Int. J. Communications, Network and System Sciences, 2009, 3, 169-247
doi:10.4236/ijcns.2009.23022 Published Online June 2009 (http://www.SciRP.org/journal/ijcns/).

On the Performance of Traffic Locality Oriented Route Discovery Algorithm with Delay

Mznah AL-RODHAAN[1], Lewis MACKENZIE[2], Mohamed OULD-KHAOUA[3]
[1]*Sponsored by King Saud University, Riyadh, Saudi Arabia*
[1,2]*Department of Computing Science, University of Glasgow, Glasgow, UK*
[3]*Department of Electrical & Computer Engineering, Sultan Qaboos University, Muscat, Oman*
Email: {*rodhaan, lewis*}*@dcs.gla.ac.uk, mok@squ.edu.om*
Received January 24, 2009; *revised April* 2, 2009; *accepted April* 5, 2009

ABSTRACT

In MANETs, traffic may follow certain pattern that is not necessarily spatial or temporal but rather to follow special needs as a part of group for collaboration purposes. The source node tends to communicate with a certain set of nodes more than others regardless of their location exhibiting traffic locality where this set changes over time. We introduce a traffic locality oriented route discovery algorithm with delay, TLRDA-D. It utilises traffic locality by establishing a neighbourhood that includes the most likely destinations for a particular source node. The source node broadcasts the route request according to the original routing used. However, each intermediate node broadcasts the route request with a delay beyond this boundary to give priority for route requests that are travelling within their own source node's neighbourhood region. This approach improves the end-to-end delay and packet loss, as it generates less contention throughout the network. TLRDA-D is analysed using simulation to study the effect of adding a delay to route request propagation and to decide on the amount of the added delay.

Keywords: MANETs, On-Demand Routing Protocols, Route Discovery, Delay, Congestion, Simulation Analysis

1. Introduction

When mobile devices such as notebooks and PDAs appeared, users wanted wireless connectivity and this duly become a reality. Wireless networks could be infrastructure-oriented as in access point dependent networks [1] or infrastructure-less multi-hop such as Mobile Ad hoc NETworks (MANETs) [1,2]. 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 [2].

The design of an efficient routing strategy is a very challenging issue due to the limited resources in MANETs [1]. MANETs routing protocols can be divided into three categories: proactive, reactive, and hybrid [3]. In proactive routing protocols (table-driven), the routes to all the destinations (or parts of the network) are determined statically at the start up then maintained using a periodic route update process. An example of this class of routing protocols is the Optimized Link State Routing Protocol (OLSR) [4]. However, in reactive routing protocols (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 [5]. Examples of this class are Dynamic Source Routing (DSR) [6] and Ad Hoc On Demand Distance Vector (AODV) [7]. Finally, hybrid routing protocols combine the basic properties of the first two classes of protocols; so they are both reactive and proactive in nature. Zone Routing Protocol (ZRP) [8] is an example belonging to this class.

In on-demand routing protocols, when a source node needs to send messages to a destination it initiates a broadcast-based route discovery process looking for one or more possible paths to the destination where the broadcasting of the route request dominates most of the routing overhead.

In this paper, a traffic locality oriented route discovery algorithm that uses delay, TLRDA-D, is introduced.

Copyright © 2009 SciRes.

Moreover, TLRDA-D is analysed using simulation to understand the relationship between congestion and delay and ease the decision on the amount of the added delay.

The rest of the paper is organised as follows: Section 2 presents the related work while Section 3 presents the proposed algorithm; evaluates the performance and describes the simulation environment and observation. Finally, Section 4 concludes this study.

2. Related Work

The principle of locality was first applied in memory referencing behaviour [9] then it was subsequently observed in the use of other resources such as file referencing [10]. The locality of reference concept deals with the process of accessing a single resource more than once. It includes spatial and temporal locality [11,12]. 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 [13]. The importance of traffic locality concept is recognized in networking. Traffic locality concept is a motivation factor behind network clusters and workgroups [14]. While in infrastructure wireless networks, traffic locality is utilized to improve load balancing in base stations [1,15]. 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 within the near past have high probability of re-communicating in the near future leading to temporal locality [16]. Sometimes a node communicates with a certain set of nodes more than others within a particular time regardless of their locations, highlighting the traffic locality [17].

