Center for Communications and Digital Signal Processing
(CDSP)

TECHNICAL REPORT

TR-CDSP-00-52

Asymptotic Behavior of Mobile Ad Hoc Routing Protocols with respect to Traffic, Mobility and Size.

Cesar Santiváñez

Department of Electrical & Computer Engineering
Northeastern University, Boston, MA 02115.

(email: cesar@ece.neu.edu)

October, 2000

Asymptotic Behavior of Mobile Ad Hoc Routing Protocols with respect to Traffic, Mobility and Size.

César Santiváñez
Department of Electrical & Computer Engineering
Northeastern University
Boston, MA 02115, USA

Abstract

Mobile Ad hoc NETworks (MANET) are a result of an association of mobile nodes in an environment lacking supporting infrastructure. MANET networks are usually limited in terms of bandwidth, energy, processing capabilities, and security (among others). However, recent technological developments regarding a node transmission rate, battery life, and processing power have motivated an increasing interest in ad hoc networks. Thus, it is of paramount importance to understand the fundamental limits on the performance of these networks.

This report presents results regarding the asymptotic bandwidth consumption required for several known ad hoc network protocols : Plain Flooding (PF), Standard Link State (SLS), Dynamic Source Routing (DSR), Hierarchical Link State (HierLS), Zone Routing Protocol (ZRP); as well as results for a novel protocol fruit of DARPA's Density and Asymmetry-adaptive Wireless Network (DAWN) project, namely the Hazy Sighted Link State (HSLS) protocol. The emphasis in bandwidth consumption is due to the author's believe that bandwidth is likely to be the most limiting factor on a MANET network performance.

The results presented in this report will help network designers to better identify the class of protocols to engage depending on their operation scenario.

1 Introduction

A Mobile Ad hoc NETwork (MANET) may be envisioned as a quickly deployable association of mobile nodes in an environment lacking supporting infrastructure. MANET networks appear in military and disaster recovery operation, or in a freely association of users as for example in a conference room or in a university campus.

MANET networks are usually limited in terms of bandwidth, energy, processing capabilities, and security (among others). However, recent technological developments regarding a node transmission rate, battery life, and processing power have motivated an increasing interest in ad hoc networks. Thus, it is of paramount importance to understand the fundamental limits on the performance of these networks.

This report presents results regarding the asymptotic behavior of several known ad hoc network protocols : Plain Flooding (PF), Standard Link State (SLS), Dynamic Source Routing (DSR) [1], Hierarchical Link State (HierLS) [2], Zone Routing Protocol (ZRP) [3]; as well as results for a novel protocol fruit of DARPA's Density and Asymmetry-adaptive Wireless Network (DAWN) project [5], namely the Hazy Sighted Link State (HSLS) protocol.

This work focuses on the bandwidth consumption required to successfully route packets from the sources to the destinations. This is done because it is the author opinion that bandwidth is likely to be the most limiting factor on a MANET network performance. Obviously, this is not the case for all the possible networks (for example, sensor networks may be primarily limited by the processing power of the nodes).

The remaining of this section presents a new definition of *total overhead* that allows for comparison among different protocols. Roughly speaking, the lower the *total overhead* the greater the available capacity for data forwarding. Thus, in a bandwidth constrained network, *total overhead* may be the single most important performance figure. However, it is not the only one as delay, energy consumption, memory and processing requirement, may each has even greater importance depending on the particular application. This work, however, only focuses in analyzing the *total overhead* required for different protocols.

Section 2 presents the assumptions made in our analysis. These assumptions restricted our attention to a (broad) class of networks. Section 3 analyzes the asymptotic behavior of the *total overhead* induced by the Plain Flooding (PF) algorithm. Section 4 presents similar results for the Standard Link State (SLS) protocol. Section 5 presents an asymptotic lower bound for DSR's *total overhead*. Sections 6 and 7 analyze HierLS and ZRP respectively, and Section 8 briefly describes and analyzes the Hazy Sighted Link State (HSLS) protocol. Section 9 summarizes these results and finishes this report with some final remarks.

1.1 Scalability and Total overhead

Traditionally, the term *overhead* has been used to describe only the amount of bandwidth employed in constructing a route towards a destination. Thus, in proactive approaches *overhead* has been equated to the number of packets being exchanged between nodes, in order to maintain the node's forwarding tables up-to-date. In reactive approaches, *overhead* has been equated to the bandwidth consumed by the route request/reply messages (global or local). It has been a main objective in the analysis/design of protocols to keep the aforementioned value of *overhead* low.

While it is true that the above mentioned sources of overhead are of crucial importance in a protocol behavior, they do not provide enough information to do a proper performance assessment of a given protocol since they fail to weight the impact of sub-optimal routes in the protocol's performance. As the network size increases above 100 nodes, keeping route optimality imposes an unacceptable cost in both proactive and reactive approaches, and sub-optimal routes become a fact of life in any scalable routing protocol. Sub-optimal routes are introduced in reactive protocols since they try to maintain the current source-destination path as long as it is valid, although it may no longer be optimal. Also, local repair techniques try to reduce the *overhead* induced by the protocol at the expense of longer, non optimal paths. Proactive approaches introduce sub-optimal routes by limiting the scope of topology information dissemination (e.g. hierarchical routing) and/or limiting the time between successive topology information updates dissemination so that topology updates are no longer instantaneously event-driven (e.g GSR [6]).

Thus, it is necessary to revise the concept of *overhead* so that it weights the effect of sub-optimal routes in capacity scarce systems, that is, **sub-optimal routes not only increase the end-to-end delay but also incur in a greater bandwidth usage than required**. This extra bandwidth may be comparable to the other sources of overhead. Approaches that attempt to minimize only the other sources of overhead may create the illusion that they are *scalable* by inducing a fixed amount of the aforementioned overhead, but in practice they may cause severe performance degradation as the extra overhead induced by sub-optimal routes increases.

Thus, the real definition of overhead – which will be assumed in the remainder of this report – is that **total overhead is the total amount of bandwidth used in excess to the minimum amount of bandwidth required to forward data through shortest distance (in the bandwidth sense) paths as if the nodes had instantaneous full-topology information**. The above definition then includes the proactive information exchange between nodes, the bandwidth consumed by route request/reply in reactive approach, the extra hops in hierarchical proactive approaches (e.g. using a 4 hop path when there is a 3 hop path available), as well as the extra bandwidth consumed when flooding a data packet through the entire network instead of using only the shortest path.

The above definition of *total overhead* provides a comparison ground among otherwise very different protocols; but it is not intended to be the *only* criteria of comparison among protocols, as different applications may impose additional criteria such as, for example, end-to-end delay, or jitter. However, in the case of a bandwidth-limited network, and as the size and mobility of the network increase, the remaining capacity (i.e. the initial network capacity minus the *total overhead*) may well be the single most important performance metric because it defines the achievable throughput.

The different sources that add up the *total overhead* may be grouped as *reactive*, *proactive*, and *sub-optimal route* cost.

Reactive cost is the amount of bandwidth consumed to build paths to any destination, given that there is a traffic flow into that destination. In static networks, reactive cost is a function of the rate of generation of new flows. In dynamic (mobile) networks, however, paths are (re)built not only due to new flows but also due to link failures in an already active path. Thus, in general, the reactive cost is a function of traffic and topology change.

Proactive cost is the amount of bandwidth consumed by protocols to propagate route information *before* it is needed (whether periodically or as a response to topological changes). In other words, *proactive* cost is the amount of bandwidth consumed (to propagate/build routes) that has not been considered in the *reactive* cost computation.

Sub-optimal route cost is the difference between the bandwidth consumed when transmitting data from all the sources to their destinations and the bandwidth that would have been consumed should the data have followed the shortest available path(s). For example, consider a source that is 3 hops away from its destination. If a protocol chooses to deliver one packet following a k ($k > 3$) hop path (maybe because of out-of-date information, or because the source has not yet learn of the availability of a 3 hop path), then we will have to add $(k-3) * packet\_length$ to the *sub-optimal route* cost. Similarly, if Plain Flooding (PF) is employed in a network of size N – requiring a packet to be retransmitted N times (one per node) – then we will have to add $(N-3) * packet\_length$ bits to the *sub-optimal route* cost per each packet transmitted.

The above classification of the different sources of *total overhead* will prove useful in deriving the *total overhead* expressions in the next sections. However, we need first to introduce the set of assumptions adopted in this work.

2 Network model and assumptions

Let N be the number of nodes in the network, d be the average in-degree (i.e. the average number of incoming links into a node), L be the average length of a source destination pair, λ_{lc} be the expected number of link status changes that a node detects in a second, λ_t be the average traffic that a node generates in a second, and λ_s be the average number of new sessions generated by a node in a second (clearly, $\lambda_s \leq \lambda_t$).

In order to analyze the asymptotic behavior of quantities that depend on different parameters, some assumptions are made. [1]

A.1 As network size increases, the average in-degree d remains constant.

A.2 Let A be the area covered by the N nodes of the network, then the average area covered by a set of n nodes increases as $\Theta(\sigma * n)$, where $\sigma = A/N$ is the network average density.

A.3 The number of nodes that are at distance of k or less hops away from a source node increases (on average) as $\Theta(d * k^2)$. The number of nodes exactly at k hops away increases as $\Theta(d * k)$.

A.4 The maximum and average paths among nodes in a connected subset of n nodes both increase as $\Theta(\sqrt{n})$. In particular, the maximum path across the whole network and the average path across the network (L) increases as $\Theta(\sqrt{N})$.

A.5 The traffic that a node generates in a second (λ_t), is independent of the network size N (number of possible destinations). As the network size increases, the total amount of data transmitted/received by a single node will remain constant but the number of destinations will increase (the destinations diversity will increase). Similarly, λ_s is also independent of the network size N.

A.6 For a given source node, all possible destination ($N - 1$ nodes) are equiprobable and as a consequence the traffic from one node to a particular destination decreases as $\Theta(1/N)$.

A.7 Unless otherwise stated, it is assumed that the cause of link status change is node mobility. λ_{lc} is directly proportional to the node speed.

A.8 Mobility models. Let $f_{1/0}(x, y)$ be the probability distribution function of a node position at time 1 second, given that the node was at the origin $(0, 0)$ at time 0. Then, the probability distribution function of a node position at time t given that the node was at the position (x_{t_0}, y_{t_0}) at time t_0 is given by $f_{t/t_0}(x, y) = \frac{1}{(t-t_0)^2} f_{1/0}(\frac{x-x_{t_0}}{t-t_0}, \frac{y-y_{t_0}}{t-t_0})$. Similarly, let $g_{0/1}(x, y)$ be the probability distribution function of a node position at time 0 second, given that it is known that the node position at time 1 will be $(0, 0)$. Then, the probability distribution function of a node position at time $t < t_1$ given that the node will be at the position (x_{t_1}, y_{t_1}) at time t_1 is given by $g_{t/t_1}(x, y) = \frac{1}{(t-t_0)^2} g_{0/1}(\frac{x-x_{t_1}}{t_1-t}, \frac{y-y_{t_1}}{t_1-t})$.

