Robust and Timely Communication over Highly Dynamic Sensor Networks

Tian He§, Brian M. Blum†, Qing Cao‡,
John A. Stankovic†, Sang H. Son† and Tarek F. Abdelzaher‡

†Department of Computer Science, University of Virginia
§Department of Computer Science and Engineering, University of Minnesota
‡Department of Computer Science, University of Illinois, Urbana-Champaign

Abstract

Highly dynamic sensor networks, such as mobile robotic sensor networks, have been applied in various kinds of application scenarios such as real-time planet exploration and deep-ocean discovery. In these types of networks, mobility and energy management protocols change the connectivity among the neighboring nodes quickly. Traditional state-based protocols, designed for static and/or low-mobility networks, suffer excessive delay in updating their routing or neighborhood tables, leading to severe packet loss and communication delay in the highly dynamic situations. To provide robust and timely communication, we exploit the concept of *Lazy-Binding* to deal with the elevated network dynamics. Based on this concept and the knowledge of the node positions, we introduce Implicit Geographic Forwarding (IGF), a new protocol for highly dynamic sensor networks that is altogether *state-free*. We compare our work against several typical routing protocols in static, mobile and energy-conserving networks under a wide range of system and workload configurations. In the presence of mobility and other dynamics, IGF achieves as much as 10 times improvement in the delivery ratio and significant reduction in both the end-to-end delay and control overhead. In addition to extensive simulations, we also implement and evaluate the IGF protocol on the Berkeley mote platform.

I. INTRODUCTION

Highly dynamic sensor networks, such as mobile robotic sensor networks, have been widely used to explore environments that are difficult and dangerous for humans. In the exploration missions of the red planet, scientists employ the robotic sensor devices (Rover) to discover the possibility of water activity. In the deep-ocean exploration, robotic sensors are used to access the risk of potential earthquakes, volcanoes and tsunamis. In addition, robotic sensors are used to investigate dangerous sites such as radioactive environments and mine fields. These robotic sensors are normally equipped with various kinds of sensors such as cameras, spectrometers and magnetometers. They can localize themselves in real-time through either GPS [1] or position tracking [2]. Distributed data processing, such as the collaborative exploration for map construction [3], requires a team of robotic sensors to communicate with each other constantly in real-time. This type of communication imposes several challenges. First, when robotic sensors move, the constant changes of the connectivity among the neighboring nodes make it difficult to maintain freshness of the routing states in traditional state-based routing protocols. Second, energy management protocols [4]–[6] transit the robotic sensors into and out of sleep states, and their participation in the network becomes probabilistic at any given point in time. These unique challenges demand a new routing solution. In this paper, we exploit the concept of **lazy-binding**, which is widely used in other research areas, such as the programming language design and operating systems. Specifically here, we define *lazy-binding as deferring mapping the system physics (e.g., the network topologies) into the volatile states (e.g., the route state), required by a certain operation, to the last possible moment allowed by the operations.* Since lazy binding defers binding the volatile states as late as possible, it enables the system to cope with real-time changes in the network topology. Our first installment based on this concept is a location-based routing protocol, called Implicit Geographic Forwarding (IGF). IGF allows a sender to determine a packet's next-hop online in real-time. By combining lazy-binding and location-address semantics, IGF becomes a pure state-free protocol, which does not depend on the knowledge of the network topology or the presence/absence of other nodes. This characteristic of being state-free is valuable to the highly dynamic sensor networks, as it supports fault tolerance and makes protocols robust to real-time topology shifts or node state transitions. Further, a state-free solution eliminates the bandwidth-consuming packets required in the state-based solutions for routing and neighbor table upkeep.

The remainder of the paper is organized as follows: Section II motivates the need for lazy-binding in highly dynamic networks. Section III describes the IGF protocol in detail. Section IV presents our simulation experiments and analysis in mobile and other environments. Section V describes our implementation on the MICA2 platform and its evaluation. We discuss the state-of-the-art and future work in Sections VI and VII, respectively. We conclude the paper in Section VIII.

II. THE MOTIVATION FOR LAZY-BINDING

Advances in the protocol design [7]–[11] continuously expand our ability to deal with the high dynamics inside the network. These protocols have been designed with robustness in mind, however, the level of fault tolerance is usually designed to adapt to occasional node failures and infrequent topology migration. In order to cope with the elevated transition of network topologies, the state-based solutions are required to refresh the routing states in real time to reflect changes, which consequently introduces significant overhead and network congestion. Eventually, the performance of these algorithms might degrade dramatically, as the real-time maintenance of the routing states might not keep up with the transition rate of the network topologies. We observe that **the delay** between **the time when a physical network topology maps to the routing states** and **the time when these states are actually used for packet forwarding** is the root cause of state invalidation and routing failures. We term this delay as the binding delay. A long binding delay leads to a high probability that recorded states are invalid by the time they are used. This problem increases as the network dynamics increase. In addition, since routing states are volatile and become outdated at a much faster rate in highly dynamic networks, it is inefficient to maintain state proactively and eagerly. According to the binding time, we categorize the routing protocols into four categories as shown in Figure 1.

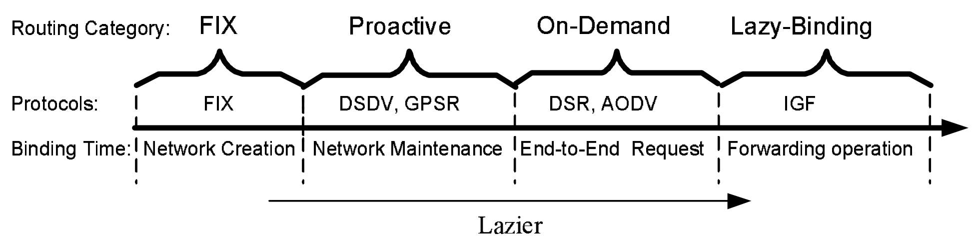


Fig. 1. Difference in Binding Time

- Fixed routing schemes are rarely used due to their rigid early-binding at the deployment time. The binding delay of this type of network could be infinite.
- Proactive schemes such as DSDV [10] and GPSR [11] maintain the network states aggressively. The routing states are refreshed regardless whether there is need of data delivery. This eager and proactive binding property is suitable for the networks with a small rate of topology change. The binding delay is the interval between consecutive routing updates.
- On-demand algorithms such as DSR [8], AODV [9] and Directed Diffusion [7] bind the routing state to physical topologies with a lazier approach - On-demand Route Discovery. The on-demand property allows them to defer the binding of the routing states to the physical network topologies until there is a need for end-to-end delivery. Those schemes have been proven effective [12] in dealing with moderate mobility and failures. The binding delay of on-demand algorithm is the time since the route discovery.
- Different from the on-demand schemes, the IGF protocol proposed in this work goes one-step further. It defers the binding of the routing states to the physical network topology *until the packet forwarding operation actually happens at a sending node.* This design allows: 1) The elimination of the communication overhead to maintain the state proactively, reducing the unnecessary update of the volatile routing states, 2) The real-time detection of the node failure, migration, and transition into a sleep state and 3) The real-time utilization of recently awoken or newly arriving nodes.

III. IGF PROTOCOL DESIGN

In this section, we introduce the IGF protocol as an exemplar instance of the lazy-binding concept applied to routing.

A. System Model and Assumptions

IGF is targeting to the high-end sensor networks (e.g., mobile sensor networks), where each sensor node can obtain its location (x, y) through GPS [1] or a position tracking technique [2]. The IGF communication supports the location-address semantic, in which locations are specified as the routing destinations, instead of using a particular node ID. This location-address semantic are valid in many sensor networks, because sensor data, such as temperature readings, are normally tagged with the location-context, and therefore can be addressed directly by the location, eliminating the overhead to translate the target destinations into a set of node IDs. Since the packet size in high-end sensor networks is relatively large, our main design uses handshaking to avoid the hidden and exposed terminal problems in wireless communication [13], and an alternative solution for small-packet delivery (e.g., Tinyos Message) is discussed separately in Section III-H.4. For the sake of simplicity, we describe IGF in Section III-B assuming a sufficient node density. The issues related to the density, radio irregularity and localization error are resolved in the later sections.

