

Adaptive and fuzzy approaches for nodes affinity management in wireless ad-hoc networks

Essam Natsheh and Tat-Chee Wan

School of Computer Sciences, Universiti Sains Malaysia, USM 11800, Penang, Malaysia

Tel.: +604 653 2062; Fax: +604 657 3335; E-mail: dr_natsheh@hotmail.com, tcwan@cs.usm.my

Abstract. In most of the ad-hoc routing protocols, a static link lifetime (LL) is used for a newly discovered neighbors. Though this works well for networks with fixed infrastructures, it is inadequate for ad-hoc networks due to nodes mobility and frequent breaks of links. To overcome this problem, routing protocols with estimated LL using nodes affinity were introduced. However, these protocols also used the static estimated LL during the connection time. In contrast to that, in this paper two methods are presented to estimate LL based on nodes affinity and then continually update those values depending on changes of the affinity. In the first method, linear function is used to map the relationship between the signal strength fluctuation and LL. In the second method, fuzzy logic system is used to map this relationship in a nonlinear fashion. Significance of the proposed methods is validated using simulation. Results indicate that fuzzy method provides the most efficient and robust LL values for routing protocols.

Keywords: Ad-hoc networks, AODV, link lifetime, fuzzy systems, intelligent networks

1. Introduction

Wireless ad-hoc networks consist of mobile nodes and have no fixed infrastructure. The mobility makes the nodes continually having new neighbors and losing some others. Each node must have self-detection capability to discover the nodes around it (neighbors) and through them it can reach far nodes (multi-hop concept). Therefore, one of the most concerns of the nodes is to have reliable values for links lifetime (LL) with their neighbors. If LL is too short, the node will suffer from continually checking the neighbors' connectivity. On the other hand, if LL is too high, the nodes will suffer from late discovering of broken links. Therefore, LL value is one of the most important parameters that the designer of ad-hoc routing protocol must takes into account.

To overcome choosing a suitable value for LL, Paul et al. [3] proposed to use the nodes affinity as a metric for LL. The idea behind it is strongly connected nodes must have higher values for LL than weakly connected nodes. Using this idea many protocols have been proposed in literature to determine LL values based on signal strength. These protocols outperform traditional protocols with static estimation of LL. However, these protocols use the estimated LL as a static value during the whole connection time. In this paper, the design and implementation of two adaptive mechanisms are described. They can determine suitable value of LL based on nodes affinity, and then they continually adapt this value based on changes of signal strength. The first mechanism, called adaptive-LL, optimizes LL linearly and have simple implementation. To provide non-linear optimization, fuzzy logic system is used which can provide this

option effectively, and hence, provide more accurate LL estimation. The second mechanism is called fuzzy-LL. These two mechanisms help the routing protocols to minimize packet loss while providing reliable LL.

To implement the proposed methods, Ad-hoc On-demand Distance Vector (AODV) routing protocol [1] is utilized as the underlying routing platform. AODV is a reactive routing protocol where the routes are determined as per needed only. It manages local connectivity using two parameters: `hello_interval` and `allowed_hello_loss`. The `hello_interval` specifies the time between two Hello messages, usually it is set to 1-second. If a neighbor does not receive any packets (Hello messages or otherwise) for more than `allowed_hello_loss × hello_interval` seconds, the node should assume the link to this neighbor is broken. Broch et al. [28] find that using `allowed_hello_loss` equal to 3 can produce best protocol performance. Therefore, LL takes a static value equal to 3-seconds.

Although our proposed methods are evaluated with AODV, they are suitable to use in any other ad-hoc routing protocols as well.

The rest of this paper is organized as follows. Section 2 summarizes related work on using signal strength to optimize neighbor connectivity. Followed by the implementation of adaptive-LL method, fuzzy-LL method, performance analyzes of the proposed methods, and finally the conclusions.

