A Scalable Logical Coordinates Framework for Routing in Wireless Sensor Networks *

Qing Cao and Tarek Abdelzaher
Department of Computer Science, University of Virginia, Charlottesville, VA 22904
e-mail:{qingcao,zaher}@cs.virginia.edu

Abstract

Routing is one of the key challenges in sensor networks that directly affects the information throughput and energy expenditure. Geographic routing is the most scalable routing scheme for statically placed nodes in that it uses only a constant amount of per-node state regardless of network size. The location information needed for this scheme, however, is not easy to compute accurately using current localization algorithms. In this paper, we propose a novel logical coordinate framework that encodes connectivity information for routing purposes without the benefit of geographic knowledge, while retaining the constant-state advantage of geographic routing. In addition to efficiency in the absence of geographic knowledge, our scheme has two important advantages: (i) it improves robustness in the presence of voids compared to other logical coordinate frameworks, and (ii) it allows inferring bounds on route hop count from the logical coordinates of the source and destination nodes, which makes it a candidate for use in soft real-time systems. The scheme is evaluated in simulation demonstrating the advantages of the new protocol.

1. Introduction

Recent technology has made exciting progress in large-scale sensor networks, which opens the door for myriads of civil, meteorological and military applications. Large-scale sensor networks can be deployed to carry out various tasks without the need for human intervention. Efficient data dissemination among different parts of the network is crucial for overall application performance. Such dissemination hinges on the design and implementation of efficient routing protocols.

Current routing protocols for sensor networks (and more generally for ad hoc wireless networks) broadly fall into two categories; address based [13, 20, 14] and content based [25, 4]. The former category requires an explicit destination address. The latter implicitly defines a set of destinations by their attributes and delivers the data to all matching destinations. It is likely that future sensor networks need both types of routing protocols. Content-based routing may be used as an efficient multicast mechanism that discovers a set of destinations matching given criteria (and returns their addresses to the sender if needed). Address-based routing can then be used to unicast data individually to particular destinations in the content-based groups as dictated by application logic. In this paper, we focus on the latter type and assume that when the address-based routing is needed, the addresses of the destinations have been obtained in advance, presumably through some content based mechanisms.

With the exception of geographic routing schemes [9, 1, 14, 15, 8] (where nodes are addressed by their location), address-based routing schemes are typically not scalable in that their routing state grows as some function of either the network size or the number of active destinations. In contrast, geographic routing needs a *constant* amount of state that is only related to the node's immediate neighborhood. Unfortunately, it also needs location information. While a plethora of localization services have been proposed to estimate node locations using a small number of GPS-enabled anchors [23, 18, 3, 16], accurate localization services remain hard to implement. It has been shown that errors in node location information lead to routing failures [11]. This difficulty motivates the development of scalable routing protocols (i.e., those that maintain a constant amount of state) that do not rely on geographic knowledge. This paper presents logical coordinate routing, which belongs to this category.

The main idea of our scheme is for each node to maintain hop counts to a small number of landmarks. This hop count vector is the logical coordinate of the node. The difference vector between two node vectors represents the distance between them. The routing scheme forwards packets in the direction that minimizes the magnitude of the remaining difference vector. Compared with current routing pro-

* The work reported in this paper was supported in part by the National Science Foundation grants EHS-0208769 and ITR EIA-0205327, DARPA grant F33615-01-C-1905, and MURI grant N00014-01-1-0576.

Proceedings of the 25th IEEE International Real-Time Systems Symposium (RTSS 2004)

tocols, this simple scheme has several advantages. First, it translates the routing problem into a different logical domain in which the state kept on each node is constant (only immediate neighbor coordinates). Moreover, it can encode a configurable amount of topological information that depends on the number of chosen landmarks, which we call *configurable dimensionality*, from which we observe several positive implications. Most importantly, since the logical coordinate dimensions can be arbitrarily enriched by increasing the number of landmarks, logical vectors contain inherent redundancy, which significantly improves robustness with respect to node failures and physical voids. Selecting an appropriate number of landmarks at suitable locations makes it possible to eliminate voids in the logical coordinate space despite their existence in the physical space. Compared to location-based approaches, the logical coordinates directly encode connectivity relationships among nodes, rather than physical proximity. Hence, they reflect and abstract the more relevant topological information in a simple and efficient manner. In this paper, we call routing in the new coordinate space *logical coordinate routing* (LCR). Finally, the logical coordinates of each node can be used to bound the actual hop distance between two arbitrary nodes. As a result, they can be leveraged to predict the delivery delay between nodes, which is of use in soft real-time sensor networks such as EnviroTrack [7] and Mobicast [12].

Recently, other efficient routing protocols emerged that are geographical location independent [21, 17]. Compared with them, our protocol is the first one to use configurable logical dimensionality, and directly encode a bound on hop count. Besides, simulation-based comparisons show that our protocol demonstrates a considerable performance improvement in delivery ratio, especially in the presence of voids.

In the rest of this paper, we present an exploration of the new logical coordinate framework and its possible applications. The paper is organized as follows. Section 2 describes our assumptions, the design of the logical space and its properties. The design of a specific simple-forwarding based routing protocol is presented in Section 3. Section 4 describes our experiments and an in-depth analysis of the data collected. We review related work in Section 5 and conclude in Section 6.

2. Design of the Logical Coordinates

In this section, we present the assumptions, principles and properties of the logical coordinate framework.

