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

Scaling Without Hierarchies. Or, 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

On this paper, the issue of scalability on ad hoc networks is addressed. We present a framework powerful enough to allow for the analysis of a wide variaty of protocols in the literature. We provide the first asymptotic results for a representative set of protocols in the literature: Plain Flooding (PF), Standard Link State (SLS), Dynamic Source Routing (DSR) [5], Hierarchical Link State (HierLS) [6], Zone Routing Protocol (ZRP) [7], and HSLS. These results show that HSLS presents the best scalability properties, even though it is much easier to implement than the other competitive protocols, namely HierLS and ZRP. In addition, the analytical results are complemented with a simulation experiment comparing HSLS and HierLS under a light-loaded scenario. An immediate contribution of this paper is to show that HSLS is a competitive, low-cost alternative to Hierarchical approaches for ad hoc networks. A longer term contribution of this work, is to improve our understanding of the limits and trade-off inherent to mobile adhoc networks.

1 Introduction

An ad hoc network is a wireless network where the nodes (possibly mobile) communicate (possibly using multiple hops) without the presence of an infrastructure. Ad hoc networks are suitable in situations where the network must be rapidly deployed and function without an infrastructure, such as military communications, disaster relief, campus networks, etc. . MANET networks are typically limited in terms of bandwidth, energy, processing capabilities, and security (among others).

Recent years have witnessed a surge in the interest in ad hoc networks. Spurred by ever-decreasing form-factors and cost of wireless transceivers and processors, a multitude of new applications are emerging. These include short-range ad hoc wireless networks for ubiquitous computing, larger range indoor wireless LANs that operate in ad hoc mode, metropolitan area networks (e.g. Metricom [1], Rooftop [2]), and sensor networks [3]. Standards such as Bluetooth, HomeRF, and IEEE 802.11 are giving impetus to the growth in the number of ad hoc communication enabled devices.

At the same time, recent technological developments regarding wireless transmission rates (11 Mbps wireless LANs (IEEE 802.11b) are here [4] and 54 Mbps (HiperLAN2 Forum) in the near future), battery life, and processing power will enable ad hoc networks to grow beyond their traditional limits (currently no more than tens of nodes), and to handle traffic intensive application as for example digital images and video. Thus, this new scenario (bigger size, higher capacity, higher traffic load) requires a review of the balance between data traffic and control overhead in traditional routing protocols.

Traditional solutions for large networks in fixed scenarios as the internet, relay on a hierarchical structure being imposed into the network. Hierarchical extensions have also been proposed for ad hoc networks. However, the inherent mobility of ad hoc networks requires complex hierarchy maintenance procedures (not required in their fixed network counterparts) that difficult their implementation. Also, the performance degradation imposed for such hierarchy maintenance is not well understood. *Is hierarchical routing the only (or the best) alternative to scale to large ad hoc networks?.*

In this paper, the above question is answered negatively by showing that a novel link state variant, the Hazy Sighted Link State (HSLS) protocol, presents scalabilty properties that are at least as good as the one obtained by hierarchical approaches. HSLS, developed in the context of DARPA's Density and Asymmetry-adaptive Wireless Network (DAWN) project [9], was first introduced in [11]. HSLS is by far much easier to implement than the hierarchical approaches.

Furthermore, analytical study of several protocols in the literature and the identification of HSLS as the one with better scalability (with respect to size , mobility, and traffic) provides a better understanding of the fundamental limit of ad hoc networks, and provides an insight into the answer to the timely question: *how scalable are ad hoc networks.*

In addition, the analytical results are complemented with a simulation experiment comparing

HSLS and HierLS under a light-loaded scenario. The simulation results points out that HSLS not only outperfroms HierLS in the extreme cases (high load, large networks) , but it may also outperform HierLS under lighter network conditions (where the asymptotic results do not really hold). In conclusion, for practical scenarios, the choice between HSLS and HierLS (ignoring implementation cost) is not trivial. Thus, an immediate contribution of this paper is to present HSLS as an attractive alternative to HierLS. A longer term contribution of this work, is to improve our understanding of the limits and trade-off inherent to mobile adhoc networks.

Our contributions include the following. We present a framework powerful enough to allow for the analysis of a wide variaty of protocols in the literature. We provide the first asymptotic results for a representative set of protocols in the literature: Plain Flooding (PF), Standard Link State (SLS), Dynamic Source Routing (DSR) [5], Hierarchical Link State (HierLS) [6], Zone Routing Protocol (ZRP) [7], and HSLS. These results show that HSLS presents the best scalability properties, even though it is much easier to implement than the other competitive protocols, namely HierLS and ZRP. In addition, the analytical results are complemented with a simulation experiment comparing HSLS and HierLS under a light-loaded scenario. The simulation results points out that HSLS not only outperfroms HierLS in the extreme cases (high load, large networks) , but it may also outperform HierLS under lighter network conditions (where the asymptotic results do not really hold). In conclusion, an immediate contribution of this paper is to show that HSLS is a competitive, low-cost alternative to Hierarchical approaches for ad hoc networks. A longer term contribution of this work, is to improve our understanding of the limits and trade-off inherent to mobile adhoc networks.

