Hierarchical Routing Overhead in Mobile Ad Hoc Networks

John Sucec, *Member*, *IEEE*, and Ivan Marsic, *Member*, *IEEE*

Abstract—Hierarchical techniques have long been known to afford scalability in networks. By summarizing topology detail via a hierarchical map of the network topology, network nodes are able to conserve memory and link resources. Extensive analysis of the memory requirements of hierarchical routing was undertaken in the 1970s. However, there has been little published work that assesses analytically the communication overhead incurred in hierarchical routing. This paper assesses the scalability, with respect to increasing node count, of hierarchical routing in mobile ad hoc networks (MANETs). The performance metric of interest is the number of control packet transmissions per second per node (ϕ). To derive an expression for ϕ, the components of hierarchical routing that incur overhead as a result of hierarchical cluster formation and location management are identified. It is shown here that ϕ is only polylogarithmic in the node count.

Index Terms—Mobile ad hoc network, routing, hierarchical techniques, scalability, control overhead.

1 Introduction

Mobile ad hoc networks (MANETs) are comprised of mobile nodes that perform multihop datagram forwarding over wireless links. The mobility of network nodes combined with the transient nature of wireless links results in a frequently changing network topology that is far more dynamic than the topologies typical of wired networks. Further, it is commonly assumed for MANETs that the wireless links tend to be low capacity homogeneous links (i.e., no hierarchy in the physical topology of the network). This means that neither traffic aggregation nor summarization of routing information can be achieved through hierarchically proportioned physical links. Thus, not only is maintaining and acquiring routing information in MANETs difficult to achieve due to link state volatility, but so is achieving this in a manner that scales well with increasing network size.

This paper addresses the scalability, with respect to increasing node count, of hierarchical routing in MANETs. The performance metric under consideration is the control overhead required by hierarchical routing in terms of *packet transmissions per node per second* (ϕ). This assessment considers the overhead due to the maintenance of routing tables and hierarchical clustering as well as the overhead due to address management (or, equivalently, location management). An important finding of this paper is that ϕ is a polylogarithmic function of the network node count (N).

Numerous papers have been published on hierarchical routing. Among these are [1], [2], [3], [4], [5], [6], [7]. Although a number of these papers provide detailed assessments of hierarchical routing performance, only [3] and [7] attempt to quantify analytically control packet overhead. In [7], control packet overhead required for constructing routing tables in a *two*-level hierarchically organized network is considered. However, the analysis was performed to address primarily the overhead of updates due to link cost changes and did not address the control packet overhead incurred by node mobility. In [3], the control packet overhead required for routing table maintenance is also considered but for a three-level hierarchical network. However, the assessment in [3] does not consider the case where the number of hierarchical levels is logarithmic in N or attempt to explicitly bound ϕ as a function of N.

Scalability performance metrics, considered elsewhere, include routing table storage overhead and the ratio of hierarchical path length to least-hop path length. Although these metrics are important, control packet overhead is of chief interest here. The justification for focusing on ϕ is as follows: First, control packet overhead is arguably more critical than routing table size because scarce wireless link capacity poses a more severe performance limit than the available memory in today's computers. Second, while extensive earlier work exists that analyzes hierarchical path lengths (e.g., [8]), little analysis has been published that assesses ϕ.

The remainder of this work is organized as follows: Section 2 describes notation and assumptions related to the essential features of the network environment. Section 3 provides an overview of hierarchical routing. Section 4 derives an expression for the frequency of cluster link state change events that is essential for subsequent control overhead assessment. Section 5 applies the result proven in Section 4 to evaluate ϕ. Conclusions are provided in Section 6.

- *The authors are with the Center for Advanced Information Processing (CAIP), Rutgers University, 96 Frelinghuysen Road, Piscataway, NJ 08855. E-mail: {jsucec, marsic}@caip.rutgers.edu.*

Manuscript received 8 June 2002; revised 11 July 2003; accepted 22 Oct. 2003. For information on obtaining reprints of this article, please send e-mail to: tmc@computer.org, and reference IEEECS Log Number 1-062002.

2 Network Environment

2.1 Assumptions

The underlying physical topology of a MANET is represented here by a *connected*, undirected graph, $G = (V, E)$, where V is the set of nodes and E is the set of bidirectional links. It is assumed that, at any time, nodes are situated randomly throughout the network area in accordance with a two-dimensional uniform random variable distribution. For the purpose of analyzing the frequency of location update events, the random waypoint model for node mobility, employed in [9], with zero pause time is assumed.

Each node is equipped with a single network interface card (NIC) having a transmission radius of R_{TX}. If the distance separating a pair of nodes is less than R_{TX}, then a bidirectional link connects them and they are considered to be neighbors. Otherwise, the nodes are not connected. Each NIC employs carrier sense multiple access with collision avoidance (CSMA/CA).

The scalability of a routing protocol may be assessed in terms of a number of distinct criteria. Among these are scalability with respect to increasing node count (N) and increasing average node density (nodes per unit area). In order to isolate the performance of hierarchical routing with respect to increasing N, it is assumed that average node density and average node speed (m/s) are held constant. It is shown in [10] that, when the node density is constant with respect to increasing node count (i.e., network area is proportional to N), the average hop count on the shortest path between an arbitrary pair of nodes in a two-dimensional network consisting of N nodes is proportional to $\sqrt{N}$.

Last, to realize a hierarchical architecture, nodes organize themselves into clusters in accordance with some clustering algorithm. It is assumed that the clustering is performed recursively, resulting in a hierarchical cluster *level* count (L) that is logarithmic in N.

2.2 Notation

- $V_k \equiv$ Set of level-*k* clusters at level-*k* of the cluster hierarchy ($V_0 = V$).
- $E_k \equiv$ Set of level-*k* links at level-*k* of the cluster hierarchy ($E_0 = E$).
- $N \equiv$ Number of network nodes ($N = |V|$).
- $N_k \equiv$ Number of level-*k* clusters, i.e., $N_k = |V_k|$.
- $L \equiv$ Number of levels in the cluster hierarchy.
- $\phi \equiv$ Average number of control packet transmissions per second per node.
- $\psi \equiv$ Average volume of control overhead in bits per second per node.
- $c_k \equiv N/N_k$, i.e., the average number of level-0 clusters subsumed by a level-*k* cluster.
- $A_k \equiv$ Average area covered by a level-*k* cluster.
- $f_k \equiv$ Average frequency of level-*k* cluster link state change events.
- $g_k \equiv$ Average frequency of level-*k* cluster *migration* events, i.e., the frequency at which level-*k* clusters migrate sufficiently close to one another to create a new level-*k* link or migrate sufficiently far from one another to delete a level-*k* link.
- $\alpha_k \equiv N_{k-1}/N_k$, $k \in \{1, 2, \ldots, L\}$.
- $h \equiv$ Average number of hops separating a pair of communicating nodes.
- $h_k \equiv$ Average hop count, in terms of level-0 clusters (i.e., nodes), across a level-*k* cluster.
- $d_k \equiv$ Average number of level-*k* neighbors for a level-*k* cluster.
- $\mu \equiv$ Average node speed.
- $p_{j,k} \equiv$ Probability that relative mobility between a pair of level-*j* clusters, which triggers a level-*j* cluster link state change, also triggers one or more level-*k* cluster link state changes, $j \le k$, $p_{k,k} = 1$.
- $l_{j,k} \equiv$ Average number of level-*k* cluster link state changes per level-*j* cluster link state change trigger that impacts level-*k*, $j \le k$, $l_{k,k} = 1$.
- $g(x) = \Theta(f(x)) \equiv \exists\, a > 0$ and $\exists\, b > 0$ and $\exists\, x_0$ such that $\forall\, x > x_0$: $0 \le a \cdot f(x) \le g(x) \le b \cdot f(x)$.
- $g(x) = \mathrm{O}(f(x)) \equiv \exists\, b > 0$ and $\exists\, x_0$ such that $\forall\, x > x_0$: $0 \le g(x) \le b \cdot f(x)$.
- $g(x) = \Omega(f(x)) \equiv \exists\, a > 0$ and $\exists\, x_0$ such that $\forall\, x > x_0$: $0 \le a \cdot f(x) \le g(x)$.

