Analysing the Impact of Topology Update Strategies on the Performance of a Proactive MANET Routing Protocol

Yangcheng Huang
University College London
Dept of Computer Science
London WC1E 6BT, UK
yangchenghuang@ieee.org

Saleem Bhatti
University of St. Andrews
School of Computer Science
Fife KY16 9SX, UK
saleem@cs.st-andrews.ac.uk

Søren-Aksel Sørensen
University College London
Dept of Computer Science
London WC1E 6BT, UK
s.sorensen@cs.ucl.ac.uk

Abstract

This paper presents an analysis of several topology update strategies for proactive MANET routing protocols. Although there have been a number of performance studies of proactive MANET routing protocols, little attention has been paid to the impacts of topology update strategies on routing performance. The goal of this paper is to better understand how topology update strategies can contribute to topology maintenance in proactive mobile ad hoc networks and thus impact the overall performance. Our contribution includes (1) a quantitative analysis on the impacts of proactive update intervals on the routing performance; (2) evaluating the performance of **reactive** *topology updates and* **proactive** *updates for proactive routing protocols.*

1 Introduction

Each node in a proactive Mobile Ad hoc Network (MANET) maintains routing information to every other node in the network at all times. The routing information can be either topological repositories (such as in OLSR [5]) or distance to other nodes (such as DSDV [8]). Due to frequent topology changes caused by mobility, the routing information in each node has to be updated to reflect any topology changes and maintain the correctness of route selection. This requires the that nodes broadcast topology updates with certain strategies.

Proactive routing protocols like OLSR use periodic updates to maintain the routing information about each node in the network. For example, an OLSR node propagates *topological control (TC)* messages among all the nodes of the network to advertise the link status between itself and its neighbours, as well as *HELLO* messages effectively as keep-alives but also containing routing information. Each node in the OLSR network receives the topology update messages and updates its local state repositories correspondingly.

Despite the simplicity and robustness of such periodic topology update strategy, there have been several concerns about its performance.

- Topology changes may be too dynamic to be captured by periodic updates. Link breakage might have to wait for a period of the topology update intervals before being advertised. In the presence of frequent topology changes, the performance of the periodic updates needs to be re-evaluated.
- Although it is commonly believed that a smaller topology update interval could speed up adaptation to changes, the *quantitative* impacts of update intervals on routing performance is still not clear. Considering the resource constraints of MANETs, the topology update performance needs to be quantified in order to enhance system efficiency and scalability.

While there have been a number of performance studies of proactive MANET routing protocols, little attention has been paid to the impacts of topology update strategies on routing performance. Recent studies on topology update strategies [4] [9] have been focused on overhead reduction. Samar and Haas [9] propose several update strategies to maximum the update period while maintaining the performance and satisfying certain requirements (such as bounded-delay). Clausen applies the concept of fisheye [7] into OLSR [4], to reduce the control overhead by introducing temporal partiality into proactive updates.

In this study, we investigate the impacts of the topology update strategies on the routing performance.

The goal of this paper is to better understand how topology update strategies can contribute to topology maintenance in proactive mobile ad hoc networks and thus impact the overall performance.

Our contributions include:

- a quantitative analysis on the impacts of proactive update intervals on the routing performance. We define *consistency* probabilistically and use it to evaluate the topology update performance. A novel resilience model is presented for periodic topology updates in mobile ad hoc networks.
- evaluation of the performance of *reactive* topology updates and *proactive* updates with a combination of model-based analytical study and a simulation based performance evaluation study.

The rest of the paper is organised as follows. Section 2 discusses the topology update strategies in existing proactive MANET routing protocols. Section 3 presents an analytical study on the performance of the strategies. Section 4 presents our simulation based performance evaluation for OLSR. Conclusions are summarised in section 5.

2 Topology Update Strategies in Proactive MANET Routing Protocols

In this section, we briefly discuss the topology update strategies in the existing MANET routing protocols.

From the scope of the update messages, existing topology update strategies can be categorised into *global updates* and *localised updates.*

Global Updates. Proactive protocols such as OLSR[5] use global topology updates. In those protocols, each node periodically exchanges its topology information with every other node in the network. The disadvantage of global updates is that they consume significant amount of bandwidth.

Localised Updates. To reduce the overheads in topology updates, in the protocols such as DSDV [8] and FSR [7], each node only propagate route updates within a localised region. For example, DSDV nodes maintain the distances to all the other nodes in the network and broadcast periodically such distance information only to its neighbours. The Fisheye State Routing(FSR)[7] adopts the strategy of *temporal partiality*, whereby a node only advertises information about closer nodes and exchanges link state information with its neighbours.

