International Journal of Parallel, Emergent and Distributed Systems
Vol. 23, No. 6, December 2008, 447–460

The impact of topology update strategies on the performance of a proactive MANET routing protocol

Yangcheng Huang[a]*, Saleem Bhatti[b,1] and Søren-Aksel Sørensen[c,2]

[a]*Ericsson Ireland Research Centre, Athlone, Ireland;* [b]*School of Computer Science, University of St Andrews, St Andrews, Fife KY16 9SX, UK;* [c]*Department of Computer Science, University College, London WC1E 6BT, UK*

(*Received 9 May 2007; final version received 6 November 2007*)

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

Keywords: MANET; proactive; routing; topology; OLSR

1. Introduction

Network dynamics and resource constraints are two of the most critical issues in wireless *ad hoc* networks and sensor networks. For example, each node running a *proactive* mobile *ad hoc* network (MANET) routing protocol maintains up-to-date routing information to every other node in the network at all times. This is in comparison to *reactive* routing protocols, in which routes are determined on demand. The routing information can be either topological repositories (such as in optimised link state routing (OLSR) [5]) or distance to other nodes (such as destination sequenced distance vector (DSDV) [11]). Due to the frequent changes caused by mobility, the routing information has to be updated to reflect topology changes and guarantee the correctness of route selection. This requires nodes using a proactive routing protocol to *broadcast* topology updates with optimal strategies. However, the traffic overhead introduced by the topology state advertisements may lead to performance degradation and consume battery power.

Topology control and maintenance algorithms have been proposed to maintain network connectivity while reducing resource consumption (including energy consumption and bandwidth consumption). The key idea of topology control is to *define* the network topology by forming the proper neighbour relationships under certain criteria. The motivations for topology control/maintenance algorithms are to achieve optimisations in resource usage. However, network performance *must* be maintained. For example, nodes with certain energy-efficient topology control algorithms could determine their

*Corresponding author. Email: yangcheng.huang@ericsson.com

ISSN 1744-5760 print/ISSN 1744-5779 online

DOI: 10.1080/17445760801945524
http://www.informaworld.com

transmission power collaboratively, instead of transmitting using the maximal power. This would reduce energy consumption while maintaining network capacity.

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-alive 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 a periodic topology update strategy, there have been several concerns about its impact on 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 (e.g. situations in which there is a high rate of node mobility), 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, this is likely to result in an increase in the level of *control traffic* or *control overhead* – the messages that are not data packets and are generated due to the operation of the routing protocol for its own use. However, the *quantitative* impacts of update intervals on routing performance are still not clear. Considering the resource constraints of MANETs, the performance impact of topology update strategies need to be quantified in order to improve 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 [4,13] have been focused on overhead reduction. Samar and Haas [13] propose several update strategies that calculate the update periods under certain constraints (such as bounded-delay) and under assumptions on the distribution of link change events. The basic motivation is to maximise the update period while maintaining performance. The simulation results show that the proposed schemes, while maintaining satisfactory network performance (e.g. throughput), can lead to more than 45% control traffic reduction. Clausen applies the concept of *fish eye routing* [10] into OLSR [4] and demonstrates, using simulations, a reduction in control overhead by introducing temporal partiality into proactive updates.

In this study, we investigate the impacts of different topology update strategies on the routing performance of OLSR. The goal of this paper is to give better insight into how topology update strategies can contribute to topology maintenance in proactive MANETs and thus assess the likely impact on the overall performance of a MANET network.

Our contributions include:

- A quantitative analysis, through simulations, on the impacts of proactive update intervals on the routing performance of OLSR. We define *consistency* probabilistically and use it to evaluate the topology update performance. A novel resilience model is presented for periodic topology updates in MANETs.
- Evaluation of the impact in performance of *reactive* and *proactive* topology updates with a combination of a model-based analytical study and a simulation based performance evaluation study.