The first assumption follows since imposing a fixed degree in a network is both desirable and achievable. It is desirable, because allowing the density to increase without bound will jeopardize the achievable network throughput[2]. It is achievable, because there are effective power control

[1] Standard asymptotic notation is employed. A function $f(n) = \Omega(g(n))$ [similarly, $f(n) = O(g(n))$] if there exists constants c_1 and n_1 [similarly, c_2 and n_2] such that $c_1 * g(n) \le f(n)$ [similarly $f(n) \le c_2 * g(n)$] for all $n \ge n_1$ [similarly, $n \ge n_2$]. Also, $f(n) = \Theta(g(n))$ if and only if $f(n) = \Omega(g(n))$, and $f(n) = O(g(n))$.

[2] See the topology (power) control results in DAWN [5]

mechanism currently available (see for example [4]). In general, a topology control algorithm should will try to make the density as small as possible without compromising connectivity.

The first assumption drives us apart of random network models where the probability of a link between 2 nodes is fixed to a value p. In those networks, as size increases, the average node degree ($N * p$) goes to infinity and the shortest path among the 2 nodes furthest apart approach 2. We do not believe that such networks will be representative of future mobile ad hoc networks.

The second assumption is motivated by the observation that at large scales (large number of nodes), one expects to see some uniformity. For example, it is natural to assume that half the area covered by the network contains around one half of the nodes in the network.

Assumptions A.3 and A.4 are based on assumption A.2. For example, if we consider a circular area centered at node S, of radio R with n nodes in it; we see that doubling the area radius ($2R$) will quadruple the area, and therefore quadruple the number of nodes inside the area. On the other hand, the distance (in meters) from S to the furthermost nodes will have only double, and assuming that the transmission range (after power control) of the nodes do not change, then the distance (in hops) will also only double (in average). Similarly, the 'boundary' area (where the nodes furthermost apart from S are) will increase linearly (as the circumference of a circle does) with the radius.

Assumption A.5 and A.6 follow from behavior observed in the telephone and internet networks, that is, as the network size increases (i.e. networks interconnect) the total amount of traffic required by a user does not increase but only diversifies. For example, availability of cheap long distance service all over the world allows a home user to talk with all their family members and friends (wherever they are) but does not increase the time the user has to spare for personal phone calls. Similarly, with the increase in size and content of the internet, a user may found more web pages he/she will like to visit (destination set diversifies) but if the amount of bandwidth and time available for the user to connect to the internet is fixed, he/she will have to limit the total time (and therefore traffic) he/she spend on the internet.

The above statements are not final. For example, the increase in size and content on the internet may prompt users to stay more time on-line, or to perform more documents download operations, effectively increasing the incoming traffic to these destinations. Also, traffic sources involved in advertisement will try to send a fixed amount of information to all (or a constant fraction) of the users. For these 'advertisers' the total traffic injected into the network will not be constant but will increase with network size.

Thus, assumptions A.5 and A.6 are motivated by human users behavior, and other networks may violate these assumptions. For example, in sensor networks each node may want to transmit its sensed information either to all other nodes (causing λ_t to increase as $\Theta(N)$), or to a central node (causing the destination set to consist of only 1 node, violating assumption A.6).

The traffic assumption is of paramount importance in the analysis since it will determine the effect of sub-optimal routes in the network performance. For example, if almost all the traffic is limited to a locality of the source then hierarchical routing and ZRP will be greatly benefited. On the other hand, having a small set of destinations will favor DSR. In general, the most demanding scenario is when all destinations are equally probable, so we focus our analysis in this case.

Assumption A.7 stresses our interest in analyzing mobile networks. In particular, we assume that short-term variations on link quality may be buffered off by a effective link control mechanism. However, we note that this is not necessarily true, as radio behavior is quite unpredictable and long-lived link degradation is possible even if there is no mobility (e.g. due to obstructions, small displacement, rain, etc.).

Assumption A.8 is motivated by mobility models where the velocity of a mobile over time is highly correlated . For example, if the speed and direction are constant but unknown. Obviously, this assumption does not hold for a random walk model. However, a random walk model will induce smaller node displacements over time (randomness tends to cancel out) and as a consequence they impose a somewhat less demanding scenario for routing protocols. Once again, we decide to focus in the most demanding scenario, that is larger displacements, and assume that speed and direction are random processes with a slow decaying autocorrelation function; which justify assumption A.8.

The following sections present our results for *total overhead* for several current routing protocols.

3 Plain Flooding (PF)

In PF, each packet is (re)transmitted for every node in the network (except the destination). Thus, we need $N-1$ transmissions for each data packet, when the optimal value (on average) should have been L. Since there are $\lambda_t * N$ data packets being generated at each second, the extra bandwidth required for transmitting all this packets is $size\_of\_data * (N-1-L) * \lambda_t * N$ bps. And since $L = \Theta(\sqrt{N})$, then PF *sub-optimal route* cost per second is $\Theta(\lambda_t * (N^2 - N^{1.5})) = \Theta(\lambda_t * N^2)$.

PF does not try to find routes toward the destination, so it does not induce neither *reactive* nor *proactive* cost. Thus, PF *total overhead* per second is $\Theta(\lambda_t * N^2)$.

4 Standard Link State

In Standard Link State (SLS), a node sends a Link State Update (LSU) to the whole network each time it detects a link status change. A node also sends periodic, soft-state LSUs every T_p seconds.

SLS does not induce *reactive* cost, and since the paths it generates are optimal, it does not induce *sub-optimal route* cost either.

To compute SLS *proactive* cost, consider that each node generates a LSU at a rate of λ_{lc} per second, and this LSU is retransmitted at least once per each node (i.e. N times). Thus, each LSU generated a cost of $size\_of\_LSU * N$. Since there are $N * \lambda_{lc}$ LSUs being generated at any given second, then SLS proactive cost is $size\_of\_LSU * \lambda_{lc} * N^2$ bps.

Thus, SLS *total overhead* per second is $\Theta((k_1 + k_2 * d) * \lambda_{lc} * N^2)$, where we used the fact that the LSU size depends on the number of neighbors of a node (i.e. the node degree d), which is bounded independently of N (assumption A.1).

5 Dynamic Source Routing (DSR)

In Dynamic Source Routing (DSR) no proactive information is exchanged. When a node (source) needs to reach a destination, it floods a route request (RREQ) message into the network. When a RREQ message reaches the destination (or a node with a cached route towards the destination) a route reply message, including the newly found route, is sent back to the source. The source attaches the new route to the header of all subsequent packets to that destination, and any intermediate node along the route uses this attached information to learn the identity of the next hop in the route.

5.1 DSR without Route Cache (DSR-noRC)

DSR-noRC reactive cost must account for RREQ messages originated for new session requests (generated at a rate λ_s per second per node) and the RREQ messages generated by failures in links that are part of a path currently being used. If we only consider the RREQ messages originated for new session requests, then a lower bound can be obtained.

Each new route request is flooded to the whole network, which implies at least $N-1$ retransmissions (only the destination does not need to retransmit the route request). Thus, each route request cost $size\_of\_RREQ * (N-1)$ bits, and there are $\lambda_s * N$ RREQ messages being generated in a second due to new session requests. Thus, DSR-noRC reactive cost per second is $\Omega(\lambda_s * N^2)$ bits.

Note that an upper bound can also be obtained if we consider that in the worst case each link failure required a new global RREQ (i.e. no local repair possible). Then, the overhead induced by this repair RREQ will be $number\_repairs * size\_of\_RREQ * N$ bits, and considering that the number of repairs requested in a second is smaller than the number of link failures (over the entire network) in the same second ($\lambda_{lc} * N$), we conclude that the bandwidth cost induced by these repairs is lower than $\lambda_{lc} * N * size\_of\_RREQ * N$ bps. Thus, the cost induced by route repairs is $O(\lambda_{lc} * N^2)$, and the *reactive* cost of DSR is $O((\lambda_s + \lambda_{lc}) * N^2)$ bps.

For DSR *sub-optimal route* cost let's only consider the extra bandwidth required for appending the source-route into each data packet. Once again, a lower bound will be obtained. The number of bits appended to each data packet will be proportional to the length L_i of path i. This length will be equal to or larger than the length L_i^{opt} of the optimal path i^{opt}. Thus, we can use the optimal path values as a lower bound. Then, the extra bandwidth consumed by a packet delivered using a path i (with at least L_i^{opt} retransmissions) will be at least $(\log_2 N) * (L_i^{opt})^2$, where $\log_2 N$ is the minimum length of a node address. Then, the average extra bandwidth per packet over all paths is $E\{(\log_2 N) * (L_i^{opt})^2)\} \geq (\log_2 N) E\{L_i^{opt}\}^2 = (\log_2 N) L^2$ bits. Thus, for each packet going from source to destination there is at least (an average) *sub-optimal route* cost of $(\log_2 N)L^2$. Then, since there are $\lambda_t * N$ packets being transmitted at any given time (assumption A.5), the *sub-optimal route* cost induced over the whole network is at least $\lambda_t * N(\log_2 N)L^2$ bps. Thus, recalling that $L = \Theta(\sqrt{N})$ (assumption A.4), DSR-noRC *sub-optimal route* cost is $\Omega(\lambda_t * N^2 \log_2 N)$ bps.

Finally, DSR-noRC *total overhead* per second is $\Omega(\lambda_s * N^2 + \lambda_t * N^2 \log_2 N)$.

6 Hierarchical Link State (HierLS)

In *m-level* Hierarchical Link State (HierLS) routing, (level 1) nodes are grouped in level 1 clusters, level 1 clusters (level 2 nodes) are grouped in level 2 super-clusters, and so on up to the m level. So in general, level i nodes are grouped into level i cluster, which become level $i+1$ nodes. And so on, until the number of highest level nodes is below a threshold and therefore they can be grouped (conceptually) in a single level m. Thus, the value of m is determined dynamically based on the network size, topology, and threshold values.

Link state information inside a level i cluster is aggregated (limiting the rate of LSU generation) and transmitted only to other level i nodes belonging to the same level i cluster (limiting the scope of the LSU). Thus, individual link status change may not be sent outside the level 1 cluster (if they do not cause a significant change to higher levels aggregated information), greatly reducing the proactive overhead.

HierLS relies on another service, namely Location Management, to inform a source node S the address of the highest level cluster that contains the desired destination D but does not contain the source node S. For example, consider a 4-level network as shown in Figure 1. S and D are level 1 nodes; $X.1.1$, $X.1.2$, etc. are level 2 nodes (level 1 clusters); $X.1$, $X.2$, etc. are level 3 nodes (level 2 clusters); X, Y, V, and Z are level 4 nodes (level 3 clusters); and the entire network forms the level 4 cluster. The Location Management (LM) service provides S the address of the highest level cluster that contains D but does not contain S (e.g. the level 3 cluster Z in Figure 1). Node S can then construct a source-based route toward the destination. This route will be formed by a set of links in node S level 1 cluster ($X.1.1$), a set of level 2 links in node S super-cluster (level 2 clusters $X.1$), and so on. In Figure 1 the route found by node S is : $S-n_1-n_2-X.1.5-X.1.3-X.2-X.3-Y-Z-D$. When a node outside node S level 1 cluster receives a packet, the node will likely produce the same high-level route towards D, but it will 'expand' the high-level links that traverse its cluster using lower level (more detailed) information. In Figure 1 this expansion is shown for the segment $Z-D$.

