

Redefining Link Duration: Making Routing Sensitive to Mobility

Namusale Chama*, Rute Sofia*
IANLab, SITI, University Lusófona

Abstract—This paper is focused on a discussion of parameters and heuristics that are expected to assist multihop routing in becoming more sensitive to node mobility. We provide a discussion concerning existing and a few proposed parameters. Moreover, the work also discusses two new heuristics based on the notion of link duration. The heuristics are compared based on a meaningful set of scenarios that hold different mobility properties.

Index Terms—multihop routing, mobility, wireless networks

I. INTRODUCTION

The most recent paradigms in wireless architectures describe environments where nodes present a somewhat dynamic behavior (e.g. *Mobile Ad-hoc Networks*, *MANETS*) or even a highly dynamic behavior (e.g., *User-provided Networks*, *UPNs*). Nodes in such environments correspond to wireless devices which are carried or controlled by humans and hence exhibit movement patterns which mimic the ones of humans - *social mobility patterns*. Moreover, in such environments, data transmission is based on multihop routing, namely, on *single-source shortest-path* approaches. In terms of metrics, the most popular multihop routing approaches rely on static link cost metrics such as hop count. The result is that when facing movement of nodes, multihop routing has its own shortfalls e.g., the need to recompute paths frequently if nodes exhibit high variability in movement of the nodes. In other words, current multihop approaches lack sensitivity in what concerns nodes movement. The routing metrics currently being considered cannot capture node speed variation or acceleration, node movement pattern, or even direction. Hence, a topology change is merely interpreted as a trigger to perform path re-computation. There are, however, cases where node movement may actually not represent a link break. Or, instead, a link change due to node movement may be so subtle that in fact it would not require any update to the topology, and hence, current approaches may result in useless path re-computation, of which the cost relates to additional signaling overhead and latency.

This paper is focused on a brief analysis of mobility impact on routing and proposes a category of routing metrics related to different notions of link duration. Our expectations are that such a metric may make multihop routing more sensitive to node movement and hence, assist in improving the trade-off of node movement vs. network efficiency. Under such category we provide several metrics and a discussion on the impact that they have, under different parameters.

The remainder sections are organized as follows. Section II goes over work that is related to ours, highlighting the relation between previous work, and our contribution. Section III gives an overview on the notion of node movement and how it impacts routing and the definition of a wireless link. Section IV gives an overview of the existing mobility tracking parameters. Section V describes our proposed routing metrics and how they improve routing sensitivity to node movement, while section VI provides an initial performance comparison of the proposed metrics. We conclude in section VII.

*{namusale.chama,rute.sofia}@ulusofona.pt. UITEC (Research Unit in Computational Technologies/Unidade de Investigação em Tecnologias Computacionais), Universidade Lusófona. Campo Grande 376, 1749-024 Lisboa, Portugal.

II. RELATED WORK

A number of approaches have been dealing with detection and measurement of accurate node mobility as well as counterbalancing mobility impact on routing. A first category relates to applying signal strength measurement at the receiver as a way to estimate the distance variation between two nodes, having as ultimate goal providing a way to build more robust paths. For instance, the *Mobility Prediction Routing Protocol (MAODV)*[8] considers the variation of the received signal power by a particular node to predict link breaks. Dube et al. [2] have also applied signal strength measurement as a way to build more robust paths. Manoj et al. applied the notion of received signal strength to track the distance variation between two nodes, in a specific time period [6]. Their distance change tracking approach is a desirable feature as it can capture movement of nodes and the impact on links. However, and similarly to the other work in this category, it does not track movement patterns.

A second category of work that tries to make routing more sensitive to node mobility relates to throughput variation measurement as a way to determine node mobility. For instance, Suyang and Evans have used the slope of change of throughput in a link vs. the link load to estimate topology changes [9]. Based on history, through throughput monitoring, a decrease according to Suyang and Evans means that there is a true change in the physical topology. They have attributed the changes to increase in the node distance and increase in interference. Even with good attributes of avoiding interference and detecting mobility collectively, node mobility individually has attributes that have not been addressed. Nodes may exhibit movement that does not necessarily impact the route stability.