Some consequences related to this notation and the earlier stated assumptions are as follows:

$$f_0 = \Theta\left(\frac{\mu}{R_{\mathrm{TX}}} \cdot |E|\right) = \Theta\left(\frac{\mu}{R_{\mathrm{TX}}} \cdot N \cdot \frac{d_0}{2}\right) = \Theta(N), \tag{1}$$

$$h_k = \Theta\left(\prod_{j=1}^{k} \sqrt{\alpha_j}\right) = \Theta(\sqrt{c_k}), \tag{2}$$

$$d_k = \Theta(1), \forall\, k \in \{0, 1, \ldots, L-1\}, \tag{3}$$

$$c_k = \prod_{j=1}^{k} \alpha_j = \Theta(A_k). \tag{4}$$

Equation (2), of course, follows from the result of [10]. Last, it is noted that, in this paper, the expressions "level-*k* node" and "level-*k* cluster" may be used interchangeably.

3 Hierarchical Routing Overview

3.1 Hierarchical Routing Principles

Fig. 1 illustrates the fundamental concept of a cluster hierarchy. All network nodes (i.e., V) are level-0 clusters. Level-0 clusters organize themselves into level-1 clusters, via some clusterhead election process such as one of the methods of [11], [12], [13], [14]. The level-1 clusters, in turn, organize themselves into level-2 clusters. That is, a level-*k* node which is elected as the clusterhead for a level-*k* cluster becomes a level-(*k*+1) node. This clustering procedure is performed recursively until the desired number of cluster levels has been constructed.

The principles of hierarchical routing have seen application in military-based packet radio networks, such as the Survivable Packet Radio Network (SURAN) described in [2] and [3]. More recently, the Hierarchical State Routing (HSR) protocol proposed in [4], [5] and multimedia support for mobile wireless networks (MMWN) proposed in [6] represent hierarchical approaches that support group mobility and multimedia, respectively, in MANETs.

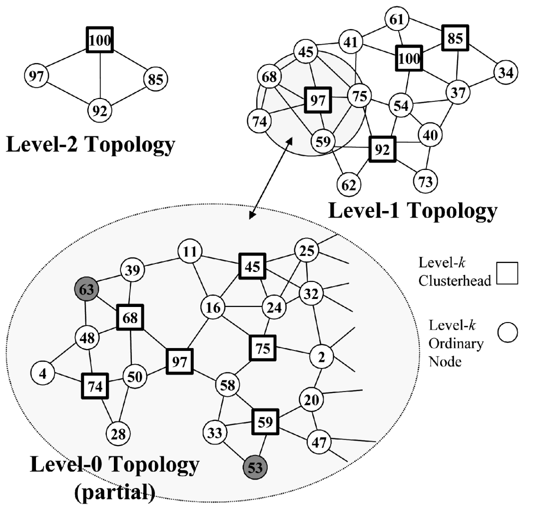

Fig. 1. Example of 3-level hierarchy.

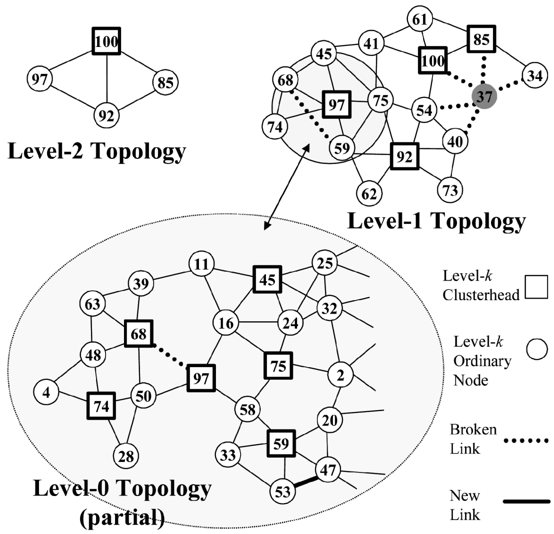

Fig. 2. Examples of cluster link state changes.

The analysis of this paper assumes *strict hierarchical routing*, based on the description provided in [15], to be in effect. HSR and MMWN are examples of strict hierarchical routing implementations. An implementation recommended for SURAN in [3] also falls into this category. Three important concepts concerning packet forwarding in hierarchical networks are as follows:

1. Packet-forwarding decisions are based on the hierarchical address of the destination node.
2. Every node has a $\Theta(\log N)$ hierarchical topology map for the clusters of the network hierarchy to which it belongs.
3. The forwarding of datagrams need *not* be directed through clusterheads and are forwarded via clusterhead and/or nonclusterhead nodes alike to the destination.

Last, an example of a hierarchical address is given based on Fig. 1. Node 63 is considered. Node 63 is a member of the level-1 cluster 68 ($68_1.0_0$) which, in turn, is a member of level-2 cluster 97 ($97_2.0_1.0_0$). Thus, the hierarchical address for 63 (63_0) is $97_2.68_1.63_0$. A source node, say 53 ($97_2.59_1.53_0$), can initiate unambiguous packet forwarding toward 63 simply by knowing the hierarchical address of 63 (i.e., $97_2.68_1.63_0$) and by each intermediary node having a copy of the $\Theta(\log N)$ hierarchical topology map for the clusters of the network hierarchy to which it belongs. That is, packet forwarding to 63 ($97_2.68_1.63_0$) would consist of first forwarding datagrams to the level-2 cluster 97 then to the level-1 cluster 68 and finally to 63 itself.

3.2 Clustering Techniques

A number of clustering schemes have been proposed in previous literature (e.g., [11], [12], [13], [14]). Of particular interest here are the max-min D-hop clustering strategy of [11] and the linked cluster algorithm (LCA) of [12]. Each of these approaches is an ID-based clustering technique. The max-min D-hop strategy is shown to converge in $O(D)$ rounds and generates only $O(D)$ messages per node. It represents, therefore, a scalable clustering procedure. The 1-hop clustering case is equivalent to an *asynchronous* version of the LCA. It is an asynchronous version that is assumed to be in effect for election of level-k clusterheads, $k \in \{0, 1, \ldots, L\}$, known here as asynchronous LCA (ALCA).

Based on the analysis of [11], formation of 1-hop clusters requires only two rounds of communication. This fact is significant for bounding the overhead required for cluster maintenance. That is, once level-k clusterheads are initially elected by the ALCA, $k \in \{0, 1, \ldots, L-1\}$, if a single perturbation to the network topology (e.g., cluster creation) triggers a clusterhead reorganization, then reorganization will incur only two rounds of messaging (i.e., the messaging required to react to a topology perturbation is confined to a two-hop radius about the perturbation). Thus, unlike some of the other clustering approaches where a single level-k perturbation can subsequently affect the entire set of level-k clusterheads, a level-k topology change has only local effect for the ALCA.

The ALCA election process is now described briefly. Essentially, a level-k node v_k is elected as a level-k clusterhead by a neighbor u_k if the node ID of v_k is the largest among all nodes in the closed neighborhood of u_k. For example, in the level-0 topology of Fig. 1, node 97 is elected to serve as a clusterhead because it is the largest node in its neighborhood. As another example, node 68 is also elected to serve as a clusterhead because it has the largest node ID in the level-0 neighborhood of node 63, even though 68 is *not* the largest node in its own level-0 neighborhood. The recursive application of this election process is illustrated in Fig. 1 by the level-1 and level-2 topologies.

Referring to Figs. 1 and 2, a discussion of hierarchical cluster link state changes is given. First, link state changes may be due to link creations or link deletions. Fig. 2 depicts a new link that does not appear in Fig. 1 between nodes 47 and 53 while the link between nodes 68 and 97 that appears in Fig. 1 has been deleted in Fig. 2. Such level-0 link state changes are due to node mobility. Second, link state changes may occur at any of the L levels in the cluster hierarchy. For example, as shown in Fig. 2, the deletion of the level-0 link between nodes 68 and 97 also resulted in the level-1 link between $59_1.0_0$ and $68_1.0_0$ being broken. Third, level-*k* links may be created/deleted as a result of creation/deletion of level-*k* nodes. For example, the deletion of the level-1 cluster $37_1.0_0$ in Fig. 2 results in multiple level-1 links being deleted. This latter phenomenon is discussed in greater detail in Section 4.

3.3 Hierarchical Location Management

As discussed already, packet forwarding may be achieved provided a hierarchical map is available at each node and the hierarchical address of the destination is known. In order for an originating node to learn the hierarchical address of an intended destination, an address management or, equivalently, a *location management* scheme is required.

