A Robust Path Metric for Mobile Wireless Networks

TIK Report 273

Rainer Baumann*, Simon Heimlicher*, Vincent Lenders† Martin May*
*Computer Engineering and Networks Laboratory
ETH Zurich, Switzerland
†Department of Electrical Engineering
Princeton University, NJ, USA

Abstract— **In this paper, we propose and evaluate a path metric to select robust paths in mobile wireless networks. This path metric selects paths that offer a good trade-off between long routes consisting of many reliable links and short routes consisting of only few but unreliable links. In contrast to the widely used minimum hop count metric, which chooses arbitrarily among those paths with minimum length, our path metric estimates the reliability of a link based on the measured received signal strength (RSSI). Every node determines a preferred signal strength and assigns each neighbor a link cost depending on how close to this strength its RSSI value is. Using simulations with realistic mobility traces of a large city, we compare our path metric to the minimum hop count metric. With nodes moving at car speed, using our metric increases the average route lifetime by a factor of 7 and reduces the packet loss ratio due to route breaks during communication by a factor of up to 15.**

I. INTRODUCTION

Wireless networking is expected to continue to grow in popularity, leading to abounding connectivity in urban areas. Routing in such dense networks involves selecting among multiple paths the one with the most desirable characteristics. Routing protocols determine the path to use based on a routing metric. The most commonly used metric is minimum hop count, which selects a arbitrary path among those with the minimum number of hops to the destination. This routing method is also called *shortest path routing* and can be represented by a link cost metric that assigns every link a cost of 1; leading to a route cost equal to the number of hops.

Shortest path routing is useful under the assumption that there is no measurable difference in the quality of the links. In fixed networks, this assumption is reasonable, but in mobile wireless networks, the quality of links varies widely and is subject to many influences, e.g. (i) the geographical length of the link, (ii) physical disturbances the link is exposed to (including interference and obstacles), and (iii) the relative speed of the end points of the link. Hence, the shortest path route may involve unreliable wireless links and break quickly, leading to packet loss and a connection interruption until a new route is established.

As proposed earlier, paths can be selected based on more than only their hop count by assigning each link a cost figure that reflects the quality of the link. The straightforward method to rate wireless links is to consider their *Received Signal Strength (RSSI)* value, which can easily be obtained from most wireless interfaces. In order to avoid links that are very long or cross areas of high interference, links with low RSSI are assigned high cost. However, selecting paths based on this simple link metric leads to routes with a large number of short links. Even though the individual links appear to be very reliable, the probability of a route break increases with the number of hops. Furthermore, every hop increases the end-to-end delay and incurs additional transmissions that add to the interference level.

Thus, selecting paths based on the RSSI values of their links is a trade-off between long paths over many reliable links and short paths over few unreliable links. We propose a link cost metric that differs from earlier proposals as follows. Our metric not simply prefers the strongest links, but rather uses a cost function that essentially rates the signal strength in relation to a pre-determined *preferable signal strength (PSS)*. However, even if every node selects the link with the lowest cost, the determined end-to-end path may not be the most robust. Therefore, we develop a path metric based on our link metric that assigns the lowest cost to the most robust path. For practical applications, this path metric can for instance be incorporated in a distance vector routing protocol, but it could also be used in other contexts.

In order to demonstrate the effectiveness of our path metric, we compare the routes it selects with the shortest path routes using the network simulation software Glomosim. We use a static and two mobile scenarios with node speeds comparable to pedestrians and cars, respectively. To derive as realistic results as possible, the movement of pedestrians and cars follows the vectorized street map of a real city with around half a million citizens. Our results show that the probability of route breaks is decreased considerably compared to shortest path routing. In the car speed scenario, the average route lifetime is increased by a factor of 7 and the packet loss ratio (fraction of packets lost due to route interruptions) is reduced by up to a factor of 15.

This paper makes the following main contributions. First, we introduce a link cost metric that allows to select robust links in mobile wireless networks. Second, we provide a path metric to select robust end-to-end paths based on our link metric. Third, we present an implementation of those metrices in a simple proactive distance vector routing protocol.

The rest of this paper is organized as follows. In the following section, we highlight related work. Then, we introduce our metrics in Section III. The implementation of our metrics in a routing protocol is discussed in Section IV. In Section V, we describe the simulation environment we use for our experiments, and in Section VI we present the results of the evaluation. We conclude the paper in Section VII.

II. RELATED WORK

In this paper, we propose a new link metric for mobile wireless networks that allows a routing protocol to select paths that are more robust than those selected by the minimum hop count metric.

Even in fixed wireless networks, the bad performance of shortest path routing has been reported in different contexts. For example, Aguayo et al. [2] observe that in the Roofnet wireless mesh network, links typically have a packet delivery ratio that occupies the whole range between 0 and 1. As a consequence, a minimum hop count metric often chooses lossy links that cause a lot of retransmissions and degrade the end-to-end performance. Yarvis et al. [6] observe that hop count performs poorly as a routing metric for sensor networks.

One way to discriminate links is to consider the packet loss ratio. De Couto et al. proposed ETX [7], a metric that maximizes the expected transmission count instead of minimizing the hop count. Draves at al. [8] compared this metric in a static wireless indoor network and confirm an increased end-to-end throughput. Others have proposed to completely avoid links with a loss ratio above a certain threshold [15].