Another category of work that we cite relates to an attempt that is not directly tackling mobility impact on routing in the sense of reducing computation but instead addresses a latency reduction. In such category falls the *On-Demand Multipath Distance Vector (AOMDV)* [7]; the work of Kim et al. [4] which considers disjoint paths in AODV to counter link failure in case the alternative path shares some links with the primary one. Albeit multipath assists in reducing latency and packet loss in the event of link failure, the cost of path re-computation is still present and affects the network operation.

A fourth category of work relies on *link sensing* as a measure of improving routing in terms of mobility sensitivity. Benzaid et al. [1] have proposed the *fast Optimized Link State Routing Protocol (Fast-OLSR)* whose basic idea is to detect link changes in a quicker way, by increasing the HELLO sending rate. Albeit interesting, such rate only assists in understanding that some nodes may be on the move, but not exactly which.

All of the mentioned approaches have in common the aspect that they fail in being able to distinguish between a topology change that is long-lasting, and a topology change that is so short that in fact it should not result in route re-computation. Our belief is that by defining a multihop routing metric more sensitive to node mobility, multihop routing can become more robust and better adjusted to the current wireless dynamic scenarios.

III. NODE MOBILITY IMPACT ON ROUTING

This section provides a brief discussion on the impact of node mobility on routing. *Node mobility* here refers to a change in speed and/or direction for a node. Hence, *node movement* embodies three main aspects: node position; speed; direction.

The impact of mobility on routing can be measured mostly by analysing the trade-off in robustness (e.g. the need to recompute paths more) vs. signaling overhead (more messages sent to quicker detect link breaks). Furthermore, node mobility impacts routing on a different number of ways and here we shall address the main ones, namely, relation to distance ; movement pattern (and how it affects links); relative movement (link remains stable due to similar movement of the nodes that compose the link); impact on the different stages of the routing process (e.g. route discovery and maintenance phase).

Intuitively, it seems that node movement can be better captured just by sending additional signaling messages. However, it is not always the case that by increasing signaling overhead one may better prevent link breaks and reduce the cost of re-computation; on the other hand, node movement patterns affect links temporarily or permanently.

To give a concrete example let us consider Figure 1 which illustrates a wireless topology where A and B exhibit some movement. The figure considers three different cases. In Figure 1 i), A exhibits confined movement eventually returning to its original position. In ii) B is the node moving in a *ping-pong* pattern, i.e. B is jumping back and forth between two different positions. The final case (cf. Figure 1 iii)) corresponds to the case where B permanently moves away from its original position. Upon movement of at least one of the nodes A and B, the corresponding link quality is affected. If the nodes exhibit frequent movement, frequent path re-computation may occur. For scenarios such as the one illustrated in Figure 1, it may happen that two neighbor nodes move away from each other resulting in a link being broken and consequently, resulting in topology re-computation, to return to their original position a few milliseconds later. Were it for a protocol truly adaptive to node movement, the decision on whether or not to trigger path re-computation should be based on metrics that can capture node mobility patterns.

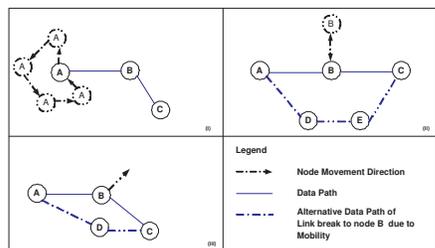


Figure 1. Examples of node mobility.

Node mobility impact on routing is also distance dependent, i.e., related to link *size*. For instance, a link formed by two nodes far apart (*long link*) can be affected and broken even by a small, insignificant node movement. If instead we have a *short link* (small distance between the two nodes), the movement of a node has to be significant to result in a link break. It should be noticed that link quality may degrade also due to mobility, but what we are highlighting here is that changes in distance are not sufficient to define a routing metric sensitive enough to node movement. It is also necessary to incorporate some sensitivity to a node's movement pattern and this is not a trivial task given the possible mobility patterns. For instance, a node moving between two different positions A and B can just move from A to B; move away from A to B and come back to A; or it can be ping-ponging between A and B. As mentioned previously, such movement

pattern may be insignificant in terms of impact on a link (e.g. because the distance between the nodes is short). In contrast, a move between A and B will impact link capacity heavily, and a ping-pong movement will result in a wide wireless link capacity variation. For both the aforementioned cases, a link is not truly broken and yet, today path re-computation would occur.

