, reactive routing, and dynamic cluster-based routing. Proactive routing protocols periodically distribute routing information throughout the network in order to precompute paths to all possible destinations. Although this approach can ensure higher quality routes in a static topology, it does not scale well to large highly dynamic networks. By contrast, flooding-based routing requires no knowledge of

Manuscript received June 15, 1998; revised March 17, 1999.

A. B. McDonald is with the Department of Information Science and Telecommunications, University of Pittsburgh, Pittsburgh, PA 15260 USA and the Department of Neurophysiology, Children's Hospital of Pittsburgh, Pittsburgh, PA 15260 USA (e-mail: tudball@neuronet.pitt.edu).

T. F. Znati is with the Department of Computer Science and the Department of Information Science and Telecommunications, University of Pittsburgh, Pittsburgh, PA 15260 USA (e-mail: znati@cs.pitt.edu).

Publisher Item Identifier S 0733-8716(99)04804-0.

network topology. Packets are broadcast to all destinations with the expectation that they will eventually reach their intended target. Under light traffic conditions flooding can be reasonably robust. However, it generates an excessive amount of traffic in large networks, and it is difficult to achieve flooding reliably [30] when the topology is highly dynamic. Consequently, it does not seem that a routing strategy based exclusively on proactive routing or flooding can achieve the objectives required for ad hoc routing.

In a reactive routing strategy, the design objective is accomplished by maintaining paths on a demand-basis using a query–response mechanism. This limits the total number of destinations to which routing information must be maintained, and consequently, the volume of control traffic required to achieve routing. The shortcomings of this approach include the possibility of significant delay at route setup time, the large volume of far-reaching control traffic required to support the route query mechanism, and lower path quality relative to proactive routing. Furthermore, despite the objective of maintaining only desired routes, the route query could propagate to every node in a network during the initial path setup, causing each node to establish paths even when they are only required by certain sources.

In dynamic cluster-based routing, the network is dynamically organized into partitions called clusters, with the objective of maintaining a relatively stable effective topology [21]. The membership in each cluster changes over time in response to node mobility and is determined by the criteria specified in the clustering algorithm. In order to limit far-reaching reactions to topology dynamics, complete routing information is maintained only for intracluster routing. Intercluster routing is achieved by hiding the topology details within a cluster from external nodes and using hierarchical aggregation, reactive routing, or a combination of both techniques. The argument made against dynamic clustering is that the rearrangement of the clusters and the assignment of nodes to clusters may require excessive processing and communications overhead, which outweigh its potential benefits. If the clustering algorithm is complex or cannot quantify a measure of cluster stability, these obstacles may be difficult to overcome.

A desirable design objective for an architectural framework capable of supporting routing in large ad hoc networks subject to high rates of node mobility incorporates the advantages of cluster-based routing and balances the tradeoff between reactive and proactive routing while minimizing the shortcomings of each. Furthermore, the consequences of node mobility suggest the need to include a quantitative measure of mobility directly in the network organization or path selection process. Specifically, a strategy capable of evaluating the probability of path availability over time and of basing clustering or routing decisions on this metric can help minimize the reaction to topological changes. Such a strategy can limit the propagation of far-reaching control information while supporting higher quality routing in highly mobile environments.

The purpose of this paper is to present the (α, t) cluster framework, which defines a strategy for dynamically organizing the topology of an ad hoc network in order to adaptively balance the tradeoff between proactive and demand-based routing by clustering nodes according to node mobility. This is achieved by specifying a distributed asynchronous clustering algorithm that maintains clusters which satisfy the (α, t) criteria that there is a probabilistic bound α on the mutual availability of paths between all nodes in the cluster over a specified interval of time t. In order to evaluate the (α, t) criteria, a mobility model is proposed that characterizes the movement of nodes in large ad hoc networks. It is shown how this model is used to determine the probability of path availability when links are subject to failure due to node mobility.

Based on the (α, t) cluster framework, intracluster routing requires a proactive strategy, whereas intercluster routing is demand-based. Consequently, the framework specifies an adaptive-hybrid scheme whose balance is dynamically determined by node mobility. In networks with low rates of mobility, (α, t) clustering provides an infrastructure that is more proactive. This enables more optimal routing by increasing the distribution of topology information when the rate of change is low. When mobility rates become very high, cluster size will be diminished and reactive routing will dominate. The (α, t) cluster framework decouples the routing algorithm specification from the clustering algorithm, and thus, it is flexible enough to support evolving ad hoc network routing strategies [13], [15], [27], [29] in both the intra- and intercluster domains.

The remainder of the paper is organized as follows: Section II presents a review of the significant contributions in the area of dynamic clustering for ad hoc networks. The characterization of the (α, t) cluster and the cluster routing methodology is described in Section III. Details of the the (α, t) cluster algorithm are presented in Section IV. The mobility model used to characterize link and path availability is developed in Section V, and simulation results demonstrating the effectiveness of the (α, t) cluster framework are presented in Section VI. Finally, conclusions of this work are presented in Section VII.

II. Related Work

Several dynamic clustering strategies have been proposed in the literature [10], [21], [25], [31]. While these strategies differ in the criteria used to organize the clusters and the implementation of the distributed clustering algorithms, none of the proposed schemes uses prediction of node mobility as a criteria for cluster organization. Clustering decisions in each of these schemes are based on static views of the network at the time of each topology change. Consequently, they do not provide for a quantitative measure of cluster stability. In contrast, the (α, t) cluster strategy forms the cluster topology using criteria based directly on node mobility. According to [31], the ability to predict the future state of an ad hoc network comprised of highly mobile nodes is essential if the network control algorithms are expected to maintain any substantive quality-of-service (QoS) guarantees to real-time connections.

The multimedia support for wireless network (MMWN) system proposed by Ramanathan and Steenstrup [31] is based upon a hybrid architecture that includes the characteristics of

ad hoc and cellular networks. Their framework uses hierarchical routing over dynamic clusters that are organized according to a set of system parameters that control the size of each cluster and the number of hierarchical levels. Aggregation of routing information is used to achieve scalability and limit the propagation of topological change information. A multilevel strategy is used to repair virtual circuit (VC) connections that have been disturbed due to node mobility. MMWN does not predict node movement. Consequently, it is unable to provide a quantitative bound on the stability of its cluster organization.

Krishna *et al.* [25] proposed a scheme that dynamically organizes the topology into k clusters, where nodes in a cluster are mutually reachable via k-hop paths. The algorithm considers $k = 1$ and reduces to finding cliques in the physical topology. Using a first-fit heuristic, the algorithm attempts to find the largest cliques possible. Although the algorithm does not form optimal clusters, it still requires a three-pass operation each time a topology change occurs: one for finding a set of feasible clusters, a second for choosing the largest of the feasible clusters that are essential to maintain cluster connectivity, and a third to eliminate any existing clusters that are made superfluous by the new clusters.

The objective of the scheme proposed by Lin and Gerla [21] differs significantly from the previous examples. Rather than using clustering to minimize the network's reaction to topological changes, their scheme is intended to provide controlled access to the bandwidth and scheduling of the nodes in each cluster in order to provide QoS support. Hierarchical routing and path maintenance were a secondary concern. The proposed algorithm is very simple and uses node ID numbers to deterministically build clusters of nodes that are reachable by two-hop paths.

The zone routing protocol (ZRP) proposed by Haas and Pearlman [13] is a hybrid strategy that attempts to balance the tradeoff between proactive and reactive routing. The objective of ZRP is to maintain proactive routing within a zone and to use a query–response mechanism to achieve interzone routing. In ZRP, each node maintains its own hop-count constrained routing zone; consequently, zones do not reflect a quantitative measure of stability, and the zone topology overlaps arbitrarily. These characteristics differ from (α, t) clusters, which are determined by node mobility and do not overlap. Both strategies assume a proactive routing protocol for intrazone/cluster routing, and each organizes its topology based upon information maintained by that protocol. ZRP also defines the query control scheme to achieve interzone routing. Although ZRP is not a clustering algorithm and the (α, t) cluster framework is not a routing protocol, the comparison demonstrates a close relationship that could be leveraged by incorporating the (α, t) cluster into ZRP. The use of (α, t) clusters in ZRP could achieve more efficient and adaptive hybrid routing without significantly increasing its complexity.

III. (α, t) Cluster Framework

Hierarchical routing has been shown to be essential in order to achieve at least adequate levels of performance in very large networks [17], [18]. In fixed infrastructure networks, hierarchical aggregation achieves the effect of making a large network appear much smaller from the perspective of the routing algorithm. Cluster-based routing in ad hoc networks can also make a large network appear smaller, but more importantly, it can make a highly dynamic topology appear much less dynamic. Unlike the cluster organization of a fixed network, the organization of an ad hoc network cannot be achieved offline. The assignment of mobile nodes to clusters must be a dynamic process wherein the nodes are self-organizing and adaptable with respect to node mobility. Consequently, it is necessary to design an algorithm that dynamically implements the self-organizing procedures in addition to defining the criteria for building clusters.

The objective of the (α, t) cluster framework is to maintain an effective topology that adapts to node mobility so that routing can be more responsive and optimal when mobility rates are low and more efficient when they are high. This is accomplished by a simple distributed clustering algorithm using a probability model for path availability as the basis for clustering decisions. The algorithm dynamically organizes the nodes of an ad hoc network into clusters where probabilistic bounds can be maintained on the availability of paths to cluster destinations over a specified interval of time.

The (α, t) cluster framework can also be used as the basis for the development of adaptive schemes for probabilistic QoS guarantees in ad hoc networks. Specifically, support for QoS in time-varying networks requires addressing: 1) connection-level issues related to path establishment and management to ensure the existence of a connection between the source and the destination and 2) packet-level performance issues in terms of delay bounds, throughput, and acceptable error rates. Ideally, it is desirable to guarantee that the QoS requirements of ongoing connections are preserved for their entire duration. Unfortunately, this is not possible in a time-varying network environment as connections may fail randomly due to user mobility. A more realistic and practical approach is to provide some form of probabilistic QoS guarantees by keeping connection failures below a prespecified threshold value and by ensuring with high probability that a minimum level of bandwidth is always available to ongoing connections.

Based upon the intracluster routing model proposed in Section III-B, and using the estimates of path availability and other link status metrics provided through the routing algorithm, a connection admission control algorithm could determine with high probability whether or not sufficient resources are available to support the requirements of an intracluster connection over a specific period of time. In order to achieve similar QoS guarantees across the intercluster domain, the (α, t) cluster framework could be extended to support a dynamic hierarchical architecture in which resource information within each cluster is aggregated and a second level of clustering algorithm maintains (α, t) paths between clusters using virtual links.[1] Hierarchical QoS-based routing and admissions control schemes are not considered further in this paper.

[1] A virtual link represents the set of physical links that connect nodes in one cluster to nodes in another cluster.

The remainder of this section is organized as follows: the (α, t) cluster is formally characterized in Section III-A. The implementation of routing is discussed in Section III-B. Finally, a methodology for selecting the system parameters α and t is presented in Section III-C.

A. (α, t) Cluster Characterization

The basic idea of the (α, t) cluster strategy is to partition the network into clusters of nodes that are mutually reachable along cluster internal paths[2] that are expected to be available for a period of time t with a probability of at least α. The union of the clusters in a network must cover all the nodes in the network.

Definition 1: Let $\mathcal{P}^k_{m,n}(t)$ indicate the status of path k from node n to node m at time t. $\mathcal{P}^k_{m,n}(t) = 1$ if all the links in the path are active at time t, and $\mathcal{P}^k_{m,n}(t) = 0$ if one or more links in the path are inactive at time t. The path availability $\pi^k_{m,n}(t)$ between two nodes n and m at time $t \geq t_0$ is given by the following probability expression:

$$\pi^k_{m,n}(t) \equiv \Pr\big(\mathcal{P}^k_{m,n}(t_0 + t) = 1 | \mathcal{P}^k_{m,n}(t_0) = 1\big).$$

Definition 2: Let $\pi^k_{m,n}(t)$ be the path availability of path k from node n to node m at time t. Path k is defined as an (α, t) path if and only if

$$\pi^k_{m,n}(t) \geq \alpha.$$

Definition 3: Node n and node m are (α, t) available if they are mutually reachable over (α, t) paths.

Definition 4: An (α, t) cluster is a set of (α, t) available nodes. Definition 4 states that every node in an (α, t) cluster has a path to every other node in the cluster that will be available at time $t_0 + t$ with a probability $\geq \alpha$. The cluster characterization, as previously defined, requires a model which quantifies the (α, t) path availability as given in Definition 1. Path availability is a random process that depends upon the mobility of the nodes which lie along a given path. Consequently, the mobility characteristics of the nodes play an important role in the characterization of this process. In Section V, a mobility model for large ad hoc networks is proposed, and the probability distributions for the aggregate distance and trajectory covered by a node over time are derived. These distributions provide the basis for developing analytical models for link availability. It is also shown how this model can be used to derive expressions for path availability which can be efficiently evaluated by the (α, t) cluster algorithm.