Aside from these metrics based on measured packet loss ratio, several signal strength-based routing protocols have been proposed. Examples are ABR [21], SSR [9], and RABR [1]. The common idea of these protocols is to prefer stable links over transient ones. Therefore, these protocols select paths that are based on links that had a strong signal strength for a long time in the past. In contrast, our link cost metric assigns lowest cost to links with a signal strength that is closest to a certain value.

Cheng and Heinzelamn [5] explore different design trade-offs to maximize the route lifetime in ad hoc networks. However, they do not propose specific mechanisms or metrics that can be applied by a routing protocol.

An alternative way to cope with frequent path failures is to maintain alternative routes such that end-to-end connectivity can be re-established quickly. Preemptive routing [11] and pro-active route maintenance [19] have been proposed to determine an alternative route when a link is expected to break (e.g., indicated by low received signal strength) on an active route. Multi-path routing protocols [14], [17], [22] establish multiple routes to a destination and thus have immediately an alternative route when the active route breaks. All these approaches are complementary to the metric we propose in this paper and they might well benefit from using our metric to determine path cost.

In [18], Naumov et al. propose to use the received signal strength for a discrete rating scheme of links. This idea was employed for controlled flooding applications. In contrast, we propose in this paper a continuous link rating scheme and incorporate this scheme into a routing metric to increase route robustness in mobile wireless networks.

III. METRIC DESIGN

In this section, we present the concept of our link cost metric which prefers certain neighbors based on their received signal strength[1]. We first explain how to derive a meaningful average of the signal strength and then we describe how this average is used to calculate the link cost metric.

A. Link Cost Metric

The aim of preferring certain neighbors over others for routing data is to minimize the probability of link breaks and thus also minimize the route break probability. In mobile wireless networks that route according to the shortest path paradigm, link breaks due to mobility are very common. The reason is that the distance of two neighbors is optimized such that it is as close to the maximal communication range as possible and thus the number of hops is minimized. Thus, links selected by shortest path routing are likely to break as soon as the nodes move away from each other.

Estimating the distance between neighbors without location data is usually done based on the received signal strength (RSSI) since this measurement value provides an indication about the quality of a link and is easily obtainable from most wireless interfaces. However, the RSSI value is susceptible to fluctuations due to fading and similar temporary disturbances. In order to mitigate the effects of random signal fading, we use an averaging technique widely used in telecommunication: Exponential Weighted Moving Average (EWMA).

We calculate the average signal strength, $\overline{s}_{i,k}$, for neighbor k considering the packet with index i according to Alg. 1. By $U := [u_1, ..., u_{|U|}]$, we denote the set of neighbors and by $S_k := [s_{k,1}, ..., s_{k,|S|}]$, the set of signal strengths measured from the packets received from neighbor k. The smoothing factor α determines the weight of each measured value $s_{i,k}$ and reduces the weight of a single measurement, if there are many current measurements available. n_k denotes the number of packets received from neighbor k during the last second.

Using the average signal strength $\overline{s}_{i,k}$, we calculate the link cost according to the cost function defined in Alg. 2. Note, that we have also run simulations with exponential cost functions. But since we have not found a parametrization that outperforms the simple linear function, we use the linear function for all simulations reported in this paper. The linear function consistently delivers high performance, is easy to understand, and its parameters can be chosen intuitively as shown in Section III-B.

---

[1] By *received signal strength* of a neighbor we refer to the strength of the signal that is received *from* this neighbor.

Algorithm 1 Exponential Weighted Moving Average of Signal Strength.

```
for $u_k \in U$ do
    $\alpha = \frac{n_k}{n_k+1}$
    for $s_{i,k} \in S_k$ do
        $\overline{s}_{i,k} = (1-\alpha)\ \overline{s}_{i-1,k} + \alpha\ s_{i,k}$
    end for
end for
```

Algorithm 2 Calculation of link cost c_x.

```
if $\overline{s}_{i,k} > PSS$ then
    $c_k = \frac{PSS-\overline{s}_{i,k}}{PSS-rxThresh} \cdot (c_{out} - c_{pss}) + c_{pss}$
else
    $c_k = \frac{\overline{s}_{i,k}-PSS}{rxMax-PSS} \cdot (c_{in} - c_{pss}) + c_{pss}$
end if
```