B. IGF Details

We begin our introduction of IGF with an straightforward example. Figure 2 depicts a scenario where the node S is transmitting a packet towards the final destination D. We define the dark nodes within the 60 degree sector (shown in Figure 2) as forwarding candidates (We address the case when there are no candidates inside the specified forwarding area in section III-G). Among these candidates, we highlight two nodes, R and A, to represent the chosen next-hop and an alternate "competing" node, respectively. In addition, the gray node N represents a node within the communication range of S that is not a candidate node. When the node S initiates a packet transmission, the communication handshake goes through following steps: (the timeline of the IGF Handshake is shown in Figure 3)

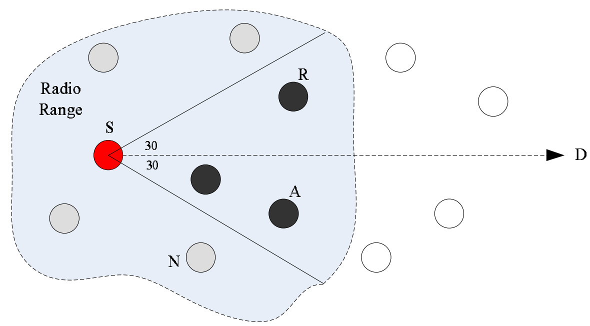


Fig. 2. Forwarding Area for Source S

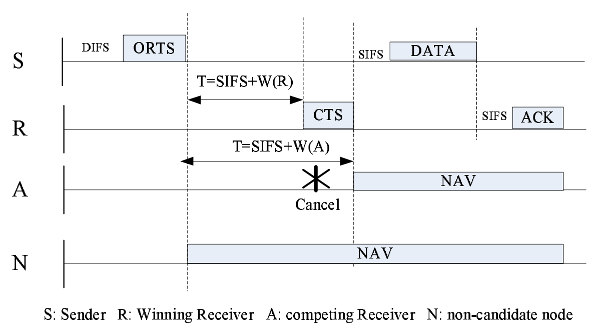


Fig. 3. IGF Handshake

1) ORTS PHASE: With modifications to the 802.11 DCF MAC protocol, the IGF handshake begins when the sender S's Network Allocation Vector (NAV) timer is zero and it senses an idle channel for DIFS time; at this point the node S sends, via broadcast, an Open RTS (ORTS) packet. This ORTS packet contains the locations of the sending node S and the final destination D.
2) CTS-WAIT: While all nodes within the communication radius of the node S receive and process this ORTS packet, only the forwarding candidates (dark nodes) set a CTS_Response timer (T_{cts_wait}). This timer controls an appropriate amount of time that a forwarding candidate must wait before responding to the received ORTS packet. The value of T_{cts_wait} can depend on the link quality, the progress in distance towards the destination and/or the energy remaining at the potential receiver. The nodes that are not forwarding candidates (gray nodes) set their NAV timer in accordance with 802.11 semantics to avoid interference with this ongoing transmission.
3) CTS: While all forwarding candidates set their CTS_Response timer, only a single node, with the shortest T_{cts_wait} value (the node R in the Figure 2 scenario), responds to the ORTS with a CTS packet. To prevent multiple responses, other forwarding candidates overhearing this CTS packet cancel their timers and set their appropriate NAV timers. In addition, the sender S, having already received a valid CTS packet, ignores further CTS packets heard in response to the now antiquated ORTS. We consider the IGF lazy-binding done, when the sender S decides that the node R is the receiver for this packet.
4) DATA: After the sender S is bound with a specific receiver (R), the sender S sends DATA to the node R.
5) ACK: The node R acknowledges the sender S, if DATA is received successfully.

C. More on Forwarding Candidates

This section gives a detailed discussion on how a node determines whether it is a valid forwarding candidate. We prefer that every node within the forwarding area is capable of hearing one another, to prevent the interference among the forwarding candidates. Accordingly, as depicted in Figure 2, we choose the candidate nodes that reside within a ± 30-degree angle of the line connecting the sender and the final destination. Using the sender and the receiver's own location, as well as the final destination location, each node (e.g., the node R) apply simple trigonometry to test whether itself is within the forwarding area. The formula to calculate the angle $\angle RSD$ in Figure 2 is:

$$Degree_{\angle RSD} = \text{acos}\left(\frac{|SR|^2 + |SD|^2 - |RD|^2}{2|SR||SD|}\right) \tag{1}$$

In the ideal case, the shape of the forwarding area ensures all forwarding candidates, responding to an ORTS packet, are located within the communication range of one another; this eliminates the chance multiple CTS are sent in response to a single ORTS packet. However, in reality, due to an irregular communication radius [14], the node A might still fail to know that a response to the node S's ORTS has already been transmitted by the node R. In this rare case, the sender S needs to resolve duplicate CTS packets by choosing only one of those responses.

As stated before, the neighboring nodes that receive an ORTS, but are not within the forwarding area, simply set their NAV timer to reflect the duration of communication. This prevents collisions due to the hidden terminal problem [13].

D. More on Setting Response Wait Times

This section provides more discussion on how to set the T_{cts_wait} value. Having determined that it is within the forwarding area of communication, a node can adopt different policies in setting its T_{cts_wait} according to any combination of available metrics including the reception quality of the link, the progress in distance toward the destination, the energy remaining at the receiver, the statistics of packet loss, the processor load or the single hop delay. While many metrics can be used to decide the T_{cts_wait} delay according to the application specifics, without loss of generality, in the current IGF implementation, we adopt following formula:

$$\begin{aligned} F &= \frac{W_P * (1 - progress/radius) + W_R * rand()}{W_P + W_R} \\ T_{cts\_wait} &= SIFS + (DIFS - SIFS) * FF \in [0, 1) \end{aligned} \tag{2}$$

In Equation 2, $progress$ is the advance in distance toward the destination; $Radius$ is the nominal radio range. $Rand()$ generates a random number between 0 and 1; W_P and W_R are the weights of progress and randomness, respectively; $SIFS$ delay is the Short Inter Frame Spacing and $DIFS$ delay is the Distributed Inter Frame Spacing as defined in the 802.11 standard. Equation 2 probabilistically allows the nodes that relay packets further to wait for a smaller period of time before responding. In addition, the randomization included in Equation 2 can disperse the system workload among multiple equally eligible nodes. It should be noted that Equation 2 is designed to be compatible with the timing rule of 802.11 DCF by guaranteeing: 1) The minimum value of T_{cts_wait} is larger than or equal to the SIFS delay. 2) The maximum value of T_{cts_wait} is smaller than the DIFS delay to prevent other nodes from initiating a new transmission.

E. More on identifying a unique candidate

If more than two forwarding candidates choose similar T_{cts_wait} values (within propagation delay τ), the transmission of CTS would overlap each other, leading to collision. This section provides analysis on the chance of collision under different node densities. Here we use the time slotted approach (e.g. in ALHOA and CSMA) to analyze the performance of the contention-based protocols and establish a system model. The analytic result from the slotted approach serves as the worse-case bound of the un-slotted case. Let N_{node} be the average number of competing nodes within the forwarding area and K_{slot} be the number of back-off slots. A CTS packet encounters a collision when it overlaps with the transmission of at least one other CTS packet from other competing nodes (two or more CTS packets choose the same slot). A unique candidate can be identified as long as the sender receives at least one CTS response from any node within the forwarding area. According to the generalized birthday problem [15], the expected number of slots containing exactly one CTS packets with N_{node} competing CTS packets is :

$$E(N_{node}) \ = N_{node}\left(1 - \frac{1}{K_{slot}}\right)^{N_{node}-1} \tag{3}$$

According to Equation 3, we plot $E(N_{node})$ values under different N_{node} and K_{slot} settings. Figure 4 suggests that the collision-free slots increase almost linearly with the total slots available.

