Weak State Routing for Large Scale Dynamic Networks

Utku Günay Acer
Shivkumar Kalyanaraman
Alhussein A. Abouzeid

Department of Electrical, Computer and Systems Engineering
Rensselaer Polytechnic Institute
Troy, NY, USA
aceru@rpi.edu, {shivkuma,abouzeid}@ecse.rpi.edu

ABSTRACT

Routing in communication networks involves the *indirection* from a *persistent name (or ID)* to a *locator* and delivering packets based upon the locator. In a large-scale, highly dynamic network, the ID-to-locator mappings are both large in number, and change often. Traditional routing protocols require high overhead to keep these indirections up-to-date. In this paper, we propose Weak State Routing (WSR), a routing mechanism for large-scale highly dynamic networks. WSR's novelty is that it uses random directional walks biased occasionally by weak indirection state information in intermediate nodes. The indirection state information is *weak*, i.e. interpreted not as absolute truth, but as *probabilistic hints*. Nodes only have partial information about the region a destination node is likely to be. This method allows us to *aggregate* information about a number of remote locations in a geographic region. In other words, the state information maps a *set-of-IDs* to a *geographical region*. The intermediate nodes receiving the random walk use a method similar to longest-prefix-match in order to prioritize their mappings to decide how to bias and forward the random walk. WSR can also be viewed as an unstructured distributed hashing technique. WSR displays good rare-object recall with scalability properties similar to structured DHTs, albeit with more tolerance to dynamism and without constraining the degree distribution of the underlying network.

Through simulations, we show that WSR offers a high packet delivery ratio, more than 98%. The control packet overhead incurred in the network scales as $O(N)$ for N-node networks. The number of mappings stored in the network appears to scale as $\Theta(N^{3/2})$. We compare WSR with Dynamic Source Routing (DSR) and geographic forwarding (GPSR) combined with Grid Location Service (GLS). Our results indicate that WSR delivers more packets with less overhead at the cost of increased path length.

Categories and Subject Descriptors: C.2.2 [Computer-Communication Networks]: Network Protocols – *Routing protocols*

General Terms: Algorithms, Design, Performance

Keywords: Dynamic Networks, Routing Algorithms, Unstructured Distributed Hashing, Weak States

MobiCom'07, September 9–14, 2007, Montréal, Québec, Canada.

1. INTRODUCTION

Routing in communication networks involves the *indirection* from a *persistent name (or ID)* to a *locator*. The primary goal of a routing protocol is to gather information, which gives directions on performing these indirections and how to reach a node with a specific ID. In a dynamic network where the network topology and the neighborhoods of nodes continuously change, the information the nodes use to make routing decisions should also change and remain up-to-date for resilient routing.

One approach in routing is to use the geographical location information where forwarding is achieved by leveraging Cartesian properties like distance and direction as in Greedy Perimeter Stateless Routing (GPSR) protocol [15]. In dynamic networks, such a mechanism may be favored over link state protocols because the entropy of the link state is usually higher than the node location entropy. Still, ID-to-location indirection is a problem and nodes require an efficient location service which tracks down location changes in the network through receiving location updates from all the nodes. Updating location information in remote location servers causes overhead and deteriorates the capacity of the network [3].

In this paper, we propose a routing algorithm for dynamic networks, the *Weak State Routing (WSR) Protocol*, which uses the prior location information for the nodes. We evaluate the weak states in the context of connected highly mobile ad-hoc networks. In our approach, both the locator and ID-to-location mapping are dynamic. Unlike traditional routing table state, the states we use are *weak*, i.e. are interpreted not as absolute truth, but *probabilistic hints*. In a mobile network, a deterministic state that has strong semantics is rapidly invalidated due to dynamism. The nodes require control traffic to refresh strong states. On the other hand, weak state has probabilistic semantics and it is more stable. Weak state can be updated locally without requiring control traffic. Weak state does not require hard limit on expiration in contrast to traditional strong routing states. Hence, weak state is a generalization of *soft* state. Weakening a soft state ages it and is equivalent to a soft *timeout*. Once the strength of the semantics is below a threshold value, the information the state provides becomes insignificant and the state is effectively timed out.

Using the state information, a node ID is mapped to a geographical region, or GeoRegion, where the node is likely to be located at present instead of a point in the plane. We use unstructured random directional walks to route data packets. This is different from a classical random walk that is forwarded a neighbor with equal probabilities. Instead, the source node randomly picks a direction and sends the random walk in that direction. These random directional walks are occasionally biased using the weak indirection state information at intermediate nodes. The biasing points are similar to *rendezvous* concepts in earlier work (eg: ORRP, [7]), except

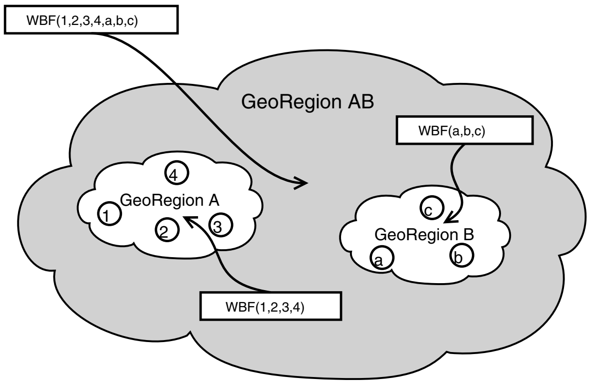

Figure 1: Weak State Concept: Indirection aggregates: Weak Bloom Filter (WBF) mapped to aggregated geographic regions

that WSR may require multiple biases to get the random walk to the destination. Note that we propose weak states as a solution to the routing problem for dynamic networks; the scope of weak states is not limited to mobile ad-hoc networks. They can also be deployed in scenarios where the connectivity is intermittent such as Delay Tolerant Networks (DTN). In such scenarios, link state is arguably less useful since the network is disconnected most of the time, which is the main reason we use geographic information in our mechanism.

In WSR, we aggregate the ID-to-location mappings in an unstructured manner. In other words, we have mappings for *SetofIDs-to-GeoRegion* that is represented by a weak Bloom filter (WBF), instead of individual *ID-to-location* mappings. If two mappings are close in terms of their GeoRegion portions, we can combine these mappings. We refer to these aggregate mappings as *indirection aggregates*. These indirection aggregates naturally capture the uncertainty in the location of a remote node as we move from GeoLocation to GeoRegion. In particular, more specific indirections exist for nodes in nearby locations, and walks are progressively guided with greater precision. (see Fig. 1).

The weak state routing concept is an unstructured and distributed hashing technique. We note that point-to-point routing is equivalent to rare object lookup in P2P networks [19]. WSR provides distributed hashing functionality without a structured overlay as in traditional Distributed Hash Tables (DHT). Our distributed hashing and aggregation mechanisms are also similar to subnetting mechanism in IP, where the aggregation is structured and based on common prefixes. We make the forwarding decisions using a method similar to "longest-prefix-match". In our case, we use the *strongest semantics match* to prioritize the indirections in order to decide how to bias the random walk. We look at the mappings which render the strongest signal on the destination node's membership. Instead of forcing a network structure (i.e. hierarchy, IDs in a subnet have same prefix etc), we assume a flat network and provide an unstructured aggregation mechanism for scalability.

WSR routes packets with scalable overhead and storage. While WSR routes packets to the destination nodes with high probability without using absolute state information, the paths taken by the packets may be longer, in terms of hops and distance, than the shortest paths. The total routing protocol overhead increases linearly with the number of nodes in the network regardless of the node mobility level. WSR offers a scalable indirection method by aggregating weak states. Empirically, we see that the number of mappings in the entire network scales as $\Theta(N^{3/2})$, N being the number of nodes. WSR enjoys these benefits at the cost of increased path length. Alternatively, we can use a random walk at the beginning of a flow in order to locate the destination node. Once the location of the destination is found, we can use geographical forwarding to deliver the data packets. Since we consider weak states as a mechanism that can also be deployed in DTNs, we use opportunistic forwarding in this paper. Our simulations show that WSR outperforms flooding based routing schemes and geographical routing schemes that use a location service in terms of overhead and delivery ratio. While the packets routed by WSR follow longer paths than what a location service provides, the average end-to-end delay performance is better.

The rest of this paper is organized as follows: In Section 2, we overview the related work. We define our weak state concept and present our routing mechanism in Section 3. In Section 4, we provide the performance analysis through simulations. We discuss how weak states can be extended to dynamic networks other than ad-hoc networks and present our plans for future work in Section 5. We conclude the paper in Section 6.

2. RELATED WORK

Routing protocols are classified into four major types: reactive, proactive, position-based, and hybrids of these approaches.