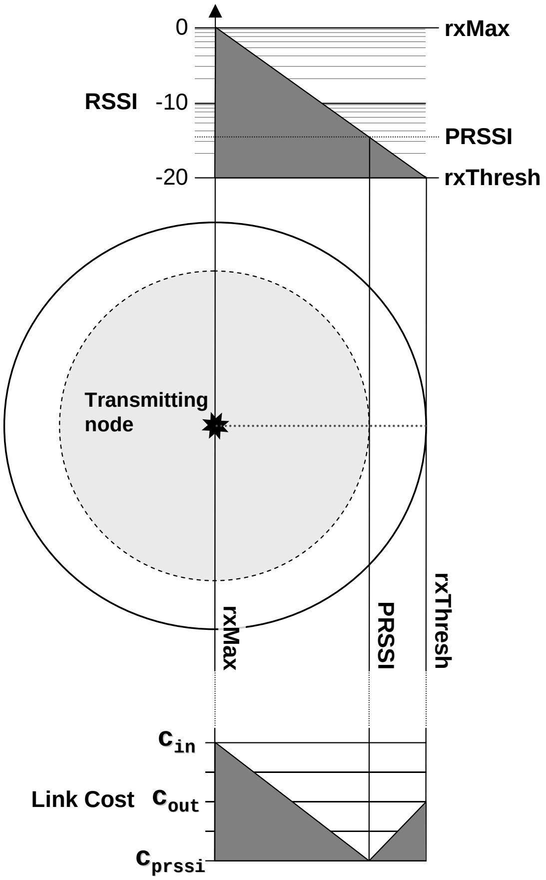

Fig. 1. Dependence of the Link Cost Metric on the Signal Strength. *Note: for simplicity, we depict the communication range (rxThresh) of a node and the power level of the preferred signal strength as circles, but our algorithm by no means assumes that the range is indeed circular.*

Our linear cost function has three parameters: c_{pss}, c_{out}, and c_{in}. These parameters determine, which link cost is assigned to a link with a given signal strength. By defining c_{pss} as lower than c_{out} and c_{in}, links whose received signal strength is close to the preferred signal strength (PSS) will have minimal link costs corresponding to highest preference, as depicted in Fig.1. Links with much lower or much higher RSSI will be assigned higher cost, indicating lower preference.

The preferred signal strength parameter depends on the scenario and in particular on the mobility of the nodes. At high node speeds, a value close to the maximum signal strength (rxMax) should be chosen because links that do not have a high signal strength are likely to break quickly. In such an environment, paths consisting of several short links provide better performance.

In contrast, if nodes move slower, the preferred signal strength should be closer to the minimal signal strength that still allows successful reception (rxThresh value). The reasoning is that at low speed, the probability that two nodes move outside the communication range is much lower, even neighbors with low signal strength may remain within range for a considerable period of time. Hence, the routing performance is better if paths with lower hop count are used.

B. Robust Path Metric

After describing our link cost metric, we show how it can be used as a path metric. An optimal path with respect to the link cost metric comprises exactly those nodes whose received signal strengths is very close to the preferred signal strength. An example is depicted in Fig. 2. Node C is exactly at the PSS of node D and node B is exactly at the PSS of node C. However in reality, it is unlikely that every node will always have a neighbor that is located exactly at the preferred signal strength range. And if there exists such a path, it might not lead to the destination on a straight line but might involve detours across a large number of links. In most cases, the robustness of such an overly long path would be no higher than the robustness of the shortest path.

Therefore, we propose a path metric that provides a trade-off between the preferred signal strength and the hop count to the destination. Our definition of the optimal path p_{opt} from the set of possible paths L, is the path that minimizes the sum of its link costs:

$$p_{opt} = p_l : \min_{p_l \in L} \sum_{k \in p_l} c_k \tag{1}$$

This definition is commonly used in networking. For example, the minimum hop-count metric uses the same definition except that the link costs c_k are set to 1 for all links.

To illustrate the impact of this new path metric on the link selection, we plot in Fig. 3 the path chosen by this metric and compare it with the path that would be chosen by the minimum hop-count metric. This path is one hop shorter, since our path metric trades in low hop count for more reliable links in order to increase the robustness of the path.

C. Cost Parameter Selection

The robustness of a route depends mainly on how good the trade-off between hop count and hop reliability fits the scenario. The link cost function proposed in Alg. 2 can be parametrized to optimize the path selection for many scenarios.

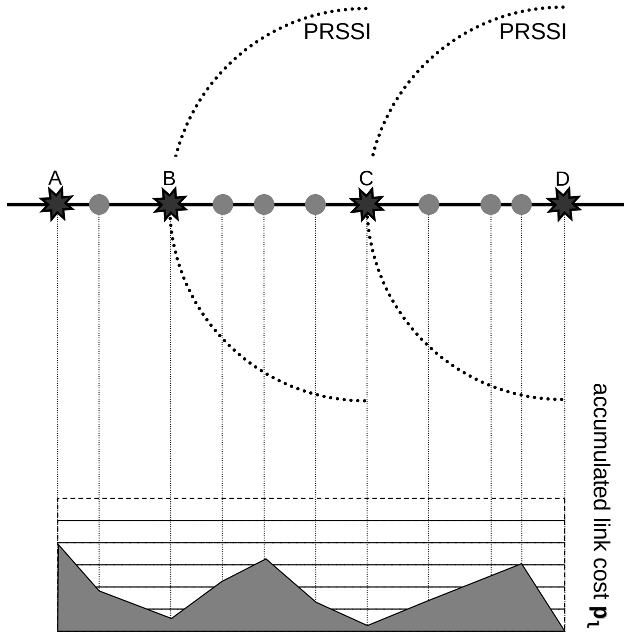

Fig. 2. Path Costs for a Route from A to D using our Link Cost Metric.