2. Related work

Dube et al. [2] proposed Signal Stability-based Adaptive (SSA) routing protocol using signal strength and stability of individual hosts as route selection criteria. According to the signal strength between two neighbors, SSA classifies the link between them as strongly/weakly connected. However, many studies [3–5] show that discovering a strong link does not necessarily lead to discovering a long lifetime route since the other links extending from the end node with the strong link may be very short LL. To overcome that, Paul et al. [3] introduce a parameter – *affinity* – which characterizes the relationship strength between two neighbors. This parameter utilizes LL as a measure of the strength of link. The affinity between two nodes n and m , a_{nm} , is:

$$a_{nm} = \begin{cases} \text{high} & \text{if } \Delta S_{nm(ave)} > 0 \\ \frac{S_{thresh} - S_{nm(current)}}{\Delta S_{nm(ave)}} & \text{otherwise} \end{cases} \quad (1)$$

where S_{thresh} is the threshold signal strength below which the link will be assumed broken. $\Delta S_{nm(ave)}$ is the average rate of change of signal strength over past few samples, where ΔS_{nm} is defined as:

$$\Delta S_{nm} = \frac{S_{nm(current)} - S_{nm(previous)}}{\Delta t} \quad (2)$$

Many researchers inherited the concept of *affinity* in designing their routing protocol. Agarwalet al. [4] proposed Route-Lifetime Assessment Based Routing protocol (RABR), which incorporates the residual-route-lifetime prediction on the basis of affinity appraisal. In addition, the protocol also considered the path length while choosing an optimal route for a TCP source.

In [6–11], the authors used *single to noise ratio* (SNR) as a measure of affinity between two neighbors. The authors in [12] used *propagation loss* factor while in [13,14] used *signal strength threshold* to achieve the same target. Although those factors are good metrics of nodes affinity, they works at a physical layer of ad-hoc networks and they depends on the antenna and network adapter, hence, they needs interlayer

```

At every  $\Delta t$  seconds:
Calculate  $\Delta S_{nm}$ ;
if ( $\Delta S_{nm} > 0$  and Link_Lifetime  $\leq 6$ )
    Link_Lifetime = Link_Lifetime +  $\alpha$ ; // increment
else if ( $S_{nm} < 0$  and Link_Lifetime  $\geq 0.2$ )
    Link_Lifetime = Link_Lifetime *  $\beta$ ; // decrement

Fixed Parameters:
 $\Delta t = 0.5$  seconds; // sampling interval time.
 $\alpha = 0.2$ ; // increase factor.
 $\beta = 0.9$ ; // decrease factor.

```

Fig. 1. Adaptive-LL algorithm.

interaction with routing layer to choose stable paths. More research is needed to see how the routing layer should use information received from the physical layer and its effect on performance optimization [15].

Other studies used the signal strength fluctuation as an indicator of possible link breakage. In [16], the authors argue that when signal strength is going to be stronger it means that two nodes will be closer and the link between them would have longer lifetime. In [17,18], the authors used this concept to suggest an algorithm of early searching for an alternative route when the original route going to be weaker. In [11], the authors exploit the variation tendency of signal strength to detect the relative movement between nodes, which is utilized to identify leaving and thus unreliable neighbors.

3. Adaptive links lifetime method

Figure 1 presents an on-line algorithm for adaptively changing the LL according to the observed signal strength. The idea behind this algorithm is to infer whether LL should become longer or shorter by examining the variations in signal strength. If ΔS_{nm} is positive, it indicates that the affinity is increasing and the signal strength become stronger, then LL should be longer. On the other hand, if ΔS_{nm} is negative, it indicates that the affinity is decreasing and the signal strength become weaker, then LL should be shorter.

To ensure that LL will be in acceptable range, we restrict it to stay within the range [0.2, 6] seconds. This ensures the acceptable range of LL, does not matter, whether the signal strength is very weak or very strong.

In recommending values for α and β , it is needed to ensure that a single variation of signal strength does not result in large variation in LL. We note that it takes at least 30 intervals for LL to increase from 0.2 seconds to 6 seconds or to decrease from 6 seconds to 0.2 seconds, which is 15 seconds for our default parameters for Δt , α and β . More accurate variation of LL could be achieved using non-linear capability of fuzzy logic method proposed in the next section.