In particular, a unique feature of our work is that our results are derived from a mobility-based, probabilistic analytical model. Thus, our results have a broad applicability and help to understand the limits on scalability for large mobile ad hoc networks.

The next subsection review some previous work. The remaining of this section reviews the concept of *total overhead* that was first introduced in the prequel ([11]), where HSLS was derived. The definition of *total overhead* provides a framework powerful enough to allow the comparative evaluation of the different protocols. Roughly speaking, the lower the *total overhead* the greater the available capacity for data forwarding. Thus, the *total overhead* may be the single most important performance figure in a bandwidth constrained network. in a bandwidth constrained network. Although performance figures such as the delay, energy consumption, memory and processing requirements, may be as or even more important depending on the particular environment, this work focuses only on the *total overhead* induced for different protocols.

Section 2 presents the assumptions adopted for the analysis; these assumptions have 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 SLS protocol. Section 5 presents an asymptotic lower bound for DSR's *total overhead*. Sections

6 and 7 consider HierLS and ZRP respectively, and Section 8 briefly describes and analyzes the HSLS. Section 9 summarizes these results and completes this paper with some final remarks.

1.1 Previous work

There have been a vast amount of research into assesing the relative performance of routing protocols for mobile networks. Some of the most recent works include [13], [14], [15], and [16]. Unfortunately, ad hoc networks protocols have eluded a complete theoretical analysis, and most of the comparisons have been based on simulations. These simulation results, although extremely useful, fail to provide a deeper understanding of the limit of the protocols and their performance dependence in the network parameters outside the range being simulated. A notable exception is the work in [16] where analytical and simulation results are integrated in a comparative study. [16] provides valuble insight into different protocols relative performance. [16], however, fails to deliver a final analytical result, as the efect of sub-optimal routes is evaluated through simulations; making it difficult to totally understand the interactions among the different network parameters. In the present work, that gap is closed, since exact (asymptotic) expressions are derived. These expressions will provide the reader with the required understanding of the dynamic interaction of network parameters under the protocols studied.

This work can be though of as a continuation to the work in [17] and [18]. In those works, the capacity (ignoring the routing protocol overhead) of an ad hoc network was studied. These analysis points out that the achievable capacity is mainly limited by the length of the average source-destination pair. They both show that at a any given time, a number of simultaneous transmissions in the prder (a fraction of) the number of nodes in the network is achieved. However, a succesful packet delivery may require more than one (re) transmission. In [17], an static network was considered and therefore the average path length increases as rapidly as the square root of the number of nodes in the network. Thus, as the number of nodes increases, the bandwidth available for a particular source decreases (as rapidly as the square root of the network size). In [18] mobility with infinite memory and delay tolerance was considered, and therefore the average path length was reduced to 2. However, the delay will increase as fast as the decrease in average path length (that is, will now increase as the square root of te number of nodes in the network, since it is the time required for a node to get 'close' to a random destination after a packet for that destination has been received). Thus, since in practical scenarios we can not usually tolerate those high delays, imposing the condition that the packets be delivered as soon as they reach the head of the transmission queue will cause the average path length to return to the values obtained in [17]. This result (stated as assumption a.4) was fundamental in our analysis. In this paper, we complement the work in [17] and [18] by considering the protocol-induced overhead, that will limit the effective capacity for data transmissions.

1.2 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 expressed in terms of the number of packets exchanged between nodes, in order to maintain the node's forwarding tables up-to-date. In reactive approaches, *overhead* has been described in terms of the bandwidth consumed by the route request/reply messages (global or local). Efficient routing protocols try to keep the aforementioned *overhead* low.

While the above mentioned overhead is an important characteristic significantly the protocol behavior, they do not provide enough information of a protocol's behavior, it is not comprenhensive enough to lead to a proper performance assessment of a given protocol, since it fail to weight the impact of sub-optimal routes. As the network size increases to, say, 100 nodes; maintaining route optimality imposes an unacceptable cost under both the 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 because they try to maintain the current source-destination path for as long as it is valid, although it may no longer be optimal. In addition, 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 (range) of topology information dissemination (e.g. hierarchical routing [6]) and/or limiting the time between successive topology information updates dissemination (e.g GSR [10]).