The Location Management (LM) service can be implemented in different ways, whether proac-

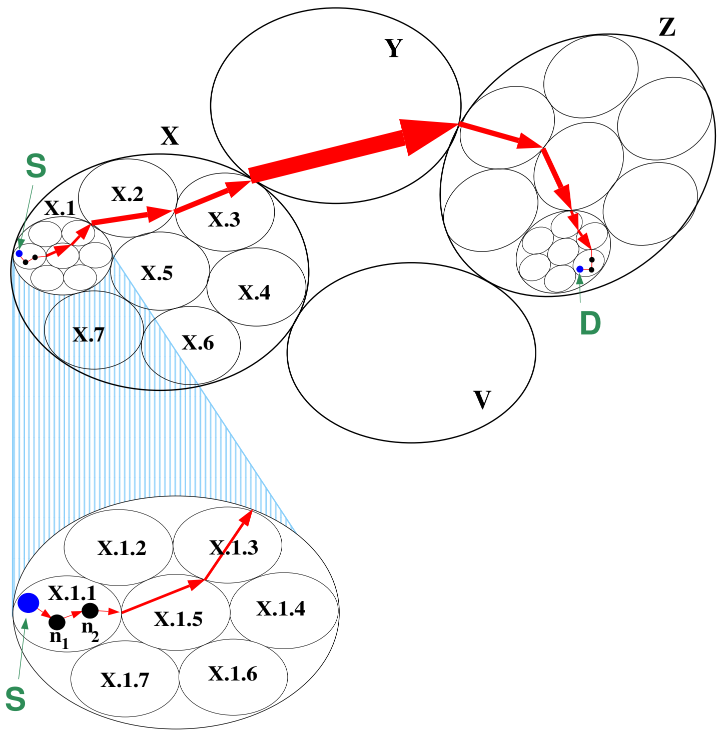

Figure 1: A Source (S) - Destination (D) path in HierLS, as computed for node S. Also, the high level link inside the destination cluster have been 'expanded'.

tive (location update messages), reactive (paging), or a combination of both. Typical choices are:

LM1 Pure reactive. Whenever a node changes its level i clustering membership but remains in the same level $i+1$ cluster, this node sends an update to all the nodes inside its level $i+1$ cluster. For example, if in figure 1 node n_2 moves inside cluster $X.1.5$, i.e. it changes its level 1 cluster membership but does not change its level 2 cluster membership (cluster $X.1$); then node n_2 will send a location update to all the nodes inside the cluster $X.1$. The rest of the nodes in the network are not informed.

LM2 Local paging. In this LM technique, one node in each level 1 cluster assumes the role of LM server. Also, one node among the level 1 LM servers inside the same level 2 cluster assumes the role of level 2 LM server, and so on up to the m level. The LM servers form a hierarchical tree. Location updates are only generated and transmitted between nodes in this tree (LM servers). When a node D changes its level i clustering membership, the LM server of its new level i cluster will send a location update message to the level $i+1$ LM server, which in turn will forward the update to all the level i LM servers inside this level $i+1$ cluster. Additionally, the level $i+1$ checks if the node D is new in the level $i+1$ cluster, and if this is the case it will send a location update to its level $i+2$ LM server, and so on.

When a level i LM server receives a location update message about node D from its level

$i+1$ LM server, it updates its local database with node D's new location information and forwards this information to to all the level $i-1$ LM servers inside its level i cluster. The level $i-1$ LM servers forward the location update message to the level $i-2$ server in its level $i-1$ cluster, and so on until all the level 1 LM servers (inside node D's level $i+1$ cluster) are informed of the new level i location information of node D.

When a node needs location information about any node in the network, the node pages its level 1 LM server for this information.

LM3 Global paging. LM3 is similar to LM2 in that they both create and maintain a LM server hierarchy. The difference between LM3 and LM2 strives in that when a level i LM server receives a location update from a higher level $i+1$ LM server, it does not forward this information to the lower level ($i-1$) LM servers. Thus, a lower level (say level $j<i$) LM server does not have location information for nodes outside its level j cluster. Additionally, a mechanism for removing outdated location information about nodes that left a level i cluster need to be added to the level i clusters LM servers. Basically, a level 1 LM server that detects that a node left its level 1 cluster will remove the entry corresponding to this node form its own database, and will inform its level 2 LM server. The level 2 LM server will wait for a while for a location update from the new level 1 cluster (if inside the same level 2 cluster) and if no such an update is received it will remove the node entry and will inform its level 3 LM server, and so on until arriving to a LM server that already has information about the new location of the node.

When a node needs location information about any node in the network, the node pages its level 1 LM server for the information. If the level 1 LM does not have the required information, it (the level 1 LM server) pages its level 2 LM server, who in turn pages its level 3 LM server, and so on, until a LM server with location information about the desired destination is found.

In this work we will initially assume that a proactive approach (LM1) is being used (because approach LM1 is easier to implement and analyze). We will refer to this protocol as HierLS-LM1. Approach LM2 (referred to as HierLS-LM2) potentially reduces the bandwidth consumption (for reasonable values of λ_s) but at the expense of complexity (selection and maintenance of LM servers) and an increase in the latency for route establishment. However, the asymptotic characteristic of HierLS do not change whether we use approach LM1 or approach LM2, as will be explained later (subsection 6.4). Approach LM3 (referred to as HierLS-LM3) is the more complex to implement and analyze. It will induce a significant amount of reactive overhead (susceptible to traffic), but will reduce the amount of overhead induced by mobility. Approach LM3 will be analyzed subsection 6.5.

In the following subsections we will develop expressions for HierLS-LM1 *proactive* and *sub-optimal route* costs and found the *total overhead* induced by this approach (since approach LM1 is used for location management, there is no *reactive* cost associated with HierLS-LM1). The last subsections modify the previous analysis to obtain the total overhead expression for HierLS-LM2, and HierLS-LM3.

6.1 HierLS-LM1 proactive cost

Let's consider a network organized in m level clusters, each of equal size k (thus the network size $N = k^m$). Note that k is fixed (predefined) meanwhile m increases with N.

If the cause of topology change is not mobility but random oscillations on the link quality due to interference, multi-path fading, short range mobility, etc. it is highly probable that there is no cluster membership change for any node. Thus, there is no location management cost. Even more, link status changes are not likely to trigger higher level (abstracted) link changes. This is because higher level links are the aggregation of several low-level links and the temporal changes in one link are not important enough to significantly affect the set.[3] Thus, if link changes are propagated inside level 1 clusters only, then each LSU will be retransmitted k times (once per each node in the same level 1 cluster as the node generating the LSU). Thus, since there are $\lambda_{lc} * N$ LSUs being generated each second, the *proactive* cost is $k * \lambda_{lc} * N$ bps, that is, *proactive* cost is $\Omega(\lambda_{lc} * N)$ bps.

In the other hand, if topology change is due to mobility, the *proactive* cost asymptotic behavior is dominated by the location management function (approach LM1). To visualize this, let's consider that the average speed of a node is s, then the time that a node takes to change its level $m-1$ cluster is directly proportional to the diameter of this level $m-1$ cluster and inversely proportional to the node speed s. Since the level $m-1$ cluster size is N/k, the cluster diameter is $\Theta(\sqrt{N/k})$. In approach LM1, the new location information will have to be forwarded to all the nodes inside the level m cluster (the whole network). Thus, every node will send a location update message to the whole network (N transmissions) each $\Theta(\sqrt{N/k}/s)$ seconds, inducing a cost of $\Theta(\sqrt{k} * s * \sqrt{N})$ bits each second. Thus, totaling over all nodes, the proactive cost due to level $m-1$ clusters membership change is $\Theta(\sqrt{k} * s * N^{1.5})$ per second. If we consider the location update generated due to level $m-i$ membership change, we will see that a level $m-i$ $(m-i+1)$ cluster is k^{i-1} times smaller than a level $m-1$ (m) cluster, and consequently a level $m-i$ cluster diameter is $k^{\frac{i-1}{2}}$ times smaller than a level $m-1$ cluster diameter. Thus, the generation rate of location updates due to level $m-i$ membership change is $k^{\frac{i-1}{2}}$ times larger than the rate induced by level $m-1$ changes. Also, since the new location information will have to be transmitted to all the nodes inside the current level $m-i+1$ cluster then the number of transmissions required for each packet decreases by a factor of k^{i-1} with respect to the number of transmissions induced by level $m-1$ changes, which results in a net reduction of $k^{\frac{i-1}{2}}$. Thus, the total cost due to location updates is :

[3] Usually, link state updates for higher level links are triggered when the difference between the current cost – however it is defined – and the last one advertised exceeds a threshold

$$\begin{aligned} Loc\_Upd\_Cost &= \Theta(\sqrt{k} * s * N^{1.5}) * [1 + \sqrt{\frac{1}{k}} + \sqrt{\frac{1}{k}}^2 + \sqrt{\frac{1}{k}}^3 + ...] \\ &= \Theta(\sqrt{k} * s * N^{1.5}) * \frac{1}{1 - \sqrt{1/k}} \\ &= \Theta(s * N^{1.5}) \end{aligned}$$

Thus, *proactive* cost in HierLS-LM1 has a lower bound (location management cost) that is $\Omega(s * N^{1.5})$ bps. Therefore, HierLS *proactive* cost per second is $\Omega(s * N^{1.5})$. Comparing this value with the $\Omega(\lambda_{lc} * N)$ cost due to LSUs in the non-mobility case, and considering that – unless mobility is highly grouped [4] – LSUs are mainly limited inside level 1 clusters (no higher level LSUs), we conclude that HierLS-LM1 *proactive* cost per second is $\Omega(s * N^{1.5} + \lambda_{lc} * N)$.

6.2 HierLS-LM1 sub-optimal route cost

To estimate the *sub-optimal route* cost, we assume that each level i (beginning at level 2) increases the actual route length by a factor f_i (f_i depends on the mobility rate and the value of k and is typically close to 1, for example $f = 1.05$ means a 5% increase in the route length). Thus, if the optimal path length is l, then the actual path length will be $\Pi_{i=2}^{i=m} f_i l$. Let f be the geometric average of the set $\{f_i\}$, that is, $f = (\Pi_{i=2}^{m} f_i)^{\frac{1}{m-1}}$. Then, the sub-optimal route cost induced by a packet transmission is $size\_of\_data * (f^{m-1} - 1) * l = size\_of\_data * (k^{(\log_k f)(m-1)} - 1) * l = size\_of\_data * (\frac{N}{k}^{\delta} - 1) * l$, where $\delta = \log_k f$.

Since there are $\lambda_t * N$ packets entering the network each second, the total *sub-optimal route* cost per second is $size\_of\_data * (\frac{N}{k}^{\delta} - 1) * L * \lambda_t * N$, and since L is $\Theta(\sqrt{N})$, we finally get that HierLS-LM1 [5] *sub-optimal route* cost per second is $\Theta(\lambda_t N^{1.5+\delta})$.

6.3 HierLS-LM1 total overhead

Combining the lower bound obtained for the *proactive* cost and the tight bound obtained for the *sub-optimal route* cost we finally obtain that the *total overhead* per second for HierLS-LM1 is $\Omega(s * N^{1.5} + \lambda_{lc} * N + \lambda_t N^{1.5+\delta})$. [6]

6.4 HierLS-LM2 total overhead

LM2 differs from LM1 in that:

[4] If nodes move as groups, higher level nodes (clusters) association may appear/disappear over time. For example, two groups may come close together or separate by going in different directions. In the other hand, if nodes mobility pattern are independent, the disappearance of some links between a pair of clusters will be likely compensated by the appearance of new links.