Another aspect to consider is on the routing process phase where movement is detected. Of utmost relevance is node mobility in the route discovery and route maintenance. Assuming the existence of a link during route discovery where one of the nodes is moving, it may happen that such a path may not be available anymore on the route maintenance phase.

Therefore, one main aspect to tackle in order to make routing more sensitive to node mobility is to consider metrics that are capable of capturing some properties of such movement. In the next section we describe a number of parameters and of metrics that can be used to achieve such goals.

IV. MOBILITY TRACKING PARAMETERS

In this section we describe several parameters related to the routing process which provide some support in terms of sensitivity to node mobility, explaining their advantages, as well as weaknesses. Link Duration or Lifetime

Link duration (LD) or *lifetime* is a parameter that is tightly related to the movement of nodes and is also as of today one of the parameters that is most popular in terms of tracking node mobility. By definition link duration is associated to the period of time where two nodes are within the transmission range of each other. In other words, it is the time period that starts when two nodes move to the transmission range of each other and that ends when the signal strength perceived by the receiver node goes under a specific threshold [5][12] [10]. Some authors then provide a variation of this definition by working the threshold value.

Today's definition of LD only assimilates node mobility in regards to its relation to signal strength. It fails, however, in terms of sensitivity to movement patterns. For instance, the current LD does not capture the case where a node jumps between its original position and a second position with a frequency that is not significant in terms of the potential delay it causes. Such movement will trigger repeated re-computation, which brings in more delay than if such frequent hopping would simply be disregarded. This can be in fact improved in a number of ways as shall be addressed in section V.

A. Node Degree Stability/Rate of Changing Neighbors

The *node degree* N_i of a node i corresponds to the number of neighbors i has at a particular instant in time. From a mobility perspective, an increase in node degree either means that other nodes moved towards node i , or that node i moved towards other nodes. It should be noticed that from our mobility analysis perspective, having nodes moving towards others is the same as having static nodes simply joining or leaving a network. Hence ND_i per se is not an adequate mobility tracking parameter. However, if one considers the variation of the node degree through time one may be able to infer some mobility properties. We define *Node Degree Stability* (NDS_i) for node i as a parameter that tracks the rate of ND changes. For the sake of clarity we provide a simplistic and initial embodiment of NDS_i in equation IV-A which corresponds to the difference between the ND at a previous instant $t - 1$ and the ND at the current instant t .

$$NDS_i = ND_{i,t-1} - ND_{i,t}$$

Let us consider the case where i is moving through clusters of other nodes. NDS_i captures, through time, the fluctuation of neighborhood

variation from the perspective of node i . Per se, it does not suffice to truly track mobility, given that i may be moving or instead, i may actually be static and its neighbors may move.

B. Average Number of Link Breaks

Another parameter that assists in tracking mobility dynamics is the *Average Number of Link Breaks* estimated in a specific interval for a node i , ALB_i [5]. If i experiences a high ALB then through time it may be a node to avoid, if the goal is to provide robust paths. This implies that it Albeit interesting due to the easy computation of such parameter, ALB can only assist in terms of the route discovery phase, given that it may assist in setting up more robust paths. However, and for the case of repetitive movement patterns, ALB cannot capture such mobility dynamics and will result in route re-computation.

C. Pause Time

Pause time is the period of time that the node is *stationary* (e.g. its speed is zero) [11]. Khamayseh et al. [3] have used pause time to determine mobility levels, by assuming that nodes with long pause times are less mobile than nodes with small pause times, and hence assist in developing more stable links. Their notion of pause time is based on an aggregate perspective and relative to the global time of a simulation. In real-time, nodes exhibiting short-pause times may or may not prevent the development of more robust links, but this is related not only to being static, but also to the movement pattern they exhibit.

D. Summary

Most of the mobility parameters previously discussed in this section are able to partially capture mobility dynamics of a node. Out of the ones described, LD seems to be the most relevant to consider in regards to attempting to develop routing metrics that can assist in tracking mobility dynamics in particular regarding node movement with patterns that exhibit some recurrent behavior (e.g. ping-pong movement). The remainder parameters are relevant and may be applied to assist parameters such as LD, in building more robust metrics.