The hierarchical location management (HLM) service is now briefly described. Its concept is inspired by the Grid Location Service (GLS) originally proposed in [16]. In both GLS and HLM, a set of location management (LM) servers is selected for each node v to function as a distributed database. The distributed database houses for v its geographic coordinates (e.g., GLS) or address information (e.g., HLM). The salient features of the distributed database are as follows:

1. The set of nodes functioning as LM servers for v are based on the relation of their node ID to that of v and their relative proximity in the grid or cluster hierarchy.
2. The density of LM servers for v in regions near v is high and low in the regions far from v.
3. The frequency at which v updates its location to nearby LM servers is high while servers situated far from v receive updates at a low frequency.

Item 1 affords, for each level in the cluster hierarchy, unambiguous selection of a candidate node to function as a member of the distributed LM server set for v. Second, it provides a means to direct *queries* properly to the server set. Third, provided a suitable hashing function is applied for server selection, it distributes the load of server functionality equitably throughout V. Items 2 and 3 provide favorable scalability to HLM. These features effectively summarize LM detail about v in regions of the network that are far from v.

It is also evident that maintaining LM data at $L = \Theta(\log N)$ levels in the cluster hierarchy means that each node acts as an LM server for $\Theta(\log N)$ nodes, on average. That is, each node serves as the level-*k* LM server for, on average, one other node, $k \in \{1, 2, \ldots, L\}$. Thus, the decentralized nature of HLM avoids concentration of LM data at dedicated servers which, as a result, would become traffic hot spots.

The LM procedure is now illustrated. It is supposed that node 28, a member of level-1 cluster 74, wishes to initiate a communication session with 63. Fig. 3 depicts the network topology of interest for this example. Since 28 and 63 do not belong to the same level-1 cluster, 28 must send an LQ message to the level-1 LM server of 63. Previously, 63 had registered its level-1 cluster membership with its level-1 LM server, $45_1.11_0$. First, 63 consulted its level-1 cluster topology map to determine which level-1 cluster ($45_1.0_0$, $59_1.0_0$, $68_1.0_0$, $74_1.0_0$, $75_1.0_0$ or $97_1.0_0$) should house its level-1 LM server. Applying a hashing function that operates on the candidate level-1 cluster IDs, $45_1.0_0$ is computed as the cluster to which 63 should forward location registration (LR) updates. Upon arrival of an LR packet from 63 at a cluster member of $45_1.0_0$, a similar (but subtly different) hashing function is applied to compute the actual cluster member of $45_1.0_0$ to which the LR packet should be forwarded. In this case, node 11 is computed as the level-1 LM server for 63. Details of the hashing functions are not necessary to understand for the purposes of this paper and, therefore, are omitted here.

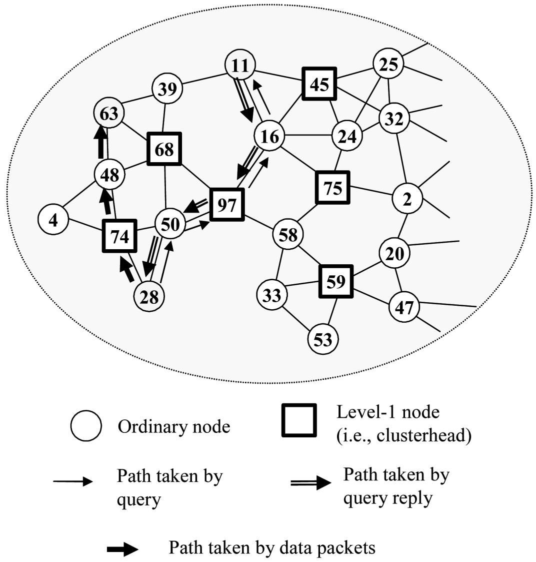

Fig. 3. Topology of level-2 cluster $97_2.0_1.0_0$ with LQ requestor (28), LQ target (63), and level-1 LM server (11).

Node 28 wishing to communicate with 63 has the same level-1 topology map as 63 and applies the same hashing function as 63 and, therefore, computes $45_1.0_0$ as the level-1 cluster to which its LQ packet should be forwarded. When the LQ packet arrives at a member of $45_1.0_0$ (node 16, in this case), a hashing function computes node 11 as the level-1 LM server for 63, applying the same procedure used above to direct the LR packet within $45_1.0_0$. Thus, LQ and LR packets alike may be addressed and forwarded correctly to the appropriate LM server. Upon arrival of the LQ at 11 (the level-1 LM server for 63), a query reply containing the level-1 cluster location of 63 ($68_1.0_0$) is addressed and forwarded in response to the requesting node, 28. Having learned the hierarchical address of 63 ($68_1.63_0$), packet forwarding from 28 to 63 may begin. Last, it is noted that the addressing of LQ and LR packets to clusters may be facilitated by *anycast*, e.g., as in IPv6 [17].

4 Cluster Link State Change Frequency

Several of the factors contributing to control overhead, considered in Section 5, depend on the frequency of cluster link state changes (f_k) at level-*k* in the cluster hierarchy. Due to the complex dynamics of the ALCA, the derivation of f_k is nontrivial for $k > 0$.

4.1 Link State Change Theorem

At level-0 (node level) in the cluster hierarchy, link state changes are due entirely to mobility of level-0 clusters (i.e., individual nodes). Thus, f_0 is simply a function of N, μ, R_{TX}, and d_0.

In general, however, cluster link state changes at some level-*k* in the hierarchy, $k > 0$, may actually be due to cluster mobility at level-*j*, $j \in \{0, 1, \ldots, k\}$. That is, a level-*k* cluster link state change may be due to the migration of level-*k* nodes or due to the migration of level-(*k*-1) nodes which resulted in the creation (or deletion) of a level-*k* cluster or due to the migration of level-(*k*-2) nodes which resulted in the creation (or deletion) of a level-(*k*-1) cluster which, in turn, resulted in the creation (or deletion) of a level-*k* cluster, etc. It is evident, therefore, that migration of level-*j* nodes can affect level-*k* cluster link states ($j < k$) as a result of recursive application of the ALCA. This is referred to here as *recursive cluster creation/deletion*.

The primary outcome of this section is the following theorem which quantifies f_k.

Theorem. *The frequency of level-k cluster link state change events is related to N and c_k as follows:*

$$f_k = \Theta\left(\frac{N}{c_k^{3/2}}\right). \tag{5}$$

4.2 Proof of Theorem

Since the frequency of cluster migration events (g_k) is used in the derivation of f_k, g_k is considered initially. Cluster migration contributes to f_k and, therefore, $g_k \leq f_k$. Clearly, a necessary condition is $g_k = \mathrm{O}\left(N \cdot c_k^{-3/2}\right)$.

Assuming $d_k \cong d_{k-1}\ \forall k > \kappa = \Theta(1)$, then, since there are α_k fewer level-*k* clusters than level-(*k*-1) clusters, there also must be α_k fewer level-*k* migration events than level-(*k*-1) migration events. Thus, a factor of c_k is accounted for because there are $\Theta(c_k)$ times as many level-0 links as there are level-*k* links. Formally, this is stated as follows:

$$g_k \leq g_0 \cdot \frac{|E_k|}{|E_0|} = \mathrm{O}\left(g_0 \cdot \frac{N/c_k}{N}\right) = \mathrm{O}\left(\frac{N}{c_k}\right). \tag{6}$$

Now, g'_k is defined as the frequency of level-*k* migration events per level-*k* cluster link. If $g'_k = \Theta\left(1/\sqrt{c_k}\right)$, then $g_k = \mathrm{O}\left(N \cdot c_k^{-3/2}\right)$ holds.

$$g'_k \equiv \frac{g_k}{|E_k|} = \frac{2 \cdot g_k}{d_k \cdot N_k} = \frac{2 \cdot c_k \cdot g_k}{d_k \cdot N} = \Theta\left(\frac{c_k \cdot g_k}{N}\right), \tag{7a}$$

$$\Leftrightarrow g_k = \Theta\left(\frac{N \cdot g'_k}{c_k}\right). \tag{7b}$$

To justify $g'_k = \Theta\left(1/\sqrt{c_k}\right)$, it is observed that node migration frequency is a function of cluster area and node speed:

$$g'_k = \Theta\left(\frac{\mu}{\sqrt{A_k}}\right) \Rightarrow g'_k = \Theta\left(\frac{1}{\sqrt{c_k}}\right), \tag{8a}$$

$$\Rightarrow g'_k = \Theta\left(\prod_{j=0}^{k} \frac{1}{\sqrt{\alpha_j}}\right). \tag{8b}$$

The result of (8a), of course, is arrived at by recalling that $c_k = \Theta(A_k)$ as per (4) and that $\mu = \Theta(1)$. Equation (8b) follows from (8a) by applying (4).

