TLRDA-D: Traffic Locality Oriented Route Discovery Algorithm with Delay

Mznah A. Al-Rodhaan[1], Lewis Mackenzie[2], Mohamed Ould-Khaoua[3]

ABSTRACT- **Communication 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. In MANETs, 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 are introducing 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. Within the neighbourhood, each node broadcasts the route request according to the original routing used. Beyond this boundary, each intermediate node broadcasts the route request with a delay to give a higher 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 both mathematical and simulation modelling to study the effect of adding different amount of delays to route request propagation and to decide on the most suitable quantity of added delay.**

Index Terms-MANETs, Route Discovery, Delay, Congestion.

I. 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 [26] or infrastructure-less multi-hop such as Mobile Ad hoc NETworks (MANETs) [26, 33]. 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 [33].The design of an efficient routing strategy is a very challenging issue due to the limited resources in MANETs [26]. MANETs routing protocols can be divided into three categories: proactive, reactive, and hybrid [1]. 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) [2]. 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 [12]. Examples of this class are Dynamic Source Routing (DSR) [21] and Ad Hoc On Demand Distance Vector (AODV) [27]. 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) [17] 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 analysed using mathematical and simulation modelling 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 environment and observation. Finally, Section 4 concludes this study.

II. RELATED WORK

The principle of locality was first applied in memory referencing behaviour [14] then it was subsequently observed in the use of other resources such as file referencing [31]. The locality of reference concept deals with the process of accessing a single resource more than once. It includes spatial and temporal locality [13, 24]. 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 [32]. The importance of traffic locality concept is recognized in networking. Traffic locality concept is a motivation factor behind network clusters and workgroups [10]. While in infrastructure wireless networks, traffic locality is utilized to improve load balancing in base stations [26, 28]. 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 [20]. Sometimes a node communicates with a certain set of nodes more than others within a particular time regardless of their locations, highlighting the

[1] Information Technology Department, College of Comp. and Info. Sci., King Saud University, Riyadh, Saudi Arabia.
[2] Department of Computing Science, University of Glasgow, Glasgow, UK G12 8QQ, Email: lewis@dcs.gla.ac.uk.
[3] Department of Electrical & Computer Engineering, Sultan Qaboos University, Al-Khodh, 123, Muscat, Oman, Email: mok@squ.edu.om.

1999-4974©2010BUJICT

traffic locality [6].

III. 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 [25] where a group of nodes communicate to accomplish a common goal.

In this paper, traffic locality concept [3, 6] is investigated more to improve the route discovery process in on-demand routing protocols for MANETs. Our algorithm, TLRDA-D [4, 5] 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 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, $h_s : N \rightarrow \mathbb{Z}^+ \cup \{0\}$ where $h_s(u)$ is the hop count between s and some other node $u \in N$ and $h_s(s) = 0$. A node, *x*, is considered to be part of the working set of a source node, s, if $h_s(x) \leq LP$. 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 $\tau_1 \cap \tau_2 = \emptyset$. Let us consider a source node s, any node $v \in \tau_1$ satisfies the condition $h_s(v) \leq LP_r$ 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 $h_s(d) \geq LP_{old}$ then the neighbourhood of s expands; otherwise it shrinks.

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

Fig. 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 $\lim_{\alpha \rightarrow \infty}(\alpha) = 1$, 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 represented by LP_{old} is given the same

1999-4974©2010BUJICT

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.

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

If y >= *max-y* then
    y = *Initial-y*
End if
$\alpha = y/(y+1)$
$LP_{new} = \alpha LP_{old} + (1 - \alpha)h_s(d)$
If $h_s(d) < LP_{old}$ then
    $LP_{new} = \text{Floor}(LP_{new})$
Else
    $LP_{new} = \text{Ceiling}(LP_{new})$
End if
$LP_{old} = LP_{new}$
y = y+1

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

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 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:

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

In TLRDA-D, upon receiving a route request; each node performs the steps shown in Fig. 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.

Steps preformed by each node upon receiving a route request in TLRDA-D

If i = 0 then $d = \log_2(LP_r)$
Else if i = 1 then $d = LP_r$
    Else if i = 2 then $d = 2*LP_r$
        Else if i = 3 then $d = 4*LP_r$
            Else if i = 4 then $d = LP_r * LP_r$ end if
            End if
        End if
    End if
End if
If route request is a duplicate
    Discard the route request
Else
    If hop_count > LP_r then
        Wait d units of time
    End if
    Process the route request
End if

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

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:

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

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:

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

A. Delay analysis

All packets (data or control) are subject to different amounts of delay while travelling from source to destination in any network such as queuing delay, processing delay, and propagation delay...etc. These delays depend on many factors such as: energy level, packet length, and contention level at that particular time. Propagation delay between two adjacent nodes is assumed to be negligible in this analysis since packets in wireless communications travels by the speed of light where propagation delays$=\frac{\text{distance}}{(3*10^8)}$. However, other delays affect the network performance.