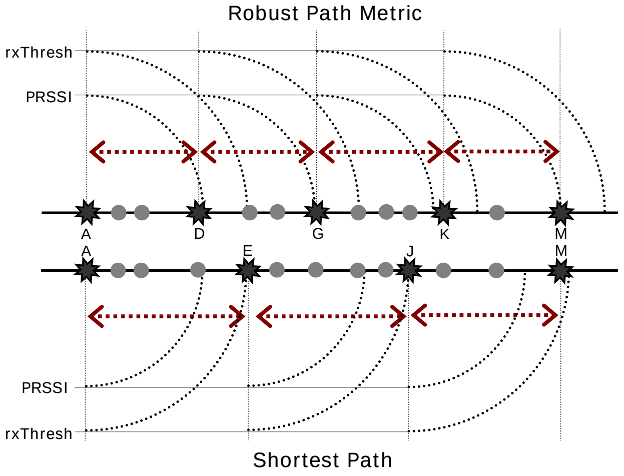

Fig. 3. Comparison of Route Selection; Source A, Destination E.

As a case study, we describe how to optimize the parameters c_{pss}, c_{out}, and c_{in} for a scenario with uniform distribution of link breaks with the following probabilities. For nodes at the border of the communication range, the link break probability be $b_{out} := 40\%$; for nodes that are next to each other, the link break probability be $b_{in} = 1\%$; and for nodes at the preferred signal strength, the link break probability be $b_{pss} := 10\%$.

In order to minimize the route break probability we have to set c_{pss}, c_{out}, and c_{in} according to the following consideration. The probability of a route break should be the same for a path over several links with preferred signal strength as for a path with much fewer links where the nodes are at communication range distance. Assuming uniform distribution of link breaks, the probability of a route break can be approximated as follows. Let b be the link break probability and h be the number of hops. Then the probability of a route break is $r = 1 - (1 - b)^h$.

Since the proposed cost function is linear and the costs over a path are accumulated, the cost parameters c_{out}, c_{in}, c_{pss} can be interpreted as the number of hops. Inserting the parameters and the route break probabilities into the above equation leads to the following system of equations:

$$
\begin{aligned}
(1 - b_{pss})^{c_{pss}} &= (1 - b_{out})^{c_{out}} && (2) \\
(1 - b_{pss})^{c_{in}} &= (1 - b_{in})^{c_{pss}}. && (3)
\end{aligned}
$$

By arbitrarily defining $c_{pss} = 1$, we find

$$
\begin{aligned}
c_{out} &= log_{(1-b_{out})}(1 - b_{pss}) \approx 5 && (4) \\
c_{in} &= log_{(1-b_{pss})}(1 - b_{in}) \approx 10. && (5)
\end{aligned}
$$

Thus, in our example, the probability of a route break is the same for a path with five links with preferred signal strength as for a path with one long link where the nodes are at the maximal communication range distance: $5c_{pss} = c_{out}$.

IV. IMPLEMENTATION

In order to compare the effect of link and route metrics with shortest path routing, we have implemented a proactive distance vector routing protocol. We have deliberately chosen a very basic implementation to minimize the influence of protocol specific features. We will first briefly describe the distance vector routing algorithm and then discuss the link and path metric implementation in more detail.

A. *Exchange of Distance Vector*

In our basic proactive distance vector routing protocol, every node maintains the distance for every known destination. Note that in the context of distance vector routing, the term distance corresponds to the cost of a link or a path. The distances are computed based on periodically exchanged messages called distance vectors. A distance vector is a table that provides destinations and their distances. To avoid the typical problems of distance vector routing (loops and the count-to-infinity problem), we use a widely established method called split horizon with poison reverse [16]).

In our implementation, all nodes build and maintain a data structure called *neighbor table* that contains the distance vectors received by neighbors and a timestamp of the last update.

Based on the neighbor table, nodes maintain their own distance vector. Whenever an entry is added, removed, or changed in the neighbor table, the node re-computes its distance vector. There are essentially three cases that necessitate an update of the distance vector:

1) New neighbor: If a distance vector from an unknown neighbor is received, a corresponding entry is added to the neighbor table. In addition, the distance vector of the node is re-computed.
2) Maintain neighbor: If the distance of a known neighbor to a destination changes, the distance of this node to this destination is recomputed.

3) Missing neighbor: If no distance vector is received from a neighbor for a certain period, its entry is removed and the distance vector of the node is re-computed.

Based on the distance vector, routing packets is straightforward and implemented on a hop-by-hop basis: A packet is always forwarded to the neighbor with the shortest distance to this destination.

B. Implementation of Robust Path Metric

The implementation of our path metric is split into two modules. The *link cost module* maintains the link cost to all neighbors based on the link cost metric, as described in Section III.

The *path cost module* maintains the distances to the destinations and provides the corresponding next hop. The distance c_d to a destination d is calculated in two steps. First, for every neighbor n_k, the sum of the distance to this neighbor, c_k, and the distance to the destination reported by this neighbor, $c_{d,k}$, is calculated. Second, the neighbor c_{k*} with the lowest distance is chosen as the next hop and stored in the distance vector along with the associated cost c_d. In the rest of the paper, we will refer to this path selection method as *route preference* scheme.