Now, f_k is clearly dependent on g_k, the frequency of level-*k* migration events. However, due to the effect of recursive cluster creation/deletion, f_k must be expressed as a weighted sum of g_j terms, $0 \leq j \leq k$:

$$f_k = \sum_{j=0}^{k} p_{j,k} \cdot l_{j,k} \cdot g_j. \tag{9}$$

The following lemma is useful in simplifying (9).

Lemma 1. $l_{j,k} = \Theta(1)$.

Proof. A proof is provided in the Appendix. □

Applying Lemma 1 to (9) yields:

$$f_k = \Theta\left(\sum_{j=0}^{k} p_{j,k} \cdot g_j\right). \tag{10}$$

As in the evaluation of g_k, the aggregation of cluster links by a factor of α_k at each level-*k* in the cluster hierarchy impacts f_k:

$$f_k \leq f_0 \cdot \frac{|E_k|}{|E_0|} = \mathrm{O}\left(f_0 \cdot \frac{N/c_k}{N}\right) = \mathrm{O}\left(\frac{N}{c_k}\right). \tag{11}$$

As indicated by (11), the effect of recursive cluster aggregation and, thus, link aggregation reduces f_k by a factor of c_k with respect to f_0. If the cluster link state change frequency (per level-*k* link) is $\mathrm{O}\left(1/\sqrt{c_k}\right)$, then the theorem is proven.

Similar to the derivation of g'_k, f'_k is defined as the frequency of level-*k* cluster link state change events per level-*k* cluster link:

$$f'_k \equiv \frac{f_k}{|E_k|} = \frac{2 \cdot f_k}{d_k \cdot N_k} = \frac{2 \cdot c_k \cdot f_k}{d_k \cdot N} = \Theta\left(\frac{c_k \cdot f_k}{N}\right), \tag{12a}$$

$$\Leftrightarrow f_k = \Theta\left(\frac{N \cdot f'_k}{c_k}\right). \tag{12b}$$

Unlike g'_k, however, due to the effect of recursive cluster creation/deletion, f'_k must be expressed as a weighted sum:

$$f'_k = \Theta\left(\sum_{j=0}^{k} p_{j,k} \cdot g'_j\right). \tag{13}$$

The $l_{j,k}$ term of (9) does not appear in (13) because f'_k is the cluster link state change frequency *per* level-*k* link. Now, it is noted from (8) that the g'_j terms of (13) may be expressed in terms of c_k or as a function of the factors α_j:

$$\Rightarrow f'_k = \Theta\left(\frac{1}{\sqrt{c_k}} \cdot \sum_{j=0}^{k} \left[p_{j,k} \cdot \prod_{i=j+1}^{k} \sqrt{\alpha_i}\right]\right). \tag{14}$$

Moving forward, the following definitions are useful.

Definition. *A level-k cluster link state change trigger that modifies the underlying level-k graph* $G_k = (V_k, E_k) \rightarrow G'_k = (V'_k, E'_k)$ *is said to be due to level-k node migration if* $V'_k = V_k$. *This is referred to as trigger event type E-1.*

Definition. *A level-k cluster link state change trigger that modifies the underlying level-k graph* $G_k = (V_k, E_k) \rightarrow G'_k = (V'_k, E'_k)$ *is said to be due to level-j node migration,* $j < k$, *as a result of recursive cluster creation/deletion if* $V'_k \neq V_k$. *This is referred to as trigger event type E-2.*

Clearly, f'_k may be due to level-*k* cluster migration (trigger event type E-1) or due to recursive cluster creation/deletion as a result of a level-*j* cluster migration event (trigger event type E-2). Type E-1 trigger events occur with frequency g'_k per level-*k* link, which is given by (8). What remains to be quantified in order to evaluate f_k is the effect of type E-2 triggers. If the cumulative frequency of such triggers is $O(g'_k)$, then (5) is proven.

Working forward from (14), it is easier to assess f'_k by considering the frequency of cluster link deletion and cluster link creation events in turn. That is, f'_k is the sum of the cluster link deletion and cluster link creation frequencies. Further, it is desirable that only cluster link deletion frequency or cluster link creation frequency (but not both) be assessed. The following lemma permits this.

Lemma 2. *The frequency of level-k cluster link deletions due to type E-2 trigger events is equal to the frequency of level-k cluster link creations due to type E-2 trigger events.*

Proof. A proof is provided in the Appendix. □

The implication of Lemma 2 is that the total frequency of cluster link state changes due to type E-2 events is simply twice the frequency of link state changes due to cluster deletion events. Thus, the details of recursive cluster deletion, only, are considered here forward. The following additional definitions and lemma are useful.

Definition. *A* critical node *at level-k in the hierarchy is a node that has been elected as a clusterhead by a single level-(k-1) node.*

Definition. $\sigma_k \equiv$ *Fraction of level-k clusters that are critical level-k clusters.*

Lemma 3. $p_{j,k} \leq \prod_{i=j+1}^{k} \sigma_i$, $j < k$.

Proof. A proof is provided in the Appendix. □

Applying Lemma 2 and Lemma 3 to (14) yields:

$$f'_k = \Theta\left(\frac{1}{\sqrt{c_k}} \cdot \sum_{j=0}^{k} \left[\prod_{i=j+1}^{k} \sigma_i \cdot \sqrt{\alpha_i}\right]\right). \tag{15}$$

A fourth lemma is now provided to bound σ_k.

Lemma 4. $\sigma_k < 1/\sqrt{\alpha_k}$, $1 \leq k \leq L$.

Proof. A proof is provided in the Appendix. □

Definition. $\beta_{\max} \equiv \max\{\sigma_i \cdot \sqrt{\alpha_i}\}$, $i \in \{j+1, j+2, \ldots, k\}$.

Applying $\beta_{\max}$ and Lemma 4 to (15) yields:

$$f'_k = \Theta\left(\frac{1}{\sqrt{c_k}} \cdot \sum_{j=0}^{k} \prod_{i=j+1}^{k} \beta_{\max}\right), \tag{16a}$$

$$\Rightarrow f'_k = \Theta\left(\frac{1}{\sqrt{c_k}} \cdot \left(1 + \sum_{j=0}^{k} \beta_{\max}^{j}\right)\right), \tag{16b}$$

$$\Rightarrow f'_k = \Theta\left(\frac{1}{\sqrt{c_k}} \cdot \frac{1 - \beta_{\max}^{k+1}}{1 - \beta_{\max}}\right), \tag{16c}$$

$$\Rightarrow f'_k = \Theta\left(\frac{1}{\sqrt{c_k}}\right). \tag{16d}$$

Combining (16d) with (12b) yields (5) and the theorem is proven. □

5 Hierarchical Routing Overhead

Communication overhead in hierarchically organized networks results from the following:

- Hello protocol (ϕ_{HELLO}).
- Cluster formation and cluster maintenance messaging (ϕ_{CL}).
- Acquisition of local topology data when nodes migrate from one cluster to another (ϕ_{ACQ}).
- Flooding of cluster topology updates to cluster members (ϕ_{FLOOD}).
- Location registration events (ϕ_{REG}).
- Handoff or transfer of location management data (ϕ_{HANDOFF}).
- Location query events (ϕ_{QRY}).
- Addressing information required in datagram headers ($\psi_{\text{CTRL-HEADER}}$).

Total communication overhead per node in hierarchically organized networks is the sum of the above factors. The following claims are made regarding the *average* control overhead:

Claim 1: $\phi_{\text{HELLO}} = \Theta(1)$ packet transmissions per second per node (Section 5.1).

Claim 2: $\phi_{\text{CL}} = O(\log N)$ packet transmissions per second per node (Section 5.2).

Claim 3: $\phi_{\text{ACQ}} = O(\log N)$ packet transmissions per second per node (Section 5.3).

Claim 4: $\phi_{\text{FLOOD}} = O(\log N)$ packet transmissions per second per node (Section 5.4).

Claim 5: $\phi_{\text{REG}} = \Theta(\log N)$ packet transmissions per second per node (Section 5.5).

Claim 6: $\phi_{\text{HANDOFF}} = \Theta(\log^2 N)$ packet transmissions per second per node (Section 5.6).

Claim 7: $\phi_{\text{QRY}} = \Theta(h)$ packet transmissions per second per node (Section 5.7).

Claim 8: $\psi_{\text{CTRL-HEADER}} = \Theta(\log N)$ *bits per control message datagram* (Section 5.8).