In MANETs, most of the delays experienced by a message are in the MAC layer due to contention. MAC protocol does not distinguish between data and control packets because there is no priority used when using DCF in IEEE 802.11 standards [30]. There is one queue in the MAC layer where all packets (data or control) are queued and process as FCFS (FIFO). MAC layer protocol has no knowledge about the importance of the data coming from higher layer such as control packets which are treated as normal payloads.

When the network gets congested, the channel contention increases which in turn increases the system delay and incurs more packet loss leading to network performance degradation and maybe saturation. Vice versa, lowering network congestion reduces the system delay. Understanding the relationship between congestion and delay in any network is essential especially in a resource-limited environment like MANETs. Thus analyzing our system using queuing theory provides us with better understanding of the congestion and the delay in such systems.

MANET can be modelled analytically as a network of queuing systems [16, 19] because a mobile node receives different kind of messages (data or control packet) with different lengths, queues them if needed, processes them then transmits them. The whole system can be modelled as a network of queuing systems operating in steady state. For simplicity, the role of mobility is ignored in this delay analysis and two assumptions were made: 1) Packet generation and arrival at each node assumed to be an independent and identically distributed. 2) Each node has infinite buffers to avoid dropped packets. So each node is modelled as M/G/1 system [9, 23] that satisfies the following conditions: 1) service delays are independent and have a general distribution because packets differ in size. 2) Packets arrive at each node according to a Poisson process with rate λ and independent of service time.

The system has a single server that serves packets in their order of arrival (FCFS). When the packet is ready to be transmitted, the node senses the shared physical media before attempting to transmit by performing the CSMA/CA access protocol at the MAC layer, so this contention time is included in the service time. Nodes in TLRDA-D are modelled as M/G/1 systems with different arriving customers such as data or control packets. Delay analysis is conducted for route requests which are divided into two classes:

Class 1: Contains route requests propagating within their source node neighbourhood region.

Class 2: Contains route requests propagating within their source node beyond-neighbourhood region.

Route requests travelling within the beyond-neighbourhood region are stored for d units of time before joining Class 1 queue where they are treated as Class 1 packets. To simplify the analysis, let us assume that separate buffers are maintained for Class 2 before joining the queue of Class 1. When the server is free and Class 1 is nonempty, the first packet in Class 1 queue enters the service. Fig. 3 shows a representation of a node as a queuing system running TLRDA-D on top of the on-demand routing algorithm used.

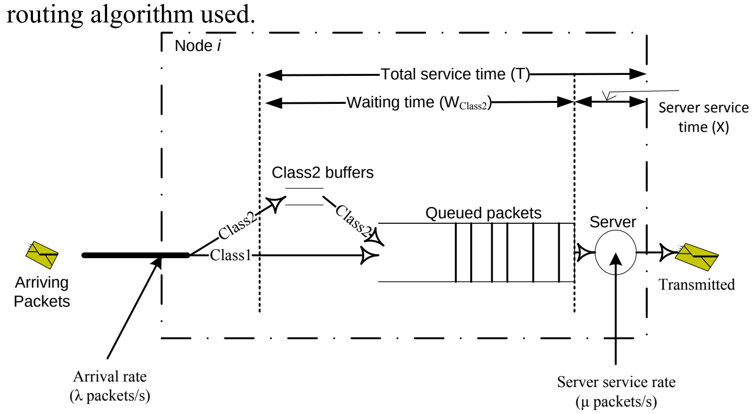

Fig. 3: A mobile node in MANETs represented as a queue for TLRDA-D.

According to TLRDA-D, when a route request propagates within the beyond-neighbourhood region; an extra amount of delay should be added to give other packets a better chance of being transmitted earlier and to reduce the contention at each node. Delaying route request packets when propagating in the beyond-neighbourhood region should not affect the discovery time of this route since most of the destinations of the route request lay within the neighbourhood region. In fact, fulfilled route requests compete with other packets to win the channel adding undesirable contention and should be given lower priorities over other data or control packets. To calculate the average waiting time in the queue for M/G/1 queue, a simple method from [9] is used. The notations used to perform the delay analysis for TLRDA-D are explained in Table 1.

