A TRAFFIC LOCALITY ORIENTED ROUTE DISCOVERY ALGORITHM FOR MANETS

Mznah A. Al-Rodhaan, Lewis Mackenzie, Mohamed Ould-Khaoua

Department of Computing Science
University of Glasgow Glasgow,
UK G128QQ
Email: {rodhaan, lewis, mohamed}@dcs.gla.ac.uk

ABSTRACT

We introduce a new approach to traffic locality then use it to develop a new route discovery algorithm. The algorithm, we named Traffic Locality-Expanding Ring Search (TL-ERS), improves the route discovery process for MANETs that exhibit traffic locality by establishing a neighbourhood that includes the most likely destinations for a particular source, then broadcasting route requests using this neighbourhood as a first locale or ring, in which to search for the target. If route discovery in this ring proves unsuccessful, then the algorithm establishes a second ring, double the size of the first, if route discovery here also fails the algorithm finally resorts to flooding. TL-ERS is adaptive and continuously updates the boundary of the source node's neighbourhood to optimize performance. Furthermore, we provide a detailed performance evaluation using simulation to demonstrate its advantages over both the AODV and the Expanding Ring Search (ERS).

Keywords: MANETs, On-demand, Route Discovery, Traffic Locality, ERS.

1 INTRODUCTION

When mobile devices such as notebooks and PDAs appeared, users wanted wireless connectivity and this duly become a reality. Wireless networks may be infrastructure-oriented as in access point dependent networks [1] or infrastructure-less such as Mobile Ad hoc Networks (MANETs) [2, 1, 3]. Some of the dominant initial motivations for MANET technology came from military applications in environments that lack infrastructure. However, MANET research subsequently diversified into areas such as disaster relief, sensors networks, and personal area networks [3].

The design of an efficient routing strategy is a very challenging issue due to the limited resources in MANETs [1]. Multi-hop routing protocols can be divided into three categories: *proactive*, *reactive*, and *hybrid* [4]. In a proactive routing protocol (table-driven), the routes to all the destinations (or parts of the network) are determined statically at the start up, and maintained by using a periodic route update process, An example of this class of routing protocols is the Optimized Link State Routing Protocol (OLSR) [5]. However, in a reactive routing protocol (on-demand), routes are determined dynamically when they are required by the source using a route discovery process. Its routing overhead is lower than the proactive routing protocols if the network size is relatively small [6]. In on-demand routing, when a source needs to send messages to a destination it initiates a broadcast-based route discovery process to look for one or more possible paths to the destination. Examples from this class are Dynamic Source Routing (DSR) [7] and Ad Hoc On Demand Distance Vector (AODV) [8]. Finally, a hybrid routing protocol combines the basic properties of the first two classes of protocols. That is, they are both reactive and proactive in nature. Zone Routing Protocol (ZRP) [9] is an example belonging to this class.

1.1 Route Discovery Process

On-demand routing algorithms are known to use low bandwidth and to have low power consumption, since the nodes have no periodical tasks, very appealing for MANET scenarios [4]. Such algorithms search for routes only when needed, as the name implies. When a source node needs to send packets to a destination, it initiates a route discovery process, using broadcasting techniques, to look for a route, or routes, to that destination. After discovering the required route(s), the source will start transmitting data packets.

In MANETs, broadcasting is an essential part of the discovery process in on-demand routing protocols. For example, DSR and AODV depend on simple flooding as a form of broadcasting, where each node may receive multiple copies of a unique route request packet and retransmit it exactly once. Unfortunately, flooding leads to a reproduction of the packet causing redundancy that will congest the network and increase the chances of collision: this is known as the broadcast storm problem [10]. So flooding consumes a lot of node resources such as bandwidth and power. These problems can be reduced if the flooding is controlled by stopping the route request as soon as possible upon the discovery of the needed route [11, 12].

In on-demand routing protocols, when a source node needs a route to a particular destination it first searches in its cached routes where any route recently discovered or overheard is stored for future use; if this is unsuccessful, it starts a route discovery process whereby a route request packet is broadcasted from a node to its neighbours until it arrives at the destination or an intermediate relay node that has a route to the destination. The request packet continues to be broadcasted until the time to live (TTL) field reaches zero.

The route discovery process often floods the network with route request packets looking for routes throughout the network. Unfortunately, the route request will keep propagating even after a route has been found and this will, of course, contribute to congestion and wastage of resources. The route discovery protocols can be optimised by minimizing such overhead and reducing or stopping the unnecessary propagation of route request packets after the route has been discovered. In this paper, we will show how exploiting the concept of traffic locality can help to reduce such overhead and improve the efficiency of the route discovery process of on-demand routing protocols such as AODV.

The rest of the paper is organised as follows: Section 2 presents the related work in the literature while section 3 discusses locality in MANETs. Section 4 describes our proposed algorithm, TL-ERS, that utilizes traffic locality; section 5 describes the simulation environment and evaluates the performance by conducting a comparative study that demonstrates the superiority of our algorithm over both AODV and Expanding Ring Search (ERS) [8] in terms of reducing route request overhead. Finally, Section 6 concludes this study.

2 RELATED WORK

In on-demand routing protocols, the broadcasting of the route request used in route discovery dominates most of the routing overhead. Several approaches have been proposed to reduce this overhead by using variations of limited broadcasting so that the overhead of the route discovery process is reduced by limiting the route request broadcast.

