Adaptive Mobile Applications in Passive Network Awareness

Sumit Kumar Sar[1], Dinesh Kumar Bhawnani[2], Chaitali Choudhary[3]
[1,]Associate professor, CSE Dept(BIT Durg),[2,3] Assistant Professor, CSE Dept(BIT Durg)

Abstract

In pervasive computing, adaptation is essential to applications. However, adaptation requires an up-to-the-minute understanding of the state of the environment, which often comes with a significant added cost in terms of computation and communication. In this paper, we explore possibilities for measuring the degree of mobility in dynamic ubiquitous networks. Existing measures of mobility degree are either global values (e.g., the average node speed) or local measures that require additional coordination (e.g., the relative speed of neighboring nodes). In contrast, we describe a completely passive approach to measuring the local mobility degree that uses knowledge about communication mechanisms to determine the network dynamics affecting a single node. This context information can subsequently be used by a node to adapt communication and application protocols to current conditions. In this paper, we define a passive metric that gives a node an approximate view of local network dynamics and compare its ability to capture dynamics with existing metrics.

1 Introduction

The increasing ubiquity of wireless devices has introduced challenges in connecting these devices. Research in mobile ad hoc networks (MANETs) has created novel solutions to route messages between these nodes without the assistance of a wired infrastructure. However, as mobile computing applications become more pervasive, it becomes apparent that the need for adaptation in both low-level communication protocols and high-level applications is essential. Common forms of context already used to adapt application behavior include location (e.g., for guide programs [1, 2]), time (e.g., for reminders [3]), or weather conditions (e.g., for automated field notes [4]). While toolkits that allow sensing such context [5, 6, 7] provide information about a device's physical environment, network context, i.e., information sensed about the status of the communication links, can also be very important. For example, an application may use information about the network's available bandwidth or provided latency to change the data it sends, e.g., to lower the fidelity of data when bandwidth is restricted. At the network layer, a routing protocol might change how it transmits messages depending on the nearby network, e.g., a hybrid of reactive and proactive routing may scale back its degree of proactive ness in highly dynamic environments. Such adaptively is essential for ubiquitous computing applications.

Traditional means of measuring context from a local region are active in that they require nodes to exchange information to calculate their context (e.g., their respective locations or speeds) or send extra control messages (e.g., ping messages sent to establish latency measures). In this paper, we explore the practicality of passively measuring context, focusing on, the local degree of mobility. Necessarily, any approach to passive context sensing generates merely an approximation of the desired context, and we seek to determine the accuracy with which the degree of mobility can be estimated without incurring additional network overhead. The remainder of this paper is organized as follows. Section 2 describes current approaches to sensing network context. In Section 3, we outline our model for passively sensing the mobility degree. Section 4 measures how the quality of our passive metric differs from the actual truth. In Section 5 provides future work & conclusions.

2 Related Work

Many existing projects provide network-awareness by requiring a separate piece of software that generates additional network traffic. For example, piecewise network-awareness [8] creates a dedicated service to actively monitor and collect network context but separates the characteristics sensed about the wireless portion of a mobile network from those sensed about the wired portions. Network-awareness using mobile agents [9] integrates network sensing tasks into an agent but requires each agent to periodically beacon messages to measure network status. Active network monitoring has been explicitly separated from passive network monitoring. Komodo [10] differentiates the two, defining passive context sensing as any mechanism that does not add network overhead. However, Komodo requires knowledge of the entire network (even network links not currently in use), so the project implements an active sensing approach. Likewise, [11] defines active and passive network-awareness, but favors a distinct separation of the communication and network sensing, resulting in an active approach. Passive measurement of a mobility degree has been explored in a scheme that uses perceived signal strength to adapt a routing protocol [12]. This approach requires that nodes are able to easily discern the signal strength of incoming packets and requires each node to store information about the past signal strengths for all of its neighbors. The approach also requires nodes to send periodic hello messages to monitor their neighbor set, which adds network overhead. The approach in [13] monitors packet traffic to provide routing protocols information about packets dropped at the TCP layer. This information allows protocols to more quickly react to route failures. We undertake a similar approach in this work but focus on gathering a local measure of mobility instead of boosting performance on a particular network flow. With respect to metrics addressing mobility degree, early work [14] defined the mobility degree as the average relative speed between all pairs of nodes. As a sensed metric to provide to applications, this approach requires nodes to be equipped with some location monitor (e.g., GPS) and to exchange this information periodically. In addition this metric has a global perspective and hence is useful for statistics purposes rather than providing dynamic support for adaptation. More recently, it has been shown that network topology dynamics are impacted by more than just physical node mobility [15], but no concrete proposal has been provided for sensing these changes as context and incorporating them into applications or communication protocols. Our approach builds on the successes (and failings) of these previous approaches to create a passive metric for sensing local network dynamics. We use information gathered from eavesdropping on existing network traffic to enable each node to create a local view of its network dynamics.