We also simulate the process to identify a unique candidate. Figure 5 shows the probability of success under different N_{nodes} and K_{slots} values. Figure 5 indicates that with a sufficient and reasonable number of back-off slots (e.g., 20) the success ratio approaches 100%.

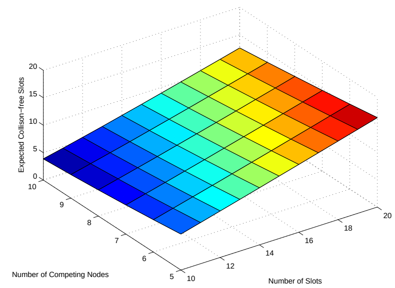


Fig. 4. Expected number of slots that are collision-free

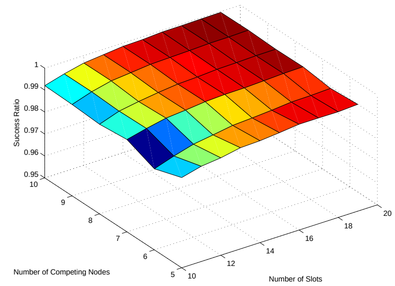


Fig. 5. Success ratio in identifying unique candidates

F. About lazy binding in IGF

IGF is an extension of the location-based protocols with the addition of lazy binding. In the location-based protocols such as GPSR, routing depends on up-to-date local neighborhood tables. Normally the neighbor table is updated through periodic beaconing. The binding of a specific forwarding node to a certain geographic location is eagerly established when the neighboring nodes exchange beacons. This eager-binding would be invalid quickly due to node mobility or sleep state transitions, which lead to stale routing information and unnecessary beacon exchanges. Moreover, this eager binding is not synchronized with the packet forwarding operations. In GPSR, AODV, DSR, if the chosen forwarding node of a sender fails or moves out of range, the MAC layer of the sender drops the packet and notifies the network layer about the routing failure. The network layer has to resolve this failure by attempting another backup route if available, which might be invalid too, or alternately waiting for an update to the neighborhood table, suffering a latency proportional to the beacon period in a scale of seconds. In contrast, IGF adopts lazy binding to discover the next hop the instant it is needed. The worse case back-off delay introduced by lazy binding is tens of microseconds according to the 802.11a standard. This is four orders of magnitude shorter than the period of the neighbor table update through beaconing found in other protocols. The worse case back-off delay in our implementation on the MICA platform is higher due to a low data rate; however, it is still two orders of magnitude shorter. In addition, the route maintenance found in Directed Diffusion [7], DSR [8], AODV [9] and LAR [16] normally takes at least tens of milliseconds or seconds to fix a broken link (depending on the size of network and the cause of failure). In contrast, IGF binds the node that is able to forward the packets, moments before (about 50us) the actual forwarding operation takes place. This lazy binding property dramatically reduces the chance that packets are forwarded to a node that fails or moves out of range. As a result, IGF shows as much as 10 times performance improvement in the delivery ratio when compared with several classical and state of the art solutions in the presence of high rate changes of the network topology.

G. Optimizations for Sparse Networks

IGF targets sensing-covered dense sensor networks in which greedy forwarding has been proven to guarantee delivery [17]. However, we note that without the capability of circumventing voids (e.g., the absence of forwarding candidate nodes), IGF results in communication failure in sparse sensor networks. The stateless property of IGF precludes utilizing the perimeter-forwarding rule for planarized graphs, such as the one used in GPSR [11].

To improve the delivery performance under sparse sensor networks, we have designed and implemented a history-based forwarding area shift technique in IGF. This mechanism activates when a void is detected through a MAC layer notification of failure to IGF. The sender then retransmits the packet, requesting a shift of the forwarding area to *search* for an available receiver. The sequence of shifts is shown Figure 6. Those shifts allow IGF to utilize communication areas outside of the initial forwarding area. Since the area shifts allow backtracking, we must make sure that IGF is loop-free while maintaining the state-free property. Unfortunately, it has been proven in [18], a memoryless location-based routing algorithm is not loop-free

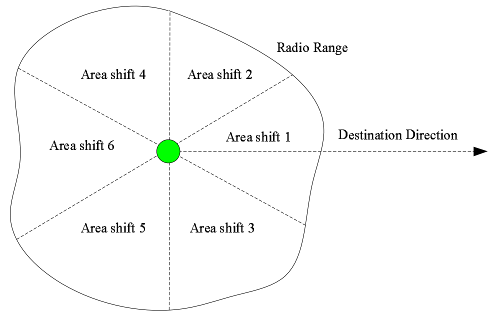


Fig. 6. IGF Handshake

if backtracking is allowed. To address this issue, IGF places a trace-history into the packet header to remember the nodes this packet visited recently, and no state is maintained in the nodes. To avoid the infinite loop, during the backtracking, a node choose the next hop forwarding node with an ID that is not in the trace-history. We note that this trace-history starts to accumulate only when the backtracking is activated, it does not incur overhead whenever greedy forwarding is possible. With this void avoidance capability, in the empirical study later shown in the evaluation, IGF is able to achieve a 100% delivery ratio with a small length of history added to a packet header.

H. Design Issues

This section completes our approach with several practical design issues.

1) Radio Irregularity: For the sake of clarity, IGF is described with a nominal symmetric radio range. However, IGF does work with asymmetric irregular range [14]. First, we enforce a symmetric channel by an ORTS-CTS-DATA-ACK handshaking procedure; second, though it is possible that an asymmetric channel among forwarding candidates still exists, which might introduce multiple CTS responses to a single ORTS, the sender can resolve the duplicate packets by simply choosing one and ignoring the others.

2) Localization Error Impact: IGF can be regarded as an extension to location-based protocols, whose performance can be affected by localization errors. Results from [19] show that the performance of the Geographic Forwarding (GF) protocol degrades as the localization error increases. In our evaluation section IV-E, we demonstrate our IGF scheme as well as GPSR achieves 100% delivery ratios in the presence of up to 50% radio range errors.

3) Energy Implications: The 802.11b standard allows a node to turn off the radio [20] after the node overhears a RTS packet that is not targeted to itself. However, IGF requires forwarding candidates to remain in the listening mode to overhear any CTS for about 5×10^{-5} seconds. This causes a slightly increase in the energy consumption. We note, however, that this increase is negligible when considering the higher delivery ratio, the reduction in control packets, and the smaller end-to-end delay we are able to achieve.

4) Alternative MAC Implementation: Without loss of generality, IGF is currently built and evaluated on the top of 802.11 DCF. Considering the bandwidth available in the mobile robotic sensor networks, it is a good solution in dealing the hidden and exposure terminal problems. However, we note that IGF is not bound to 802.11 DCF and it can be implemented with several existing MAC protocols. For example, IGF can be built on the MAC protocol suggested in [21] that uses the implicit ACK. In this scenario, forwarding candidates wait for a random delay before starting to relay a packet, and the one with the smallest delay forwards the data packet first. This data packet serves as both an acknowledgement to the sender and a cancellation signal to the rest of the forwarding candidates. (These handshaking sequences are shown in Figure 7). Based on [32], we have built the IGF protocol on the Berkeley mote platform and evaluated it with results shown in Section V.

5) Other Implications: Depending on the localization method used, IGF requires either additional energy [1] or more control messages [19] to localize the nodes, especially in the mobile environments. In addition, IGF doesn't fix the routes during the forwarding, which might lead to packet reordering due to the MAC contention. Consequently, the final receiver should ensure the data can be re-assembled correctly.

IV. EXPERIMENTS AND ANALYSIS

To assess the performance, we implement the IGF protocol in GloMoSim [22], a simulator for wireless sensor, ad hoc, and mobile networks. GloMoSim provides a high fidelity simulation for wireless communication with detailed modeling of communication propagation, radio and MAC layers. In addition, we also implement the IGF protocol on Berkeley mote platform (section V).