B. (α, t) Cluster Routing Methodology

The logical relationship between the (α, t) cluster algorithm, the routing algorithm, and the other network-layer entities is depicted in Fig. 1. The cluster algorithm resides logically between the routing-layer and the Internet MANET[3]

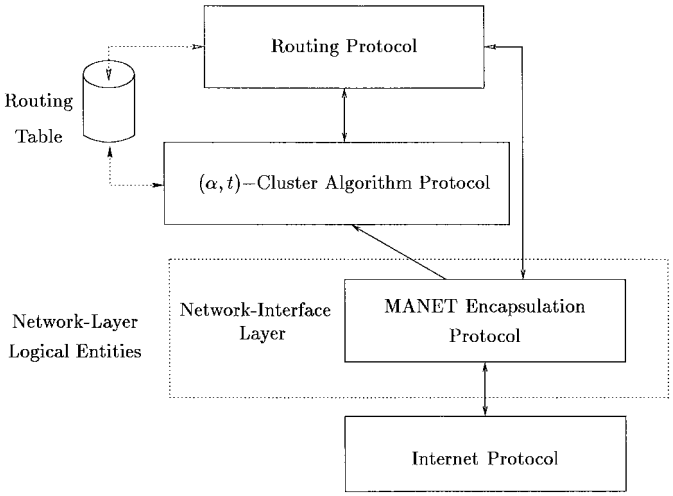


Fig. 1. Logical relationships among MANET network-layer entities.

encapsulation protocol (IMEP)[4] [6]. As such, the cluster algorithm presents a logical topology to the routing algorithm, and it accepts feedback from the routing algorithm in order to adjust that logical topology and make clustering decisions. To support the (α, t) cluster framework, IMEP or an equivalent protocol must identify a node's cluster identifier number (CID) to neighboring nodes and include the CID in the encapsulation of the routing information packets. A protocol that provides the functionality of IMEP along with these enhancements will be referred to in this paper as a network-interface layer protocol.

A two-level routing algorithm adaptively subdivides the task of establishing and maintaining routes to mobile destinations. Intracluster routing uses a proactive strategy, whereby each node in a cluster maintains topology information and routes to every cluster destination for the duration of the time that the node remains in a given cluster. Routes to destinations outside of a node's cluster are established on a demand basis only. Consequently, a reactive routing strategy must be implemented to setup intercluster routes.

1) Intracluster Routing: Intracluster routing can be implemented with any distributed routing algorithm that can proactively maintain routes to a set of mobile destinations. Similar to ZRP, which uses hop-count [11], the (α, t) cluster uses a path availability based membership to limit the propagation of routing updates. Those MANET protocols that have been designed specifically to operate as reactive protocols can still function as intracluster protocols in which the demand for a route is produced by the cluster algorithm. However, because the (α, t) criteria establishes a lower bound on the availability of cluster paths, preference is given to those algorithms capable of incorporating link and path availability information as a routing metric and of using maximum availability as an optimization criteria when establishing paths. Arguments have been made against path optimization in ad hoc networks [7], [27]; however, these arguments are based upon the assumption of a monolithic network without clustering. (α, t) clusters gradually adapt the cluster topology to maintain a consistent

[2] A cluster internal path consists exclusively of nodes that are members of the cluster.

[3] A MANET is a mobile ad hoc network.

[4] The IMEP layer is designed to provide services to upper-layer network entities such as link status sensing, neighbor discovery, one-hop neighbor broadcast, control packet aggregation, and address resolution.

level of path stability such that path optimization becomes effective.

In Section V-C, it is shown how the path availability can be calculated from the individual link availabilities along the path. This can be accomplished based on aggregated path information using a modified Bellman–Ford algorithm to find the maximum availability path or using Dijkstra's algorithm if complete link status information is available. Because path availability is a time-varying quantity that depends on the individual link properties, it is more efficient if the characteristics of the links are known at each node. To overcome the shortcomings of link-state protocols, several alternatives that provide complete link status information along selected paths have been proposed for use in highly dynamic environments. Examples that are well-adapted for intracluster routing include the link vector algorithm (LVA) [9] and the wireless routing protocol (WRP) [24]. Based on complete link status information, it is possible to estimate the current link availability anywhere in the cluster without periodic routing updates. If aggregated routing information is used, the path availability information will become outdated without periodically updating the path status.

A node's membership in a cluster is implied by the contents of the routing tables distributed among the active nodes in that cluster. No distinct cluster table is required since it is the routing algorithm as modulated by the cluster algorithm which determines the set of nodes that are in the cluster. Consequently, a node's routing table gives a complete picture of its current view of the cluster. Accordingly, cluster convergence is simply a matter of the convergence of the routing tables in the cluster.

2) Intercluster Routing: Intercluster routing is achieved using a demand-based protocol that establishes paths by executing a path-search query and response algorithm. With respect to this process, each node can be considered as a route cache for the set of nodes in its cluster. In the worst case, the response phase will begin as soon as the first node in the target destination's cluster receives the route query. Queries will never be propagated further into the cluster in which the target destination resides.

One methodology for maintenance of end-to-end routing between destinations in different clusters is direct implementation of a flat-routed reactive routing protocol, such as the temporally ordered routing algorithm (TORA) [26], [27] or the ad hoc on-demand distance vector algorithm (AODV) [28]. Specifically, each node requiring a route first searches for the desired destination in its cluster routing table that is proactively maintained by the intracluster routing protocol. If the destination is not found, the node initiates a route discovery process if it is the source node, or it propagates the request if it is processing another node's route query. As such, every node will participate in two routing protocols: one within a cluster and one for noncluster destinations. Consequently, each node will be able to maintain routes to any connected destination.

The problem with the flat-routed reactive approach is that it fails to take advantage of the cluster topology, which could be used to more efficiently manage the route discovery and maintenance processes. To address this shortcoming, an improved methodology for (α, t) cluster interconnection is proposed that leverages the cluster topology in order to more efficiently discover and maintain end-to-end routing between nodes in different destinations by adapting the ZRP interzone routing protocol (IERP). IERP assumes a topology comprised of a sequence of overlapping zones and specifies a bordercasting technique that is used to efficiently construct routes across multiple zones. However, IERP as defined for ZRP cannot be directly applied to the interconnection of (α, t) clusters since these clusters are designed not to overlap. To bridge the differences between the requirements of IERP and the properties of the (α, t) clusters, the following adaptations are required.

1) The border nodes of an (α, t) cluster consist of the set of nodes which are adjacent to nodes that are not members of the same cluster. Each border node treats adjacent clusters[5] as supernodes and advertises reachability to those supernodes within their (α, t) cluster. Consequently, each node in a cluster has knowledge of, and proactively maintains routes to, the set of adjacent clusters identified by the border nodes.
2) The bordercasting process defined by IERP must be modified to allow the exchange of the route query from the egress border node of one cluster to the ingress border node of its adjacent clusters.
3) Each (α, t) cluster egress border node processing an IERP route query will append its CID to the route query and forward one copy of the query to those neighboring nodes which are ingress border nodes in adjacent clusters. Consequently, a sequence of CID's is accumulated, which represents an intercluster route to the desired destination.
4) Each (α, t) cluster ingress border node searches its routing table for the destination. If the destination is found, the node appends its CID to the accumulated sequence of CID's in the route query and returns it in the response message that is sent back along the accumulated sequence of clusters in reverse order.

Unlike ZRP, this modified scheme builds a route as a sequence of clusters rather than nodes. The specific paths across each cluster are determined dynamically by the intracluster routing algorithm. Since each node in a cluster maintains routes to the set of adjacent clusters, this methodology provides a strategy which is robust, such that routes will remain viable so long as the cluster adjacencies remain intact—even if the specific border nodes change. Thus, it is highly adaptive with respect to node mobility and requires less reactive route maintenance.

C. (α, t) Cluster Parameters

Evaluation of (α, t) paths requires specification of two system parameters, α and t. The effects of these parameters are tightly coupled, making it difficult to select optimal values. Large values for t seem desirable, as they imply more cluster stability and reduce the computational requirements of cluster

[5] An adjacent cluster to a node n is one with a border node that is adjacent to a border node in the node n's cluster.

maintenance. However, large t will drive down the path availability between nodes of the cluster for the same mobility patterns, which makes it more difficult to achieve the required lower bound α. Consequently, large values of t will tend to result in smaller clusters, whereas small values of t will increase the cluster size, which results in more optimal routing with increased routing overhead.

Since α establishes a lower bound on the probability that a given cluster path will remain available for time t, it controls the cluster's inherent stability. Thus, for a given level of stability, the role of t is to manage the cluster size, which controls the balance between routing optimality and efficiency. Given that no single pair of values for α and t can be optimal or even sufficient in all situations, at least one of the parameters should be adaptive. In particular, appropriate bounds on path availability should consider the level of traffic and possibly the QoS requirements of connections routed through the cluster in order to ensure a sufficient level of cluster stability that will support those connections with high probability. In the remainder of this section, a methodology is proposed for adaptive maintenance of the system parameter α.

The previous observations suggest that the value chosen for α should reflect traffic conditions. Assuming that path availability is an ergodic process, it represents the average proportion of time an (α, t) path is available to carry data. Consequently, α places a lower bound on the effective capacity of the path over an interval of length t. This bound must be sufficient to support the current traffic load. If QoS support is considered, the bound must be high enough to support the bandwidth which has been allocated to real-time connections.

If the average packet delay at a node along an (α, t) path exceeds the availability of that path, excessive queuing delays may be incurred in the network. Consequently, a necessary condition for satisfying the traffic requirement is to establish an effective capacity over any interval of length t, which results in average delays that are significantly less than t. Based on this observation, queuing models can be used to determine a lower bound for path availability. The development of complete queuing models is beyond the scope of this paper; however, a simplified model is presented next in order to illustrate this concept.

Assume, without loss of generality, that t is identical at every node in a cluster. If the cluster's topology remains stable over the interval of length t, then routing will be deterministic during this interval, and standard assumptions [18] permit the ad hoc network to be modeled as a network of Jackson queues [19].

Let the link capacity be C bits/s and the mean packet length $1/\mu$ bits. The effective packet service rate μ_{eff} over the interval t can be determined based upon the path availability according to (1). Based on the Jackson model, each node can be treated as an independent M/M/1 queue. Using knowledge of the current aggregate arrival rate λ and the effective service rate μ_{eff}, the M/M/1 results can be applied to find the mean total packet delay T. Since this delay must be less than t, this approach establishes a lower bound on the path availability, as shown in (4)

$$\mu_{eff} = \alpha C \mu \tag{1}$$

$$T = \frac{1}{\mu_{eff} - \lambda} \tag{2}$$

$$t \geq \frac{1}{\alpha C \mu - \lambda} \tag{3}$$

$$\alpha \geq \left(\frac{1 + \lambda t}{\mu t C} \right). \tag{4}$$

An effective adaptive strategy for determining the value of α controls the minimum level of cluster stability required to support the traffic load and QoS requirements of established connections. The choice of the parameter t is a system design decision that determines the maximum cluster size achievable for different rates of mobility when no traffic exists in the network.

IV. (α, t) Cluster Algorithm

Two key requirements motivate the design of a successful dynamic clustering algorithm: 1) the algorithm should achieve a stable cluster topology and 2) it should do so with minimal communications overhead and computational complexity. Consequently, in a highly dynamic environment, the algorithm should be distributed, operate asynchronously, and require minimal coordination among the nodes. Furthermore, it is highly unlikely that an optimal cluster topology will be achievable. Therefore, optimal clustering should not be a concern—rather an egalitarian view of clustering should be adopted with the objective of achieving good clusters. This means that clusters are stable relative to the overall topology, that clustering decisions are made fairly, and that the cluster topology converges to meet the clustering criteria. Finally, the algorithm should be self-starting and robust, in that after finite periods of network instability, it eventually converges to a stable and efficient clustered topology.

Clearly, if the underlying topology is so unstable that flooding is the only viable routing strategy, then no algorithm will be able to achieve the key requirements. Furthermore, if the topology is static or quasi-static, then the nature of clustering, and thus the design criteria, will be substantially different. In this case, optimal clustering can be achieved either with offline approaches or using centralized algorithms based on complete topological information. Somewhere in between these two extremes lies the domain for dynamic clustering. This is where the (α, t) cluster algorithm is designed to operate.

(α, t) clusters are dynamic entities that are created, expanded, contracted, and eventually terminated based upon routing information that is maintained on a set of cooperating mobile nodes. Other than the dissemination of topology information by the intracluster routing protocol, the cluster algorithm does not require any additional message types; however, it must be able to trigger a routing update when joining or leaving a cluster. The strength of the (α, t) cluster algorithm is that it is minimal and requires no far-reaching internodal coordination when initiating clustering activity. Its role is to modulate the actions of the intracluster routing

algorithm by effectively filtering its view of the network. This is achieved by manipulating the node's CID and exploiting functionalities that already exist or can easily be incorporated into existing protocols at the routing and network-interface layers.

A distributed asynchronous algorithm is specified for maintaining (α, t) clusters. The algorithm is simple, efficient, and self-starting. Every node in a cluster participates in a proactive routing protocol wherein the scope of routing information propagation is controlled by the nodes' view of their cluster membership. A node neither processes nor propagates routing information from nodes that are not identified as belonging to its own cluster. However, routing information from nodes that do belong to its cluster is processed and disseminated. No centralized control over the clustering process is required. Nodes can asynchronously join, leave, or create clusters. The algorithm is event driven, and its actions depend upon the nodes' ability to satisfy the (α, t) criteria with respect to their current cluster or the cluster they are attempting to join.

The (α, t) cluster algorithm is driven by both hard-state and soft-state events. Specifically, topological changes, which are detected locally or learned through routing updates, trigger specific actions by the algorithm. Hard-state events include node activation, node deactivation, link activation, and link failure. In general, the algorithm requires clustered nodes to determine whether or not the (α, t) criteria continues to be satisfied following a topological change. Additionally, soft-state is maintained at each node through the use of a timer referred to as the α timer. This timer determines the maximum time t for which the node can guarantee path availability to each cluster destination with probability $\geq \alpha$. The expiration of the α timer is treated by the algorithm as a topological change requiring the node to reevaluate the (α, t) criteria with respect to its cluster.

All cluster actions are implied by information received through routing and network-interface layer protocol information. In the remainder of this section, the five events which drive the (α, t) cluster algorithm, namely, node activation, link activation, link failure, expiration of the α timer and node deactivation, and the actions taken by a node in response to each of these events, are described. The section concludes with a discussion of the major properties of the (α, t) cluster algorithm.