The rest of this paper is organised as follows. We present a discussion on a variety of possible topology update strategies in Section 2, followed by an analytical study on the performance of the strategies in Section 3. The experimental design is illustrated in Section 4, including detailed simulation parameters and metrics. We present our observations based on NS2 simulations in Section 5. We finally summarise our work in Section 6.

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.

2.1 Optimised flooding in OLSR

OLSR uses proactive methods to maintain the routing information about each node in the OLSR network. That is, a node using OLSR maintains a routing information base to all other nodes on the network, even if those nodes are currently not destinations for flows currently traversing this node. All nodes detect neighbour changes by exchanging periodic *HELLO* messages (keep-alive messages), and propagate TC messages among all the nodes of the network to discover available routes in the presence of mobility and resource constraints.

OLSR inherits the concept of link state (LS) routing but with flooding optimisations. In traditional LS-based routing protocols, each node sends its local link-state information to its adjacent nodes once it detects the link changes between itself and its neighbours, and the adjacent nodes then forward the information to their neighbours resulting in *flooding* of the routing messages.

Unlike the traditional LS method, OLSR uses multi-point relays (MPRs; [8,9,12]) to optimise the flooding mechanism. Each node selects a set of its neighbour nodes as MPRs. A node, which has selected its neighbour *A* as its MPR, is called the *MPR Selector* of node *A*.

The selective flooding based on MPR is *efficient* in terms of control message delivery. In Ref. [9] it is shown that such flooding eventually reaches all the nodes in the graph. Also, for each node pair in the network, the subgraph consisting of the unidirectional MPR links in the network and all adjacent links (of the node pair) contains a *shortest* path with respect to the *original* graph.

In particular, the MPR optimisations include the following three aspects.

Firstly, only the MPR nodes are responsible for forwarding control traffic. This reduces significantly the number of message copies required to flood a message to all nodes in the network.

Secondly, the partial LS is advertised in order to provide shortest path routes. In OLSR, only the states from the *MPR selector set* is advertised in the topology control messages. Although bi-directional, only the *unidirectional* LSs (i.e. the link status from the MPR nodes towards their corresponding MPR selectors) are advertised. From this, the nodes eventually obtain a *directed* MPR subgraph of the whole network topology. The motivation for such partial state advertisement is to reduce the size of the topology control TC messages.

Thirdly, only MPR nodes generate the TC messages, since the MPR selector set in non-MPR nodes is *NULL*. This reduces the number of the topology messages generated in the network.

With all the optimisations above, the MPR mechanism provides an efficient method for flooding control traffic by reducing the number of transmissions required and the amount of control traffic flooded. Further, details of OLSR and MPR can be found in Refs. [5,8,9,12].

2.2 Localised topology update in DSDV

Here, we examine briefly an alternative approach, used in DSDV [11].

Each node running DSDV in the network maintains a routing table that records distance vectors, i.e. the number of hops to all of the possible destinations within the network and the corresponding next-hop nodes.

DSDV requires each node to advertise its own routing table by broadcasting its entries to each of its current neighbours *locally*. In order to reduce the amount of state information carried in each update and help alleviate the potentially large amount of topology update traffic, DSDV employs two types of update packets:

- *Full dump updates* carry all available routing information and might require multiple network protocol data units (NPDUs). Full dumps can be transmitted relatively infrequently when no movement of mobile nodes occurs.
- *Incremental dumps* carry only information changed since the last full dump. The size of incremental dumps is smaller than that of full dumps, and therefore can fit into a standard NPDU. When movement becomes frequent, the size of an incremental dump increases and approaches the size of a NPDU. Then a full dump can be scheduled so that the next incremental will be smaller.

2.3 Types of topology update strategies

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 can consume a significant amount of bandwidth.

Localised updates

To reduce the overheads in topology updates, in the protocols such as DSDV [11] and Fisheye State Routing (FSR) [10], each node only propagates 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. FSR adopts the strategy of *temporal partiality*, whereby a node only advertises information about closer nodes and exchanges LS 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.