Reactive protocols like Dynamic Source Routing (DSR) [13] flood the network to perform route discovery when a path between a source node and a destination node is needed. Data is not transmitted until a route is found. Typically, the changes in the routes are handled by route maintenance schemes, in which the node experiencing route failure flood the network to find a new route. Flooding route discovery and maintenance packets may not be a problem in small scale networks. However, the cost of flooding the network is significant in large scale networks. By contrast, proactive protocols like Destination Sequence Distance Vector (DSDV) [20] periodically broadcast route information across the entire network. When a node needs to send a packet to a destination node, it knows the path to this node without the route discovery phase. Similar to reactive protocols, network floods of control packets cause high overhead. In dynamic networks, the nodes need to broadcast state information frequently in order to keep up with the changes in the network topology.

Position based protocols such as GPSR [15], provide a scalable solution to routing problem by leveraging the geographic coordinates of nodes to route packets. A packet is forwarded in the "general direction" of the destination until the destination is reached. The nodes maintain little to no state in order to forward packets to their destination given the locations of the destination nodes are known. Hence, they heavily rely on a location service such as Grid Location Service (GLS) [18]. GLS partitions the network into structured grids forming a structured geographical hierarchy, which is hard to maintain as the network size and node mobility increases.

The routing problem in large and dynamic networks is also addressed in [8], [11], and [23]. A reactive protocol, FRESH is proposed in [8] where the route discovery process is composed of several small scope searches instead of single large scope network flood. In each search, the intermediate node that keeps the packet looks for a node that has encountered the destination more recently than itself. In [11], Grossglauser and Vetterli propose an algorithm named EASE, in which every node maintains an encounter history consisting of the location as well as the time of its last encounter with every other node. The authors show that the nodes efficiently find routes using this history. A scalable link state routing protocol, Hazy Sighted Link State (HSLS) is proposed in [23], where nodes propagate link state information to farther nodes at decreasing rates by flooding link state update messages with variable TTL values.

Although these algorithms significantly reduce the routing overhead, they still depend upon strong state semantics which comes under strain when the network becomes both large and highly dynamic. To the extent that state about a dynamic part of the network, individual links or individual nodes need to be disseminated to a large subset of nodes, it implies higher control traffic to maintain/refresh brittle indirection state. WSR takes a hybrid proactive/reactive approach and uses efficient random directional walks both for the dissemination ("put") and discovery ("get") phases. The state of a dynamic entity (ID-to-location mapping) is also disseminated to fewer nodes and aggregated/weakened for scalability.

Probabilistic routing is a concept that draws attraction in P2P networks. [9] and [19] use unstructured pure random walks to locate an object in a P2P network. In [22] and [17], Bloom filters are used to bias random walks. Kumar et al. introduce exponentially decaying Bloom filters (EDBF) in [17], which is a “weak” representation of a set of objects. We apply this concept to store a probabilistic set-of-IDs corresponding to nodes placed within an area. The difference between EDBF and our weak states is that in [17], EDBF is used to set up implicit gradients between the nodes by comparing the signals obtained in successive nodes. On the other, we use decayed Bloom filters to set up explicit mappings between a weak set-of-IDs to a geographical region.

Routing in Delay Tolerant Networks (DTN) can be categorized into two classes. *Predictable* routing uses oracles about the global future view of the network [12]. *Opportunistic* techniques on the hand are deployed where the nodes have no global information on the network. Most of the opportunistic routing mechanisms do not use explicit states to assist routing the data packets. Rather, they rely on the natural node mobility [24], [26]. These works are inspired by the model in [10]. If the mobility scope is small relative to the size of the network, packets may not be delivered to destination. Similar to our mechanism, PROPHET positions itself between the two extremes. It maintains transitive probabilities for each destination like a probabilistic distance vector. The probabilities are updated during encounters with nodes and also aged over time. Though it provides the value of probabilistic hints, it has not been tied to a scalable scheme. It has no notion of aggregation or explicit mapping.

The weak state mechanism can also function as a distributed hashing method. Therefore, WSR resembles with the distributed hash tables (DHT) that provide lookup services at large scale P2P networks. In a DHT, every node stores a range of keys and any node can locate a the node a key is stored using *consistent hashing* [14]. The DHTs rely on a structured overlay network, which can be in the form of ring [25], virtual d-dimensional torus [21], etc. The source of dynamism in a P2P network is dynamic node addition and node deletion. On the other hand, in ad-hoc and delay tolerant networks, the dynamism stems from node mobility. [6] shows that maintaining the structure is hard and may require substantial overhead in P2P systems. This claim is also true in a mobile network setting. A routing protocol for wireless networks inspired by DHTs, Virtual Ring Routing (VRR), is proposed in [5]. The authors show that increasing the node mobility and network size degrades the network performance even though the protocol does not require network flooding. GIA [6] attempts to make Gnutella (a flooding-based unstructured lookup method) scalable. However, GIA achieves this by depending upon the heterogeneity, in particular by leveraging nodes with higher storage and connectivity capabilities. In contrast WSR can achieve similar objectives of scaling and rare-object recall without constraints on the degree distribution or dependence on super-nodes in the context of large-scale dynamic networks.

Dimension	WSR	GLS	Prophet	HSLS	VRR
Weak**S**trong Semantics	W	S	W	S	S
State Aggregation	YES	NO	NO	NO	NO
Unstructured \\ **S**tructured	U	S	U	U	S
Probabilistic Forwarding	YES	NO	YES	NO	NO
Explicit**I**mplicit Indirection	E	E	I	E	E

Table 1: The comparison of the key design dimensions in WSR and other schemes

The comparison of WSR and the related work in terms of the key design dimensions is summarized in Table 1.

3. WEAK STATE ROUTING

In this section, we detail the assumptions, weak state concept, specifications of WSR. Specifically, we address the following:

1. Assumptions made by WSR
2. Explanation of state concept and weakening the states
3. Proactive element of WSR: location announcements
4. Packet forwarding strategy

3.1 Assumptions

The assumptions WSR makes are not very different from those made by a traditional location based routing protocol: All nodes know their positions on a 2-D plane, either using a GPS device or through any other means. By using periodic single hop beacon messages, the nodes also know their neighbors and their positions. The nodes have uniform omnidirectional antennas. The information on node mobility is limited by the maximum node speed. The source nodes cannot mark packets with a definite location of the destination through an omniscient location database.

3.2 Weak States

We now explain the weak state concept we propose. Note that our design is only one instance of the weak state concept. Other realizations are also possible.

Component Notation	Description
B	Weak Bloom Filter containing the SetofIDs portion
X	X coordinate of the center of the GeoRegion
Y	Y coordinate of the center of the GeoRegion
R	Magnitude of the radius of the GeoRegion

Table 2: Components that constitute a weak state

Each state corresponds to an indirection or a mapping from a persistent node ID or a collection of IDs (SetofIDs) to a geographical region (GeoRegion) in which the node is believed to be in currently[1]. The state information captures the uncertainty in the mapping. We call such state information *weak state*. A weak state is composed of two portions: the SetofIDs portion that is represented

[1]In the rest of the paper, we use the terms mapping and indirection interchangeably.

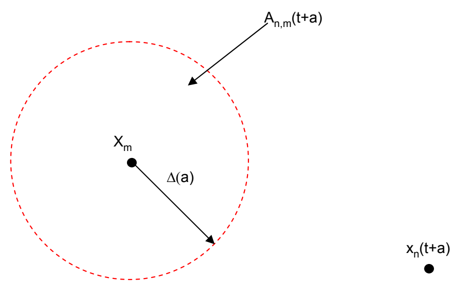

Figure 2: Geographical decaying: If node n knows that node m was located at X_m at time t, at $t + a$ the node is located within the circle centered at X_m with radius $\Delta(a)$, worst case displacement in a.

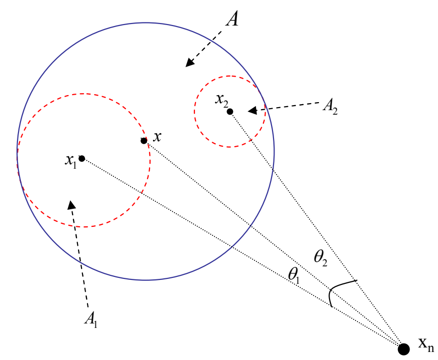

Figure 3: If the GeoRegion portions of two mappings are located close to each with respect to the location of the node maintaining these mappings, they are combined into one mapping. The new GeoRegion is the smallest circle that contains both GeoRegions

by a weak Bloom filter and the GeoRegion portion described by a circle with coordinates of its center and the magnitude of its radius. The components of a weak state are summarized in Table 2.