In MANETs, packets are jittered by a random duration [11] before being broadcasted at each node. Random Rebroadcast Delay (RRD) is used to prevent broadcast synchronization. The extra amount of delay imposed to route requests in the beyond-neighbourhood is independent of RRD. Moreover, RRD

is used with broadcasted packets whereas unicast packets face another kind of delay due to the handshaking mechanism; just for simplicity we assume that these two delays are equal.

TABLE 1

Parameters of the Queuing Network Model for TLRDA-D.

N_Q	Average number of packets waiting in the queue
N_i	number of packets waiting in the queue when the i^{th} packet arrives
λ	Arrival rate
μ	Service rate
ρ	Utilization factor of the server ($\rho < 1$)
W_{Class1}	Average packet waiting time in the queue for Class 1
W_{Class2}	Average packet waiting time in the queue for Class 2
T_{Class1}	Average of total service time per packet for Class 1 packets
T_{Class2}	Average of total service time per packet for Class 2 packets
w_i	Waiting time for the i^{th} packet in the queue
R	Average residual service time
r_i	Residual service time is the remaining time of the packet currently in service when the i^{th} packet arrived
X	Average server service time
x_i	Server service time for the i^{th} packet
M	Average amount of jitter added to any broadcast packet
m_i	jitter added to the i^{th} broadcast packet
d	Amount of delay added to Class 2 packets

Class 1 packets

Class 1 contains route requests propagating in their source node's neighbourhood region. TLRDA-D processes Class 1 packets according to routing algorithm used. The service time for any packet $(x_1, x_2, \cdots)$ is a discrete random variable where the average service time $\overline{X} = 1/\mu$ and $E\{X\} = \overline{X}$.

So the average waiting time in the queue for the ith route request (w_i) is consisting of service times (x_j) of the packets currently waiting in the queue, residual time (r_i), and RRD (m_i):

$$w_i = \sum_{j=i-N_i}^{i-1} x_j + r_i + m_i \quad (5)$$

Since M is discrete random variable, the k moment, $\overline{M^k}$, of the jitter time is computed as

$$E\{M^k\} = \sum_m P(m)\, m^k$$

$$E\{w_i\} = E\left\{\sum_{j=i-N_i}^{i-1} E\left\{x_j \middle| N_i\right\}\right\} + E\{r_i\} + E\{m_i\} \quad (6)$$

Knowing that N_i is a random variable and independent of x_j.

$$E\{w_i\} = \overline{X} E\{N_i\} + E\{r_i\} + E\{m_i\} \quad (7)$$

Following the analysis in [9] where all long-term averages viewed as limits when packet index converges to infinity, assuming these limits exist. This assumption is true if $\rho < 1$. In other words, the arrival rate $(\lambda) <$ the service rate (μ) so the node can handle the packet received in reasonable time and avoid the unpleasant effect of saturation [23].

$$\lim_{i\to\infty} E\{w_i\} = \overline{X} \lim_{i\to\infty} E\{N_i\} + \lim_{i\to\infty} E\{r_i\} + \lim_{i\to\infty} E\{m_i\} \quad (8)$$

$$W_{Class1} = \overline{X} N_Q + R + M \quad (9)$$

Applying Little's Theorem as in [9]

$$N_Q = \lambda W_{Class1} \quad (10)$$

Substituting equation (10) in (9) and using $\rho = \overline{X}\lambda$:

$$W_{Class1} = \rho W_{Class1} + R + M \quad (11)$$

$$W_{Class1} = \frac{R + M}{(1 - \rho)} \quad (12)$$

Where the average residual time as stated in [9] is:

$$R = \frac{\lambda\, \overline{x^2}}{2} \quad (13)$$

The second moment $(\overline{x^2})$ of service time is computed as in [29]:

$$E\{X^2\} = \sum_{x_i} P(x_i)\, {x_i}^2$$

The average of waiting time formula can be obtained similar to [9, 23] by substituting (13) into (12):

$$W_{Class1} = \frac{\lambda \overline{x^2} + 2M}{2\ (1-\rho)} \quad (14)$$

Total service time for a Class 1 packet can be obtained from adding the waiting time in queue to the average server service time and the waiting time for the channel to be free.

$$T_{Class1} = W_{Class1} + X \quad (15)$$

Class 2 packets

When route requests travel in the beyond-neighbourhood region, they are delayed for d units of time at each node in this region. The average waiting time in the queue for the ith route request is:

$$w_i = \sum_{j=i-N_i}^{i-1} x_j + r_i + m_i + d \quad (16)$$

Following the same analysis from equation (5) to (11):

$$W_{Class2} = \rho W_{Class2} + R + M + d \quad (17)$$

And by substituting R from equation (13) in (17):

$$W_{Class2} = \frac{\lambda \overline{x^2} + 2M + 2d}{2(1 - \rho)} \quad (18)$$

In average, route requests that belong to Class 2 will experience a delay equal to the delay of other packets from Class 1 plus an extra amount of delay.

