Scalable OSPF Updates for MANETs

Jangeun Jun and Mihail L. Sichitiu
Department of Electrical and Computer Engineering
North Carolina State University
Raleigh, NC 27695
email:{jjun,mlsichit}@ncsu.edu

Abstract— **Recently, mobile ad hoc networks (MANETs) have evolved as an irreplaceable networking technology for situations with sparse or inexistent infrastructure. Adapting OSPFv3 to MANET environments has several advantages. The protocol has proven its maturity in the wired Internet and is now the de facto standard Intra-AS routing protocol. When used in wireless ad hoc networks with proper extensions, OSPFv3 can provide better interoperation between the ad hoc domain and the Internet. This is one of the essential requirements for any ad hoc networks (including wireless mesh networks) that require connectivity to the Internet. The main challenge is that since OSPF is highly optimized for wired-oriented environment, it needs further enhancement to be fully functional in ad hoc scenarios. The most serious obstacle is the large routing overhead associated with the frequent topology changes due to volatile wireless links and node mobility. Various overhead reduction schemes have been proposed by focusing on how to reduce hello packets, flooding nodes, or the number of adjacencies. We propose a completely different approach by taking into account the very essential characteristics of wireless ad hoc networks - namely, distance effect. We implemented the proposed scheme on a simulator that uses the real OSPFv3 as its routing module. The simulation experiments prove that the proposed scheme allows significant reduction in OSPF routing overhead at small loss of route optimality.**

I. INTRODUCTION TO SCALABLE UPDATES IN MANETs

The main characteristic of ad-hoc networks is the absence of pre-planning. The topology of the network is discovered on the fly, after the network's deployment. Thus, such a network must exchange a number of messages which are used to "set-up" various parameters in the network. Example of such parameters are the very existence of other nodes in the network, their position, information about their neighbors, what they offer (e.g., local maps, files, printing facilities). The location information and the neighbors will change with mobility, and thus will have to be updated. As the number of nodes and the mobility increases, the updates will start to be a significant percentage of the total traffic in the network, and at some point, the network will not be able to carry all the updates, let alone the "useful" user traffic.

Information about other nodes' neighbors and/or their location (as well as distance vector or link state information) is especially useful in routing. On one hand routing is essential for the functioning of the ad-hoc network, so one cannot do without it. On the other hand the updates consume available

bandwidth in the ad-hoc network. The bandwidth consumed by the control overhead reduces the bandwidth for user applications. This leads to a significant problem in typical MANETs where bandwidth and processing power are scarce resources.

Various schemes have been proposed to alleviate the overhead problem in MANETs. Fisheye State Routing (FSR) [1] is a link state routing protocol for MANETs that attempts to address the scalability problem. The main idea of FSR is that the updates corresponding to closer nodes propagate more frequently. Since the topology map maintained at each node relies on periodic link state packets that are not flooded as in conventional protocols, it is expected to require less overhead than other link state protocols with full flooding. OLSR [2] is another link state routing protocol for MANETs that reduces the overhead by reducing the number of nodes participating in link state flooding. Some protocols (e.g., [3] [4]) rely on clustering scheme to contain the overhead within a certain boundary. DSR [5] uses aggressive caching to reduce the flooding. Location information is often incorporated into routing protocols to reduce the overhead and increase the scalability. Some examples are LAR [6], DREAM [7] and ZRP [8]. Existing solutions show improvements in flooding overhead using different approaches. However, they require additional protocol complexity (e.g., formation and maintenance of routing hierarchy), storage (e.g., caching) or information (e.g., location).

We aim at providing a distributed, efficient and scalable scheme that can be either applied to existing protocols as an extension or implemented as a stand-alone protocol. The main characteristics of the proposed scheme can be described as:

- It does not rely on any external information other than simple network layer information readily available (e.g., hop count).
- It is fully distributed and simple enough to require no central control of the topology.
- It has the analytical property that the overhead asymptotically increases as $O(1)$ with the scaled network size.
- It is not protocol-specific and thus, it can be easily applied to almost all the classes of existing ad hoc routing protocols that employ flooding as its main or supplementary mechanism.