Weak states have two aspects of probabilistic behavior: the SetofIDs portion is probabilistic in terms of the membership, and the GeoRegion is probabilistic in terms of scope. The weakness of the SetofIDs allows it to remain valid, be locally updated without explicit protocol messaging and exhibit persistence rather than having a strong semantics which can be quickly invalidated. This way a large part of the network can have some weak state about the location of any particular node. An explicit mapping from SetofIDs to GeoRegion gives a concrete bias that can be reinforced as we get closer to the destination. Since the uncertainty of mappings increases in time, we periodically weaken the state information.

Let $x_n(t)$ denote the location of node n at time t and consider the case where node n knows that node m is located at point X_m at time t, $x_m(t) = X_m$ (See Fig. 2). At time $t+a$ where $a \geq 0$, node n is not certain about the location of node m. The location now becomes a random process based on the mobility pattern of the node. To convey this uncertainty, we decay the location information: if node n knows the maximum possible speed S a node can move, it can determine the region, $A_{n,m}(t)$ where the probability of node m being in that region is 1, $P\{x_m(t + a) \in A_{n,m}(t)|x_m(t) = X_m\} = 1$. $A_{n,m}(t)$ is a circular area centered at X_m with radius $\Delta(a)$, the worst case displacement of a node in a time interval of a, i.e. $\Delta(a) = S \times a$. Now, the state corresponds to a mapping from a node ID to a GeoRegion $A_{n,m}(t)$.

Such an approach to mapping a node ID to a geographical region enables us to combine several mappings of which GeoRegion parts are close enough, into one state. This way we can have mappings from *SetofIDs* to GeoRegion. Consider the scenario in Fig. 3, where node n maintains two states with corresponding GeoRegions A_1 centered at x_1 and A_2 centered at x_2. Since the distance between the two GeoRegions is small in comparison to the distance between the either GeoRegion and the location of node n, the node can aggregate these two mappings. The GeoRegion of the new aggregated mapping is the smallest circle A centered at x, midpoint of x_1 and x_2, that contains both A_1 and A_2. Note that for node n, the deflection from x_1 and x_2 to x, θ_1 and θ_2 respectively, is minor. The corresponding SetofIDs portion of the new indirection aggregate is the union of the SetofIDs parts of the two mappings before the aggregation. After the mappings are aggregated, we keep decaying the mapping geographically. This way, the mapping gives the smallest area that contains all the nodes in the SetofIDs portion of the new mapping with probability 1.

By broadening the GeoRegion portion of the mapping, we weaken the state information: the uncertainty in the location of a node that is an element of the SetofIDs portion of the mapping increases every time the GeoRegion is expanded. However, a mapping with a very large uncertainty level is not useful for making forwarding decisions. Therefore, we use a limit on the radius of the GeoRegion portion of the mapping. This threshold is not static and depends on the distance between the center of the GeoRegion and the location of the node maintaining this state because we want to have more definite location information as we get closer to the destination. Once the perimeter of the GeoRegion reaches the location of the node, we no longer broaden GeoRegion portion of the mapping.

Once the limit on the radius of the GeoRegion is achieved, we stop decaying the mapping geographically. In this case, the probability that a node in the SetofIDs portion of the mapping is placed in this GeoRegion portion of the mapping is not 1. In order to represent this uncertainty, we decay the SetofIDs portion of the mapping and diminish the strength of our belief of that node being in this area.

To represent the SetofIDs part of the mapping, we use a variant of the Bloom filter data structure [4]. A Bloom filter is described by an array of u bits, which are all initialized to 0 at the beginning. A fixed number k of hash functions are employed. When inserting an element m (node ID in our case), the bits in k array positions $h_1(m), \ldots, h_k(m)$, which are obtained by feeding m to each of the hash functions, are set to 1. The union of the filters is a new filter with the same size and characterized by the same hash functions, and is obtained by the bitwise OR operation. In a regular Bloom filter, the membership query for an element n yields *yes* only if all the bits in array positions $h_1(n), \ldots, h_k(n)$ are 1.

Bloom filters are subject to false positives and the false positive rate increases with the number of elements added to the filter. To reduce the false positives, we also use a limit on the total number of bits set to 1 in Bloom filter B, which we call the cardinality of B and denote by $|B|$. Similar to the limit on the radius of the GeoRegion portion of the mapping, reaching the cardinality limit triggers the decay of the mapping in time. The decaying schedule

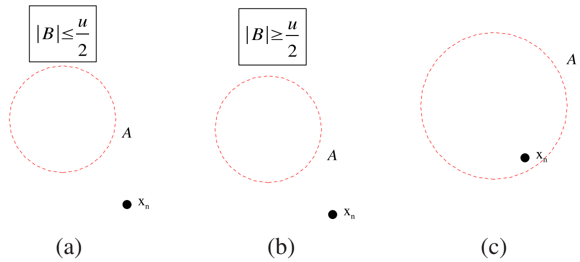

Figure 4: Conditions on decaying of mapping with GeoRegion A and SetofIDs portion B that is maintained at node with location x_n. u is the WBF length. At each decaying instant, if the node is located outside the GeoRegion and the cardinality of the SetofIDs portion $|B|$ is below $u/2$ (a), the GeoRegion is decayed. Weakening the SetofIDs portion is triggered if the WBF fills up (b), or the GeoRegion grows so that the node is located within the GeoRegion (c).

of a mapping is summarized in Fig. 4. In addition, if the union of two mappings violate either criteria, we do not combine them.

Once either the limit on the maximum radius of the GeoRegion portion of the mapping or the limit on the cardinality of the SetofIDs portion of the mapping is reached, the nodes decay the SetofIDs portion of the indirections they maintain. We refer to this weakened SetofIDs portion of the indirection as a "weak Bloom filter" (WBF). A WBF is similar to exponentially decaying Bloom filter (EDBF) proposed for P2P networks in [17]. In that method, the Bloom filter data structure is used to represent a set of resources that can be reached through a particular node either directly or over multiple hops after that node. The filters are decayed at every hop they propagate by setting a random subset of bits to 0. In other words, the filter becomes a probabilistic or fuzzy set, and a query yields a *signal* (or probability) of membership, rather than a binary yes/no or 0/1 answer. A random walk in a P2P network would sense a local gradient in the "signal" between a node and its neighbors and use that information for biasing its progress.

In an ad-hoc network, the state information is not only a decayed Bloom filter. The mappings contain both a SetofIDs and an associated GeoRegion value. This is an explicit mapping to a geographical region rather than a implicit gradient between neighbors. Therefore, we decay the SetofIDs portion of the indirection aggregates *over time* rather than *over hops*. To do this, we periodically decay the SetofIDs portion of the mapping. At each decaying instant, the bits set to 1 are reset to 0 by a fixed probability p. This way the number of 1's corresponding to every node in the SetofIDs decay exponentially with time. Using WBF, we do not need a hard timeout value to remove the stale state information of any node. Once the cardinality of the WBF is below a threshold value, we remove the mapping since it is too old to bias any packet accurately enough.

3.3 Semantic Strength of Mappings

The forwarding decisions of the intermediate nodes are based on the quality of information that the mappings offer. We now explain the mapping quality using two strength parameters: temporal strength and spatial strength.

Temporal Strength: The temporal strength of the mapping is associated with the probability of a node being placed in the GeoRegion part of the mapping. Considering two mappings M_1 and M_2 with corresponding GeoRegions A_1 and A_2, we say that M_1 is temporally stronger for node m at time t if node m is placed in A_1 at time t with a larger probability, i.e. $P\{x_m(t) \in A_1\} > P\{x_m(t) \in A_2\}$.

Given that node m is located within a region A at time t, i.e. $x_m(t) \in A$, the probability of the node being in the same area in a future time $t + a$, $P\{x_m(t + a) \in A | x_m(t) \in A\}$ is a non-increasing function of $a \in [0, \infty)$ regardless of the mobility process of the node. Therefore, a temporal strength parameter should capture the fact that among two mappings, the one that provides more recent information about a node should be temporally stronger.

The temporal strength is strongly related to the time decaying process of the mapping. A mapping is decayed in time only if the probability that a node being in that region is not 1. In this case, we reset each 1 in the WBF part of the mapping by a fixed probability, p. For a mapping whose WBF part is denoted by B, let $\theta(m) = |\{i|B[h_i(m) = 1, i = 1, \ldots k\}|$ be the number of 1's in B corresponding to node m. A larger $\theta(m)$ indicates that the mapping contains more recent information with high probability. Therefore, the probability that the node is located within the area that the GeoRegion portion of the mapping represents, is higher. As a result, we use $\theta(m)$ as an indicator of the temporal strength. Note that the temporal strength only yields the freshness of the mapping and not the actual the probability of the node being in the corresponding area.