A. Node Activation

The primary objective of an activating node[6] is to discover an adjacent node and join its cluster. In order to accomplish this, it must be able to obtain topology information for the cluster from its neighbor and execute its routing algorithm to determine the (α, t) availability of all the destination nodes in that cluster. The source node can join a cluster if and only if all the destinations are reachable via (α, t) paths. Such a cluster is referred to as a feasible cluster. The source node will continue checking each neighbor in sequence until it finds a feasible cluster or runs out of neighbors. If the source node is unable to join a cluster, it will create its own cluster, referred to as an orphan cluster, and wait for another opportunity to cluster with other nodes.

The first step upon node activation is the initialization of the source node's CID to a predefined value which indicates its unclustered status. The network-interface layer protocol is required to advertise the node's CID as part of the neighbor greeting protocol [23] and in the header of the encapsulation protocol. This enables nodes to easily identify the cluster status and membership of neighboring nodes and of the source of the routing updates—a necessary function to control the dissemination of routing information.

When its network-interface layer protocol identifies one or more neighboring nodes, the source node performs the following actions. First, the source node identifies the CID's associated with each neighbor. Next, it evaluates the link availability associated with each neighbor according to either a system default mobility profile or mobility information obtained through the network-interface layer protocol or physical-layer sensing. The precise methodology and the information required for the evaluation of link availability is described in Section V-C. Finally, the neighbors, having discovered the unclustered status of the source node, automatically generate and transmit complete cluster topology information, which they have stored locally as a result of participating in the cluster's intracluster routing protocol. This topology synchronization function is a standard feature of typical proactive routing protocols when a router discovers the activation of a link to a new router. The source node does not immediately send its topology information to any of the neighbors.

Having completed the previous actions, the source node proceeds according to the cluster status of the identified neighbors. If none of the neighbors are clustered, the source node sets a randomized backoff timer, during which time it delays any further clustering activity. The purpose of this timer is to effectively spread out the clustering of nodes that have activated more or less simultaneously. This minimizes the probability that each of these nodes is forced to create an orphan cluster. If the source node has identified one or more adjacent clusters, it will evaluate each such cluster's feasibility in turn. The precise algorithm steps for evaluating cluster feasibility depend upon the nature of the topology information—distance-vector or link-state—and the routing algorithm. If the source node determines that a cluster is feasible, it joins that cluster.

The cluster-join action is achieved asynchronously without any additional internodal coordination. The source node sets its CID to equal the CID of the cluster it is joining, and it generates its own routing update that is broadcast to its neighbors. Recognizing their own CID's in the routing update, those neighbors that are members of the target cluster process the source node's routing update. In doing so, the routing protocol automatically adds the source node as a destination in their respective routing tables, which infers cluster membership.

If the source node's network-interface layer protocol detects no adjacent nodes, or its attempts to join an adjacent cluster fail due to cluster infeasibility, the cluster algorithm generates and sets a globally unique CID that will be used in subsequent neighbor greeting exchanges. In this orphaned state, the (α, t)

[6] The activating node will be referred to as the source node.

```
Procedure Node_Activation(node)
begin
    if (node has no neighbors)
    {
      Initiate_Cluster();
      node_status = CLUSTERED ;
    }
    else
    {
      node_status = UNCLUSTERED;
      found_clustered = false ;
      for (each neighbor N)
      {
        if (N's CID != UNCLUSTERED)
        {
          found_clustered = true;
          if (Cluster_Is_Feasible (N's cluster))
          {
            Join_Cluster(N's cluster);
            node_status = CLUSTERED;
            break;
          }
        }
      }
      if (node_status == UNCLUSTERED)
      {
        if (found_clustered == true)
        {
          Initiate_Cluster();
        }
        else
        {
          Sleep(random_backoff_time);
          Node_Activation(node);
        }
      }
    }
end
```

Fig. 2. Outline of node activation algorithm.

criteria is trivial because the path availability of the source node to itself is always 1.0. In order to periodically reattempt to join a neighboring cluster, the node's α timer is set to the value of the system parameter t. Fig. 2 shows pseudocode for the algorithm executed by the source node upon activation.

B. Link Activation

A link activation detected by a clustered node that is not an orphan is treated as an intracluster routing event. Hence, the topology update will be disseminated throughout the cluster. Unlike reactive routing that responds after path failure, the dissemination of link activation updates is a key factor to an (α, t) cluster node's ability to find new (α, t) paths in anticipation of future link failures or the expiration of the α timer.

The objective of an orphan node is to either have its own cluster expanded through the actions of other nodes or to join an existing cluster unless node mobility is very high. Link activation triggers an orphan node's attempt to join a cluster. In order to receive cluster topology information from its new neighbor, the orphan node must temporarily reset its CID to indicate its unclustered status. Only information received from nodes that are in the same cluster as a destination or in the unclustered state are passed by the cluster algorithm protocol

```
Procedure Link_Activation(node)
begin
    if (node is an orphan)
    {
      tmp_CID = CID;
      CID = UNCLUSTERED;
      node_status = UNCLUSTERED;
      for (each neighbor N)
      {
        if (N's CID != UNCLUSTERED)
        {
          if (Cluster_Is_Feasible(N's cluster))
          {
            Join_Cluster(N's cluster);
            node_status = CLUSTERED;
            break;
          }
        }
      }
      if (node_status == UNCLUSTERED)
      {
        CID = tmp_CID;
        node_status = CLUSTERED;
      }
    }
end
```

Fig. 3. Outline of link activation algorithm.

to the routing layer (see Fig. 1). Thus, by changing its CID, the orphan node triggers the transmission of routing updates from its neighbor. Upon receiving the cluster topology information, the node evaluates cluster feasibility and either joins the cluster or returns to its orphan cluster status, depending upon the outcome of the evaluation. Fig. 3 shows the pseudocode for the algorithm executed by an orphan node upon detecting a link activation.

C. Link Failure

The objective of a node detecting a link failure is to determine if the link failure has caused the loss of any (α, t) paths to destinations in the cluster. A node's response to a link failure event is twofold. First, each node must update its view of the cluster topology and reevaluate the path availability to each of the cluster destinations remaining in the node's routing table. Second, each node forwards information regarding the link failure to the remaining cluster destinations. This second action is a function of the routing protocol. Each node receiving the topology update reevaluates its (α, t) paths as if it had directly experienced the link failure. When evaluating path availability to destination nodes within the cluster following a topology change, it is necessary to adjust the timing parameter to reflect that the α timer has not yet expired. Use of the full value of t would unnecessarily penalize the nodes by requiring a path availability that is higher (further out in time) than required by the cluster criteria. Thus, the estimated availabilities will reflect the probabilities evaluated at the maximum time for which this node has already made its probabilistic guarantee.

Using the topology information available at each node, the current link availability information is estimated, and maximum availability paths are calculated to each destination node in the cluster. If the node detects that a destination has become

unreachable, then the node assumes that the destination has deactivated or otherwise departed from the cluster. In this case, the destination is removed from the node's routing table and will not be considered further in the evaluation of (α, t) paths. If a node detects that any of the remaining cluster nodes are connected within the cluster but not (α, t) reachable, it will voluntarily leave the cluster. A node leaves a cluster by sending a routing update to its neighbors that indicates that the status of all its links are down or equivalently an infinite distance to itself. It then resets its own CID to the unclustered value and proceeds according to the rules for node activation. No further action is required following a link failure if the node successfully evaluated (α, t) paths to each destination in the cluster. Fig. 4 presents the pseudocode of the algorithm executed by a node that detects a link failure through the services of the network-interface layer protocol or receives a topology update which reflects a link failure.

```
Procedure Link_Failure(node)
begin
    t = time_remaining_on(alpha_timer);
    if (Cluster_Is_Feasible(my_cluster))
    {
      /* status quo */
    }
    else
    {
      Leave_Cluster(my_cluster);
      node_status = UNCLUSTERED;
      CID = UNCLUSTERED;
      found_cluster = flase;
      for (each neighbor N)
      {
        if (N's CID != UNCLUSTERED)
        {
          found_cluster = true;
          if (Cluster_Is_Feasible(N's cluster)
          {
            Join_Cluster(N's cluster);
            node_status = CLUSTERED;
            break;
          }
        }
      }
      if (node_status == UNCLUSTERED)
      {
        if (found_cluster == true)
        {
          Initiate_Cluster();
        }
        else
        {
          Sleep(random_backoff_time);
          Node_Activation(node);
        }
      }
    }
end
```

Fig. 4. Outline of link failure algorithm.

D. Expiration of α Timer

The α timer controls cluster maintenance through periodic execution of the intracluster routing algorithm at each node in a cluster. Using the topology information available at each node, the current link availability information is estimated and maximum availability paths are calculated to each destination node in the cluster. If any of the paths are not (α, t) paths, then the node leaves the cluster. α timer-based cluster maintenance is asynchronous and requires no internodal communications other than the action required for a node's departure from its cluster. The precise actions taken by a node upon the expiration of its α timer are virtually identical to the actions taken by a node when it detects a link or node failure, with the exception that the α timer triggers an orphan node to reattempt to join a cluster in a manner that is identical to link activation.

Cluster maintenance based on α timer expiration accomplishes two fundamental objectives in the (α, t) cluster framework. First, it provides the mechanism by which nodes proactively seek to extend their path availability guarantees—thereby providing the basis for achieving cluster stability. Nodes that are unable to do this leave the cluster—with the objectives of shrinking the cluster in order to improve its stability for the remaining nodes and finding a better cluster for itself. Second, α timer maintenance leads to topological synchronization, which provides the basis for cluster convergence. The issue of cluster convergence is discussed further in Section IV-F.

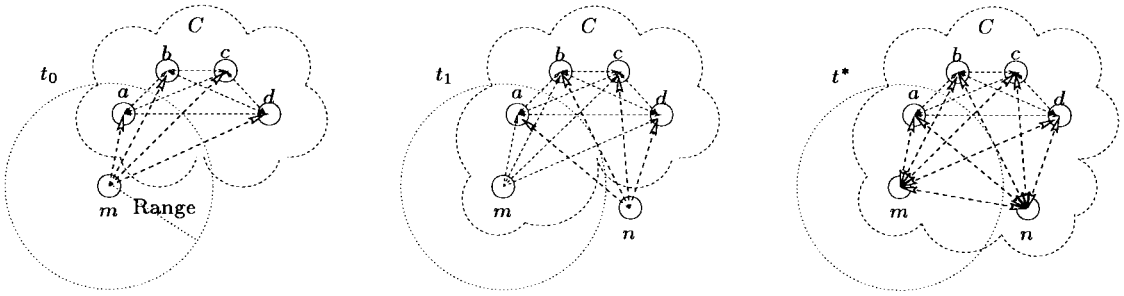
Since each node in a cluster maintains an independent α timer, which is started when a node joins the cluster, a natural phase shift exists across the nodes in a cluster, which produces the desirable effect of distributing the collective reactions to changes in path availability. This allows for gradual adjustments in cluster routing and membership. Because it is treated as a topology change, the pseudocode in Fig. 4 also applies to α timer expiration.

E. Node Deactivation

The event of node deactivation encompasses four related events, namely, graceful deactivation, sudden failure, cluster disconnection, and voluntary departure from the cluster. In general, each of these events triggers a response by the routing protocol. As a result, nodes determine that the node that has deactivated is no longer reachable.

In the cases of graceful deactivation and voluntary departure, the deactivating node announces its departure by disseminating a topology update to all the nodes in the cluster, which indicates the failure of all its incident links. Nodes receiving this status update effectively erase the node from their own view of the cluster.

If a node becomes disconnected from the cluster due to mobility, or the node fails suddenly, the response of the nodes will depend on the specific sequence of events that lead to the convergence of routing in the cluster. A node can recognize for itself that is has become cluster disconnected by virtue of losing paths to the entire set of nodes in the cluster. Hence, it becomes an orphan node and proceeds according to the rules previously described for an activating node. However, the remaining nodes in its original cluster may not immediately be able to determine that this node is unreachable and will attempt to reevaluate their (α, t) paths to the destination. In this case, these nodes may determine that the destination is no longer reachable via an (α, t) path, and consequently, they



(a) (b) (c)

Fig. 5. Convergence of (α, t) cluster algorithm. (a) Node m joins cluster C. (b) Node n joins cluster C. (c) Cluster C converges.

will also leave the cluster voluntarily, which is a recoverable, although suboptimal, response.

Each node receiving a topology update that reflects the node deactivation in the cluster uses the new topology information in order to evaluate (α, t) paths to the remaining cluster-connected destinations in the cluster. Should the node fail to find an (α, t) path to any of the destination nodes in the cluster, it leaves the current cluster and proceeds according to the rules of node activation previously described. These actions are also reflected by the pseudocode in Fig. 4.

F. Discussion

The design of the (α, t) cluster algorithm was predicated upon two basic tenets: 1) optimality is inherently difficult or impossible to achieve in highly dynamic environments that are constrained by the limitations of the physical media and 2) efficiency is more important than optimality in these environments in order to achieve acceptable levels of performance. Consequently, the overriding design principle was based on this fundamental tradeoff.

Specifically, two observations can be made about the (α, t) cluster algorithm with respect to the aforementioned design tradeoffs. First, no attempt was made to maintain or specify the criteria for optimal cluster organization. This is a difficult problem even in a fixed topology network in which cluster size is typically used as the basis for optimization. Second, minimization of internodal coordination and communications was substantially more important than ensuring complete consensus at all times with respect to each clustering decision, so long as the cluster topology converges under stable conditions. Furthermore, the rate of topology change and network latency make synchronization of each clustering action through distributed consensus protocols infeasible—even if bandwidth and processing resources are plentiful. The remainder of this section presents discussion of the properties of the (α, t) cluster algorithm with respect to cluster convergence and partitioning.