OSPF [9] is by far the most popular intra autonomous system routing protocol in the Internet. In the latest version (OSPF for IPv6 [10]), the protocol was dissociated from the
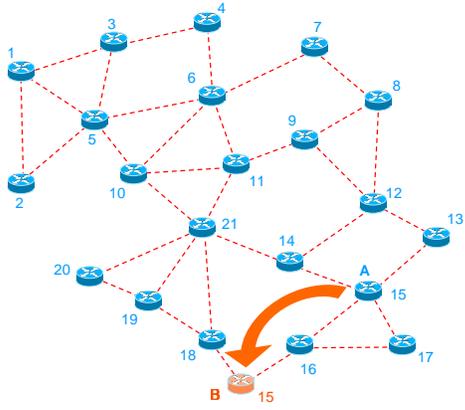
Fig. 1. When node 15 moves from point A to point B, it must send updates to other nodes such that those nodes will be able to reach it.

```
char from_node[n] = {'y','y','y',.....,'y'};

proces_update (received_update)
{
    if (from_node[received_update->source] == 'y')
    {
        from_node[received_update->source] == 'n';
        forward (received_update);
    }
    else
    {
        from_node[received_update->source] ← 'y';
    }
}
```

Fig. 2. Pseudo-code for processing the received updates.

Fig. 3. A regular chain topology chosen for the analysis of asymptotic efficiency.

addressing scheme. This change allows the protocol to work not only with IPv6 addresses but with more general addressing schemes. Adapting OSPFv3 to MANET environments has several advantages. The protocol is mature in the wired Internet and is now the de facto standard. When used in wireless ad hoc networks with proper extensions, OSPFv3 can provide better interoperation between the ad hoc domain and the Internet. The most serious obstacle in using OSPF in MANETs is the large flooding overhead associated with the frequent topology changes. We propose to use the scheme described above to alleviate this problem.

II. PROPOSED SOLUTION

In many MANET protocols, the updates are done by flooding [2], [11]. This means, that when a node needs to update the others it will send (i.e., locally broadcast) the update to all its neighbors, and the neighbors to their neighbors, and so on. In this way, the update will eventually reach all nodes.

Assume that in Fig. 1, node 1 wants to send a message to node 15. When node 15 is in position A the messages from 1 to 15 will likely use the following route: 1-5-10-21-14-15. When node 15 moves to position B the route will likely be: 1-5-10-21-18-15. The first hop of node 1 is node 5 regardless of the position of node 15. Thus, we can say that 15's update going all the way to 1 is "useless". On the other hand, the route from node 12 to node 15 changes substantially when node 15 moves. The general rule is that the closer neighbors should be updated more often than the nodes which are far away. The idea is present in other papers [7], [1] but the implications were not fully explored.

A different way to update is not to forward all the updates, but only a few of them. For example, a node may forward only half of the updates it receives. Figure 2 depicts the pseudo-code of the routine processing the received updates. Under this policy, the nodes one hop away from the node sending the updates will get all updates, the nodes two hops away will get half of the updates, the nodes three hops away will get a quarter of the updates, etc. A node h hops away from the source of the updates will get $\frac{1}{2^{h-1}}$ of the updates.

A. Analysis

In this section, the asymptotic efficiency of the proposed scheme is presented. Two regular topologies are considered in this analysis. We first define some important variables used commonly in the analysis of the two cases as follows:

- $k \equiv$ the total number of updates generated at the center node
- $i \equiv$ the distance (i.e., number of hops) between the center node and another node in the network
- $N(i,k) \equiv$ the total number of updates (including both the generated updates at the center node and the relayed updates by other nodes) as a function of i and k
- $R(i) \equiv$ the update generation rate induced by the center node in the network of diameter i, $R(i) = \frac{N(i,k)}{k}$
- $\beta \equiv$ the update relaying factor ($0 < \beta < 1$) at each node, e.g., one out of every $\frac{1}{\beta}$ arriving updates is relayed in exponential reduction