Due to the above mentioned shortcomings, the authors revised the concept of overhead in[11]; so that it would also capture the effect of sub-optimal routes in capacity limited systems, since **sub-optimal routes not only increase the end-to-end delay but also result in a greater bandwidth usage than required**. This extra bandwidth may be comparable to that associated with the other sources of overhead. Although approaches that attempt to minimize the other sources of overhead may appear to be *scalable* by inducing a fixed amount of the aforementioned overhead, they may turn out to lead a poor performance due to increased overhead induced by the sub-optimal routes. Thus, a more effective definition of the overhead – which will be considered in the remainder of this work – was presented :

Definition : **The *total overhead* is equal to the total amount of bandwidth consumed in excess of the minimum required to forward data through the shortest distance (in number of hops) paths assuming that the nodes had instantaneous full-topology information**.

The above definition of *total overhead* provides for a common basis for comparing otherwise very different protocols. It is not claimed to be the *only* criterion for comparison among protocols, as additional criteria – such as end-to-end delay, or jitter – may be imposed by different applications. In the case of a bandwidth-limited network, and as the size and mobility of the network population increase, the remaining capacity (i.e. the total network capacity minus the *total overhead*) may well be the single most important performance metric. In addition, the *total overhead* expression defined above is directly related to the energy consumed by a node due to transmissions, which is an important (limiting) factor on battery-constrained system. [1]

The different sources that contribute to the *total overhead* may be grouped as *reactive*, *proactive*, and *sub-optimal route* overheads.

The *reactive* overhead of a protocol is the amount of bandwidth consumed by the specific protocol to build paths from a source to a destination, *after* a traffic flow to that destination has been generated at the source. In static networks, the *reactive* overhead 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* overhead is a function of both traffic *and* topology change.

The *proactive* overhead of a protocol is the amount of bandwidth consumed by the protocol in order to propagate route information *before* it is needed. This may take place periodically and/or in response to topological changes. Notice that the *proactive* overhead is the amount of bandwidth consumed (to propagate/build routes) that is not included in the *reactive* overhead.

The *sub-optimal route* overhead of a protocol is the difference between the bandwidth consumed when transmitting data from all the sources to their destinations using the routes determined by the specific protocol, 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 the source is not (yet) aware of the availability of a 3 hop path), then $(k-3) * packet\_length$ bits will have to be added to the *sub-optimal route* overhead computation. 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 $(N-3) * packet\_length$ bits will have to be added to the *sub-optimal route* overhead computation each time a packet is transmitted.

The above classification of the different sources of *total overhead* will prove useful in deriving the *total overhead* expressions in the next sections. First, the set of assumptions adopted in this work are introduced in the next section.

2 Network model

Let $\mathcal{G} = (\mathcal{N}, \mathcal{V})$ be a directed graph, where $\mathcal{N}$ and $\mathcal{V}$ are the sets of nodes and links in the network, respectively. Let $N = |\mathcal{N}|$ be the number of nodes in the network, d be the average in-degree, L be the average path length over all source destination pairs, λ_{lc} be the expected number of link status changes that a node detects per second, λ_t be the average traffic rate that a node generates in a

[1] However, if power consumption due to reception, processing, etc. is not neglectable, then the present work should be extended before it can be applied to energy-constrained networks.

second (in bps). The following assumptions, motivated by geographical reasoning, define the kind of scenarios targetted on this work: [2]

a.1 As the 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, and $\sigma = N/A$ be the network average density. Then, the expected (average) number of nodes inside an area A_1 is approximately $\sigma * A_1$.

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 (in hops) 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).

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

a.7 It is assumed that link status changes are due to mobility. λ_{lc} is directly proportional to the relative node speed.

a.8 Mobility models : time scaling. 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 $g_{t/t_1}(x, y, x_{t_1}, y_{t_1}) = \frac{1}{(t-t_1)^2} g_{0/1}(\frac{x-x_{t_1}}{t_1-t}, \frac{y-y_{t_1}}{t_1-t})$.

The first assumption (a.1) 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 would jeopardize the achievable network throughput[3]. It is achievable, because there are effective power control mechanisms currently available (see for example [8]). In general, a topology control algorithm should try to make the density as small as possible without compromising (bi)connectivity.

[2]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 $c_1 * g(n) \leq f(n)$ [similarly $f(n) \leq c_2 * g(n)$] for all $n \geq n_1$ [similarly, $n \geq n_2$]. Also, $f(n) = \Theta(g(n))$ if and only if $f(n) = \Omega(g(n))$, and $f(n) = O(g(n))$.

[3]See the topology (power) control results in DAWN [9]

The first assumption excludes random network models where the probability of a link between 2 nodes is fixed to a value p. In those networks, as the size increases, the average node degree ($N*p$) goes to infinity and the shortest path among the 2 nodes furthest apart approaches 2. Such networks are not believed to be representative of future mobile ad hoc networks.

The second assumption (a.2) 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. Thus, geographical reasoning may not define one hop connectivity (where multipath fading, obstacles, etc. are more important), but it obviously influences connectivity at larger scales. Thus, we can talk about the 'geographical' and 'topological' regions. In the 'geographical' (large-scale) region, geographical-based reasoning shapes routing decisions. In the 'topological' region, it is the actual – and apparently arbitrary – link connectivity (topology) what drives the routing decisions, and geographical insights are not useful.