It is, however, our belief, that LD requires a more thorough characterization to be able to integrate routing metrics that are more sensitive to movement.

V. MAKING ROUTING MOBILITY AWARE, LINK DURATION HEURISTICS

In this section we provide our proposal concerning heuristics that augment the current definition of LD in a way that optimizes the routing process, by being able to better capture when route re-computation is truly required.

A. Link Duration based on Signal-to-Noise Ratio Threshold

Today's LD definition simply relates to the receiver's perceived signal strength, as mentioned in section IV. As long as two nodes are within each other transmission range, then the LD keeps increasing [5]. If two nodes i and j share a short link and exhibit a synchronous mobility pattern thus keeping the same average distance while moving, then LD also increases. Let us now consider the same type of movement pattern between two nodes that are at a larger distance, i.e., the difference of behavior if a link is considered long or short. Micro-movement of nodes will impact longer links more than shorter links in what concerns route re-computation. However, metrics that just consider LD cannot capture this behavior.

Hence, this first definition relates to applying a threshold for SNR, so that LD becomes dependent of a "good" SNR level, that can assist in reaching a better trade-off in terms of route re-computation vs. signaling overhead. In other words, LD would become the instant in time that starts when two nodes encounter each other with SNR above the required threshold and which ends when the perceived SNR at the receiver is lower than a pre-defined threshold. This definition is evidently highly dependent on the choice of the SNR threshold.

B. Relaxed Link Duration: LD with a Tolerance Interval

One way to assist routing in becoming more tolerant to frequent and temporary movement the LD definition can be relaxed in regards to time. Hence we propose the *Relaxed Link Duration (RLD)* heuristic. Let us assume that a link has been stable since instant t_0 and breaks at an instant t_1 and that at instant t_2 the node that originated such break returns to its original position. With the current LD definition, the link would be considered broken at instant t_1 thus originating a route re-computation. In such case, $LD = t_1 - t_0$. At instant t_2 the route would again have to be recomputed. Following the same example provided, at instant t_1 the node detecting the link break would wait Δt time units before issuing a route re-computation. Assuming that Δt is large enough (for our example, larger than $t_2 - t_1$), then the node would return, on instant t_2 , to its original position and the routing process would disregard the temporary link break. The key aspect here is the choice of an adequate Δt . If Δt is too large, then for nodes that may cross frequently their original position, link breaks will not be detected and hence signaling is reduced. The flip-side to this is that for nodes that exhibit movement that result in a permanent link break, there is an additional delay added.

C. Spatial Stability-based Link Duration

In section IV we have introduced NDS_i and ALB_i . The combination of NDS_i and of ALB_i assists in capturing some properties of node mobility. To provide an example of how this can be achieved we define *Spatial Stability-based Link Duration (SSLD)* as an extension of LD based on a correlation between ALB_i and NDS_i . It should be noticed that the work here provided is intended to be initial and hence we do not provide a concrete instantiation of a formula that may represent SSLD. Instead, we explain the rationale for this heuristic.

Let us consider a node i that has a high NDS and a high ALB. In terms of node mobility and its impact on routing, links related to such node are expected to be less robust, given that there is a strong movement associated to the perspective of such node. Hence what SSLD can provide is a way to, at an early instant in time, discard successors of a node because they exhibit some mobility variability. As mentioned before, the node may be static and yet, the result are less stable paths (because most neighbors may be moving). For a more detailed explanation we provide in Table I the full set of NDS vs. ALB, by considering a high and low value. For each combination we provide examples of scenarios that may result in such values. Moreover, we also explain what the correlation may provide for each combination.

Scenarios where a node i holds both a low NDS (be it negative or positive) and low ALB imply that despite the reduction on the number of neighbors, the rate of link breaks is low. Potential cases for such result are scenarios where i is static and has a few nodes around it moving, or the node and its neighbors exhibit a group movement pattern. In both cases, the correlation between NDS_i and ALB_i can capture that mobility is not having particular impact on routing, for links associated to i .

Table I
NDS AND ALB CORRELATION FOR NODE i .