We investigate the overhead reduction efficiency for the chain topology and then extend the method to the grid topology. Figure 3 illustrates the chain topology under consideration. We focus on the effect of the updates generated at the node located in the center of the network, namely the center node. Assuming full flooding scheme is being used, we can express $N(i,k)$ and $R(i)$ as:

$$N_{flooding}(i,k) = 2k + ... + 2k = 2ik, \qquad (1)$$

$$R_{flooding}(i) = \frac{2ik}{k} = 2i = \Theta(i). \qquad (2)$$

If the proposed scheme is applied to the same scenario, the resulting $N(i,k)$ and $R(i)$ will become:

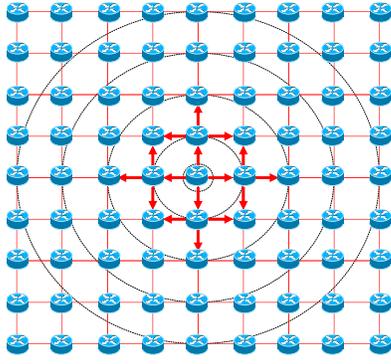$$N_{new}(i,k) = 2k \sum_{l=1}^{i} \beta^l = 2k \left[\frac{\beta(1-\beta^i)}{1-\beta} \right] \qquad (3)$$

Fig. 4. A regular grid topology chosen for the analysis of asymptotic efficiency.

Fig. 5. Overhead reduction behavior can be customized by configuring prescaler parameters.

$$R_{new}(i) = \frac{2\beta}{1-\beta}\left(1 - \beta^i\right) = \Theta(1). \tag{4}$$

The results from (2) and (4) indicate that the asymptotic update generation of the new scheme provides significant enhancement in scalability over the full flooding scheme.

Figure 4 illustrates the grid topology. We investigate the effect of the updates generated at the center node using the same variables used in the chain topology case. With full flooding scheme, $N(i,k)$ and $R(i)$ are:

$$N_{flooding}(i,k) = \left[\left(8\sum_{l=1}^{i} l\right) - 4i\right]k = 4ki^2, \tag{5}$$

$$R_{flooding}(i) = \frac{4ki^2}{k} = \Theta(i^2). \tag{6}$$

If the proposed scheme is used, the resulting $N(i,k)$ and $R(i)$ will become:

$$
\begin{aligned}
N_{new}(i,k) &= 4k\sum_{l=1}^{i} l\beta^l \\
&= \frac{4k\beta}{(\beta-1)^2}\left[(\beta i - i - 1)\beta^i + 1\right]
\end{aligned} \tag{7}
$$

$$R_{new}(i) = \frac{N_{new}(i,k)}{k} = \Theta(1). \tag{8}$$

The results from the grid topology is consistent with the chain topology. Equations (6) and (8) show that the asymptotic update generation of the new scheme has significantly reduced overhead compared to the full flooding scheme. Note that the result holds for any β that satisfies $0 < \beta < 1$. Therefore, β can be varied depending on the performance requirement.

B. Parameters

The proposed scheme provides great flexibility because it turns into virtually infinite number of variations by simply changing a few parameters.

The scheme used for analytical result in Section II-A is a specific case where the number of relayed updates are exponentially reduced as they propagate from the center of
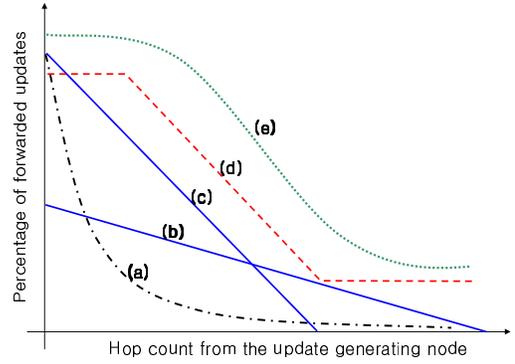
the generating node. We define "prescaler" as the entity that performs the functionality of overhead reduction and β as the prime parameter that determines the intensity of reduction. Note that the prescaler need not always be exponential, rather, it can be hyperbolic or linear depending on the design of a certain routing protocol. Thus, the prescaler's "mode" is another parameter that can be used to customize the reduction behavior.