$$W_{Class2} = W_{Class1} + \frac{d}{(1 - \rho)} \quad (19)$$

The total service time per packet for Class 2 (T_{Class2}) is the waiting time in the queue (W_{Class2}) plus the average server service time (X) which includes the waiting time for the line to be free.

$$T_{Class2} = W_{Class2} + X \quad (20)$$

The average waiting time of T_{Class2} is more than T_{Class1} by $d/(1 - \rho)$ units of time. This increment should not affect the end-to-end delay of the network due to the fact that the delay is added in the beyond-neighbourhood region.

MANETs consist of N nodes and each node process different kind of packets where T_{Class1} and T_{Class2} represent packet's delays within each node. Analyses of the route request latency and end-to-end delay are done for the whole network where stations of the queuing network corresponded to the nodes in MANET as in Fig. 4.

This delay reduces network congestion so the average service time, X, and the average waiting times W_{Class1} and W_{Class2} are reduced in both T_{Class1} as well as T_{Class2} compared to high congested network which indeed improve the network performance. Network congestion is reduced when adding more delay to route requests propagation which reduces channel contention but to a certain extent because there are other factors affecting channel contention such as fading and transmutation errors. Let us denote channel contention due to congestion by C where $0 \leq C \leq C_{max}$ while contention due to other factors by C_{min} thus the total channel contention is $C_{min} + C$. So the total channel contention in a heavy congested network is $C_{max} + C_{min}$ such as a network using a routing algorithm that uses simple flooding. However, the total channel contention in a well controlled network is equal to C_{min}.

End-to-end delay analysis:

The end-to-end delay is the route discovery time plus the average delay experienced by the data packet from the time it is sent by the source node until it is received at the destination. Route discovery time is the round trip time of route request and route reply between source node and finder of the needed route.

Route discovery time (RDT) for one route request in TLRDA-D is analysed as:

$$RDT = \begin{cases} T_{Class1} & h_s(f) = 0 \\ 2T_{Class1} h_s(f) & h_s(f) \leq LP_r \\ T_{Class1}(LP_r + h_s(f)) + T_{Class2}(h_s(f) - LP_r) & h_s(f) > LP_r \end{cases} \quad (21)$$

So the end-to-end delay can be calculated as:

$$End - to - end\ delay = RDT + T_{Class1} * h_s(d) \quad (22)$$

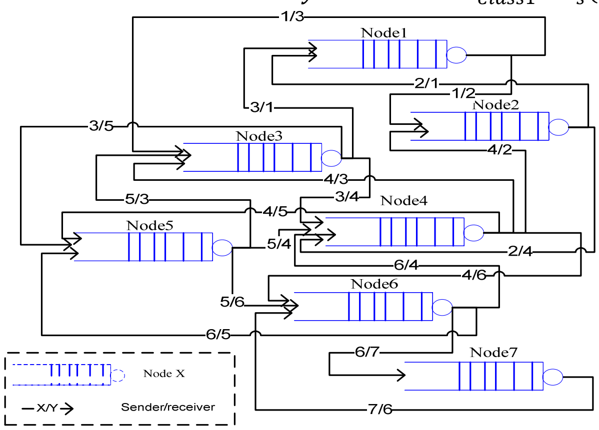

Fig. 4: A mobile ad hoc network of size seven represented as a network of queuing systems.

Route request latency analysis:

Assuming that the route request propagates through the network until it is faded or TTL reaches zero where the diameter of the network is D. Route Request Lifetime (RRL) can be calculated as follow where node B is the nearest boundary node:

$$RRL = \begin{cases} T_{Class1} & h_s(f) = 0 \\ T_{Class1} LP_r + T_{Class2}(\min\{D, h_s(B)\} - LP_r) & h_s(f) > 0 \end{cases} \quad (23)$$

