Straight Line Routing for Wireless Sensor Networks

Cheng-Fu Chou, Jia-Jang Su, and Chao-Yu Chen
Computer Science and Information Engineering Dept., National Taiwan University
ccf,jiajing,chaoyu@cmlab.csie.ntu.edu.tw

Abstract

Sensor networks are large-scale distributed sensing networks comprised of many small sensing devices equipped with memory, processors, and short-range wireless communication radio. Instead of broadcast-based routing protocols, in this paper we propose a novel energy-efficient routing protocol, which is called Straight Line Routing Algorithm[1] (SLR), for wireless sensor networks. To achieve the routing task without broadcasting, the source host constructs the event path and the sink host constructs the query path respectively. That is, the routing path is found as the query path and the event path first intersect. Moreover, the SLR is able to build both the query path and the event path without any help of the geographic information.

We evaluate the performance of Straight Line Routing and Rumor Routing protocols through extensive simulations. The simulation results indicate that compared with Rumor Routing, the SLR can save more energy consumption, provide better path quality, and improve the successful ratio of routing as well.

1. Introduction

In the near future, advances in processor, memory, and radio technology will enable small and cheap nodes capable of wireless communication and significant computation. Networks comprised of such nodes can coordinate to perform distributed sensing of environment phenomena. Therefore, there are a lot of applications for sensor networks. For instance, sensor nets often are installed in monitoring system, such as Structure Health Monitoring (SHM), which could be used to monitor human healthy anytime and anywhere. Another example is to monitor the degree of temperature or the density of dust around a volcano to predict the eruption time.

[1] **This work was partially supported by the National Science Council and the Ministry of Education of ROC under the contract No. NSC92-2622-E-002-002 and 89E-FA06-2-4-8**

Since sensor nodes are scattered over the entire sensing area, communication between sensor nodes is usually in a hop-by-hop manner. This is similar as the traditional distributed routing protocols such as Destination Sequenced Distance Vector [2] (DSDV), Ad Hoc On-Demand Distance Vector [1] (AODV), and Dynamic Source Routing [3] (DSR) do in ad hoc networks. However, these broadcast-based routing protocols are not appropriate for a sensor network since the broadcast is a costly operation. Frequent broadcasts drain the sensor battery off quickly. In addition, it is difficult to recharge or replace the sensor nodes that run out of energy and are distributed widely over the geographic area.

Another type of routing protocols is the random-walk-based protocol. Without triggering every sensor node to generate a routing message, random-walk-based protocol limits the propagation of routing message among partial sensor nodes. Gossip [4] and Rumor [5] are two famous random-walk-based routing protocols. Gossip has focused on multicast or broadcast services in the Internet and does not take wireless environment features, such as power constrain and high error rate of the wireless channel, into account. Thus, in this paper we are interested in another routing protocol - *Rumor Routing*. In *Rumor Routing* protocol, every node has to maintain its neighbor list. When propagating a routing message, the node appends its neighbor list in that routing message. Consequently, the message will record every node that has received this message. By examining such visited list, the node can choose a neighbor node that has not received this message, and keep the routing away from growing in the "backward" direction. The advantage of *Rumor Routing* is simple to implement but with the following drawbacks.

Spiral Problem

Rumor routing is able to avoid searching the path in the backward direction, but unable to figure out a better direction for the routing path. In other words, such routing scheme might generate a lot of meanderings along the routing path. The worst case could be formed as a spiral. Furthermore, the winding path consists of more nodes than the straight path does, i.e., compared with the straight path, the response time of the winding path is longer and the total consumed energy of the winding path is larger as well.

Waste Energy in recording visited nodes in packet's payload

In *Rumor Routing*, paths are constructed in a hop-by-hop manner. In order to avoid choosing backward nodes, the current node first examines the visited list to select an unvisited node as the next hop, appends all its neighbors' IDs in packet payload and then transmits the routing packet to that chosen node. Hence, the size of the routing packet is expected to become larger and larger and the node has to consume more and more energy to transmit this routing packet.

To avoid the spiral problems, the intuition is to reduce the number of meanderings in the routing path. Therefore, in this work we propose a novel random-walk-based routing protocol --- Straight Line Routing (SLR), which aims to make the routing path grow as straight as possible.

2. Network Assumption

In this section we first state some basic assumptions about the sensornets in this work. Sensornets comprise the nodes that are spread out, randomly or in some pattern over some well-defined area. Nodes only have short-range communication, but also are inside radio range of several other nodes. The communication power of all nodes in sensornets is equal. Energy is a scarce resource for sensor nodes. Using a node's wireless communication requires energy.