3. Traffic Locality Oriented Route Discovery Algorithm with Delay (TLRDA-D)

MANETs are very useful in applications that need immediate collaboration and communication with the absence of network infrastructure where a temporary connection can be established for quick communication. These collaborative jobs demand traffic to be between known source-destination pairs to accomplish specific tasks. So if this pattern of traffic is found in an application then the design of the algorithm should utilize it.

Looking at the traffic behaviour of MANETs, the traffic may follow a certain pattern, not purely spatial or temporal, in which the source node tends to communicate with a set of nodes more than others regardless of their locations in a connected network. The traffic locality of a particular source node is captured in its working set. The working set is a set of nodes that the source node is mostly communicating with, not necessarily neighbours where members of the working set change over time. Moreover, the traffic locality is identified by the intensity of traffic within the working set over some time interval. If a source node 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 far away.

MANETs exhibit traffic locality due to the communication requirements of the users carrying and operating them. One common application that exhibits traffic locality in MANETs is a group communication ad hoc network [18] where a group of nodes communicate to accomplish a common goal.

In this paper, traffic locality concept [17] is utilized to improve the route discovery process in on-demand routing protocols for MANETs. It is used to develop a new adaptive route discovery algorithm, TLRDA-D. The algorithm works by gradually building up the node neighbourhood as a region centred at the source node and expected to contain most of the members of its working set where the whole connected network consists of two disjoint regions: *neighbourhood* and *beyond-neighbourhood*.

Establishing this neighbourhood is a challenging endeavour as it must adapt according to the traffic in an effort to build then maintains the neighbourhood region that reflects the current working set. Upon joining the network, the new node needs a start-up period during which it uses the original broadcast algorithm depending on the routing algorithm used.

Since the neighbourhood region contains the source node's working set, no extra delays are imposed in this region to avoid delaying the route discovery process. On the other hand, delaying a fulfilled route request in the beyond-neighbourhood region reduces channel contention without adding any latency to the discovery process.

Due to the scarce resources in MANETs, the algorithm is kept simple by avoiding the collection or manipulation of large amount of data. Furthermore, the global information is avoided because it is unavailable in a real environment that uses no external resources.

Each node has a locality parameter *LP* where $LP \in \mathbb{N}^*$ which corresponds to the current estimated depth of its neighbourhood as it might be defined by the *weighted average* of hop counts between that source node and destinations as in Equation 1 including route finder. 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.

Let $s \in N$ be a source node in a network of N nodes and define a function, $h_s : N \to \mathbb{Z}^+ \bigcup \{0\}$ where $h_s(u)$ is the hop count between s and some other node $u \in N$

Copyright © 2009 SciRes.

and $h_s(s)=0$. A node, *x*, is considered to be part of the working set of a source node, *s*, if $h_s(x) \le LP$. In TLRDA-D algorithm, source node broadcasts route requests after adding the value of its *LP* to the route request packet so intermediate nodes can decide if the route request is within its source node's neighbourhood or not. To avoid ambiguity we will use LP_r to refer to the *LP* stored in the route request. Also to calculate *LP,* the source node needs to store locally the number of its previous route requests.

Formally, we can view the issue as a two tier-partition where the two tiers $\{\tau_1, \tau_2\}$ are the neighbourhood and beyond-neighbourhood respectively in a network that exhibits traffic locality. It is obvious that the two tiers are disjoint sets so $\tau_1 \bigcap \tau_2 = \phi$. Let us consider a source node *s*, any node $v \in \tau_1$ satisfies the condition $h_s(v) \le LP_r$ and any node $u \in \tau_2$ should satisfy the condition $h_s(u) > LP_r$. *LP* is continuously tuned to adapt to the current situation using the values of $h_s(d)$.

The algorithm is adaptive and adjusts its neighbourhood depth, *LP*, to expand or shrink the neighbourhood boundary. If the destination is outside the neighbourhood then this requires the neighbourhood to be adjusted by the following strategy: *LP* is adjusted by taking the weighted average of the current value of *LP* and the new hop count extracted from the received route reply packet.