The two parameters of mode and β affect the functionality of the prescaler in a deterministic manner. That is, when a node receives an update, it determines whether or not to relay it based on a predefined schedule (e.g., relay two for every five received updates). On the other hand, the decision need not be hard fixed. Sometimes it is complicated to come up with a clear cut schedule for a given overhead reduction goal. In such a case, one can employ a scheme that makes the relaying decision based on a random function with parameter β. This provides another prescaler parameter that allows one to fine tune the reduction operation.

In many cases, the closest neighbors (e.g., within one or two hop distance) of a node require highly accurate update information. One example is that a node moves away from its neighbor and the neighbor does not realize the lost link until the update arrives detouring the previous link. This problem can be solved if the detouring update travels fast enough up to a certain distance. Therefore, it is useful to allow updates to be flooded to a predefined distance boundary. We define such a boundary as "inner flooding bound". The prescaler comes into play once the update travels beyond the inner flooding bound.

On the other hand, if a prescaler mode is used with a small β for a network with large diameter, the nodes far from the generating node will barely receive an update. Note that if the overhead is exponentially reduced with the distance, the time interval between updates is exponentially increased. Therefore, there has to be a boundary beyond which the prescaler decides to relay all the received updates. We define such a boundary as "outer flooding bound".

We have defined and discussed a few parameters that can be used for customizing the behavior of the prescaler. Following

summarizes the parameters with brief descriptions:

- **Prescaler mode:** one of exponential, hyperbolic, discrete, linear, or any other overhead reduction scheme.
- **Prescaling intensity** $(= \beta)$**:** defines the reduction intensity with which a prescaler operates.
- **Stateful relay decision:** deterministically relays update based on the state (i.e., reception counter) per destination.
- **Stateless relay decision:** relays update based on a random number generator with parameter β.
- **Inner flooding bound:** defines the distance range within which updates are unconditionally relayed.
- **Outer flooding bound:** defines the distance range beyond which updates are unconditionally relayed.

Figure 5 shows some examples of the configuration of these parameters. The x-axis of the graph represents the number of hops between the update generating (i.e., originating) node and other nodes in the network that receives the update. The y-axis represents the percentage of updates a node transmits because of the originating node. Curves (a) and (e) show the overhead reduction trend obtained by setting the prescaler in exponential or hyperbolic mode with a certain β. In curve (a), the inner flooding bound is one hop and the outer flooding bound is infinity. Curves (b), (c), and (d) show the case of linear prescaler mode with different slopes (determined by β). Curves (d) and (e) show the effect of inner and outer flooding bounds. Smooth inflection in curve (e) indicates stateless (i.e., randomized) relay decision.

The proposed scheme assumes that all the nodes in the network are initially updated before the topology starts to change with mobility or node failure. In real situation, each node joins the network at different time which invalidates the assumption. This can be solved by setting the prescaler to always relay the initial updates it receives. Thus, the initial update of a node (no matter when it joins the network) will be flooded throughout the network and all the other nodes will receive the update.

C. Trade-offs

In this section, the trade-off between the overhead efficiency and route optimality is discussed. Because of the reduction in updates, it is possible that some routes become stale or suboptimal. If the update arrival interval at a receiving node is larger than the update change interval at the originating node, the information stored at the receiving node may become more and more incorrect until the correct update arrives. Although the node has incorrect information, the user traffic forwarded by the node may follow the correct path as it travels towards the final destination.