Spatial Strength: The spatial strength of the mapping involves the uncertainty in the GeoRegion portion of the mapping. Again, consider two mappings M_1 and M_2 with GeoRegion portions A_1 and A_2, respectively. Given that $P\{x_m(t) \in A_1\} = P\{x_m(t) \in A_2\}$ for node m, we say that M_1 is spatially stronger if A_1 represents a smaller region, i.e. its radius is smaller than that of A_2 since M_1 directs packets destined for node m to a more definite region.

3.4 Dissemination of Location Information (Put)

Our routing mechanism is based on forwarding data packets toward the region where we believe the destination node now is, using the weakened information that was generated in the past. To generate the mappings, the nodes should know where a node is located at a time in the past. We take advantage of the fact that nodes know the location of their neighbors through periodic beacon messages. Once two nodes get out of each other's transmission range, they create a mapping for each other using their last known location. We also use periodic announcements to disseminate location information. Creating a mapping is analogous to the `Put` operation where a value corresponding to a hash key is stored in the network.

In order to form rendezvous points between data packets and weak states in intermediate nodes, we forward the location announcements in *random directions*. Notice that this is different from a standard random walk where the random walker can proceed to each neighbor with equal probability. In this case, a node selects the direction of the announcement packet randomly and sends the announcement in that direction. The node first picks an angle uniformly between 0 and 2π radians. The direction on which the location announcement is sent is determined by this angle. We pick a point that is far from the location of the node along this direction (a point outside the area covered the network) and use geographical routing to forward the announcement.

When a node receives an announcement from node m, it creates a weak state with a WBF at which only the entries at indices $h_1(m), \ldots, h_k(m)$ are 1, and a GeoRegion in which the center is the location of node m and the radius is 0. Right after creating this state, the node checks if it can combine it with a previously generated mapping.

Location announcement concept we use is similar to the *Orthogonal Rendezvous Routing Protocol* (ORRP) proposed for static networks in [7]. In this paper, the authors show that a pair of orthogonal lines intersect at a point in space with high probability, so that rendezvous points can be formed between source-destination pairs in 2-D Euclidian space. In a mobile environment, it is difficult to maintain fixed, orthogonal straight lines. Instead, we send our announcements in random directions. Also, note that the prior work assumed no location information but local directional communications through directional antennas. On the other hand, we assume that the nodes use omni-directional antennas, but they are aware of their location.

By radially sending announcements in random directions at different points in time, we increase the probability that a random walk will intersect with one of these lines. Assuming the nodes are placed uniformly on the plane, the nodes that are far away from a particular node receive location announcement packets at a lower rate than the nodes closer to this node. Therefore, the uncertainty in the location of a node and the weakness of the states increase as the distance to this node increases. Even though a node has uncertain information about the location of this destination, it can bias the packet towards a region where the packet will rendezvous with a node that has more definite information.

Because both the beacon messages and the location announcements are sent proactively, the total number of control packets generated within a time interval is fixed. With N nodes in the network, the beaconing overhead is $O(N)$. On the other hand, the number of the location announcement packets forwarded within the network is bounded by $N \times T$ where T is the TTL of the announcement packets. T should be selected so that the announcement packet is able to traverse to network from one end to any other. In other words, T should depend on the network diameter. Within a fixed area, increasing the network density does not change the diameter. This implies that the total announcement overhead also scales as $O(N)$.

3.5 Forwarding Data Packets (Get)

Our data forwarding mechanism is a simple greedy algorithm. Similar to announcement packets, a data packet is initially sent in a random direction and subsequently biased at each intermediate node if the node has a weak state about the location of the destination. Since we consider WSR as a biasing technique that can be extended to dynamic networks other than ad hoc networks such as DTNs, we deploy an opportunistic packet forwarding scheme. We do not use discovery messages to locate a destination node. We also do not deploy an end-to-end feedback mechanism; the destination node does not acknowledge data packets to the source node indicating whether or not the packets were received. Therefore, every packet is transmitted only once.

This method acts as the `Get` operation which locates a key in a P2P network and retrieves the value corresponding to the key.

The strategy we use for biasing data packets is similar to the longest-prefix-match method and summarized in Algorithm 1. We use the strongest semantics match in order to prioritize the indirections to decide how to bias the random walk. We first consider the temporal strength of the mapping to find the area that the destination node is most likely in. To do that, the destination is node ID, m, is looked up in the WBF part of each mapping maintained by the current node. For each mapping i, the result of this lookup is a value Φ, which is the total number of bits set to 1 in locations indexed by $h_j(m), j \in 1 \ldots k$ in the WBF part of the mapping i, i.e. Φ is the temporal strength of the mapping. The current bias of the packet is carried in the packet since not every intermediate node receiving the packet has information about the location of the destination node. If the mapping is stronger than the current bias, i.e. temporally stronger than the current bias or spatially stronger while its temporal strength is equal to the current bias, the new bias is determined by this mapping. Ties among mappings are broken randomly. The packet is forwarded toward the center of the GeoRegion part of the mapping of the strongest encountered mapping through a geographical routing scheme such as GPSR. An illustration of the a route determined by WSR is given in Fig. 5.

Algorithm 1 Algorithm for biasing packets in WSR

`ForwardPacket` (p)
//Consider the bias previously given to the packet
$m \leftarrow destination(p)$
$\Theta \leftarrow Temporal(p)$
$R \leftarrow Spatial(p)$
$(x, y) \leftarrow TargetLocation(p)$
//Find the strongest local mapping indicating the whereabouts of the node
for all mapping i **do**
  $\Theta_i \leftarrow Lookup(i, m)$
  if ($\Theta_i \geq \Theta$) OR ($\Theta_i = \Theta$ AND $Radius_i < R$) **then**
    $\Theta \leftarrow \Theta_i$
    $R \leftarrow Radius_i$
    $(x, y) \leftarrow Center_i$
  end if
end for
$Temporal(p) \leftarrow \Theta$
$Spatial(p) \leftarrow R$
$TargetLocation(p) \leftarrow (x, y)$
Use a geographic forwarding scheme to send the packet to $TargetLocation(p)$

`Lookup`(i, m)
$\Phi \leftarrow 0$
for all $q \in \{1, 2, \ldots k\}$ **do**
  $\Phi \leftarrow \Phi + WBF_i[h_q(m)]$
end for
Return Φ

In our mechanism, an intermediate node drops a data packet if the packet cannot rendezvous with a state information within a some number of hops or the packet TTL expires. If a data packet cannot be further forwarded in the initially selected random direction, the intermediate node picks a new random direction and sends the data packet in this direction.

4. PERFORMANCE EVALUATION

In this section, we provide the performance evaluation of our weak state paradigm. First, we present how WSR behaves under various mobility levels. Then, we compare WSR with DSR and GLS location service combined with GPSR protocol. Similar to WSR, GLS is also based on a distributed hashing mechanism. However, GLS relies on a underlying structure similar to traditional DHTs. Through GLS, we present how unstructured methods enhance the network performance in dynamic settings.

The simulations were performed using Network Simulator (ns2) [2]. In order to forward data packets toward the point the weak states bias, we use GPSR. For DSR, we used the standard implementation in the ns2 distribution. For GLS-GPSR scheme, we trustfully used the implementation available in [1].

In our simulations, the mobile nodes move in a 2500 m $\times$ 2500 m area. At the MAC layer we use IEEE 802.11 standard. The nodes

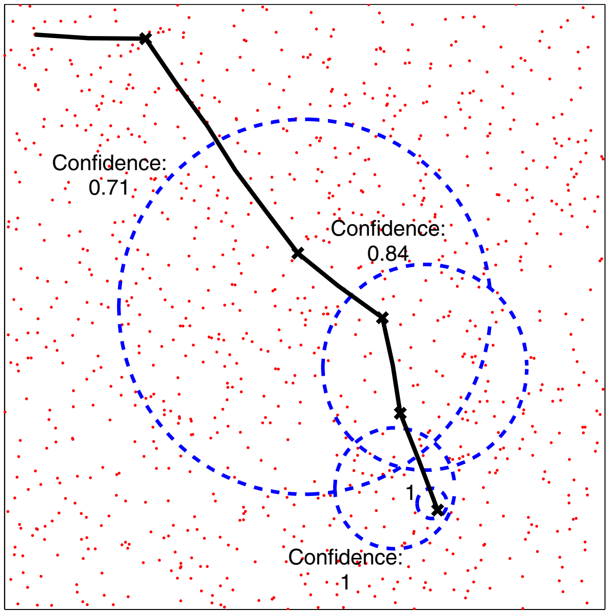

Figure 5: An example WSR route for 1000 nodes. The dots represent the nodes in the network. The solid line is the path that the packet follows. The packet is biased by the nodes that are represented as crosses. The GeoRegions are shown by the dashed circles. The confidence values near the circles given by $\theta(m)/k$ where m is the destination node, $\theta(m)$ is the temporal strength of the mapping biasing the packet. k is the number of hash functions.

have 250 m omnidirectional transmission range. The nodes move with a modified version of random waypoint mobility model which is originally defined in [13]. In this model, a node chooses a random destination point in the simulation area and a speed value uniformly from a range defined by a minimum and a maximum node speed values; v_{min} and v_{max}, respectively. In [27], it has been shown that using 0 minimum speed causes instantaneous average node speed to decrease continuously over time, which mislead to incorrect conclusions on the performance. Hence, we use a positive minimum speed value. Once the node arrives at the chosen waypoint, it pauses for a uniformly distributed period of time.