NDS_i	Low	Low	High	High
ALB_i	Low	High	Low	High
Examples resulting in such parameter values	<p>Case 1 A static node i has only a few mobile neighbors.</p> <p>Case 2 Node and its neighbours are moving exhibiting a group movement pattern.</p>	<p>Case 3 A static node i has a few mobile nodes which either exhibit ping-pong behavior or which are on the move.</p> <p>Case 4: A node i keeps a stable rate of change of neighbors based on different neighbors (and hence different links).</p>	<p>Case 5 Node i is static and experiences a significant change in terms of neighborhood.</p> <p>Case 6 Node i is mobile and experiences an increase in its neighbor degree, thus implying that there is a group movement pattern - old links are kept.</p>	<p>Case 7 A node moves towards a more dense region.</p> <p>Case 8 A node experiences a significant change in its dense neighborhood.</p>
What the correlation assists in capturing	- Realizes that there is low mobility impact - Ensures robustness of node as potential successor	- Realizes that there is significant mobility impact. - Node may be ruled out as potential successor.	- Indication of group movement pattern; - Stability in the neighborhood independently of having the node static or moving.	- High variability in terms of movement and of neighborhood.

Let us now consider a scenario with a low NDS and a high ALB value. This means that despite the fact that i has stability in terms of neighbors (i.e. it keeps a steady rate between neighbors leaving and joining), it can be captured that the neighborhood of i shows high variability. In contrast, if only NDS would be considered, then it would be assumed that i was a node stable in terms of links and hence a robust choice for a successor on a path. Also, if only ALB would be considered on such a scenario, then it would be immediately assumed that i was a node to rule out.

As example of scenarios which result in a high value for NDS and low value for ALB we provide cases 5 and 6. For both cases i sees an increase in neighbors (new nodes are joining); yet, such change has no impact in terms of links already established. If i is a static node (cf. case 5) then this means that its neighborhood is mobile in its majority. If instead i is moving, there is a clear group movement pattern, given that there is no increase in terms of link breaks.

The final discussion here provided relates to having both a high NDS and a high ALB . Potential examples of scenarios where this occurs are cases 7 and 8 are where node i is respectively experiencing a sudden and significant change in terms of neighborhood or moving through a dense region.

In the next section we provide a brief analysis related to the performance comparison for the heuristics described in this section.

VI. PERFORMANCE COMPARISON ANALYSIS

This section provides an analysis on the performance comparison for the heuristics proposed on section V. We consider a specific set of scenarios and evaluate the different heuristics in terms of route re-computation in the form of signaling overhead reduction, delay reduction, as well as improved throughput. As underlying scenario for this comparison we consider Figure 1 main cases: i) confined movement to a specific region where node a is making confined movements; ii) ping-pong movement as a result of periodic node mobility of node b ; iii) free movement in a large region. The confined movement example stands for a case where the node exhibits some features that may be captured, as happens in social mobility models, but its mobility is confined to a specific region and hence there may be some probability of returning to origin. In this case it is also likely that most of the node's movements do not truly impact the links it is part of severely. In other words, movement does not always imply a link break. The second case stands for an example of movement of a node which returns to the original position frequently. The third case represents an example of movement where the node will not return to origin.

In addition to the movement patterns based on Figure 1, we also consider other parameters in our analysis. From a topology perspec-

tive we shall analyse the different cases assuming different distances for links, namely, short and long links, given that as explained, movement of nodes impacts differently the network depending on the distance of nodes to neighbors. We also consider *network density*, namely, low or high density. A third parameter that we take into consideration is *movement frequency* in the form of *low frequency* (the node moves seldom thus exhibits long pause times) and *high frequency* (the node is highly dynamic; pause time is short).

In Table II we provide a summarized comparison of the performance of each heuristic, namely, LD based on today's definition (benchmark), and the two proposed heuristics: LD with a tolerance interval, as well as LD based on Spatial Stability.

The table contemplates the three scenarios (confined movement category, ping-pong movement, free movement) where we have considered the combination of network density and movement frequency by relying on the two extreme values *low* and *high*. For each scenario, we also consider what may happen if the topology is mostly based on short or long links.