To make our evaluation close to the latest Telos mote capability proposed for use in the WSN environments [23], we set our system parameters as shown in Table I. We expect the typical communication patterns inside a sensor network to be established

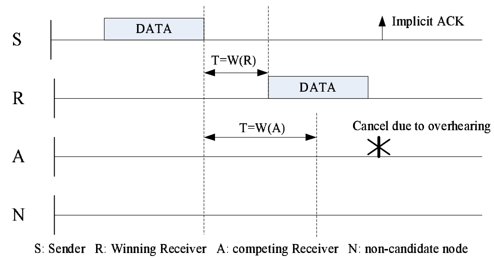


Fig. 7. IGF Handshake

TABLE I
SYSTEM PARAMETERS PARAMETERS

Parameters	Settings
Radio Range	$40.0m$
Terrain	$150X150m^2$
Collision Range	$71.2m$
Nodes	100 nodes,Uniform
Radio Range	$40.0m$
Bandwidth	$200kbps$
Radio	Lossy channel
Packet Size	32 byte Payload
W_P	W_P = 2
W_R	W_R = 1

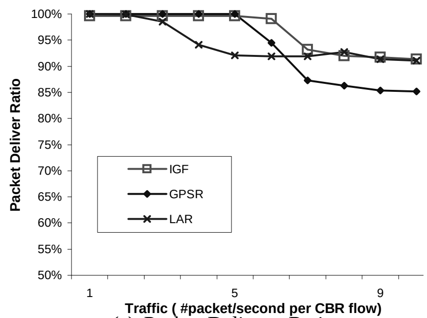
based on request and retrieval semantics for data delivery between sensor nodes and a querying entity. One-to-one, many-to-one and many-to-many communication patterns are representative workloads in sensor networks. One-to-one communication happens when one node detects some activity that needs to be reported to a remote entity. Alternatively, a querying entity will require periodic reports from the whole sensor area, which take the form of many-to-one communication. It is more common that multiple applications run simultaneously and the traffic flows interleave with each other, shown by the many-to-many cross-traffic pattern.

We evaluate 120 system configurations under different traffic loads, node mobility and energy conserving schedules. For each configuration we average 60 runs with different random seeds (hence 60 different network topologies and node placements) to ensure adequate confidence of our results. The 90% confidence interval is within 3% to 10% of the mean for GPSR and IGF, and 8% to 15% for LAR. Due to the space limitation, we only present more complex and interesting many-to-many scenario ($40 \times 60 = 2400$ runs). The complete data set is available upon request. In the many-to-many tests, 6 nodes, randomly chosen from the left side of the terrain, send 6 CBR flows to 2 nodes (3 each) on the right side of the terrain. The average hop count is about $4 \sim 6$ hops. We note that most well-known sensor network protocols such as Directed Diffusion [7], TTDD [24] and TBF [25] are mostly designed for static sensor networks and have never been evaluated in mobile environments. For the sake of fairness, we choose the only protocols that evaluate the mobility extensively in their publications ([8], [11], [12], [16]). Moreover, since IGF is a location-based routing protocol, it is unfair to compare IGF with other ID-based procotocol. As a result, we decide to compare IGF with two protocols: 1) LAR [16] is a protocol optimized for mobility using the location information and it is suitable for sensor network; and 2) GPSR [11] is the standard location-based sensor network protocol with greedy and planar perimeter forwarding rules. We consider these protocols in three scenarios:

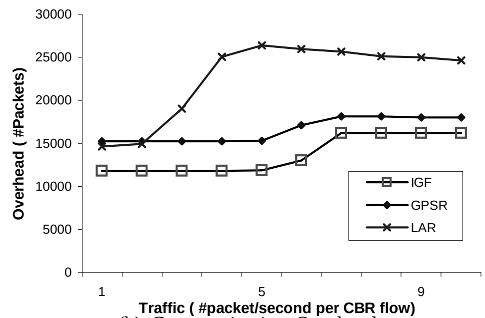
- A Static Network, where nodes are not mobile and energy conservation is not considered;
- A Mobile Network, with mobility ranging from walking to vehicular speeds;
- An Energy Conservation Network where nodes can transition into and out of dormant states.

For each experiment we choose three typical metrics on 1) the delivery ratio (the number of packets received / number of packets sent), 2) average end-to-end delay of received packets[1], and 3) overall communication overhead (total packets sent out by a node). In addition to the above experiments, we also evaluate the performance sensitivity of IGF in the presence of a low node density (voids), localization errors and location update delay in Section III-G, Section IV-E and Section IV-F, respectively.

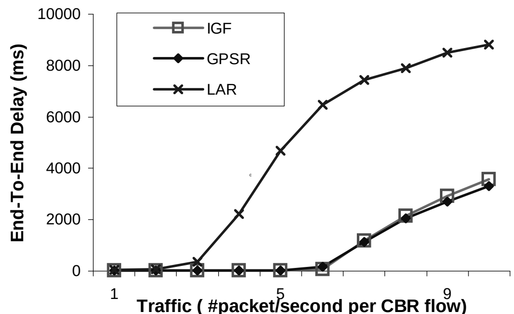
[1]We note that our evaluation does not choose deadline miss ratios as the major metrics, because such an approach reveals less information about the tradeoff between actual delays and other system performance parameters