To compare our solution with shortest path routing, we also implement a path cost module using hop count as the path metric. The only difference to the algorithm presented above is that a constant link cost of 1 is used instead of c_k.

Since some implementations of distance vector routing protocols have a rather low limit for the distance, they cannot be used with our path metric. However, the effect of preferring links at the preferred signal strength can still be shown, as follows. The distance c_d to a destination d is calculated as with the route preference scheme. However, the distance that is stored in the distance vector along with the chosen next hop is simply the sum $c_{d,k*} + 1$. In the following, we refer to this route selection scheme as the *link preference* scheme. Including this link preference scheme in the evaluation shows clearly the advantage of our route preference scheme over link preference schemes with a local view proposed in the literature discussed in Section II.

C. Parameter Selection

The preferred signal strength value should be set according to the maximal expected movement speed of the nodes and the update interval of the deployed routing protocol. To determine the preferred signal strength, we consider two nodes being at this signal strength and moving away from each other with the maximal expected speed. The preferred signal strength should now be chosen such that these two nodes do not lose their connection (reach the maximal communication range) during the next two update intervals of the routing protocol. This allows the routing protocol to (i) detect the leaving node and (ii) fix the affected paths. The preferred signal strength is calculated individually per node based on the receiver sensitivity (rxThresh) of the deployed hardware, thus ensuring hardware independent operation.

Parameter	Value
c_{pss}	10000
c_{out}	50000
c_{in}	100000
PSS	rxThresh + 10dB

TABLE I
PARAMETERS.

We assume according to the established traffic speed regulation for cities a maximal node speed of 40 mph. The PSS is calculated for two routing update intervals of 1 second. The detailed parameters for the calculations of preferred neighbor routes are listed in Tab. I. Note, that since the distance values are implemented as integers, we scaled the link costs c_{pss}, c_{out}, and c_{in} by a factor of 10000 to increase the precision and to avoid rounding effects. As the distance vector update interval, we use 1 second.

V. SIMULATION SETUP

To evaluate and compare the performance of our approach, we performed simulations with Glomosim [24], a network simulator for wireless networks. As a reference for the performance of our path metric, we use the minimum hop count metric. The settings and assumptions of our experiments are described in this section; the results are given in Section VI.

A. Radio Settings

Our simulations are based on a WiFi network. All nodes are equipped with a 802.11b radio with a bandwidth of 11 $Mbps$ and a nominal range of 250 meters. As MAC layer protocol we use the 802.11 DFWMAC-DCF w/RTS/CTS and as propagation model two-ray ground. Due to the large network sizes, we were unable to simulate the effect of intermediate buildings in our city scenarios. However, we expect that the trends of our results also hold when such obstacles are present.

B. City Mobility Model

The literature shows that the results of performance studies for ad hoc networks depend heavily on the chosen mobility model [4], [23] and that simple mobility models like random way point [3] are not very accurate. However, real traces are available for the movement of pedestrians and vehicles in cities. Saha and Johnson proposed to use street maps from the TIGER database to generate realistic mobility patterns [20]. However, they do neither account for speed limits nor adequately consider the exact geometry of the streets. In contrast, we use detailed vectorized street maps with complete speed limit data.

The road maps of these cities are extracted from a geographic information system (GIS) that includes vectorized building and street maps with speed limit data. The vectorized map of the center of the city that we use in this paper is shown in Fig. 4.

The actual node movement is modeled according to the steady-state random trip mobility model [12] on the vectorized maps. That is, a node chooses a random destination in

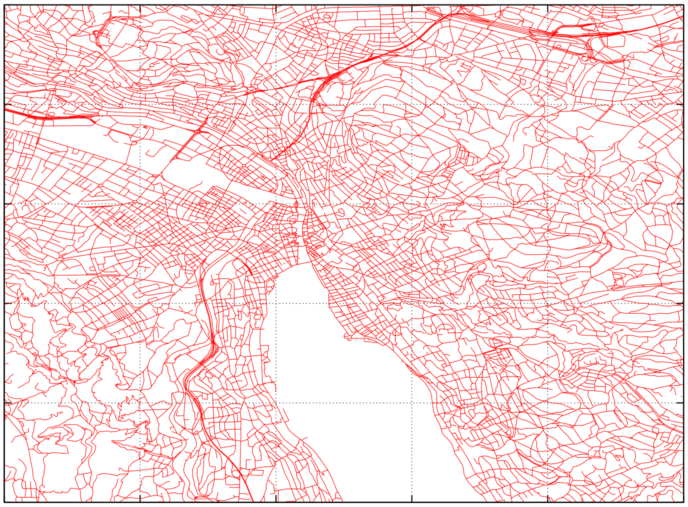

Fig. 4. Vectorized street map of a city (10km by 10km).

the city and moves to this position with a constant speed along the shortest path. The model captures the behavior of pedestrians as well as cars since both move along the streets of a city. We generate three scenarios: (i) a scenario with mobile nodes moving at pedestrian speeds (i.e., node speeds that are uniformly distributed between 2 mph and 9 mph), (ii) a scenario including nodes moving at car speeds according to the maximal allowed speed on that street (i.e., $3/4\ max\_speed \leq car\_speed \leq max\_speed$, usually 20 mph – 40 mph), and (iii) a static scenario for model verification and benchmarking. In each scenario, we model 5000 nodes on an area of 10km by 10km in the investigated city.