Assumptions a.3 and a.4 are based on assumption a.2. For example, consider a circular area centered at node S of radius R with n nodes in it. Then doubling the area radius ($2R$) will quadruple the covered 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 doubled, and assuming that the transmission range (after power control) of the nodes does not change, then the distance (in hops) will also only double (on the average). Similarly, the 'boundary' area (where the nodes furthermost away from S are) will increase linearly (as the circumference of a circle does) with the radius.

Assumption a.5 and a.6 follow from the 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 find more web pages he/she would 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 spends on the internet.

he above arguments are somewhat debatable and may change. For example, the increase in size and content of the internet may prompt users to stay more time on-line, or perform more document 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 [6] and ZRP [7] will be greatly benefited. On the other hand, having a small set of destinations will favor DSR [5]. In general, the most demanding scenario is when all destinations are equally probable, and for this reason the analysis is focused in this case.

Assumption a.7 stresses our interest in analyzing mobile networks. In particular, this assumes that short-term variations in link quality may be buffered off by an effective link control mechanism, for example by requiring a high fading margin before declaring a link up (so, small oscillations will not affect connectivity), or by waiting for several seconds of link inactivity before declaring a link down (so that short-lived link degradation will not trigger link state updates). However, this may not necessarily be true, as radio behavior is quite unpredictable and long-lived link degradation is possible even if there is no mobility (e.g. due to rapidly varying multipath fading caused by small displacement, obstructions, rain, etc.).

Assumption a.8 is motivated by mobility models where the velocity of a mobile over time is highly correlated . For example, this is the case if the unknown speed and direction are constant. 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 consequently they impose a somewhat less demanding scenario for routing protocols. Once again, the study of this paper considers the most demanding scenario (that is, larger displacements) and assumes that the 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.

In the present work, only DSR without the route cache option (DSR-noRC) will be considered.

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

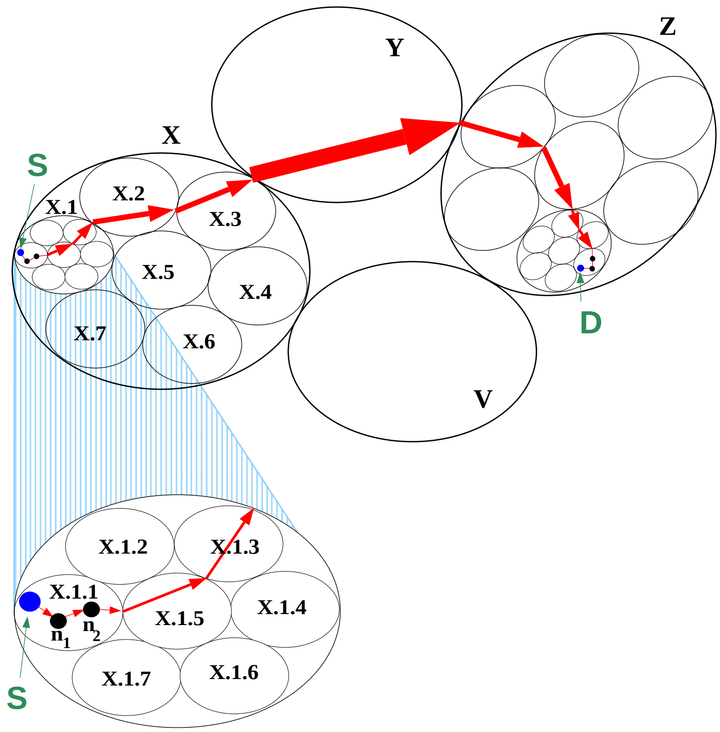

Figure 1: A Source (S) - Destination (D) path in HierLS, as computed for node S. Also, the high level link inside the destination cluster have been 'expanded'.

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 proactive (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.[4] 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

[4] 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

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 :

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

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 [5] – 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

[5] 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.

HierLS-LM1 [6] *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 $\Omega(s * N^{1.5} + \lambda_{lc} * N + \lambda_t N^{1.5+\delta})$. [7]

6.4 HierLS-LM2 total overhead

LM2 differs from LM1 in that:

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

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

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). [8] 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 $\Theta(k)$. [9] 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*

[8] 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.

[9] 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. $\Theta(k)$).