From the triggering mechanism of the topology updates, existing topology update strategies can be categorised into *proactive updates* and *reactive updates.*

Proactive Updates. The routing protocols broadcast topology updates periodically, even without topology changes. Some protocols such as OLSR [5] broadcast the updates with a fixed interval. IARP [6] and the fast-OLSR extension [2] set the update intervals inversely proportional to the maximum velocity of the nodes. TBRPF [1] generates two types of updates: full-topology periodic updates and differential updates. DSDV [8] advertises the routing information periodically and incrementally as topological changes are detected.

Reactive Updates. Traditional link-state routing protocols such as OSPF sends an update when a link becomes invalid or when a new node joins the network. The benefits of this strategy are that, if the network topology or conditions are not changed, no update packets are sent, which eliminates redundant periodic update dissemination into the network; in addition, topology changes can be captured and broadcasted quickly with no delay, which might reduce packet drops because of mobility.

In this study, we investigate the impacts of the topology update strategies on the routing performance. In particular,

- We study the performance of the *proactive update* strategy by tuning topology update intervals and investigating the impacts on the routing throughput.
- We compare the performance of the *proactive update* strategy with that of the *reactive update* strategy. The following two *reactive* update options are proposed.
  - *Localised reactive update (etn1).* We apply the concept of FSR into the reactive topology updates. Whenever a link change is detected, the node sends its topology updates to its neighbours only. Correspondingly, the node has a more accurate state of the nearer nodes than the farther nodes.
  - *Global reactive update (etn2).* Each node engaged in the proactive routing protocols broadcasts topology updates to every other node in the network whenever a link change is detected.

3 Model Based Analytical Study

3.1 Performance Metrics & Assumptions

In the following analytical study, we use *consistency* to evaluate the performance of topology update strategies. In our study, *consistency* is calculated by the probability that, at time t, the state tuple (i.e. neighbour tuple or topology tuple) corresponding to key k (i.e. either node identifier for neighbour tuple, or node pair for topology tuple) is the same at both state installer (a node that receives and uses state information in a routing update) and state holder (the node for which that state information applies). In this paper, we define the consistency metric as follows.

Definition Let $R = \{r_1, r_2, ..., r_K\}$ be a set of routing states of node i. Let $t(r_k)$ be the period during which route state r_k is the same with remote connectivity. The *consistency,* c of R during a period T is the average proportion of $t(r_m)$ over T, where $c = \frac{\sum_{k=1}^{K} t(r_k)}{K*T}$

In order to evaluate the consistency of routing states, we analyse the *state inconsistency time* L, i.e. the period to convergence from the change occurrence (i.e. inconsistency occurs) to the time the nodes in the network update the state repositories (i.e. achieving state consistency again).

Route consistency in proactive protocols is closely correlated with routing performance (i.e. throughput), as without correct routes in place,packets cannot be delivered to their intended destinations. In a proactive routing protocol, packet drops occur when the routing information is inconsistent or there is congestion. So, in this study, the average routing throughput of proactive routing protocols is assumed to be *proportional* to route state consistency, and we do not model congestion.

Without losing generality, we assume that the arrival of a change event (either link changes or route changes) is an independent, identically distributed Poisson process with arrival rate λ. The assumption is reasonable, if the node degree is small and the nodes are moving randomly so that the process of route change is totally random.

3.2 A Probabilistic Topology Update Model

Consider an arbitrary period of time, starting at t_0. Let X be the time of $first$ link change occurrence after t_0. Let r be the topology update interval.

Table 1. Symbols in the Analytical Model

r	topology update interval
λ	topology change rate
L	state inconsistency time
ϕ	state inconsistency ratio
ψ	derivative of ϕ with respect to r