We note that target signal amplitude attenuates, as a monotonically decreasing function of the distance from the source, according to an inverse distance squared law or exponentially. When a node receives a packet successfully, it is able to tell which node sending this packet and measure the signal strength. That is, we assume that all nodes are aware of energy model. After receiving a packet, each node has the ability of determining the distance from the source according to the signal strength.

Now, we give an introduction about how to figure out a routing path in the sensor network. When a sensor node detects an interesting event, it starts to invoke the routing mechanism for the event path. This routing path for the event is growing straight until it hits the border or the path length is equal to some constant. That is, a fixed-length path is constructed, and all nodes along this path keep the event information. On the other hand, there is a special node that will request events on the sensor network. We refer to this node as the "Sink". Similarly, when the sink queries an event, it also executes the routing mechanism. A query path will be constructed by the same routing scheme. When the query path intersects the corresponding event path, the routing path is completed. The intersection node is called the anchor node, and the anchor needs to reply an *ACK* message to the Sink when it has been created.

3. SLR Overview

The main idea of the SLR protocol is to keep the routing path straight. Like *Rumor Routing*, SLR constructs the path in the hop-by-hop type. In each hop, we need to choose a node, which lies on the extended line of the path, as the next hop. Fig. 1 illustrates the original idea. Suppose node *b* in Fig. 1 is the newest hop, node *a* is the pre-hop of node *b*, (Obviously, the direction of the path is from *a* to *b*) and the distance between them is $R/2$ where R is the radio distance. We focus on the intersection of *a*'s and *b*'s radio ranges and consider our new coordinate system. We define the first dimension as the distance from node *a* and the second dimension as the distance from node *b*. It is clear the node, whose location is (R, $R/2$), is the most suitable node to be the next hop in the intersection message.

Unfortunately, the node in the position (R, $R/2$) does not always exist. We need to modify our algorithm for adapting to this situation. Before introducing the algorithm, we define some interest regions associated with a node. *Outside Band* and *Inside Band* of node *n* are referred to the circular band regions where the center is node *n* and the radius are R and $R/2$, respectively.

The basic concept of Straight Routing Algorithm is illustrated as Fig. 2. We assume node *A* is the current hop, and node *B* is the previous hop. Shown as Figure 2, we can observe that the intersected region of *Inside Band* of node *A* and *Outside Band* of node *B* provides

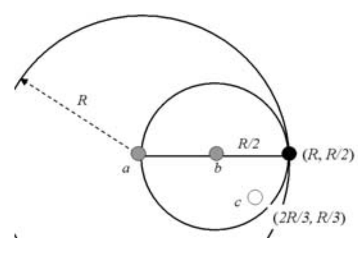


Fig. 1. the ideal case of SLR. the first dimension is the distance form node *a*, and the second is the distance from node *b*. The node in the position (*R*, *R*/2) is most suitable to be the next-hop.

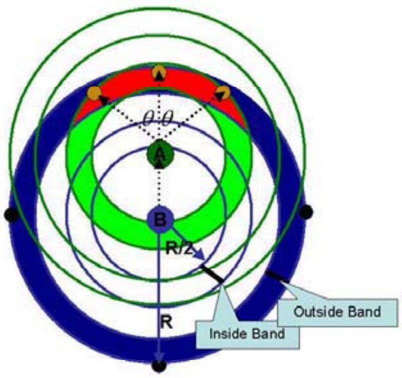


Fig. 2. the concept graph of SLR. Node A is the current hop, and node B is the pre-hop. The radiuses of *Onside Band* and *Inside Band* are R and R/2, respectively.

a proper set of nodes for choosing the next hop. We name this intersected region *Candidate Region*, and the next hop will be chosen from nodes inside it.

We designed two steps in our routing protocol. First step, the *Candidate Region* will be determined. Then, the second step is to choose a node from *Candidate Region* as the next hop.

Step 1:

For each routing, every node will maintain two variables, $Flag_{In}$ and $Flag_{Out}$. Based on our assumptions, after a node receives a route request, this node can calculate the distance from the sender to itself. By the way, the node can recognize itself in which band of the source. If it is in the *Inside band*, it will enable its $Flag_{In}$. On the other hand, if it is in the *Outside band,* it will enable its $Flag_{out}$.. A node will start contending to be the next hop only when its two variables, $Flag_{In}$ and $Flag_{Out}$, both have been enabled, because it is in the *Candidate Region*.