The broadcast of the route request can be controlled using the TTL field in the IP header of the route request packet. Expanding Ring Search (ERS) is one of the route request optimization techniques that lower the overhead cost for DSR as presented in [13], and for AODV as proposed in [14]. In ERS the source node searches for the target in multi-ring scheme instead of one-to-all scheme. This is achieved by increasing the TTL value from an initial value to a predefined threshold to expand the radius of the search linearly. Research in [15] tried to find the best initial value for TTL theoretically. They found that the pessimistic search, the worst case where the initial ring may contain the needed route, gives the best performance if destination's speed is known to the source node but the absence of this information makes it a challenging task. Another study [16] proposed two approaches. The first assumes the probability distribution of the destination is known prior to the discovery process while the second one assumes the probability distribution of the destination is not known, fitting more with the unpredictability of MANETs. The second approach uses a sequence of random TTL values to minimize the worst-case search cost. Approaches in [16] was later investigated in [17] on DSR and was observed that when caching of previous routes is taking into consideration the route discovery has similar overhead but higher latency compared to the basic route discovery in DSR.

Hop-Wise Limited broadcast (HoWL) [18] is another approach that limits a route request by predicting the destination location from old routes. It sends the route request packet with a TTL equal to the average of old routes, maybe stale, to that particular destination if it targeted that destination before; otherwise it uses the simple flooding.

3 TRAFFIC LOCALITY IN MANETS

The locality of reference concept deals with the process of accessing a single resource more than once. It includes spatial and temporal locality [19]:

Spatial locality: a resource has a higher chance to be referenced if a neighbouring resource was just referenced.

Temporal locality: a resource that is been referenced now will be referenced again sometime in the immediate near future.

This concept deals with the process of accessing a single resource more than once. It includes spatial and temporal locality [19]. This concept is the principle behind memory and disk caches [20], where instructions and data are placed in higher-speed memory to exploit the probability that future accesses will exhibit locality of reference.

A particular instance of this is observed in paging systems [21]. In a given time period a process will typically tend to concentrate memory references on a particular subset of pages. The term *working set* [21], refers to the collection of pages that a process is actively referencing. Similarly, in networking, locality is observed through the fact that devices within the same geographical area tend to communicate more often than those that are further apart, and exhibit both temporal and spatial locality [21]. This has motivated the concepts of network clusters and workgroups. The working set of a particular node in a network normally refers to the set of nodes which the node is mostly communicating with, not necessarily neighbours, along with intermediate nodes on routes to those targets, during some time interval of interest. In MANETs, locality is observed through the fact that neighbours, nodes in the same geographical area, tend to receive communication from the same sources, highlighting the spatial locality. Also, nodes communicated with in the near past have high probability of re-communicating in the near future leading to temporal locality. Looking at the behaviour of MANETs overall, we observe that the traffic follows a certain pattern, not purely spatial or temporal, in which the source node tends to communicate with a certain set of nodes more that others. This observation has motivated us to introduce another form of locality in MANETs, referred to henceforth as *traffic locality*.

Traffic locality is based on the working set concept, identifying the set of nodes that a given source is mostly communicating with. These nodes are not necessarily identified by spatial or temporal locality but rather by intensity of traffic within the working set over some time interval. Moreover, if a source exhibits traffic locality with a certain destination, the intermediate node comprising the route in question will also be a member of the source node's working set until one of them moves away.

The reasons as to why MANETs exhibit traffic locality are related to the communication requirements of the users carrying and operating the nodes. One common application that exhibits traffic locality in MANETs is a communication group ad hoc network [22] where a group of nodes communicate with each other regardless of their location.

We use the traffic locality concept to optimize the route discovery process. This can be achieved by an adaptive route discovery algorithm that will gradually build up the node neighbourhood as a region centred at the source node and expected to contain most of the members of its working set. Establishing this neighbourhood is a challenging task as it must adapt by expanding and shrinking according to traffic in an effort to reflect the working set. Each node needs a start-up period upon joining the network within which uses simple flooding. Once a locality neighbourhood is reasonably estimated, the algorithm broadcasts any route request with the full capacity of the channel to all nodes within that neighbourhood, in an attempt to minimise the route discovery time. Due to the scarce resources in MANETs, the algorithm must be kept simple by avoiding the collection or manipulation of large amount of data. Each source node has a locality parameter *LP* which reflects the current estimated depth of its neighbourhood as it may be defined by the *weighted average* of hop counts between that source node and *route finders* of previous routes: the finder of a route is the first node that finds the route in its cache table whether it is the destination or an intermediate node. The source node may also keep a counter of its previous route requests whether a weighted or standard average is used. A node, *x*, is considered to be part of the neighbourhood set of a source node, s, if the hop count between *s* and *x* is less than or equal to *LP*.

The algorithm is adaptive and adjusts its neighbourhood. If a route finder is outside the neighbourhood then this requires that the neighbourhood be altered via some adaptation strategy. One such strategy is as follows: *LP* is adjusted by taking the weighted average of the current value of *LP* and the new hop count between the source and the route finder. Alternatively a doubling strategy [11] could be used but this lacks a countervailing shrinking ability and so will not be considered. Fig. 1 shows how the algorithm would shrink or expand the neighbourhood of the source node *s*. In analogy, the expansion and shrinking correspond to page faulting in the context of memory management.

To illustrate the neighbourhood adjustment process, let us consider the source node *s* after it has completed its start-up phase and let $N \quad n_1, n_2, \mathbf{L}, n_l$ be the set of nodes in some network of diameter, *d*, where the diameter of MANET is the path with the smallest number of hops between the two furthest apart nodes in the network [23]. Let $s \quad N$ be a source node and define a function, $h_s : N \quad \mathbb{Z}^+ \quad 0$ where $h_s(v)$ is the hop count between s and some other node $v \quad N$ and $0 \quad h_s(v) \quad d$ .When the source node *s* communicates with any node *f* that is $h_s(f)$ hops away and *y* is the number of previous route requests that already been sent by *s*, the following formula is used by *s* to update its *LP*:

$$LP_{new} \begin{cases} \lceil LP_{old} \quad (1 \quad) \ h_s(f) \rceil & h_s(f) \quad LP_{old} \\ \lfloor LP_{old} \quad (1 \quad) \ h_s(f) \rfloor & h_s(f) \quad LP_{old} \end{cases} \qquad (1)$$

where $y/(y \quad 1)$. It is clear that if $h_s(f) \quad LP_{old}$ then the neighbourhood of s will expand and it will shrink if $h_s(f) \quad LP_{old}$.