To visualize this phenomenon, a simple distance vector routing protocol with the proposed scheme is implemented in QualNet simulator [12]. We also implemented an animation tool in MATLAB that helps visualizing the dynamic change in topology and node movement based on the trace file generated by the simulator. The resulting route topology and the node position is shown in Fig. 6.
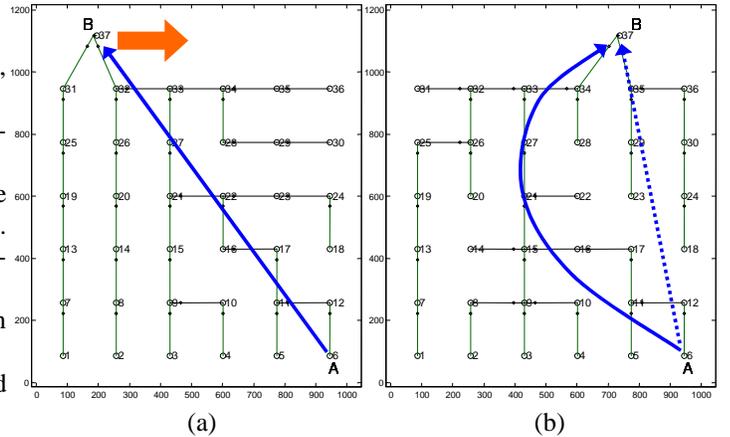

Fig. 6. Two topologies showing (a) optimal route paths in steady state convergence, and (b) transient suboptimal route paths due to overhead reduction and mobility.

For clarity, all the nodes are assumed to be stationary except for node B that moves from left to right as indicated by the block arrow. All the fixed nodes are deployed in a grid to form a 4-regular graph. The route topology before and after the movement are shown in Fig. 6-(a) and (b), respectively. To avoid cluttering the figure, only the route vectors for the mobile node are presented. Thus, each solid line attached to every node in the figure is pointing at its next hop neighbor towards the destination node B. Figure 6-(a) shows that in the converged steady state, all the route vectors are optimal. The distance of multihop paths from every fixed node to node B is minimal. For example, node A reaches node B in ten hops. In contrast, Fig. 6-(b) shows transient suboptimal routes where node B reaches node A in ten hops (following the curved arrow) while the shortest path should be seven hops (following the dotted arrow).

### III. APPLICATION TO OSPFv3 PROTOCOL

This section describes how the proposed overhead reduction scheme can be applied to OSPFv3 implementation.

The operation of OSPF protocol comprises three sub-protocols. First, hello protocol is used to establish or tear down adjacencies when an OSPF router arrives at or disappears from the network. Second, when building adjacencies, synchronization protocol makes sure that two new neighbors commonly share the most up-to-date link state database. Third, flooding protocol ensures that any change in the link state is immediately propagated throughout the network. Although OSPF supports hierarchical topologies with multiple areas, we assume single area scenario because the areas in legacy OSPF are manually configured and no dynamic OSPF area formation scheme is known to the best of authors' knowledge.