[5] This result is also valid for HierLS-LM2 and HierLS-LM3.

[6] The author considers this expression to be a tight bound and that the *total overhead* induced by HierLS-LM1 is $\Theta(s * N^{1.5} + \lambda_{lc} * N + \lambda_t N^{1.5+\delta})$.

- LM1 transmit location update to all the nodes inside a level i cluster. LM2 transmit this updates only to the level 1 LM servers.

- LM2 induces a reactive cost when paging the level 1 LM servers asking for a destination location information.

The first difference implies that LM2 reduction factor on location update cost with respect to LM1 is in the order of the ratio of the number of nodes (N) to the number of level 1 LM servers ($\Theta(N/k)$). This ratio is $\Theta(k)$; where k, the number of nodes in a level 1 cluster, is fixed (predetermined, bounded). Thus, HierLS-LM2 *proactive* cost asymptotic behavior is the same as HierLS-LM1's .

The second difference implies a reactive cost component in HierLS-LM2 *total overhead* expression. Indeed, in the worst case, each time a packet is transmitted the source has to page its level 1 location server. This paging message need to be retransmitted in average $\Theta(\sqrt{k})$ times. The important observation is that this number is bounded as the maximum size of a level 1 cluster is predetermined independently of the size N (what changes with N is the number of levels m). Thus, in the worst case, each second $\lambda_t * N$ paging messages will have to be retransmitted a constant number of times, therefore inducing a bandwidth consumption per second that is $\Theta(\lambda_t * N)$. Thus, the *reactive* cost induced by HierLS-LM2 (upper bounded) is $O(\lambda_t * N)$. This value is smaller than the *sub-optimal route* cost ($\Theta(\lambda_t * N^{1.5+\delta})$) and therefore has no impact in the total overhead expression.

Thus, HierLS-LM2 *total overhead* is $\Omega(s * N^{1.5} + \lambda_{lc} * N + \lambda_t N^{1.5+\delta})$, and shows the same asymptotic properties as HierLS-LM1.

6.5 HierLS-LM3 total overhead

When the LM3 approach is used, a node change in its level i membership will be informed (for the new level i LM server) to the other level i LM servers (k nodes on average). The number of transmission needed to reach each of these level i LM servers will be of the same order of magnitude that the diameter (D_i) of a level i cluster. The generation rates of these level i location updates (for a given node) will roughly be $\Theta(s/D_i)$ where s is the node speed as explained in subsection 6.1. Thus, the bandwidth consumption due to level i location updates induced by one node is $\Theta(k * D_i) * \Theta(s/D_i) = \Theta(k * s) = \Theta(s)$. Thus, considering the location updates due to levels 1, 2, .., m; we get that the bandwidth consumption due to all the location updates induced by a node is $\Theta(s * m) = Theta(s * log_k N)$. And, considering the bandwidth consumed by all the N nodes in the network we get that HierLS-LM3 location update cost is $\Theta(s * N \log N)$.

For the paging (*reactive*) cost, we recall that the fraction of nodes in a source (say S) $m-1$ level cluster is (on average) $1/k$. Thus, most of the nodes belong to a different level $m-1$ cluster. Since all the nodes are equiprobable destinations (assumption A.6), we conclude that the majority of destinations will require long pages, that is, will require pages that will travel all the way to the level $m-1$ LM server. Thus, we may simplify the analysis by considering only the cost of paging for information to destination outside one node level $m-1$ cluster. Thus, each page will require at least $\Omega(\sqrt{N/k})$ transmissions (assuming that optimal routes are available and because of assumption A.4). Also, a page is generated at least every new session and the fraction of these pages that refer to destination outside the source (S) level $m-1$ cluster is $k-1/k \approx 1$. Thus, each second there are at least $\frac{k-1}{k} * \lambda_s * N$ pages being generated (in the entire network), inducing a *reactive* cost of at least $\Omega(\frac{k-1}{k} * \lambda_s * N\sqrt{N/k}) = \Omega(\lambda_s N^{1.5})$.

Note that an upper bound can also be found if we consider that all pages are far reaching, that a page is triggered for each data packet (at a rate of $\lambda_t * N$ packets per second) and that the average number of transmissions required for each page is $\Theta(N^{0.5+\delta})$ (see subsection 6.2 about sub-optimal routes). Then, the paging cost obtained is $O(\lambda_t N^{1.5+\delta})$. Thus, the *reactive* cost (paging LM servers) can be absorbed by the *sub-optimal route* cost expression ($\Theta(\lambda_t N^{1.5+\delta})$), and its inclusion in the *total overhead* expression will have no effect.

Finally, HierLS-LM3 *total overhead* per second is $\Omega(s * N \log N + \lambda_{lc} * N + \lambda_t N^{1.5+\delta})$, slightly different from the expressions for HierLS-LM1 and HierLS-LM2.

7 Zone Routing Protocol (ZRP)

Zone Routing Protocol (ZRP) is a hybrid approach, combining a proactive and a reactive part. ZRP tries to minimize the sum of the proactive and reactive overhead.

In ZRP, a node propagates event-driven (Link State) updates to its k-hop neighbors (nodes at a distance, in hops, of k or less). Thus, each node has full knowledge of its k-hop neighborhood and may forward packets to any node on it. When a node needs to forward a packet outside its k-hop neighborhood, its sends a route request message (similar to DSR), but this packet do not need to be send to all the nodes in the network but only to a subset of them (namely, 'border nodes'). The nodes in this subset will have enough information about their k-hop neighborhoods as to decide whether to reply to the route request or to forward it to its own set of 'border' nodes. The route formed will be described in terms of the 'border' nodes only, thus allowing 'border' nodes to locally recover from individual link failures, reducing the route maintenance cost.

ZRP's *total overhead* components will be analyzed in the next subsections, where lower bounds will be derived.

7.1 ZRP proactive cost

After a node S detects a link status change, it generates and propagates a LSU with a Time To Live (TTL) field set to k. Thus, the LSU is retransmitted for all the nodes that are $k-1$ or less hops away from S. In average there are $\Theta((k-1)^2) = \Theta(k^2)$ such nodes (assumption A.3). Thus, in ZRP each LSU induces $\Theta(k^2)$ transmissions. Since the LSU generation rate (for the whole network) is $\lambda_{lc} * N$, then we conclude that ZRP *proactive* cost per second is $\Theta(k^2 * \lambda_{lc} * N)$.

Similarly, if we let n_k represent the average number of nodes inside a node (say S) 'zone' (i.e. less that k hops away from S), and recalling (assumption A.3) that $n_k = \Theta(k^2)$, then we obtain that ZRP *proactive* cost per second is $\Theta(n_k * \lambda_{lc} * N)$.

7.2 ZRP reactive cost

For ZRP's *reactive* cost, a lower bound will be provided. This lower bound will be obtained by considering only the bandwidth consumed by new session's route request (RREQ). The bandwidth consumed for new RREQ required for repairing paths (which could be significant in a highly mobile environment) is ignored, therefore the result obtained is just a lower bound.

To compute the bandwidth consumed by new session's route request (RREQ), we first consider the bandwidth consumed by one (1) route request (RREQ) generated by a node (say S) due to the beginning of a new session.

Let $\{B_i^S\}$ the set of border nodes that will need to be paged by node S. ZRP tries to efficiently minimize the number of nodes paged. Thus, considering that each border node 'covers' in average $n_k = \Theta(k^2)$ nodes (assumption A.3) and that each node is at least 'covered' by a border node, we conclude that adding up the number of nodes covered for any of the border nodes or the source, this value should be (at least) greater than N (i.e. the entire network is 'covered' when trying to find a destination). As a consequence $|\{B_i^S\}| + 1) * n_k \geq N$. The last inequality implies $|\{B_i^S\}| \geq \frac{N}{n_k} - 1$. Thus, the number of border nodes is $\Omega(N/n_k)$. The more effective the implementation of ZRP, the tighter the bound.

Now, each border node has to receive at least once the RREQ message originated by node S. For example, border node B_i^S received the RREQ for the first time from border node B_j^S. This RREQ packet has to travel k hops from B_j^S to B_i^S and therefore it consumes $size\_of\_RREQ * k$ bits. Adding up all such transmissions over all border nodes we get a bandwidth consumption of $size\_of\_RREQ * k * |\{B_i^S\}|$ bits. The last quantity is a lower bound for the bandwidth required for propagation of a node (S) RREQ message, since it does not consider duplicate transmission and back-propagation of RREQ messages, that although minimized by ZRP verification process, can not be totally eliminated. In general, the more efficient the verification algorithms, the tighter the bound.

Then, since each second there are $\lambda_s * N$ new RREQ being generated, ZRP *reactive* cost per second is lower bounded by $\lambda_s * N * size\_of\_RREQ * k * |\{B_i^S\}| \geq \lambda_s * N * size\_of\_RREQ * k * (\frac{N}{n_k} - 1)$. Finally, ZRP *reactive* cost per second is $\Omega(\lambda_s N^2/k) = \Omega(\lambda_s N^2/\sqrt{n_k})$.

7.3 ZRP sub-optimal route cost

ZRP paths degrade over time. Thus, ZRP's *sub-optimal route* cost increases with mobility and session duration, and decreases with the 'zone radius'. In one extreme if one node's zone is the entire network, this cost is zero; in the other if the zone radius is the smallest possible we got DSR performance.

We may obtain an upper bound if we consider that the worst possible path length for a ZRP session grows as $\Theta(N/k)$. To visualize this, consider a sequence of border nodes from source (S) to destination (D) long after the initial path was constructed and after several repair procedures have taken place : $S - B_1^S - B_2^S - B_3^S - B_4^S - ...D$. One property of this sequence is that 2 non-adjacent members of the list can not belong to each other zone (i.e. can not be less that k hops apart). [7] This property causes (as will be explained below) that the largest possible sequence increases as $\Theta(N/k^2)$. Also the number of transmissions required to forward one packet from one node of the sequence to the next is in average $\Theta(k)$. [8] Then, the maximum bandwidth that may be required to forward one packet (a long time after the session was first created, and assuming high mobility) is $\Theta(\frac{N}{k^2}) * \Theta(k) = \Theta(\frac{N}{k})$ bits. Since there are $\lambda_t * N$ packets being generated each second, an upper bound for *sub-optimal route* cost per second for ZRP is given by the difference between this maximum bandwidth employed ($\Theta(\frac{N}{k} * \lambda_t * N)$) and the optimal value ($\Theta(\lambda_t * N * L) = \Theta(\lambda_t * N^{1.5})$) that would have been obtained if full topology information were available. Thus, ZRP *sub-optimal route* cost per second is $O(\lambda_t * \frac{N^2}{k})$ (upper bound).

We note that the above term (upper bound) presents the same asymptotic behavior that the lower bound for ZRP *reactive* cost (considering that λ_t and λ_s are directly proportional, where the proportional constant is the average number of packets transmitted as part of a session). Thus, it is safe to say that the asymptotic behavior of ZRP is captured by the reactive and proactive cost alone, and that we can analyze the *total overhead* asymptotic behavior based on these values only. Thus, we will not try to further improve the loose upper bound on *sub-optimal route* presented here.