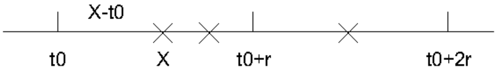


Figure 1. Periodic Topology Update

According to the definition,

$$L = (t_0 + r - X)^+$$

According to the assumption,

$$X - t_0 \sim Exp(\lambda)$$

Therefore, the expected inconsistency time is

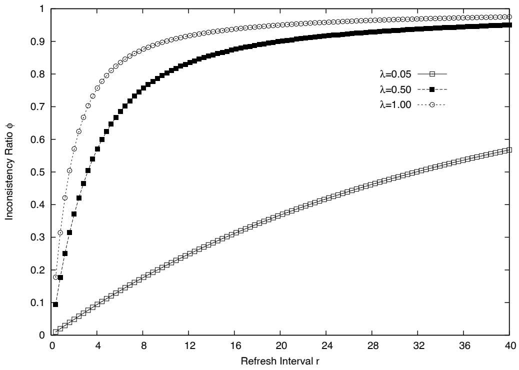
$$\begin{aligned}
E(L) &= E(t_0 + r - X)^+ \\
&= E(r - \gamma)^+ (where \gamma = X - t_0 \sim Exp(\lambda)) \\
&= \int_0^\infty (r - \gamma)^+ \lambda e^{-\lambda\gamma} d\gamma \\
&= \int_0^r (r - \gamma)^+ \lambda e^{-\lambda\gamma} d\gamma \\
&= r + \frac{e^{-r\lambda} - 1}{\lambda} \\
&= \varphi(r, \lambda) \qquad (1)
\end{aligned}$$

The expected inconsistency ratio, which is defined as the fraction of inconsistency time, is

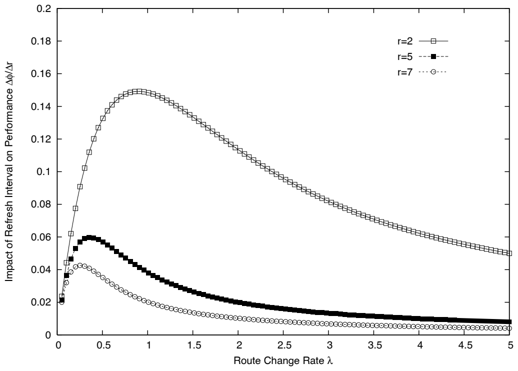
$$\begin{aligned}
\phi(r, \lambda) &= \frac{\varphi(r, \lambda)}{r} \\
&= 1 + \frac{e^{-r\lambda} - 1}{r\lambda} \qquad (2)
\end{aligned}$$

Consider the impacts of topology update interval r on state inconsistency ratio ϕ, i.e. the derivative of ϕ with respect to r ($\frac{d\phi}{dr}$).

$$\begin{aligned}
\psi(r, \lambda) &= \phi'(r) \\
&= \frac{d\phi}{dr} \\
&= \frac{1}{r^2\lambda} - \frac{1 + r\lambda}{r^2\lambda e^{r\lambda}} \qquad (3)
\end{aligned}$$



(a) ϕ vs. r



(b) ψ vs. λ

Figure 2. Impacts of Refresh Interval

3.3 The Impacts of Topology Update Intervals

Based on Equation (2) and (3), we present an analysis on the impacts of topology update intervals on routing performance.

As shown in Fig 2(a), state consistency (i.e. $1 - \phi$) drops as expected with the increase of topology update interval r. However, such impact largely depends on the state change rate λ. For example, when the state change rate is relatively low (i.e. $\lambda = 0.05$), the consistency reduces gradually with the increase of refresh interval; in addition, the maximum inconsistency ratio is moderate, 57% in this case. On the other hand, when the state change rate is relatively high (i.e. $\lambda = 0.5$ or 1), the consistency drops sharply to 20% when the update interval increases from 1s to 4s; after that, the consistency ratio goes smoothly and increasing refresh intervals does not have significant impact on the performance.

Fig 2(b) demonstrates such observations more intuitively. When state change rate λ is relatively high (i.e. $\lambda > 0.25$ when $r = 5s$), with the increase of route change rate λ, the impact of refresh interval r on route consistency drops. Especially with larger refresh intervals (i.e. r = 5s or 7s), update interval has no significant impacts on consistency (i.e. $\frac{d\phi}{dr} < 0.06$).

Table 2. Symbols in Overhead Analysis

HELLO	hello messages in neighbor detection
TC	topology control messages
α_{hello}	control overhead by HELLO messages
α_{tc}	control overhead by TC messages
$\lambda(v)$	link change rate under node velocity v

In summary, the impacts of update intervals on consistency largely depend on state change rate. Under frequent state changes, tuning topology update intervals doesn't have much impact on state consistency.

3.4 Control Overhead Analysis

In this section, we present a brief discussion on the control overhead under different topology update strategies. The symbols used in this section can be found in Table 2.