The intrazone routing protocol [6] and the fast-OLSR extension [2] set the update intervals inversely proportional to the maximum velocity of the nodes. Topology Broadcast using Reverse-Path Forwarding [1] generates two types of updates: full-topology periodic updates and differential updates. DSDV [11] advertises the routing information periodically and incrementally as topological changes are detected.

Reactive updates

Traditional link-state routing protocols such as open shortest path first (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 broadcast quickly with no delay, which may reduce packet drops because of mobility.

2.4 Topology update strategies in this study

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 analytic study

3.1 Performance metrics and assumptions

In the following analytical study, we use *consistency* to evaluate the performance of topology update strategies. In our study, *consistency* is calculated as 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 $R = \{r_1, r_2, \ldots, r_K\}$ 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 $c = (\sum_{k=1}^{K} t(r_k))/(KT)$.

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 a link changes or a 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 (Figure 1). The symbols used in this section can be found in Table 1.

3.2 A probabilistic topology update model

Consider an arbitrary period of time, starting at t_0. Let X be the time of the *first* link change occurrence after t_0. Let $\gamma = X - t_0$ be the waiting time until the first occurrence after t_0. Let r be the topology update interval as depicted in Figure 1.

We define L, the state inconsistency time, as:

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

We make the assumption that occurrences of link changes are exponentially distributed with a mean rate λ:

$$X - t_0 \sim \mathrm{Exp}(\lambda).$$

Therefore, the expected inconsistency time is:

$$E(L) = E(t_0 + r - X)^+ = E(r - \gamma)^+ = \int_0^\infty (r-\gamma)^+ \lambda \mathrm{e}^{-\lambda\gamma} \mathrm{d}\gamma = \int_0^r (r-\gamma)^+ \lambda \mathrm{e}^{-\lambda\gamma} \mathrm{d}\gamma$$

$$= r + \frac{\mathrm{e}^{-r\lambda} - 1}{\lambda} = \varphi(r, \lambda). \tag{1}$$

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

$$\phi(r, \lambda) = \frac{\varphi(r, \lambda)}{r} = 1 + \frac{\mathrm{e}^{-r\lambda} - 1}{r\lambda}. \tag{2}$$

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

$$\psi(r, \lambda) = \phi'(r) = \frac{\mathrm{d}\phi}{\mathrm{d}r} = \frac{1}{r^2\lambda} - \frac{1 + r\lambda}{r^2\lambda \mathrm{e}^{r\lambda}}. \tag{3}$$

Table 1. Definitions 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

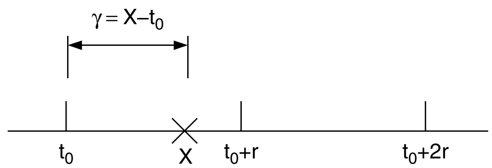

Figure 1. Periodic topology update.

3.3 The impacts of topology update intervals

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

As shown in Figure 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 1 s to 4 s. After that, the consistency ratio has a lesser rate of increase, and increasing the refresh interval does not have significant impact on the performance.

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

In summary, the impact of update intervals on consistency largely depends on the state change rate. Under frequent state changes, tuning topology update intervals does not 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 definitions used in this section can be found in Table 2:

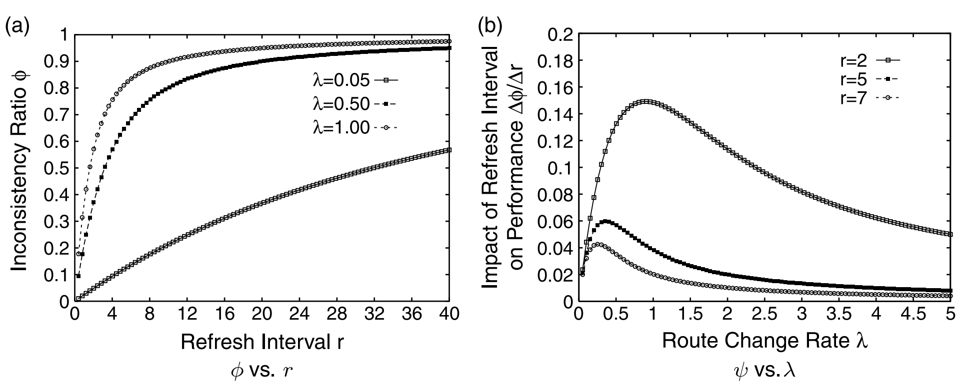

Figure 2. Impacts of refresh interval. (a) ϕ vs. r. (b) ψ vs. λ.