We use MAC layer failure notifications from the 802.11 MAC layer. Use of this feedback informs nodes about the loss of a neighbor and provides retransmission opportunities for data packets in case of transmission failure. If a link loss occurs when a location announcement packet is being transmitted, the packet is dropped.

The performance metrics we use to evaluate performance are the data packet delivery ratio, control routing overhead, number of states maintained in the network, the normalized path length, number of transmissions per successfully delivered packets and end-to-end delay for data packets. We examine these metrics with respect to the growth of network sizes. When comparing WSR with the prior work, we used $v_{min} = 5$ m/s and $v_{max} = 10$ m/s. To evaluate the effect of node mobility on WSR, the other node speed values we used are: $v_{min} = 10$ m/s, $v_{max} = 10$ m/s and $v_{min} = 10$ m/s, $v_{max} = 20$ m/s. For all scenarios, the mean pause time is 10 seconds.

We simulate WSR for 1000 seconds and other protocols for 500 seconds because DSR and GLS simulations require too much memory. In WSR, we deploy constant bit rate connections between 60 randomly selected source-destination pairs. In DSR and GLS, the number of connections is 30. For each connection a 512 byte data packet is sent every second for 100 seconds. All results are averaged over 5 different topologies with different mobility and connection scenarios. The connections are initiated at random times making sure the last packet is sent before the end of the simulation.

The parameters we used for WSR are as follows: The beaconing interval we used is 1 second while the announcement interval is 5 seconds. The beaconing interval should be set such that the node can keep up-to-date neighbor tables. The announcement interval should be in the order of the time in which node displacement becomes comparable to transmission range. Another choice would be to send an announcement packet once the node displacement is above some threshold value. Our results indicate a constant announcement interval is sufficient in terms of overhead and reachability when nodes move with speeds up to 20 m/s. In high mobility scenarios, the probability of a node being located within a given area decreases faster; therefore, the temporal strength of a mapping should decrease faster as the node mobility increases. Hence, the probability value to decay mappings should be larger in high mobility cases. The probability we used to decay SetofIDs portion of the mappings is $p = 0.01$ in the first two scenarios. In the high mobility case, we decay the SetofIDs portions of the mappings with probability $p = 0.02$. Note that we assume the only information the nodes possess about the mobility model is their maximum possible speed. Since, the maximum speed is the same in first scenarios, we decay the mappings with the same value in both scenarios. Using an arbitrarily large probability value causes wild variations on the mapping distribution and strength. In this case, the nodes store less mappings, and the delivery ratio and and path quality terribly suffer. The WBF width is $u = 2048$ bits and the number of hash functions is $k = 32$. The maximum WBF cardinality value we allow is 1024. For data packets, we used the default TTL value of GPSR protocol, 128. We want our announcement packets to be able to traverse to entire network. To do this, the TTL of the announcement packets should be in the order of the diameter of the network. In the scenario we are considering, the maximum distance between any points in the network is around 14.142 times the transmission range of the nodes. We set the TTL of the announcement packets to 16, which enables any packet to traverse the network and also is the TTL of the control packets in DSR.

The vertical bars in the graphs shown in this section correspond to 95% confidence interval. All figures in this section other than Fig. 10 and Fig. 11 contain the confidence intervals.

4.1 WSR Results

The results in this section only involve the performance of WSR. We show how the performance of WSR changes with respect to the node mobility level and number of nodes in the network.

Fig. 6 shows how the fraction of data packets delivered to their destination nodes change with respect to the network size under different speed values. In all cases, the WSR succeeds delivering at least 98% of the packets. The figure shows that the delivery ratio is very high regardless of the number of the nodes and mobility level. No matter how fast nodes are moving, weak states bias almost all data packets towards their destination.

At each location announcement, nodes that receive this announcement creates a mapping for the announcing node and keep this mapping until the strength of mapping is below some threshold value. Since the announcements are sent along random directions, the nodes receiving each announcement will be different with high probability. Therefore, a large part of the network maintain a weak location information on every node. The number of transmissions along a random direction until the data packet encounters with the

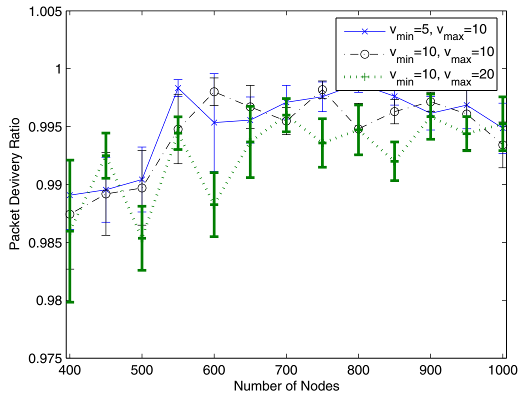

Figure 6: Packet delivery success rate with respect to the number of nodes. The delivery ratio is high regardless of the network size. 95% confidence intervals shown are also small in comparison to the mean values.

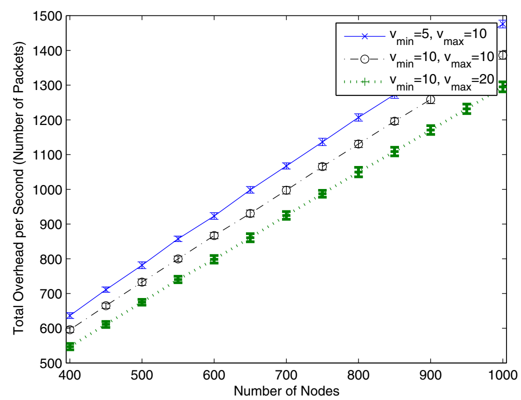

Figure 7: Routing Protocol Overhead in WSR. The overhead is the sum beacon and location announcement packets forwarded in the network. Overhead increases linearly with the number of nodes.

location information can be approximated by a exponential random variable, whose tail approaches to 0 quickly. Hence, the packet is quickly received by an intermediate node maintaining a weak state about the destination, and biased along the direction the corresponding mapping yields. If this mapping is strong, the data packet will encounter the destination node or a previous neighbor of the destination that maintains a stronger mapping as it proceeds along the direction biased by the mapping with high probability. If the mapping is weak, then the announcement that created the state information is received a long time ago, implying a large number of more recent announcements are sent out by the destination. In this case, the probability of encountering a node that received a fresher announcement and thus maintaining a stronger mapping is high. In any case, the data packet will be received by a intermediate node that has a stronger mapping until it reaches at the destination. Consequently, the delivery ratio is very high.

Fig. 7 shows the control overhead incurred per second in terms of the total number of times control packets transmitted (originated or forwarded). The control packets for WSR consist of the periodic beacons and the location announcements. Since the both beacons and the location announcement packets are sent proactively, the expected number of control packets generated by a node within a fixed time interval is constant. The beacons are one hop packets and they are not further forwarded. Since an announcement packet has a fixed TTL, the number of times it is forwarded is bounded by a constant. Fig. 7 shows that the overhead increases linearly with the number of nodes in the network independent of node speed, which is consistent with our discussion in Section 3.4. Since the location announcement packets are dropped in case of transmission failures, the overhead slightly decreases with increasing mobility level because of more frequent neighbor losses.

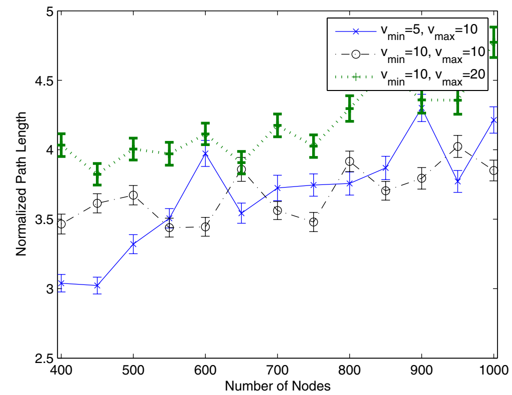

Figure 8: Path efficiency of WSR vs. Number of Nodes. WSR packets follow long paths due to the state weakness.