3.4.1 Control Overhead under *Proactive* Strategy

Let α be the amount of control overhead. Let r be the topology update interval. Then

$$\alpha = \alpha_{hello} + \alpha_{tc}$$

In this study, we keep the HELLO intervals, h, and node density constant. Therefore,

$$\alpha_{tc} \propto \frac{1}{r}$$

The control overhead under *proactive* update strategy can be approximated by,

$$\alpha = \frac{\alpha_0}{r} + c \tag{4}$$

From Equation (4) we can see that, under a *proactive* update strategies, the control overhead has no relationship with network conditions such as node velocity.

3.4.2 Control Overhead under *Reactive* Strategy

The control overhead under a *reactive* update strategy is analysed as follows.

We assume that the link change inter-arrival time distribution can be approximated by an exponential

Table 3. MAC/PHY Layer Configurations

MAC Protocol	IEEE 802.11
Radio Propagation Type	TwoRayGround
Interface Queue Type	DropTailPriQueu
Antenna Model	OmniAntenna
Radio Radius	250m
Channel Capacity	2Mbits
Interface Queue Length	50

distribution with fairly high accuracy [10]. The link change inter-arrival time density function can be expressed as,

$$f(t) = \lambda(v)e^{-\lambda(v)\,t} \tag{5}$$

The average link change interarrival time is $\frac{1}{\lambda(v)}$. Correspondingly, the average reactive update interval is inversely proportional to $\lambda(v)$. Then the control overhead can be approximated by,

$$\alpha = \alpha_0\lambda(v) + c \tag{6}$$

From Equation (6) we can see that, under *reactive* update strategies, the routing overhead increases *linearly* with link change rate. Since the link change rate increases with node velocity, radio range and node density [10], *reactive* update strategies would introduce much more control overhead in high-density, high-mobility networks.

4 Simulation based Performance Evaluation

4.1 Simulation Set-up

We implement the proposed options in the OLSR implementation which runs in version 2.9 of NS2 and uses the ad-hoc networking extensions provided by CMU. The detailed configuration is shown in Table3.

We use a network consisting of n nodes: $n = 20$ to simulate a low-density network, $n = 50$ to simulate a high-density network. The nodes are randomly placed in an area of 1000m by 1000m. All simulations run for 100s.

We use the Random Trip Mobility Model, "a generic mobility model that generalises random waypoint and random walk to realistic scenarios" [3] and performs perfect initialisation. Unlike other random mobility models, Random Trip reaches a steady-state distribution without a long transient phase and there is no need to discard initial sets of observations.

The mean node speed, v, ranges between 1m/s to 30m/s. For example, when the mean node speed is 20m/s the individual node speeds are uniformly distributed between 0m/s and 40m/s. The average node pause time is set to 5s.

A random distributed CBR (Constant Bit Rate) traffic model is used which allows every node in the network to be a potential traffic source and destination. The rate of each CBR traffic is 10kb/s. The CBR packet size is fixed at 512 bytes. There are at least $n/2$ data flows that cover almost every node.

For each sample point presented, 100 random mobility scenarios are generated. The simulation results are thereafter statistically presented with the mean of the metrics and the errors. This reduces the chances that the observations are dominated by a certain scenario which favours one protocol over another.

In each simulation, we measure each CBR flow's *throughput* and *control traffic overhead* and then calculate the mean performance of each metric as the result of the simulation.

Throughput is computed as the amount of data transferred (in bytes) divided by the simulated data transfer time (the time interval from sending the first CBR packet to receiving the last CBR packet).

Control overhead is calculated by summing up the size of all the control packets *received* by each node during the whole simulation period.

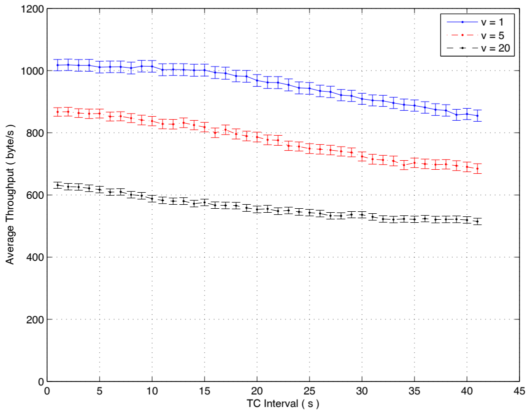
4.2 Observations

In this section, we present the observations on the routing performance under various factors, such as node velocity node density and topology advertisement redundancy options. For each figure, we hold the interval of HELLO messages h (seconds), node number, n, and radio range, rr (metres), constant.