Table 2. Definitions in overhead analysis.

HELLO	Hello messages in neighbor detection
TC	Topology control messages
α	Control traffic overhead
α_{hello}	Control overhead due to HELLO messages
α_{tc}	Control overhead due to TC messages
$\lambda(v)$	Link change rate under node velocity v

(1) *Control overhead under* proactive *strategy*: Let α be the amount of control overhead. Let r be the topology update interval. Then

$$\alpha = \alpha_{\text{hello}} + \alpha_{\text{tc}}. \tag{4}$$

In this study, we keep the HELLO intervals and node density constant. Because of the periodic nature of topology updates, control overhead due to TC messages (α_{tc}) is (approximately) proportional to topology update rate. Therefore,

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

From Equation (4) we can see that the control overhead under the *proactive* update strategy can be approximated by,

$$\alpha = \frac{\alpha_0^{\text{p}}}{r} + c^{\text{p}}, \tag{5}$$

where α_0^{p} and c^{p} are constants determined by the size of the network, the size of the control packets and HELLO intervals.

(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 distribution with fairly high accuracy [14]. The link change inter-arrival time density function can be expressed as,

$$f(t) = \lambda_v \mathrm{e}^{-\lambda_v t}. \tag{6}$$

The average link change interarrival time is $(1/\lambda_v)$. Correspondingly, the average reactive update interval is inversely proportional to λ_v.

Then the control overhead can be approximated by,

$$\alpha = \alpha_0^r \lambda_v + c^r, \tag{7}$$

where α_0^r and c^r are constants determined by the size of the network, the size of the control packets and HELLO intervals.

With the increase of node velocity, the link change inter-arrival time decreases and the link change rate increases [14]. Therefore, under reactive update options, the average update interval decreases and the topology control overhead increases *linearly* with link change rate.

3.5 Summary of analytic study

Our recent study on route duration statistics [7] has shown that, in a MANET with moderate or high mobility, the route duration can be approximated by an exponential distribution with appropriate parameters. In particular, most of the route durations (i.e. more than 75%) are smaller than 4 s. Correspondingly, the route change rate is larger than 0.25/s.

Based on the analytic results in this section we can infer that, because of frequent route changes:

- Topology refresh intervals would not have significant impact on routing throughput.
- Reactive update options would introduce much more traffic overhead than proactive options.

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 Carnegie Mellon University (CMU)[3]. The detailed configuration is shown in Table 3.

The nodes are randomly placed in an area of 1000 m by 1000 m. We use a network consisting of n nodes: $n = 20$ to simulate a low-density network, $n = 50$ to simulate a high-density network. To illustrate, if the nodes are *evenly* distributed, in a low-density network, each node has 4 nodes in its transmission range (Figure 3(a)), while in a high-density network, each node has 12 nodes in its transmission range (Figure 3(b)).

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

All simulations run for 100 s. The steady-state simulations enable us to collect the results within a short simulation period (i.e. 100 s), which could represent long-run averages (such as 1000 s). On the other hand, it is enough to collect mobility data within 100 s, since the nodes at moderate or high velocity ($v > 10$ m/s) could travel from one edge of the area to the opposite edge within 100 s.

A randomly distributed constant bit rate (CBR) traffic model is used which allows every node in the network to be a potential traffic source and destination. The rate of each

Table 3. MAC/PHY layer configurations.