Fig. 2 shows the steps of updating the locality parameter *LP* by the source node after receiving the route reply so the neighbourhood region will be ready for next route request. For clarity, the function *Ceiling* will return the smallest integer greater than or equal to its parameter while the function *Floor* will return the greatest integer less than or equal to its parameter. To prevent from approaching too close to 1 as y getting bigger due to $\lim_{y} (\quad) \quad 1$, where only the function Ceiling or Floor will affect the value of *LP,* we need to reset y to an initial value, Initial$_y$, when y reaches some threshold, Threshold$_y$.

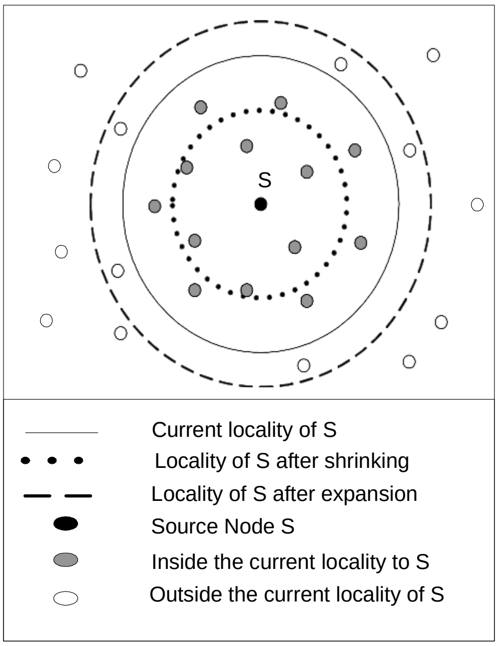

S
Current locality of S
Locality of S after shrinking
Locality of S after expansion
Source Node S
Inside the current locality to S
Outside the current locality of S

Figure 1: Neighbourhood of source node *s* with expansion and shrinking.

```
Algorithm preformed by source node upon receiving a route reply
y = previous number of route requests.

If y >= Threshold_y then
    y = Initial_y
End if
  $y/(y \quad 1)$
$LP_{new} \quad LP_{old} \quad (1 \quad ) LP_{old}$
If hop_count < LP_old then
    LP_new = Floor(LP_new)
Else
    LP_new = Ceiling(LP_new)
End if
LP_old = LP_new
y =y+1
```

Figure 2: locality parameter *LP* update at the source node.

4 TRAFFIC LOCALITY-EXPANDING RING SEARCH

Network-wide flooding is a very expensive process. A process such as this should be avoided in a resource-limited environment such as a MANET. One way to search for a route without covering the whole network is to use Expanding Ring Search (ERS) [14]. It works by searching successively larger areas centred around the source node, until the required route is located. The basic idea behind ERS is to find a local node with a valid route to the destination and avoid flooding the entire network in search of such a route. Therefore, the source node starts the search by broadcasting a route request with TTL= TTL_START to flood the first ring. Each time the source times-out without receiving a reply it re-initiates the route request with TTL incremented by TTL_INCREMENT. This process continues until a TTL_THRESHOLD is reached. If no route has been located by this time, flooding is used with TTL = network diameter (*d*).All nodes in a connected network use the same fixed predefined values for TTL_START, TTL_INCREMENT and TTL_THRESHOLD. For instance, when ERS is used to optimize AODV protocol described in [8] it employs TTL parameters as in Table 1 and follows Fig. 3.

The route request time (RRT) is the time from the first initiation of a route request until it is discarded. RRT for ERS is:

$$RRT_{ERS} \quad \begin{cases} \sum_{i\ \ 1}^{r}(2i\ \ \ 1) & h_s(f)\ \ \ T \\ \sum_{i\ \ 1}^{T}(2i\ \ \ 1)\ \ \ d & h_s(f)\ \ \ T \end{cases} \qquad (2)$$

Where $h_s(f)\ \ T$ means success in ring r, $h_s(f)\ \ T$ means last attempt which is whole network coverage, T = TTL-THRESHOLD and r is the ring that contains $h_s(f)$.

To obtain a search result that is as close as possible to the optimal search result whilst costing less, the search strategy has to be set to suit the application scenario and system configuration. ERS is not necessarily better than simple flooding if the ERS parameters are not selected properly [24, 16, 17, 15]. Selecting the initial TTL value for the first search ring is an important step towards a more effective search [15].

Traffic Locality-Expanding Ring Search (TL-ERS) is a modification of ERS using the traffic locality concept. The main difference between ERS and TL-ERS is in the TTL parameters. ERS uses a fixed radius for all nodes in the network depending on the search ring but TL-ERS uses adaptive values. TL-ERS uses the value of *LP* as the radius of the first ring. *LP* differs from source node to source node and always updated to reflect the current environment. It uses the parameters stated in Table 1 and follows Fig. 4. Moreover, TL-ERS limits the number of rings in the worst case to three rings to achieve lower cost broadcasting as discussed in [24].