5.1 Hello Protocol

The Hello protocol is employed for nodes to learn and verify adjacencies. Discovery of adjacencies can be facilitated by periodic broadcast of a single Hello message over the shared CSMA/CA media. This is sufficient for a node v to announce itself to all nodes within R_{TX} of it. Once an adjacency is discovered, robust exchange of neighborhood data can be facilitated by periodic one-to-one Hello messaging between v and its neighbors. This is done, on average, with d_0 neighbors. Considering the periodic broadcast of the Hello message and recalling (3), the total number of Hello message transmissions per Hello interval is $1 + d_0 = \Theta(1)$.

The frequency of Hello messages is also proportional to μ and inversely proportional R_{TX}. That is, $f_{\mathrm{HELLO}} = \Theta(\mu/R_{\mathrm{TX}})$ and applying the network framework assumptions of $\mu = \Theta(1)$ and $R_{\mathrm{TX}} = \Theta(1)$ yields $f_{\mathrm{HELLO}} = \Theta(1)$. Combining $f_{\mathrm{HELLO}} = \Theta(1)$ with $1 + d_0 = \Theta(1)$ transmissions per Hello interval means $\phi_{\mathrm{HELLO}} = \Theta(1)$, as per Claim 1.

5.2 Hierarchical Clustering

5.2.1 Cluster Formation

The initial formation of level-(k+1) clusters,

$$k \in \{0, 1, \ldots, L-1\},$$

involves the recursive application of the ALCA. At each level-k node, two rounds of communication must be performed with its level-k neighbors to elect a level-k clusterhead. The communication with level-k neighbors is via a path consisting of

$$h_k = \Theta\left(\prod_{j=1}^{k} \sqrt{\alpha_j}\right)$$

level-0 node hops. This would suggest that the overhead due to level-k cluster formation increases with k. However, the increase in path length between level-k nodes is offset by a decrease in the number of nodes $N_k = N/\prod_{j=1}^{k} \alpha_j$ at each successively higher level in the hierarchy.

Following the procedure of [11], level-1 cluster formation requires two rounds of communication between neighbors and $\Theta(N)$ network-wide communication overhead. Level-2 cluster formation is now evaluated. It is recalled that $h_1 = \Theta(1)$ represents the average number of hops separating adjacent level-1 nodes. Therefore, each level-1 node must communicate two rounds of cluster formation messaging with, on average, $d_1 = \Theta(1)$ neighboring level-1 nodes over paths consisting of, on average, h_1 hops. Since there are N/α_1 level-1 clusters in the network, the aggregate communication overhead due to level-2 cluster formation is

$$\begin{aligned}&(\text{two rounds of messaging}) \times (d_1 \text{ communication sessions}) \times \\ &(h_1 \text{ transmissions per message per communication session}) \times \\ &(N/\alpha_1 \text{nodes}) = 2 \cdot d_1 \cdot h_1 \cdot N/\alpha_1 = \Theta(N)\end{aligned}$$

packet transmissions. In general, the aggregate level-k cluster formation overhead ($\Phi_{\mathrm{CL-F},k}$) may be expressed as follows:

$$\Phi_{\mathrm{CL-F},k} = 2 \cdot d_{k-1} \cdot h_{k-1} \cdot N_{k-1} = \Theta(h_{k-1} \cdot N_{k-1}), \tag{17a}$$

$$\Rightarrow \Phi_{\mathrm{CL-F},k} = \mathrm{O}(N). \tag{17b}$$

Equation (17b) follows from (17a) by applying (2)-(4) and the definition of c_k.

Since there are $\Theta(\log N)$ cluster levels, the aggregate number of packet transmissions due to ALCA cluster formation is $\mathrm{O}(N \cdot \log N)$. Dividing by N yields $\mathrm{O}(\log N)$ overhead *per node* each time the network cluster hierarchy is initialized (presumably a rare occurrence).

5.2.2 Cluster Maintenance

The assessment of level-k cluster maintenance follows logic similar to that given for level-k cluster formation with the following differences. First, unlike cluster formation which occurs only during network initialization or when the network is reset for management purposes, level-k cluster maintenance events may be triggered as a result of a level-(k-1) link state change. Second, each cluster maintenance trigger impacts the two endpoint nodes of the affected link, thus, contributing a factor of two to maintenance overhead. Third, as discussed in Section 4.2, level-j link state changes can impact the level-k cluster link state ($j < k$) via recursive cluster creation/deletion. This is obviously relevant to cluster maintenance. The effect of recursive cluster creation/deletion is accounted for by f_{k-1}, as per the derivation of (5).

It is evident from (17a) and the example of cluster formation that level-k cluster maintenance depends additionally on d_{k-1} and h_{k-1}. Combining this with f_{k-1} and the fact that both endpoints affected by a link state change implement cluster maintenance yields:

$$\Phi_{\mathrm{CL-M},k} \leq 4 \cdot d_{k-1} \cdot f_{k-1} \cdot h_{k-1}, \tag{18a}$$

$$\Rightarrow \Phi_{\mathrm{CL-M}} \leq 4 \cdot \sum_{k=1}^{L} d_{k-1} \cdot f_{k-1} \cdot h_{k-1}. \tag{18b}$$

In (18), $\Phi_{\mathrm{CL-M}}$ is the aggregate control packet overhead due to cluster maintenance. Applying (2), (3), and (5) allows $\Phi_{\mathrm{CL-M}}$ to be expressed in terms of N.

$$\Phi_{\mathrm{CL-M}} = \mathrm{O}\left(\sum_{k=1}^{L} f_{k-1} \cdot h_{k-1}\right) = \mathrm{O}\left(\sum_{k=1}^{L} N\right), \tag{19a}$$

$$\Rightarrow \Phi_{\mathrm{CL-M}} = \mathrm{O}(N \cdot \log N). \tag{19b}$$

Dividing $\Phi_{\mathrm{CL-M}}$ by N yields $\phi_{\mathrm{CL-M}} = \mathrm{O}(\log N)$. Combining $\phi_{\mathrm{CL-M}}$ with the cluster formation overhead given above yields $\phi_{\mathrm{CL}} = \mathrm{O}(\log N)$, as per Claim 2.

5.3 Acquiring Cluster Topology Data

In order for a node v to forward datagrams based on strict hierarchical routing, v must know the topology for each cluster to which it belongs. That is, for each level-k cluster to which it belongs, v must know the IDs of the cluster members, the connectivity among cluster members, and to which clusters (if any) each member serves as a cluster gateway. Of these three data items, the intracluster connectivity contributes the most to the level-k routing table size—a contribution that is quadratic in the level-(k-1) neighbor count.

To assess level-*k* routing table size, $d_{\max}$ is defined as $\max\{d_0, d_1, \ldots, d_{L-1}\}$. That is, $d_{\max}$ is the maximum average degree of level-(*k*-1) link connectivity within a level-*k* cluster among all levels of the cluster hierarchy. From (3), it follows that $d_{\max} = \Theta(1)$ and $d_{\max}^2 = \Theta(1)$.

When a node v migrates from one level-*k* cluster to another, it acquires the $\Theta\left(d_{k-1}^2\right) = \Theta(1)$ routing table entries associated with that cluster. (This can be obtained from a neighbor of the new cluster by a single hop transmission.) Such level-*k* migration events occur with frequency $f_{\mathrm{MIG},k}$. Applying (4) and noting that $f_{\mathrm{MIG},k}$ is proportional to the node speed ($\mu = \Theta(1)$) and inversely proportional to the square root of the cluster area (A_k), it is evident that:

$$f_{\mathrm{MIG},k} = \Theta\left(\mu/\sqrt{A_k}\right) = \Theta(1/\sqrt{c_k}). \tag{20}$$

The aggregate topology acquisition overhead (Φ_{ACQ}), therefore, is a function of d_k and $f_{\mathrm{MIG},k}$ summed over the cluster topology levels for which cluster migration is possible:

$$\Phi_{\mathrm{ACQ}} = \Theta\left(N \cdot \sum_{k=1}^{L-1} d_{k-1}^2 \cdot f_{\mathrm{MIG},k}\right) = \Theta\left(N \cdot \sum_{k=1}^{L-1} \frac{1}{\sqrt{c_k}}\right), \tag{21a}$$

$$\Rightarrow \Phi_{\mathrm{ACQ}} = \mathrm{O}(N \cdot \log N). \tag{21b}$$

The per node topology acquisition overhead is, therefore, $\phi_{\mathrm{ACQ}} = \mathrm{O}(\log N)$ as per Claim 3.