To illustrate the neighbourhood adjustment process, let us consider the source node *s* at any time after completing its start up phase; when *s* receives a reply answering its current query it updates its *LP* using Equation 1 after extracting $h_s(d)$ from the received route reply packet and *y* is the number of previous route requests that already been sent by *s*. If $h_s(d) \ge LP_{old}$ then the neighbourhood of *s* expands; otherwise it shrinks.

$$LP_{old} = \alpha \times LP_{old} + (1-\alpha) \times h_s(d)$$

$$LP_{new} = \begin{cases} \lceil LP_{old} \rceil & h_s(d) \ge LP_{old} \\ \lfloor LP_{old} \rfloor & h_s(d) < LP_{old} \end{cases} \qquad \alpha = \frac{y}{(y+1)} \qquad \textbf{(1)}$$

Figure 1 shows the steps of updating the locality parameter *LP* by the source node after receiving the route reply so the source node 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 α from approaching 1 as y gets bigger due to $\lim_{\alpha \to \infty}(\alpha) = 1$, 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 its maximum value, *max-y*. Each time y is initialised to 1, the partial historical information

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

```
If y >=max-y then
    y = Initial-y
End if
α = y/(y+1)
LP_new = αLP_old + (1−α)h_s(d)
If h_s(d) < LP_old then
    LP_new = Floor(LP_new)
Else
    LP_new = Ceiling(LP_new)
End if
LP_old = LP_new
y = y+1
```

Figure 1. Update procedure for the locality parameter LP at the source node in TLRDA-D.

represented by LP_{old} is given the same weight as the hop count. Alternatively, if *y* initialised by zero all the weight is given to the hop count.

In TLRDA-D, D stands for a delay where TLRDA-k denotes an instant of the algorithm where the delay equals to k units of time. Intermediate nodes in TLRDA-D broadcast route requests according to the on-demand routing algorithm used while route requests propagating within the neighbourhood boundary. However, beyond this boundary TLRDA-D broadcasts route requests with a delay at each node until the route request broadcast fades or the time to live (TTL) reaches zero.

The motive for adding this delay in the beyond-neighbourhood region is to give higher priority to route requests that are broadcasted within their own source node's neighbourhood regions. Moreover, other route requests that are travelling within their source node's beyond neighbourhood regions have higher chance of being already fulfilled thus they are given lower priority. This approach not only improves the average route discovery time but also improves the latency of the whole network, as it generates less contention throughout the network.

The delay should be calculated by monotonic non- decreasing function as the route request propagates further within beyond neighbourhood region, since the chance of route request fulfilment increases with each hop when the route request moves away from the source node's neighbourhood region. The delay increment can be logarithmic, linear, polynomial, or exponential. However, the exponential increase yields a huge amount of delay that may affect the discovery time if route finder is within the beyond-neighbourhood region which makes it unsuitable for resource-sensitive environment like MANETs and

Copyright © 2009 SciRes.

hence ruled out.

The simulation is used to help us decide on the amount of delay that needs to be imposed to the route request dissemination in the beyond-neighbourhood region for TLRDA-D and whether it should be logarithmic, linear or polynomial. TLRDA-D has been implemented using five different amounts of delay (d_i) where d_i at any intermediate node takes the following values:

$$d_i = \begin{cases} \log_2(LP) & i = 0 \\ 2^{i-1} LP & i = 1,2,3 \\ LP^2 & i = 4 \end{cases} \quad \textbf{(2)}$$

In TLRDA-D, upon receiving a route request; each node performs the steps shown in Figure 2. If the route request has been received before then it is considered redundant and thus discarded. Otherwise, the receiving node compares *LP* value from the route request packet with the hop count after counting itself as an extra hop, if the node resides in the beyond-neighbourhood region of the route request initiator then the node holds the route request for *d* units of time then processes it. Otherwise, the node processes the route request according to the routing algorithm used.

If a route reply is not received within an estimated period of time called NETwork Traversal Time (*NETTT*), the source node will try again to discover the route by broadcasting another route request for a maximum number of tries. So the source node waits *NETTT* units of time to receive a reply before trying to search for the destination again. The worst case scenario is assumed and Node Traversal Time (*NTT*) follows the on-demand routing algorithm used in a network with diameter of *D* hops. TLRDA-D calculates this estimated time as:

$$NETTT = 2\{(LP * NTT) + (D - LP)(NTT + d_i)\} \quad \textbf{(3)}$$

Step preformed by each node upon receiving a route request in TLRDA-D
1: If i = 0 then $d = \log_2(LP_r)$
3: End if i = 1 then $d = LP_r$
4: End if i = 2 then $d = 2*LP_r$
5: End if i = 3 then $d = 4*LP_r$
6: End if i = 4 then $d = LP_r * LP_r$ end if
7: End if
8: End if
9: End if
10: End if
11: If route request is a duplicate
12: Discard the route request
13: Else
14: If hop_count > LP_r then
15: Wait *d* units of time
16: End if
17: Process the route request
18: End if

Figure 2. Route request messages processing at each node for TLRDA-D.