Step 2:

Subsequently, every node in *Candidate Region* will set its own timer, T_{wait}. In this work, we set the formula in the form of $T_{wait} = 1/dist_{parent} + 1/dist_{gradnparent}$. This means it has the best possibility to become the next hop node. When its timer expires, the node will issue a message to notify its neighbors. Besides the node that has become the next hop node, other nodes in *Candidate Region* will overhear the notifying message. After receiving the message, nodes will stop the contention procedure.

4. Improvements

Based on the discussion above, we consider the case of low network density. The situations that there is no node in the *Candidate Region* may often be suffered, and these situations will cause SLR can't pick up any node as the next hop. Therefore SLR is terminated usually before finishing constructing the path when the node-density is low.

Intuitively, the probability that the situation (empty *Candidate Region*) happens is inverse-proportional with the size of the *Candidate Region*. Unfortunately, in order to choose a better node and to make the path straight, SLR refines the area of the *Candidate Region* by receiving twice route messages. This feature makes the success probability of SLR degrade considerably on low-density networks. In addition to the terminated-probability, another drawback of SLR is the long delay of its routing procedure. Because of the method of refining *Candidate Region,* the longest distance of every hop (from the next node to the present) is the half of transmission radius. That means the routing procedure of SLR needs more time, and the hop count of the path is larger than general protocols such as *Rumor*.

For the purpose to adapt SLR to low-density sensor nets, the probability of successful path discovery and the path quality (i.e., hop count) can be enhanced. We devise four improvement schemes as the followed:

1. Adjusting the widths of *Inside Band* and *Outside Band*
2. SLR Dual Way
3. SLR Far Jump
4. SLR Short-Cut ACK

Adjusting the widths of *Inside Band* and *Outside Band*

Obviously, the size of the *Candidate Region* is

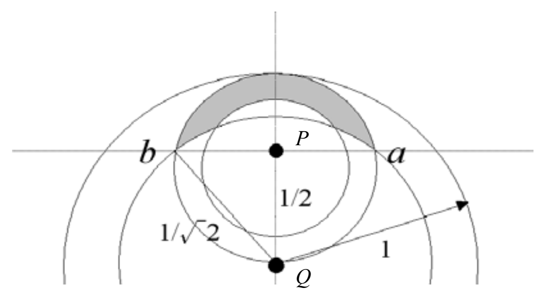


Fig. 3. the graph to obtain the range of the *Outside Band* width. Node *P* is the current hop, and node *Q* is the pre-hop. The distance of *P, Q* is 1/2.

directly related to the width of *Inside Band* and *Outside Band*. Moreover, the width affects the bending-angle of the path at every hop. For example, we observe that to decrease gb_{Out} (*Outside Band* width) or gb_{In} (*Inside Band* width) will decrease the size of the *Candidate Region* and bending-angle of the path both. Decreasing the bending-angle makes the path straighter but the terminated-probability higher. In the contrast, increasing the size of the *Candidate Region* makes bending-angle smaller, but the probability of successful path discovery becomes higher.

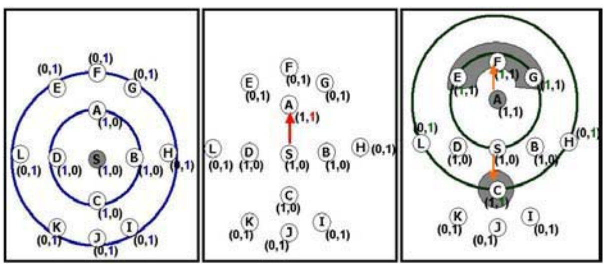
Therefore, now our task is to determine the values of gb_{Out} (*Outside Band* width) and gb_{In} (*Inside Band* width). At first we observe the Fig. 3 and suppose the transmission radius R is 1. The range of gb_{Out} value will be deduced roughly. We obtain

$$\frac{1}{\sqrt{2}} < \left(1 - gb_{out}\right) < 1$$

$$\Rightarrow 0 < gb_{out} < 1 - \frac{1}{\sqrt{2}}$$

In this work, we set the *Inside Band* a plate (i.e., gb_{in}=1), and the size *Candidate Region* just depends on the value of gb_{Out}.

SLR Dual Way



(*a*) (*b*) (*c*)

Fig. 4. the illustration of SLR Dual Way.