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 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

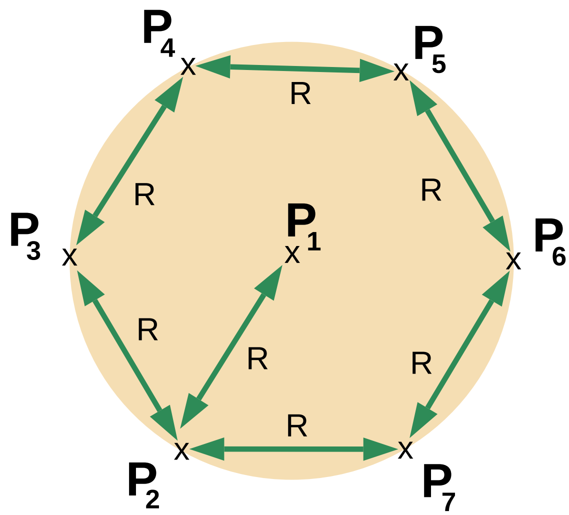

Figure 2: Geometrical interpretation of the maximum number of 'inhibitors' that 'cover' a node

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 $\Theta(\sqrt{n_k}) = \Theta(k)$, 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.[10]

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

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 $\Omega(\lambda_{lc}^{\frac{1}{3}} * \lambda_s^{\frac{2}{3}} * N^{\frac{5}{3}})$. [11]

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 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 the prequel ([11]), the HSLS protocol was introduced as the best algorithm among the family of FSLS approaches. HSLS 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 analysis in [11] hinted about the excellent asymptotic properties of HSLS, although the exact expression were not derived. That work was left to this paper.

In a highly mobile environment, the HSLS protocol 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 global LSU transmission – LSU that travels over the entire network, i.e. TTL field set to infinity – all counters are reset and the algorithm is

[11]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.

(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.[12] 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.

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 [13] when $s_i = 2^i$ (i.e. $s_1 = 2, s_2 = 4, s_3 = 8, s_4 = 16, ...$ 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 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.

Figure 3 shows an example of HSLS's LSU generation process when mobility is high and in consequence LSUs are always generated. Figure 3 assumes that the node executing HSLS computes its distance to the node farthest away from itself to be betwenn 17 and 32 hops, and therfore it replace the TTL value of 32 with the value infinity, resetting the algorithm at time $16t_e$. The reader is referred to [11] and [12] for more details about HSLS.

The analysis in [11] hinted about the excellent asymptotic properties of HSLS, although the exact expression were not derived. That task was left to the following subsection of this paper.

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

[12] 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.

[13] 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.

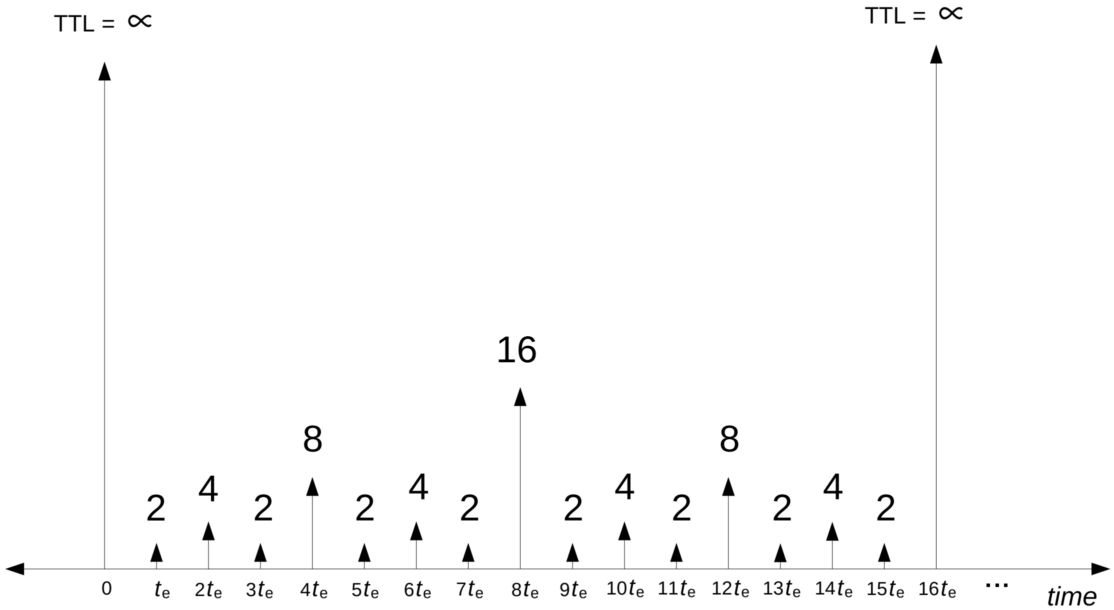

Figure 3: General LSU generation process for HSLS when relative node mobility is high, for a node whose distance to the node farthest away from itself is betwween 17 and 32.