(a) Packet Delivery Ratio



(b) Communication Overhead



(c) E2E Communication Delay

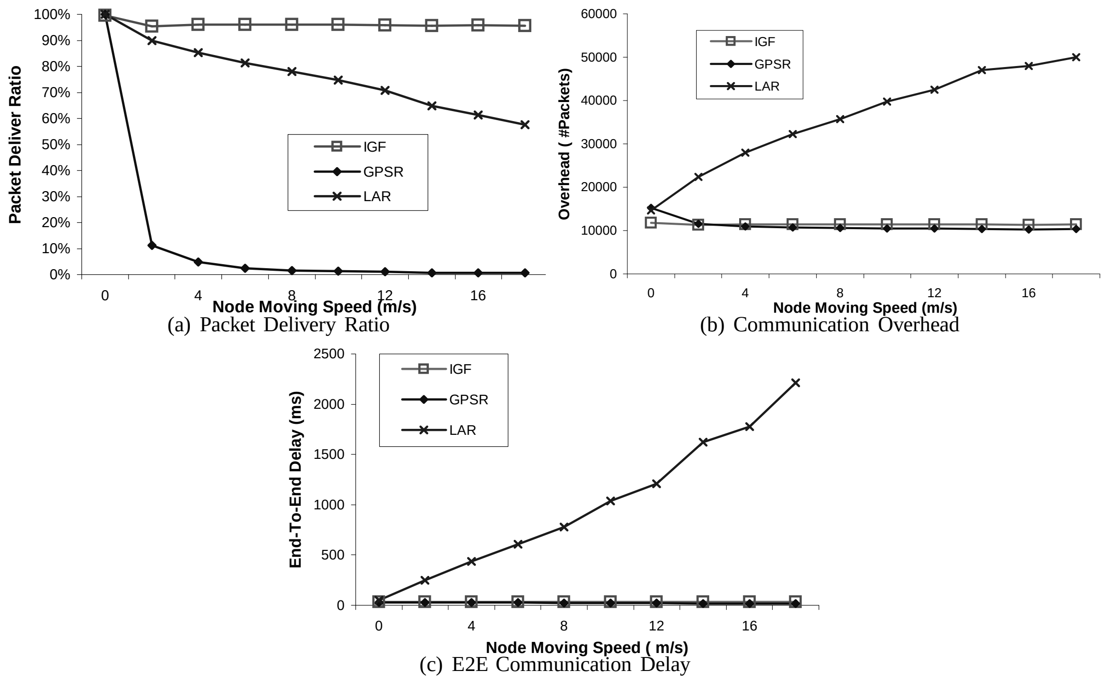
Fig. 8. Performance in Static Networks

A. Performance in Static Networks

The evaluation in static networks shows that IGF performs as well as or slightly better than GPSR and LAR, when dynamics such as mobility and energy conserving sleep cycles are not considered. In these experiments, we increase many-to-many CBR flow rate until sufficient congestion is seen. Figure 8a shows that GPSR and IGF have comparable delivery ratios under light loads, while LAR loses packets early as these protocols quickly congest the network by sending route discovery packets. When the traffic flow rates increase enough to adequately congest the network in GPSR and IGF (6+ packets/second per CBR flow), performance in GPSR degrades due to limited intersecting routes, suffering additional collision caused by neighbor table update beacons (0.1 beacon per second). LAR uses location information to keep the effects of route discovery to a minimum allowing it to maintain delivery ratios comparable to IGF. However, LAR's frequent transmission of route discovery packets toward the destination, coupled with the latency incurred awaiting the route discovery response, lead to significantly more overhead (Figure 8b) and longer delay (Figure 8c) when compared with IGF. Figure 8b demonstrates IGF's savings at low traffic loads, as IGF does not require beaconing (GPSR) or route discovery packets (LAR) required in these protocols. As traffic loads increase, congestion increases the number of MAC layer collisions in both IGF and GPSR, resulting in retransmission attempts that add to overhead as shown in Figure 8b. In GPSR and IGF we see significantly lower end-to-end delay beyond 4 packets/second per CBR flow because LAR suffers latency awaiting the return of route discovery packets. Finally, under heavy traffic, we see a slightly longer delay in IGF over GPSR due to the fact that IGF manages to deliver 10% more packets (Figure 8a). We also note that Figure 8c demonstrates that the CTS back-off delay due to lazy binding in IGF has virtually no impact on the end-to-end delay.

B. Performance under Mobility

One scenario for the evaluation under high mobility would be a group of exploring robotic sensor nodes, trying to find the survivors underneath rubble after an earthquake. They periodically update their locations to each other while searching. Another scenario would be a group of mobile robots equipped with magnetic sensors, searching for mines in a battlefield. These robots report the detections to a base station by relaying packets among themselves. As we mentioned before, nodes' locations can be obtained through GPS in such a highly dynamic systems. We choose a standard waypoint mobility model during the simulations. It should be noted that in contrast to ad hoc networks where the mobility pattern is interleaved with burst movements and long pauses, sensor robots are normally continuously moving. To reflect this scenario, we set only a 1-second pause intervals between moves ($100 \sim 1000s$ pause intervals are normal settings in ad hoc network evaluations [1]). The settings stress-test the protocols' ability to deal with continuously high mobility and reflect the mobility patterns seen in mobile sensor networks. We model speeds up to 18 meters per second ($\sim 40mph$) to evaluate a wide range of mobile scenarios in which sensors can be attached to slow robots or to high-speed vehicles. We adopt a 40m range to confirm to current sensor ability, which is much smaller than 250m setting in used in WLAN ([1] and [15]). We note that the mobility is characterized by the number of neighborhood changes per second, which is affected by both the node speed and the radio range. With the same speed, a smaller range leads to a high mobility. To validate this point, in addition to this section, we investigate the impact of the radio ranges on the routing performance under mobility in Section IV-B.1.



(a) Packet Delivery Ratio

(b) Communication Overhead

(c) E2E Communication Delay

Fig. 9. Performance in Mobile Networks

In the static network scenario (section IV-A), we use a low beacon rate (0.1 beacon per second) in GPSR to reduce the effect of congestion. To optimize GPSR to deal with mobility, we test GPSR with multiple beacon rates. Because beacons consume bandwidth, their cost offsets their savings, arriving at similar results in all beacon rates we tested. Consequently, we adopt 1 beacon per second to keep the state as fresh as possible without causing congestion. Rerouting is supported when a protocol experience a link break. From Figure 9a, we see that when nodes do not move (0 m/s), no packets are lost and the lowest delay and overhead are incurred due to minimal congestion. As we introduce mobility, increasingly affecting the validity of neighborhood and routing state with increased node speeds, we see the delivery ratios (Figure 9a) in GPSR and LAR drop off quickly while IGF continues to perform close to optimal. For example, when the node moving speed is at 4 meters/second, IGF demonstrates as much as 10 times performance gain in the delivery ratio over GPSR. For LAR, performance degrades as node migration invalidates eager-binding routes. Since LAR is specially designed to deal with mobility, its milder degradation, as seen in Figure 9a, results from location controlled flooding of route discovery packets to reestablish routes despite mobility. As an addendum to explain why GPSR performs so poorly, we note from Figure 9a that GPSR's delivery ratio quickly drops to zero at relatively low node speeds. One might assume this is because the beacon overhead leads to congestion in GPSR, hence a very low delivery ratio. However, from Figure 9b, we note that control overhead in GPSR is actually smaller than LAR. In fact, this low delivery ratio in GPSR happens because according to greedy forwarding rules in GPSR, the chosen next-hop node is normally located at the edge of the sender's communication radius. Because nodes are equally likely to move in any direction, there is a high chance that designated receiver will have moved out of communication range from the sender since the last beacon which was received seconds ago. Over multi-hop routes, the chances of failure grow exponentially. In contrast, IGF binds the next hop tens of microseconds before packet forwarding occurs. This significantly reduces the chance that a chosen node will move out of communication range during this tiny interval. Aside from the delivery ratio (Figure 9a), our evaluation shows that IGF significantly outperforms other protocols in metrics of overhead (Figure 9b) and end-to-end delay (Figure 9c) under all moving speeds. All these results are due to IGF's ability to defer the mapping between routing states and network topologies until this binding is absolutely required.

1) Radio range impact on the routing performance under mobility: In this experiment, we investigate the impact of different radio ranges on the routing performance in mobile sensor networks. When nodes move around, mobility breaks old links and establishes new links. With the same node speed, a smaller radio range causes a higher rate of change in the network topology. Figure 10 proves that GPSR is able to achieve good deliver ratios with a large radio range, which leads to a smaller mobility.

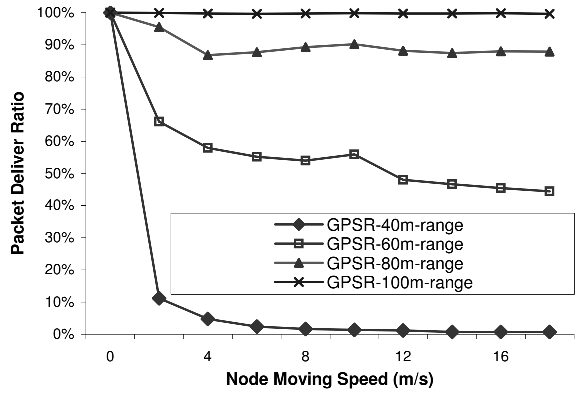
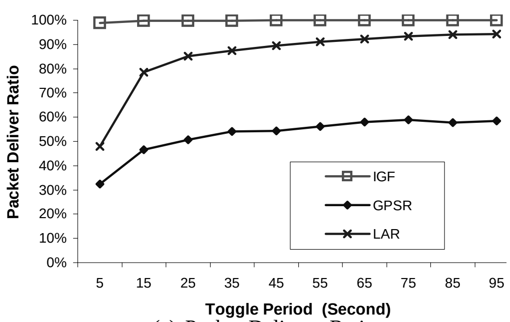


Fig. 10. Delivery Ratio under Different Radio Ranges and Speeds

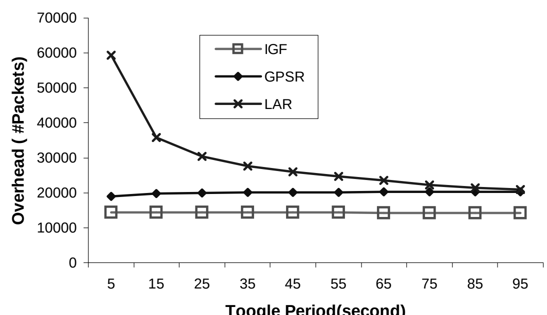
On the other hand, Figure 10 indicates GPSR's delivery performance reduces dramatically under high mobility situations.