C. Traffic Pattern

We assume that wireless networks are mostly used for applications like web browsing, messaging, chatting, etc. Therefore, we rely on an Internet traffic model as used in [10] consisting of a half-half mix of streaming and web-like traffic. Streaming traffic has a bidirectional constant bit rate of 64 kb/s and the duration of streams are exponentially distributed with an average of 480 seconds. Web-like traffic consists of sporadic 1 kB requests according to an exponentially distributed inter-request time with an average of 10 seconds, followed by response messages with a message size that is Pareto II distributed [13] (average 12 kB, minimum 0.1 kB, maximum 1000 kB).

All simulations have 500 nodes generating data traffic, a duration of at least 10000 seconds, and are always an average over at least 20 runs with different random seeds.

VI. SIMULATION RESULTS

In this section, we show the results of the evaluation of our *route preference* scheme. We compare this scheme with the *link preference* scheme and with shortest path routing. We expect that the route preference scheme outperforms the shortest path approach by a significant margin, especially in highly mobile scenarios. In the subsequent sections, we will provide detailed results, based on the following metrics.

- *Packet loss ratio* – Ratio between the number of lost packets and the number of packets sent by the source.
- *Path length* – Average path length of a route (number of hops).
- *Route break probability* – Average probability that a route breaks within a second. This corresponds to the probability that a packet is forwarded to a neighbor that is no longer in the communication range of a node.

In highly dynamic networks, such as wireless mobile networks, the routing performance is mainly measured by the packet loss ratio and the path length. We deliberately do not consider the route delay because this metric heavily depends on the traffic load and the packet loss ratios. Since packets on long routes with high delay are typically lost by protocols with high loss rate, those protocols would have a shorter average delay and protocols with low packet loss ratio would be penalized by a delay metric.

For the optimization goal of minimal route break probability, there is a trade-off between path length and link break probability. However, since there is no direct correlation between link break probability and end-to-end packet loss ratio, we do not discuss the characteristics of individual links further. Instead, we focus on the route break probability, i.e. the probability that a route breaks at *any* link.

A. Packet loss ratio

In Fig. 5, we plot the packet loss ratio against the speed of the nodes. In the static scenario, there are no link breaks, consequently, there is no significant difference among the three schemes. However, already at pedestrian walking speed, noticeable differences emerge. As expected, the differences are most pronounced at car speed. Here, the link preference scheme has a packet loss ratio of 35%, which is around one half of the ratio of shortest path routing. The route preference scheme leads to an even lower packet loss ratio of 8% – only one quarter of the packet loss ratio of link preference and less than one tenth of the shortest path method's loss ratio. These results indicate two things. Firstly, the distance between the nodes of a route is critical for communication in highly mobile networks. Secondly, optimizing these distances individually helps, but balancing these distances along the whole path is even more effective to minimize the overall packet loss ratio, as demonstrated by our route preference scheme.

The ability to select neighbors is only useful if there are multiple neighbors to choose from. In order to understand the influence of the node density, we analyze the scenario with nodes moving at car speed in more detail. The packet loss ratio is plotted against the node density in Fig. 6. Obviously, if the node density is 6 or more nodes per communication range, the packet loss ratio does not decrease anymore. Again, the packet loss ratio of shortest path routing is the highest and the ratios of the link and route preference schemes are again much lower – by as much as a factor of 2 and 10, respectively.

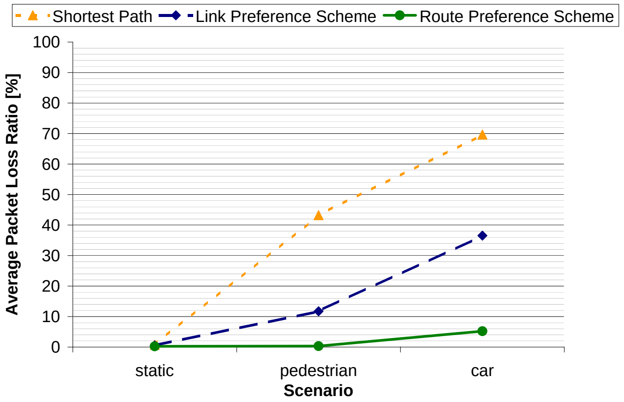

Fig. 5. Packet loss ratio.

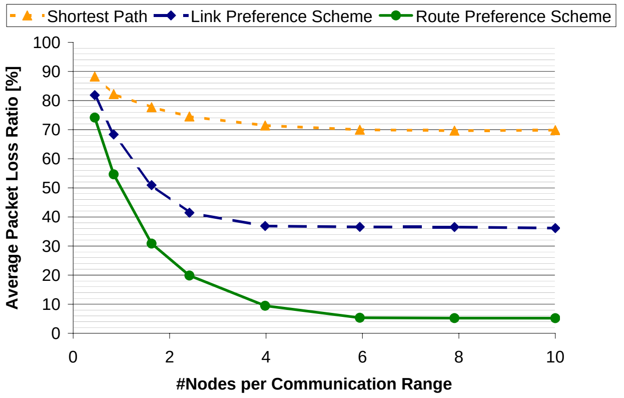

Fig. 6. Packet loss ratio vs. node density with nodes moving at car speed.