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 (global 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 global 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 global LSUs, 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 $R_x/4, R_x/8, ..., 2, 1$. [14] Finally, the total bandwidth consumption due to all the LSUs generated by node X is equal to :

$$\begin{aligned}
X_{HSLS}^{proactive_cost} &= \frac{size_of_LSU * N}{R_x * t_e} + \frac{size_of_LSU * f_x * N}{R_x * t_e} + \frac{size_of_LSU * f_x * N}{2 * R_x * t_e} + \\
&\quad + \frac{size_of_LSU * f_x * N}{4 * R_x * t_e} + \ldots \\
&= \frac{size_of_LSU * N}{R_x * t_e}[1 + f_x(1 + \frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \ldots)] \\
&\approx \frac{size_of_LSU * N}{R_x * t_e}[1 + 2 * f_x]
\end{aligned}$$

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 $\Theta(\lambda_t * N^{1.5})$. [15]. 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, [16] is p. Moreover, consider

[14] 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.

[15] 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

[16] 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.

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).

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

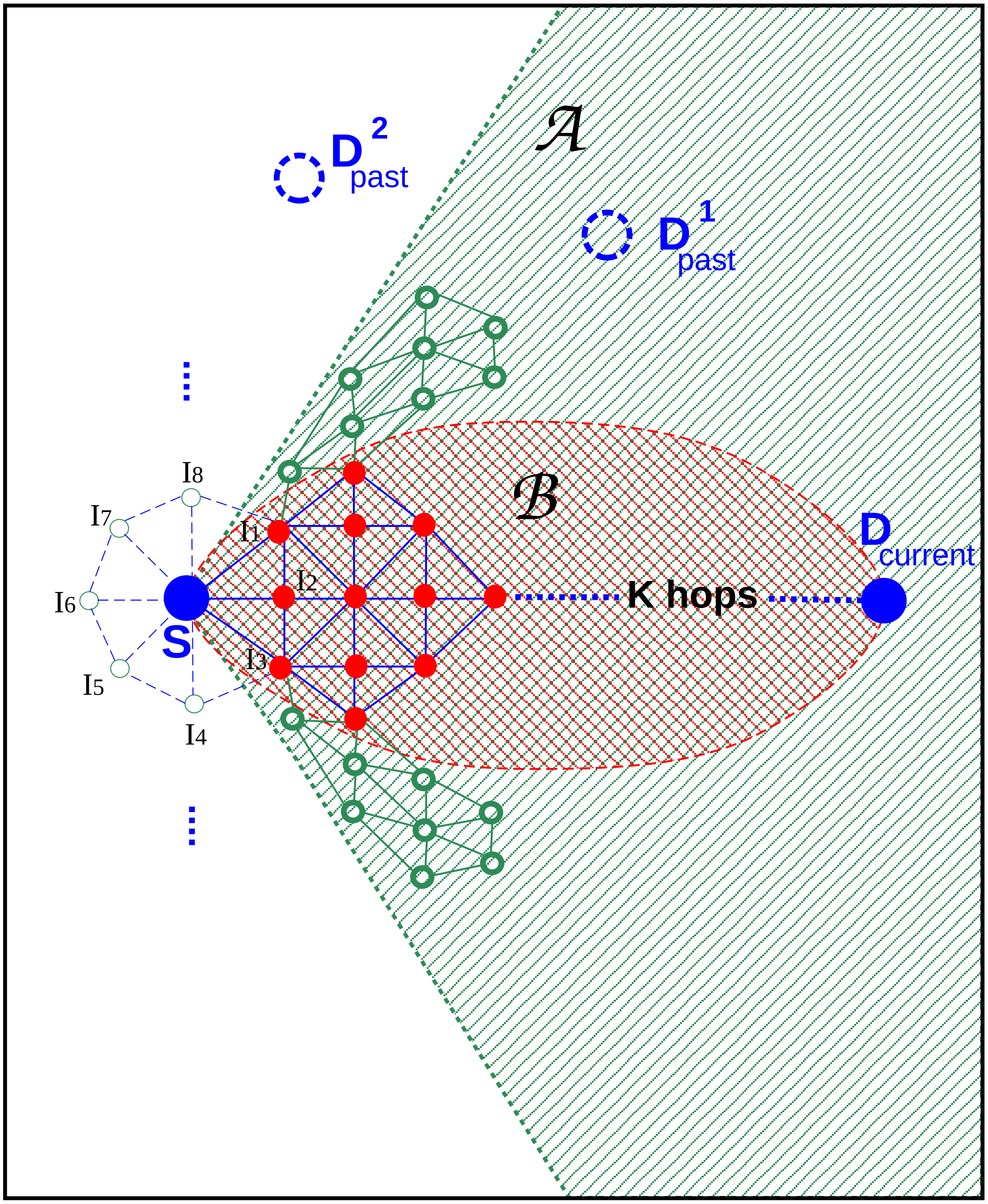

Figure 4: Good and bad next hop decision in the HSLS protocol

independent of the traffic. [17]

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 $\mathcal{A}'$.[18]

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

[17] 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 link failure detection.

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

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.

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). [19] 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.

[19] 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.

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 $\Theta(N^{1.5})$.