2.1. Assumptions

We assume that the nodes are placed on a plane. (This assumption regarding physical space is not to be confused with the configurable dimensionality in the *logical* coordinate space). Planar placement is usually an adequate characterization of many real applications. Second, we consider the position of individual nodes to be relatively static. The static model generally characterizes typical sensor networks well enough. In practice, we may recompute the logical coordinates of each node periodically to account for possible displacement due to wind and other environmental factors. Third, we assume that the approximate placement of landmarks is controllable. For example, they can be purposely placed at the boundaries of the region. We will demonstrate later that the location of the landmarks has considerable impact on the efficiency of logical coordinates. At last, we assume that the communication links between nodes are bi-directional. Although this may not be true in reality, it can be easily achieved by selecting only those links which are bi-directional for communication.

In our framework, nodes need not know their location. It is important to clarify that this paper does not argue against use of localization services. In fact, approximate knowledge of location is essential for many sensor network applications, such as target tracking. However, unlike target tracking where location errors of individual target-tracking nodes can be reduced by averaging across multiple observers, in geographic routing individual node locations play a key role in forwarding decisions. Hence, location information must be accurate not only after aggregation and trajectory fitting, but also at the individual node level. The paper therefore asserts that it is beneficial not to have to rely on the availability of such accurate location information for the benefit of routing services.

2.2. The Logical Coordinate Space

The idea of the logical coordinate approach is partly inspired by the classical distance vector concept in conventional networks in that the key structure in our framework is based on measuring hop counts between nodes. The key difference is that instead of encoding the hop count between *any* two nodes, we encode hop distance to a few reference points (landmarks) only. As we show later, it is advantageous to place the landmarks as sparsely as possible.

After the landmarks are chosen, the logical coordinate framework is constructed as follows. First, each landmark broadcasts a beacon that is forwarded once to all nodes along with a hop count parameter. The hop count is initialized to zero at the landmark and incremented at each hop. Each node that receives the beacon records the shortest distance, in hops, from itself to the corresponding landmark. If multiple beacons from the same landmark are received via different routes, the lowest count is recorded. When multiple landmarks are chosen, every node in the sensor network is expected to receive beacons from all landmarks. Each node consequently records the hop counts between itself and each of the landmarks. We call the vector formed by these hop numbers the *logical coordinate vector*. The dimensionality of this vector corresponds to the number of landmarks. For example, if we choose four landmarks, we build a four-dimensional logical coordinate space. To preserve a consistent order of elements in all logical coordinate vectors, these vectors can be sorted, for example, by unique landmark identity, statically assigned at compile time and carried along in the aforementioned beacons.

After each node records its own logical coordinate vector, the initial establishment of the logical coordinate space is complete. An example of a logical space with four landmarks (denoted by filled black circles) is shown in Figure 1. In this example, 24 nodes are deployed. Since four landmarks are chosen, the logical coordinate space is four dimensional. The four landmarks have coordinates (0,5,3,8), (5,0,8,3),(8,3,5,0) and (3,8,0,5), respectively. Note that we have transformed the target space from a physical plane to four (partially redundant) logical dimensions. Further, the coordinates of landmarks have the distinctive trait that they have one zero element in exactly one dimension; namely, the dimension corresponding to the local landmark. Other nodes have non-zero (positive) coordinates in all dimensions.

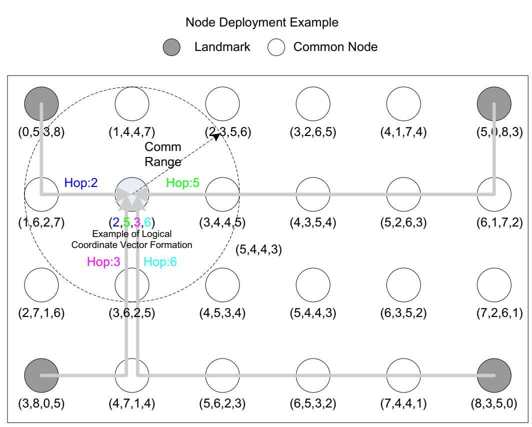

Figure 1. Logical Coordinates Construction Example

2.3. Space Maintenance

The formation of the logical coordinates must be resilient with respect to two kinds of inconsistencies. First, beacon message loss that results in missing coordinates. We call this problem *null coordinates*. In the neighbor beacon exchange, a node that finds out that it has null coordinates compared with its neighbors tries to correct each null coordinate by incrementing the lowest of the corresponding coordinates of its neighbors by one (hop) to update its own coordinate vector.

The second type of inconsistency is where neighboring nodes differ by more than one in some coordinate. This should never occur in an ideal world because neighbors are only one hop away. However, it could be observed in the presence of message loss. In the neighbor beacon exchange stage, if a node finds out that one of its neighbors has a coordinate value that differs by more than one compared with its own (for the same landmark), then the node with the higher-valued coordinate locally decrements it to remove the inconsistency. If a logical coordinate is updated on some node, it will beacon this update to refresh its neighbors. Although this refreshing has the potential to be propagated, in practice, we find that the scope of updates in the neighbor exchange stage is highly limited, and the update process converges very fast.

2.4. Properties and Concepts

Now we present useful concepts of the logical coordinate framework, which are the fundamentals of the design of our new protocol.