Now, we need to show that since the sequence of intermediate nodes from source S to destination D has the property that non-adjacent nodes are more than k hops apart then maximum possible length of such a sequence increases as $\Theta(N/k^2)$. Indeed, consider only the sub-sequence formed by the odd-placed nodes in the original sequence (i.e. S, B_2^S, B_4^S, B_6^S ...). The presence of S in the subsequence inhibits any other node in S's zone. Similarly, the presence of B_2^S in the

[7] For example, if B_1^S and B_4^S are less than k hops apart, then B_1^S will shorten the sequence from S to D as follows : $S - B_1^S - B_4^S - ...D$. Thus, by repeating the above procedure we always get a sequence with the aforementioned property.

[8] We know that the maximum distance between consecutive nodes in the sequence is k (one is in the zone of the other) and the minimum distance between nodes two position apart (e.g. B_1^S and B_3^S) in the sequence is at least $k+1$ (since they do not belong to each other zone because of the aforementioned property). Then, the average number of transmission required is between $\frac{k+1}{2}$ and k (i.e. $\Theta(k)$).

sub-sequence inhibits all other nodes in B_2^S zone. Now, consider that the maximum number of times a node j can be inhibit is r_j and let r be the average value of r_j over all the nodes, we will show later that r increases as $\Theta(1)$ with respect to network size (N) and zone radius (k). Thus adding up the number of nodes inhibit for the nodes in the subsequence can not be larger than $N * r$. Thus, the length of the subsequence is smaller than $N * r/n_k$, and the length of the original sequence is smaller than $2 * N * r/n_k$. Thus, the maximum sequence length is $\Theta(N/k^2)$ (provided r exists and is $\Theta(1)$).

Finally, it is easier to explain the behavior (bound) of r rather than showing it. So, let's try to understand intuitively the reasons r is independent of N and k. Let's consider a node X and the set of nodes $\{Y_i\}$ that inhibit node X of belonging to the aforementioned sub-sequence. We know that nodes $\{Y_i\}$ are at least k hops away from each other. Also, they must be k or less hops away from X (to be able to 'inhibit' it). Thus, the limit in the number of times node X can be inhibit is equal to the maximum number of inhibitors inside node X zone. In other words, r is the maximum number of nodes that can be inside node X zone (less than k hops away), given that they are all more than k hops away from each other. Before attempting to answer this question in a graph theoretic framework, let's formulate a similar (in view of assumptions A.2, A.3, and A.4) geometric problem: what is the maximum number of points that can be placed inside a circle of radius R, such that the minimum distance between any of this points is greater than R?. If the condition would have been "greater or equal" to R, it is easily verified that the solution would be the 7 points shown in Figure 2: P_1, the center of the circle, and the 6 vertex ($P_2 \dots P_7$) of a regular hexagon with its side lengths equal to R and its center colocated with the circle center (i.e. P_1). Thus the six vertex will be exactly on the border of the circle of radius R. Any repositioning of the points trying to give room for another one will result in the reduction of some of the distances to less than R. Thus, if 7 is the maximum we can get when we allow the distances to be "greater or equal" to R, then when we restrict the constrain to be only "greater", 7 is no longer achievable and the maximum would be at most 6. Two important observations are that this value (6 or 7) does not depend on the circle radius; and also, the maximum number of points is achieved when most of the points try to be on the border of the circle (that is, trying to expand the distances as much as possible).

We now use the insight gained in the geometrical interpretation to justify that r is bounded independently of N and k. We know that r is the (average) maximum number of nodes that can be inside node X zone (less than k hops away) given that they are all more than k hops away from each other. Form the geometric insight we know that in average the number of nodes may be maximized if we put the nodes in the limit of the node X zone. Let's assume that similar to the geometric case, we choose X as its own inhibitor (just for argument sake) and also choose the $r-1$ remaining nodes to belong to the border of the zone. Let $\mathcal{C}$ be the shortest loop (formed only by nodes inside X zone) containing all this $r-1$ nodes. The length of this cycle will increase as

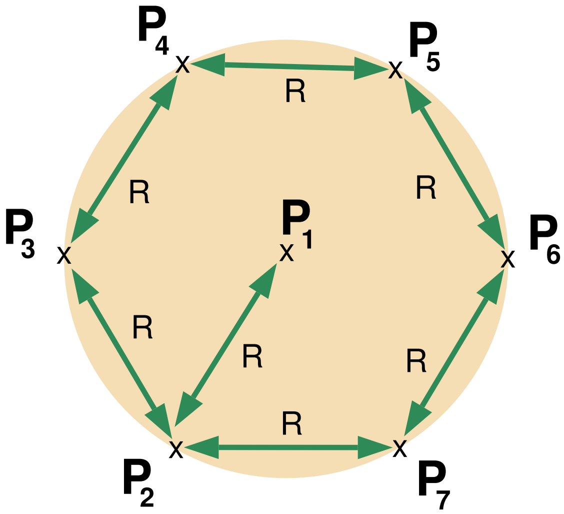

Figure 2: Geometrical interpretation of the maximum number of 'inhibitors' that 'cover' a node

$\Theta(\sqrt{n_k}) = \Theta(k)$, because assumption A.4 implies that the one-dimensional metrics (average path, maximum path, and consequently also the set 'perimeter') of a set of nodes are directly proportional to the square of the number of nodes in the set. Finally, since the loop length is $\Theta(k)$, and the distance (length) between two consecutive 'inhibitor' nodes in the loop is greater than or equal to $k+1$; then the maximum number of 'inhibitor' nodes is equal to the cycle length divided by $k+1$, which is $\Theta(k)/(k+1) = \Theta(1)$. Thus, r is bounded, and this bound is $\Theta(1)$, independent of k and N.[9]

7.4 ZRP total overhead

In the previous subsection be obtained a lower bound for ZRP's *reactive* and *proactive* cost. Also, an upper bound for ZRP *sub-optimal route* cost showed that inclusion of this term is not necessary to analyze the asymptotic properties on ZRP's *total overhead*. Thus, ZRP *total overhead* is $\Omega(n_k * \lambda_{lc} * N + \lambda_s N^2/\sqrt{n_k})$, which is obtained by adding the *reactive* and *proactive* cost only.

If we attempt to minimize the above lower bound by properly choosing the value n_k, we get that the best asymptotic behavior of the bound is obtained when (if possible) $n_k = \Theta((\frac{\lambda_s * N}{\lambda_{lc}})^{\frac{2}{3}})$, obtaining a *total overhead* that is $\Omega(\lambda_{lc}^{\frac{1}{3}} * \lambda_s^{\frac{2}{3}} * N^{\frac{5}{3}})$. [10]

Note that when $\lambda_{lc} = \Theta(\lambda_s * N)$, n_k must be $\Theta(1)$ and therefore the *total overhead* induced by ZRP becomes $\Omega(\lambda_{lc} * N + \lambda_s N^2) = \Omega(N * (\lambda_{lc} + \lambda_s * N)) = \Omega(\lambda_s * N^2)$. If λ_{lc} grows faster that $\Theta(\lambda_s * N)$, values of n_k lower than 1 does not make sense. What happen is that ZRP behaves

[9] This paragraph does not intend to be a mathematical proof, but only an intuitive explanation of the reasoning for considering r to be $\Theta(1)$.

[10] Note that in this case k, the zone radius, is $\Theta((\frac{\lambda_s * N}{\lambda_{lc}})^{\frac{1}{3}})$. Thus, k should increases with traffic and decreases with mobility as expected; but the dependency is not linear.

in pure reactive mode (similar as DSR) and therefore the *total overhead* induced by ZRP in those cases is also $\Omega(\lambda_s * N^2)$.

In the other hand, if $\lambda_{lc} = \Theta(\lambda_s/\sqrt{N})$, the best achievable value of n_k is $\Theta(N)$. Thus, ZRP *total overhead* becomes $\Omega(\lambda_{lc} * N^2 + \lambda_s N^{1.5}) = \Omega(N^2 * (\lambda_{lc} + \lambda_s/\sqrt{N})) = \Omega(\lambda_{lc} * N^2)$. If $\lambda_s/\sqrt{N}$ grows faster than λ_{lc}, then n_k can not grow more than N and therefore ZRP behaves in pure proactive mode (as SLS) and induces a total overhead of $\Omega(\lambda_{lc} * N^2)$.

Finally, ZRP total overhead is:

$$ZRP_{total\_overhead} = \begin{cases} \Omega(\lambda_{lc} * N^2) & \text{if } \lambda_{lc} = O(\lambda_s/\sqrt{N}) \\ \Omega(\lambda_{lc}^{\frac{1}{3}} * \lambda_s^{\frac{2}{3}} * N^{\frac{5}{3}}) & \text{if } \lambda_{lc} = \Omega(\lambda_s/\sqrt{N}) \text{ and } \lambda_{lc} = O(\lambda_s * N) \\ \Omega(\lambda_s * N^2) & \text{if } \lambda_{lc} = \Omega(\lambda_s * N) \end{cases}$$

8 Hazy Sighted Link State (HSLS)

In a highly mobile environment, the Hazy Sighted Link State (HSLS) approach will transmit Link Status Updates (LSU) at particular time instants that are multiples of t_e seconds. Thus, several link changes are 'collected' and transmitted every t_e seconds. The *Time To Live* (TTL) field of the LSU packet is set to a value that is a function of the current time index. After one broadcast LSU transmission – LSU that travels over the entire network, i.e. TTL field set to infinity – all counters are reset and the algorithm is (re)initialized. After this transmission, a node 'wakes up' every t_e seconds and sends a LSU with TTL set to s_1 if there has been a link status change in the last t_e seconds. Also, the node wakes up every $2 * t_e$ seconds and transmits a LSU with TTL set to s_2 if there has been a link status change in the last $2 * t_e$ seconds. In general, a node wakes up every $2^{i-1} * t_e$ $(i = 1, 2, 3, ...)$ seconds and transmits a LSU with TTL set to s_i if there has been a link status change in the last $2^{i-1} * t_e$ seconds.[11] If the value of s_i is greater than the distance from this node to any other node in the network (which will cause the LSU to reach the entire network), the TTL field of the LSU is set to infinity and the algorithm is reset. Figure 3 shows an example of HSLS's LSU generation process when mobility is high and in consequence LSUs are always generated. Note that the sequence $s_1, s_2, \ldots$ is non-decreasing. For example consider what happens at time $4 * t_e$. This time is a multiple of t_e (associated with s_1), also a multiple of $2 * t_e$ (associated with s_2) and $4 * t_e$ (associated with s_3). Note that if there has been a link status change in the past t_e or $2 * t_e$ seconds, then this implies that there has been a link change in the past $4 * t_e$ seconds. Thus, if we have to set the TTL field to at least s_1 (s_2) we also have to increase it to s_3. Similarly, if there has not been a link status change in the past $4 * t_e$ seconds, then there has not been a link change in the past t_e or $2 * t_e$ seconds. Thus, if we do not send a LSU with TTL set to s_3, we do not send a LSU at all. Thus, the resulting decision at time $4 * t_e$ (and any other time

[11] In case a node has several LSUs to transmit with TTL values s_1, s_2, etc, the node will only transmit one LSU with the highest TTL value, as will be explained later in this subsection.