Table 1: ERS Parameters

TTL parameter	ERS	TL_ERS
TTL_START	1	LP
TTL_INCREMENT	2	2LP
TTL_THRESHOLD	7	2LP

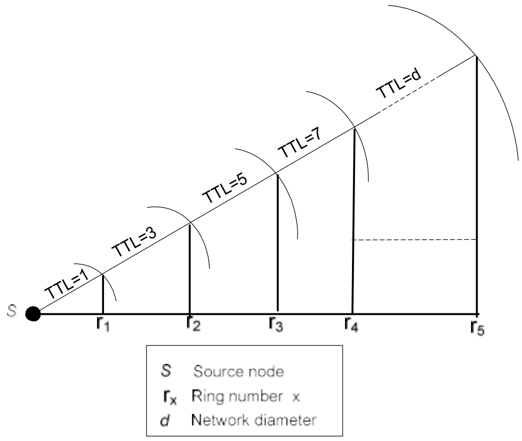


Figure 3: Successive rings in ERS

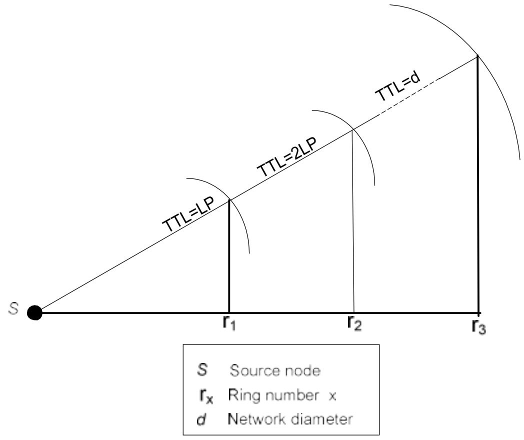


Figure 4: Successive rings in TL-ERS.

The pseudo code for initiating or reinitiating a route request with the correct TTL is described in Fig. 5.

TL-ERS works by initialising the TTL field with the value of *LP* for the first search ring. If the source node times out without receiving a route reply, it reissues the route request with TTL equal to twice *LP*. If it times out again it will flood the whole network by assigning TTL to be the network diameter. The RRT of TL-ERS equals:

$$RRT_{TL\ \ ERS} \quad \begin{cases} LP & h_s(f)\ \ \ LP \\ LP\ \ \ 2LP & LP\ \ \ h_s(f)\ \ \ 2LP \\ LP\ \ \ 2LP\ \ \ d & 2LP\ \ \ h_s(f)\ \ \ d \end{cases} \quad (3)$$

Where $h_s(f)\ \ LP$ means success in the first ring, $LP\ \ h_s(f)\ \ 2LP$ means success in the second ring and $2LP\ \ h_s(f)\ \ d$ means whole network coverage.

5 SIMULATION

Simulations have been conducted to evaluate our algorithm, TL-ERS, against both AODV and AODV with ERS [8], ERS for short, algorithms. The three algorithms were implemented using NS2 simulator version 2.29 [25] conducting extensive experiments to evaluate the performance of TL-ERS. The comparison metrics include the route request (RREQ) delay and the total number of route requests to study the route request latency and overhead. The metrics also include end-to-end delay to study the network behaviour in different scenarios.

```
Algorithm preformed by the source node upon sending a route request in TL-ERS approach for initiating or reinitiating route request.
If ring = 1 then
    TTL=LP
    broadcast the route request
Else
    If ring = 2 then
        TTL= 2LP
        broadcast the route request
    Else
        If ring=3 and 2LP < d then
            TTL= d
            broadcast the route request
        Else
            destination not found.
        End if
    End if
End if
```

Figure 5: TTL initialization steps for initiating or reinitiating a route request in TL-ERS.

Since nodes are mobile in MANETs, modelling these movements is not obvious. In order to simulate a new protocol such as TL-ERS, it is necessary to use a mobility model that reasonably represents the movements of a typical node [26]. Accurate mobility models should be chosen carefully to determine whether the proposed protocol will be useful when implemented or not. Moreover, one of the main characteristics of mobility in MANETs is the maximum speed of nodes because the average speed of nodes determines the rate of broking links which increase the overhead in on-demand protocols.

In MANETs, the entity mobility models typically represent nodes whose movements are completely independent of each other, e.g. the Random Way Point (RWP) model [13]. On the other hand, a group mobility model may be used to simulate a cooperative characteristic such as working together to accomplish a common goal. Such a model reflects the behaviour of nodes in a group as the group moves together, e.g. Reference Point Group Mobility (RPGM) model [27].

5.1 Simulation Environment

We assume all nodes are identical, all links are bidirectional and no selfishness in the network. Mobile nodes are assumed to operate in a squared simulation area of side 1000m. The transmission range is 100m and is fixed in all nodes to approximately simulate networks with a maximum hop count of 10 hops. Each run was simulated for 900 seconds of simulation time, ignoring the first 30 seconds as a start-up period for the whole network. For each topology, 30 runs were performed. The results of these runs were averaged to produce the graphs, Figs. 6-14, shown below and a 95% confidence interval is produced (shown as standard error bars in the relevant figures). Table 2 provides a summary of the chosen simulation parameter values.

Table 2: Simulation Parameters.

Parameter	Value
Transmission range	100m
Topology size	1000x1000m
Simulation time	900s
Packet size	512 bytes
Packet rate	4pkt/s
Data sessions	5,10,15,…,35
Traffic type	CBR(UDP)
Routing protocol	AODV
Number of Nodes	20,30,..,100
Runs per point	30
Antenna type	Omni Antenna
MAC protocol	IEEE 802.11
Mobility model	RGPM model
Propagation model	Two-Ray Ground
Maximum speed	5,10,…,30m/s
Minimum speed	1m/s
Threshold_y	10
Initial_y	1

The simulation area is kept constant in all scenarios to study TL-ERS performance in both scarce and dense environments, since we are interested in knowing the behaviour of our algorithm in both kinds of environments.