3 Model

In supporting envisioned ubiquitous computing applications, it becomes necessary to use infrastructure-less communications to connect multitudes of local devices. In such scenarios, mobile ad hoc networks (MANETs) and the protocols devised for them play an important role in ensuring devices are successfully interconnected. In this section, we briefly introduce important aspects of unicast routing protocols for MANETs. This is important because our passive metric uses communication inherent in these protocols to sense the mobility degree on both a global and local scale. Because existing methods for sensing local mobility degree are limited, we formally derive a local mobility metric that represents a measure of truth against which we can compare our passive local metric. Finally we introduce this passive local metric and provide some intuition into how it measures a local degree of mobility.

3.1 Overview of basics of unicast routing protocols

Unicast routing protocols for MANETs require every node to serve as a router and are classified as either table-driven or on-demand. A popular example of a table-driven protocol is Destination-Sequenced Distance Vector Routing (DSDV) [16]. In DSDV, each node maintains a table giving the best known distance to each destination and the next hop to take to get there. These tables are updated by periodically exchanging information among neighbors, generating a fairly constant overhead that is independent of the amount of useful communication. On-demand routing determines routes only when a data packet needs to be sent. Two examples are Ad hoc On-Demand Distance Vector Routing (AODV) [17] and Dynamic Source Routing (DSR) [18]. On-demand routing protocols broadcast a route request that propagates to the destination. A reply is returned from the destination along the same path. AODV stores the routing information in tables on each node, and the tables are updated via periodic exchanges of stored information. In DSR, the packet carries the routing information, and no beaconing is required. Each approach has its advantages and disadvantages [19]; details are omitted for brevity. In all these protocols, when a node detects that a path has been broken due to a failed link, it sends a route error packet to its active predecessors for that destination, i.e., neighbors known to use the node to forward packets to the destination. One common way to characterize a network's dynamics is by the number of links that break over time. Every time a link is broken, a route error packet is generated, and because mobility causes link failures, the degree of mobility of the system influences the number of route error packets. Our passive metric is based on this assumption. Node mobility is not the only cause of link breaks, and our goal in Section 4 will be to determine how well the two correlate. However, it is worth noting that, since the degree of mobility is often used to infer network stability, the rate of link breakages itself is a useful measure of mobility, from an application perspective.

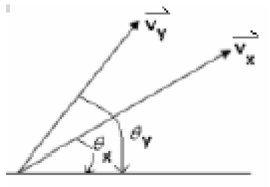

Fig. 1. Relative Velocity

$$v(x,y,t) = \sqrt{(v_{x,t}{}^2 + v_{y,t}{}^2 - 2 \times v_{x,t} \times v_{y,t} \times \cos(\theta_{x,t} - \theta_{y,t}))} \quad (1)$$

3.2 Active metrics for Mobility Degree

Measuring the mobility of a network or an area surrounding a node can give application information about the relative stability of its node's communication links. Metrics for measuring the degree of mobility can be broadly classified as global or local metrics depending on the region over which they measure mobility. An example of a global metric is the average node speed across all nodes. An example of a local metric is the average relative speed of a particular node with respect to its neighbors. In many instances, simulation results for communication protocols are often evaluated with respect to global metrics of mobility. For example, a common mobility model used in simulations is the random waypoint mobility model [20]. In this model, two parameters, the average node speed and the pause time (or the amount of time a node waits between switching trajectories) both measure a node or network's mobility. Though these measures do give a feel of the surrounding network, they can only be measured dynamically by incurring message overhead. Mobility metrics such as pause time are artifacts of simulations and have no bearing in real deployments. On the other hand, metrics such as the average node speed do not sufficiently measure dynamics. Attempts to account for this in simulation [21] measure the relative velocity with respect to all nodes. We define a local metric for mobility based on the relative velocity of a node and its directly connected neighbors. In the remainder of this paper, this metric is considered a target for our passive mobility metric. In this derivation, V (x, y, t) is the relative velocity of node x with respect to y at time t. Note that V (x, y, *t)   V(y, x, t). The relative velocity between x and y* can be calculated using the following equation [22] derived via simple vector arithmetic. Where, V_k , t is the velocity of node k at time t and T_k , t is the angle between the velocity of node k and the horizontal at time t. This relationship is depicted in Fig. 1. Let N (x, t) be the list of x's neighbors at time t. Then, |N (x, t)| denotes the number of x's neighbors. The average relative velocity of node x with respect to all of its neighbors at time t is given by:

$$\frac{1}{|N(x,t)|}\sum_{y\in N(x,t)} v(x,y,t) \quad (2)$$

$$M_x = \frac{1}{(t_2 - t_1)}\int_{t_1}^{t_2}[(\frac{1}{|N(x,t)|}\sum_{y\in N(x,t)} v(x,y,t))]dt \quad (3)$$

where Mx is a local metric defining the degree of mobility of x. To dynamically use this metric for adaptation, nodes must periodically exchange location and velocity information, resulting in increased traffic. The remainder of this paper explores a passive metric that tries to correlate well with the above local metric without incurring added communication costs.

3.3 A Passive Metric

A passive metric that can correlate well with target metric (above) can be a major advantage to the system in real time. Our passive metric uses information gleaned by eavesdropping on existing communications to infer the degree of mobility in a region of the network. As eluded to previously we use the route error packets inherent to routing protocols for this purpose. In this definition, the symbol RERR refers to the number of route error packets. We represent the varying metrics as RERR X, Y, Z, where the values of the subscripts identify different measurement strategies. The first subscript (capitalized) denotes whether the metric has local (L) or global (G) scope, while the second indicates whether it corresponds to the number of route error packets generated (by this node) (g) or seen (by this node) (s). The last category includes the route error packets this node generates in addition to the route error packets that the node forwards on behalf of other nodes. The last subscript signifies whether the metric is normalized (e.g., divided by the total number of data packets) (n) or absolute (no normalization done) (a).

- $RERR_{g, g, n}$ — the number of route error packets generated by the entire system normalized to the total number of data packets successfully received by their intended destination..
- $RERR_{g, g, a}$ — the number of route error packets generated by the entire system.
- $RERR_{l, g, n}$ — the number of route error packets generated by each node normalized to the total number of data packets successfully processed (i.e., received or forwarded) by that node.
- $RERR_{l, s, n}$ — the number of route error packets forwarded by each node normalized to the total number of data packets processed by that node. In our protocol for sensing mobility degree, we calculate these values by simply examining traffic passing through the local node. This approach eliminates overhead by taking advantage of the fact that each node is already serving as a network router.

4 Evaluating a Passive Mobility metric

To evaluate how well our passive metric correlates with standard mobility measures, we measured our metric's performance through simulation. These results were then compared to several of the standard metrics described in Section 3.2. In this section,