Among three sub-protocols, we focus on the link state flooding protocol that produces significant overhead in maintaining the topology under MANET environment. Link state update (LSU) packets are used in OSPF flooding, and one LSU packet may contain multiple link state advertisements (LSAs). When an OSPF router receives LSAs in a flooded LSU packet, it
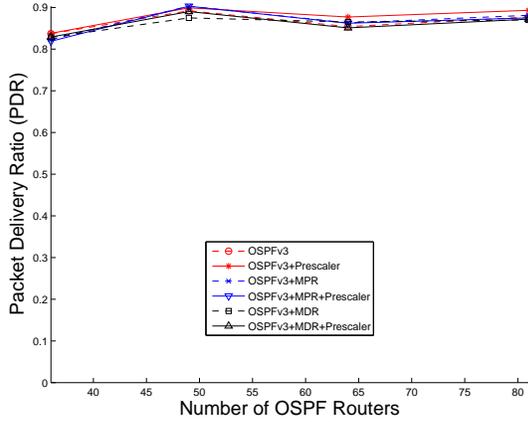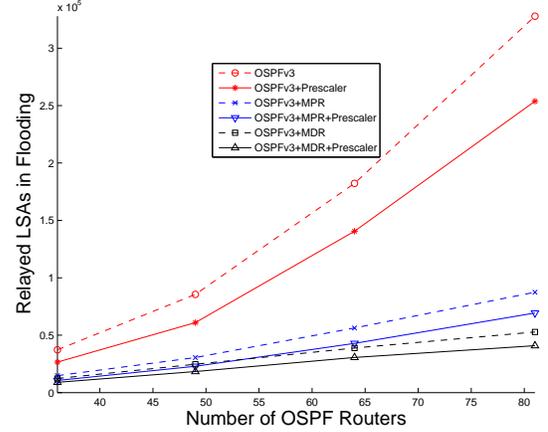
Fig. 7. The performance of different versions of OSPFv3 under low mobility: (a) packet delivery ratio and (b) number of relayed LSAs for flooding.


```
AdvRtrState[n] = {0,0,...,0}; // State per advertising router

Process-LSA (Received-LSA, SPF-Table) {
   AdvRtr ← Get-Advertising-Router (Received-LSA);
   If (AdvRtr is new) Then Reflood;
   Dist ← Get-Distance (AdvRtr, SPF-table);
   Else If (Dist ≤ FloodingInner) Then Reflood;
   Else If (Dist ≥ FloodingOuter) Then Reflood;
   Else If (FloodingMode == "Exponential & Stateful") {
      If (AdvRtrState[AdvRtr] == 0) Then Reflood;
      Else AdvRtrState[AdvRtr]←AdvRtrState[AdvRtr]+1)%(1/Beta);
   }
   Else If (FloodingMode == "Exponential & Stateless") {
      If (UniformRNG(0~1) ≤ Beta) Then Reflood;
   }
   Else If (FloodingMode == "Hyperbolic & Stateless") {
      Dist ← Max(0, Dist - FloodingInner);
      If (UniformRNG(0~1) ≤ (Dist/(Dist+1))^Beta) Then Reflood;
   }
   Else If (FloodingMode == "Linear & Stateless") {
      Dist ← Max(1, Dist - FloodingInner);
      If (UniformRNG(0~1) ≤ (Beta✖10-Dist)/(Beta✖10-Dist+1))
         Then Reflood;
   }
}
```

Fig. 8. Pseudo-code for overhead reduction scheme applied to OSPF.

first updates its own link state database and then, determines whether or not to relay (i.e., reflood) each LSA to different neighbors. In fact, the updates are multicasted, but reflooding is made reliable by scheduling retransmission only for the intended neighbors. The overhead reduction scheme should be inserted between these two steps. Among different types of LSAs, our scheme deals with only router-LSAs because all the LSAs in MANETs are router-LSAs.

Figure 8 shows the pseudo code with four prescaler modes that can be applied to the OSPFv3 protocol. Unlike a distance vector protocol where each update element corresponds to a destination, a flooded update in OSPF is associated with the router that advertised (i.e., originated) the LSA. In addition to prescaler parameters, three pieces of information are required in making reflooding decision: advertising router (AdvRtr), hop distance (Dist) between the LSA receiver and AdvRtr, and state of the advertising router (AdvRtrState). The third variable is used only when the prescaler is in a stateful mode. Note that AdvRtr and Dist are readily available from the received

LSA and SPF table, respectively. Thus, the scheme requires no extra control overhead.

When LSAs (in a LSU packet) arrive and if the prescaler is enabled, the procedure "Process-LSA" is called for each LSA. If the LSA is advertised by an unknown router indicating it is the initial update, it gets reflooded unconditionally. Otherwise, if the advertiser is within the inner flooding bound (FloodingInner) or beyond the outer flooding bound (FloodingOuter), it gets reflooded. For other cases, a uniform random number generator (UniformRNG) is used for a stateless mode and state variable per advertising router (AdvRtrState) is used for a stateful mode. In hyperbolic and linear modes, generated random numbers are compared against the value calculated from the variables Dist and Beta to produce desired behavior.