A traffic generator was used to simulate constant bit rate (CBR) with a packet data payload of 512 bytes. In this study, each source-destination connection is called a session. For example, 10 sessions represent 10 source-to-destination connections. The traffic load in the network is determined by the number of sessions allowed in the network. Moreover, data sessions, flows, between different source and destination pairs in groups of ten nodes were simulated to simulate traffic in a network that exhibit traffic locality. The source transmits data packets at a rate of four packets per second.

The RPGM model was utilized as a mobility model in all of our simulations since it models the random motion of groups of nodes and of individual nodes within the group. Group movements are based upon the movement of the group reference point following its direction and speed. Moreover, nodes move randomly within their group with speeds between 1 and 30m/s. Each group contains 10 nodes. The minimum speed was 1 while pause time of 50s.

The Two-ray Ground model was utilized as a radio propagation model in all of our simulations because it considers both the single line-of-sight path and a ground reflection path. When two-ray ground used as radio propagation model in ns-2.29, the system uses Friss-space attenuation at near distance and Two-ray Ground at far distance depending on the distance between transmitter and receiver.

In our simulation, we concentrate on three major parameters: network size, data sessions and maximum speed in three different cases by varying one parameter while keeping the other two constant as explained blow:

Case 1: Network size is the total number of nodes in the network. When the network size increases, the average hop length of routes also increases which may increase the error rate and/or increase network latency. Simulation has been performed using nine topologies with different number of nodes, multiples of 10, from 20 to 100 while other simulations use a fix number of nodes i.e. constant value.

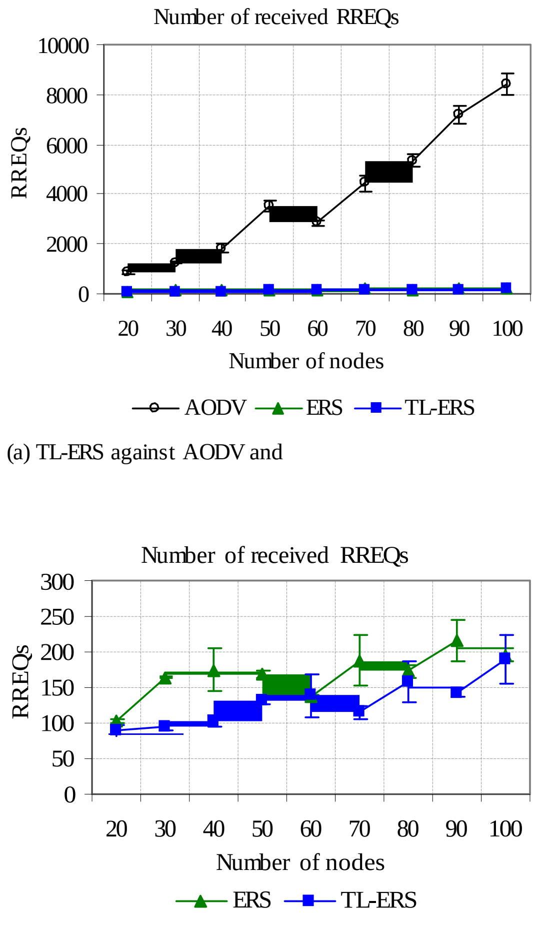
Case 2: The traffic load was varied from light through moderate to heavy traffic to evaluate our algorithm in the present of different traffic load. A traffic load of sizes 5, 10… 35 sessions were used in some simulations with network of size 70 nodes and maximum speed of 30m/s to test our algorithms using reasonably incremented amount of traffic meanwhile avoiding saturation. In the other hand, traffic load of 10 sessions were used in other simulations for networks of variable sizes or variable maximum speed.

Case 3: The maximum speed varied from 5, 10…30m/s in some simulation runs and fixed at 30m/s in others.

5.2 Simulation Analysis

Considering case 1, variable network size, Figs 6, 7 and 8 display the results of running our algorithm, TL-ERS, against both AODV and ERS for 900 seconds using networks with different number of nodes, from 20 to 100 in an area of 1000m x 1000m with RGPM as the mobility model with a minimum speed 1m/s and a maximum speed of 30m/s. The number of data sessions is fixed to ten.

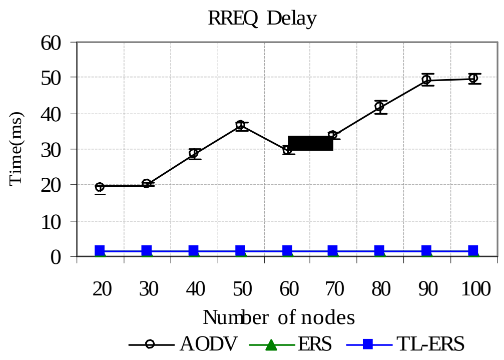
Fig. 6 shows the superiority of TL-ERS over both AODV and ERS by minimizing the route request overhead. In Fig. 6(b), we have used a larger scale for TL-ERS vs. ERS to clearly compare their behaviour and we will do the same for all the figures throughout this section. The RREQ overhead, measured by the number of RREQ received in the whole network, of TL-ERS is lower than ERS as Fig. 6(b) and significantly lower than AODV as shown in Fig. 6(a). In fact, the difference in RREQ overhead between TL-ERS and AODV increases with dense network. Nevertheless, the network latency is almost the same as shown in Fig. 7(b) between TL-ERS and ERS. However, there is great reduction in RREQ latency over AODV as in Fig. 7(a). In other words, network performance improves due to the reduction in the number of rebroadcast route requests as presented in Fig. 6 with low latency which has a generally beneficial effect on the network performance due to the fact that the data can typically travel with less congestion.