2.4.1. The Neighborhood Property The most basic property maintained by the logical coordinate space as described above is the following:

Property 1. *In a correct logical coordinate space, the corresponding coordinates for the same landmark between any two nodes which are mutual neighbors differ by at most* 1.

Proof. This property follows directly from the neighborhood maintenance protocol discussed above. □

2.4.2. Bounded Hop Count From the perspective of soft real-time applications, a very useful property of the protocol is that the hop count along the actual routing path between a source and a destination can be estimated solely from the logical coordinates of the source and destination, as follows.

Property 2. *For any two nodes $V(V_1, ..., V_n)$ and $W(W_1, ..., W_n)$, the hop count of the shortest path between them is lower-bounded by $MAX(|V_1 - W_1|, ..., |V_n - W_n|)$ and upper-bounded by $\sum_i |V_i - W_i|$.*

Proof. This follows directly from Property 1, since for any single hop, the coordinate in each dimension can change by at most one. □

In practice we found that the lower bound is especially useful. The actual hop count was usually found to be *exactly* the lower bound. Hence, this optimistic bound can be used to estimate delay in soft real-time load balanced sensor networks where deadline misses can be tolerated. For those cases that are higher than the lower bound, the marginal offset in the hop count is usually low enough compared with the overall hop number. We will validate this claim thoroughly in the experiments.

2.4.3. The Distance Concept We use vector distance, defined below, as the logical distance between nodes. Thus, for logical coordinate vectors V and W with coordinate elements V_i and W_i, respectively, the distance D between them is defined as:

$$D = \sqrt{\sum_{i=1}^{n} (V_i - W_i)^2} \quad (1)$$

As shown, we use the length of the difference vector $(V_1 - W_1, V_2 - W_2, ..., V_n - W_n)$ as the distance metric between nodes. Observe that the length computed above has no physical geometric interpretation, since the logical dimensions are not orthogonal. However, compared to physical distance, this metric reflects more accurately the topological distance between two nodes in the sensor network graph. If two nodes are connected by fewer hops, the logical distance between them tends to be smaller.

3. The Logical Coordinate Routing Protocol

In the previous section, we outlined the logical coordinate framework and its properties. We now apply these properties to the design of the LCR protocol.

3.1. Basic Protocol Design

In our routing protocol, the node with the least logical distance to the destination is chosen as the next relay. Since each node knows the logical coordinates of its neighbors, this comparison can be done locally. In most cases, greedy forwarding alone guarantees a high ratio of successful packet delivery. We refer to this scheme as the baseline design. We further augment the LCR protocol with techniques for loop avoidance and (logical) void avoidance, described in the following two subsections respectively, in order to address these less common routing anomalies.

3.2. Loop Avoidance

One possible cause of delivery failure in LCR is the presence of multiple nodes that have the same distance to the destination. This may lead to unexpected loops. This situation rarely arises in location-based routing since two nodes rarely share exactly the same distance to a destination. However, the presence of same distance nodes is more common in logical coordinates due to the coarser granularity of coordinate values. Figure 2a shows an example. As shown, although each node chooses the best node it knows in its neighbor table, a packet is forwarded back to its starting point (node 9). One intuitive solution is to record all nodes that the packet has visited in the packet header. However, the packet sizes in sensor networks are usually only tens of bytes, implying it is infeasible to record all nodes the packet has been routed to. We solve this problem by making a tradeoff, recording only a finite moving history of the delivery path in the packet. The length of moving history represents a tradeoff between the memory capacity needed for better loop avoidance and the corresponding control overhead.

One loop avoidance example is shown in Figure 2b. Observe that for purposes of loop avoidance, nodes in the recorded history can be identified by short internal identifiers (chosen at compile time) known only to themselves and their neighbors, as opposed to their full-length logical coordinates.

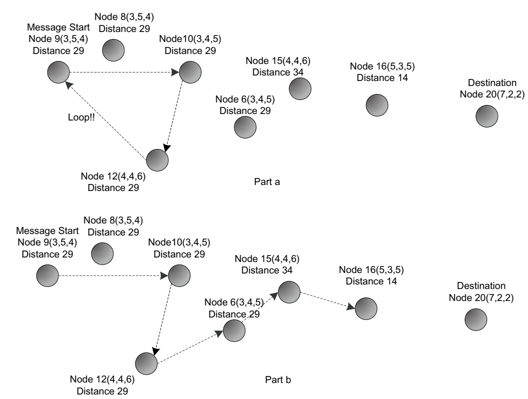

Figure 2. Loop Avoidance using Moving History Recording

Due to the limited size of the moving history record and the large number of nodes in a sensor network, it is still possible that loops may exist. For this purposes, a time-to-live field is added to the packet header, such that packets are dropped when this field reaches a threshold.

3.3. Void Avoidance

Another related problem is that some nodes may have no neighbor that is logically closer to the destination, such as node 6 in the aforementioned example. Clearly this implies that the packet has met a logical void. Our logical coordinate framework, by construction, is much less susceptible to such dead-ends than geographic coordinates. This is because our logical space morphs around physical voids, eliminating local minima. Nevertheless, situations exist when a local minimum arises in the logical space (for example upon node failures). When one node finds that it has no neighbor that is closer to the destination than itself, it will choose the best *upstream* node (i.e., the closest neighbor to the destination). The purpose of this action is to try to deliver the packet to a new node where a new downstream path might exist to the destination. In order to bound upstream forwarding, we use a parameter called *tolerance counter* stored in the packet header. Initialized to zero at the start node, this parameter increases by one whenever the packet is delivered to an upstream node. The packet is dropped if this counter exceeds a predefined threshold, $MaxTolerance$. Observe that since we also keep a moving path history in the packet, the memory of the last few hops helps the packet avoid being delivered back to the local minimum. Also note that the void avoidance mechanism can increase worst-case hop count between source and destination (stated by the bounded hop-count property in Section 2.4.2) by at most $MaxTolerance$.