A. Confined Movement Scenario

We start by discussing the performance of the heuristics in confined movement scenarios where the majority of links are short. Being a confined movement scenario, the nodes involved move in a specific personal space revolving around their initial position and hence are expected to return or to pass on its original position with some frequency. Such frequency is strongly related to the link stability.

For a low density network where most nodes are static (movement frequency is low), as shown in the first line of table II, then the expected result when relying on LD is that the path re-computation is normally triggered due to permanent link breaks. With the increase of movement frequency (cf. Table II second and fourth row), path re-computation occurs due to link breaks. A first aspect to highlight is that the LD performance in terms of routing degradation is heavily dependent on movement frequency, and pattern. A second aspect relates to the fact that LD behavior will impact routing more heavily for dense networks. Overall, LD performance is dependent on the link distance and hence the impact on routing is expected to be more significant for topologies where links are in their majority long.

Looking into the first proposed heuristic, RLD (cf. Table II rows 5 to 8), for the case of confined movement. At a first glance and in what concerns simpler topologies (low density, mostly static nodes) RLD seems to have a similar performance at the cost of a slight increase in delay, which depends on the choice of the tolerance interval. It should be noticed that RLD will experience an additional delay, but when LD is applied, there is also a delay increase due to path re-computation. Therefore, even though there is potentially an

increase in delay depending on the choice of the tolerance interval, we expect RLD to behave better given that it reduces the need to recompute paths and hence improves throughput and reduces the delay associated to path re-computation. Such improvement is not so significant in simpler topologies but becomes significant when the movement frequency increases, given that the RLD is less sensitive than LD to movement frequency.

In regards to SSLD, for simpler topologies the expected performance is similar to LD, for this scenario. The reason for this is that SSLD does not incur particular delay (in contrast to RLD) and the variability in terms of ND and of link breaks is low. When the movement frequency increases, then this heuristic works better by being able to more quickly track that there is variability in terms of movement affecting the link.

B. Ping-Pong Scenario

This is a scenario where LD does not suffice to provide routing with adequate stability in the face of movement. The ping-pong frequency is highly related to the link stability. If it is too short (e.g. a few milliseconds) it will originate frequent link breaks. If it is long, then it affects routing less, when considering LD. Hence, when LD is considered this type of movement results in significant throughput reduction and increase in delay, as well as additional signaling due to the route re-computation.

By considering RLD it is possible to decrease the impact of mobility on routing, if the tolerance interval is adequately tuned. The result is an increase in performance due to lesser path re-computation. Another aspect that seems interesting to be further explored in RLD is that it is an heuristic that seems to be less sensitive than LD to movement frequency.

SSLD is also expected to offer a good improvement in comparison to LD, for this scenario. The correlation between NDS and ALB gives the means to detect that there is some repetition in terms of neighborhood. When combined to the linear ALB, it is feasible to realize that the node is moving in a repetitive pattern and hence to take a more intelligence action in terms of the decision to recompute.

When comparing the three heuristics in this scenario for short links vs. long links, we believe that the behavior is similar to the previous scenario: there is an overall performance degradation for the case of long links, given that distance increases the need for MAC contention.

C. Free Movement Scenario

In the free movement scenario we consider the case where a node is moving away from its original position and not expected to return. The way it moves away (pattern) also impacts the movement. In this type of situation LD suffices to ensure that path re-computation is only triggered when a permanent link break occurs. When we consider RLD, then the expected performance is similar to the one of LD if the topology is mostly static but when the movement frequency increases and with the increase in network density then the behavior of RLD becomes worse as it adds delay to the moment when a link breaks. It should be noticed that such delay, albeit always expected to be present, can be tuned through time to become less significant. An exception to this may be the situation where a node moves with a slow motion and stopping on the way for long pause times. In some cases, LD may result in path re-computation that can be avoided an instant later in time (due to a slow motion node, pausing also for long times).

Considering SSLD, our analysis tells us that when considering simpler topologies, this heuristic seems to behave in a similar way to LD. But when there is an increase in the movement frequency then SSLD may assist in providing an action on the right instant. Moreover

and in comparison to RLD, there is no significant delay added when applying SSLD. As for the three types of scenarios, SSLD seems to be the better suited for the confined movement scenario.