In on-demand routing algorithms, when an intermediate node *m* receives a route request for the first time; it stores: the broadcast ID and the route request originator IP address in its routing table, if it has such a table, for a an estimated time Broadcast Cache Time (*BCT*) as part of the route request processing steps. This information is used to distinguish between new and redundant route requests. When *BCT* expires, the route request record is deleted from the routing table. TLRDA-D calculates the time as:

$$BCT = \begin{cases} BCT & h_s(m) \le LP_r \\ BCT + d_i & h_s(m) > LP_r \end{cases} \quad \textbf{(4)}$$

3.1. Simulation Analysis

A simulation has been conducted to evaluate the new algorithm, TLRDA-D, and compare it with AODV. TLRDA-D algorithm was implemented as a modification to AODV implementation in NS2 network simulator, version 2.29 [19]. NS2 was used to conduct extensive experiments for performance evaluation and comparison.

Mobile nodes are assumed to operate in a squared simulation area of 1000m × 1000m. The transmission range is fixed to 100m in all nodes to approximately simulate networks with a minimum hop count of 10 hops between two border nodes one on opposite sides in a connected network. 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 then averaged to produce the graphs shown throughout this paper and a 95% confidence interval is shown as standard error bars in the relevant figures. Table 1 provides a summary of the chosen simulation parameter values.

The comparison metrics include:

• End-to-end delay: the total delay for the application data packet while transmitted from source to destination plus the route discovery time which is the round trip time from sending a route request until receiving the route reply.

• Packet loss: the number of dropped packets in a single run.

• Route request overhead: measured by the number of received route requests in the whole network.

A traffic generator was used to simulate constant bit rate (CBR) with payload of 512 bytes. Moreover, each five communication sessions were simulated between one source and five destinations randomly selected in a group of ten nodes to simulate traffic in an application that exhibit traffic locality. Data packets are transmitted at a rate of four packets per second, assuming nodes are identical, links are bidirectional, and mobile nodes oper-

Copyright © 2009 SciRes.

ate in a flat arena.

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 [20]. However, a group mobility model may be used to simulate a coop erative 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 [21,22].

The RPGM mobility generator was used [23] to generate mobility scenarios for 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 with Speed Deviation Ratio and Angle Deviation Ratio = 0.5. Moreover, nodes move randomly within their group with a speed randomly selected between 1m/s and 15m/s with 50s as pause time. Each group contains 10 nodes.

In our simulation, we concentrate on varying three major parameters to study their effect on TLRDA-D performance: network size, traffic load, and maximum speed in three different cases by varying one parameter while keeping the other two constant.

Effect of network size: 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 (small size network) to 100 (moderate size network) with traffic load of 10 communication sessions and a maximum speed of 15m/s.

Figure 3 shows the superiority of TLRDA-D over AODV in reducing the end-to-end delay due to reducing congestion level especially when d_2, d_3 or d_4 is used as the amount of delay. For instance, in TLRDA-d_2, TLRDA-d_3, and TLRDA-d_4, the end-to-end delay was reduced by nearly 53% in small size network and by 68% in moderate size network compared to AODV. Moreover, this figure clearly shows that d_2, d_3 or d_4 yield in average almost the same end-to-end delay. The amount of delay added in TLRDA-d_2 was adequate to achieve the best discovery time in our scenarios as adding more delay will not yield further contention improvement. In average, route requests in TLRDA-D reside in the network for longer time than in the case of AODV (not shown here). This is due to the added delay which increases overhead yet reduces discovery time.

Figure 4 shows that TLRDA-D loses fewer packets compared to AODV by 1% to 30% in small size network and by 22% to 62% in moderate size network because TLRDA-D reduces congestion level. In TLRDA-D, the number of received route requests is more than that of AODV as shown in Figure 5. Some of the saved packets, gained in TLRDA-D as a result of reducing packet loss, are route requests which justify the increase in route request overhead. Those route requests might be duplicate copies but were dropped because of congestion or/and collision rather than redundancy. The rest of the saved packets can be any kind which might be useful but dropped in AODV due to high channel contention or collision. TLRDA-d_2, TLRDA-d_3, and TLRDA-d_4 lose fewer packets than TLRDA-d_0 and TLRDA-d_1 which improves network performance.