Note the transmission cost of routing protocol also includes increased length of paths that the data packets take. The efficiency of WSR in the control overhead performance comes at the cost of increased path length. Since we simulate very large networks, calculating the shortest path for every generated packet is extremely computationally expensive and hence not possible. As a result, we evaluate the cost of increased path length in terms of normalized path length, which corresponds to the ratio of the total distance a data packet travels to the distance between the source-destination pair. Even though this is a measure of Euclidean length a packet travels, it roughly indicates the hop-length and it also provides a comparison between the Euclidean straight-line distance.

Fig. 8 shows the normalized path length. The paths taken by data packets are less than 4.5 times longer than the straight line, due to the probabilistic nature of WSR. The packets are sent towards region where the destination is believed to be in instead of the exact point where the destination is located. As the network size grows, location announcement packets are distributed to a larger number of nodes. Therefore, the average bias given to data packets becomes weaker and the paths taken by the data packets get longer. The figure shows that path length slightly increases with the node speed values. Since in high mobility case there are more transmission failures, more announcement packets are dropped, which decreases the effective announcement rate.

Fig. 9 shows the effect of the number of nodes and the node speed values on the total number of mappings in the network at the end of the simulation. Each mapping corresponds to a constant length WBF, and three quantities describing the GeoRegion: the coordinates of its center and the length of its radius. Our results show that the number of mappings does not change significantly with the mobility level. Note that we decay the SetofIDs portion of the mappings more aggressively in the last scenario since the prob-

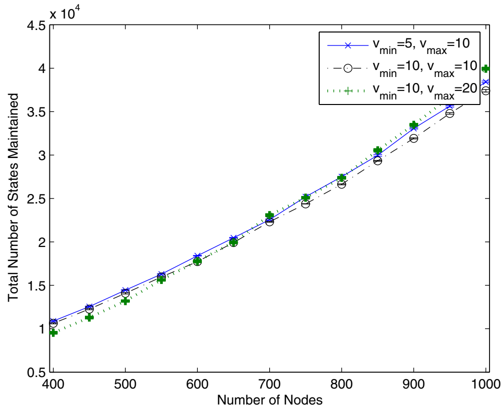

Figure 9: Number of mappings stored vs. Number of Nodes. Each state corresponds to a WBF and a GeoRegion. The number of mappings stored in the network scales as $O(N^{3/2})$

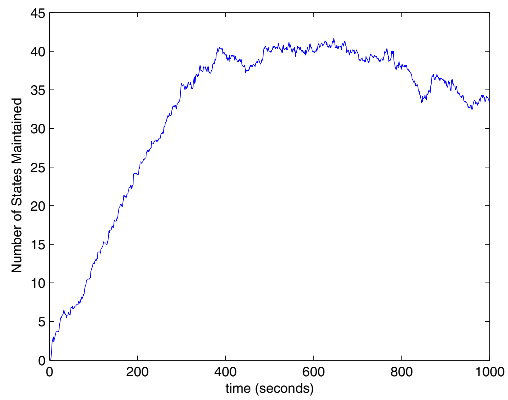

Figure 10: Evolution of the number of mappings in a node. The total number of nodes in the network is 1000. $v_{min} = 10$ and $v_{max} = 10$. Announcement rate and decaying rate matches and hence the number of states stored is bounded.

ability for nodes to remain in the same GeoRegion is lower for the high mobility case. The location announcements are sent along the random directions with a TTL value that is sufficient to traverse the network along that direction. If the nodes are uniformly distributed in the network, each announcement is received by $\Omega(\sqrt{N})$ nodes and if the rate at which the nodes create mappings matches the rate at which the nodes remove mappings (as in Fig. 10), this is also an upper bound and $O(\sqrt{N})$ nodes maintain weak states about a destination node; therefore, the number of mappings stored in the network scales as $\Theta(\sqrt{N})$ implying that the total number of mappings in the network scales as $\Theta(N^{3/2})$. Even though the nodes are not uniformly distributed in our simulations since they move according to the random waypoint mobility model, we empirically see that this argument holds and the number of mappings in WSR has similar scalability properties to the number of states stored in ORRP. According to our simulation results, the increase in the number of mappings in WSR shows similar characteristics to the increase in the number of routing table entries in ORRP. However, because of aggregation, the average number of mappings a node maintains is less in comparison to ORRP.

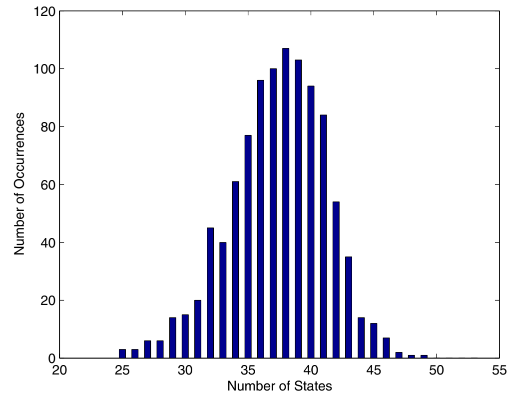

Figure 11: The distribution of the number of mappings to the nodes in the network. The total number of nodes is 1000. $v_{min} = 10$ and $v_{max} = 10$. States are well distributed with a coefficient of variation 0.1 (with standard deviation of 3.8 and mean 37.4)

The nodes continuously receive location announcements and they decay the mappings. Fig. 10 shows how the number of mappings evolve in a random node in a $N = 1000$ node network and the node speed is always 10 m/s. The figure shows that the announcement rate matches with the decaying and aggregation rate. Therefore, the number of mappings maintained at a node is bounded and oscillates within a steady state interval[2].

Fig. 11 shows the the distribution of the mappings in a 1000 node network with node speed 10 m/s. The distribution has a mean of 37.4 and a standard deviation of 3.8. The coefficient of the variation (CoV) of the distribution (the ratio of the standard deviation to the mean) is only 0.1. Hence, the distribution has a low degree of variation with respect to the mean and the mappings in the network are well distributed because of the location announcement method in the random directions. Therefore, failure of a single node will not drastically influence network performance.

4.2 Comparison with the Prior Work

Simulations in this section compare Weak State Routing with Dynamic Source Routing Protocol and Grid Location Service combined with GPSR. The simulation setup is same as the one in the previous subsection. We compare these schemes only when $v_{min} = 5$ m/s and $v_{max} = 10$ m/s because the prior works [13], [16] explicitly show that node mobility degrades the performance of their schemes.

Fig. 12 shows the fraction of data packets successfully delivered. In WSR, almost all packets are received by their destination nodes. DSR delivers only a small portion of the packets in large scale networks due to its flooding nature. GLS globally partitions the coordinate system into a hierarchy of grids with squares of increasing size. As the order of the square increases, the area also increases. Each node maintains a location server in each square and updates its location information in the location servers at frequencies decreasing with the order of square. The location server for a node in each grid is the node with the least ID greater than its own ID in the corresponding square. When a node has a packet to send to the destination, it searches the location servers of the destination

[2]The number of mappings stabilize at around t=300 seconds in all cases. Therefore, we start sending packets at this time in our simulations.

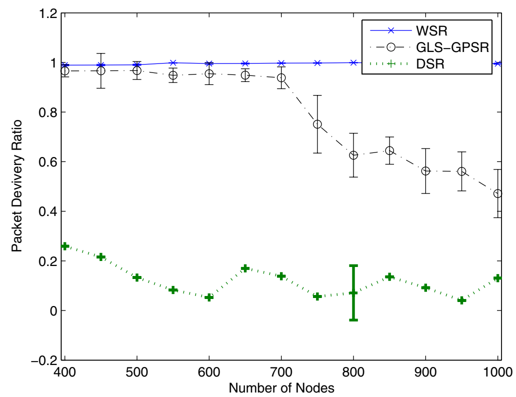

Figure 12: Packet delivery success rate with respect to the number of nodes. $v_{min} = 5$ and $v_{max} = 10$. WSR consistently delivers a very large fraction of packets. DSR has a low delivery rate and GLS performance degrades with the network size.

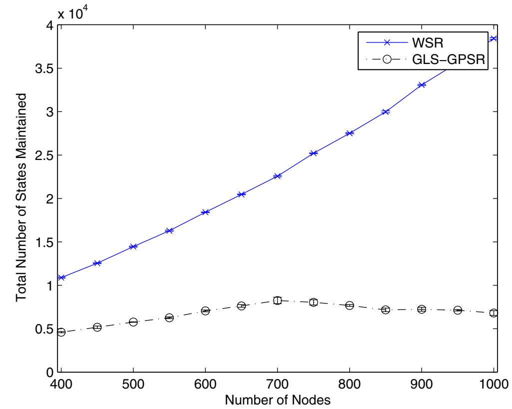

Figure 14: Total States Stored vs. The Number of Nodes. $v_{min} = 5$ and $v_{max} = 10$. Total storage size increases as $O(N^{3/2})$ in WSR. Due to overhead, update packets are lost in GLS resulting in lower update rate and decrease in the stored state.