1)(α, t) *Cluster Convergence:* As a consequence of the design tradeoffs, cluster convergence with respect to the requirements of Definition 4 is not guaranteed at every instant. Rather, each node asynchronously makes its own clustering decisions based upon the most recent information it has or can obtain directly from an adjacent node without far-reaching coordination with any other nodes. Achieving *a priori* consensus is prohibitively complex particularly because the asynchronous property of the (α, t) cluster algorithm permits any number of nodes to join a cluster simultaneously. Consequently, nodes make clustering decisions based on their ability to establish (α, t) paths in the forward direction without attempting to ensure that (α, t) paths exist in the reverse direction from all the nodes in the cluster. As a result of latency, it is therefore possible for a node to make a clustering decision without complete knowledge of the set of nodes that are in the cluster. Fig. 5 illustrates these two concepts and demonstrates how cluster convergence is achieved.

In Fig. 5(a), node m is depicted within range of node a at time t_0. Having received topology information from node a, node m has evaluated (α, t) paths to nodes a, b, c, and d, which are in a cluster together. The dashed lines in the figure represent the (α, t) paths. Node m now joins the cluster by setting its CID and disseminating its own routing information update to the existing nodes in the cluster. Note that at time t_0, no (α, t) paths have been confirmed from the existing nodes in the cluster back to node m. In Fig. 5(b), node n is shown at time t_1, just after having come into range of node d. Node d still has not established that node m has joined this cluster. Consequently, when node n evaluates (α, t) paths it does not include node m in the process. Assuming node n finds (α, t) paths to nodes a–d, it joins the cluster.

Convergence is achieved when all nodes in the cluster have a common view of cluster membership. Assuming that no more nodes join the cluster after time t_1, and that all the nodes' α timers expire in the interval $t_1 < t < t^*$ such that each node executes cluster maintenance during that interval, then Fig. 5(c) depicts the cluster state at time t^* when the cluster is guaranteed to be converged.

2)(α, t) *Cluster Partitioning:* The most basic form of cluster partitioning involves the disconnection of a single node from its cluster. As discussed in Section IV-E, when a single node recognizes that it has become cluster disconnected it leaves the cluster. The response of the remaining nodes in the cluster depends upon the precise timing of events. These nodes will either logically remove the disconnected node from their views of the cluster or depart from the cluster if the remaining nodes are no longer reachable via an (α, t) path. However, if a partitioning exists with more than one node in each partition, it is possible for the nodes in each of these partitions to decide that the nodes in the other partitions have effectively deactivated. Consequently, they will continue to

operate under the assumption that their partition is the cluster. Under these circumstances it is possible for more than one cluster to exist with the same CID. However, it is essential to prevent duplicate CID's from persisting because ambiguous CID's can lead to intercluster routing conflicts.

To resolve the problem of duplicate CID's, a renaming strategy is required. Specifically, it must be possible to detect that a cluster has partitioned and for each partition to adopt unique CID's. The strategy for achieving this is based upon embedding a globally unique node identifier (NID) into each CID. Consequently, each node in a cluster can determine the NID of the node that generated the cluster's current CID. This node will be referred to as the parent of the cluster, although it carries no other functional responsibility. Partition detection requires each node to detect when the cluster's parent node is removed or removes itself from the cluster. Renaming involves selection of a new parent and assignment of a new CID to a subset of nodes in a cluster following detection of a partition.

Cluster renaming is achieved as follows: upon detection of a cluster partition, each node determines if its NID is the lowest NID among the nodes in its cluster routing table. The node with the lowest NID generates a new CID, adopts that CID itself, and broadcasts a Cluster_Rename message, which includes the previous and the new CID's, to the nodes in its connected partition of the cluster. Each node receiving a Cluster_Rename message adopts the new CID, effectively joining a new cluster. If more than one node believes it has the lowest NID due to inconsistencies in the cluster routing tables, only the first received Cluster_Rename message will be accepted. In the worst case, the cluster partition may be subdivided into multiple new clusters. If a node does not have the lowest CID and does not receive a Cluster_Rename message within a prespecified timeout interval, the node leaves the cluster and proceeds according to the rules of node activation previously described.

V. Ad Hoc Mobility Model

In this section, a random walk-based mobility model is developed for ad hoc networks, and expressions are derived which characterize the distribution of aggregate distance and direction covered by a node over a specific interval of time. Based upon this model, and assuming that a link is active if the distance between two nodes is less than a system dependent threshold,[7] the objective is to characterize their mobility and use this characterization to determine the conditional probability that the nodes will be within range of each other at time $t_0 + t$, given that they are located within range of each other at time t_0. Assuming that link failures are independent, and the rate of node deactivation is small relative to the rate of link failure, this model shows how to evaluate path availability—providing the basis for (α, t) cluster management.

The model described in this section characterizes the aggregate behavior of nodes in a large network. In these environments, any correlation is assumed to be insignificant due to the large number of independent nodes. In addition, recent performance studies of ad hoc network routing protocols have adopted random uniform models [8], [11] or modified random models that include pause-times [4] in order to model node mobility. Pause-time random models are supported inherently by the model proposed in this paper because the speed of a mobile unit can be from any distribution, so long as it has a mean and standard deviation. Furthermore, in a large ad hoc network with many transient users, information that accurately reflects the detailed mobility characteristics of individual users is likely to be difficult or impossible to maintain. Consequently, in these types of environments the random assumption is reasonably optimal.

[7] The effective distance threshold, also referred to as the range, is a function of numerous system dependent factors including, but not limited to, signal power, fading, noise immunity, and receiver sensitivity.

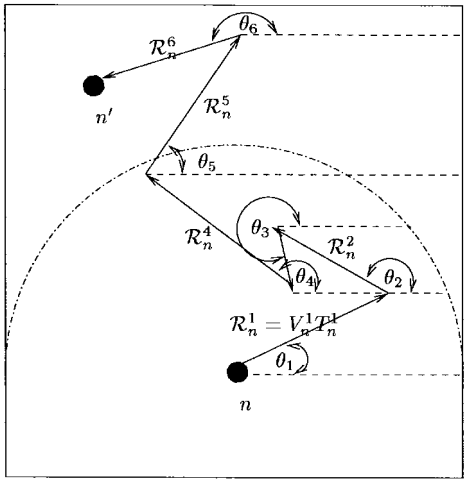
A. Random Ad Hoc Mobility

The random ad hoc mobility model proposed in this section is a continuous-time stochastic process, which characterizes the movement of nodes in a two-dimensional space. Based on the random ad hoc mobility model, each node's movement consists of a sequence of random length intervals called mobility epochs during which a node moves in a constant direction at a constant speed. The speed and direction of each node varies randomly from epoch to epoch. Consequently, during epoch i of duration T_n^i, node n moves a distance of $V_n^i T_n^i$ in a straight line at an angle of θ_n^i. The number of epochs during an interval of length t is the discrete random process $\mathcal{N}_n(t)$. Fig. 6(a) illustrates the movement of node n over six mobility epochs, each of which is characterized by its direction, θ_n^i, and distance $V_n^i T_n^i$.

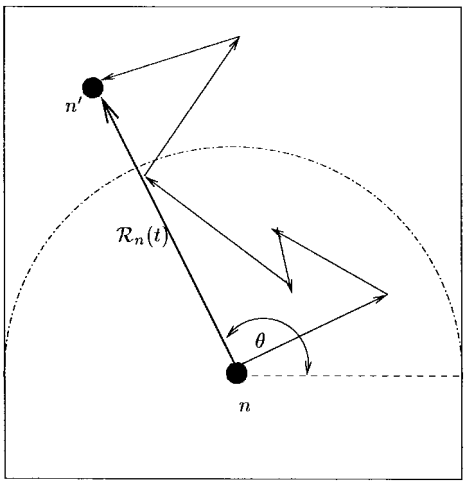
The mobility profile of node n moving according to the random ad hoc mobility model requires three parameters: λ_n, μ_n, and σ_n^2. The following list defines these parameters and states the assumptions made in developing this model.

- The epoch lengths are identically, independently distributed (i.i.d.) exponentially with mean $1/\lambda_n$.
- The direction of the mobile node during each epoch is i.i.d. uniformly distributed over (0, 2π) and remains constant only for the duration of the epoch.
- The speed during each epoch is an i.i.d. distributed random variable (e.g., i.i.d. normal, i.i.d. uniform) with mean μ_n and variance σ_n^2 and remains constant only for the duration of the epoch.
- Speed, direction, and epoch length are uncorrelated.
- Mobility is uncorrelated among the nodes of a network, and links fail independently.

Nodes with limited transmission range are assumed to experience frequent random changes in speed and direction with respect to the length of time a link remains active between two nodes. Furthermore, it is assumed that the distributions of each node's mobility characteristics change slowly relative to the rate of link failure. Consequently, the distribution of the number of mobility epochs is stationary and relatively large while a link is active. Since the epoch lengths are i.i.d. exponentially distributed, $\mathcal{N}_n(t)$ is a Poisson process with rate λ_n. Hence, the expected number of epochs experienced by



(a)



(b)

Fig. 6. Ad hoc mobility model node movement: (a) epoch random mobility vectors (b) and hoc mobility model node movement.

node n during the interval $(0, t)$ while a link is active is $\lambda_n t \gg 1$.

These assumptions reflect a network environment in which there are a large number of heterogeneous nodes operating autonomously in an ad hoc fashion, which conceptually reflects the environment considered in the design of the (α, t) cluster framework. That is, while some nodes may share similar objectives and move together, there is a large enough population of nodes and frequency of events[8] that the overall correlation is insignificant and the aggregate effective movement can be modeled by a random process.

In order to characterize the availability of a link between two nodes over a period of time $(t_0, t_0 + t)$, the distribution of the mobility of one node with respect to the other must be determined. To characterize this distribution, it is first necessary to derive the mobility distribution of a single node in isolation. The single node distribution is extended to derive the joint mobility distribution that accounts for the mobility of one node with respect to the other. Using this joint mobility distribution, the link availability distribution is derived. If the link availability metric is known for each link along a path between two mobile nodes, assuming that links fail independently, the path availability is easily determined as the product of the individual link availability metrics.

Single Node Mobility: Two definitions are central to the development of the single node mobility model. They define two random vectors that characterize the direction and distance moved by a mobile node during a single epoch and over an interval of length t, respectively.

Definition 5: The epoch random mobility vector $\vec{\mathcal{R}}_n^i$ represents the direction and distance moved by node n during mobility epoch i. It has magnitude $\mathcal{R}_n^i = |\vec{\mathcal{R}}_n^i| = V_n^i T_n^i$ which is the distance covered by node n during epoch i, and phase θ_n^i, which is the direction of node n during epoch i.

Definition 6: $\vec{\mathcal{R}}_n(t)$ is the random mobility vector for node n. The magnitude $\mathcal{R}_n(t)$ is equal to the distance from $(X(t_0), Y(t_0))$ to $(X(t_0{+}t), Y(t_0{+}t))$, where $(X(\tau), Y(\tau))$ is the position of the node at time τ. The phase angle θ_n is the angle of the line joining the two points. The random mobility vector can be expressed as a random sum of the epoch random mobility vectors $\vec{\mathcal{R}}_n(t) = \sum_1^{\mathcal{N}_n(t)} \vec{\mathcal{R}}_n^i$.

Fig. 6(a) shows a sample path for the movement of an arbitrary node n over an interval of length t. For each epoch, the figure shows the epoch vector $\vec{\mathcal{R}}_n^i$ with magnitude $V_n^i T_n^i$ and direction θ_n^i of the node during the epoch. The resulting random mobility vector $\vec{\mathcal{R}}_n(t)$ is shown in Fig. 6(b), and it can be seen that it is the vector sum of the individual epoch vectors. The following lemma characterizes the magnitude and phase distributions of $\vec{\mathcal{R}}_n(t)$.

Lemma 1: Consider a mobile node which is located at position $(X(t_0), Y(t_0))$ at time t_0 and moves according to a random ad hoc mobility profile, $\langle \lambda_n, \mu_n, \sigma_n^2 \rangle$. Let $\vec{\mathcal{R}}_n(t)$ be the resulting random mobility vector. The phase angle θ_n of $\vec{\mathcal{R}}_n(t)$ represents the aggregate direction of the mobile node and is uniformly distributed over (0, 2π), and the magnitude $\mathcal{R}_n(t)$ represents the aggregate distance moved by the node and is approximately Raleigh distributed with parameter $\alpha_n = (2t/\lambda_n)(\sigma_n^2 + \mu_n^2)$

$$\Pr(\theta_n \le \phi) = \frac{1}{2\pi}\,\phi, \qquad 0 \le \phi \le 2\pi \tag{5}$$

$$\Pr(\mathcal{R}_n(t) \le r) \approx 1 - \exp\left(\frac{-r^2}{\alpha_n}\right), \qquad 0 \le r \le \infty. \tag{6}$$

The derivation of these distributions is an application of the theory of uniform random phasor sums [3]. The basic idea is to decompose the distance moved during each mobility epoch, $\mathcal{R}_n^i = V_n^i * T_n^i$, into X and Y components, $X_n^i = \mathcal{R}_n^i \cos\theta_i$ and $Y_n^i = \mathcal{R}_n^i \sin\theta_i$. According to the random ad hoc mobility model, $\lambda_n t \gg 1$. For the Poisson distribution, this is a sufficient condition for the central limit theorem (CLT) to hold with respect to the summations of the X and Y components over all the epochs during an interval of length t. Consequently, $X_n(t) = \sum_1^{\mathcal{N}_n(t)} X_n^i$ and $Y_n(t) = \sum_1^{\mathcal{N}_n(t)} Y_n^i$ are approximately normally distributed and are shown to have zero mean and variance $= (t/\lambda_n) * (\sigma_n^2 + \mu_n^2)$. Furthermore, $X_n(t)$ and $Y_n(t)$ are shown to be uncorrelated; therefore, the product of the two normal distributions gives the joint

[8] Events include the activation of a link or node and changes in speed and direction of a node.

distribution of X and Y. This is transformed using standard methods into polar coordinates to produce the joint distribution with respect to $\mathcal{R}_n(t)$ and θ_n. The results of the lemma follow by taking the marginal distributions with respect to these random variables. Simulation results reported in [22] validate these analytical results.