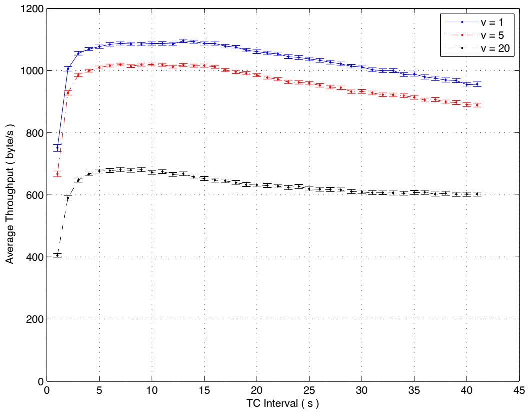
4.2.1 Routing Performance under Proactive Topology Updates

From Fig 3(a) we can see, in a low-density network, tuning topology update intervals within a certain range (e.g. $1 \leq t \leq 20$) has no significant impact on routing throughput; with the increase of topology update intervals, the throughput only drops slightly.

For example, when the node mobility is slow (e.g. $v = 1m/s$), the throughput is almost *constant* when the refresh intervals are between 1 and 10. When the node velocity is moderate (e.g. $v = 5m/s$) or relatively high (e.g. $v = 20m/s$), the increase of refresh intervals from 1 to 10 only leads to less than 5% performance degradation.

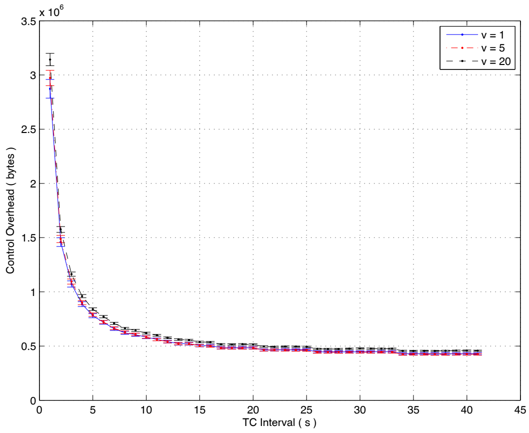


(a) Low Density

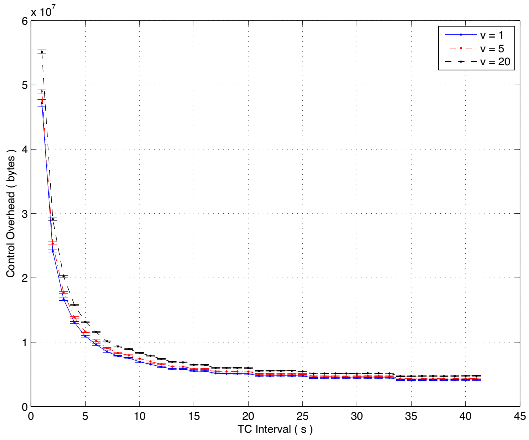


(b) High Density

Figure 3. Throughputs vs. Topology Update Intervals (h=2s rr=250m)



(a) Low Density



(b) High Density

Figure 4. Control Overhead vs Topology Update Intervals (h=2s rr=250m)

In a high-density network, however, our simulation results show the different impacts of topology update intervals on the throughput.

From Fig 3(b), when the refresh intervals are relatively small ($1 \leq t \leq 5$), reducing topology update intervals leads to up to 50% performance degradation; when the refresh intervals are larger than 10s, the throughput drops gradually with the increase of topology update intervals. This can be explained by the fact that small topology update intervals generate a large amount of control packets. This can also be observed in Fig 4(b). The extra control overhead introduced exaggerates channel contention and leads to queue overflow.

From Fig 4, the control overhead is inversely proportional to topology update intervals. This matches Equation (4) well. (From this we can infer that, in presence of network congestion, increasing topology update intervals could lower the control overhead effectively. However, we do not examine, specifically, congestion effects in this study.)

4.2.2 Routing Performance under Reactive Topology Updates

In this section, we compare the routing performance with proactive update and with reactive update strategies. From the figures we can see, among the three topology update options, the soft-state based proactive option outperforms significantly the other two reactive options. Global reactive update option (i.e. $etn2$) performs slightly better than the proactive approach (Fig5), but it introduces three times more control over-

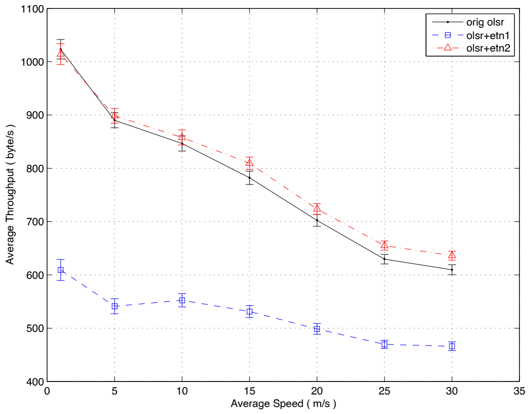


Figure 5. Throughput under Different Topology Update Options (h=2s rr=250m)

head (Fig6). Although the localised reactive update (i.e. $etn1$) introduces much less overhead, its throughput is far from satisfactory.