Effect of traffic load: Traffic load of sizes 5 (light traffic) to 35 (heavy traffic) communication sessions incremented by 5 were injected in networks of size seventy nodes and maximum speed of 15m/s. A reasonably incremented amount of traffic was used to test our algorithm meanwhile avoiding saturation.

Also in this analysis, when TLRDA-D uses d_2, d_3 or d_4 as amount of delay, the algorithm yield in average almost the same end-to-end delay as depicted from Figure 4 for these three instances among all experimented instances

Table 1. System parameters.

Parameter	Value
Transmission range	100m
Topology size	1000×1000m
Simulation time	900s
Packet size	512 bytes
Packet rate	4pkt/s
Traffic load	5,10,…,35 sessions
Traffic type	CBR(UDP)
Antenna type	Omni Antenna
MAC protocol	IEEE 802.11with RTS/CTS
Maximum speed	2,5,7,10,13,15m/s
Minimum speed	1m/s
Pause time	50s
Mobility model	RPGM model
SDR, ADR	0.5
Propagation model	Two-Ray Ground model

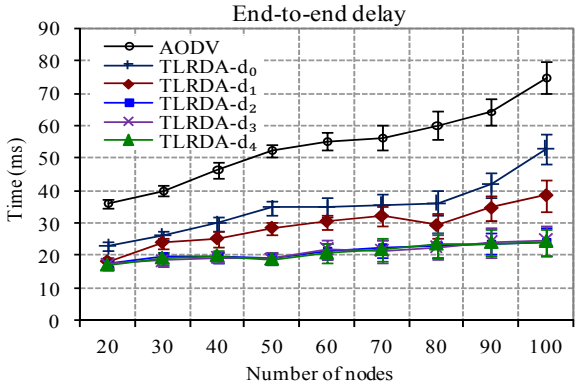

Figure 3. End-to-end delay verses network size for networks of 10 communication sessions and 15m/s as maximum speed.

Copyright © 2009 SciRes.

of TLRDA-D. The end-to-end delay was reduced by nearly 57% in light traffic and 65% in heavy traffic for TLRDA-d_2, TLRDA-d_3, TLRDA-d_4 compared to AODV. So, TLRDA-D has end-to-end delay lower than AODV from traffic load prospective.

Furthermore, TLRDA-d_2, TLRDA-d_3, and TLRDA-d_4 have almost the same end-to-end delay that is lower compare to both TLRDA-d_0 and TLRDA-d_1. This improvement in the end-to-end delay is due to the reduction in channel contention where the application data can travel earlier and quicker which improves the network performance. Moreover, TLRDA-D reduces packet loss in the whole network compared to AODV as shown in Figure 7. This improvement in TLRDA-D over AODV ranges from 3% to 65% in light traffic while it ranges between 10% and 53% in heavy traffic.

The packet loss is nearly the same for the three instances TLRDA-d_2, TLRDA-d_3, and TLRDA-d_4 and better than both TLRDA-d_0 and TLRDA-d_1. Also in this analysis, some of these saved packets in TLRDA-D might be route requests which justify the increment in route request overhead in TLRDA-D over AODV as in Figure 6.

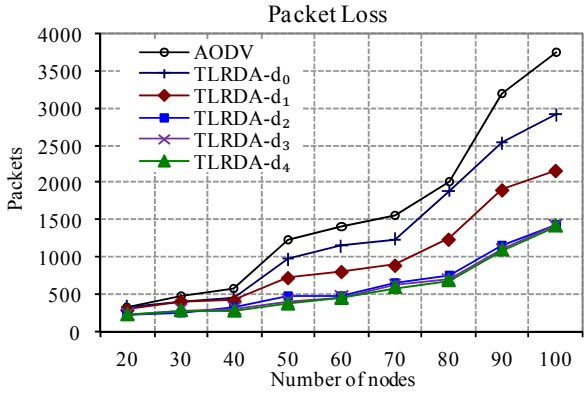

Figure 4. Packet loss verses different number of nodes for networks of 10 communication sessions and 15m/s as maximum speed.

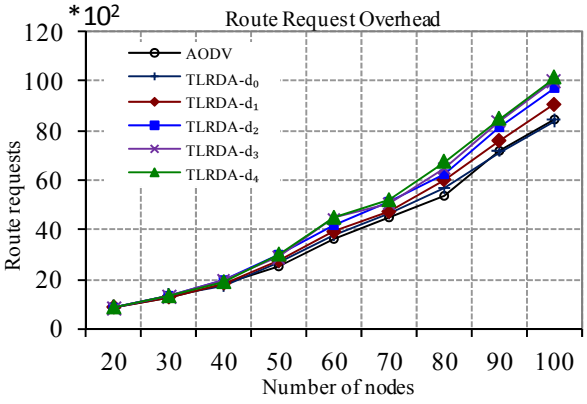

Figure 5. Route request overhead verses different number of nodes for networks of 10 communication sessions and 15m/s as maximum speed.