The results of this section show that if a mobile node moves in a random uniform direction during each mobility epoch, the random nature of the mobile's direction is preserved over several direction and speed changes. Along with the distribution of the aggregate distance, this allows for the characterization of the joint mobility of two mobile nodes by considering the relative movement of one node with respect to the other.

Joint Node Mobility: In cellular networks the characterization of mobility metrics[9] relies on the analysis of the movement of a single node with respect to a fixed point of reference [14], [32]. Based on the assumption of random link failures, the ad hoc problem can be transformed into the cellular problem by considering the mobility of two nodes at a time and fixing the frame of reference of one node with respect to the other. This transformation is accomplished by treating one of the nodes as if it were the base station of a cell, keeping it at a fixed position. For each movement of this node, the other node is translated an equal distance in the opposite direction. These concepts are reflected in the following definitions and lemmas:

Definition 7: The vector $\vec{\mathcal{R}}_{m,n}(t)$, representing the equivalent random mobility vector of node m with respect to node n, is obtained by fixing m's frame of reference to n's position and moving m relative to that point.

Lemma 2: Let two mobile nodes m and n move according to random ad hoc mobility profiles, $\langle \lambda_m, \mu_m, \sigma_m^2 \rangle$ and $\langle \lambda_n, \mu_n, \sigma_n^2 \rangle$, respectively. By Lemma 1, the random mobility vectors for each node are $\vec{\mathcal{R}}_m(t)$ and $\vec{\mathcal{R}}_n(t)$ with uniformly distributed direction and Raleigh distributed magnitude. Let α_m and α_n be the parameters of the Raleigh distributions. $\mathcal{R}_{m,n}(t)$ is the magnitude of the difference $\vec{\mathcal{R}}_m(t) - \vec{\mathcal{R}}_n(t)$, is Raleigh distributed with parameter $\alpha_{m,n} = \alpha_m + \alpha_n$, and the phase is uniformly distributed over $(0, 2\pi)$.

The X and Y components of the two uniformly distributed Raleigh phasors $\vec{\mathcal{R}}_m(t)$ and $\vec{\mathcal{R}}_n(t)$ are each approximately normal with zero mean and variance $= (t/\lambda_m) * (\sigma_m^2 + \mu_m^2)$ and $(t/\lambda_n) * (\sigma_n^2 + \mu_n^2)$, respectively. Since the two nodes move independently according to the random ad hoc model, the distributions of $X_{m,n}(t) = X_m(t) - X_n(t)$ and $Y_{m,n}(t) = Y_m(t) - Y_n(t)$ are also normal with zero mean and variance $= (t/\lambda_m) * (\sigma_m^2 + \mu_m^2) + (t/\lambda_n) * (\sigma_n^2 + \mu_n^2)$. The result follows by taking the joint distribution of $X_{m,n}(t)$ and $Y_{m,n}(t)$, transforming into polar coordinates, and taking the marginal distributions.

Lemma 3: $\vec{\mathcal{R}}_{m,n}(t) = \vec{\mathcal{R}}_m(t) - \vec{\mathcal{R}}_n(t)$ is the equivalent random mobility vector of node m with respect to node n.

Corollary 1: By Lemmas 2 and 3, the equivalent random mobility vector node m with respect to node n is approximately Raleigh distributed and has a uniformly distributed direction.

[9] Residence time and handoff rate are examples of mobility metrics.

B. Random Ad Hoc Link Availability

In this section, the distributions for mobile node distance and direction are used to derive expressions for link availability based upon different initial conditions. Assuming that link failures are independent and the rate of node deactivation is small relative to the rate of link failure, it is shown how the expression for link availability can be used to derive the path availability metric.

Corollary 1 shows how the joint mobility problem can be transformed into an equivalent problem involving the movement of a single node. In this section, the result of Corollary 1, along with the distribution of the distance covered by a single node as it moves across a cell prior to a handoff [14], is used to derive the distribution of the availability of a link between two nodes.

Definition 8: Let $\mathcal{L}_{m,n}(t)$ indicate the state of the link between nodes n and m at time t. $\mathcal{L}_{m,n}(t) = 1$ if the link is active, $\mathcal{L}_{m,n}(t) = 0$ if the link is inactive.

Definition 9: Link availability is the probability that there is an active link between two mobile nodes at time $t_0 + t$, given that there is an active link between them at time t_0. Note that a link is still considered available at time t even if it experienced failures during one or more intervals (t_i, t_j); $t_0 < t_i < t_j < t_0 + t$. More specifically, for nodes n and m, link availability is defined as

$$\mathcal{A}_{m,n}(t) \equiv \Pr(\mathcal{L}_{m,n}(t_0 + t) = 1 | \mathcal{L}_{m,n}(t_0) = 1).$$

Fig. 7 demonstrates the mobility of two nodes initially separated by a distance C. The transformation from single node random mobility vectors to the equivalent random mobility vector can be derived by noting how the progression of the distance between the nodes proceeds in an identical manner in both cases. The movement of m relative to n is shown for each epoch, along with the resulting equivalent random mobility vector $\vec{\mathcal{R}}_{m,n}(t)$. If m lies within the circular region of radius R_{eq}[10] centered at n, the link between the two nodes is considered to be active.

Depending on the initial status and location of nodes m and n, two distinct cases of link availability can be identified. In the first case, the link activation is caused by the activation of an adjacent node at time t_0. In the second case, the link activation is caused by node mobility, specifically, the nodes move into range of each other at time t_0. Assuming node n is active at time t_0, the link availability models reflect the following initial conditions.

1) Node activation: node m becomes active at time t_0, and it is assumed to be at a random location within range of node n.
2) Link activation: node m moves within range of node n at time t_0 by reaching the boundary defined by R_{eq}, and it is assumed to be located at a random point around the boundary.

[10] The transmission range of a mobile node is assumed to be bounded by an area with a hexagonal shape of radius R. $R_{\text{eq}} \approx 0.91R$ is the radius of the approximating circle with the same area [14].

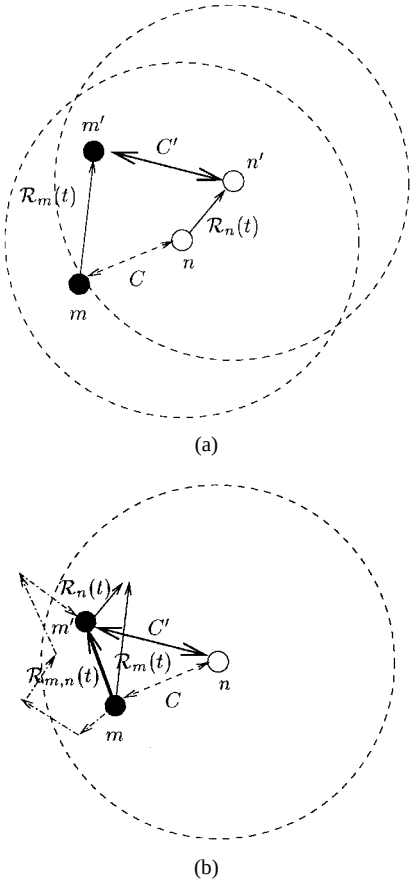


Fig. 7. Joint mobility transformation. (a) Joint node case. (b) Joint mobility transformation.

Theorems 1 and 2 characterize the link availability between two mobile nodes, n and m, as reflected by the initial conditions stated above. Proofs are presented in the Appendix, and simulation results reported in [22] demonstrate excellent agreement with these analytical models.

Theorem 1—Node Activation: If node n moves according to a random ad hoc mobility profile $\langle \lambda_n, \mu_n, \sigma_n^2 \rangle$, and node m activates at time t_0 within a uniform random distance from node n and moves according to a random ad hoc mobility profile $\langle \lambda_m, \mu_m, \sigma_m^2 \rangle$, then the distribution of the link availability over time is given approximately by the following expression, where $\Phi(a, b, z)$ is the Kummer-confluent hypergeometric function[11]:

$$\mathcal{A}_{m,n}(t) \approx 1 - \Phi\left(\frac{1}{2}, 2, \frac{-4R_{\text{eq}}^2}{\alpha_{m,n}}\right) \tag{7}$$

$$\alpha_{m,n} = 2t\left(\frac{\sigma_m^2 + \mu_m^2}{\lambda_m} + \frac{\sigma_n^2 + \mu_n^2}{\lambda_n}\right). \tag{8}$$

Theorem 2—Link Activation: Let $\langle \lambda_n, \mu_n, \sigma_n^2 \rangle$ and $\langle \lambda_m, \mu_m, \sigma_m^2 \rangle$ be the random ad hoc mobility profiles of node n and node m, respectively, and assume that and a link activates between n and m at time t_0 such that m is located at a uniform random point exactly R_{eq} from n; then, the link availability is distributed according to the following expression, where I_0 is a modified Bessel function of the first kind, and $\alpha_{m,n}$ is defined in (8):

$$\mathcal{A}_{m,n}(t) = \frac{1}{2}\left(1 - I_0\left(\frac{-2R_{\text{eq}}^2}{\alpha_{m,n}}\right)\exp\left(\frac{-2R_{\text{eq}}^2}{\alpha_{m,n}}\right)\right). \tag{9}$$

[11] In (7), $a = 1/2$ and $b = 2$, consequently, the expression reduces to $1 - e^{z/2}(I_0(z/2) - I_1(z/2))$.

C. Random Ad Hoc Path Availability

Lemma 4 completes the model developed in this section by relating path availability to individual link availabilities according to the definition of path availability given in Definition 1, and the assumption of independent link failures.

Lemma 4: Let $\mathcal{A}_{i,j}(t)$ be the availability for link $(i, j) \in$ path k between nodes n and m, as defined in Definition 9. The path availability at time t_0+t is denoted $\pi_{m,n}^k(t)$. According to the assumption of independent link failures, path availability is given by

$$\begin{aligned}\pi_{m,n}^k(t) &\equiv \Pr(\mathcal{P}_{m,n}^k(t_0+t) = 1 | \mathcal{P}_{m,n}^k(t_0) = 1) \\ &= \prod_{(i,j)\in k} \mathcal{A}_{i,j}(t_0+t).\end{aligned} \tag{10}$$

Path Availability Cost Calculation: Theorems 1 and 2 demonstrate how the link availability can be calculated, thereby providing a link metric that represents a probabilistic measure of path availability. This metric can be used by the routing algorithm in order to construct paths that support a lower bound α on availability of a path over an interval of length t as specified in Definition 2. Based on Lemma 4 and Definition 2, the availabilities of each of the links along a path are used by the (α, t) cluster protocol to determine if the path is an (α, t) path, and consequently, if a cluster satisfies the (α, t) criteria. In order to support this functionality in an ad hoc network, the routing protocol must maintain and disseminate the following status information for each link:

- the initial link activation time: t_0;
- the mobility profiles for each of the adjacent nodes: $\langle \lambda_i, \mu_i, \sigma_i^2 \rangle$, $i = m, n$;
- the transmission range of each of the adjacent nodes: R_{eq};
- the event which activated the link: 1) node activation at time t_0 or 2) nodes moving into range of each other at time t_0.

Based on this information, any node in an (α, t) cluster can estimate, at any time τ, the availability of a link at time $t+\tau$. This can be achieved because each node knows the initial link activation time t_0; hence, link availability is evaluated over the interval $(t_0, t+\tau)$. Nodes can use conditional probability to evaluate the availability of their own links because they have direct knowledge of such a link's status at time τ, whereas remote nodes do not. Specifically, for an incident link that activated at time t_0, a node will evaluate the availability at time t, given that it is available at time $\tau \geq t_0$.

In this section, the random ad hoc mobility model was developed and Theorems 1 and 2 show how this model can be used to quantify the probability that a link will be available between two nodes after an interval of duration t. Lemma 4 shows how this model can be extended to completely characterize the availability of multihop paths across an ad hoc network depending upon the initial status of each link

in the path and assuming independent failures of each link. Although it was not considered directly in these models, the extension to include node failure or deactivation can be made by considering the link failure probabilities conditioned on the status of the nodes. The total probability of link failure will then consist of the weighted contribution due to mobility and to the failure of at least one of the nodes.

VI. Simulation

The performance of the (α, t) cluster strategy can be assessed according to a variety of measures that broadly fall into two distinct categories, namely: 1) those that capture the dynamic properties of the (α, t) cluster protocol with respect to cluster stability and protocol efficiency and 2) those which characterize the packet level performance, such as delay and throughput. The dynamic packet level performance depends substantially upon the properties of the underlying routing protocols and medium access control schemes, whereas the inherent stability and efficiency of the (α, t) cluster protocol can be evaluated by considering the following objectives: the (α, t) cluster protocol should: 1) adapt to node mobility by dynamically changing the cluster size according to the (α, t) criteria; 2) provide an effective infrastructure that is more stable than the unclustered network; and 3) achieve cluster maintenance with minimal communications overhead.

Based on the routing methodology discussed in Section III-B, the (α, t) cluster strategy reduces to a flat-routed, reactive strategy when node mobility is very high on a persistent basis. The dynamic traffic performance in this worst case scenario is characterized by the performance of the intercluster routing protocol that has previously been reported in the literature [11]. However, network dynamics will not always be this severe. Furthermore, it is reasonable to expect that communications among nodes that are physically close together will be typical in many ad hoc network applications. Consequently, the probability of communications among nodes in the same or nearby clusters is expected to be high. As demonstrated by the simulation results reported in this section, the control message overhead required to achieve clustering is insignificant even at very high link failure rates. Therefore, the clustering overhead is expected to have little effect upon delay and throughput characteristics. Thus, the (α, t) cluster strategy will be able to provide improved traffic-level performance relative to a reactive routing strategy—without requiring significant control overhead.