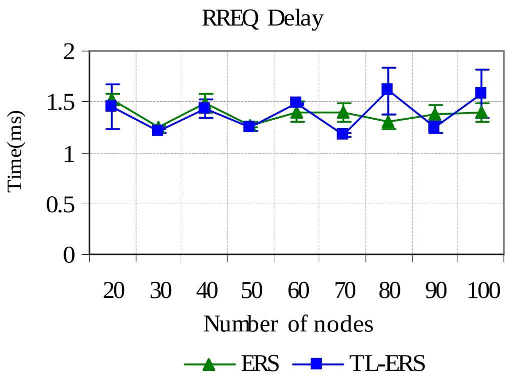
(a) TL-ERS against AODV and

(b) TL-ERS against ERS

Figure 6: Route requests overhead for different number of nodes.



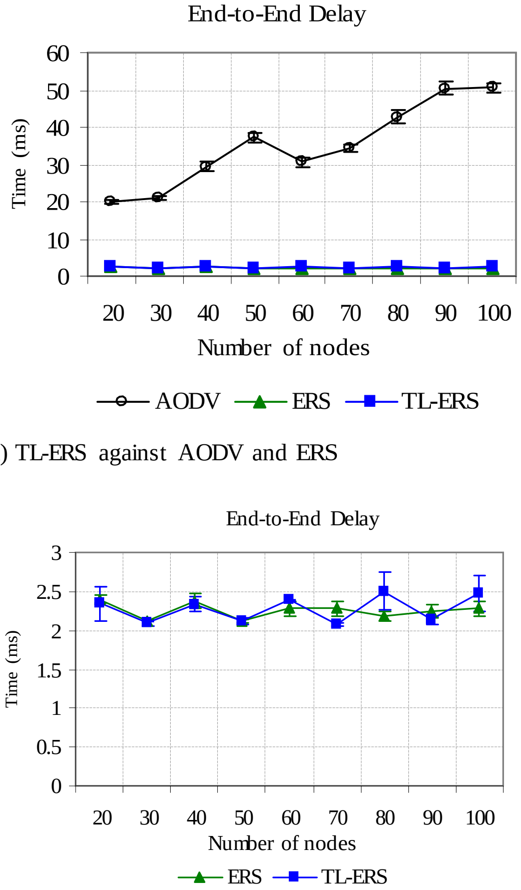
(a) TL-ERS agains t AODV and ERS



(b) TL-ERS against ERS

Figure 7: Route request delay verses network density.

The end-to-end delay is defined to be the total delay for the actual transmitted data plus the RREQ delay. Fig. 8(a) and Fig. 8(b) show the end-to-end delay and demonstrate that TL-ERS gives less network latency than both AODV and ERS due to the reduction in number of RREQ.



(a) TL-ERS against AODV and ERS

(b) TL-ERS against ERS

Figure 8: Network delay versus network density.

For case 2, simulations with different number of data sessions, Fig. 9-11 display the results of running our algorithm, TL-ERS, against both AODV and ERS for 900 seconds using networks of size 70 nodes in an area of 1000m x 1000m with a random speed range between 1 and 30m/s. The amount of traffic ranged from 5 to 35 flows incremented by five.

Figs. 9 and 10 show that TL-ERS improves the route request overhead over both AODV and ERS. This improvement increases with dense network. The three algorithms give stable averages of data throughput regardless of network size due to the fact that the communication always between nodes within the same group thus the movement follows the RGPM model.

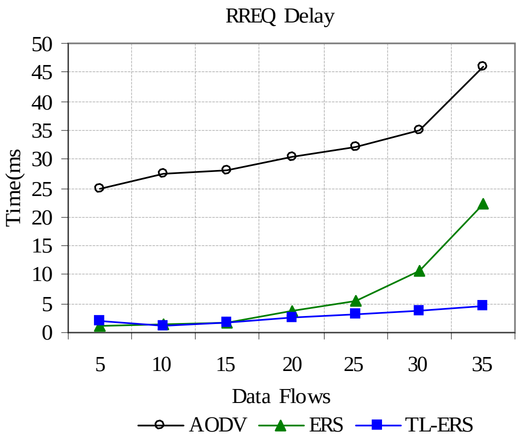


Figure 9: RREQ latency for different number of data sessions.

Fig. 11 shows that TL-ERS has lower network latency compare to both AODV and ERS due to the reduction in the number of rebroadcast route requests. The attractiveness of this stems from the fact the data can travel faster and with less congestion.

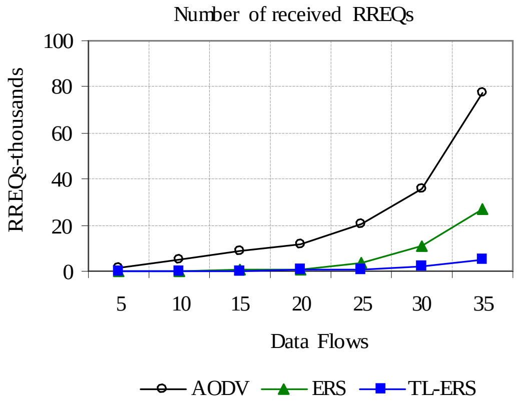


Figure 10: RREQs overhead for different number of data sessions.