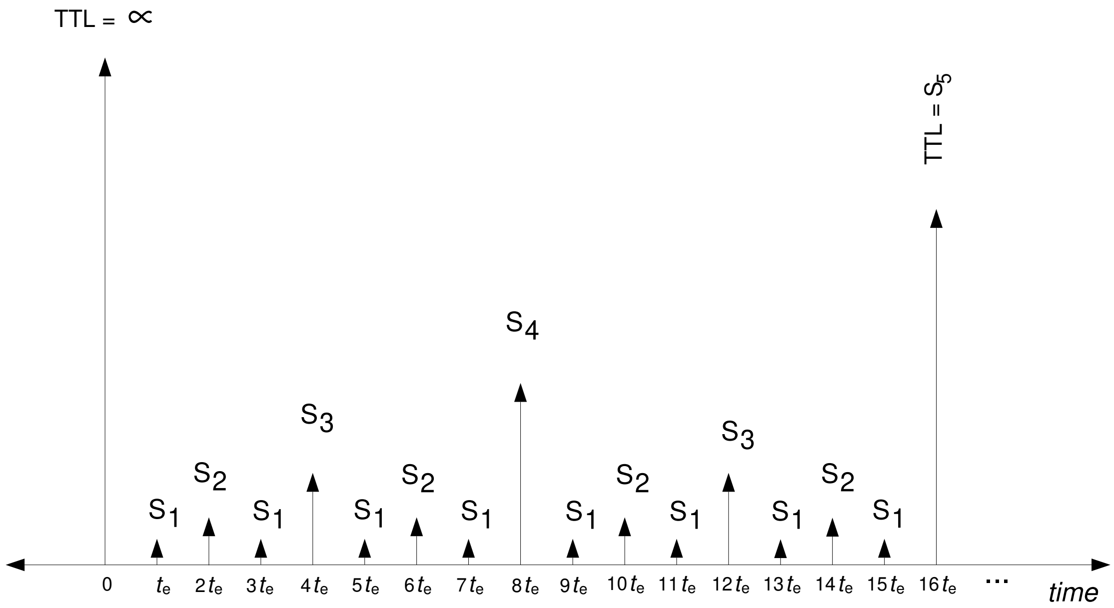

Figure 3: General LSU generation process for the Hazy Sighted Link State (HSLS) protocol

$4 * k * t_e$ where k is a odd number) is to check if there has been any link status change during the past $4 * t_e$ seconds and if that is the case, to send a LSU with TTL set to s_3.

At this point we should note that HSLS verification of link status in the past $2^{i-1} * t_e$ seconds is different for the cases $i = 1$ and $i \geq 2$. For $i = 1$, HSLS compares the current link status with the one recorded t_e seconds ago. Thus, if in the last t_e seconds a link came down and then recovered (went up again), then no LSU would be generated. On the other hand, for $i \geq 2$ a node running HSLS will check for the link status recorded for each link in the 2^{i-1} waking times (covering a period of time of $2^{i-1} * t_e$ seconds) and if all this link status are not the same, then a LSU will be generated and transmitted with TTL set to s_i.

The reader will have noticed that the algorithm definition is not complete since the values of the sequence $\{s_i\}$ has not been defined yet. Hazy Sighted Link State is based in the observation that nodes that are far away do not need to have complete topological information in order to make a good next hop decision, thus propagating every link status change over the network may not be necessary. The function s_i is chosen as to minimize the *total overhead* (as defined in the previous section). Based on assumption A.6 an uniform traffic distribution among all the nodes in the network was assumed and as a consequence the best performance is obtained [12] when $s_i = 2^i$.

Finally, $s_1 = 2, s_2 = 4, s_3 = 8, s_4 = 16, ...$ and so for. Thus, nodes that are at most two hops away from a node X will receive information about any node X's link status change at most after t_e seconds. Nodes that are more than 2 but at most 4 hops away from X will receive information

[12]The derivation of s_i is out of the scope of the present paper, but intuitively it can be seen that due to our 'uniform' distribution of traffic destinations, a linear relationship between distance and freshness of information is obtained. This linear relationship causes that the probability of a bad next hop decision remain roughly constant independently of the distance to the destination, as will be later explained in subsection 8.2.

about any of X links change at most after $2 * t_e$ seconds. In general, nodes that are more than 2^{i-1} but at most 2^i hops away from X will receive information about any of X links change at most after $2^{i-1} * t_e$ seconds.

8.1 HSLS proactive cost

Consider Figure 3. We assume that we are in a highly mobile environment so that every time interval a LSU is generated. This assumption will be relaxed later, but for now it helps to better understand the analysis. We want to add together all the different LSUs (re)transmissions due to LSUs generated by node X and then average them over time (later, we multiply this average value for the number of nodes in the network to get the *proactive* cost). We can begin grouping LSUs by their TTL value at the time they were generated. For example, we count the number of packets that has been transmitted due to LSUs that were generated with a TTL set to $s_4 = 16$ (or what is the same, that were generated at times $8 * k * t_e$, where k is odd). These LSUs (of size $\Theta(d)$, where d is the average node degree) are generated every $16 * t_e$ seconds and are transmitted to $\Theta(d * 16^2)$ nodes (assumption A.3). Thus, the bandwidth consumed by these packets (originated at times $8 * k * t_e$) will be $\Theta(\frac{d*d*16^2}{16*t_e}) = \Theta(\frac{d^2*16}{t_e})$. If we consider the bandwidth consumed by packets transmitted due to LSUs generated with TTL set to $s_3 = 8$, we notice that the generation rate increases by a factor of 2, but the number of packets (re)transmitted decreases by a factor of 4, thus the total effect is a reduction of 2 in the bandwidth consumed. Let C_i be the bandwidth consumed by the LSUs with TTL equal to 2^i, then the bandwidth consumed for the LSUs with TTL equal to 2^i or lower is equal to $C_i + C_{i-1} + C_{i-2} + ... = C_i[1 + \frac{1}{2} + \frac{1}{4} + \frac{1}{8}] \leq 2 * C_i$. Thus, an important result is that the proactive cost is dominated by the longest reaching LSUs, that is, by the LSUs with the maximum possible value of TTL (i.e. with TTL set to the maximum distance between any two nodes).

Thus, a more careful analysis begins by accounting for the LSUs with larger TTL. We already mentioned that any node (say X), computes its maximum distance to any other node ((MD_x) and if the value of TTL provided by the HSLS algorithm is greater than this value, node X will set set the TTL value to infinity (broadcast LSU) and will reset the counters and timers. Let R_x be the power of 2 such that $R_x < MD_x \leq 2R_x$. Thus, at time $R_x * t_e$ the node X sends a LSU to the entire network and resets the counter to zero. Thus, every $R_x * t_e$ seconds node X induces N transmissions, and therefore the bandwidth consumption due to this broadcast LSUs is $\frac{size\_of\_LSU*N}{R_x*t_e}$.

The second larger TTL is R_x, and LSUs with this TTL are generated at times $\frac{R_x}{2} * t_e$. Recalling that the timer are reset at time $R_x * t_e$, we notice that the interval between consecutive generation times is $(R_x * t_e - \frac{R_x}{2} * t_e) + \frac{R_x}{2} * t_e = R_x * t_e$. Thus, the generation rate of LSUs with TTL equal to R_x is $\frac{1}{R_x*t_e}$ (the same as the generation rate of broadcast LSU, i.e. LSUs with TTL set to infinity). These LSUs (with TTL set to R_x) induces s_{R_x} transmissions (where $s_{R_x} = \Theta(R_x^2)$). It is clear that these LSUs will not reach all the nodes in the network so $s_{R_x} < N$. Let $f_x = s_{R_x}/N$, thus from assumption A.3, f_x should be around $(R_x/MD_x)^2$, i.e., $f_x \in [0.25, 1]$. In practical situations, due to boundary effects (i.e. the number of nodes at a maximum distance MD_x is small), we got that typically f_x is in the interval $[0.5, 1]$. Thus, the bandwidth consumption due to LSUs with TTL equal to R_x is $\frac{size\_of\_LSU * f_x * N}{R_x * t_e}$.

For the remaining TTL values we do not need to consider 'boundary' conditions anymore. Thus, for TTL equal to $R_x/2$, the generation rate doubles and the number of transmissions induced per LSU is reduced by a factor of 4, thus the total effect is a reduction by a factor of 2 regarding the bandwidth consumption due to LSUs with TTL equal to R_x. The same argument applies for TTL equal to $R_x/4, R_x/8, ..., 2, 1$. [13] Finally, the total bandwidth consumption due to all the LSUs generated by node X is equal to :

$$\begin{aligned} X_{HSLS}^{proactive\_cost} &= \frac{size\_of\_LSU * N}{R_x * t_e} + \frac{size\_of\_LSU * f_x * N}{R_x * t_e} + \frac{size\_of\_LSU * f_x * N}{2 * R_x * t_e} + \\ &+ \frac{size\_of\_LSU * f_x * N}{4 * R_x * t_e} + \ldots \\ &= \frac{size\_of\_LSU * N}{R_x * t_e}[1 + f_x(1 + \frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \ldots)] \\ &\approx \frac{size\_of\_LSU * N}{R_x * t_e}[1 + 2 * f_x] \end{aligned}$$

Thus, since the size of a LSU only depends on the node density (bounded in average); f_x is bounded below 1; and R_x is $\Theta(\sqrt{N})$ (assumption A.4); the *proactive* cost per second induced by one node is $\Theta(\frac{N^{0.5}}{t_e})$. Since there are N nodes, the *proactive* cost per second induced by the whole network is $\Theta(\frac{N^{1.5}}{t_e})$.

8.2 HSLS sub-optimal route cost

The key to the derivation of HSLS *sub-optimal route* cost is to understand that if the probability of making an incorrect next hop decision at any time is independent of network size N and traffic λ_t, then the *sub-optimal route* cost increases with respect to traffic and size as $\Theta(\lambda_t * N^{1.5})$. [14]. To visualize this, consider a simple network model where each node's probability of making a bad next hop decision, independently of the distance k to the destination, [15] is p. Moreover, consider the scenario shown in Figure 4 where a node S must forward a packet towards a destination D (in general, S is not the source of the packet but it is relaying it).

[13] Although assumptions A.3 and A.4 are not applicable to small values of TTL because they are asymptotic conditions, previous discussion has already shown that the impact of LSUs with small TTL in the total proactive cost induced in a large network is not significant and we can safely avoid more exact analysis for the bandwidth induced by this (small TTL) LSUs.

[14] Note that the *sub-optimal route* cost also depends on mobility and/or rate of topological change so the above expression is not complete. Later, a more precise expression will be derived

[15] Later in this subsection we will show that for HSLS, the probability of a bad next hop decision roughly remains constant (or at least lower- and upper- bounded) with respect to the distance to the destination.