Of course, the void avoidance approach is, by nature, heuristic, and does not guarantee packet delivery. However, our simulation results validate its effectiveness compared with other routing algorithms.

3.4. Dynamic Issues: Node Failure, Sleep, and Replacement

The main source of dynamic changes in topology is the failure or replacement of old nodes, or topological changes introduced by power management. When new nodes are put into the network, they contact nearby neighbors to retrieve their logical coordinate vectors. The new nodes then construct their own LCVs in the same manner as described in Section 2.3, and use the constructed LCVs for future routing purposes. Observe that in most power management schemes such new nodes are generally awakened to replace those that go to sleep, leveraging redundancy that exists in the sensor network [5, 24]. The new nodes will tend to assume the same logical coordinates as those of the departed ones due to physical proximity. Thus, only localized adjustments need to be made in the logical coordinate grid.

Similarly, when nodes fail, generally other nodes do not need to adjust their logical coordinates. The inherent redundancy in the construction of logical coordinates makes them tolerant to a certain percentage of node failures. Greedy algorithms will generally remain successful in finding a logical route to the destination. If local minima are reached, the void avoidance mechanism introduced in Section 3.3 will resolve the problem.

4. Performance Evaluation

In this section, we present the performance evaluation results of LCR. We start by presenting the simulation setup, followed by a description of the experiments and results. The simulation environment is GloMoSim, a discrete event simulator developed by UCLA. GloMoSim simulates at the packet level, thus allowing us to gather accurate data on a variety of aspects.

The simulation setting is constructed with respect to the settings used in previous similar routing research projects, such as [2] and [14]. There are three settings, each contains a sparse scenario and a dense scenario, denoted by A and B, respectively.

In the following simulations, we choose DSR, GF and GPSR as comparison candidates. We don't compare against DSDV and AODV considering that the work in [2] has demonstrated the performance superiority of DSR compared to these two protocols in ad-hoc environment. We also include a delivery performance comparison between LCR and the virtual coordinate approach proposed by [21] in Section 4.2.1.

Scenario	Sparse(A)	Dense(B)	Region
Scenario 1	30	50	$1500m * 300m$
Scenario 2	120	200	$3000m * 600m$
Scenario 3	72	120	$1250m * 1250m$

Table 1. Simulation Setting

We use a reliable MAC protocol in order to isolate delivery failures due to the routing layer from those due to the MAC layer. A reliable MAC-layer is currently available for MICA II motes [22] (our target implementation platform) since the recent release of TinyOS 1.1 [10]. The evaluation is focused on comparing the ability of routing protocols to exploit topological connectivity information. Performance is measured in data packet delivery ratio, path hop counts, and number of routing protocol packet transmissions. These metrics tend to be independent of MAC layer details (other than reliability). Thus, while we choose 802.11 in the MAC layer, results presented in this section could be generalized to any reliable MAC protocol. We assume that individual nodes have a communication range of 250m. Each packet sent in the network has a TTL of 64.

Our evaluation contains two parts. The first part explores optimal parameter settings of our protocol and justifies design decisions. In the second part, we evaluate the routing protocol performance.

4.1. Evaluation of Design Choices

Different design choices in LCR can significantly influence its performance. In this section, we give a quantitative evaluation of their effects. The evaluation below is for an underloaded network. Effects of congestion are investigated in future work.

4.1.1. The Impact of Distance Metrics As mentioned earlier, our distance metric does not have a physical interpretation because the coordinates are not orthogonal. Hence, it is important to evaluate its performance. In this section, we compare the packet delivery ratio of routing based on this distance metric compared to that based on other distance metric candidates. In particular, we compare against the common metric of manhattan distance, which simply adds up the absolute differences in coordinate values in all dimensions.

Four landmarks are positioned on the boundary of the area. Zero tolerance for upstream packets is assumed (i.e. $MaxTolerance = 0$). No loop-avoidance measures are adopted. We select all pairs of nodes that can reach each other and let them send one packet to each other. The packet delivery ratio is plotted for two greedy routing policies: one that minimizes the difference vector length (DV Length) as described in this paper, and one that minimizes manhattan distance. All scenarios tested are dense (set A) scenarios.

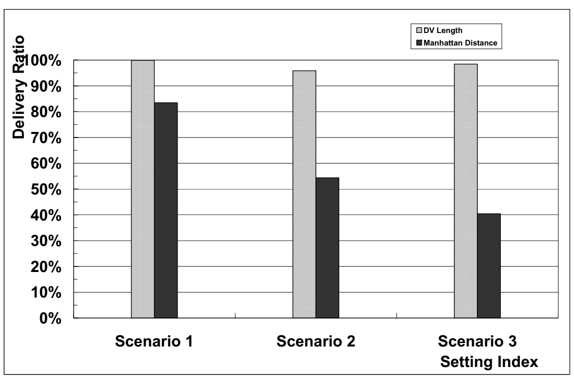

Figure 3. Impact of Distance Definition

Figure 3 shows the simulation results. We observe that the difference vector length approach performs significantly better than the manhattan distance approach. We also notice that LCR successfully delivers almost all packets even without the tolerance and loop-avoidance optimizations in dense scenarios, which reflects the inherent delivery properties of the logical coordinates framework.