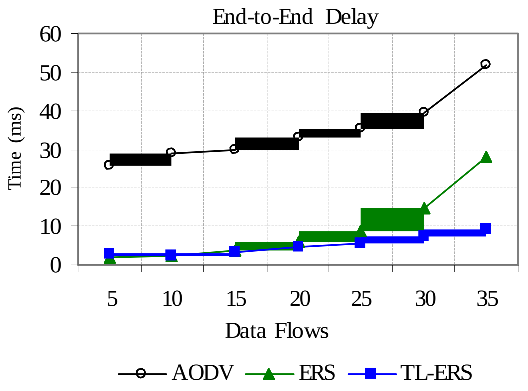
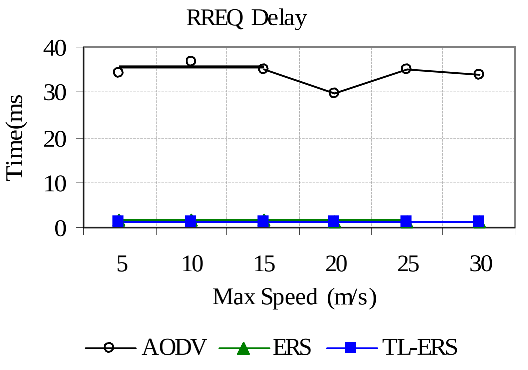
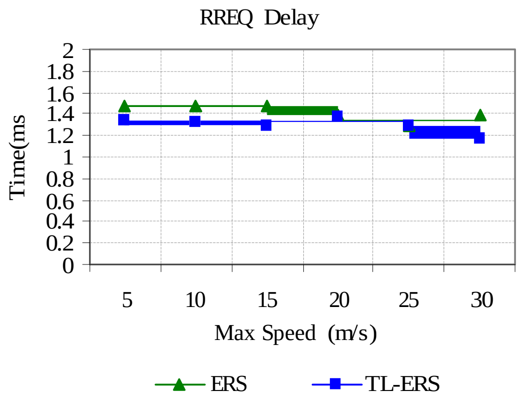


Figure 11: End-to-end delay for different number of data sessions.

For case 3, Figs. 12-14 were extracted from simulating the three algorithms while increasing the maximum speed by five steps starting from 5 to 30m/s. The number of nodes was fixed at 70 and data sessions at 10. Fig. 12(a) shows a great reduction of RREQs latency in both ERS and TL-ERS over AODV because the success case is the dominant in both TL-ERS and ERS while Fig. 12(b) shows that TL-ERS gives little improvement in RREQ delay compared to ERS regardless of speed. Also the overhead of the RREQs in AODV is higher than both TL-ERS and ERS as in Fig. 13(a). Nevertheless, the RREQ overhead of TL-ERS is better than ERS as shown in Fig. 13(b) which will improve the network latency as in Fig. 14.

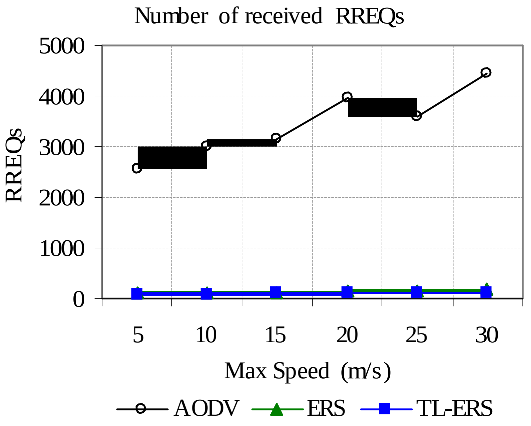


(a) TL-ERS against AODV and

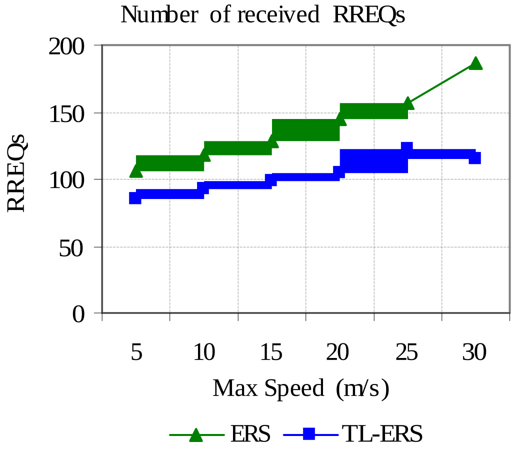


(b) TL-ERS against ERS

Figure 12: RREQ latency for different maximum speeds.

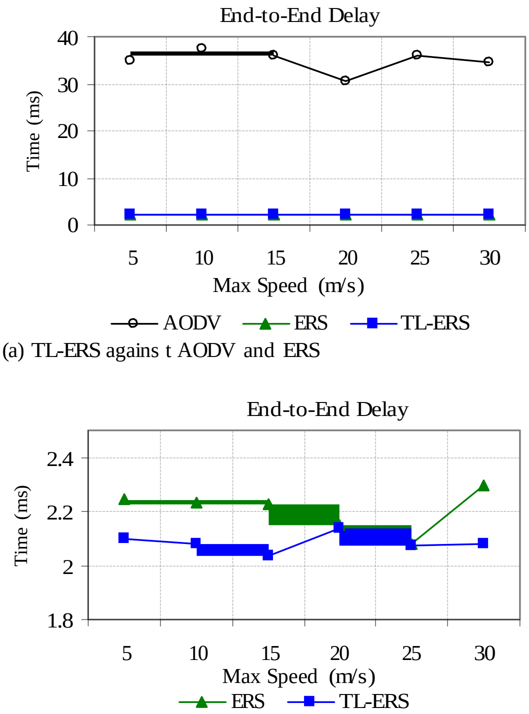


(a) TL-ERS against AODV and ERS



(b) TL-ERS against ERS

Figure 13: RREQs overhead for different maximum speeds.



(a) TL-ERS agains t AODV and ERS

(b) TL-ERS against ERS

Figure 14: End-to-end delay for different maximum speeds.

6 CONCLUSION

In this paper, we introduced a new approach to traffic locality then used it to develop a new route discovery algorithm for MANETs, named as TL-ERS. It reduces the route request overhead during the route discovery process by exploiting traffic locality. It works by including most of the likely destinations for the source node on hand in the first ring and broadcasts the route requests first within this ring. If no route is found in the first ring, the ring search will be doubled as a second attempt. If this is unsuccessful, a network wide broadcast is performed. The algorithm is adaptive and continuously updates the boundaries of the first and second ring to provide better performance. Furthermore, we have provided a performance evaluation for TL-ERS and compared it against both AODV and Expanding Ring Search (ERS) algorithm. Our evaluation has shown that TL-ERS exhibits a lower route request overhead since it broadcasts lower number of route requests than both AODV and ERS.