VII. CONCLUSIONS AND FUTURE WORK

This paper addresses ways to make multihop routing more sensitive to movement. We introduce the problem space of impact of mobility on routing and discuss a few existing parameters and also some novel parameters that may be considered to track mobility in routing. Based on the analysis and discussion of such parameters we propose heuristics based on the notion of link duration to attempt to make routing more sensitive to node mobility. Such heuristics are then compared against the current definition of link duration for a meaningful set of scenarios.

Albeit being initial work that requires further delving, the provided comparison hints that the two heuristics being proposed, namely, the RLD and the SSLD, are relevant enough to be considered as potential candidates to assist multihop routing in terms of mobility sensitivity. While the RLD seems to be more relevant for scenarios that exhibit some repetitive motion pattern, SSLD seems to provide an overall good performance, in particular for confined movement scenarios.

As future work we intend to further detail the two proposed heuristics, and to provide an evaluation of them against LD for the most popular forms of multihop routing (distance vector and link-state approaches).

REFERENCES

- [1] M. Benzaid, P. Minet, and K. Al Agha. Integrating fast mobility in the olsr routing protocol. In *Mobile and Wireless Communications Network, 2002. 4th International Workshop on, IEEE, 2002*.
- [2] R. Dube, C.D. Rais, Kuang-Yeh Wang, and S.K Tripathi. Signal stability-based adaptive routing (ssa) for ad hoc mobile networks. *Personal Communications, IEEE, 4:36–45, 1997*.
- [3] Y. Khamayseh, O. Darwish, and S. Wedian. Ma-aodv: Mobility aware routing protocols for mobile ad hoc networks. In *Systems and Networks Communications, 2009. ICSNC '09. Fourth International Conference on, pages 25–29, 2009*.
- [4] J. Kim, S. Moh, I. Chung, and C. Yu. *Advanced Web and Network Technologies, and Applications*, chapter Robust Multipath Routing to Exploit Maximally Disjoint Paths for Wireless Ad Hoc Networks, pages 306–309. Springer, 2006.
- [5] Q. Liang and T. Kunz. Mobility metrics to enable adaptive routing in manet. In *Wireless and Mobile Computing, Networking and Communications, 2006. (WiMob'2006). IEEE International Conference on, pages 1–8, 2006*.
- [6] B.S. Manoj, R. Ananthapadmanabha, and C.S.R. Murthy. Link life based routing protocol for ad hoc wireless networks. In *Computer Communications and Networks, 2001. Proceedings. Tenth International Conference on, IEEE, pages 573–576, 2001*.
- [7] M.K. Marina and S.R. Das. On-demand multipath distance vector routing in ad hoc networks. In *Network Protocols, 2001. Ninth International Conference on, IEEE, pages 14–23, 2002*.
- [8] L. Meng, W. Fu, Z. Xu, J. Zhang, and J. Hua. A novel ad hoc routing protocol based on mobility prediction. *Inform. Technol. Journal, Asian Network for Scientific Information, 7:537–540, 2008*.
- [9] Ju. Suyang and J.B. Evans. Mobility-aware routing protocol for mobile ad-hoc networks. In *Communications Workshops, 2009. ICC Workshops 2009. IEEE International Conference on, pages 1–6, 2009*.
- [10] C. Tsao, T. Wu, W. Liao, and J. Kuo. Link duration of the random way point model in mobile ad hoc networks. In *Wireless Communications and Networking Conference, 2006. WCNC 2006. IEEE, volume 1, pages 367–371, April 2006*.
- [11] J. Tsumochi, K. Masayama, H. Uehara, and M. Yokoyama. Impact of mobility metric on routing protocols for mobile ad hoc networks. In *Communications, Computers and signal Processing, 2003. PACRIM. 2003 IEEE Pacific Rim Conference on, volume 1, pages 28–30, Aug 2003*.
- [12] Y. Wu, W. Liao, C. Tsao, and T. Lin. Impact of node mobility on link duration in multihop mobile networks. *IEEE Transactions on Vehicular Technology, 58:2435–2442, June 2009*.

Table II
COMPARISON OF THE LD HEURISTICS.