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 Comparative study

The previous sections results are summarized in Tables 1 and 2. Table 1 present the overhead results per source type (*proactive*, *reactive*, and *sub-optimal route*). [20] 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 derived before).

These results increase our understanding of the limits and provide valuable insight about the behavior of several representative routing protocols. The better understanding of these limits will help network designers to better identify the class of protocols to engage depending on their operation scenario.

[20] 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	$\Omega(sN^{1.5} + \lambda_{lc} N)$	–	$\Theta(\lambda_t N^{1.5+\delta})$
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	$\Omega(sN^{1.5} + \lambda_{lc} N + \lambda_t N^{1.5+\delta})$ $\Omega(s * N \log N + \lambda_{lc} * N + \lambda_t N^{1.5+\delta})$	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.

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 with respect to traffic. (*total overhead* is 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. [21]

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.[22] 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.

The authors were mainly interested in large networks, and therefore we focused our attention on HSLS and HierLS. It can be noted that HSLS has better asymptotic properties than HierLS, which means that as size increases HSLS eventually outperform HierLS. The idea of HSLS – being much more simple to implement – outperforming HierLS is counter-intuitive. A first reaction to this result will likely be to assume that the constants involved in the asymptotic analysis may be too large, preventing HSLS from outperform HierLS under 'reasonable' scenario. Thus, the authors relied on a couple of simulation experiment to validate if, in effect, HSLS may outperform HierLS even under moderate network size and traffic load.

9.1 A simulation experiment: HSLS vs. HierLS-LM1

Table 3 shows the simulation results obtained by OPNET for a 400-node network where nodes are randomly located on a square of area equal to 320 square miles (i.e. density is 1.25 nodes

[21] 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 $1/t_e$

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

per square mile). Each node choose a random direction among 4 possible values ($45^o, 135^o, 225^o$, and 315^o), and move on that direction at 28.8 mph (8 milimiles per second). Upon hitting the area boundaries, a node bounces back. The radio link capacity was 1.676 Mbps, and each source (there were 60) send data at the average rate of 2 packet per second (packet interarrival time was exponential). Each packet was 4 kbit long (to load the network without increasing simulation time too much). Thus, 60 8Kbps streams were generated at each second. Simulation were run for 350 seconds, leaving the first 50 seconds for protocol initialization, and transmitting packets/collecting statistic for the remaining 300 seconds. The HierLS approach simulated was of the type HierLS-LM1, and corresponds to the DAWN project [9] modification of the MMWN clustering protocol [6]. The minimum and maximum cluster size was set to 9 and 35 respectively.

The metric of interest is the throughput, interpreted as the fraction of packets successfully delivered. The simulation results presented are not a comprehensive study of the relative performance of HierLS versus HSLS under all possible scenarios. They just presents and example of a real-life situation where HSLS outperform HierLS, and complement our theoretical analysis. the theoretical analysis focuses in asymptotically large network, heavy traffic load, and saturation conditions where the remaining capacity determines the protocol performance. The simulation results, in the other hand, refer to medium size networks with light loads, where depending on the MAC employed, other factors may have more weight over the protocols performance.

The results in Table 3 show that HSLS may outperform HierLS in medium size, more realistic, scenarios. However, both protocols performance is quite poor. The above is a consequence of the MAC protocol being employed, which was unreliable CSMA. For the network load being induced, the non-neglectable probability of collision reduced the chance of packets reaching destination more than a few hops away. Thus, these results are for comparison sake only, since they suggest to use more elaborated MAC algorithms as the use of the RTS/CTS handshake.

Two main reasons contributed to HSLS outperforming HierLS for such a wide margin:

1 Since min-hop routing was used, the routing protocols tends to choose paths with 'longer' links (i.e. greater distance between the 2 nodes at each extreme of the link). As nodes move, these links deteriorate faster than the 'shorter' ones, and as a consequence packtes are being loss (unreliable MAC). HierLS has to wait until a 'degraded' link is declared DOWN before switching the packet transmission to different ones. HSLS, in the other hand, is benefited from quick feedback about a node one-hop neighborhood by eavesdropping the HELLO messages (beacons) exchanged by the neighbor discovery modules. Note that HierLS design philosophy prevents it from using such information, since it relies on all nodes inside the same cluster having the same view of the intra-cluster topology. HSLS, in the other hand, was designing under the assumption that nodes that are closer should be updated more frequently, so that having HELLOs messages interpreted as LSUs with TTL equal to 1 falls naturally into HSLS framework.

Protocol	Thoughput	Delay
DSR-noRC	$\Omega(\lambda_s N^2 + \lambda_t N^2 \log_2 N)$	Always
HierLS-LM1	0.0668	0.0134
HSLS	0.2454	0.0163
HSLS-2	0.1556	0.0141

Table 3: Throughput results for a 400-node network for HSLS and HierLS.