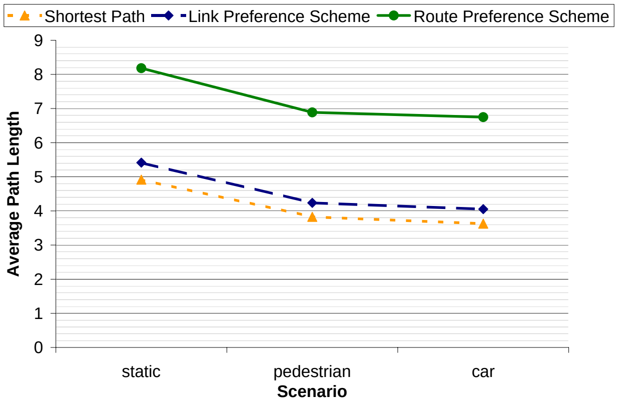

Fig. 7. Path length.

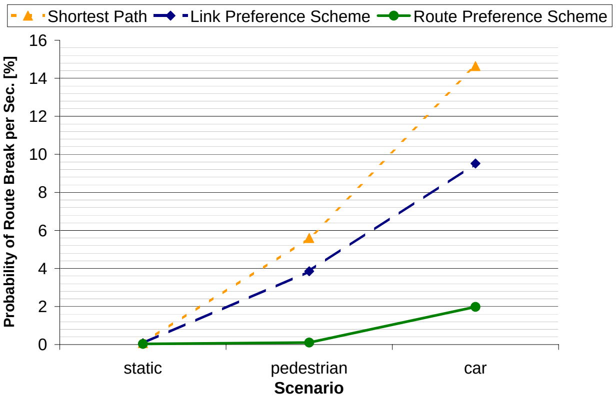

Fig. 8. Route break probability.

At very low node densities of 3 or less nodes per communication range, the packet loss ratio increases almost exponentially not only with shortest path, but also with the link and route preference schemes. This is not surprising given that the freedom to select among multiple neighbors evades as the node density decreases, leaving not much room for optimization.

B. Path length

Path length correlates with the end-to-end delay and also has an influence on the route break probability. In Fig. 7, we plot the path lengths for the scenarios used previously in Fig. 5. The shortest path routing method optimizes for this metric and is obviously the optimal case here. Any scheme that uses nodes that are not on one of the shortest paths is bound to extend the path length. This plot shows that the route preference scheme incurs much longer routes. However, the links used by those routes seem to be much more reliable, such that the route preference scheme still leads to substantially more robust routes and lower packet loss ratio as shown previously in Fig. 5.

C. Route break probability

The robustness of routes manifests itself in the packet loss ratio. In Fig. 8, we plot another metric that corresponds to route robustness: the probability that a route breaks within the next second. As expected, in the static case, no difference is visible. However, as the node speed increases, the effect of preferring certain neighbors becomes apparent. What is surprising is that the route preference scheme, which has the longest paths of the three schemes, has by far the lowest route break probability. At car speed, the probability of a route break is almost a factor of 7 lower than with shortest path routing.

Consequently, the average route lifetime of the route preference scheme is the highest – around 8 times higher than with shortest path (see Tab. II).

VII. CONCLUSION

In mobile wireless networks, the problem with shortest path routing is that the routes often involve links at the edge of the communication range that are likely to break due to interference or node mobility. In contrast, in this paper,

Routing scheme	*Average route lifetime*
Shortest path routing	6.8s
Link preference scheme	10.5s
Route preference scheme	50.5s

TABLE II
AVERAGE ROUTE LIFETIME WITH NODES MOVING AT CAR SPEED.

we propose to select paths that provide a good trade-off between their hop count and the reliability of their links. We introduce a new link cost metric for mobile wireless networks. Furthermore, we propose a path metric that allows to select robust paths based on our link metric. Compared to shortest paths, the paths selected by our metric exhibit a lower packet loss ratio and a lower break probability because they consist of more reliable links.

Our link cost metric gives preference to links that offer a good compromise between length and reliability using a cost function. Every node assigns its neighbors a link cost that depends on how close to its preferred signal strength the neighbor's RSSI value is. Based on this link cost, our path metric then selects the most robust path.

We demonstrate the effectiveness of our approach in a simulation-based study with realistic mobility traces where nodes move according to the vectorized street map of a major city. With nodes moving at car speeds (up to $40mph$), paths selected by our path metric are a factor of 7 less likely to break than paths chosen by minimum hop count. In terms of end-to-end performance, the packet loss ratio caused by route interruptions during the transmission of data is reduced by a factor of up to 15.

We conclude that selecting paths based on the RSSI value of their constituting links allows to increase route robustness considerably. Furthermore, our results indicate that it is possible to build city-wide wireless networks. However, to what extent the performance we derived in our simulation experiments is achievable in real networks, remains to be determined by a real-world study.