C. Performance under Energy Conservation

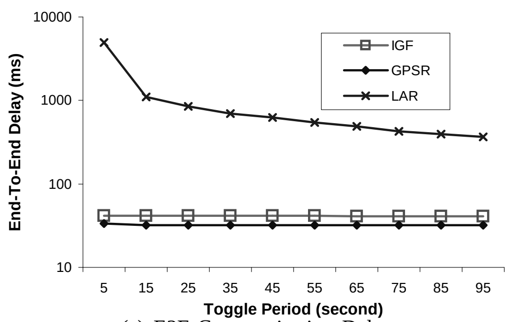
It is crucial for sensor network systems to support energy conservation. The most practical way to reduce total energy consumption is to turn on/off the nodes on demand of events [4]–[6]. However, these operations disrupt the network topologies. In this experiment, we test IGF, GPSR and LAR in the presence of orthogonal energy conserving protocols by periodically transiting nodes into and out of sleep states. To prevent congestion, and therefore isolate the effects of the awake-sleep transition in our analysis, we set the flow rate to 1 packet per second. We note that two key parameters in energy-conserving protocols can affect the routing performance:



(a) Packet Delivery Ratio



(b) Communication Overhead



(c) E2E Communication Delay

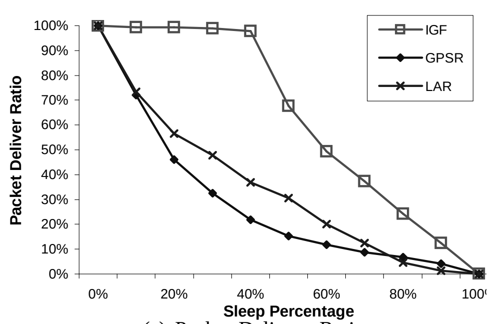
Fig. 11. Performance under Varied Toggle Periods

- **Toggle Period:** Toggle Period is the time interval between consecutive transitions into a sleep state. This parameter reflects how fast a routing state is invalidated due to sleep-awake transitions. We change this value from 5 seconds to 95 seconds in increments of 10.
- **Sleep Percentages:** The percentage of time a node is in the sleep mode. We note that sleeping can significantly affect the active node density, as this reduces the number of nodes participating in routing at any point in time.

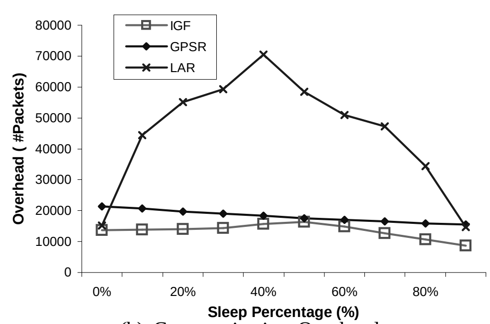
1) Performance under Varied Toggle Periods: Figure 11a shows the results for many-to-many flows where the Sleep Percentage is set at 30% for varying Toggle Periods. It shows that IGF outperforms all other protocols at all toggle periods investigated. GPSR utilizes a beaconing mechanism to proactively bind network topologies into neighbor states. This binding can be quickly invalidated due to nodes' awake-sleep transitions. As a result, packets may be forwarded to nodes that were turned off since the last beacon and then dropped by the MAC layer. This leads to a poor delivery ratio in GPSR (Figure 11a). In LAR, a node requires the network layer to handle transmission failures by initiating route discovery. Due to the on-demand nature of those algorithms, LAR outperform GPSR, as the recently returned route discovery packet traverses nodes that are currently awake and therefore able to act as routers. Finally, we see IGF performing significantly better than other protocols, at times showing more than 3 times improvement in packets delivered when compared to GPSR. We attribute this performance to the IGF's ability to utilize whatever neighbors are currently awake en route to the destination. We note the Toggle Periods here only range from 5 to 95 seconds. When the Toggle Periods increase further, less dynamics are introduced into the network topologies and routing states can remain fresh for a longer period of time. In this scenario, higher delivery ratios are expected for other algorithms. Theoretically, when the Toggle Period approaches infinity, energy conserving networks become traditional static networks, for which we have shown performance comparisons in section IV-A.

2) Performance under Varied Sleep Percentage: We next assess routing performance varying Sleep Percentage for the highly volatile case where the Toggle Period is set to 5 seconds. This not only allows us to compare our work under varied Sleep Percentage times, but allows us to stress test our protocol under highly dynamic system settings. In this experiment, we increase the sleep percentage of each node from 0% (always awake) to 100% (always asleep) in increments of 10%.

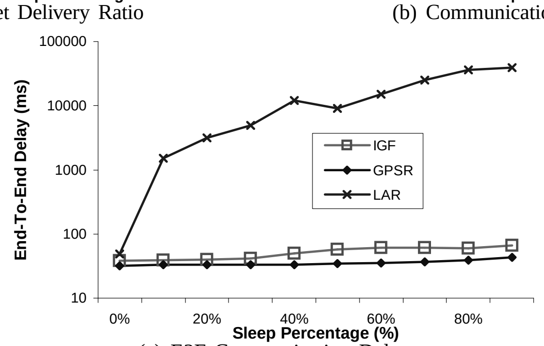
Figures 12a, b and c all demonstrate IGF's better performance over varied Sleep Percentages. Figure 12a shows that IGF delivers the highest percentage of packets under all Sleep Percentage settings, while incurring the small end-to-end delay (Figure 12c) and the lowest transmission overhead (Figure 12b). For example, Figure 12a shows that at a 50% sleep percentage, IGF delivers 340% more packets than the GPSR protocol. The drastic drop in overhead (Figure 12b) as seen in LAR also can also be attributed to this drop in the Packet Delivery Ratio. Since LAR is designed to adapt to occasional node failures, as we expect, in such highly dynamic networks, it takes a huge end-to-end delay to repeatedly fix these routes (Figure 12c). GPSR shows the lowest end-to-end delay (Figure 12c) because it delivers a tiny percentage of packets when compared to the IGF. Those packets go through the networks quickly by chance. Only IGF has a highest delivery ratio and a small delay. This is due to the fact that IGF can immediately detect node transitions into sleep states and immediately utilize recently awoken nodes.



(a) Packet Delivery Ratio



(b) Communication Overhead



(c) E2E Communication Delay

Fig. 12. Performance under Varied Sleep Percentage

D. Performance in Sparse Sensor Networks

The typical density of sensing-covered sensor network systems [4] is about $20 \sim 25$ nodes/radio range in order to provide high fidelity in localization, detection and tracking. In previous evaluations, we use 22 node/radio ranges as a typical setting. However, it is important to understand how IGF performs under various node density settings. To prevent congestion, and therefore isolate the effects of density in our analysis, we set the per node flow rate to 1 packet per second. To change the density of the network, instead of increasing the number of nodes in the terrain, we keep the number of nodes constant at 100, and increase the side length of the square terrain from 100 meters to 250 meters in increments of 10. Figure 13 shows that with the history-based forwarding- area shifts, IGF achieves a 100% delivery ratio when the node density is larger than 12 nodes per nominal radio range. Figure 13 reveals that when density if relatively high (≥ 9 node/radio range), longer trace-history does not help much, however when the network become sparse, longer history can improve the delivery ratio.

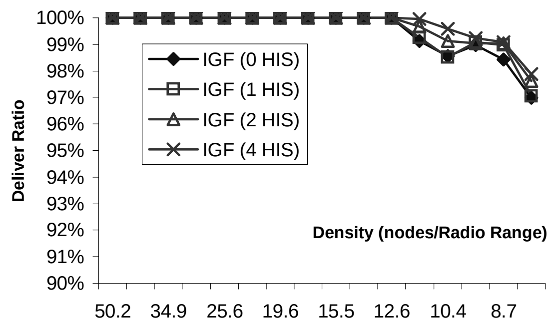


Fig. 13. Delivery Ratio Vs. Node Density

E. Performance under Localization Errors

While most work in location-based routing assumes perfect location information, the fact is that erroneous location estimates are virtually impossible to avoid. In this experiment, we investigate location error impact on the IGF protocol. To prevent congestion, and therefore isolate the effects of the localization error, the traffic loads are set to the rate of 1 packet/second. We compare IGF, GPSR with the basic geographic forwarding(GF) [26], which forwards a packet to the node that makes the most progress toward the destination. We increase the localization error from 0% to 50% of the radio range in increments of 5% to measure the end-to-end delivery ratios. Figure 14 demonstrates that both the IGF and GPSR protocol perform much better in the presence of localization errors while the GF protocol suffers as location errors increase.