Effect of mobility: The value of the maximum speed where 2m/s used as slow speed and 15m/s is fast speed. The end-to-end delay in TLRDA-D is reduced com can be 2, 5, 7, 10, 13, or 15m/s with networks of 70 nodes and traffic load of 10 communication sessions where 2m/s used as slow speed and 15m/s is fast speed. The end-to-end delay in TLRDA-D is reduced compared to AODV for different maximum speed as in Figure 9 where discovery time increases in both TLRDA-D and AODV with fast speed because speed affects routes and may result in broken links. This figure reveals the difference in the end-to-end delay among all five instances of TLRDA-D where TLRDA-d_2, TLRDA-d_3, and TLRDA-d_4 reduce end-to-end delay more than TLRDA-d_0 and TLRDA-d_1.

TLRDA-D reduces packet loss compared to AODV as shown in Figure 8. Packet loss increases with faster movements in both algorithms. TLRDA-D improves packet loss over AODV by 14% to 87% in slow speed and by 21% to 62% in fast speed. Moreover, these packets include route requests which increases route request overhead in TLRDA-D over AODV as shown in Figure 9.

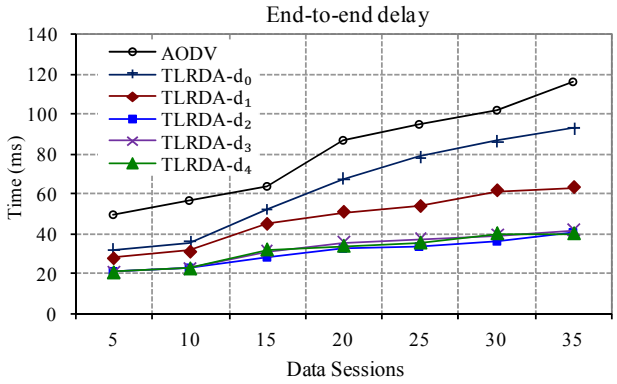

Figure 6. End-to-end delay versus traffic load with a network70 nodes and 15m/s as maximum speed.

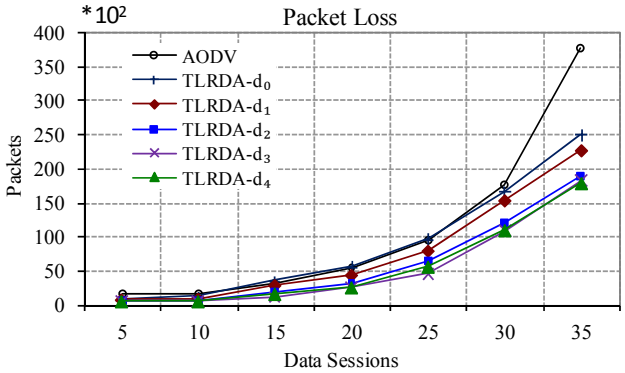

Figure 7. Packet loss versus traffic load with a network70 nodes and 15m/s as maximum speed.

Copyright © 2009 SciRes.

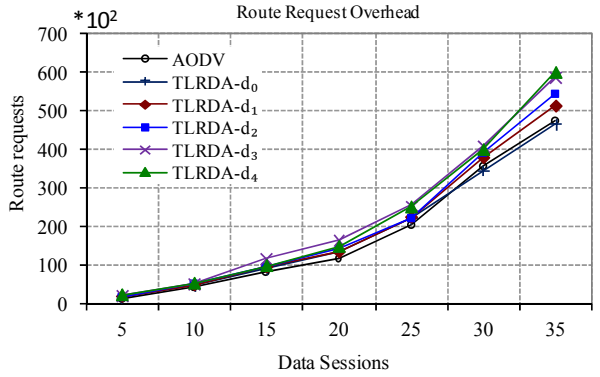

Figure 8. Route request overhead versus traffic load with a network70 nodes and 15m/s as maximum speed.