In basic SLR, the initial direction of the path is chosen by source randomly. However, if the initial direction is totally backward the destination, the path is getting worse and worse during path construction, because the SLR always keeps the path straight. Nevertheless, it deserves to be mentioned that the inverse direction is a good choice in this case. Based on this observation, we develop another scheme to improve SLR, which is the so-called SLR_DW.

In the beginning (shown in (*a*)), the source node *S* broadcasts a routing message within its communication range. After receiving this message the nodes in the *Inside Band* of *S* (ex: *A, B, C, D*) set their $Flag_{In}$ 1, while the nodes in the *Outside Band* (ex: *E, F, G, L, H, I, J, K*) set their $Flag_{out}$ 1.

Then (illustrated with (*b*)) a node (e.q., node *A*) has been chosen as the next hop, so the direction from *S* to *A* is the initial direction.

After node *A* rebroadcasts this message, the nodes in the *Inside Band* of *A* (e.q. *E, F, G*) set their $Flag_{In}$ 1 (demonstrated by (*c*)). At the same time, the nodes in the *Outside Band* of *A* (node *C*) set their $Flag_{out}$ 1. That means there is the path whose initial direction is from *A* to *C* has been generated. Hereafter, the both paths of different directions will be constructed individually.

SLR Far Jump

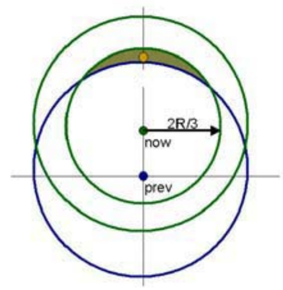


Fig. 5. the graph to show the SLR Far Jump. The new radius of *Inside Band* is 2R/3, and the gray area is suitable to be the new *Candidate Region*.

The prerequisite to enlarge the area of the *Candidate Region* is that every node is able to detect whether the *Candidate Region* is empty or not. For this reason, when a node transfers a routing message, it will set a timer and wait. No notifying message heard before the timer expires indicates that no node in the *Candidate Region*.

In this subsection we introduce the mechanism to enlarge the *Candidate Region*, SLR Far Jump. When the timer expires, the node will issue the *SLR_FJ* message, which the new length of *Inside Band* radius is recorded in. Therefore it is expectable that more neighbors will set their $Flag_{In}$ 1 after hearing this *SLR_FJ* message. However, dislike the basic SLR, the nodes whose pair ($Flag_{In}$, $Flag_{Out}$) is equal to (1,0), instead of (1,1), will contend to be the next hop. The reason can be explained through Fig. 5. For the current node, the gray area of Fig. 5 is suitable to be the new *Candidate Region*. The pairs ($Flag_{In}$, $Flag_{Out}$) of the nodes in this area are equal to (1,0).

Nevertheless, it is emphatic that the principle described above has to be used once again after the SLR Far Jump. This is because the distance of the hop decided by SLR Far Jump is longer than *R*/2. Consider the next hop choice. If we apply the principle of the basic SLR (That is, ($Flag_{In}$, $Flag_{Out}$) = (1,1)), the *Candidate Region* decided may not be suitable. Conversely, adopting the rule above (($Flag_{In}$, $Flag_{Out}$) = (1,0)) once again will determine the *Candidate Region* just like Fig. 5, which is more reasonable.

SLR Short-Cut ACK

In basic SLR, the distance per hop is a half of radio distance at most. That means the routing procedure of SLR needs more time, and the hop count of the path is larger. In this subsection, we take advantage of *ACK* messages to reduce the hop count of the path that has been constructed. We call this mechanism SLR Short-Cut ACK.

As the network assumptions, when two corresponding paths cross, the anchor (i.e., the intersection node) will reply the *ACK* message to the Sink node. It is interesting the propagation of the *ACK* message will stride across two hops at least. As the case illustrated by Fig. 6, when a node transfers an *ACK* message, at least two nodes along the path will receive it. In fact, according to the feature of triangles (shown in Fig. 6 (*c*)) there are three nodes receiving the message. Because the node will retransfer the *ACK* after receiving it, the response time and path length (hop count) are shorter than the original.