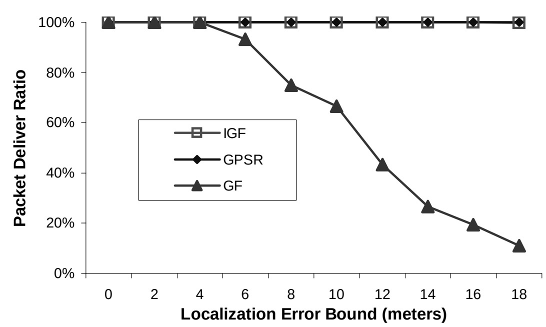


Fig. 14. Localization Error Impact

F. Performance under Different Localization Update Intervals

IGF obtains location updates from GPS or other localization schemes. Since the update rate affects the amount energy consumed to obtain the locations, the location is normally updated intermittently. Consequently, nodes have to make the routing decisions based on the last localization result, which might cause the routing failures if the the update delay is too long. In this section, we investigate the impact of the location update delay to the end-to-end delivery ratio. Figure 15 shows that the location update delay doesn't affect the static and energy conservation networks since nodes don't move in such networks. As for the mobile networks, a moderate location update delay (e.g., $\leq 1 second$) doesn't noticeably affect the delivery ratio, however, a large delay cause more routing failures. Figure 15 also indicates that the impact of the update interval is affected by nodes' speed. With the same update intervals, a faster node speed leads to a lower delivery ratio.

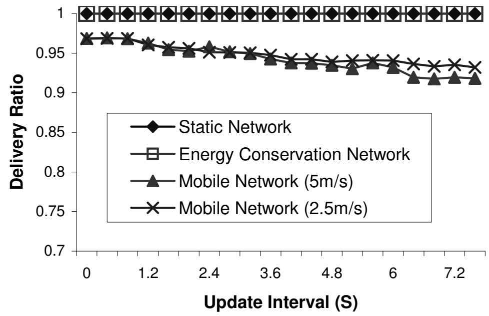


Fig. 15. Impact of Location Update Delay

V. IMPLEMENTATION ON MOTES

We have implemented the IGF protocol on the Berkeley motes platform [27] with a code size of 11,606 bytes (code is available through CVS at [28]). Currently, this implementation is built on the top of a MAC protocol with the implicit ACKs mentioned in section III-H.4. Three applications including data placement, target tracking and CBR data streaming are also built to run on top of IGF. Due to physical constraints and the un-availability of state-of-the-art protocols on such a platform, it is difficult to perform as extensive evaluation as we did in the wireless simulator. We, therefore, only present initial results here as a study for developing a more complete solution and evaluation in the future mote platform. As we mentioned in section III-B, IGF does not task a specific node to route packets a priori. This feature is beneficial for load balancing among the nodes inside the forwarding area. In this experiment, we use 25 motes to form a 5 by 5 grid. To evaluate the load balancing capability of IGF we send a CBR data stream from node 24 to node 0, which is the base station. We collect the number of packets relayed by intermediate motes ($1 \sim 23$) and compare this with the result obtained from the GF protocol which we also implemented on the motes. While both GF and IGF achieve nearly 100% delivery ratio, GF tends to relay packets via a fixed route which might lead to unbalanced traffic. This is shown in Figure 16 as node 19 relays 250 packets while node 18 doesn't forward any packets. Instead, by distributing traffic loads, IGF effectively balances energy consumption. We argue that in sensor networks, balanced energy consumption can prevent some nodes from dying faster than others, therefore increasing the network lifetime.

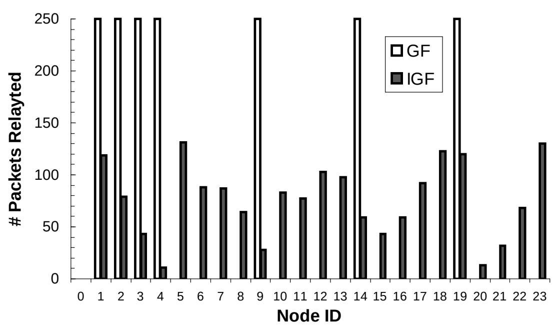


Fig. 16. Traffic Balance

VI. RELATED WORK

In this section we discuss prior research in distributed computing that is pertinent to the design of IGF. Various protocols [29]–[34] have been introduced to reduce packet loss through reliable communication in sensor networks. Alec Woo [33] chooses reliable routes based on link connectivity statistics obtained dynamically from a EWMA estimator. RMST [30] tracks packet fragments so that receiver initiated requests can be satisfied when individual pieces of an application payload get lost. PSFQ [31] caches packets along the path to the sender, initiating fragment recovery as required, starting with its local neighborhood. Robust data delivery [29] simultaneously sends packets along multiple paths at the expense of increases in communication overhead. While these ARQ/FEC-based solutions have proven effective when dealing with interference and collisions, their robust and reliable features might not be able to handle failures due to high dynamics in network topologies. We consider them orthogonal and complementary to our work.

Many routing algorithms have been proposed for ad hoc and sensor networks. With regard to the mechanisms used to bind network topology to the routing state, we divide these routing protocols into three categories. The first category we term as proactive eager-binding routing algorithms. DSDV [10] requires each node to proactively broadcast routing updates periodically. Global routing tables are refreshed regardless whether there is need for data delivery. Location-based routing algorithms such as GPSR [11] remove the requirement that a protocol maintains a global view of the network (i.e. end-to-end routing tables), therefore reduces communication overhead by eliminating its dependence on the network wide state information. However, they still depend on up-to-date local neighborhood tables, requiring control overhead to maintain such tables and suffering latency and packet loss when a node's neighborhood state changes between updates. To minimize unnecessary overhead incurred by proactive updates, a set of on-demand algorithms are proposed to defer route acquisition until data delivery is required. We term the second category as reactive eager-binding algorithms. It has been proved in [12] that AODV [9] and DSR [8] can successfully deal with moderate mobility with long pause intervals ($100 \sim 1000$ seconds). However, the eager binding of the routing states at the route acquisition phase make them less effective to deal with high dynamics in which network topologies change at a much faster rate than the duration of connections. Routing maintenance and rediscovery are proposed in [8] [9] to remedy this situation partially at the cost of higher delay and expensive control overhead. LAR [16] extends the on-demand idea proposed by AODV [9] and DSR [8], utilizing location information to limit the scope of route requests. While LAR significantly reduces routing overhead by only propagating queries to relevant portions of the network, it still needs to maintain or establish an explicit path before transmitting a packet. Current reactive eager-binding algorithms can successfully deal with occasional node failures and moderate mobility. However, the elevated dynamics due to the continuous mobility and power conservation inside sensor networks challenge researchers to develop a new category of routing protocols based on the lazy binding concept.

The first state-free protocol IGF belongs to this third category. ExOR [35] also decides the forwarding candidate on the fly. However, before transmitting a packet, the sender needs to specify the forwarding candidates in the packet header, which requires maintaining the state information about neighboring nodes. GeRaF [36] proposes a similar packet forwarding technique and it focuses on the multi-hop performance in terms of the average number of hops to reach a destination. Both ExOR and GeRaF do not model the effect of channel contention; while this work provides a detailed implementation and evaluation through both simulation and a running system.