A low route requests overhead in TL-ERS will have a positive impact on network performance since the transmission of data packets will start earlier and with less congestion so the end-to-end network latency improves. As future work, we will investigate other route request optimization techniques in on-demand routing algorithms.

7 REFERENCES

[1] S. Murthy and B. Manoj: Ad Hoc Wireless Networks: Architectures and Protocols, Prentice Hall, (2004).

[2] I. Chlamtac, M. Conti and J. J. N. Liu: Mobile ad hoc networking: imperatives and challenges. Ad Hoc Networks,(1). **1**: p. 13-64 (2003).

[3] A. Tanenbaum: Computer Networks, Pearson Education, (2003).

[4] M. Abolhasan, T. Wysocki and E. Dutkiewicz: A review of routing protocols for mobile ad hoc networks. Ad Hoc Networks,(1). **2**: p. 1-22 (2004).

[5] C. Adjih, T. Clausen, P. Jacquet et al.: "Optimized Link State Routing Protocol," The Internet Engineering Task Force, IETF, RFC 3626 (2003).

[6] S. R. Das, R. Castaneda, Y. Jiangtao and R. Sengupta: Comparative performance evaluation of routing protocols for mobile, ad hoc networks (1998).

[7] D. Johnson, D. Maltz and Y.-C. Hu: "The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR)," The Internet Engineering Task Force, IETF, draft-ietf-manet-dsr-09.txt, (2003).

[8] C. Perkins, E. Belding-Royer and S. Das: "AODV Ad Hoc On-Demand Distance Vector Routing," The Internet Engineering Task Force, IETF, RFC 3561, (2003).

[9] Z. J. Haas, M. R. Pearlman and P. Samar: "The Zone Routing Protocol (ZRP) for Ad Hoc Networks," IETF MANET Working Group, (2002).

[10] T. Yu-Chee, N. Sze-Yao, C. Yuh-Shyan and S. Jang-Ping: The broadcast storm problem in a mobile ad hoc network. Wireless Networks,(2/3). **8**: p. 153-167 (2002).

[11] L. Gargano, M. Hammar and A. Pagh: Limiting flooding expenses in on-demand source-initiated protocols for mobile wireless networks. in 18th International Parallel and Distributed Processing Symposium (2004).

[12] M. Al-Rodhaan, L. Mackenzie and M. Ould-Khaoua: A New Route Discovery Algorithm for MANETs with Chase Packets. International Journal of Simulation Systems, Science & Technology,(Performance Modelling of Computer Networks, Systems and Services). **Volume 8, Number 3**: p. 1-12 (2007).

[13] D. B. Johnson and D. A. Maltz: "Dynamic Source Routing in Ad Hoc Wireless Networks," Mobile Computing, I. a. Korth (Editor), vol. 353, Kluwer Academic Publishers, Norwell, MA, pp. 153-181,(1996).

[14] L. Sung-Ju, E. Belding-Royer and C. Perkins: Scalability study of the ad hoc on-demand distance vector routing protocol. Int. J. Netw.

Manag.,(2). **13**: p. 97-114 (2003).

[15] K. Wu, H.-C. Yang and F. Li: Pessimism is mostly the best in the expanding ring search for wireless networks. in Proc. of IEEE Global Telecommunications Conference (Globecom'2006). San Francisco, California (2006).

[16] N. Chang and M. Liu: "Revisiting the TTL-based controlled flooding search: optimality and randomization," Proceedings of the 10th annual international conference on Mobile computing and networking, ACM Press, Philadelphia, PA, USA, (2004).

[17] D. Koutsonikolas, S. M. Das, H. Pucha and Y. C. Hu: On optimal TTL sequence-based route discovery in MANETs. in Second International Workshop on Wireless Ad Hoc Networking (2005).

[18] M. Minematsu, M. Saito, H. Aida et al.: HOWL: an efficient route discovery scheme using routing history in ad hoc networks. in 27th Annual IEEE International Conference on Local Computer Networks (LCN'02) (2002).

[19] Wikipedia: "Locality of reference," (2007).

[20] P. Denning: The locality principle. Commun. ACM,(7). **48**: p. 19-24 (2005).

[21] A. Silberschatz, P. Galvin and G. Gagne: Operating Systems Concepts, John Wiley & Sons, (2005).

[22] M. Mosko and J. Garcia-Luna-Aceves: Performance of group communication over ad hoc networks. in IEEE Int. Symp. Computers and Communications ISCC. Italy (2002).

[23] S. Kurkowski, T. Camp and M. Colagrosso: MANET simulation studies: the incredibles. SIGMOBILE Mob. Comput. Commun. Rev.,(4). **9**: p. 50-61 (2005).

[24] Z. Cheng and W. Heinzelman: "Flooding strategy for target discovery in wireless networks," Proceedings of the 6th ACM international workshop on Modeling analysis and simulation of wireless and mobile systems, ACM Press, San Diego, CA, USA, (2003).

[25] K. Fall: "ns Notes and Documentation," The VINT Project, (2000).

[26] T. Camp, J. Boleng and V. Davies: A Survey of Mobility Models for Ad Hoc Network Research. Wireless Communications & Mobile Computing (WCMC): Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications,(5). **2**: p. 483-502 (2002).

[27] X. Hong, G. Mario, P. Guangyu and C. Ching-Chuan: "A group mobility model for ad hoc wireless networks," Proceedings of the 2nd ACM international workshop on Modeling, analysis and simulation of wireless and mobile systems, ACM Press, Seattle, Washington, United States, (1999).