5.4 Link State Packet Flooding

When a level-*k* cluster link state change occurs (i.e., a link creation or link deletion between a pair of level-*k* clusters), a link state update must be flooded to the $\Theta(c_{k+1})$ members of the affected level-(*k*+1) cluster. Applying the theorem of Section 4, level-*k* link state changes occur with frequency f_k given by (5). The size of the link state update packet is $\Theta(1)$ as the number of level-*k* neighbors for the originating node is also $\Theta(1)$ by (3). Combining c_{k+1} with (3), (5), $\alpha_k = \Theta(1)$, and summing over all $L = \Theta(\log N)$ levels in the cluster hierarchy yields Φ_{FLOOD}:

$$\Phi_{\mathrm{FLOOD}} = \sum_{k=0}^{L-1} c_{k+1} \cdot d_k \cdot f_k = \sum_{k=0}^{L-1} \alpha_{k+1} \cdot c_k \cdot d_k \cdot f_k, \tag{22a}$$

$$\Rightarrow \Phi_{\mathrm{FLOOD}} = \sum_{k=0}^{L-1} \Theta(c_k \cdot f_k) = \sum_{k=0}^{L-1} \Theta(N/\sqrt{c_k}), \tag{22b}$$

$$\Rightarrow \Phi_{\mathrm{FLOOD}} = \mathrm{O}(N \cdot \log N). \tag{22c}$$

Dividing (22c) by N yields $\phi_{\mathrm{FLOOD}} = \mathrm{O}(\log N)$ as per Claim 4.

5.5 Location Registration

The communication overhead due to location registration (LR) for hierarchical location management (LM) is now assessed. LR at a level-*k* server occurs whenever a node migrates from one level-*k* cluster to another. Whenever a level-*k* location update is triggered, a single datagram of size $\Theta(1)$ is sent to a level-*k* server.

Considering level-0 clusters, these are just the individual nodes themselves and, obviously, no LR is required. Considering level-1 clusters, no LR is required as the local topology is flooded within each level-1 cluster. Considering now LM within level-*k* clusters ($k > 1$), LR is required whenever a node changes level-(*k*-1) clusters. Recalling $f_{\mathrm{MIG},k}$ from Section 5.3, the LR overhead incurred by nodes migrating between level-*k* clusters ($\Phi_{\mathrm{REG},k}$) is as follows:

$$\Phi_{\mathrm{REG},k} = N \cdot f_{\mathrm{MIG},k} \cdot h_{k+1}, \qquad k \in \{1, 2, \ldots, L-1\}. \tag{23}$$

The level-(*k*+1) hop distance (h_{k+1}) appearing in (23) is due to the fact that, when a node migrates to a new level-*k* cluster, it must notify its level-*k* LM server situated somewhere within its level-(*k*+1) cluster. The baseline average hop cost (when $k = 1$), therefore, corresponds to $h_2 = \Theta(1)$, the average hop distance between nodes in a level-2 cluster. Summing (23) over all k and applying (2) and (20) yields the aggregate registration overhead:

$$\Phi_{\mathrm{REG}} = N \cdot \sum_{k=1}^{L-1} f_{\mathrm{MIG},k} \cdot h_{k+1} = N \cdot \sum_{k=1}^{L-1} f_{\mathrm{MIG},k} \cdot \sqrt{\alpha_{k+1}} \cdot h_k \tag{24a}$$

$$\Rightarrow \Phi_{\mathrm{REG}} = N \cdot \sum_{k=1}^{L-1} \Theta(1) = \Theta(N \cdot \log N). \tag{24b}$$

Dividing (24b) by N yields LR overhead of $\phi_{\mathrm{REG}} = \Theta(\log N)$ as per Claim 5.

5.6 Location Management Handoff

There are three factors that contribute to location management (LM) handoff overhead. First, in a completely distributed hierarchical LM system, every node maintains LM information for, on average, $\Theta(\log N)$ other nodes. Thus, whenever a node participates in LM handoff, it must generate $\Theta(\log N)$ datagrams to complete the handoff.

Second, when a level-*k* cluster link state change occurs that triggers LM handoff for a level-*k* cluster, the $\Theta(\log N)$ LM datagrams that are generated must be forwarded across a level-(*k*+1) cluster. Thus, on average, each datagram due to level-*k* handoff must be forwarded across h_{k+1}-hop paths, $h_{k+1} = \Theta(\sqrt{\alpha_{k+1} \cdot h_k}) = \Theta\left(\sqrt{h_k}\right) = \Theta\left(\sqrt{c_k}\right)$.

Third, when a level-*k* cluster link state change triggers LM handoff for a level-*k* cluster, each level-0 cluster member must transfer/receive $\Theta(\log N)$ LM entries to/from the appropriate nodes of the level-(*k*+1) cluster it just exited or joined. That is, on average, c_k nodes must transfer $\Theta(\log N)$ LM entries across $\Theta\left(\sqrt{c_k}\right)$-hop paths.

Combining the three factors contributing to level-*k* LM handoff, it is evident that the overhead per level-*k* LM handoff trigger is $\Theta\left(c_k^{3/2} \cdot \log N\right)$. Defining $f_{\mathrm{HANDOFF},k}$ as the frequency of level-*k* LM handoff triggers, the overhead due level-*k* LM handoff is clearly $\Theta\left(f_{\mathrm{HANDOFF},k} \cdot c_k^{3/2} \cdot \log N\right)$ and the aggregate LM handoff overhead is given as follows:

$$\Phi_{\mathrm{HANDOFF}} = \sum_{k=0}^{L-1} \Theta(f_{\mathrm{HANDOFF},k} \cdot c_k^{3/2} \cdot \log N). \tag{25}$$

What remains to be considered is the frequency of level-*k* LM handoff triggers. Clearly, $f_{\mathrm{HANDOFF},k} < f_k$ as the frequency of level-*k* LM handoff triggers cannot exceed the level-*k* cluster link state change frequency. Further, $f_{\text{HANDOFF},k} = \Theta(f_k)$ because a nonnegligible fraction of level-*k* link state changes will trigger level-*k* handoff. Applying this fact to (25) yields:

$$\Phi_{\text{HANDOFF}} = \Theta\left(\sum_{k=0}^{L-1} f_k \cdot c_k^{3/2} \cdot \log N\right). \tag{26}$$

Applying (5) to (26) yields:

$$\Phi_{\text{HANDOFF}} = \Theta\left(\sum_{k=0}^{L-1} \frac{N}{c_k^{3/2}} \cdot c_k^{3/2} \cdot \log N\right), \tag{27a}$$

$$\Rightarrow \Phi_{\text{HANDOFF}} = \Theta\big(N \cdot \log^2 N\big). \tag{27b}$$

Last, dividing Φ_{HANDOFF} by N yields the per node LM handoff overhead $\phi_{\text{HANDOFF}} = \Theta\big(\log^2 N\big)$ as per Claim 6.

5.7 Location Queries

Each node, at some average frequency $f_{\text{QRY}} = \Theta(1)$ per node, initiates location queries. The overhead due to a location query (LQ) depends on the hop count to the highest level LM server in the LM hierarchy that must be queried. Further, ϕ_{QRY} depends on the prevailing communication pattern of the network. For example, a possible communication pattern is one where each node is equally likely to communicate with any other node in the network. In this case, the probability of success in querying a level-*k* server is proportional to the number of nodes in a level-(*k*+1) cluster and the average probability is of the form c_{k+1}/N, $c_L = N$. Clearly, with large probability, an LQ will be forwarded to the highest-level LM server, i.e., level-(*L*-1). Thus, each LQ will typically be forwarded over a h_L-hop path. Since $h_L = \Theta\big(\sqrt{N}\big)$ and $f_{\text{QRY}} = \Theta(1)$, $\phi_{\text{QRY}} = \Theta\big(\sqrt{N}\big)$ when nodes are equally likely to initiate communication sessions with any other node.

On the other hand, a different communication pattern can yield a very different ϕ_{QRY}. For example, a scenario is considered now where nodes are more likely to communicate with a peer if the peer is situated "nearby." Defining s_k as the frequency of communication sessions where the highest-level query reaches a level-*k* server, if $s_k = 1/h_k$, then the aggregate level-*k* query overhead is $\Theta(N \cdot h_k \cdot s_k) = \Theta(N)$. Summing over all L levels of the LM hierarchy yields an aggregate query overhead $\Phi_{\text{QRY}} = \Theta(N \cdot \log N)$ and a per node overhead of $\phi_{\text{QRY}} = \Theta(\log N)$.