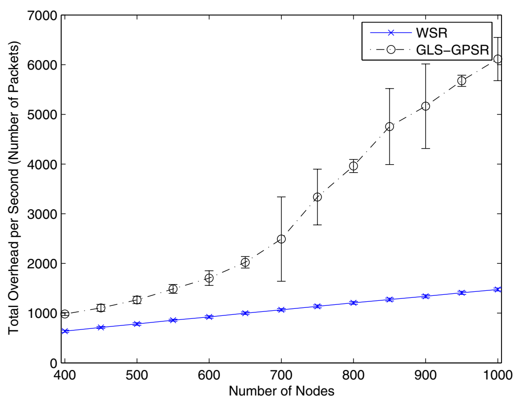

Figure 13: Control Packet Overhead vs Number of Nodes. $v_{min} = 5$ and $v_{max} = 10$. Overhead increases linearly in WSR and superlinearly in GLS.

using a request packet. The request packet follows these location servers until it finds the destination node. In a large network finding the correct location server is a difficult task. As the number of nodes increases, the number of nodes getting in and out of a square increases. In this case, the location servers of a node change frequently and the location server in which a destination's location is looked up may be different from the server the destination most recently updated its location. Also, due to superlinearly increasing overhead (see Fig 13), more location announcement packets are lost. This situation decreases the effective update rate and the number of entries stored in the network decreases because the timeout rate remains the same (see Fig. 14). Hence, the number of times request packets cannot locate the destination nodes increases, which causes undelivered packets. Overhead also causes buffer overflow and dropping data packets. Fig. 12 shows that these effects are much more visible after 700 nodes. Still, note that even when the network size is small, WSR delivers packets with higher rates.

Fig. 13 shows the protocol overhead for WSR and GLS-GPSR. We do not include the overhead for DSR because it is too high, the scale of the graph prevents the reader from seeing the difference between WSR and GLS-GPSR. In WSR, the total overhead increases linearly with the number of nodes in network as we discussed before. In GLS-GPSR, the overhead increases much more aggressively in order to keep location information in location servers up-to-date. Note that there is a correlation between the packet delivery ratio and overhead in GLS. If a location request packet fails to locate the destination, the source node issues another request and thus the overhead increases. The resulting control packet overhead increases superlinearly with the number of nodes.

In Fig. 14, we compare the number of mappings stored in network when WSR is used with the number entries in GLS location database. We do not include a curve for DSR here as in the overhead analysis. In DSR, the number of states would correspond to the number of entries in the route caches. Since the routing information is disseminated through flooding the network, the route caches quickly build up in DSR. To make the comparison between GLS and WSR clearer, we do not include number DSR cache entries in this figure. Even if WSR scales well in storage ($\Theta(N^{3/2})$), GLS is more efficient in terms of database size. We note that when the number the number of nodes increases, more location update packets are lost due to overhead (see Fig. 13). This in turn decreases the effective location update rate and causes the number of states to reduce since out-of-date location information is removed with the same rate.

The average normalized path lengths of the protocols are given in Fig. 15. These results only include the packets that are successfully delivered to their destination nodes. Since DSR heavily uses caching to forward data packets, the packets do not always end up following the shortest path. GLS also uses caching. However, the cache is used only for forwarding requests and updates. For successfully delivered packets, it is certain that the exact location of the destination node is known. Therefore, the data packets follow the efficient routes given by geographic forwarding scheme.

Normalized path length metric is useful, but it may not capture the costs arising from longer paths. Even though we are unable to compute the shortest path for every generated packet to compare the length of paths, relative comparison of "transmission effort" of each protocol is useful. The transmission effort metric is the average number of transmissions for each successfully delivered packet. Note that this metric also includes the number of retransmissions.

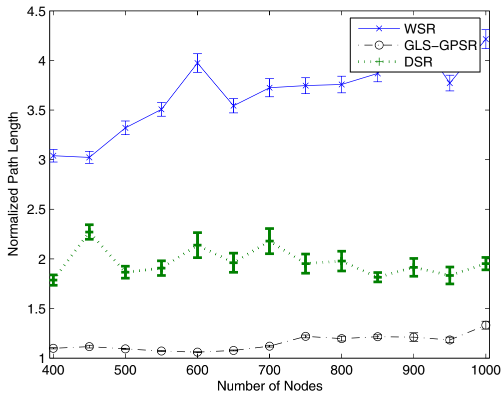

Figure 15: Path efficiency of WSR vs. Number of Nodes in the Network. $v_{min} = 5$ and $v_{max} = 10$. Data packets follow optimal routes in GLS and sub-optimal routes in DSR. WSR are paths are much longer.

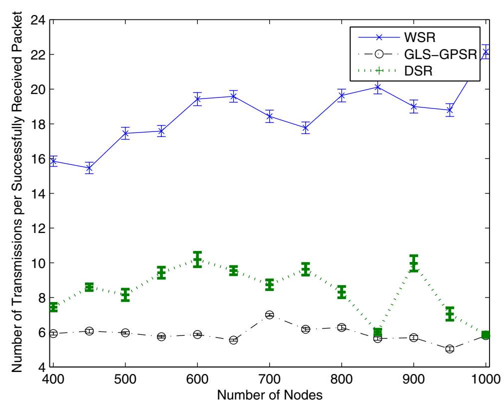

Figure 16: Number of times a data packet is forwarded vs. Number of Nodes. $v_{min} = 5$ and $v_{max} = 10$. With GLS, the data packets are sent to the exact location of destination, with fewest number of transmissions. The weakness of the location information causes more transmissions in WSR.

Fig. 16 shows that WSR delivers packets with the highest transmission effort as the normalized path length metric suggests. Since GLS delivers packets to the destinations that the protocol is able to locate, geographic routing delivers the packet with the least effort. Due to usage of cache in DSR, the packets may follow paths longer than the shortest available one.

One drawback of following routes with higher hop-count is that the data packets suffer from a larger end-to-end delay. Contrary to DSR and GLS, there is no route discovery or location query phases in WSR. The packet forwarding has opportunistic nature. Fig. 17 presents the average end-to-end delay values. Even though one request packet is sent for each flow in DSR and GLS, until the time the source node receives a route reply, all the arriving packets are queued in the source node. As we discussed before, in large networks GLS usually makes more than one attempt to locate the destination when the first location request packet is not able to locate the destination which increases the time until the queued packets are sent. The delay values are only calculated for successfully received packets. In large networks, DSR delivers only a small fraction of packets between close source - destination pair with small delay values.

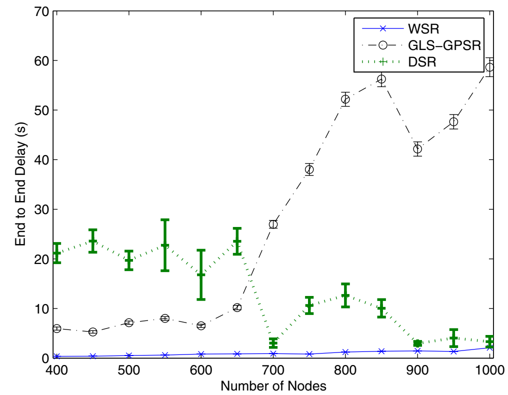

Figure 17: Data packet delay vs the total number of nodes. $v_{min} = 5$ and $v_{max} = 10$. WSR quickly delivers packets. GLS and DSR requires route discovery, which takes a large amount of time

Another factor in high end-to-end delay is the overhead. In case of link failures, DSR tries to maintain the path using route error messages. Similarly, in GLS location servers require frequent location updates and location queries for data packets in mobile environments since the formerly obtained information becomes invalid very soon. As the number nodes increases, the number of control packets stored in the intermediate node buffers increases much faster in DSR and GLS. The transmission of control packets cause extra delay. Since WSR has no location or route discovery phase and average per-node overhead remains the same, the end-to-end delay is smaller than these schemes.

Note that delay is not the only issue affected by long paths. As the number hops a packet takes increases, the a network can serve fewer connections since more nodes participate in forwarding a single session. For networks that require short paths due to reasons like constrained energy or capacity, the simulation results may suggest using GLS. However, it is possible to use WSR as a location service as we discussed before. At the beginning of a connection, a random walk that follows weak states can be issued to locate the destination. This way, data packets can follow short paths with slightly increased control overhead and delay.

5. FUTURE WORK

ns-2 simulations has shown that WSR delivers packets with high reliability, low overhead and number of stored states at the cost of increased path length. We are currently developing a mathematical analysis of the WSR. We will use a probabilistic framework to mathematically prove that source initiated random walks rendezvous with state information with high probability and the strength of the state information increases as the biased random walk gets closer to the destination. With this framework, we aim to determine WSR parameters more systematically.