5. Performance Evaluation

In this section, we evaluate the performance of SLR by simulations. In our simulations, nodes are scattered randomly on a two-dimensional field. A simple radial propagation model was used, where each node could reliably send packets to any node within its communication range R_{TX}. Four combinations of protocols, which are SLR/SLR, SLR/RR, RR/SLR, RR/RR respectively are compared. The notation formed "*a/b*" indicates the protocols to construct the event path (initialized from sensor nodes) and the query path (initialized from the sink node), respectively. ("RR" means the *Rumor Routing*). The criteria are energy cost, successful ratio, and hop count

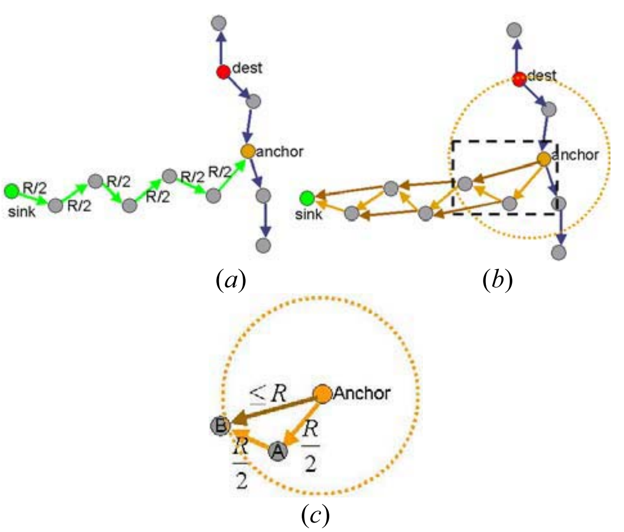


Fig. 6. a case of SLR Short-Cut ACK. The *ACK* message can stride across two hops at least (*a*) due to the feature of the triangle (*b*).

N_{Total}	the total number of sensor network
B_X	the network boundary length of X-axis
B_Y	the network boundary length of Y-axis
R_{TX}	the transmission range of the sensor node.
$nPath_{src}$	Number of Event paths
$nHop_{src}$	Number of hops for each event path
$gb^{src}_{In}=1$	The width of *Inside Band* applied in event path.
$nPath_{dst}$	Number of query paths
$nHop_{dst}$	Number of hops in each query path
$gb^{dst}_{In}=1$	The width of the *Inside Band* applied in query path.
gb^{dst}_{Out}	The width of the *Outside Band* applied in query path.

Table 1: the definition of notation in simulation

of the path averagely per routing. The definitions of the notations we used are listed in Table 1.

We evaluated SLR in two scenarios, small and large topologies respectively. The gb_{In} value is always set 1. We set gb^{src}_{Out} = 0.7 for high-density environments, and gb^{src}_{Out}= 0.5 for low-density networks. We tuned the transmission range R_{TX} to control the density of the network, and the range of R_{TX} is from 60 to 140. Besides, we did not evaluate the flooding algorithm in term of successful ratio and energy cost, because its successful ratio is assumed 100% and its energy cost is the maximum.

For each parameter set, we randomly created 100 topologies. There were 100 event-query pairs for every topology.

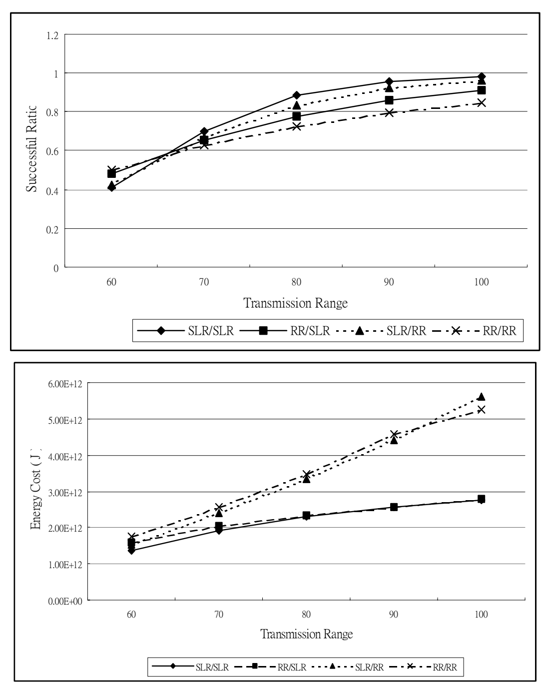


Fig. 7. Comparison of successful ratio and energy cost in the small topology.

small topology

In this scenario, we set N_{Total}=500, B_X=500, B_Y=500, $nPath_{src}$=1, $nPath_{dst}$=1, $nHop_{dst}$=10, gb^{src}_{Out}=0.3 and gb^{dst}_{Out}=0.6.