4. Fuzzy logic based links lifetime method

In this section, the proposed concept and rules for fuzzy affinity algorithm, used with AODV are introduced. In the following subsection, we discuss the need for fuzzy configuration of routing protocols parameters, followed by three subsections determines the effect of some node parameters on nodes' affinity. These parameters are then used in subsection E to create the rules of the fuzzy system.

A method to design their membership functions is presented in subsection F. The overall system design is presented in subsection G, followed by the discussion of its compatibility with original static method.

4.1. Fuzzy configuration of routing protocols parameters

In this technique the fuzzy reasoning is used to dynamically configure the protocols parameters instead of using static values. The dynamical configuration can adapt to the changing of the network topology and improve the protocol performance. In other hand using static parameters for the protocols in ad-hoc environment that suffer from frequent change of network topology and different traffic intensity may degrade the routing protocols performance.

Wang et al. [29] used a fuzzy reasoning to dynamically configure five routing parameters of AODV routing protocol. They used mathematical models to represent nodes moving mode and their traffic mode. These models were used to categorize the network environments to 9 categorize. The fuzzy reasoning was used to estimate the nodes membership degree in these environments. Depending on the node membership degree, the values of the protocol parameters are increased or decreased.

Actually, the fuzzy reasoning can be used more effectively to accurately calculate the real values of protocol parameters that map the status of the node and its links [30]. We did this to dynamically adapt routes lifetimes [31], 'Hello' messages interval time [32] and active queue management for congestion control [33]. In this paper we apply the fuzzy reasoning to dynamically adapt links lifetime.

4.2. Effect of node transmission power on links lifetime

Transmission power is a main parameter that determines the number of neighbors for nodes in this proposed ad-hoc network. Transmission power (*TrPower*) is the strength by which the signal is sent.

Here, signal power degradation is modeled by the *free-space propagation model* [19], where the received signal strength is:

$$P_r(d) = \frac{P_t G_t G_r \lambda^2}{(4\pi)^2 d^2 L} \quad (3)$$

where P_r and P_t are the receive and transmit powers (in Watts), G_t and G_r are the transmit and receive antenna gains, d is the transmit-receive separation distance, L is a system loss factor ($L = 1$ in our simulations which indicates no loss in the system hardware), and λ is the carrier wavelength (in meters) which is related to the carrier frequency (f_c) as:

$$\lambda = \frac{c}{f_c} \quad (4)$$

where c is the speed of light (3×10^8 m/s). Assuming a unity gain antenna with a 900 MHz carrier frequency, Fig. 2 shows the relation between the transmission range (*TrRange*) and the transmission power (*TrPower*). From this figure, if the *TrPower* of a node is too low, then the signal will reach to a few neighbors only and the links with those neighbors may be weak and easy to break. Consequently, LL must be small enough to get fast update for neighborhood changes. In contrast, high *TrPower* of a node will lead to high average number of its neighbors and hence increase the lifetime of its links, consequently the LL must be long due to fewer changes in node's neighborhood. Hence the following rules are proposed:

R1: If *DTTRR* is high then LL must be low

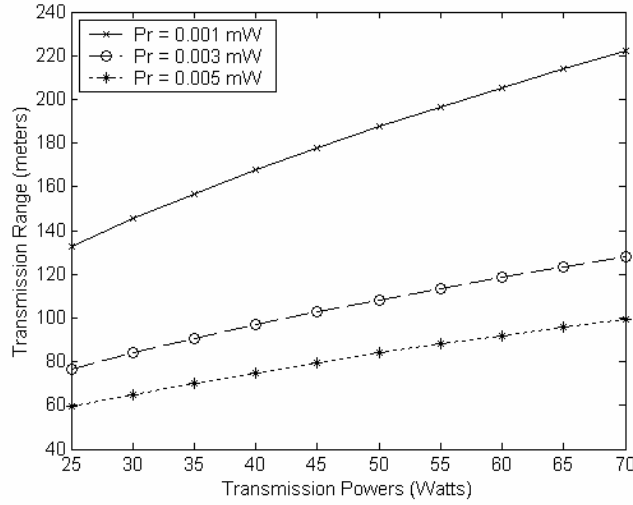


Fig. 2. TrRange-TrPower relationship.