MAC protocol	IEEE 802.11
Radio propagation type	TwoRayGround
Interface queue type	DropTailPriQueu
Antenna model	OmniAntenna
Radio radius	250 m
Channel capacity	2Mbits
Interface queue length	50

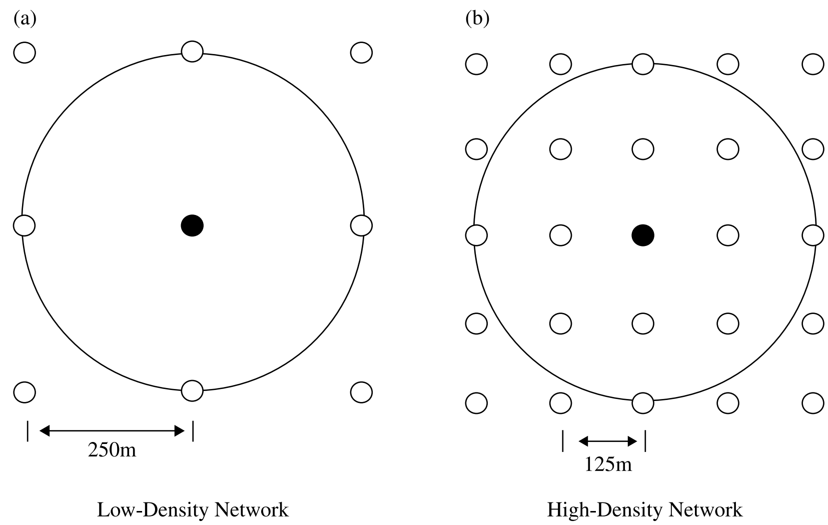

Figure 3. Low-density network vs. high-density network ($rr = 250$): (a) low-density network and (b) high-density network.

CBR flow is 10 kb/s. The CBR packet size is fixed at 512 bytes. There are at least $n/2$ data flows that cover almost every node.

4.2 Metrics

In every simulation, we measure the throughput of each CBR flow and the control traffic overhead, and then use the simulation results to calculate the average performance of each metric.

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). Since, we keep the data rate constant, the throughput in this study could represent the end-to-end packet delivery ratio.

In order to gauge the routing protocol overhead, we measure both the number of routing messages, including HELLO messages and TC messages, and the number of bytes in the routing packets transmitted. Considering the broadcast nature of the control message delivery, the packets are counted by summing the size of all the control packets *received* by each node during the whole simulation period.

In order to gain good confidence in the measurement results, we run the simulations 100 times for each data point, with different mobility pattern files, i.e. different starting states for the node positions. 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.

5. 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.

5.1 Routing performance under proactive topology updates

From Figure 4(a) we can see, in a low-density network, tuning topology update intervals within a certain range (e.g. $1 \le t \le 20$) 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 low (e.g. $v = 1$ m/s), the throughput is almost *constant* when the refresh intervals are between 1 and 10. When the node velocity is moderate (e.g. $v = 5$ m/s) or relatively high (e.g. $v = 20$ m/s), the increase of refresh intervals from 1 to 10 only leads to less than 5% performance degradation.

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

From Figure 4(b), when the refresh intervals are relatively small ($1 \le t \le 5$), reducing topology update intervals leads to up to 50% performance degradation; when the refresh intervals are larger than 10 s, the throughput drops gradually with the increase of topology