In summary, $\phi_{\text{QRY}} = h_{\text{QRY}}$, where h_{QRY} is the average hop distance of the query path to the highest level LM server that must be queried. Since h_{QRY} is within a scaling constant of h (i.e., $h_{\text{QRY}} = \Theta(h)$), $\phi_{\text{QRY}} = \Theta(h)$ as per Claim 7.

5.8 Hierarchical Addressing

To facilitate unicast packet forwarding via hierarchical routing, each datagram header must contain the hierarchical address of the target node t. The hierarchical address consists of the concatenation of the cluster IDs of the clusters to which t belongs as well as the ID of t, itself. Thus, the hierarchical address consists of $L \cdot B$ bits, where B is the number of bits in a node ID. Further, the length of the node ID itself is $\Theta(\log N)$. (This is also true for address-based nonhierarchical routing protocols.) In such an implementation, $\psi_{\text{UNICAST-HEADER}} = \Theta\big(\log^2 N\big)$ bits of overhead are added to the header of every unicast datagram of which, $\Theta(\log N)$ is due to hierarchy.

However, the overhead due to concatenated hierarchical addresses does not contribute to control overhead. This is because all control messaging due to hierarchical routing may be implemented via anycast or broadcast (for flooding). This incurs greater hop-by-hop processing at each node, but it eliminates the presence of a hierarchical address in the datagram header. Thus, for control messaging, $\psi_{\text{CTRL-HEADER}} = \Theta(\log N)$ bits (for the node ID) as per Claim 8.

6 Conclusions

This paper has assessed the communication overhead due to hierarchical routing in MANETs. The assessment considered overhead related to the creation, maintenance, and dissemination of hierarchical routing tables as well as the overhead related to the registration, query, and handoff of location management (i.e., address management) data. Considering the control overhead factors that are invariant with respect to the prevailing network traffic pattern (i.e., disregarding $\phi_{\text{QRY}} = \Theta(h)$), control packet transmission count is dominated by LM handoff events and, thus, $\phi = \phi_{\text{HANDOFF}} = \Theta\big(\log^2 N\big)$ packet transmissions per second per node.

The significance of deriving a result for ϕ is that it quantifies the scalability of hierarchical routing in MANETs. One application of this figure is that it allows network designers to specify the link capacity required for the transceivers at each node in order to accommodate the control overhead due to hierarchical routing. Given the polylogarithmic result derived here, it is evident that transceiver link capacity (C) must be polylogarithmic in the node count, i.e., combining ϕ and $\psi_{\text{CTRL-HEADER}}$ means $\psi = \Theta\big(\log^3 N\big) \rightarrow C > \psi \Leftrightarrow C = \Omega\big(\log^3 N\big)$.

A second application of the result derived here is that it affords a quantitative comparison with other routing architectures. For example, a nonhierarchical (i.e., flat) implementation of link state routing generates packet transmission overhead per node that is linear in the node count as well as $\Theta(\log N)$ bits for the node ID in each datagram header. Comparing ψ with the overhead of flat link state routing shows hierarchical routing to have a scalability advantage over flat routing by a factor of $\Theta\big(N/\log^2 N\big)$, relaxing the requirement on C by a similar factor.

A third application of this work is that it identifies the type of communication patterns for which hierarchical routing will be most applicable. Clearly, given

$$\psi_{\text{UNICAST-HEADER}} = \Theta\big(\log^2 N\big),$$

the benefit of $\psi = \Theta\big(\log^3 N\big) \Rightarrow C = \Omega\big(\log^3 N\big)$ afforded by hierarchical routing can be exploited only if $h = \mathrm{O}(\log N)$. Design of hierarchical networks, therefore, should also consider structuring communication patterns to exploit the hierarchical routing architecture (i.e., exploit *spatial locality*). Traffic patterns that yield small h (and, similarly, small h_{QRY} and ϕ_{QRY}) may be realized by nodes that have inherent hierarchical affiliations with one another. Such affiliations may be common in battlefield networks. For example, the organization of infantry into squads, platoons, companies, etc. may naturally yield beneficial hierarchical communications.

APPENDIX

Proof of Lemma 1. A level-j cluster link state change that propagates up the hierarchical tree due to recursive cluster creation/deletion remains confined to an area of the network consisting of $\Theta(1)$ level-k nodes, $k \geq j$. This is due to the nature of the ALCA. That is, at any level-j of the hierarchical tree, a cluster creation/deletion event impacts only clusters within a $\Theta(1)$-hop radius about the endpoints of the affected link and does not propagate throughout the level-j topology. Thus, a level-j cluster link state change ($j < k$) affects at worst $\Theta(1)$ level-k links. □

Proof of Lemma 2. This lemma is based on the principle of *dynamic steady state*. That is, since, over time, the average number of level-k clusters converges to a steady state limit, the average frequency of level-k cluster creations, and the average frequency of level-k cluster deletions must be equal. The same principle also applies to cluster link state changes such that the average frequency of cluster link creations and the average frequency of cluster link deletions are equal.

Again, level-k cluster link state changes may be due to level-k cluster migration (type E-1) or due to recursive cluster creation/deletion (type E-2) incurred by level-j cluster migration ($j < k$). Isolating the level-k cluster link state changes due to E-1, it is noted that a type E-1 event is equally likely to create a level-k link as it is to delete a level-k link (i.e., the principle of dynamic steady state applies here, as well). Further, it is noted that the frequency of level-k link deletions is the sum of the average frequencies of link deletions due to type E-1 and type E-2 events. Combining these two observations with the fact that the average frequency of level-k cluster deletions is equal to the average frequency of level-k cluster creations (due to the principle of dynamic steady state), it is evident that the average frequencies of level-k cluster link deletions and level-k cluster link creations due to type E-2 events must also be equal. □

Proof of Lemma 3. An arbitrary level-k cluster x is a critical level-k cluster (i.e., only a single level-(k-1) cluster elected x to be its clusterhead) with probability σ_k. If x is a critical level-k cluster, then x is *not* a level-k clusterhead. This is because, if x is a level-k critical node, then there must be a subset of level-k neighbors of x that can serve as the level-k clusterhead set for the closed level-k neighborhood about x. By the same reasoning, deletion of x from V_k, by itself, would not create a new level-k clusterhead. Thus, the only the possibility for the deletion of x from V_k to affect V_{k+1} is if the level-k clusterhead elected by x, say y, loses its level-k clusterhead status (i.e., loses its level-(k+1) membership). Deletion of y from V_{k+1} occurs only if y itself is a critical level-(k+1) cluster and this is the case with probability σ_{k+1}.

Three conclusions summarize the discussion here. One, a level-j node migration event, which triggers the deletion of x from V_{j+1}, impacts level-k cluster link status ($j < k$) only by recursive cluster deletion, but not by some combination of both cluster creation and cluster deletion events. Two, each level-i cluster deletion trigger impacts level-(i+1) in the hierarchy with probability σ_{i+1}. Three, the cluster deletion at level-i is an event that occurs independent of cluster deletion events at other levels. Combining these conclusions proves the lemma.□

Proof of Lemma 4. To prove that $\sigma_k < 1/\sqrt{\alpha_k}$, an arbitrary level-($k$-1) clusterhead x is considered. Since x is a level-k cluster, there must be at least one level-(k-1) cluster, say v, for which the ID of x is the largest among all nodes within one (level-(k-1)) hop of v. If v is the only level-(k-1) neighbor of x for which this is true, then x is a critical node. On the other hand, if there is at least one other neighbor of x that has elected x as its clusterhead, then x is not a critical node. For example, the nearest neighbor of v, say u, is also likely a neighbor of x. Since the distance r separating u from v is likely to be relatively small, u and v share a large number of common neighbors. Thus, if v has elected x as its clusterhead, then with high likelihood, so has u. The probability that u has *not* elected x as its clusterhead is assessed to upper bound σ_k.