REFERENCES

[1] Sulabh Agarwal, Ashish Ahuja, Jatinder Pal Singh, and Rajeev Shorey. Route-lifetime assessment based routing (RABR) protocol for mobile ad-hoc networks. In *ICC (3)*, pages 1697–1701, 2000.

[2] Daniel Aguayo, John Bricket, Sanjit Biswas, Glenn Judd, and Robert Morris. Link-level Measurements from an 802.11b Mesh Network. In *Proceedings of ACM SIGCOMM'04*, Portland, Oregon, USA, October 2004.

[3] Josh Broch, David A. Maltz, David B. Johnson, Yih-Chun Hu, and Jorjeta Jetcheva. A performance comparison of multi-hop wireless ad hoc network routing protocols. In *MobiCom '98: Proceedings of the 4th annual ACM/IEEE international conference on Mobile computing and networking*, pages 85–97, New York, NY, USA, 1998. ACM Press.

[4] T. Camp, J. Boleng, and V. Davies. A survey of mobility models for ad hoc network research. *Wireless Communications & Mobile Computing (WCMC): Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications*, 2(5):483–502, 2002.

[5] Zhao Cheng and Wendi B. Heinzelman. Exploring long lifetime routing (llr) in ad hoc networks. In *MSWiM '04: Proceedings of the 7th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems*, pages 203–210, New York, NY, USA, 2004. ACM Press.

[6] Jasmeet Chhabra and Brent Elliott. Real-world experiences with an interactive ad hoc sensor network. In *ICPPW '02: Proceedings of the 2002 International Conference on Parallel Processing Workshops*, page 143, Washington, DC, USA, 2002. IEEE Computer Society.

[7] Douglas S. J. De Couto, Daniel Aguayo, John Bicket, and Robert Morris. A high-throughput path metric for multi-hop wireless routing. In *MobiCom '03: Proceedings of the 9th annual international conference on Mobile computing and networking*, pages 134–146, New York, NY, USA, 2003. ACM Press.

[8] Richard Draves, Jitendra Padhye, and Brian Zill. Routing in multi-radio, multi-hop wireless mesh networks. In *MobiCom '04: Proceedings of the 10th annual international conference on Mobile computing and networking*, pages 114–128, New York, NY, USA, 2004. ACM Press.

[9] R. Dube, C. Rais, K. Wang, and S. Tripathi. Signal stability based adaptive routing (ssa) for ad hoc mobile networks, 1997.

[10] A. Feldmann, A. C. Gilbert, Polly Huang, and Walter Willinger. Dynamics of IP traffic: A study of the role of variability and the impact of control. In *SIGCOMM*, pages 301–313, 1999.

[11] T. Goff, N. Abu-Ghazaleh, D. Pathak, and R. Kahvecioglu. Preemptive routing in ad hoc networks. In *Proceedings of ACM MobiCom*, 2001.

[12] J.-Y. Le Boudec and M. Vojnovic. Perfect Simulation and Stationarity of a Class of Mobility Models. In *IEEE Infocom*, 2005.

[13] N. L. Johnson, S. Kotz, and A. W. N. Balakrishnan. *Continuous univariate distributions: Vol. 1*. Wiley, New York, 1994.

[14] S. Lee and M. Gerla. Split multipath routing with maximally disjoint paths in ad hoc networks. *Proceedings of the IEEE ICC*, pages 3201–3205, June 2001.

[15] Vincent Lenders, Joerg Wagner, and Martin May. Analyzing the Impact of Mobility in Ad Hoc Networks. In *ACM/Sigmobile Workshop REALMAN 2006*, Florence, Italy, May 2006.

[16] G. Malkin. RIP Version 2. RFC 2453 (Standard), November 1998.

[17] M. Marina and S. Das. On-demand multipath distance vector routing in ad hoc networks. *Proceedings of the International Conference for Network Procotols (ICNP)*, November 2001.

[18] Valery Naumov, Rainer Baumann, and Thomas Gross. An evaluation of inter-vehicle ad hoc networks based on realistic vehicular traces. In *MobiHoc '06: Proceedings of the seventh ACM international symposium on Mobile ad hoc networking and computing*, pages 108–119, New York, NY, USA, 2006. ACM Press.

[19] Liang Qin and Thomas Kunz. Pro-active route maintenance in dsr. *Mobile Computing and Communications Review*, 6(3):79–89, July 2002.

[20] A.K. Saha and D.B. Johnson. Modeling mobility for vehicular ad hoc networks. In *In poster at ACM VANET*, 2004.

[21] C. K. Toh. Associativity based routing for ad hoc mobile networks. *Wireless Personal Communications Journal, Special Issue on Mobile Networking and Computing Systems*, 4(2):103 – 139, March 1997.

[22] Z. Ye, Krishnamurthy, and S.V. Tripathi. A framework for reliable routing in mobile ad hoc networks. *IEEE INFOCOM*, 2003.

[23] J. Yoon, M. Liu, and B. Noble. Random waypoint considered harmful. In *INFOCOM*, 2003.

[24] Xiang Zeng, Rajive Bagrodia, and Mario Gerla. Glomosim: A library for parallel simulation of large-scale wireless networks. In *Workshop on Parallel and Distributed Simulation*, pages 154–161, 1998.