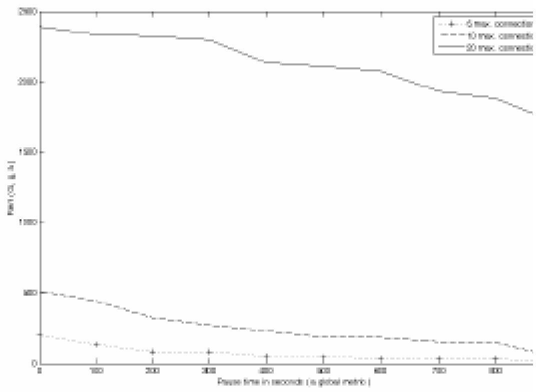

Fig. 2. RERR(G, g, a) Vs. Pause time

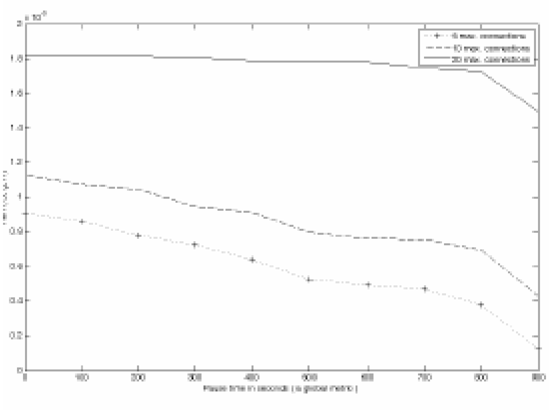

Fig. 3. RERR(G,g,n) Vs. Pause time

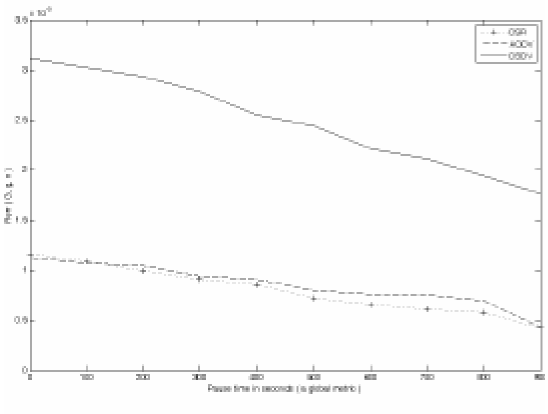

Fig. 4. RERR(G,g,n) Vs. Pause time

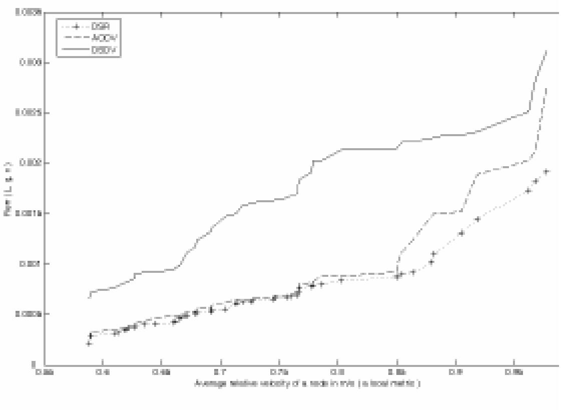

Fig. 6. RERR(L,g,n) Vs. Avg. relative veloc

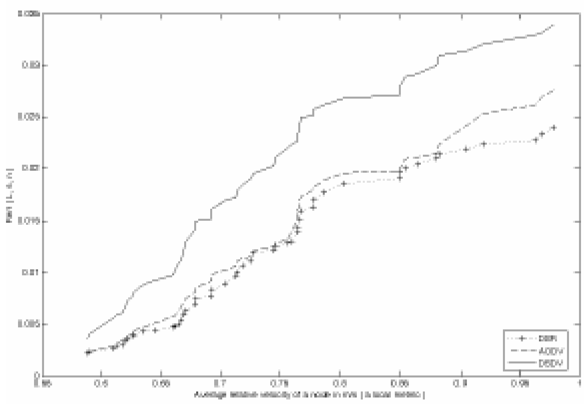

Fig. 7. RERR(L,s,n) Vs. Avg. rel. velocity

we describe the results of these comparisons.

4.1 Simulation Setup

In our measurements, we utilized node mobility patterns based on the random waypoint mobility model previously mentioned. In all of these network setups, we distributed 50 nodes randomly in a rectangular area of 1500m x 300m, and, unless otherwise specified, the nodes moved at speeds uniformly distributed between 0 and 20 m/s. Each node's radio transmission range was assumed to be 150m and had a channel capacity of 2Mbps. The simulations use the two-ray ground propagation model. Each result point in the following charts indicates 10 sample networks, for which each run lasted 900 simulation seconds. To vary the offered load in the sample networks, we simulated networks in which there were 5, 10, or 20 active connections at any time, each supporting constant bit rate (CBR) traffic at a rate of 4 packets/second.

4.2 Results

The metric described in Section 3.3 measures the number of route error packets a particular node either itself generates or that the node passes on as part of its duties as a router. In MANET routing protocols, the number of route error packets is extremely dependent on the