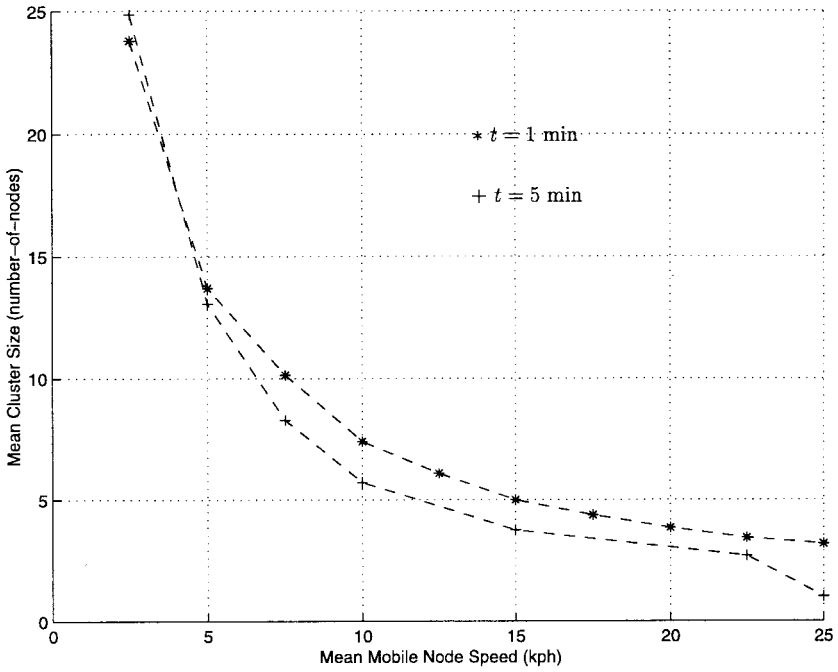
Based upon the previous observations, a simulation model was developed to evaluate the inherent stability and efficiency of the (α, t) cluster protocol. Specifically, the simulation was used to measure the strategy's effectiveness in terms of mean performance metrics including the mean cluster size, the probability of a node being clustered, the mean node residence time within a given cluster, the mean cluster survival time, and the per-node control message processing rate. The remainder of this section discusses the simulation model and presents analysis of the results.

The simulation was developed to model an ad hoc network in which nodes activate and deactivate according to exponential distributions. Once active, each node moved according to the mobility model presented in Section V of this paper. A range of node mobility with mean speeds between 5.0–25.0 km/h was simulated. The speeds during each mobility epoch were normally distributed, and the direction was uniformly distributed over $(0, 2\pi)$ [11]. Each node changed its speed and direction at random times. Although the simulation and the analytical models for link availability support distributions which include random pause times [4], this performance evaluation assumed nodes to be in constant motion. Thus, extreme node mobility was used to produce a maximally dynamic environment. Link activations and failures were detected through a process running on each node that modeled a periodic link-sensing function. This was achieved by adjusting the new positions of all the nodes in the simulation according to their current position, speed, and direction and by checking the distance between the sensing node and all other nodes.

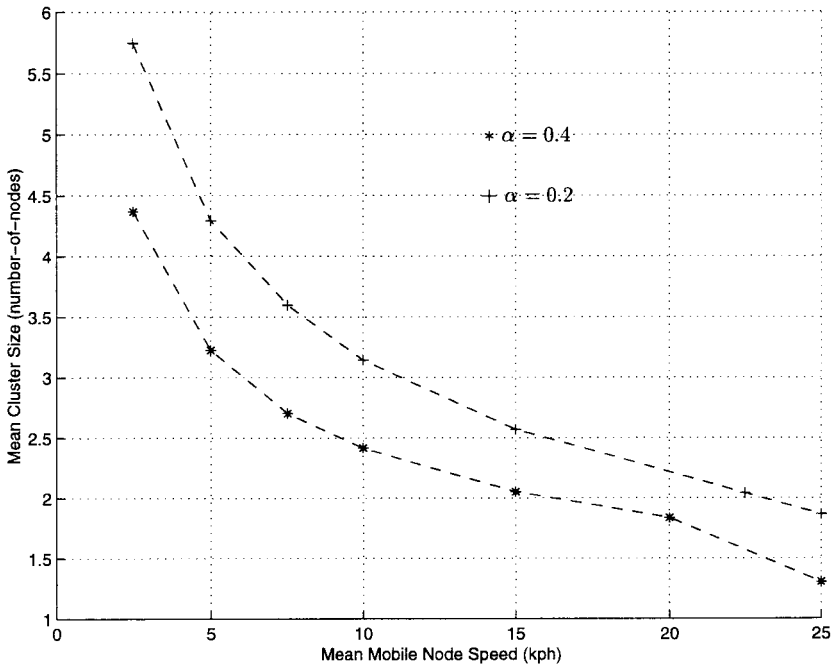
The results reported in this section were based upon a node activation rate of 250 nodes/hr. The mean time to node deactivation was 1 h. Using an approach similar to [11], nodes were initially randomly activated within a bounded region of 5×5 km. Nodes that moved beyond this boundary were no longer considered to be part of the ad hoc network and were effectively deactivated. Each node's actions within the boundary were determined according to the (α, t) cluster algorithm described in Section IV. An ideal link-state protocol was assumed for the distribution of topology information within each cluster; topology updates were sent to every node in the cluster following any link activation or failure detected by a clustered node according to the requirements of Section V-C. Link availability was estimated for the entire cluster topology by each node, following link failures or the expiration of the node's α timer according to the methodology presented in Section V-C. Finally, (α, t) path availability was evaluated using Dijkstra's algorithm.

For each simulation run, data was collected by sampling the network status once per second over an observation interval of 1 h. The first 2 h of each run were discarded to eliminate transient effects, and each simulation was rerun ten times with new random seeds. Results are shown for two cases of transmission range, namely, $R_{\text{eq}} = 1.0$ km and $R_{\text{eq}} = 0.5$ km. For the case of $R_{\text{eq}} = 1.0$ km, results are shown for $\alpha = 0.4$ using two values of t, 1 min and 5 min. For the case of $R_{\text{eq}} = 0.5$ km, results are shown for $t = 1$ minusing two values of α, 0.4 and 0.2, respectively. These values, although not comprehensive, demonstrate a wide range of possible values for the system parameters. Furthermore, the node mobility model is intended to demonstrate the behavior of clustering under the worst-case scenario, as typified by the totally random movement of nodes over a wide range of speeds. Subject to these harsh conditions, it is physically impossible to achieve significantly high probabilities of path availability. Consequently, relatively low values were used for α in the simulations. While this limits the path availability bound that can be guaranteed, the simulation results show that it does not effect cluster stability.

Fig. 8(a) and (b) shows the effects of mobility on mean cluster size. The results show the adaptive property of the



(a)

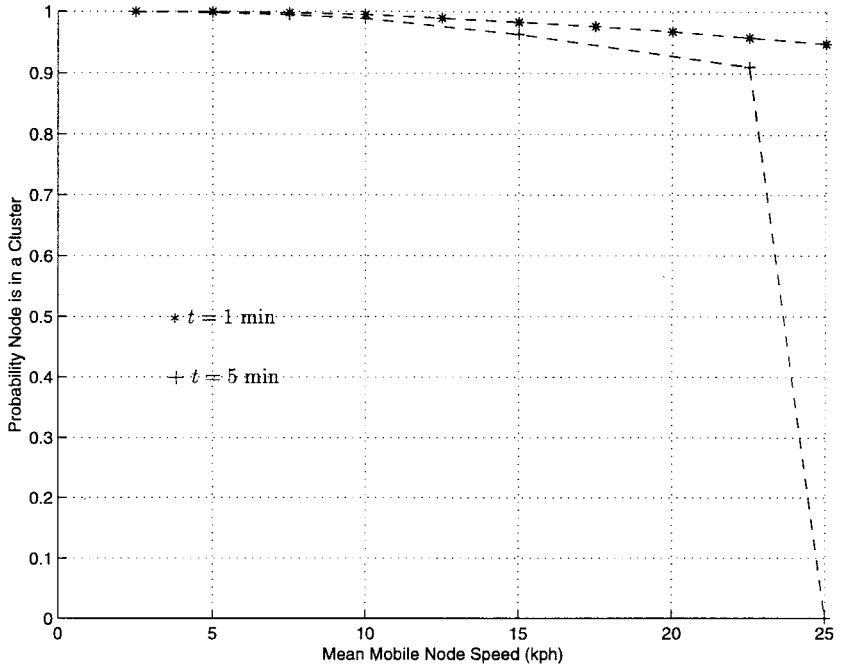


(b)

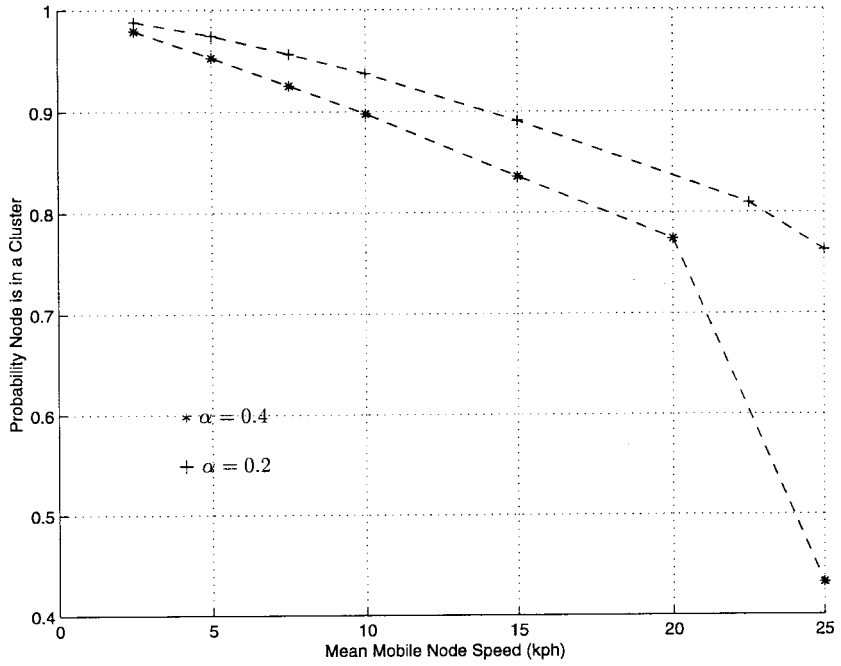
Fig. 8. Simulation results. (a) Cluster size ($R_{\mathrm{eq}} = 1000$ m). (b) Cluster size ($R_{\mathrm{eq}} = 500$ m).

(α, t) cluster algorithm and also the significant effect due to the nodes' effective transmission range. It is worthwhile to point out that it is unlikely that very low range transmitters could effectively be used in ad hoc networks with nodes moving at high rates of speed unless the nodes are moving together. These results demonstrate a desirable feature of the (α, t) cluster protocol, namely, that it adapts cluster size to node mobility. Specifically, it maintains larger clusters under lower mobility to benefit from more optimal routing, while reducing cluster size in response to greater mobility.

Fig. 8(c) and (d) shows the effects of mobility on the probability that a node is clustered. The results demonstrate a desirable property of the (α, t) cluster protocol, namely, that nodes still remain clustered with high probability even at high rates of mobility. It is interesting to observe the effects of the system parameters on this metric. Specifically, for the case of $R_{\mathrm{eq}} = 1$ km and the value of $t = 1$ min, nodes remain clustered with probability ≥0.90 even at the highest mobility rates. However, if the (α, t) criteria demands (α, t) paths over a 5 min interval, then the node's ability to achieve clustering



(c)



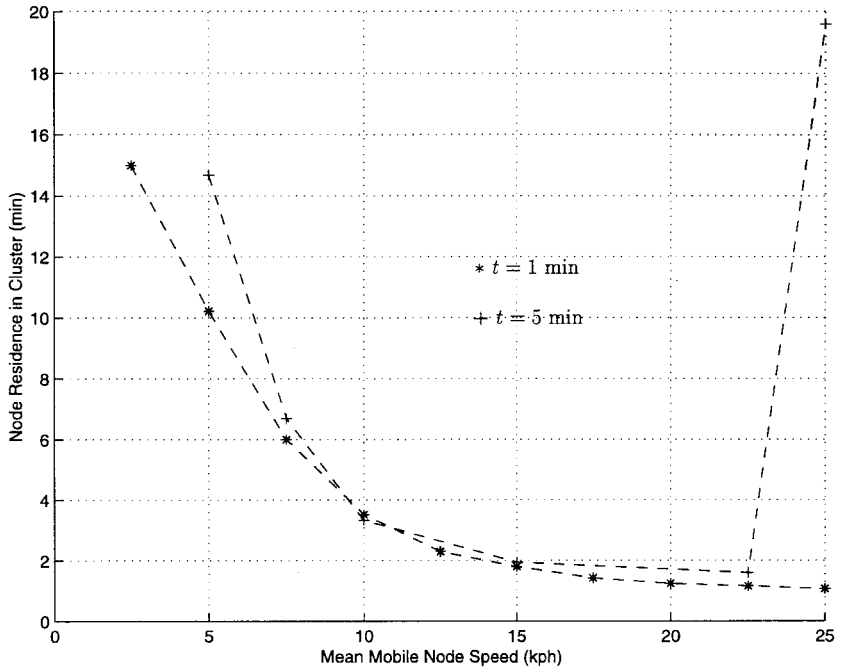
(d)

Fig. 8. *(Continued.)* Simulation results. (c) Cluster probability ($R_{\mathrm{eq}} = 500$ m). (d) Cluster probability ($R_{\mathrm{eq}} = 1000$ m).