4.1.2. The Impact of Landmark Choice Another important aspect of the design of LCR is how to choose the landmarks wisely. In this experiment, we investigate the effect of landmark placement and the effect of the number of landmarks on routing performance. The simulation shows six different landmark configurations that differ in landmark positions and count as depicted in Figure 4. Random landmark placement (labeled *random*) is compared to uniform placement at the network circumference (labeled *corner* for different numbers of landmarks and different node densities. Again, we assume zero $MaxTolerance$. Considering that packets must contain the destination's logical coordinates, and that the size of the packet header is limited, we cannot choose too many landmarks. Fortunately, we notice that performance improvement reaches diminishing returns with as few as four landmarks. At this point, the routing service finds almost all routes available. Considering that, in these experiments, we do not use tolerance and loop avoidance measures, these results are quite encouraging and serve as a lower bound for actual performance when such measures are invoked.

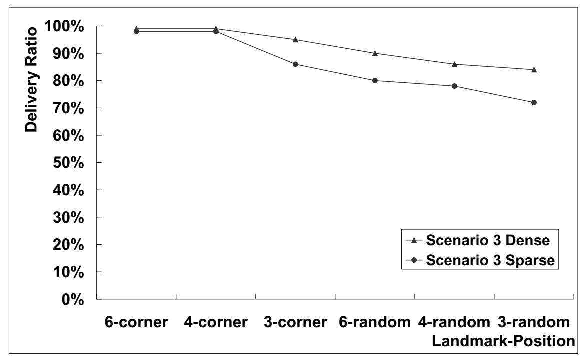

Figure 4. Impact of Landmark Choice

4.2. Routing Protocol Performance Evaluation

4.2.1. Packet Delivery Ratio Figure 5 shows the packet delivery ratio under different scenario settings. We ensure that the network is not partitioned. Traffic is generated by the simplified scenario in which each pair of nodes alternately exchange packets over the whole simulation period. We deliberately choose the period to be long enough to avoid the effect of congestion. Precise location information is assumed to be available for GPSR and greedy GF. The delivery success ratio is evaluated by recording how many packets each node receives during the whole period. It is no surprise that under such assumptions, DSR achieves a 100% delivery ratio in all settings. We don't draw the DSR line for legibility. Theoretically, GPSR should also achieve a 100% delivery ratio. In reality, its actual performance is slightly lower because the TTL of a single packet is set to 64. In some cases, GPSR packets start their traversing process, resulting in much longer (suboptimal) paths causing the TTL to expire. Some packets are therefore dropped. LCR performs comparably well to GPSR and significantly better than greedy GF. This performance is measured when no localization error exists.

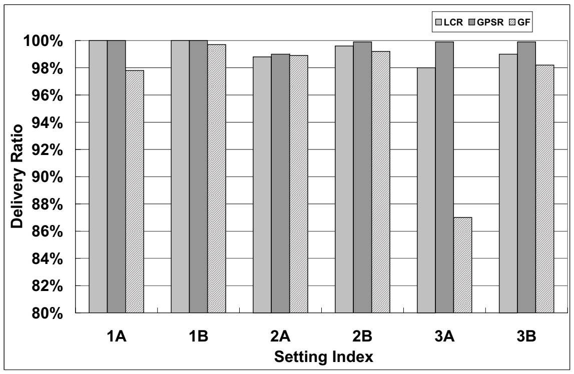

Figure 5. Comparison of Delivery Ratio

Previous work, such as [11], has demonstrated that greedy GF suffers a substantial performance degradation when the localization service can not provide accurate location information. We show in Figure 6 that a similar degradation is seen with GPSR. As shown in this graph, the performance of GPSR is severely affected when the localization error exceeds 40% of the individual node communication radius. In fact, even when nodes are dense, if the localization error is as large as the communication range, the performance of GPSR is still severely undermined. On the other hand, the performance of LCR is not affected by localization errors, thus making it more preferable in scenarios where accurate location information is not available.

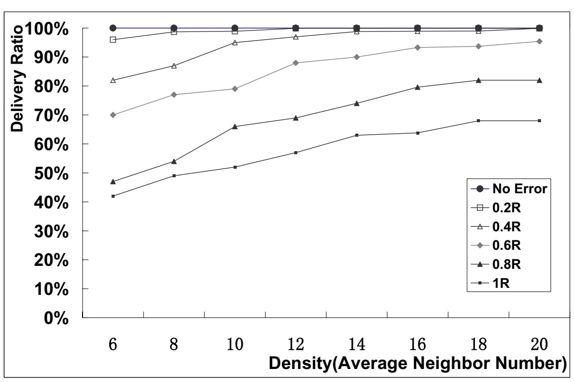

Figure 6. Delivery Ratio of GPSR Under Location Error

Recent previous work also proposed similar network coordinate encoding services that do not require geographical location. It is thus very interesting to provide direct comparison between LCR and the previously proposed services. In this paper, we compare LCR to the virtual coordinates approach mentioned in [21]. To provide a precise and fair comparison, we use their scenario setting. We also use the published results in [21] regarding geographical locations and virtual coordinates. The setting contains two different densities, one node per 12.5 square units and one node per 19.5 square units, respectively. There are four network size settings, from 50 to 3200 nodes. The nodes are uniformly deployed. Figure 7 shows the comparison results between LCR and the virtual coordinate approach. In LCR, we do not use loop avoidance and void avoidance measures, since it is conceivable that both of these optimizations can be applied to either routing framework. By doing this, we provide a fair comparison of the inherent packet delivery properties of both network coordinate encoding schemes.