Fig. 7 shows the success ratio and energy cost of four protocol pairs respectively. We can find that SLR can find more paths. Moreover, SLR/SLR and SLR/RR are much superior to RR/RR when transmission range is 70. This means the improvement of SLR adapts to the sparse networks respectably. When the radio radius is 100, because the successful ratio of SLR/RR is much higher, SLR/RR consumes more batteries than RR/RR.

In addition, we compared the average hop count of the path. It is noticed that we only compared the paths which were searched successfully. The results are listed on the table below, and these show the benefit of the SLR Short Cut ACK. We specially observed the performance of one flooding-based protocol, AODV. The path searched by flooding-based routing scheme is viewed as the shortest path. As we expected, the path length of SLR/SLR is the smallest. Conversely, RR/RR makes the longest path for the same reason. By the way, the path of SLR/RR is shorter than RR/SLR's because the event path is fix-length. When the spiral problem happens, the query message needs to pass more nodes in order to reach the event path.

Scheme	Average hops
SLR/SLR	5.497667
SLR/RR	5.682305
RR/SLR	7.310626
RR/RR	7.616045
Flooding Based	3.3534

large topology

We set parameters as N_{Total}=2000, B_X=1000, B_Y=1000, $nPath_{src}$=1, $nPath_{dst}$=1, $nPath_{dst}$=10, gb^{src}_{Out}=0.3 and gb^{dst}_{Out}=0.6, to create a large-scale network. By comparing the correspond results in Fig. 7 and 8, we can find that the performance of RR degrades very much in large-scale networks, but SLR still maintains the high efficiency. This is because the spiral problem becomes serious in the large-scale network. In small-scale networks, the distance of the sink and the event is not too long. Even there are many meanderings in the query path, but it is very probable the query path crosses the event path. However, these situations happen seldom when the scale of the network is large. Therefore, the successful ratio and energy cost of RR deteriorate in the huge-scope sensor network, because the routing message of RR will pass numbers of nodes, and the unfixed searching direction makes the two paths intersect more difficultly

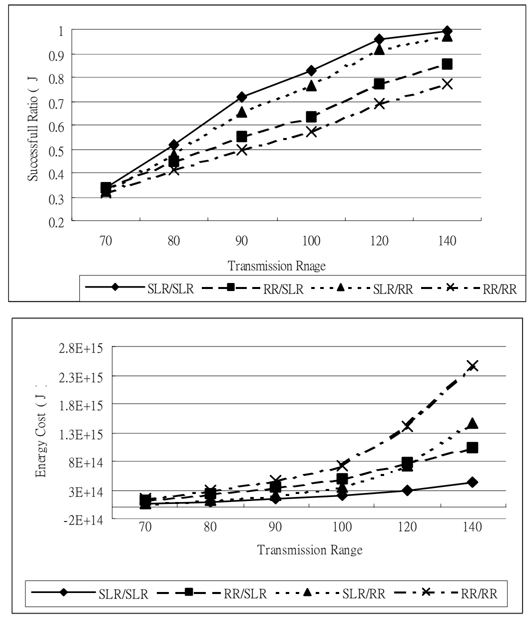


Fig. 8. Comparison of successful ratio and energy cost in the large topology.

6. Conclusions

Random-walk-based routing is a new routing protocol category without broadcasting procedures. To reduce the number of meanderings in the path, a novel random-walk routing protocol, SLR, has been proposed in this paper. This protocol aims to choose every hop in the original direction of the path. The simulation shows that the straight path can enhance the successful ratio of routing, and lower the energy cost.

7. References

[1] C.E. Perkins and E.M. Royer, "*Ad-hoc on-demand distance vector routing"* WMCSA '99. Second IEEE Workshop on, 25-26 Feb. 1999, pp. 90 – 100

[2] C.E. Perkins, P. Bhagwat, "*Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers*", in Proc. of the ACM SIGCOMM'94, vol. 24, no. 4, pp.234-244, Oct., 1994.

[3] D.B. Johnson, D.A. Maltz, "*Dynamic Source Routing in Ad Hoc Wireless Networking*", in Mobile Computing, pp.153-181,1996.

[4] M. Lin, K. Marzullo, S. Masini, "*Gossip versus deterministic flooding: Low message overhead and high reliability for broadcasting on small networks,*" UCSD Technical Report TR CS99-0637. http://citeseer.nj.nec.com/278404.html

[5] D. Braginsky and D. Estrin, "*Rumour routing algorithm for sensor networks,"* in First ACM International Workshop on Wireless Sensor Networks and Applications, 2000