IV. PERFORMANCE EVALUATION

We implemented the prescaler described in Section III on a real OSPFv3 code ported from the Quagga routing software suite [13]. The experiments were performed using a unique simulator [14] that runs the Quagga OSPFv3 code on GTNetS simulator [15]. Implementing the prescaler on the Quagga software allows the proposed scheme to be quickly ported to and deployed on real OSPF routers. On the other hand, using GTNetS allows simulation experiments for large networks.

Six different versions of OSPFv3 protocol models are used for performance evaluation. Baseline OSPFv3, OSPFv3 with Multi-Point Relays (MPR) extension [16], and OSPFv3 with MANET Designated Routers (MDR) extension [17] are used with and without prescaler capability. Both MPR and MDR commonly attempt to solve the overhead problem [18] by reducing the number of the OSPF routers participating in flooding, but they differ in the algorithm that selects relays (i.e., flooding nodes). Their additional overhead reduction features such as differential hellos and smart peering are disabled for fair comparison of flooding suppression performance.

At each node, wireless signal range is set to 250 m and IEEE 802.11b is used for the physical and the MAC layers. Initially, nodes are randomly deployed in 1000×1000 m^2
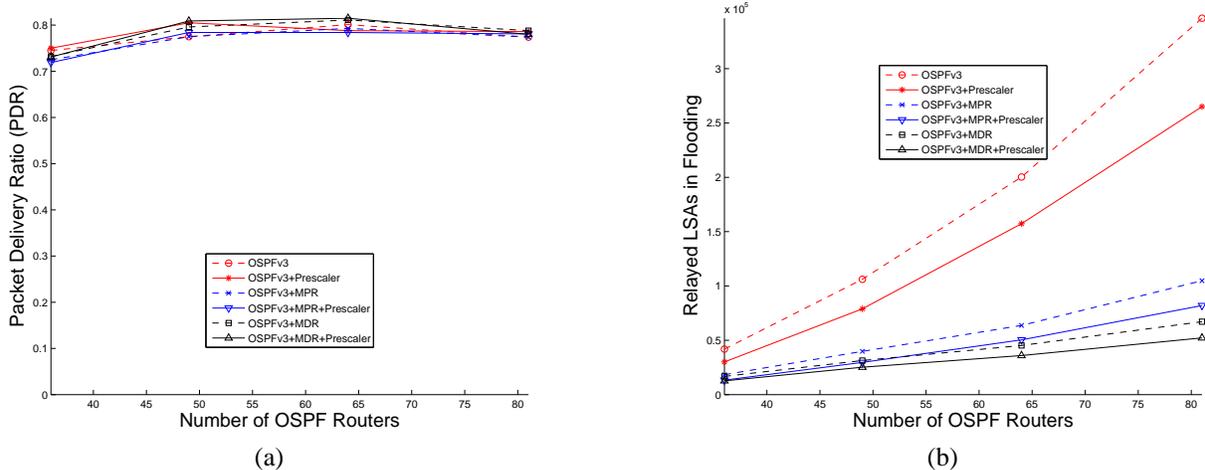
Fig. 9. The performance of different versions of OSPFv3 under high mobility: (a) packet delivery ratio and (b) number of relayed LSAs for flooding.

area. Random Waypoint model is used with pause time 40 s. Simulations are run for 600 s for both low and high mobility scenarios. The maximum node velocity is 5 m/s for the low mobility and 10 m/s for the high mobility scenario, respectively. In each case, the number of nodes is incremented from 36 to 81. CBR traffic is offered to probe route correctness. Overall traffic load offered to the network is 5 p/s between randomly chosen pairs of sources and destinations, and packet size is 40 B. As performance metrics, we measure both packet delivery ratio (PDR) and the total number of relayed LSAs.