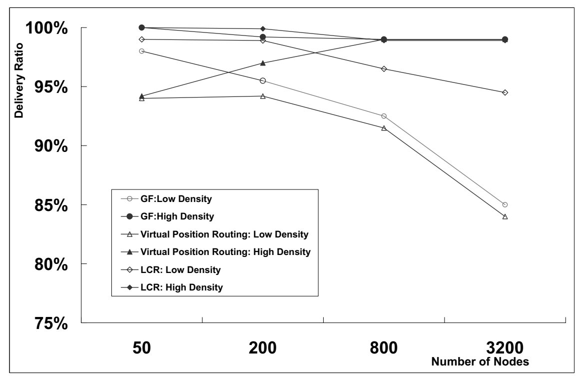

Figure 7. Comparison of Logical Coordinate Routing Schemes

As shown in Figure 7, when the density is high, there is little performance difference between the three types of greedy routing approaches. On the other hand, when the density is low, we observe a considerable advantage to LCR.

4.2.2. Routing Path Length Figure 8 shows a comparison of the packet delivery path length distribution of three routing protocols, DSR, LCR and GPSR (with different localization errors expressed as a percentage of the communication radius). The data presented are concerned with the distribution of the number of hops beyond the shortest hop count. For example, 1 means that the route found by the protocol in question is one hop longer than the best route. We use scenario 1 with 50 nodes deployed. The data are the results of ten randomized simulation rounds. For GPSR, we also consider the presence of localization errors and the corresponding results are plotted accordingly. Only those packets that are successfully delivered are included in the analysis.

When no localization error exists, GPSR performs the best. However, when the localization error is 40% of the

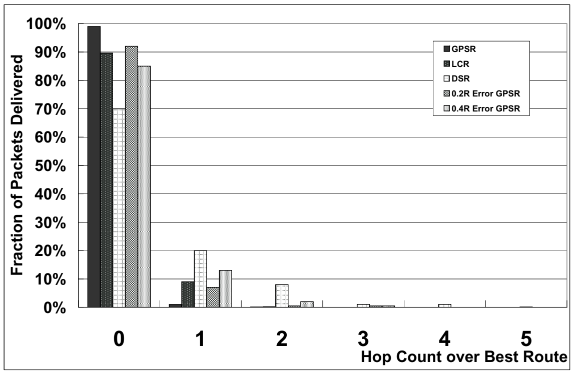

Figure 8. Packet Path Length Beyond the Best Route

communication range, the performance of GPSR is degraded and it becomes inferior to LCR.

DSR and LCR are both localization-error free, but the performance of DSR in terms of packet length is degraded by its proactive caching schemes, which causes packets to be aggressively sent along sub-optimal routes. As a result, it has a longer packet path than both LCR and GPSR.

Packet path optimality is closely related to power consumption. In sensor networks, one of the main sources of power consumption is the transmission of packets. As a result, Figure 8 indirectly indicates the power-efficiency of a particular routing scheme. If no localization errors exist, GPSR should be the most energy-efficient. However, when localization error is taken into account, LCR is better. We recognize that MAC-layer effects, such as message retransmissions, have a marked influence on power consumption. However, we do not investigate these effects because they are MAC-specific and are not an inherent property of our protocol.

4.2.3. Path Hop Prediction and Real Time Applications We now study the accuracy of hop prediction promised by Property 2. An estimate of path length is particularly useful for soft real time applications. We show that prediction according to the lower bound computed from the logical coordinates of the source and the destination is particularly accurate in practice.

We simulate six scenarios each for fifty rounds with different deployments. For each pair of nodes, the prediction and the actual hop number are compared. Figure 9 shows the distribution of the correct prediction ratio (i.e., what fraction of the time the lower bound is equal to the actual path length). As shown, under all six situations, the lower bound prediction is correct at least 70% of the time. Furthermore, for nearly all cases where prediction is wrong, the actual path length is only one hop longer than the lower bound. Observe that the lower bound is trivially correct if the source or destination is one of the landmarks since one of the coordinates would then explicitly measure the hop distance to that landmark.

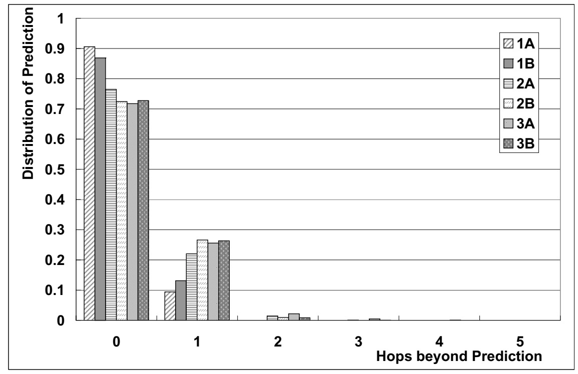

Figure 9. The Prediction of Hop Count

4.2.4. Routing Protocol Overhead Figure 10 shows a comparison of the routing protocol overhead of three routing protocols, measured by the number of protocol packets sent during the total simulation period. We still use scenario 1 with 50 nodes deployed. We gradually increase the number of data sources to monitor the protocol overhead. The packet transmissions are initiated from different nodes. We only test non-partitioned network scenarios and assume each node sends only one packet. DSR is a reactive routing protocol, which means its protocol traffic increases with the number of transmissions. We also notice that due the aggressive caching, the increase in transmitted packets has a diminishing effect on the increase of routing overhead.