Heuristic	Network density	Movement frequency	CONFINED MOVEMENT		PING-PONG		FREE MOVEMENT	
			Short links	Long links	Short links	Long links	Short links	Long links
LD	Low	Low	- Path re-computation hardly triggered without need	- Path re-computation hardly triggered without need but error probability higher than with short-links	-Some path re-computation -Some throughput reduction and delay increase. - only affects parts of the network	-Some path re-computation -Some throughput reduction and delay increase - Worse performance overall than in short-link	- Path re-computation triggered without need if node speed is low.	- Path re-computation triggered without need if node speed is low.
	Low	High	- Path re-computation triggered without need - Movement frequency dictates performance degradation	- Path re-computation triggered without need - Movement frequency dictates performance degradation	- Path re-computation triggered without need - Movement frequency dictates performance degradation - only affects parts of the network	- Path re-computation triggered without need - Movement frequency dictates performance degradation - Worse performance overall than in short-link	- Path re-computation triggered without need if node speed is low. - Link break hard to predict/anticipate	- Path re-computation triggered without need if node speed is low. - Link break hard to predict/anticipate
	High	Low	- Path re-computation hardly triggered without need	-Path re-computation hardly triggered without need - Slightly worse performance overall than in short-link	-Some path re-computation	-Some path re-computation - Worse performance overall than in short-link	- Path re-computation triggered without need if node speed is low. - Plenty of alternative paths	- Path re-computation triggered without need if node speed is low. - Link break hard to predict/anticipate
	High	High	- Path re-computation hardly triggered without need -Throughput reduction and delay increase due to MAC contention	-Heavy path re-computation -Heavy throughput reduction and delay increase due to MAC contention - Worse performance overall than in short-link	-Heavy path re-computation -Heavy throughput reduction and delay increase due to MAC contention	-Heavy path re-computation -Heavy throughput reduction and delay increase due to MAC contention - Worse performance overall than in short-link	- Path re-computation triggered without need if node speed is low. -Plenty of alternative paths	- Path re-computation triggered without need if node speed is low. - Link break hard to predict/anticipate
RLD	Low	Low	- Similar performance to LD but slight increase in delay	- Similar performance to LD but slight increase in delay	- slight improvement when compared to LD	- slight improvement when compared to LD	- Slightly worse performance than LD	- Slightly worse performance than LD
	Low	High	- Good improvement compared to LD - Less sensitivity to movement frequency	- Good improvement compared to LD - Less sensitivity to movement frequency	- Good improvement compared to LD	- Good improvement compared to LD - Worse performance overall than in short-link	-Good improvement to LD if node speed low and node pause time is high and frequent	-Good improvement to LD if node speed low and node pause time is high and frequent
	High	Low	- Good improvement compared to LD but slight increase in delay	- Good improvement compared to LD but slight increase in delay	- Good improvement compared to LD	- Good improvement compared to LD - Worse performance overall than in short-link	- Similar to LD	- Similar to LD
	High	High	- Good improvement compared to LD - Less sensitivity to movement frequency	- Good improvement compared to LD - Better sensitivity to movement frequency	- Good improvement compared to LD	- Good improvement compared to LD - Worse performance overall than in short-link	-Good improvement to LD if node speed low and node pause time is high and frequent	-Good improvement to LD if node speed low and node pause time is high and frequent
SSLD	Low	Low	- Similar performance to LD	- Similar performance to LD	- Similar performance to LD	- Similar performance to LD	- Similar performance to LD	- Similar performance to LD
	Low	High	- Slight improvement compared to LD - No delay added	- Slight improvement compared to LD -No delay added	- Good improvement compared to LD	- Good improvement compared to LD	- Slightly better performance than LD	- Slightly better performance than LD
	High	Low	- Slight improvement compared to LD - No delay added	- Slight improvement compared to LD - No delay added	- Slight improvement compared to LD - No delay added	- Slight improvement compared to LD - No delay added	- Slightly better performance than LD	- Slightly better performance than LD
	High	High	- Good improvement compared to LD - No delay added	- Good improvement compared to LD - No delay added	Slight improvement compared to LD - No delay added	- Slight improvement compared to LD - No delay added	- Slightly better performance than LD	- Slightly better performance than LD