The average route request latency per hop, if $h_s(f) > 0$, RRL is divided by number of hop counts that the route request propagates through the network where RRL is calculated in equation (23). So Route Request Latency can be calculated as follows:

$$Route\ Request\ Latency = RRL / min\{D, h_s(B)\} \quad (24)$$

Comparison between TLRDA-D and AODV

When analysing the delay, the total service time is reduced as a result of the reduction in channel contention for all instance of TLRDA-D. In TLRDA-D, Class 1 and Class 2 packets total service times are reduced by $(C_{max} - C)$ units of time. To illustrate, let us consider $0 \leq C \leq 0.8$ i.e. $C_{max} = 0.8$ and $C_{min} = 0.2$. Since AODV uses simple flooding, the channel contention is very high which makes $C = 0.8$ also all AODV packets belong to Class 1 assuming that $T_{Class1} = 1$ for this algorithm. In TLRDA-D, when the delay added to route requests increases the channel contention decreases. In TLRDA-d_0, the amount of delay added is small which makes the reduction in channel contention small as well so we will assume $C = 0.7$. Since the delay in TLRDA-di is almost double the delay in TLRDA-d_{i-1} when the delay is linear, channel contention in TLRDA-d_i is assumed to be half the channel contention in TLRDA-d_{i-1} so the values for C are 0.34, 0.175, and 0.087 for TLRDA-d_1, TLRDA-d_2, and TLRDA-d_3 respectively. Since the delay in TLRDA-d_4 is very large, channel contention is reduced even more where $0 \leq C \leq 0.04$ so $C = 0.02$ is used for TLRDA-d_4. The values of LP calculated as $LP = h_s(f) + 1$. Moreover, Class 2 packet's total service time is calculated according to the values of the delay added because $T_{Class2} = T_{Class1} + d/(1-\rho)$ units of time assuming lightly loaded network where $\rho = 0.2$. Furthermore, the hop count is assumed to be the network size divided by 10. Networks are of sizes 20, 30… 100 nodes so hop counts are 2, 3… 10 for different sources and route finders under the same environment.

Fig. 5 shows that end-to-end delay increases with the increment of network size with all instances of TLRDA-D and AODV.

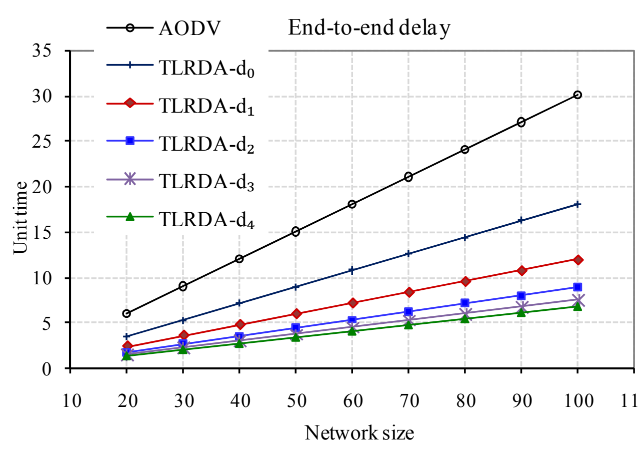

Fig. 5: End-to-end delay versus network size when $h_s(f) \leq LP_r$.

AODV discovers new routes later than all instances of TLRDA-D due to high contention. When the delay added to the route request dissemination is increased, the discovery time of the new route needed is improved because this delay is imposed after discovering the needed route. So TLRDA-d_4 gives the lowest end-to-end delay among all instances. Due to the delay added to the route request dissemination in the beyond-neighbourhood region, the average of route request latency increases in TLRDA-D more than AODV as shown in Fig. 6. Furthermore, route request latency increases with the increment of the delay added so TLRDA-d_4 gives the highest route request latency among all instances.

B. 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 [15]. 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 x1000m. 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.

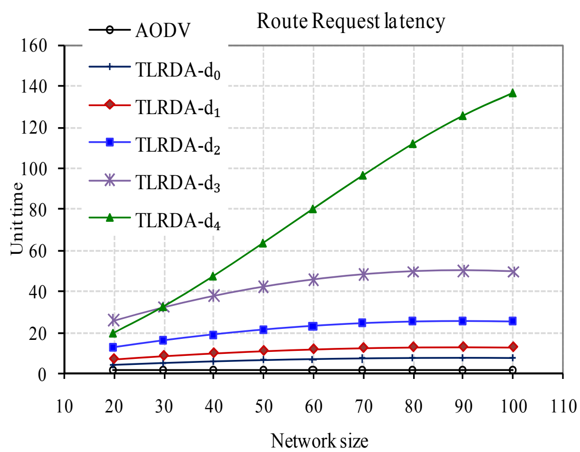

Fig. 6: Route request latency versus network size when $h_s(f) \leq LP_r$.

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 operate 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 [22]. However, 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 [8, 18].

TABLE 2:
System parameters

Parameter	Value
Transmission range	100m
Topology size	1000x1000m
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

The RPGM mobility generator was used [7] 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.

Fig.7 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.