Both LCR and GPSR send out beacons periodically with no respect to the number of data transmissions. Consequently, both LCR and GPSR send out roughly a constant number of protocol control packets. LCR utilizes a one-time flooding process to broadcast the location beaconing packets. As a result, LCR requires a higher routing protocol overhead than GPSR.

4.2.5. Void Avoidance We evaluate in this section the performance of different routing protocols in the presence of void areas. We also generalize the results to the discussion of the robustness of LCR, especially in the face of node failures. We simulate the presence of voids and the failure of nodes by reducing active node density, since both voids and node failures essentially decrease the number of usable nodes. We use scenario 3 in this experiment. We reduce the average number of neighbors per node from twenty to six in steps of one and we keep the network unpartitioned. In order to make comparisons between different routing protocols under realistic scenarios, we also include the perfor-

／	DSR	LCR	GF	GPSR
Principle	ID Based	Logical Coordinates	Location Based	Location Based
Protocol Overhead	Large	Small	Small	Small
Delivery Ratio	Theoretically Perfect	High	Moderate - High	Theoretically Perfect
Route Optimality	Good	Better	Best when not degraded	Best when not degraded
Degradation	No	No	Yes	Yes
Void Avoidance	Excellent	Excellent	Moderate	Excellent

Table 2. Routing Protocol Performance Overview

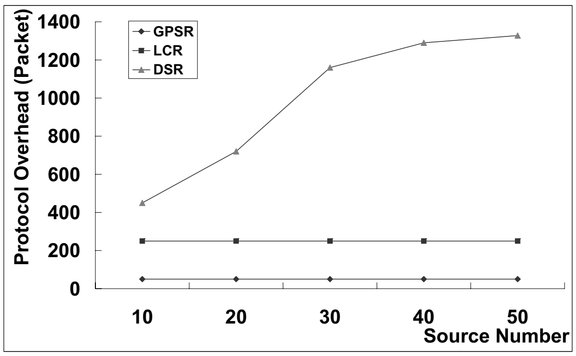

Figure 10. Routing Protocol Overhead

mance of GPSR under localization errors. The results of this experiment are shown in Figure 11.

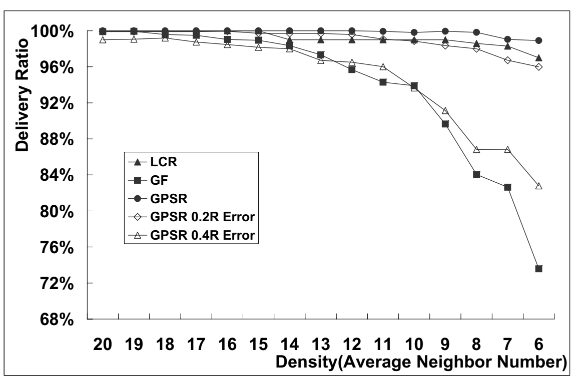

Figure 11. Delivery Ratio under Different Node Density

In these experiments, DSR realizes a 100% delivery ratio, and thus is not plotted. We set the TTL of individual packets to 64. For this reason, some GPSR packets are dropped due to the suboptimal perimeter traversing process. Greedy GF has no mechanism for dealing with voids. Its performance therefore degrades rapidly. The performance of GPSR under localization errors also degrades considerably at higher errors. In contrast, LCR reliably delivers the great majority of packets even in the presence of voids in sparser networks.

4.2.6. Performance Evaluation Summary Based on the experimental results above, we conclude that every protocol has its desirable features. Although DSR and GPSR are two schemes that both theoretically guarantee a 100% delivery ratio, they have drawbacks in sensor networks. The route discovery process makes DSR less suitable in terms of packet overhead, while GPSR can be severely degraded by localization inaccuracy. The design of LCR avoids these problems. Like DSR, it is not affected by localization errors thus guaranteeing a better delivery ratio than greedy GF and degraded GPSR. Its protocol overhead, however, is low and it exhibits exceptional ability to avoid voids in the network. Table 2 summarizes the different characteristics of the routing protocols studied in this paper and serves as a guide for future protocol design in wireless sensor networks.

5. Related Work

Previous literature on ad hoc routing contains many valuable protocols, each with varying assumptions and applicability. Two trends of address-based routing draw our attention. The first type makes routing choices based on real geographical locations [9, 1, 14, 19, 6, 8]. The second type, proposed relatively recently, tries to explore the possibility of routing without geographical information, but with certain types of substitutes through various network encoding approaches [21, 17]. For geographic routing, one potential problem is the performance degradation introduced by location inaccuracy. Since it is usually not economical to install one GPS receiver on each node, localization services must be leveraged [23, 3, 18, 16, 11]. However, these services introduce location inaccuracy, which is known to have serious consequences on the performance of routing protocols (Section 4.2.1). Although such inaccuracies might be alleviated by introducing more complex localization services, the additional complexity either requires more costly hardware or introduces more overhead in the network.