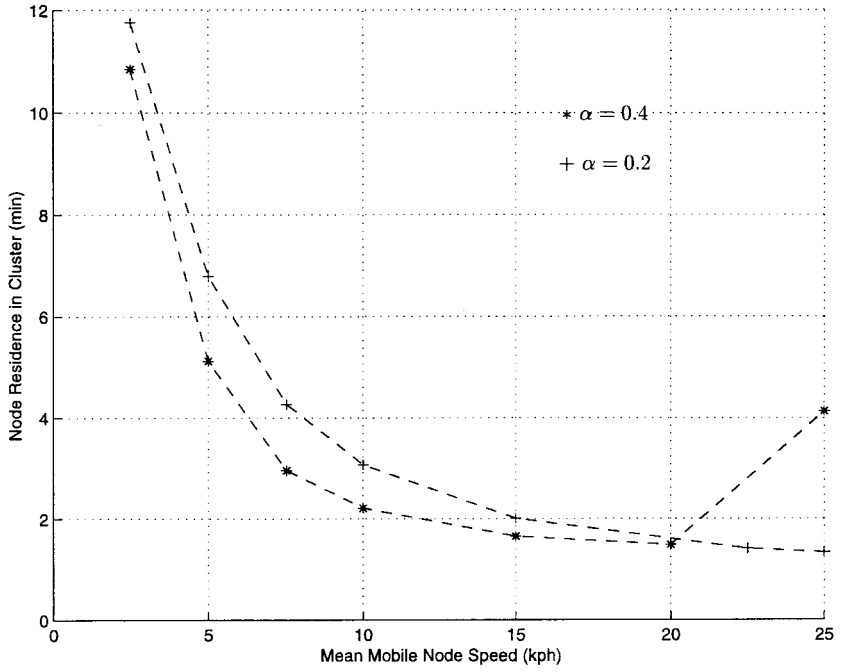
collapses above 20 km/h. For the case of $R_{\mathrm{eq}} = 0.5$ km, a similar effect is observed for variations in α. Lower values of α permit nodes to cluster more easily, although referring to Fig. 8(b) shows that the clusters are significantly smaller.

Fig. 8(e)–(h) demonstrates additional stability properties of the (α, t) cluster. Residence time is defined as the time a node remains resident in a given cluster. Longer residence times are desirable, although taken in conjunction with the probability of being clustered, smaller residence times can still be acceptable in terms of system performance. This is true because the overhead of clustering in the (α, t) cluster strategy is minimal. The enormous jump in residence time at 25 km/h observed in Fig. 8(e) is due to the very low probability of a node actually being clustered at that rate; therefore, the number represents a very small portion of the nodes in the network. Had the simulation included pause times, it is likely that cluster residence times would increase substantially.

Cluster survival time was measured by taking the amount of elapsed time each currently active cluster had existed at each sampling instant. Thus, it represents a measure of cluster



(e)

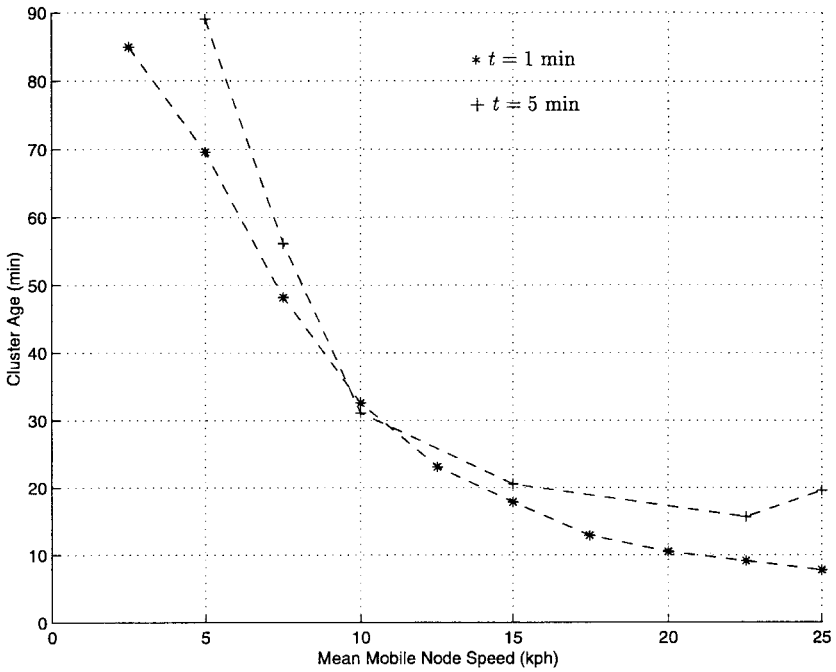


(f)

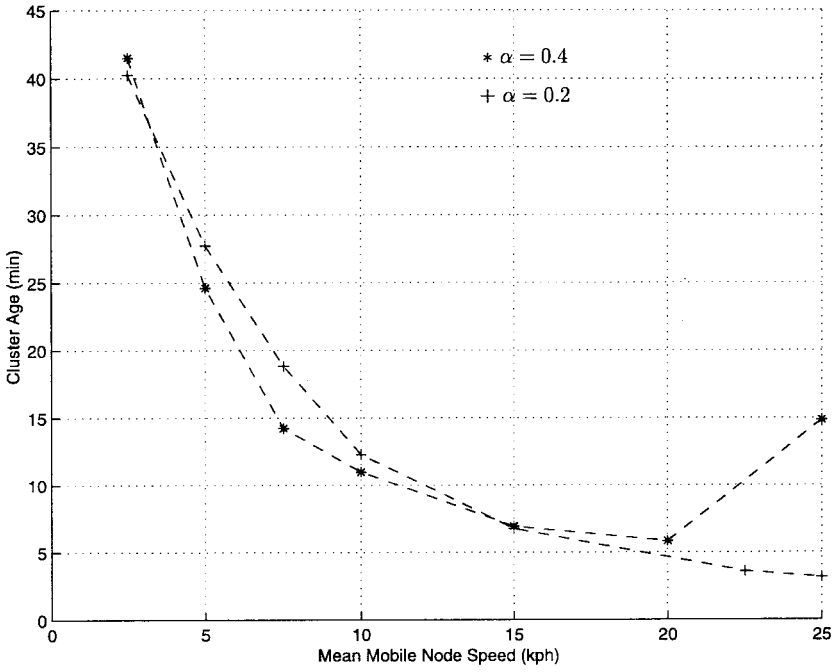
Fig. 8. *(Continued.)* Simulation results. (e) Residence time ($R_{\mathrm{eq}} = 500$ m). (f) Residence time ($R_{\mathrm{eq}} = 500$ m).

lifetime. A stable cluster topology should have relatively long cluster lifetimes. The link failure rates that were observed in these simulations range from less than 1 failure/s at the lower mobility to upwards of 2–3 failures/s at the higher rates. Similar rates were observed for link activations. Given the high rates of link failure that were observed, the cluster survival times shown in Fig. 8(g) and (h) are reasonable.

Finally, the rate of control messages processed per node, as depicted in Fig. 8(i) for $R_{\mathrm{eq}} = 1$ km (similar results were seen for $R_{\mathrm{eq}} = 0.5$ km), provides a measure of the efficiency of the (α, t) cluster algorithm. This was measured by counting the number of routing updates, including those required to join and leave clusters, that were processed by each node every second. It is interesting to observe that the algorithm essentially protects the nodes from the effects of topology changes as node mobility increases. The shape of the curve can be explained as follows: the initial increase in the received message rate is due to the substantial increase in topology changes. Although the cluster size is diminished, it is more than compensated for by the increased rate of



(g)



(h)

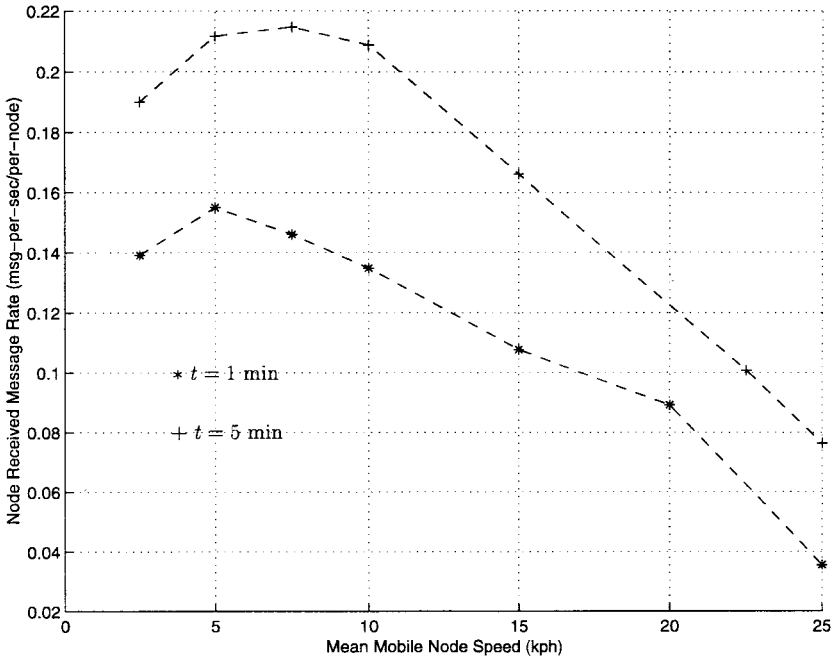
Fig. 8. *(Continued.)* Simulation results. (g) Cluster survival ($R_{\mathrm{eq}} = 1000$ m). (h) Cluster survival ($R_{\mathrm{eq}} = 500$ m).

node clustering activity and topology changes. Finally, the algorithm's adaptive property drives the message rate down by reducing the cluster size significantly as mobility increases beyond 10 km/h.

In this section, a simulation model was developed in order to show the effectiveness of the (α, t) cluster strategy in terms of its inherent properties, namely, adaptiveness to node mobility, cluster stability, and protocol efficiency. The results demonstrate that the scheme performs well and is well adapted to meet its stated objectives in the environments for which it has been designed to operate.

VII. Conclusions

The (α, t) cluster framework defines a strategy for adaptively organizing ad hoc networks into clusters in which the probability of path availability between nodes is bounded over time. The purpose of this dynamic arrangement is to support an adaptive hybrid approach to routing that is efficient under all conditions and yet can achieve more optimal routing when mobility patterns favor it. The concept of the (α, t) path was introduced and analytical models were developed that show how these paths can be evaluated. The (α, t)



(i)

Fig. 8. *(Continued).* Simulation results. (i) Message rate ($R_{\rm eq} = 1000$ m).

criteria was defined that specifies the conditions required for the management of (α, t) clusters. Finally, an algorithm was described that efficiently maintains the cluster topology with very little additional processing or internodal coordination. Simulation results show the inherent adaptive properties and stability of the (α, t) cluster protocol. Based upon the proposed routing methodology it was argued that existing reactive routing strategies provide a lower bound on the traffic-level performance of the (α, t) cluster strategy and that in most cases the performance will be improved. Future work includes detailed analysis of traffic-level performance and adaptation of admission and connection control algorithms to support probabilistic QoS guarantees using the (α, t) cluster framework.

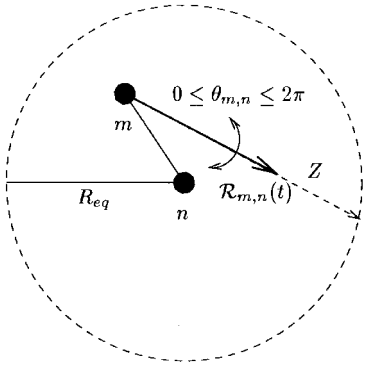


Fig. 9. Node activation model.

APPENDIX

This appendix presents proofs of Theorems 1 and 2 that characterize the link availability between two mobile nodes. Refer to Section V-B for statements of the theorems.

Proof of Theorem 1: The analysis in [14] presents the derivation of the distance Z that a mobile node must travel before reaching the boundary of a cell when the mobile moves in a random uniform direction over (0, 2π). If the cell has an effective radius of $R_{\rm eq}$, and the mobile is initially located anywhere within the cell with equal probability, then the pdf of the distribution of this distance is

$$f_Z(z) = \frac{2}{\pi R_{\rm eq}^2}\sqrt{R_{\rm eq}^2 - (z/2)^2}, \qquad 0 \le z \le 2R_{\rm eq}.$$

According to the analysis in Section V-A, the joint mobility problem can be transformed into a single node mobility problem. Assume that node n is located at the center of a circular region of radius $R_{\rm eq}$ and that node m is located within a uniform random distance $0 \le C \le R_{\rm eq}$ along a uniform random trajectory angle over (0, 2π). The equivalent direction of m is uniform over (0, 2π), and the distance $\mathcal{R}_{m,n}(t)$ moved in time t is approximately Raleigh distributed. In the node activation case, m is assumed to activate at time t_0 anywhere within a distance of $R_{\rm eq}$ from n with equal probability. Fig. 9 illustrates the relationship among these variables. Consequently, the probability that m is still within a distance of $R_{\rm eq}$ at time t is the probability that the equivalent distance it travels in that time is less than the distance to the boundary of the approximating circle given by the distribution of Z. This probability is equivalent to the link availability as expressed in Definition 9

$$\mathcal{A}_{m,n}(t) \equiv \Pr(\mathcal{R}_{m,n}(t) < Z). \tag{11}$$

This probability can be evaluated using the result of Corollary 1 and the distribution of Z by conditioning on $Z = z$. The integral is evaluated by expanding the coefficients of the exponential and integrating term-by-term. In what follows, let