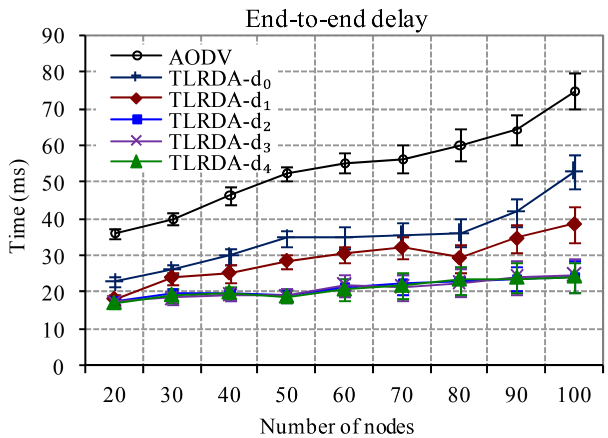

Fig. 7: End-to-end delay verses network size for networks of 10 communication sessions and 15m/s as maximum speed.

Fig. 8 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.

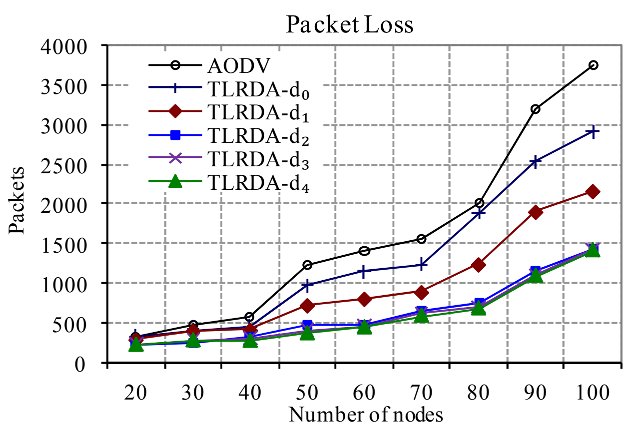

Fig. 8: Packet loss verses different number of nodes for networks of 10 communication sessions and 15m/s as maximum speed.

In TLRDA-D, the number of received route requests is more than that of AODV. 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 as in Fig. 9. 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.

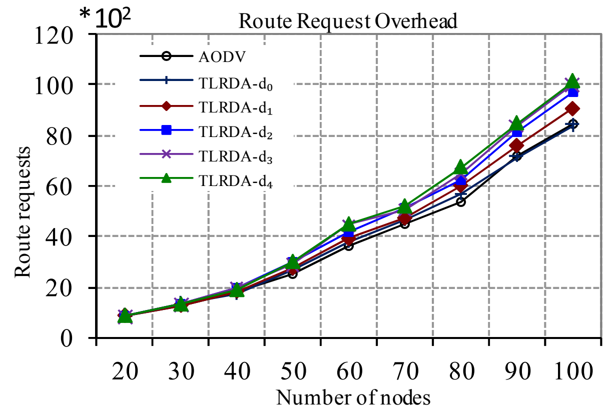

Fig. 9: Route request overhead verses different number of nodes for networks of 10 communication sessions and 15m/s as maximum speed.

<u>Effect of traffic load</u>: 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 Fig. 10 for these three instances among all experimented instances 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.

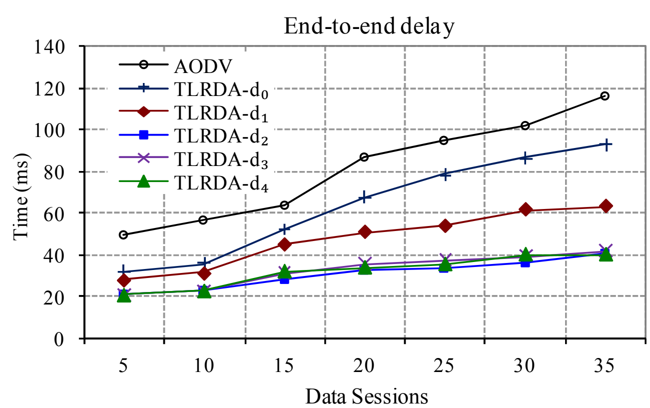

Fig. 10: End-to-end delay versus traffic load with a network70 nodes and 15m/s as maximum speed.

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 Fig. 11. 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.

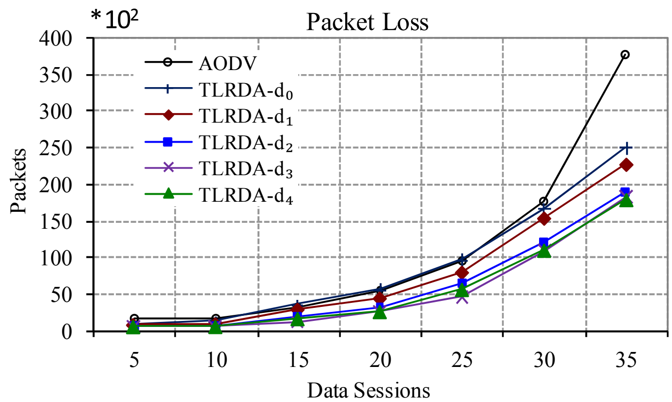

Fig. 11: Packet loss versus traffic load with a network70 nodes and 15m/s as maximum speed.