VII. FUTURE WORK

In this work, IGF assumes a localization service or the GPS capability. This is justified as sensor network applications require location information to make sensor data meaningful. We note that lazy-binding is a general concept to deal with high network dynamics and its applicability does not intrinsically depend on the location service. It is promising to apply lazy-binding to ID-Based protocols such as Directed Diffusion [7]. To extend [7], we can keep the hop-count-to-a-sink as a non-volatile state with respect to the node failures, and we perform forwarding operations with the parents of each node. We note that in this ID-based case, the state-free property is not maintained, however, lazy-binding, which is independent of the state-free property, is still beneficial in dealing with the failure of the parent nodes. Due to the space constraints, we leave this as future work.

VIII. CONCLUSIONS

In highly dynamic sensor networks, the maintenance of freshness of routing states is costly. The state update, resulting from eager-binding, directly contributes to network congestion, wasting precious energy and increasing the end-to-end transmission latency. To prevent the adverse affects that dynamic factors such as high mobility have on the state-based eager-binding routing protocols, we advocate using the concept of lazy-binding to cope with high dynamics in sensor networks. Based on this concept, we introduce IGF, a unicast protocol that is altogether state-free. In simulation, we compare our work against protocols designed for mobile environments and sensor networks. IGF demonstrates more than 10 times improvement in the packet delivery ratio when the sensor network is highly mobile. IGF also achieves significant reduction in delay and overhead when considering mobility and energy-conservation. In addition, the IGF protocol has been implemented on the Berkeley motes platform to serve as an initial study in developing a more complete solution in the future.

ACKNOWLEDGMENTS

This work was supported in part by NSF grant CCR-0098269 and NSF grant CSR-0615063.

REFERENCES

[1] B. H. Wellenhoff, H. Lichtenegger, and J. Collins, *Global Positions System: Theory and Practice,Fourth Edition*. Springer Verlag, 1997.
[2] J. Borenstein, H. Everett, and L. Feng, *Navigating Mobile Robots: Systems and Techniques*. A. K. Peters, Ltd., Wellesley, 1996.
[3] W. Burgard, M. Moors, D. Fox, R. Simmons, and S. Thrun, "Collaborative Multi-Robot Exploration," in *In Proceeding of IEEE International Conferenceon Robotics and Automation (ICRA)*, 2000.
[4] T. He, S. Krishnamurthy, J. A. Stankovic, and T. Abdelzaher, "An Energy-Efficient Surveillance System Using Wireless Sensor Networks," in *The Second International Conference on Mobile Systems, Applications, and Services (MobiSys)*, June 2004.
[5] T. Yan, T. He, and J. Stankovic, "Differentiated Surveillance Service for Sensor Networks," in *First ACM Conference on Embedded Networked Sensor Systems (SenSys 2003)*, November 2003.
[6] F. Ye, G. Zhong, S. Lu, and L. Zhang, "PEAS: A Robust Energy Conserving Protocol for Long-lived Sensor Networks," in *Proc. of International Conference on Distributed Computing Systems (ICDCS)*, May 2003.
[7] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks," in *the Sixth Annual International Conference on Mobile Computing and Networks*, 2000.

[8] J. Broch, D. B. Johnson, and D. A. Maltz, *DSR : The dynamic source routing protocol for multihop wireless ad hoc networks*. Ad Hoc Networks, Addison Wesley, 2001.
[9] C. E. Perkins and E. M. Royer, "Ad-hoc on demand distance vector routing," in *WMCSA*, 1999.
[10] C. E. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers," in *SIGCOMM*, 1994.
[11] B. Karp and H. T. Kung, "Greedy perimeter stateless routing for wireless networks," in *International Conference on Mobile Computing and Networking*, 2000.
[12] J. Broch, D. A. Maltz, D. B. Johnson, Y.-C. Hu, and J. Jetcheva, "Performance comparison of multi-hop wireless ad hoc network routing protocols," in *the Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking*, 1998.
[13] I. . L. S. Committee, "Ansi/ieee, ieee 802.11 wireless local area networks," http://grouper.ieee.org/groups/802/11/.
[14] G. Zhou, T. He, and J. A. Stankovic, "Impact of Radio Irregularity on Wireless Sensor Networks," in *The Second International Conference on Mobile Systems, Applications, and Services (MobiSys)*, June 2004.
[15] "Generalized birthday problem," http://www.mathpages.com/home/kmath057.htm.
[16] Y. Ko and N. H. Vaidya, "Location-aided routing (lar) in mobile ad hoc networks," in *International Conference on Mobile Computing and Networking*, 1998.
[17] G. Xing, C. Lu, R. Pless, and Q.Huang, "On greedy geographic routing algorithms in sensing-covered networks," in *MobiHoc'04*, 2004.
[18] I. Stojmenovic and X. Lin, "Gedir: Loop-free location based routing in wireless networks," in *Int. Conf. on Parallel and Distributed Computing and Systems*, 1999.
[19] T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. Abdelzaher, "Range-Free Localization Schemes in Large-Scale Sensor Networks," in *Proc. of the Intl. Conference on Mobile Computing and Networking (MOBICOM)*, September 2003.
[20] W. Ye, J. Heidemann, and D. Estrin, "An energy-efficient mac protocol for wireless sensor networks," in *INFOCOM*, 2002.
[21] A. Woo and D. Culler, "A Transmission Control Scheme for Media Access in Sensor Networks," in *Proc. of Mobile Computing and Networking (Mobicom)*, 2001.
[22] X. Zeng, R. Bagrodia, and M. Gerla, "Glomosim: a library for parallel simulation of large-scale wireless networks," in *the 12th Workshop on Parallel and Distributed Simulations*, 1998.
[23] *Telos Mote IEEE 802.15.4 Compliant Mote Data Sheet*, CrossBow, http://www.moteiv.com/products/telos-brochure.pdf.
[24] F. Ye, H. Luo, J. Cheng, S. Lu, and L. Zhang, "A Two-tier Data Dissemination Model for Large-scale Wireless Sensor Networks," in *7th ACM MOBICOM Conference*, 2002.
[25] D. Niculescu and B. Nath, "Trajectory Based Forwarding and its Applications," in *MobiCom*, 2003.
[26] G. G. Finn, "Routing and addressing problems in large metropolitan-scale internetworks," USC/ISI, Tech. Rep. ISI/RR-87-180, 1987.
[27] *Mica2 data sheet*, CrossBow, 2003, available at http://www.xbow.com.
[28] T. He, L. Gu, B.Blum, and J. Xie, "Nest project source code," http://sourceforge.net/projects/vert/.
[29] D. Ganesan, R. Govindan, S. Shenker, and D. Estrin, "Highly resilient, energy efficient multipath routing in wireless sensor networks," *Mobile Computing and Communications Review*, vol. 1, no. 2, 2002.
[30] F. Stann and J. Heidemann, "RMST: Reliable Data Transport in Sensor Networks," in *SNPA '03*, 2003.
[31] C. Y. Wan, A. T. Campbell, and L. Krishnamurthy, "PSFQ: A Reliable Transport Protocol for Wireless Sensor Networks," in *WSNA '02*, 2002.
[32] C. Wan, S. Eisenman, and A. T. Campbell, "CODA: COngestion Detection and Avoidance in Sensor Networks," in *Sensys 03*, 2003.
[33] A. Woo, T. Tong, and D. Culler, "Taming the Underlying Challenges of Reliable Multihop Routing in Sensor Networks," in *First ACM Conference on Embedded Networked Sensor Systems (SenSys 2003)*, November 2003.
[34] J. Zhao and R. Govindan, "Understanding Packet Delivery Performance In Dense Wireless Sensor Networks," in *Sensys 03*, 2003.
[35] S. Biswas and R. Morris, "Exor: opportunistic multi-hop routing for wireless networks," *SIGCOMM Comput. Commun. Rev.*, vol. 35, no. 4, pp. 133–144, 2005.
[36] M. Zorzi and R. R. Rao, "Geographic Random Fowarding (GeRaF) for ad hoc and sensor networks: Multihop performance," *IEEE Transaction on Mobile Computing*, vol. 2, no. 4, 2003.