traffic flowing through the node in question. For this reason, to unify the comparisons among different network samples and routing protocols, our metric normalizes the number of route error packets counted by a particular node with respect to the number of data packets counted by that same node. Our first set of experiments evaluates the effect of this normalization on our results. The results we show here use the AODV routing protocol under three different traffic conditions described above (i.e., 5, 10, and 20 active connections). The results are quite similar for other routing protocols. Fig. 2 plots the number of route errors generated in the entire network over the entire simulation time (a passive global measure of mobility) as a function of the mobility model's pause time (a simulation. level global measure of mobility). Fig. 2 only has value when taken in conjunction with Fig. 3, which plots the same metric, normalized to the total number of data packets successfully sent through the network. Together, these two charts demonstrate that the normalization makes the comparison across traffic conditions fairer in that the number of route error packets counted is now a factor of the number of data packets that traveled through the network. The intuition is that the non-normalized representation of this data (Fig. 2) unfairly penalizes the 20-flow case (i.e., counts more route error packets) but does not give it the benefit it deserves for successfully delivering more data packets. Fig. 3, on the other hand, accounts for this difference, and the difference in the traffic flows in Fig. 3 is due entirely to the added congestion resulting from the increased number of flows. In the remainder of the measurements we report, we always normalize the route error packets counted to some measure of data packets successfully delivered. The next set of experiments aim to measure how well the global version of our passive metric correlates with standard global metrics that cannot be passively sensed, e.g., pause time within the network simulation and the average node speed (across all nodes). The results we report for this measure show our metric's performance using information from three different MANET routing protocols—DSDV, AODV, and DSR in an attempt to draw generalizable conclusions across many protocols. In this case, we fixed the number of connections at any given time to be a maximum of ten. Fig. 4 compares our passively sensed global metric to changing pause times. In random waypoint mobility model, as the pause time decreases, the mobility of the network is also decrease (i.e., a network with a pause time parameter set at 900 seconds is said to display less mobility than a network with a pause time parameter of 0 seconds). The figure demonstrates that as the mobility of the network decreases (as measured in the standard metric, pause time), the degree of mobility measured by our passive metric also decreases, i.e., our passively sensed global metric is well-correlated with a standard view of global mobility degree. Fig. 5 correlates the same passively sensed global metric as above with a different standard measure of mobility: the overall average node speed. In this case, our simulation used the random waypoint mobility model with a fixed pause time of 450 seconds but varied the average node speed from 1 to 50 m/s (where a network with an average node speed of 50 m/s is assumed to be significantly more mobile than a network with an average node speed of 1 m/s. As the average speed of the nodes increases, the number of route errors generated also increases steadily, indicating a good degree of **correlation** between these two metrics. It is comforting that the passively sensed global metrics are well-correlated with standard metrics, but the goal of our work is to create a passively sensed local metric that applications can directly use to adapt their behavior. Therefore, we shift our focus to the more pragmatic metric that serves the system in real-time. The last two sets of experiments demonstrate the correlation between our passively sensed local metric and the targeted local metric that we defined in Section 3.2. These scenarios again use random waypoint mobility to dictate the movement of the nodes and fix the simulation's pause time at 450 seconds and the maximum speed of the nodes at 50m/s. The target local metric, i.e., the average relative velocity of a node with respect to its neighbors averaged across the simulation time, is compared to the number of route errors generated by that node normalized to the data traffic processed by that node. Fig. 6 demonstrates that, for a given node, as the local metric M_x increases, the number of route errors generated also increases. Therefore our locally sensed passive metric correlates well with our target comparison metric that directly measures local mobility degree. The previous measurement is restricted to giving node information about the mobility degree only within its one-hop neighborhood because it only counts route error packets resulting from broken links that were connected to the node in question. The same simulation settings as in the previous scenario are retained for the results shown in Fig. 7, but instead of correlating the target metric against the number of route errors a particular node generated, we use the number of route errors the node saw (i.e., either generated or forwarded for some other node). This gives the node in question a slightly wider network neighborhood to consider, because it incorporates information about links more than one hop distant. Because this metric accounts for a wider range of network traffic flowing through the node, we feel it is a better measure of mobility degree. In addition because all of the processing is local, the computation overhead required computing this metric in comparison with the previous one is negligibly small. Our simulations have corroborated our belief that we can create a passively sensed measure of mobility that can provide higher level protocols and applications a metric on which to base their adaptation decision. It is this last metric—the number of route errors any node processes, normalized to the number of data packets processed—that we provide to applications as a measure of the local region's mobility.

5 Future Work & Conclusion

The previous sections have outlined an approach for passively sensing the dynamics of nodes in ubiquitous computing application deployments that rely on MANETs for communication. As described previously, this metric can be of great use to adaptive applications and communication protocols. Future work on our approach will come on three fronts. First, we will explore additional network aspects that can be passively sensed in a manner similar to our approach for sensing the mobility degree. As one example, information about aspects of wireless channel conditions can be useful to adaptive applications. Second, we are creating a context-provision middleware that allows us to interpret these varying pieces of context and provide the information to applications through an intuitive programming interface. Thirdly, we have created a model for adaptive communication for pervasive computing. This model explores a combination of reactive (i.e., purely on-demand) routing and proactive (i.e., advertisement based) routing. We are incorporating the use of our passively sensed mobility degree into the communication protocol to enable the protocol to increase the ratio of reactive behavior to proactive behavior when the degree of mobility in the network increases. Our initial results indicate that such adaptation will provide the optimal balance between efficiency and overhead in our communication protocol. In summary, we presented a novel approach to network-awareness in ubiquitous communications, namely passively sensing a local region's degree of mobility. We have shown that our new metric correlates well with commonly accepted measures for mobility degree. In addition, our approach offers significant benefits over existing approaches in that it requires no increased communication overhead to sense.