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.

Effect of mobility: The value of the maximum speed 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.

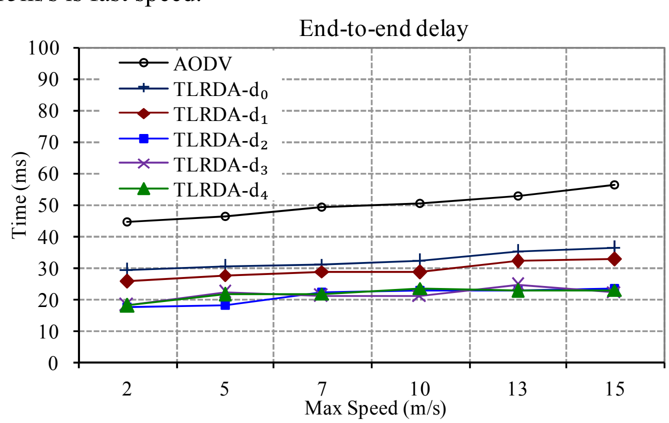

Fig. 12: End-to-end delay versus maximum speed in networks of 70 nodes and 10 communication sessions.

The end-to-end delay in TLRDA-D is reduced compared to AODV for different maximum speed as in Fig. 12 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 Fig. 13. 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

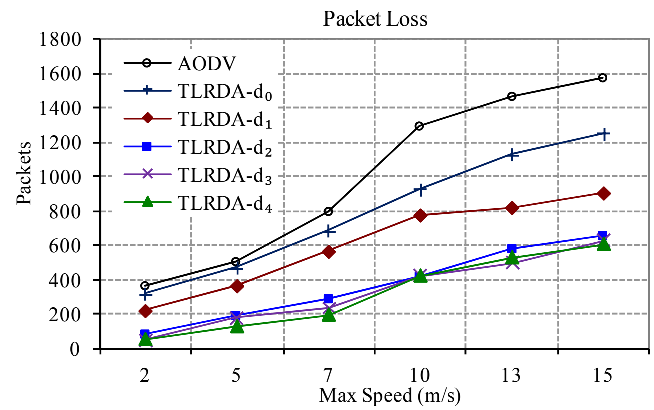

Fig. 13: Packet loss versus maximum speed in networks of 70 nodes and 10 communication sessions.

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.

IV. CONCLUSIONS

By utilising the traffic locality concept, the route discovery process can be improved in on-demand routing algorithms for

MANETs running applications that exhibit traffic locality. We have introduced a traffic locality oriented route discovery algorithm with delay, TLRDA-D, that works by establishing a neighbourhood which includes the most likely destinations for a particular source node. The source node broadcasts the route request immediately within its neighbourhood boundary 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 disseminating within their beyond-neighbourhood region to reduce channel contention which reduces the discovery time of other route requests.

TLRDA-D improves route discovery process which has a good impact over the end-to-end delay as it generates less channel contention throughout the network and reduces packet loss. We have analysed TLRDA-D using both mathematical simulation modelling to study the affect of adding a delay to route request propagation and to decide on the proper amount of delay to be added. Both analyses showed that when TLRDA-D uses twice the locality parameter as a delay, it gave the best improvement among the experimented scenarios.

REFERENCES

[1] M. Abolhasan, T. Wysocki, and E. Dutkiewicz, "A review of routing protocols for mobile ad hoc networks," *Ad Hoc Networks*, vol. 2, pp. 1-22, 2004.

[2] C. Adjih, T. Clausen, P. Jacquet, A. Laouiti, P. Minet, P. Muhlethaler, A. Qayyum, and L. Viennot, "Optimized Link State Routing Protocol," The Internet Engineering Task Force, IETF,RFC 3626, 2003.

[3] M. Al-Rodhaan, L. M. Mackenzie, and M. Ould-Khaoua, "On the Performance of Traffic Locality Oriented Route Discovery Algorithm with Chase Packets," *Bahria University Journal of Information & Communication Technology (BUJICT)*, vol. 1, pp. 9-16, 2008.