In this paper, we have evaluated the performance of our mechanism on ns-2 simulator. To compare it with prior work, we trusted publicly available models developed on the same platform by their designers. We plan to compare WSR with protocols that are more suitable for the scenarios we considered.

While we have considered WSR in the context of large, mobile and connected ad-hoc networks in this paper, we believe the weak state concept can be also adopted to networks that may have occasional disconnections such as metro-scale vehicular networks. In our simulations, we used node speed values up to 20 m/s, which are consistent with metropolitan region vehicular speeds, and showed that increased speed does not have drastically deteriorates the network performance. However, announcing position information cannot result in an efficient location information dissemination in intermittently connected networks. A straightforward strategy for this setting is that two nodes can exchange their indirection information once they become neighbors. This way, our random walk & weak state based scheme can be directly extended to support a store-carry-forward paradigm which is the fundamental concept in DTN routing. This property will not be true for a routing protocol that uses absolute states. We will investigate the concept of diffusing weak states together with an aggregation policy since intermittent connectivity may require a different strategy than the ad hoc networks.

6. CONCLUSION

In this paper, we present the Weak State Routing (WSR) protocol, an unstructured forwarding paradigm based on the partial knowledge about the node locations. WSR offers high data packet delivery ratio in large and highly dynamic ad-hoc networks without incurring extensive overhead as flooding based routing mechanism do. The intermediate nodes forward data packets in the absence of location service. The nodes periodically announce their locations on random directions. The nodes receiving these announcements create ID-to-location mappings and combine it with the information about the nodes that are proximate to the announcing node. A routing state is composed of a weak Bloom filter (WBF) which contains a set of nodes and a geographical region where the nodes are believed to be in. WBF also yields the strength of the belief that a node in WBF is in this region. When a node has a data packet, the packet is sent in a random direction hoping that it will rendezvous with an intermediate node that maintains a state information about the destination node. The packet is then biased toward the geographical region that this state directs using a geographical forwarding mechanism such as GPSR. Our biasing strategy is similar to longest prefix match method. In our case, the packet is biased by the strongest mapping. Our weak state mechanism provides a distributed hashing functionality similar to Distributed Hash Tables (DHTs). However, it does not rely on a structured overlay network. Hence, it can tolerate more dynamism.

WSR provides high data packet delivery (at least 98%), with overhead increasing linearly with the number of nodes N and the total number of mappings scaling as $\Theta(N^{3/2})$. We also showed that increasing node speed up to 20 m/s does not significantly affect the performance. WSR enjoys these benefits at the cost of increased path length. Especially for very large scale networks (with nodes more than 700), our simulation results show that WSR significantly outperforms DSR and GLS with GPSR retaining high reachability, low overhead and delay but with higher number hops to reach the destination. Even though GLS stores less number of entries in its location database, the number of states WSR stores has good scalability properties.

Acknowledgement

This material is based upon work supported by the National Science Foundation under Grant No. 0546402 and 0627039. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

7. REFERENCES

[1] Hls patch for ns-2.29. http://www.cn.uni-duesseldorf.de/staff/kiess/software/hls-ns2-patch.

[2] Network simulator. ns-2. http://www.isi.edu/nsnam/ns.

[3] BISNIK, N., AND ABOUZEID, A. A. Capacity deficit in mobile wireless ad hoc networks due to geographic routing overheads. In *26th IEEE International Conference on Computer Communications (INFOCOM'07).* (2007), pp. 517–525.

[4] BLOOM, B. H. Space/time trade-offs in hash coding with allowable errors. *Communications of the ACM 13*, 7 (1970), 422–426.

[5] CAESAR, M., CASTRO, M., NIGHTINGALE, E. B., O'SHEA, G., AND ROWSTRON, A. Virtual ring routing: Network routing inspired by dhts. *Computer Communication Review 36*, 4 (2006), 351 – 362.

[6] CHAWATHE, Y., RATNASAMY, S., BRESLAU, L., LANHAM, N., AND SHENKER, S. Making gnutella-like p2p systems scalable. *Computer Communication Review 33*, 4 (2003), 407 – 418.

[7] CHENG, B.-N., YUKSEL, M., AND KALYANARAMAN, S. Orthogonal rendezvous routing protocol for wireless mesh networks. *IEEE International Conference on Network Protocols (ICNP)* (2006).

[8] DUBOIS-FERRIERE, H., GROSSGLAUSER, M., AND VETTERLI, M. Age matters: Efficient route discovery in mobile ad hoc networks using encounter ages. *Proceedings of the International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)* (2003), 257 – 266.

[9] GKANTSIDIS, C., MIHAIL, M., AND SABERI, A. Random walks in peer-to-peer networks. *Proceedings - IEEE INFOCOM 1* (2004), 120 – 130.

[10] GROSSGLAUSER, M., AND TSE, D. N. Mobility increases the capacity of ad hoc wireless networks. *IEEE/ACM Transactions on Networking 10*, 4 (2002), 477 – 486.

[11] GROSSGLAUSER, M., AND VETTERLI, M. Locating mobile nodes with ease: Learning efficient routes from encounter histories alone. *IEEE/ACM Transactions on Networking 14*, 3 (2006), 457 – 469.

[12] JAIN, S., FALL, K., AND PATRA, R. Routing in a delay tolerant network. vol. 34, pp. 145 – 157.

[13] JOHNSON, D. B., AND MALTZ, D. A. Dynamic source routing in ad hoc wireless networks. In *Mobile Computing*, T. Imielinski and H. Korth, Eds., vol. 353. Kluwer Academic Publishers, 1996.

[14] KARGER, D., LEHMAN, E., LEIGHTON, T., LEVINE, M., LEWIN, D., AND PANIGRAHY, R. Consistent hashing and random trees: Distributed caching protocols for relieving hot spots on the world wide web. *Conference Proceedings of the Annual ACM Symposium on Theory of Computing* (1997), 654 – 663.

[15] KARP, B., AND KUNG, H. Gpsr: Greedy perimeter stateless routing for wireless networks. *Proceedings of the Annual International Conference on Mobile Computing and Networking, MOBICOM* (2000), 243 – 254.

[16] KIESS, W., FÜSSLER, H., WIDMER, J., AND MAUVE, M. Hierarchical location service for mobile ad-hoc networks. *ACM SIGMOBILE Mobile Computing and Communications Review (MC2R) 8*, 4 (2004), 47–58.

[17] KUMAR, A., XU, J., AND ZEGURA, E. W. Efficient and scalable query routing for unstructured peer-to-peer networks. *Proceedings of IEEE INFOCOM 2* (2005), 1162 – 1173.

[18] LI, J., JANNOTTI, J., DE COUTO, D., KARGER, D., AND MORRIS, R. A scalable location service for geographic ad hoc routing. *Proceedings of the Sixth Annual International Conference on Mobile Computing and Networking(Mobicom)* (2000), 120 – 30.

[19] LV, Q., CAO, P., COHEN, E., LI, K., AND SHENKER, S. Search and replication in unstructured peer-to-peer networks. *Proceedings of the International Conference on Supercomputing*, (2002), 84 – 95.

[20] PERKINS, C. E., AND BHAGWAT, P. Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers. *Computer Communications Review 24*, 4 (1994), 234 –.

[21] RATNASAMY, S., FRANCIS, P., HANDLEY, M., KARP, R., AND SHENKER, S. A scalable content-addressable network. *Computer Communication Review 31*, 4 (2001), 161 – 172.

[22] RHEA, S. C., AND KUBIATOWICZ, J. Probabilistic location and routing. *Proceedings - IEEE INFOCOM 3* (2002), 1248 – 1257.

[23] SANTIVANEZ, C., AND RAMANATHAN, R. Hazy sighted link state (hsls) routing: A scalable link state algorithm. Tech. Rep. BBN-TM-1301, BBN Technologies, Cambridge, MA, 2001.

[24] SPYROPOULOS, T., PSOUNIS, K., AND RAGHAVENDRA, C. S. Spray and wait: An efficient routing scheme for intermittently connected mobile networks. pp. 252 – 259.

[25] STOICA, I., MORRIS, R., KARGER, D., KAASHOEK, M., AND BALAKRISHNAN, H. Chord: A scalable peer-to-peer lookup service for internet applications. *Computer Communication Review 31*, 4 (2001), 149 – 160.

[26] WANG, Y., JAIN, S., MARTONOSI, M., AND FALL, K. Erasure-coding based routing for opportunistic networks. pp. 229 – 236.

[27] YOON, J., LIU, M., AND NOBLE, B. Random waypoint considered harmful. *Proceedings - IEEE INFOCOM 2* (2003), 1312 – 1321.