References

1. Abowd, G., Atkeson, C., Hong, J., Long, S., Cooper, R., Pinkerton, M. Cyberguide: A mobile context-aware tour guide. ACM Wireless Networks, 3:421–433, 1997.

2. Cheverst, K., Davies, N., Mitchel, K., Friday, A., Efstratiou, C. Experiences in developing and deploying a context aware tourist guide: the guide project. In Proc. of MobiCom, pages 20–31, 2000.

3. Dey, A., Abowd, G. Cybreminder: A context-aware system for supporting reminders. In Proc. of the 2nd Int'l. Symp. on Handheld and Ubiquitous Comp.,

pages 172–186, 2000.
4. Pascoe, J. Adding generic contextual capabilities to wearable computers. In Proc. of the 2nd Int'l. Symposium on Wearable Computers, pages 92–99, 1998.
5. Dey, A., Salber, D., Abowd, G. A conceptual frame work and a toolkit for supporting the rapid prototyping of context-aware applications. Human Comp. Interaction, 16(2–4):97–166, 2001.
6. Hackmann, G., Julien, C., Payton, J., Roman, G.-C. Supporting generalized context interactions. In T. Gschwind and C. Mascolo, editors, Software Engineering and Middleware: 4th Int'l. Wkshp, Revised Selected Papers, volume 3437 of LNCS, pages 91–106., March 2004.
7. Hong, J., Landay, J. An infrastructure approach to context aware computing. Human Computer Interaction, 16(2–4), 2001.
8. Cheng, L., Marsic, I. Piecewise network awareness service for wireless/mobile pervasive computing. Mobile Networks and Applications, 7:269–278, 2002.
9. Peysakhov, M., Artz, D., Sultanik, B., Regli, W. Network awareness for mobile agents on ad hoc networks. In Proc. of AAMAS, pages 368–376, 2004.
10. Ranganathan, M., Acharya, A., Sharma, S., Saltz, J. Network-aware mobile programs. In Milojicic, D., Douglis, F., Wheeler, R., editors, Mobility: Processes, Computers, and Agents, pages 567–581, 1999.
11. Caripe, W., Cybenkoo, G., Moizumi, K., Gray, R. Network awareness and mobile agent systems. IEEE Communications, pages 44–49, July 1999.
12. Basu, P., Khan, N., Little, T. A mobility based metric for clustering in mobile ad hoc networks. In Proc. of the Int'l. Conf. on Distributed Comp. Wkshps., pages 413–418, 2001.
13. Yu, X. Improving tcp performance over mobile ad hoc networks by exploiting cross-layer information awareness. In Proc. of MobiCom, pages 231–244, 2004.
14. Johansson, P., Larsson, T., Hedman, N., Mielczarek, B., Degermark, M. Scenario-based performance analysis of routing protocols for mobile ad hoc networks. In Proc. of MobiCom, pages 195–206, 1999.
15. Schmitz, R., Torrent-Moreno, M., Hartenstein, H., Effelsberg, W. The impact of wireless radio fluctuations on Ad hoc Network performance. In Proc. of the 29t h Int'l. Conference on Local Computer Networks, pages 594–601, 2004.
16. Perkins, C.E., Bhagwat, P. Highly dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers, Comp. Comm. Rev.,pages 234-244, 1994.
17. Perkins, C.E., Royer, E.M. Ad-hoc on-demand distance vector routing, MILCOM, 1997.
18. Johnson, D.B., Maltz, D.A., Broch, J. DSR: The Dynamic Source Routing Protocol for Multi-Hop Wireless Ad hoc Networks, Mobile Computing, 2001.
19. Royer, E.M., Toh, C.K., A review of current routing protocols for ad hoc mobile wireless networks. IEEE Personal Communications magazine, pages 46--55, April 1999.
20. Bettstetter, C. Smooth is Better than Sharp: A Random Mobility Model for Simulation of Wireless Networks, MSWiM, 2001.

21. Johansson, P., Larsson, T., Hedman, N., Mielczarek, B., Degermark, M. Scenario-based Performance Analysis of Routing Protocols for mobile Ad hoc Networks, Proc. of the 5th annual ACM/IEEE Int'l. Conf. on Mobile Comp. and Networking, pp. 195–206, 1999.
22. http://www.physicsclassroom.com/Class/vectors/U3L1e.html