5 Conclusions

In this study, we present a quantitative analysis of the impacts of topology update strategies on the performance of a proactive MANET routing protocol. Our analytical and simulation results have shown that,

- Reducing topology update intervals has little improvements on the performance of proactive routing protocols, but with a significant increase of control overhead.
- Reactive topology update approach, as adopted in traditional link state routing protocols such as OSPF, doesn't perform as well as proactive topology update approach. Particularly, due to the frequent topology changes, the global reactive update approach introduces too much control overhead. The localised reactive approach, although introducing the least overhead, has the worst packet delivery performance. Therefore, the proactive approach is more suitable for the topology update in MANETs.

These findings do not conflict with [4] and [9]. Instead, our results offer further explanations for previous studies. Because of the insignificant impacts on routing performance, the topology update period can be maximised within a certain range of values, without degrading the performance.

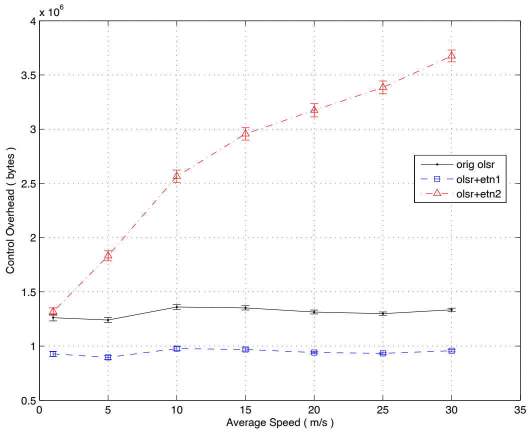


Figure 6. Control Overhead under Different Topology Update Options (h=2s rr=250m)

The motivation of this study is to gain better understanding on how topology update strategies can impact proactive routing performance in dynamic and resource-contrained networks. Because of the wide deployment of the state maintenance processes, the results of this study may provide insightful guidance on designing efficient state maintenance mechanisms for a variety of wireless protocols and applications, and thus exert positive effects on the performance of wireless networks.

The original data, the ns source code and the scripts used in this study are all available from the authors' website (http://www.cs.ucl.ac.uk/staff/y.huang/topo_strategy).

References

[1] B. Bellur and R. G. Ogier. A reliable, efficient topology broadcast protocol for dynamic networks. In *IEEE Infocomm*, pages 178–186. IEEE, Mar. 1999.

[2] M. Benzaid, P. Minet, and K. Agha. Integrating fast mobility in the olsr routing protocol, 2002.

[3] J. Y. L. Boudec and M. Vojnovic. Perfect simulation and stationarity of a class of mobility models. In *Proc. IEEE Infocom Conference*, Miami, USA, 2005.

[4] T. Clausen. Combining temporal and spatial partial topology for. manet routing - merging olsr and fsr. In *WPMC*, 2003.

[5] T. Clausen, P. Jacquet, A. Laouiti, P. Muhlethaler, a. Qayyum, and L. Viennot. Optimized link state routing protocol. In *IEEE INMIC Pakistan*, 2001. Best paper award.

[6] Z. J. Haas and M. R. Pearlman. The performance of query control schemes for the zone routing protocol. *IEEE/ACM Trans. Netw.*, 9(4):427–438, 2001.

[7] G. Pei, M. Gerla, and T.-W. Chen. Fisheye state routing in mobile ad hoc networks. In *ICDCS Workshop on Wireless Networks and Mobile Computing*, Taipei, Taiwan, 2000.

[8] C. E. Perkins and P. Bhagwat. Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers. In *SIGCOMM*, New York, NY, USA, 1994.

[9] P. Samar and Z. Haas. Strategies for broadcasting updates by proactive routing protocols in mobile ad hoc networks. In *MILCOM*, 2002.

[10] P. Samar and S. B. Wicker. On the behavior of communication links of a node in a multi-hop mobile environment. In *MobiHoc*, pages 145–156, 2004.