$a = 1/2, b = 2, z = (-4R_{\text{eq}}^2)/(\alpha_{m,n})$ and $(a)_k$, $(b)_k$ are Pochhammer symbols: $(a)_k = a(a+1)(a+2)\cdots(a+k-1)$. $\alpha_{m,n}$ is defined in (8)

$$\begin{aligned}
&\Pr(\mathcal{R}_{m,n}(t) < Z) \\
&\quad = \int_{-\infty}^{\infty} \Pr(\mathcal{R}_{m,n}(t) < Z | Z = z) f_Z(z)\, dz \\
&\quad = \int_{-\infty}^{\infty} \Pr(\mathcal{R}_{m,n}(t) < z) f_Z(z)\, dz \\
&\quad = \int_0^{2R_{\text{eq}}} \left(1 - \exp\left(\frac{-z^2}{\alpha_{m,n}}\right)\right) \frac{2}{\pi R_{\text{eq}}^2} \\
&\qquad \cdot \sqrt{R_{\text{eq}}^2 - (z/2)^2}\, dz \\
&\quad = 1 - \int_0^{2R_{\text{eq}}} \frac{2}{\pi R_{\text{eq}}^2} \sqrt{R_{\text{eq}}^2 - (z/2)^2} \\
&\qquad \cdot \left(\sum_{i=0}^{\infty} (-1)^i \frac{z^{2i}}{i! \alpha_{m,n}^i}\right) dz \\
&\quad = 1 - \left(1 - \frac{R_{\text{eq}}^2}{\alpha_{m,n}} + \frac{R_{\text{eq}}^4}{\alpha_{m,n}^2} - \frac{5R_{\text{eq}}^6}{6\alpha_{m,n}^3} + \cdots\right) \\
&\quad = 1 - \left(1 + \frac{a}{b}\, z + \frac{a(a+1)z^2}{b(b+1)2!} + \sum_{k=3}^{\infty} \frac{(a)_k z^k}{(b)_k k!}\right). \quad (12)
\end{aligned}$$

The expression in (12) is the hypergeometric series, which is the series expansion for the confluent hypergeometric function $\Phi(a, b, z)$. **QED**

Proof of Theorem 2: The analysis in [14] presents the derivation of the distance Z that a mobile node entering a cell must travel before reaching the boundary of the cell when the mobile moves in a random uniform direction. Reflecting the assumption that the mobile is entering the cell, the direction of Z is random uniformly distributed over (0,π). The value for Z along any other trajectory must be zero since the mobile would never enter the cell. Consequently, the pdf of the distribution of Z is conditional with respect to $0 \le \theta_{m,n} \le \pi$ and is given by

$$f_{Z|0\le\theta_{m,n}\le\pi}(z) = \frac{1}{\pi} \frac{1}{\sqrt{R_{\text{eq}}^2 - (z/2)^2}}, \qquad 0 \le z \le 2R_{\text{eq}}.$$

In transforming the joint node mobility problem into a single-node fixed-reference mobility problem, node m moves in an equivalent direction that is uniform over (0, 2π), and node n remains in a fixed position. Over an interval of length t, the distance $\mathcal{R}_{m,n}(t)$ moved by node m relative to node n is approximately Raleigh distributed. Fig. 10 illustrates the relationship among R_{eq}, Z, $\mathcal{R}_{m,n}(t)$, and $\theta_{m,n}$.

Proceeding in the same manner as in the derivation of Theorem 1, the link availability is determined according to (11). However, the direction of node m is uniform over (0, 2π), whereas the direction of Z is uniform over (0, π). Consequently, conditional probability must be used to solve the link availability problem, as follows:

$$\begin{aligned}
&\Pr(\mathcal{R}_{m,n}(t) < Z) \\
&\quad = \Pr(\mathcal{R}_{m,n}(t) < Z | 0 \le \theta_{m,n} \le \pi) \Pr(0 \le \theta_{m,n} \le \pi) \\
&\qquad + \Pr(\mathcal{R}_{m,n}(t) < Z | \pi \le \theta_{m,n} \le 2\pi) \Pr(\pi \le \theta_{m,n} \le 2\pi). \quad (13)
\end{aligned}$$

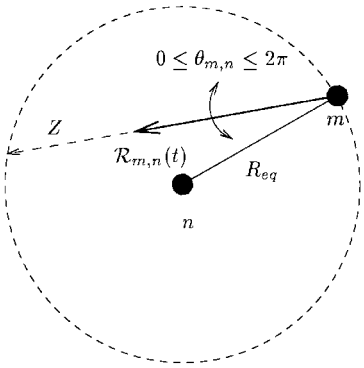


Fig. 10. Link activation model.

The conditional distribution of $\mathcal{R}_{m,n}(t)$ for $0 \le \theta \le \pi$ can be determined as follows, where $\alpha_{m,n}$ is defined in (8):

$$\begin{aligned}
&\Pr(\mathcal{R}_{m,n}(t) < Z | 0 \le \theta_{m,n} \le \pi) \\
&\quad = \int_{-\infty}^{\infty} \Pr(\mathcal{R}_{m,n}(t) < Z | Z = z) f_{Z|0\le\theta_{m,n}\le\pi}(z)\, dz \\
&\quad = \int_{-\infty}^{\infty} \Pr(\mathcal{R}_{m,n}(t) < z) f_{Z|0\le\theta_{m,n}\le\pi}(z)\, dz \\
&\quad = \int_0^{2R_{\text{eq}}} \frac{1}{\pi} \frac{\left(1 - \exp\left(\frac{-z^2}{\alpha_{m,n}}\right)\right)}{\sqrt{R_{\text{eq}}^2 - (z/2)^2}}\, dz \\
&\quad = 1 - I_0\left(\frac{-2R_{\text{eq}}^2}{\alpha_{m,n}}\right) \exp\left(\frac{-2R_{\text{eq}}^2}{\alpha_{m,0n}}\right). \quad (14)
\end{aligned}$$

The distribution of node m's trajectory is uniform over (0, 2π). Consequently, the probability that the trajectory is in the range (0, π) is exactly 0.5. Furthermore, since the value of Z over (π, 2π) is zero, the conditional probability $\Pr(\mathcal{R}_{m,n}(t) < Z | \pi \le \theta \le 2\pi)$ is equal to zero. Based on these observations, (13) reduces to the following expression, which combined with (14), yields the final result that according to (11) is the link availability:

$$\Pr(\mathcal{R}_{m,n}(t) < Z) = \tfrac{1}{2} \Pr(\mathcal{R}_{m,n}(t) < Z | 0 \le \theta_{m,n} \le \pi).$$

QED

ACKNOWLEDGMENT

The authors wish to thank Dr. R. J. Sclabassi of the Departments of Electrical Engineering and Neurosurgery at the University of Pittsburgh for his support and continuing guidance in the completion of this work. They also wish to thank M. Stover of Siemens ICN, and the anonymous referees and editors for their valuable comments that greatly improved the content and readability of this paper.

REFERENCES

[1] A. Ephremides, J. E. Wieselthier, and D. Baker, "A design concept for reliable mobile radio networks with frequency hopping signaling," *Proc. IEEE*, vol. 75, no. 1, pp. 56–73, 1987.

[2] A. Alwan, R. Bagrodia, N. Bambos, M. Gerla, L. Kleinrock, J. Short, and J. Villasenor, "Adaptive mobile multimedia networks," *IEEE Personal Commun. Mag.*, vol. 3, no. 2, pp. 34–51, Apr. 1996.

[3] P. Beckmann," *Probability in Communication Engineering.* New York: Harcourt Brace World, Inc., 1967.
[4] J. Broch, D. Maltz, D. Johnson, Y. Hu, and J. Jetcheva, "A performance comparison of multi-hop wireless ad hoc routing protocols," in *Proc. Fourth Ann. ACM/IEEE Int. Conf. Mobile Computing and Networking*, Oct. 1998.
[5] C. Barnhart, J. E. Wieselthier, and A. Ephremides, "Admission-control policies for multihop wireless networks," *Wireless Networks*, vol. 1, no. 4, pp. 373–389, 1995.
[6] M. S. Corson, S. Papademetriou, P. Papadopoulos, V. Park, and A. Qayyum, "An internet MANET encapsulation protocol (IMEP) specification," Internet Draft, Aug. 1998.
[7] M. Scott Corson and A. Ephremides, "A distributed routing algorithm for mobile wireless networks," *Wireless Networks*, vol. 1, pp. 61–81, 1995.
[8] S. Das, R. Castaneda, J. Yan, and R. Sengupta, "Comparative performance evaluation of routing protocols for mobile, ad hoc networks," in *Proc. Seventh Ann. ICCCN*, Oct. 1998, pp. 153–161.
[9] J. J. Garcia-Lunes-Aceves and J. Behrens, "Distributed, scalable routing based on vectors of link states," *IEEE J. Select. Areas Commun.*, vol. 13, no. 8, pp. 1383–1395, Oct. 1995.
[10] M. Gerla and J. T. Tsai, "Multicluster, mobile, multimedia radio network," *Wireless Networks*, vol. 1, pp. 255–265, Oct. 1995.
[11] Z. J. Haas and M. Perlman, "The performance of query control schemes for the zone routing protocol," in *Proc. ACM SIGCOMM'98*.
[12] ——, "Panel report on ad hoc networks," *Mobile Comput. Commun. Rev.*, vol. 2, no. 1, 1997.
[13] Z. J. Haas and M. Pearlman, "The zone routing protocol (ZRP) for ad hoc networks," Internet Draft, Tech. Rep., Nov. 1997.
[14] D. Hong and S. Rappaport, "Traffic models and performance analysis for cellular mobile radio telephone systems with prioritized and nonprioritized handoff procedures," *IEEE Trans. Veh. Technol.*, vol. 35, no. 3, pp. 77–92, Aug. 1986.
[15] D. Johnson and D. Maltz, "Dynamic source routing in ad hoc wireless networks," in *Mobile Computing*, T. Imielinski and H. Korth, Eds. Norwell, MA: Kluwer, 1996.
[16] J. Jubin and J. D. Tornow, "The DARPA packet radio network protocols," *Proc. IEEE*, vol. 75, no. 1, 1987.
[17] F. Kamoun and L. Kleinrock, "Hierarchical routing for large networks: Performance evaluation and optimization," *Comput. Networks*, vol. 1, pp. 155–174, 1977.
[18] ——, "Stochastic performance evaluation of hierarchical routing for large networks," *Comput. Networks*, vol. 3, pp. 337–353, 1979.
[19] L. Kleinrock, *Queueing Systems Volume 1: Theory*. New York: Wiley, 1975.
[20] G. Lauer, "Packet-radio routing," M. Steenstrup, Ed. Englewood Cliffs, NJ: Prentice-Hall, 1995.
[21] C. R. Lin and M. Gerla, "Adaptive clustering for mobile wireless networks," *IEEE J. Select. Areas Commun.*, vol. 15, no. 7, pp. 1265–1275, Sept. 1997.
[22] A. B. McDonald and T. Znati, "Link availability models for mobile ad-hoc networks," Comput. Sci. Dept., Univ. Pittsburgh, Pittsburgh, PA, Tech. Rep. 99-07, 1999.
[23] ——, "Performance evaluation of neighbor greeting procotols: ARP versus ES-IS," *Comput. J.*, vol. 39, no. 10, pp. 854–867, 1996.
[24] S. Murthy and J. J. Garcia-Lunes-Aceves, "An efficient routing protocol for wireless networks," *ACM Balzer Mobile Networks Applicat. J.*, vol. 1, no. 2, pp. 183–197, 1996.
[25] N. H. Vaidya, P. Krishna, M. Chatterjee, and D. K. Pradhan, "A cluster-based approach for routing in dynamic networks," *ACM Comput. Commun. Rev.*, vol. 27, no. 2, Apr. 1997.
[26] V. Park and S. Corson, "Temporally-ordered routing algorithm (TORA) version 1," Internet draft, Aug. 1998.
[27] ——, "A highly adaptive distributed routing algorithm for mobile wireless networks," in Proc. *IEEE INFOCOM*, Apr. 1997, pp. 1405–1413.
[28] C. Perkins and E. Royer, "Ad hoc on demand distance vector (AODV) routing," Internet draft, Nov. 1998.
[29] C. R. Perkins and P. Bhagwat, "Highly dynamic destination sequenced distance vector routing (DSDV) for mobile computers," in *Proc. ACM SIGCOMM*, Oct. 1994, pp. 234–244.
[30] R. Perlman, "Fault-tolerant broadcast of routing information," *Comput. Networks*, vol. 7, pp. 395–405, 1983.
[31] R. Ramanathan and M. Steenstrup, "Hierarchically-organized, multihop mobile wireless networks for quality-of-service support," *Mobile Networks and Applications*, vol. 3, no. 1, pp. 101–119, 1998.
[32] M. Zonoozi and P. Dassanayake, "User mobility modeling and characterization of mobility patterns," *IEEE J. Select. Areas Commun.*, vol. 15, no. 7, pp. 1239–1252, Sept. 1997.

A. Bruce McDonald (S'94) received the B.S. degree in electrical engineering from Northwestern University, Evanston, IL, in 1986 and the M.S. degree in telecommunications from the University of Pittsburgh, Pittsburgh, PA, in 1994. He is currently working toward the Ph.D. degree at the University of Pittsburgh.

He has worked in industry as both a Systems and Network Engineer and is currently a Senior Computer Engineer in the Department of Neurophysiology at the Children's Hospital of Pittsburgh, Pittsburgh, PA. Prior to his current position, he completed an internship in the Applied Network Research Group at Bellcore, Redbank, NJ. His research interests are in routing algorithms, ad hoc networks, mobility management, communications protocols, and distributed systems.

Taieb F. Znati (A'91) received the M.S. degree from Purdue University, West Lafayette, IN, and the Ph.D. degree in computer science at Michigan State University, East Lansing, in 1988.

In 1988, he joined the University of Pittsburgh, Pittsburgh, PA, where he is currently an Associate Professor in the Department of Computer Science. He also holds two joint appointments, one in telecommunications with the Department of Information Science and another in computer engineering. His current research interests focus on the design of network protocols for wired and wireless communication networks to support multimedia applications' quality-of-service requirements, the design and analysis of medium access control protocols to support distributed real-time systems, and the investigation of fundamental design issues related to distributed systems. He is also a member of the Editorial Board of the *International Journal of Parallel and Distributed Systems and Networks*.

Dr. Znati served as the General Chair of CNDS'99 and the 32nd Annual Simulation Symposium. He also served as a Program Chair for numerous conferences and workshops in networking and distributed multimedia systems.