Since the number of nodes within a given area is described by a Poisson random variable with parameter γ equal to the product of the area and the node density, the expected distance from v to its nearest neighbor u is $r = \sqrt{1/(\pi \cdot d_{k-1})}$. Here, d_{k-1} also corresponds to the average number of level-(k-1) nodes within one unit area of a node v. This is consistent with the fact that the radius of a circular area for $\gamma = 1$ is $r = \sqrt{1/(\pi \cdot d_{k-1})}$. Scaling $r \rightarrow r' = \sqrt{1/d_{k-1}}$ so that area is computed with respect to a radius of 1, the scaled area (A') of the region outside the overlapping neighborhoods of u and v is given as follows:

$$A' = 4 \cdot \int_{r'/2}^{1} \sqrt{1 - z^2} dz, \tag{A-1a}$$

$$\Rightarrow A' = \pi - r' \cdot \sqrt{1 - r'^2/4} - 2 \cdot \sin^{-1}(r'/2). \tag{A-1b}$$

Noting that, for small r', $\sqrt{1 - r'^2/4} \approx 1$ and

$$\sin^{-1}(r'/2) \approx r'/2,$$

(A-1b) may be simplified:

$$A' = \pi - 2 \cdot r', \tag{A-2a}$$

$$\Rightarrow A' = \pi - 2 \cdot \sqrt{\frac{1}{d_{k-1}}}. \tag{A-2b}$$

Normalizing the area of (A-2b) to be a portion of a unit area circle yields the fraction (A) of the neighborhood of u that overlaps the neighborhood area of v:

$$A = A'/\pi = 1 - \frac{2}{\pi \cdot \sqrt{d_{k-1}}}. \quad \text{(A-3)}$$

Applying (A-3) and recalling that the expected number of nodes in a given area is equal to the product of the area and node density yields a Poisson parameter λ for the number of nodes that are neighbors of u but are not neighbors of v:

$$\lambda = d_{k-1} \cdot (1 - A) = \frac{2 \cdot \sqrt{d_{k-1}}}{\pi}. \quad \text{(A-4)}$$

Letting ω_k be the probability that no node in the region with area $1 - A$ has a node ID greater than x (the clusterhead of v), the probability that x is a level-*k* critical node is upper bounded as follows:

$$\sigma_k \leq 1 - \omega_k. \quad \text{(A-5)}$$

The expected fraction (ξ) of level-(*k*-1) clusters for which the node ID of x is larger can be easily underbounded by the following:

$$\xi \geq \frac{\alpha_k - 1}{\alpha_k}. \quad \text{(A-6)}$$

Noting that the number of nodes in a region of area $1 - A$ is described by a Poisson random variable with parameter λ given by (A-4) and that the probability of n nodes having node IDs smaller than x is ξ^n, ω_k may be approximated as follows:

$$\omega_k \geq \sum_{n=0}^{d_{k-1}} e^{-\lambda} \cdot \frac{(\lambda \cdot \xi)^n}{n!}, \quad \text{(A-7a)}$$

$$\Rightarrow \omega_k \geq e^{-\lambda \cdot (1-\xi)} \cdot \sum_{n=0}^{d_{k-1}} e^{-\lambda \cdot \xi} \cdot \frac{(\lambda \cdot \xi)^n}{n!}, \quad \text{(A-7b)}$$

$$\Rightarrow \omega_k \geq e^{-\lambda \cdot (1-\xi)} \quad \text{(A-7c)}$$

$$\Rightarrow \omega_k \geq e^{-\frac{2 \cdot \sqrt{d_{k-1}}}{\pi \cdot \alpha_k}} \approx 1 - \frac{2 \cdot \sqrt{d_{k-1}}}{\pi \cdot \alpha_k}. \quad \text{(A-7d)}$$

Applying (A-7d) to (A-5) yields:

$$\sigma_k \leq \frac{2 \cdot \sqrt{d_{k-1}}}{\pi \cdot \alpha_k}. \quad \text{(A-8)}$$

Comparing with the lemma statement, $\sigma_k < 1/\sqrt{\alpha_k}$, it is evident that the lemma is true if $\alpha_k > 4 \cdot d_{k-1}/\pi^2$. Simulation results verify that this condition consistently holds for $d_{k-1} \leq 15$. □

ACKNOWLEDGMENTS

This work was supported in part by US Army CECOM research grants DAAB07-00-D-G505 and DAAB07-02-C-P301.

REFERENCES

[1] L. Kleinrock and F. Kamoun, "Hierarchical Routing for Large Networks," *Computer Networks*, vol. 1, no. 3, pp. 155-174, 1977.
[2] J. Westcott and G. Lauer, "Hierarchical Routing for Very Large Networks," *Proc. IEEE Military Comm. Conf.*, pp. 214-218, Oct. 1984.
[3] G.S. Lauer, "Hierarchical Routing Design for SURAN," *Proc. IEEE Int'l Conf. Comm.*, pp. 93-102, June 1986.
[4] G. Pei, M. Gerla, X. Hong, and C.-C. Chiang, "A Wireless Hierarchical Routing Protocol with Group Mobility," *Proc. IEEE Wireless Comm. Networking Conf.*, pp. 1538-1542, Sept. 1999.
[5] A. Iwata, C.-C. Chiang, G. Pei, M. Gerla, and T.-W. Chen, "Scalable Routing Strategies for Ad Hoc Wireless Networks," *IEEE J. Selected Areas in Comm.*, pp. 1369-1379, Aug. 1999.
[6] R. Ramanathan and M. Steenstrup, "Hierarchically-Organized, Multihop Mobile Wireless Networks for Quality-of-Service Support," *ACM/Baltzer Mobile Networks and Applications*, vol. 3, pp. 101-119, 1998.
[7] C.V. Rammamoorthy and W.-T. Tsai, "An Adaptive Hierarchical Routing Algorithm," *Proc. IEEE CSAC*, pp. 93-104, Fall 1983.
[8] J.J. Garcia Luna Aceves and N. Shacham, "Analysis of Routing Strategies for Packet Radio Networks," *Proc. IEEE Infocom*, pp. 292-302, 1985.
[9] J. Broch, D.A. Maltz, D.B. Johnson, Y.-C. Hu, and J. Jetcheva, "A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols," *Proc. Mobicom '98*, pp. 85-97, 1998.
[10] L. Kleinrock and J. Sylvester, "Optimum Transmission Radii for Packet Radio Networks or Why Six Is a Magic Number," *Proc. IEEE Nat'l Telecomm. Conf.*, pp. 4.3.1-4.3.5, Dec. 1978.
[11] A.D. Amis, R. Prakash, T.H.P. Vuong, and D.T. Huynh, "Max-Min D-Cluster Formation in Wireless Ad Hoc Networks," *Proc. IEEE Infocom*, pp. 32-41, 2000.
[12] D.J. Baker and A. Ephremides, "The Architectural Organization of a Mobile Radio Network via a Distributed Algorithm," *IEEE Trans. Comm.*, pp. 1694-1701, Nov. 1981.
[13] D.J. Baker, A. Ephremides, and J.A. Flynn, "The Design and Simulation of a Mobile Radio Network with Distributed Control," *IEEE J. Selected Areas Comm.*, pp. 226-237, Jan. 1984.
[14] C.R. Lin and M. Gerla, "Adaptive Clustering for Mobile Wireless Networks," *IEEE J. Selected Areas Comm.*, vol. 15, no. 7, pp. 1265-1275, Sept. 1997.
[15] M. Steenstrup, "Cluster-Based Networks," *Ad Hoc Networking*, C.E. Perkins, ed., chapter 4, pp. 75-138, Addison-Wesley, 2001.
[16] J. Li, J. Jannotti, D. De Couto, D. Karger, and R. Morris, "A Scalable Location Service for Geographic Ad-Hoc Routing," *Proc. MobiCom 2000*, pp. 120-130, Aug. 2000.
[17] R. Hinden and S. Deering, "IP Version 6 Addressing Architecture," IETF RFC, no. 2373, July 1998.

John Sucec received the PhD degree in electrical and computer engineering from Rutgers University, New Brunswick, New Jersey, in 2003. Previously, he studied at the University of Connecticut (BS 1990 and MS 1992, both in electrical engineering) and Binghamton University (MS 1996, computer science). His research interests include routing in IP-based networks and wireless network communication protocols. He is a member of IEEE.

Ivan Marsic received the BS and MS degrees in computer engineering from the University of Zagreb, Croatia, in 1982 and 1987, respectively, and PhD degree in biomedical engineering from Rutgers University, New Brunswick, New Jersey, in 1994. He is an associate professor of electrical and computer engineering at Rutgers University. His research interests are in mobile computing and collaboration, groupware, distributed computing, and human-computer interfaces. His research is supported by grants from the US National Science Foundation, Department of Defense, and industry. He has authored more than 100 journal and conference papers, one book, and three book chapters. He is a member of the IEEE, ACM, IFIP WG2.7/13.4, and International Academy for Advanced Decision Support and has been a consultant to industry and government.