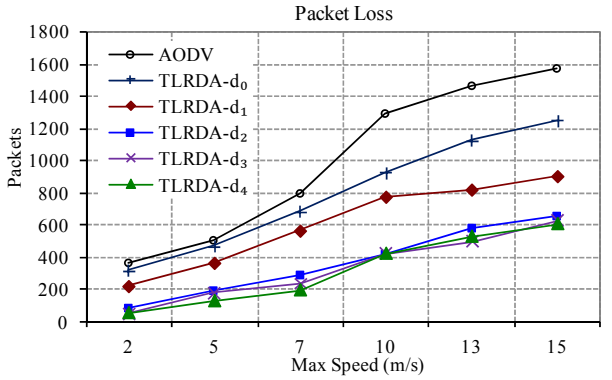

Figure 10. Packet loss versus maximum speed in networks of 70 nodes and 10 communication sessions.

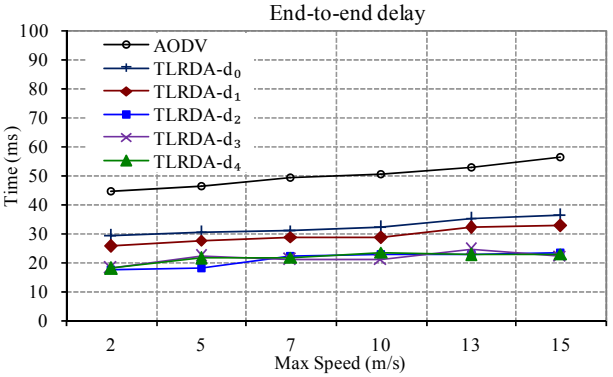

Figure 9. End-to-end delay versus maximum speed in networks of 70 nodes and 10 communication sessions.

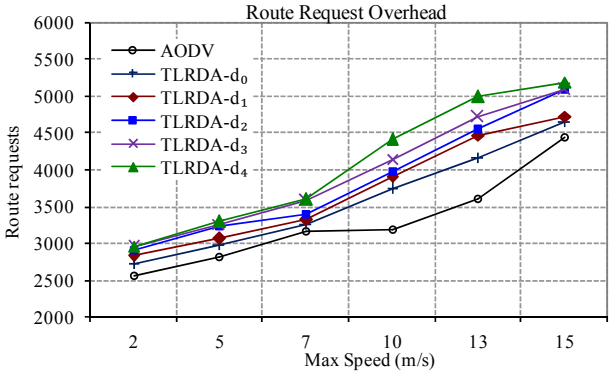

Figure 11. Route request overhead versus maximum speed in networks of 70 nodes and 10 communication sessions.

Both algorithms have almost the same number of transmitted route request; so extra route requests received in TLRDA-D might be duplicate copies but were dropped because of congestion or collision. Furthermore, the number of saved packets is greater than the increment in route requests overhead where the minimum difference ranges from 8% to 70% in slow speed and from 16% to 45% in fast speed. The extra saved packets can be any kind of packets which might be useful but dropped in AODV due to many reasons i.e. contention, congestion or collision. These saved packets in TLRDA-D have a good impact on network performance.

In summary, TLRDA-D reduces discovery time, packet loss, and end-to-end delay over AODV. However, it increases route request lifetime in justifiable manner. The best delay function would be a linear one. In particular, for the considered scenarios in our experimental study the doubling function $d_2 = 2LP_r$ gave the best performance among all scenarios performed in this study. It is worth mentioning that TLRDA-D reduces end-to-end delay despite the fact that it works by delaying, by definition, route request within their source node's beyond-neighbourhood region.

4. Conclusions

When on-demand routing algorithms for MANETs run applications that exhibit traffic locality, the route discovery process can be improved by utilising the traffic locality concept. We introduce a traffic locality oriented route discovery algorithm with delay, TLRDA-D. It works by establishing a neighbourhood that includes the most likely destinations for a particular source node. The source node broadcasts the route request without adding any delay within its neighbourhood boundary. In an effort to improve the route discovery process for MANETs that exhibit traffic locality. This adaptive route discovery algorithm gradually build up the node neighbourhood as a region, with the ability to change, centred at the source node and expected to contain most of the members of its working set. Furthermore, TLRDA-D adds a delay to route requests travelling within their beyond-neighbourhood region to reduce channel contention which reduces the discovery time of other route requests. One of the main advantages of TLRDA-D is improving route discovery process which improves the end-to-end delay as it generates less channel contention throughout the network

Copyright © 2009 SciRes.

which reduces packet loss. We have analysed TLRDA-D using simulation to study the affect of adding a delay to route request propagation and to decide on the proper amount of delay to be added. The simulation analysis showed that when TLRDA-D uses twice the locality parameter as a delay, it gave the best improvement among the experimented scenarios.