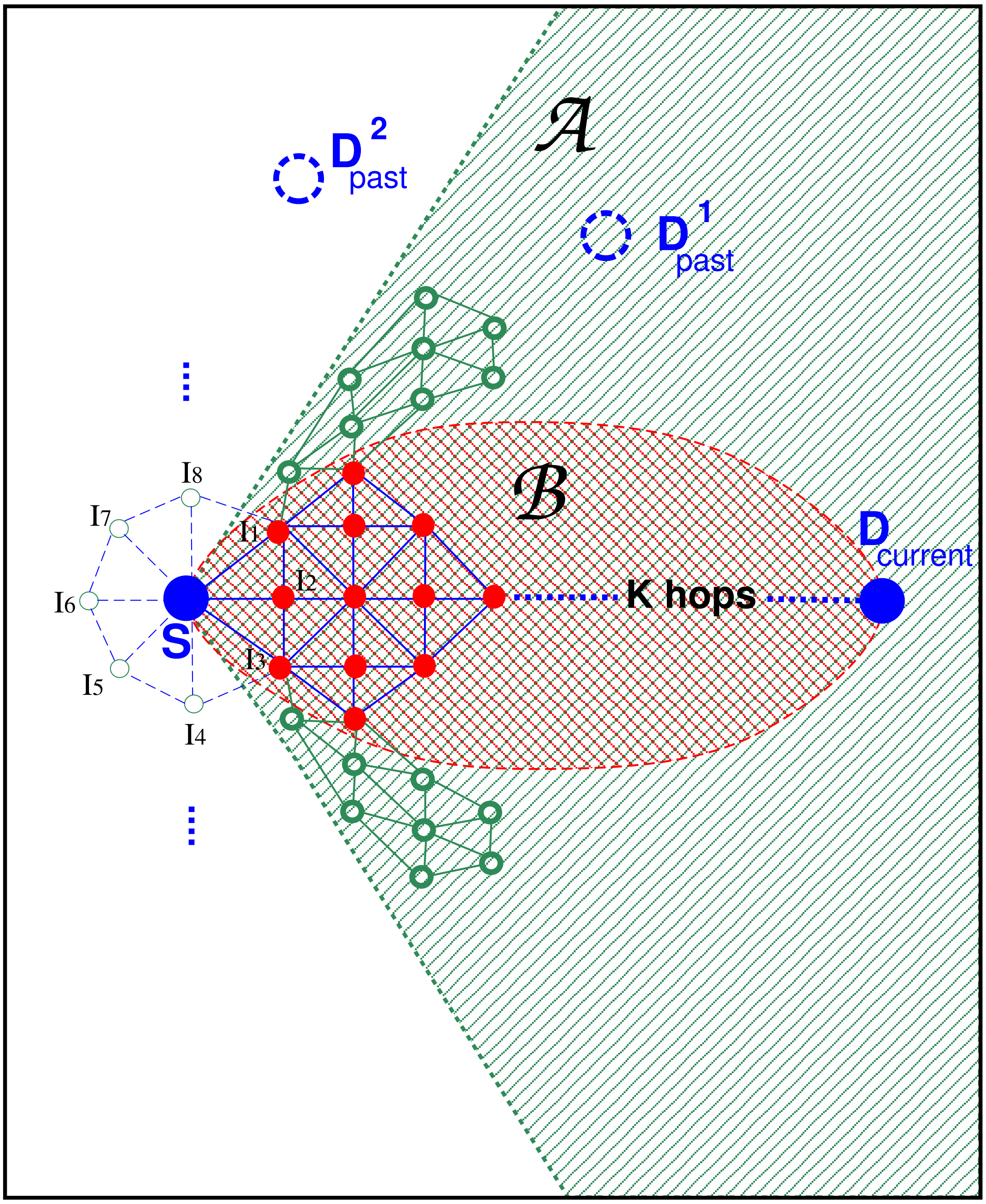

Figure 4: Good and bad next hop decision in the HSLS protocol

Let k be the minimum distance (in hops) from S to D. There are several paths that achieve this distance, and the set of nodes forming part of these paths are enclosed in the small red area including I_1, I_2, and I_3. Thus a correct next hop decision will be made if the packet is delivered to any of this three nodes (I_1, I_2, or I_3). Clearly, if the packet is delivered to any of these nodes, the distance from the new location of the packet (i.e. I_1, or I_2, or I_3) to D will decrease by one to $k-1$. If a bad next hop decision is made (packet delivered to $I_4, \ldots, I_8$, the distance from the new location of the packet to D will be k (remain the same) or increase to $k+1$ (can not be more since there is at least one path of $k+1$ hops from any I_j to D, i.e. the path $I_j - S - minimum\_path_{S\_to\_D} - D$). It can be intuitively seen that the probability that a next hop error actually produces an increase in the distance to the destination is very rare in a network, unless the density is very small (sparse network). So, we can simplify our model and assume that a packet can be successfully delivered to a proper next hop node (thus reducing the distance to the destination by 1) with a probability $1-p$, or can be delivered to a wrong node leaving the packet's distance to the destination unaltered with probability p. If we further assume that consecutive routing decisions regarding a packet are independent, then we can estimate the expected number of packets transmissions (trials) necessary to move a packet from a distance k to a distance $k-1$ as $\frac{1}{1-p}$. And since the optimal number of trials is 1, then the (average) wasted bandwidth in this one hop transmission is $size\_of\_data * (\frac{1}{1-p} - 1) = size\_of\_data * \frac{p}{1-p}$. Thus, the average number of transmissions wasted when forwarding a packet from a source L_i hops away from the destination is $size\_of\_data * \frac{p}{1-p} * L_i$. Finally, since each second there are $\lambda_t * N$ packets being generated, and the average (optimal) path length of this packets is L, the bandwidth wasted due to sub-optimal routes is $\lambda_t * N * size\_of\_data * \frac{p}{1-p} * L = \Theta(\lambda_t * N^{1.5})$, where the last equality holds since $L = \Theta(\sqrt{N})$ (assumption A.4).

The above assumption that consecutive routing decision are independent is obviously not true since a node routing decisions (and therefore mistakes) are highly correlated over time (until new link status information is available or some other mechanism – as for example loop detection – is provided to help not to make the same mistake twice) and space. Thus, this model ignores the presence of loops and other phenomena. However, the model is good enough to make the point that if p does not depend on size or traffic, a protocol *sub-optimal route* cost increases as $\Theta(\lambda_t * N^{1.5})$ with respect to size and traffic.

The remaining of this subsection will focus in showing that HSLS probability of a bad next hop decision is independent of the size and approximately constant for different distances to the destination.. The independence of p with traffic is obvious since HSLS is a proactive approach where routing information is propagated as a consequence of events (link status changes) that are independent of the traffic. [16]

[16] In some protocols LSU generation and traffic could be correlated. For example, if eavesdropping of application level acknowledgments is used to estimate the status of a link. This case has not been considered in this work, although it can be intuitively understood that in such a case the protocol performance will only improve due to quick

To analyze the probability of a bad next hop decision we need to go back to Figure 4. There $D_{current}$ represents the actual (topological) position of node D. D^j_{past}, with $j = 1, 2$ represents to possibilities of node D's topological position as seen by node S, k hops away (who may not have up to date information). The small (red) region $\mathcal{B}$ is the set of nodes that belong to any of the minimum distance path from S to D. I_1, I_2, and I_3 belong to this set, and therefore if S chooses one of these nodes as the next hop, a good next hop decision will have been made. Now, the larger (green) region $\mathcal{A}$ represents the set of nodes for whom the shortest path first algorithm run over S's (out-of-date) topology table gives as output I_1, I_2, or I_3. In our example D^1_{past} belongs to this set whether D^2_{past} do not. At this point the reader may be confused since Figure 4 seems to present areas meanwhile our discussion refers to topologies. In general the sets aforementioned do not have to cover whole areas and may have arbitrary shapes. Figure 4 presents what is expected to be an average case (due to our geometrical analogies motivated by assumptions A.2, A.3, and A.4). Thus, it is expected that the set of nodes described above cover a more-or-less compact area and that the success of the next hop decision made by S is intimately related to the fact that at the time of the last LSU received from D the physical position (that induces the network topology) of node D was inside the green area (as for example is the case on D^1_{past}). This assumption is more realistic when dealing with large distances.

In the above setting, the probability of a bad next hop decision p (at least for an asymptotically large network) will be the probability that at the time t_{past} when the last LSU was received (or assumed) the node position was not inside the green area ($\mathcal{A}$) given that at the current time $t_{current}$ the node is in the position $D_{current}$. This probability clearly depends on the time elapsed ($t_{elapsed} = t_{current} - t_{past}$) since we last received 'fresh' information, and on the node mobility model. Assumption A.8 implies that p is a function of $\frac{t_{elapsed}}{k}$. The particular form of this function will depend on $g_{0/1}(x, y)$ (i.e. the traffic model) and the normalized area $\mathcal{A}'$ (result of compressing area $\mathcal{A}$ so that the distance from S to D is unity). Thus $p = h(\frac{t_{elapsed}}{k}, g_{0/1}(x, y), \mathcal{A}')$. $h(., ., .)$ form may be complicated but it is clear that it will be nondecreasing with $\frac{t_{elapsed}}{k}$ and non-increasing with $\mathcal{A}'$.[17]

Thus, if we can lower bound $\mathcal{A}'$ and upper bound $\frac{t_{elapsed}}{k}$ when N increases to infinity we will have shown that p is bounded independent of N (we already assumed that a node mobility model – defined by its function $g_{0/1}(x, y)$ is independent of the number of nodes). To see that the average $\mathcal{A}'$ is lower bounded, consider that in the worst case the set of 'good' next hop decisions is formed by just one node (out of the d neighbors – in average – of a node). Thus, if the network is balanced (in average), the number of nodes that will return the 'good' next hop node as the output of their shortest path first algorithm should be roughly N/d. Therefore, the region $\mathcal{A}'$ will consist of a fraction of (in average) at least $1/d$ of the total area.

link failure detection.

[17]i.e. If one region totally contains another, h evaluated in the former may not be greater than h evaluated in the latter.

To see that $\frac{t_{elapsed}}{k}$ is upper bounded, we must consider that a node (say S) that is k hops away of another (say D), where $2^i < k \leq 2^{i+1}$ will receive updates about any link change detected by node D at most after $2^i * t_e$ seconds. Thus, if no LSU has been received in a long time, then at time t the node (S) knows that up until time $t - 2^i * t_e$ no link status change has been experienced by node D (which is equivalent to say that the relative position of node D with respect to their neighbors has not change much). [18] It still remains the possibility that D's neighbors move as a group but this will be detected by nodes closer to S and S will be alerted of this changes. Thus, it is safe to say that the $t_{elapsed}$ since S heard about D's whereabouts for the last time is lower than $2^i * t_e$. Thus $\frac{t_{elapsed}}{k} < \frac{2^i * t_e}{k} = \frac{2^i}{k} * t_e < t_e$. And since t_e is independent of the network size, we conclude that p is bounded as N grows to infinity. Which shows that HSLS *sub-optimal route* is $\Theta(\lambda_t * N^{1.5})$.