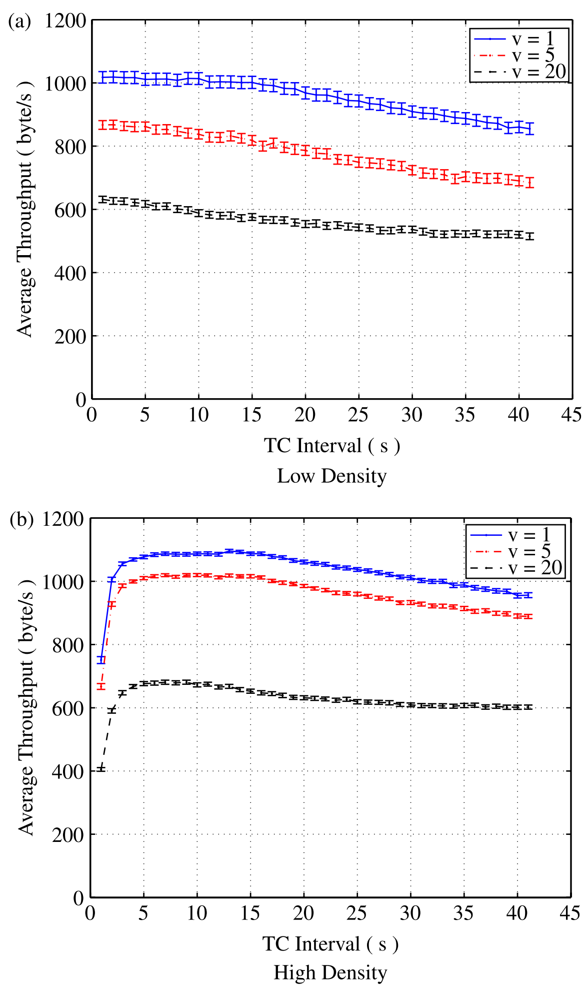

Figure 4. Throughputs vs. topology update intervals ($h = 2$ s $rr = 250$ m): (a) low density and (b) high density.

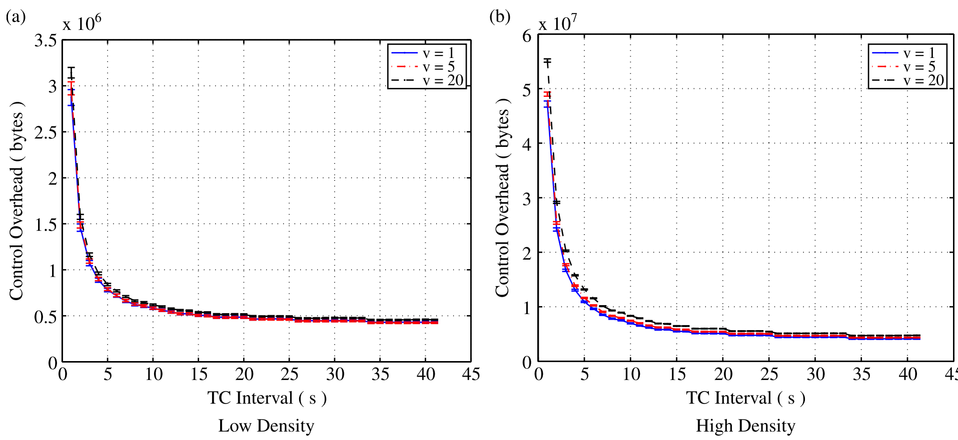

Figure 5. Control overhead vs. topology update intervals ($h = 2\,\mathrm{s}$ $rr = 250\,\mathrm{m}$): (a) low density and (b) high density.

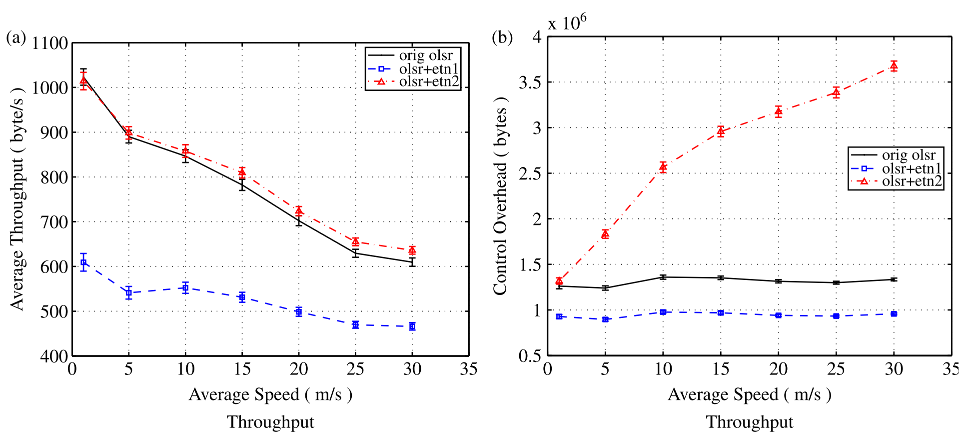

Figure 6. Routing performance under reactive topology update options ($h = 2\,\mathrm{s}$ $rr = 250\,\mathrm{m}$): (a) throughput and (b) control overhead.