[4] M. Al-Rodhaan, L. M. Mackenzie, and M. Ould-Khaoua, "On the Performance of Traffic Locality Oriented Route Discovery Algorithm with Delay " *Int'l J. of Communications, Network and System Sciences*, vol. 2, pp. 203-210, 2009.

[5] M. Al-Rodhaan, L. M. Mackenzie, and M. Ould-Khaoua, "Traffic Locality Oriented Route Discovery Algorithm with Chase Packets for Mobile Ad Hoc Networks," presented at The 2009 International Conference on Wireless Networks (ICWN'09), Las Vegas, Nevada, USA,, 2009.

[6] 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, pp. 58-68, 2007.

[7] 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.

[8] F. Bai, N. Sadagopan, B. Krishnamachari, and A. Helmy, "Modeling path duration distributions in MANETs and their impact on reactive routing protocols," *IEEE Journal on Selected Areas in Communications*, vol. 22, pp. 1357-1373, 2004.

[9] D. Bertsekas and R. Gallager, *Data Networks*, Second ed, Prentice Hall, 1992.

[10] F. Borgonovo, "ExpressMAN: Exploiting Traffic Locality in Expressnet," *IEEE Journal on Selected Areas in Communications*, vol. 5, pp. 1436-1443, 1987.

[11] T. Clausen, C. Dearlove, and B. Adamson, "Jitter Considerations in Mobile Ad Hoc Networks (MANETs)," The Internet Engineering Task Force, IETF, RFC 5148, 2008.

[12] S. R. Das, R. Castaneda, Y. Jiangtao, and R. Sengupta, "Comparative performance evaluation of routing protocols for mobile ad hoc networks," presented at The 7th IEEE International Conference on Computer Communications and Networks (IC3N'98), Lafayette, Louisiana, 1998.

[13] P. Denning, "The locality principle," *Commun. ACM*, vol. 48, pp. 19-24, 2005.

[14] P. Denning, "The working set model for program behavior," *Commun. ACM*, vol. 11, pp. 323-333, 1968.

[15] K. Fall, "NS Notes and Documentation," http://www.isi.edu/nsnam/ns/ns-documentation.html, 2000, [May 2010].

[16] R. Groenevelt, P. Nain, and G. Koole, "The message delay in mobile ad hoc networks," *Performance Evaluation*, vol. 62, pp. 210-228, 2005.

[17] 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.

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

[19] P. Jacquet, A. M. Naimi, and G. Rodolakis, "Routing on asymptotic delays in IEEE 802.11 wireless ad hoc networks " presented at 1st workshop on Resource Allocation in Wireless NETworks, in conjunction with IEEE WiOpt 2005, Riva Del Garda, Italy, 2005.

[20] L. Jinyang, B. Charles, S. J. D. C. Douglas, L. Hu Imm, and M. Robert, "Capacity of Ad Hoc wireless networks," presented at 7th annual international conference on Mobile computing and networking, Rome, Italy, 2001.

[21] 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.

[22] D. B. Johnson and D. A. Maltz, "Dynamic Source Routing in Ad Hoc Wireless Networks," in *Mobile Computing*, vol. 353, I. a. Korth, Ed. Kluwer Academic Publishers, pp. 153-181, 1996.

[23] L. Kleinrock, *Queueing Systems, Volume 2: Computer Applications.*, John Wiley & Sons, Inc., 1976.

[24] C. Kozierok, *The TCP/IP Guide*, 1st ed, No Starch Publishing, 2005.

[25] M. Mosko and J. Garcia-Luna-Aceves, "Performance of group communication over ad hoc networks," presented at IEEE Int. Symp. Computers and Communications ISCC, Italy, 2002.

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

[27] 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.

[28] M. Prasant and K. Srikanth, *AD HOC NETWORKS: Technologies and Protocols*, Springer-Verlag New York, Inc., 2004.

[29] S. Ross, Introduction to Probability and Statistics for Engineers and Scientists, Second ed, Academic Press, 2000.

[30] Y. Shaohu, Y. Shaohu, Z. Yongning, W. Shiqi, and A. W. G. Wei Guo, "Priority backoff algorithm for IEEE 802.11 DCF," presented at International Conference on Communications, Circuits and Systems (ICCCAS '04), 2004.

[31] M. Shikharesh and B. B. Richard, "Measurement and analysis of locality phases in file referencing behaviour," presented at ACM SIGMETRICS joint international conference on Computer performance modelling, measurement and evaluation, Raleigh, North Carolina, United States, 1986.

[32] A. Silberschatz, P. Galvin, and G. Gagne, *Operating Systems Concepts*, 7th ed, John Wiley & Sons, 2005.

[33] A. Tanenbaum, *Computer Networks*, 4th ed, Pearson Education, 2003.