5. References

[1] S. Murthy and B. Manoj, "Ad hoc wireless networks: Architectures and protocols," Prentice Hall, 2004.

[2] A. Tanenbaum, "Computer networks," Pearson Education, 2003.

[3] M. Abolhasan, T. Wysocki, and E. Dutkiewicz, "A review of routing protocols for mobile ad hoc networks," Ad Hoc Networks, Vol. 2, No. 1, pp. 1−22, 2004.

[4] C. Adjih, T. Clausen, P. Jacquet, *et al*., "Optimized link state routing protocol," The Internet Engineering Task Force, IETF, RFC 3626, 2003.

[5] S. R. Das, R. Castaneda, Y. Jiangtao, *et al*., "Comparative performance evaluation of routing protocols for mobile ad hoc networks," pp. 153−161, 1998.

[6] 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, April 2003.

[7] C. Perkins, E. Belding-Royer, and S. Das, "AODV ad hoc on-demand distance vector routing," The Internet Engineering Task Force, IETF, RFC 3561, July 2003.

[8] Z. J. Haas, M. R. Pearlman, and P. Samar, "The Zone Routing Protocol (ZRP) for ad hoc networks," IETF MANET Working Group, INTERNET-DRAFT, July, 2002.

[9] P. Denning, "The working set model for program behavior," Communications of the ACM, Vol. 11, No. 5, pp. 323−333, 1968.

[10] M. Shikharesh and B. B. Richard, "Measurement and analysis of locality phases in file referencing behaviour," Proceedings of the ACM SIGMETRICS Joint International Conference on Computer Performance Modelling, Measurement and Evaluation, Raleigh, North Carolina, United States, 1986.

[11] P. Denning, "The locality principle," Communications of the ACM, Vol. 48, No. 7, pp. 19−24, 2005.

[12] C. Kozierok, "The TCP/IP guide," 1st Edition, No Starch Publishing, 2005.

[13] A. Silberschatz, P. Galvin, and G. Gagne, "Operating systems concepts," 7th Edition, John Wiley & Sons, 2005.

[14] F. Borgonovo, "ExpressMAN: Exploiting traffic locality in expressnet," IEEE Journal on Selected Areas in Communications, Vol. 5, No. 9, pp. 1436−1443, 1987.

[15] M. Prasant and K. Srikanth, "Ad hoc networks: Technologies and protocols," Springer-Verlag New York, Inc., 2004.

[16] J. Y. Li, C. Blake, D. S. J. De Couto, H. I. Lee, and R. Morris, "Capacity of ad hoc wireless networks," Proceedings of the 7th Annual International Conference on Mobile Computing and Networking, Rome, Italy, 2001.

[17] M. Al-Rodhaan, L. Mackenzie, and M. Ould-Khaoua, "A traffic locality oriented route discovery algorithm for MANETs," Ubiquitous Computing and Communication Journal (UBICC), Vol. 2, No. 5, pp. 58−68, 2007.

[18] M. Mosko and J. Garcia-Luna-Aceves, "Performance of group communication over ad hoc networks," Proceedings of the IEEE International Symposium Computers and Communications ISCC, Italy, pp. 545−552, 2002.

[19] K. Fall, "NS notes and documentation," in The VINT Project, 2000.

[20] D. B. Johnson and D. A. Maltz, "Dynamic source routing in ad hoc wireless networks," in Mobile Computing, I. A. Korth, Ed., Kluwer Academic Publishers, Norwell, MA, Vol. 353, pp. 153−181, 1996.

[21] F. Bai, N. Sadagopan, B. Krishnamachari, *et al*., "Modeling path duration distributions in MANETs and their impact on reactive routing protocols," IEEE Journal on Selected Areas in Communications, Vol. 22, No. 7, pp. 1357−1373, 2004.

[22] X. Y. Hong, M. Gerla, G. Y. Pei, and C.-C. Chiang, "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, Seattle, Washington, United States, 1999.

[23] F. Bai, N. Sadagopan, and A. Helmy, "IMPORTANT: An evaluation framework to study the impact of mobility patterns on routing in ad-hoc NETworks," University of Southern California, 2005.

Copyright © 2009 SciRes.