It is interesting investigate HSLS's *sub-optimal route* cost dependency with t_e and if possible with speed (s). In order to gain insight with tractable solutions, some extra assumptions need to be made. For example, consider that $g_{0/1}$ is such that the functional form of p follows the probability distribution function typically used for analyzing residence time in cellular systems, we can consider mobility models that produce a value of $p = 1 - e^{\frac{-s\bar{t}_{elapsed}K_1}{k}}$, where K_1 is a constant that depends on the topology, average node degree, etc., and $\bar{t}_{elapsed}$ is the average time elapsed since correct link status information regarding the destination was available. Such a function is based on the underlying assumption that the expected node position after a given time varies linearly with the speed, thus the speed is directly proportional to the rate of topological change (i.e. doubling the speed will be equivalent to 'compress' the time between event by a factor of 2). Then, we can focus in networks where the functional form of p is defined by $p = 1 - e^{\frac{-\lambda_{lc}\bar{t}_{elapsed}K_2}{k}}$. Note that this expression for p depending on the rate of link changes (λ_{lc}) makes more sense when dealing with topologies and may even be true if we relax our mobility constraints (assumptions). In networks where the above assumptions are true, HSLS *sub-optimal route* cost is equal to $K_3 \frac{p}{1-p} \lambda_t N^{1.5} = K_3 * (e^{\frac{\lambda_{lc}\bar{t}_{elapsed}K_2}{k}} - 1)\lambda_t N^{1.5}$, where K_2 and K_3 are constants. Considering the ratio $\frac{\bar{t}_{elapsed}}{k}$ it has already been shown that a node (S) that is k hops away of another (D) with $2^i < k \leq 2^{i+1}$ will experience a delay in the reception of new link state information about D that is bounded by $2^i * t_e$ seconds. It is not difficult to visualize that on average node S will experience a delay of $\frac{2^i * t_e}{2}$. Thus the average ratio $\frac{\bar{t}_{elapsed}}{k} = \frac{2^i * t_e}{2 * k}$ will be bounded by $\frac{t_e}{2} > \frac{\bar{t}_{elapsed}}{k} \geq \frac{t_e}{4}$. Thus, HSLS *sub-optimal route* cost is equal to $K_3 * (e^{\lambda_{lc} t_e K_4} - 1)\lambda_t N^{1.5}$, where K_3 and K_4 are constants.

8.3 HSLS total overhead

HSLS does not induce reactive overhead, thus taking into account the *proactive* ($K_5 * \frac{N^{1.5}}{t_e}$, where K_5 is a constant) and *sub-optimal route* cost ($\frac{p}{1-p}\lambda_t N^{1.5}$) we obtain that HSLS total cost is $\Theta(N^{1.5})$.

[18] Recall that in this normalized model the distance between 2 neighbors is small compared with the area cover by a large number of nodes since density is not allowed to increase beyond a limit.

Now, let's consider that the value of t_e is not fixed but can adapt with λ_{lc} and λ_t. What is the best HSLS can do?

HSLS *total overhead* for the class of networks analyzed in the previous subsection is :

$$HSLS_{total\_overhead} = N^{1.5}[K_5 \frac{1}{t_e} + K_3 * (e^{\lambda_{lc} t_e K_4} - 1)\lambda_t]$$

For a moment, let's use the approximation $e^x - 1 \approx x$, where $x = \lambda_{lc} t_e K_4$. Thus:

$$HSLS_{total\_overhead} \approx N^{1.5} * [\frac{K_5}{t_e} + K_6 * \lambda_{lc} * \lambda_t * t_e]$$

Thus, choosing the value of t_e that minimizes the above expression we get $t_e = \Theta(\frac{1}{\sqrt{\lambda_{lc}\lambda_t}})$, $x = \Theta(\frac{\sqrt{\lambda_{lc}}}{\sqrt{\lambda_t}})$, and $HSLS_{total\_overhead} = \Theta(\sqrt{\lambda_{lc}\lambda_t} N^{1.5})$. The previous expression would define the asymptotic behavior of HSLS *total overhead* only if our approximation $e^x - 1 \approx x$ is valid. Indeed, if λ_t grows asymptotically faster than λ_{lc}, the value of x goes to zero and the approximation $e^x - 1 \approx x$ is valid.

In the other hand, if λ_{lc} grows asymptotically faster than λ_t, the approximation will not be valid. In this case, since the exponential function is the fastest growing, it is desirable to maintain the product $\lambda_{lc} * t_e$ (and therefore the value of p) bounded and therefore we choose $t_e = \Theta(\frac{1}{\lambda_{lc}})$. Thus, HSLS *total overhead* in this scenario becomes $\Theta(N^{1.5} * (\lambda_{lc} + \lambda_t)) = \Theta(\lambda_{lc} * N^{1.5})$, where the last equality holds due to our assumption that λ_{lc} grows asymptotically faster than λ_t and therefore λ_{lc} dominates the previous expression.

Thus, based on the above assumption we can say that HSLS *total overhead* is :

$$HSLS_{total\_overhead} = \begin{cases} \Theta(\sqrt{\lambda_{lc}\lambda_t} N^{1.5}) & \text{if } \lambda_{lc} = O(\lambda_t) \\ \Theta(\lambda_{lc} * N^{1.5}) & \text{if } \lambda_{lc} = \Omega(\lambda_t) \end{cases}$$

9 Conclusions and comparisons

Based on a set of realistic (in the sense that there are networks for which the assumptions hold), the asymptotic *total overhead* consumed by SLS, DSR, PF, HierLS, ZRP, and HSLS has been analyzed. The results are summarized in Tables 1 and 2. Table 1 present the overhead results for cost type (*proactive*, *reactive*, and *sub-optimal route* cost). [19] Table 2 presents the results for *total overhead* when the tunable parameters are set as to optimize performance (or at least, optimize the lower bounds).

These results will help network designers to better identify the class of protocols to engage depending on their operation scenario.

For example, if the designer's main concern is network size; it can be noted that HierLS and HSLS scale better than the others. Similarly, if traffic intensity is the most demanding requirement, we see that SLS, and ZRP are to be preferred since they scale better (*total overhead* is $\Theta(1)$,

[19] Unless otherwise stated, the HierLS results correspond to HierLS-LM1.

Protocol	Proactive	Reactive	Sub-optimal
PF	–	–	$\Theta(\lambda_t N^2)$
SLS	$\Theta(\lambda_{lc} N^2)$	–	–
DSR-noRC	–	$\Omega(\lambda_s N^2)$ $O((\lambda_s + \lambda_{lc}) N^2)$	$\Omega(\lambda_t N^2 \log_2 N)$
HierLS	$\Omega(s N^{1.5} + \lambda_{lc} N)$	–	$\Theta(\lambda_t N^{1.5+\delta})$
ZRP	$\Theta(n_k \lambda_{lc} N)$	$\Omega(\lambda_s N^2/\sqrt{n_k})$	$O(\lambda_t N^2/\sqrt{n_k})$
HSLS	$\Theta(N^{1.5}/t_e)$	–	$\Theta((e^{\lambda_{lc} t_e K_4} - 1)\lambda_t N^{1.5})$

Table 1: Asymptotic results for several routing protocol for mobile ad hoc networks.

Protocol	Total overhead (best)	Cases
PF	$\Theta(\lambda_t N^2)$	Always
SLS	$\Theta(\lambda_{lc} N^2)$	Always
DSR-noRC	$\Omega(\lambda_s N^2 + \lambda_t N^2 \log_2 N)$	Always
HierLS	$\Omega(s N^{1.5} + \lambda_{lc} N + \lambda_t N^{1.5+\delta})$ $\Omega(s * N \log N + \lambda_{lc} * N + \lambda_t N^{1.5+\delta})$	LM1 or LM2 approach used LM3 approach is used
ZRP	$\Omega(\lambda_{lc} N^2)$ $\Omega(\lambda_{lc}^{\frac{1}{3}} \lambda_s^{\frac{2}{3}} N^{\frac{5}{3}})$ $\Omega(\lambda_s N^2)$	if $\lambda_{lc} = O(\lambda_s/\sqrt{N})$ if $\lambda_{lc} = \Omega(\lambda_s/\sqrt{N})$ and $\lambda_{lc} = O(\lambda_s N)$ if $\lambda_{lc} = \Omega(\lambda_s N)$
HSLS	$\Theta(\sqrt{\lambda_{lc} \lambda_t} N^{1.5})$ $\Theta(\lambda_{lc} N^{1.5})$	if $\lambda_{lc} = O(\lambda_t)$ if $\lambda_{lc} = \Omega(\lambda_t)$

Table 2: Best possible total overhead bounds for mobile ad hoc networks protocols.

independent of λ_t) with respect to traffic. HSLS follows scaling as $\Theta(\sqrt{\lambda_t})$, and PF, DSR, and HierLS are last since their *total overhead* increases linearly with traffic. [20]

Similarly respect to rate of topological change, we get that PF may be preferred (if size and traffic is small and the rate of topological changes increases too rapidly), since its *total overhead* is independent of the rate of topological changes. Provably next will be ZRP and DSR since their lower bounds are independent of the rate of topological changes. The bounds are not necessarily tights and ZRP and DSR should depend somewhat of the rate of topological change. Finally, for SLS, HierLS, and HSLS we know (as opposed to DSR and ZRP where we suppose) that they increase linearly with the rate of topological change.

It is interesting to note that when only the traffic or the mobility is increased (but not both), ZRP can achieve almost the best performance in each case.[21] However, if mobility and traffic would increase at the same rate; that is, $\lambda_{lc} = \Theta(\lambda)$ and $\lambda_t = \Theta(\lambda)$ (for some parameter λ), then ZRP's *total overhead* ($\Omega(\lambda N^{1.66})$ will present the same scalability properties than HSLS ($\Theta(\lambda N^{1.5})$) and HierLS ($\Omega(\lambda N^{1.5+\delta})$) with respect to λ, with the difference that ZRP does not scale as well as the other two with respect to size.

These and more complex analysis can be derived from the expression presented in this paper, when different parameters are increased at the same time accordingly with the scenario the designer is interested in.

References

[1] D. B. Johnson and D. Maltz,"*Dynamic Source Routing in Ad Hoc Wireless Networks.*", In Mobile Computing, edited by Tomasz Imielinski and Hank Korth. Kluwer Academic Publishers, 1995.

[2] S. Ramanathan, M. Steenstrup, "Hierarchically-organized, Multihop Mobile Networks for Multimedia Support", *ACM/Baltzer Mobile Networks and Applications*, Vol. 3, No. 1, pp 101-119.

[3] Z. Haas and M. Pearlman, "The performance of query control schemes for the zone routing protocol," in *ACM SIGCOMM*, 1998.

[4] R. Ramanathan and R. Hain, "Topology Control of Multihop Radio Networks using Transmit Power Adjustment," in *Proceedings of IEEE Infocom*, Tel Aviv, Israel, 2000

[5] http://www.ir.bbn.com/projects/dawn/dawn-index.html

[20] It is interesting to note that HSLS scale better with traffic intensities than HierLS (the only other protocol that scales well with size). This is mainly due because HierLS never attempts to find optimal routes towards the destination, even in slow moving scenarios. HSLS in the other hand, may eventually obtain full topology information – and therefore optimal routes – if the rate of topological changes is small with respect to $1/t_e$

[21] Almost, because ZRP can not achieve the independence of *total overhead* with respect to mobility. PF does.

[6] B. A. Iwata, C.-C. Chiang, G. Pei, M. Gerla, and T.-W. Chen. "Scalable Routing Strategies for Ad Hoc Wireless Networks". *IEEE Journal of Selected Areas on Communications*, vol. 17, no. 8, pp. 1369-1379, Aug. 1999.