However, since some of HierLS shortcoming can be aliviated by techniques such as alternate path routing or by including 'stability' as a factor in the route selection, the authors tried to remove some of the bias towards HSLS. For this reason, HSLS-2 was also simulated. In HSLS-2, the routing protocol is prevented of eavesdropping the HELLO messages, and no level 1 LSU is transmitted. This was done for comparison sake only, and it is not the intention of the authors to propose such an aproach. LSUs with TTL equal to 1 , being inexpensive, improve significantly the protocol performance – as can be seen from the difference between HSLS and HSLS-2 in Table 3 – so they should always be transmitted. Instead, the author would propose to improve HierLS to address the previous issues and improve performance. Unfortunately, due to time-constraint, the aforementioned approach had to be implemented (i.e downgrading HSLS instead of upgrading HierLS). Even with the above modifications, HSLS-2 outperformed HierLS.

2 HierLS provided longer routes that made extremely difficult to the packets to reach their destination without colliding. HSLS also suffered from collisions, but the paths that HSLS provided for destination close by, tend to be smaller than the ones provided by HierLS. For example, when HSLS provided a 4 hop path, HierLS would provide a 6 six path (for a destination in a neighboring cluster). The extra path length (2 hops) may seem neglectable, but in a scenario where after 6 hops was almost certain that a packet would collide, it make a great difference. Since we were moderately loading the network, the probability of collision was high, and packet are not travelling more than 6 hops in average.

It can be seem that the previous results are highly influence for another factors such as the MAC protocol being used, the quality of the links that neighbor discovery declares up, the latency on detecting link failures, etc.

So, whether HSLS or HierLS should be preferred for a particular scenario, depends on the particular constraints (for example, if memmory or processing time is an issue, HierLS may be preferred since it require to store/process an smaller topology table). The present work, however, provides some guidelines, suggesting that as traffic, network size, and data rate increases, and a better MAC is employed (allowing to achieve the full channel capacity), HSLS should tend to be preferred.

10 Conclusions

We presented a powerful framework (the *total overhead* criteria) that allows for an analytical comparison, and better understanding, of routing protocols for mobile networks. This framework was first introduced in [11] to analyze a family of link state protocol variants, but it was extended here for application to a wide variaty of protocols in the literature.

We presented the first asymptotic results for the *total overhead* for several representative protocols. These results thread a new light into the understanding of the fundamental limits and trade-offs present in mobile networks in general, and in these protocols, in particular. The results facilitates comparison among otherwise quite diverse protocols.

Finally, our results for HSLS – a novel, easy-to-implement link state variant – show that the implementation of a complex hierarchy is not mandatory to scale to larger networks. A more focused comparison between HierLS and HSLS was undertaken, and as a result, we established HSLS as a competitive alternative to HierLS.

References

[1] http://www.metricom.com

[2] http://www.rooftop.com

[3] G. Pottie and W. Kaiser, "Wireless Sensor Networks", *Communications of the ACM, 2000.*

[4] http://www.zoom.com/zoomair/index.html

[5] 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.

[6] 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.

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

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

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

[10] 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.

[11] C. Santivanez, S. Ramanathan, and I. Stavrakakis,“Making Link State Routing Scalable”, To appear in *Proceedings of MobiHOC'2001*, Long Beach, CA, Oct. 2001.

[12] C. Santivanez, “Making Link State Routing Scalable”, *BBN Technical Report* ..... , Cambridge, MA, August 2001. Available at http://my-paper.html

[13] J. Broch, D. Maltz, D. Johnson, Y. Hu, and J. Jetcheva, “ A Performance Comparison of Multihop Wireless Ad Hoc Network Routing Protocols,” *Proceedings of MOBICOM'98*, Dallas, TX., October 1998.

[14] V. D. Park, and S. Corson, “ A Performance Comparison of the Temporally-Ordered Routing Algorithm and Ideal Link-State Routing,” *Proceedings of IEEE Symposium on Computers and Communications ISCC‘98*, Athens, Greece, June 1998.

[15] C. E. Perkins, E. M. Royer, S. R. Das, and M. K. Marina, “Performance Comparison of Two On-Demand Routing Protocols for Ad Hoc Networks”, *IEEE Personal Communications Magazine*, Vol. 8, No. 1, Feb. 2001.

[16] P. Jacquet and L. Viennot, ‘Overhead in Mobile Ad-hoc Network Protocols”, *INRIA Research Report 3965, Institut National de Recherche en Informatique et en Automatique (INRIA)*, France, June 2000.

[17] P. Gupta and P.R. Kumar. “The Capacity of Wireless Networks”, *IEEE Transaction on Information Theory*, 46 (2):388-404, March 2000.

[18] M. Grossglauser and D. Tse. “Mobility Increases the Capacity of Ad-hoc Wireless Networks”, in *Proceedings of IEEE Infocom'2001*, Anchorage, Alaska, April 2001.