Figure 7-(a) and (b) show the PDR and flooding performance results of six protocols for low mobility and Fig. 9-(a) and (b) for high mobility. While maintaining almost the same PDR as shown in Fig. 7-(a) and Fig. 9-(a), the proposed scheme shows significant reduction in the number of flooded LSAs as shown in Fig. 7-(b) and Fig. 9-(b). The flooding reduction is around 30 % for the network of 36 nodes. The amount of reduction grows as the number of nodes increases, but reduction ratio stays between 20 % and 25 % for the large network with 81 nodes. Note that our scheme almost equally affects all the three versions of OSPFv3. This is because MPR and MDR functionalities are independent of our scheme. While their schemes reduce flooding by selecting a subset of nodes that participate in flooding (where updates are still flooded throughout the network), ours reduce flooding by attenuation of flooded updates as they travel far from the source of change.

V. CONCLUSION

A distributed, efficient and scalable update scheme is proposed to alleviate the routing overhead problem in MANETs. The proposed scheme uses hop count distance to effectively reduce routing updates. The scheme does not require the formation of any hierarchy or externally obtained information. Asymptotically, it enables $O(1)$ overhead growth as the network is scaled. With various parameters that determine the

behavior of overhead reduction, the scheme can be customized and applied to different classes of existing ad hoc routing protocols that rely on flooding. The simulation experiments show that the scheme provides significant flooding overhead reduction in all the scenarios investigated.

REFERENCES

[1] M. G. G. Pei and T. Chen, "Fisheye state routing: A routing scheme for ad hoc wireless networks," in *Proc. of ICC 2000*.
[2] T. Clausen and P. Jacquet, "Optimized link state routing protocol (OLSR)." RFC 3626, Oct. 2003.
[3] C. Chiang, H. K. Wu, W. Liu, and M. Gerla, "Routing in clustered multihop mobile wireless networks with fading channel," in *Proc. of IEEE Singapore International Conference on Networks*, 1997.
[4] K. Xu and M. Gerla, "A heterogeneous routing protocol based on a new stable clustering scheme," in *Proc. of the Military Communications Conference (MILCOM)*, (Anaheim, CA), Oct. 2002.
[5] D. B. Johnson and D. A. Maltz, "Dynamic source routing in ad hoc wireless networks," in *Mobile Computing* (Imielinski and Korth, eds.), vol. 353, Kluwer Academic Publishers, 1996.
[6] Y. B. Ko and N. H. Vaidya, "Location aided routing (LAR) in mobile ad hoc networks," *Wireless Networks*, vol. 6, pp. 307–321, Sept. 2000.
[7] S. Basagni, I. Chlamtac, V. Syrotiuk, and B. Woodward, "A distance routing effect algorithm for mobility (DREAM)," in *Proc. of ACM Mobicom'98*, (Dallas, TX), pp. 76–84, Oct. 1998.
[8] Z. J. Haas and M. R. Pearlman, "The zone routing protocol (zrp) for ad hoc networks," tech. rep., IETF MANET Working Group.
[9] J. Moy, "OSPF version 2." RFC 2328, Apr. 1998.
[10] R. Coltun, D. Ferguson, and J. Moy, "OSPF for IPv6." RFC 2740.
[11] C. Bhagwat, "Highly dynamic destination-sequenced distance vector routing (DSDV) for mobile computers,"
[12] Scalable Network Technologies, Inc., "QualNet User's Manual, version 3.8." http://www.scalable-networks.com/, 2003.
[13] "GNU quagga routing software." http://www.quagga.net/.
[14] T. R. Henderson, P. A. Spagnolo, and G. Pei, "Evaluation of OSPF MANET extensions."
[15] G. F. Riley, "Large-scale network simulations with GTNetS," in *Proceedings of the 2003 Winter Simulation Conference*.
[16] M. Chandra, "Extensions to OSPF to support mobile ad hoc networking." Internet Draft draft-chandra-ospf-manet-ext-03, Apr 2005.
[17] R. Ogier and P. Spagnolo, "MANET extensions of OSPF using CDS flooding." Internet Draft draft-ogier-manet-ospf-extension-04, Jul 2005.
[18] F. Baker, M. Chandra, R. White, J. Macker, T. Henderson, and E. Baccelli, "Problem statement for OSPF extensions for mobile ad hoc routing." draft-baker-manet-ospf-problem-statement-00, Sep. 23 2003.