It is therefore natural to search for location free routing paradigms for sensor networks that preserve the simplicity and efficiency of location-based schemes. Protocols in [21] and [17] are of this type.

A preliminary comparison of delivery ratio between LCR and [21] has shown that LCR has a higher delivery ratio, especially in sparse scenarios. We also point out that our protocol requires only a one hop neighbor table to achieve satisfactory performance, while both [21] and [17] require that at least two hop neighbor information is collected. Last but not least, our protocol distinguishes itself by its inherent suitability for soft real-time applications. We hope by leveraging these properties of LCR, it becomes more natural to use in emerging sensor networks.

6. Conclusions and Future Work

In this paper, we presented a simple logical coordinate framework, together with a scalable (i.e., constant state) routing protocol, LCR, that uses logical coordinates in lieu of geographic information. Being location-independent, LCR has the distinct advantage of independence from localization errors. It is attractive for soft real time applications in that bounds on the hop count between any pair of nodes can be estimated from the source and destination coordinates. Moreover, its logical distance metrics have the potential of masking the existence of physical voids and irregularities. We performed extensive simulation experiments to compare LCR with other logical coordinate frameworks, observing a performance advantage in the treatment of voids. Based on these results, we conclude that our protocol performs very well in realistic wireless sensor networks where voids abound, accurate localization is unavailable or costly, and state must be minimal due to resource constraints.

References

[1] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia. Routing with guaranteed delivery in ad hoc wireless networks. *Wireless Networks*, 7(6):609–616, 2001.

[2] J. Broch, D. A. Maltz, D. B. Johnson, Y.-C. Hu, and J. Jetcheva. A performance comparison of multi-hop wireless ad hoc network routing protocols. In *Proceedings of MobiCom*, 1998.

[3] N. Bulusu, J. Heidemann, and D. Estrin. Gps-less low cost outdoor localization for very small devices, 2000.

[4] A. Carzaniga, D. Rosenblum, and A. Wolf. Content-based addressing and routing: A general model and its application, 2000.

[5] A. Cerpa and D. Estrin. Ascent: Adaptive self-configuring sensor networks topologies. In *Proceedings of the IEEE Infocom*, 2002.

[6] S. B. et al. A distance routing effect algorithm for mobility (dream). In *Proceedings of MobiCom*, 1998.

[7] T. A. et al. Envirotrack: Towards an environmental computing paradigm for distributed sensor networks. In *International Conference on Distributed Computing Systems (ICDCS)*, 2004.

[8] A. Z. Fabian Kuhn, Roger Wattenhofer. Worst-case optimal and average-case efficient geometric ad-hoc routing. In *Proceedings of MobiHoc*, 2003.

[9] G. Finn. Routing and addressing problems in large metropolitan-scale internetworks. In *Technical Report*, volume ISI/RR-87-180. USC/ISI, March 1987.

[10] W. for TinyOS. http://webs.cs.berkeley.edu/tos/.

[11] T. He, C. Huang, B. Blum, J. Stankovic, and T. Abdelzaher. Range-free localization schemes in large scale sensor networks. In *Proceedings of MobiCom*, 2003.

[12] Q. Huang, C. Lu, and G.-C. Roman. Spatiotemporal multicast in sensor networks. In *First ACM Conference on Embedded Networked Sensor Systems (SenSys'03)*, 2003.

[13] D. B. Johnson and D. A. Maltz. Dynamic source routing in ad hoc wireless networks. In *Mobile Computing*, volume 353. Kluwer Academic Publishers, 1996.

[14] B. Karp and H. T. Kung. GPSR: greedy perimeter stateless routing for wireless networks. In *Proceedings of Mobicom*, 2000.

[15] Y.-B. Ko and N. H. Vaidya. Location-aided routing (lar) in mobile ad hoc networks. In *Proceedings of MobiCom*, October 1998.

[16] R. Nagpal. Organizing a global coordinate system from local information on an amorphous computer. In *A.I. Memo 1666, MIT A.I. Laboratory*, 1999.

[17] J. Newsome and D. Song. Gem: Graph embedding for routing and datacentric storage in sensor networks without geographic information. In *Proceedings of the SenSys 2003*, 2003.

[18] D. Niculescu and B. Nath. Dv based positioning in ad hoc networks. In *Journal of Telecommunication Systems*, 2003.

[19] D. Niculescu and B. Nath. Trajectory based forwarding and its applications. In *Proceedings of Mobicom*, 2003.

[20] C. Perkins and E. M. Royer. Ad-hoc on demand distance vector routing. In *WMCSA'99*, February 1999.

[21] A. Rao, C. Papadimitriou, S. Shenker, and I. Stoica. Geographic routing without location information. In *Proceedings of Mobicom*, 2003.

[22] T. M. sensor node data specifications. http://www.xbow.com/products.

[23] Y. Shang, W. Ruml, Y. Zhang, and M. P. J. Fromherz. Localization from mere connectivity. In *Proceedings of Mobihoc*, pages 201–212. ACM Press, 2003.

[24] T. Yan, T. He, and J. A. Stankovic. Differentiated surveillance for sensor networks. In *Proceedings of ACM SenSys*, 2003.

[25] H. Zhou and S. Singh. Content based multicast (cbm) in ad hoc networks. In *Proceedings of MobiHoc*, pages 51–60, 2000.