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 Figure 5(b). The extra control overhead introduced exaggerates channel contention and leads to queue overflow.

From Figure 5, the control overhead is inversely proportional to topology update intervals. This matches Equation (5) 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.)

5.2 *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 (Figure 6(a)), but it introduces three times more control overhead (Figure 6(b)). Although the localised reactive update (i.e. *etn1*) introduces much less overhead, its throughput is far from satisfactory. This is caused by topology state inconsistency between nodes in the network.

6. 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. Note that some proactive MANET routing protocols like DSDV [10] do not require any network-wide topology advertisements as they use local advertisements only. Therefore, the proposed topology advertisement strategies are not applicable to these protocols. Our analytical and simulation results have shown that the context of OLSR:

- Reducing topology update intervals provides little improvement on the performance of the routing protocol, but with a significant increase of control overhead.
- A reactive topology update approach, as adopted in traditional LS routing protocols such as OSPF, does not perform as well as a proactive topology update approach. Particularly, due to frequent topology changes, the global reactive update approach may introduce significant control overhead. The localised reactive approach, although introducing the least overhead, has the worst packet delivery performance. Therefore, the proactive approach is would be suitable for topology update in MANETs.

These findings do not conflict with Refs. [4] and [13]. 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.

The motivation of this study was to gain better understanding on how topology update strategies can impact proactive routing performance in dynamic and resource-constrained 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. However, our simulations were restricted to a single protocol, OLSR, and only in simulation, so further investigations, using a wider range of protocols, coupled with some empirical studies, would be needed to verify the conclusions we have mooted.

The original data, the NS2 source code and the scripts used in this study are all available from the authors on request.

Notes

1. Email: saleem@cs.st-andrews.ac.uk
2. Email: s.sorensen@cs.ucl.ac.uk
3. *The rice Monarch project*: wireless and mobility extensions to ns-2. http://www.monarch.cs.rice.edu/cmu-ns.html

References

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

[2] M. Benzaid, P. Minet, and K. Agha, *Integrating fast mobility in the OLSR routing protocol*, in *Proc. Fourth IEEE Conference in Mobile and Wireless Communications Networks*, Stockholm, Sweden, September, 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 et al., ed. *Optimized link state routing protocol*, in *IEEE INMIC Pakistan*, Best paper award 2001.
[6] Z.J. Haas, and R. Pearlman, *The performance of query control schemes for the zone routing protocol*, IEEE/ACM Trans. Netw. 9(4) (2001), pp. 427–438.
[7] Y. Huang, S. Bhatti, and S. Sorensen, *A statistical approach on end-to-end proactive routing performance in mobile* ad hoc *networks*, Technical Report RN/06/16, UCL (2006).
[8] P. Jacquet et al., *Performance analysis of OLSR multipoint relay flooding in two* ad hoc *wireless network models*, Technical Report 4260, INRIA, 2001.
[9] ———, *Increasing reliability in cable-free radio LANs low level forwarding in HIPERLAN*, Wirel. Pers. Commun. 4(1) (1997), pp. 51–63.
[10] 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.
[11] C.E. Perkins, and P. Bhagwat, *Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers*, in *SIGCOMM*, New York, NY, USA, 1994.
[12] A. Qayyum, L. Viennot, and A. Laouiti, *Multipoint relaying: An efficient technique for flooding in mobile wireless networks*, Technical Report Research Report RR-3898, INRIA, 2000.
[13] P. Samar and Z. Haas, *Strategies for broadcasting updates by proactive routing protocols in mobile* ad hoc *networks*, in *MILCOM*, 2002.
[14] B. Wicker, *On the behavior of communication links of a node in a multi-hop mobile environment*, in *MobiHoc*, 2004, pp. 145–156.