R2: If *DTTRR* is medium then LL must be medium

R3: If *DTTRR* is low then LL must be high

where *DTTRR* is distance to TrRange ratio. When *DTTRR* is high that mean the neighbor is far away from the node but it's still inside the TrRange. In contrast, when *DTTRR* is low that mean the neighbor is near to the node and it's even closer than the half of TrRange.

4.3. Effect of distance between neighbors on links lifetime

By increasing the distance between a node and its neighbor, the transmission loss or propagation loss can also be increased. For free-space propagation, the loss L_f is defined as [19]:

$$L_f = \frac{P_t}{P_r} = \frac{1}{G_r G_t} \left(\frac{4\pi d}{\lambda} \right)^2 \tag{5}$$

For omnidirectional transmit and receive unity gain antennas, L_f are given by:

$$L_f = \left(\frac{4\pi d}{\lambda} \right)^2 = \left(\frac{4\pi f_c d}{c} \right)^2 \tag{6}$$

L_f in decibels can be written as:

$$L_f = 32.45 + 20 \log_{10}(f_c) + 210 \log_{10}(d) \tag{7}$$

Figure 3 shows the propagation loss characteristics. From this figure, it is obvious that if a node have a neighbor and they moves away from each other, then the connectivity strength between them will be weaker and vice versa. Hence, the following rules are proposed:

R4: If Δd is negative then LL must be high

R5: If Δd is zero then LL must be medium

R6: If Δd is positive then LL must be low

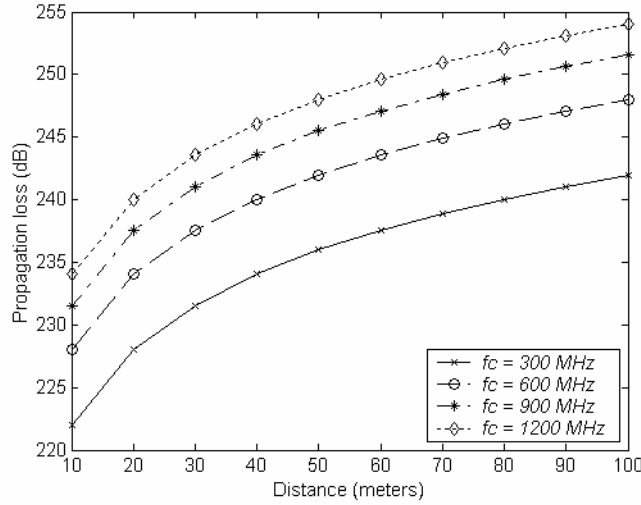


Fig. 3. Distance-propagation-loss relationship.

where Δd is the difference between the current neighbor position and its previous position. When Δd is negative that means the neighbor becomes closer to the node and, as described above, the connectivity strength will be stronger. In contrast, when Δd becomes positive, the neighbor moves away and the connectivity will be weaker.

4.4. Effect of node speed on links lifetime

Ad-hoc networks experience dynamic changes in network topology because of unrestricted mobility of the nodes. If a node has fast movement, this leads to an increase in probability of links breaks with its neighborhood. The node movement can be measured by its speed. High-speed of a node results a high probability of losing some of the current neighbors and acquiring new ones.

Galluccio et al. [20] calculated the neighborhood time, T_n , between two node n_1 and n_2 as:

$$T_n = \frac{2 \times \sqrt{R^2 - P_{n1}^2}}{v_{n1}} \quad (8)$$

where R is the transmission range, P_{n1} is the position of n_1 according to n_2 , and v'_{n1} is the relative velocity and can be calculated as:

$$v'_{n1} = \sqrt{v_{n1}^2 + v_{n2}^2 - 2v_{n1}v_{n2} \cos(\Phi)} \quad (9)$$

where v_{n1} is the magnitude of the vector \vec{v}_{n1} , v_{n2} is the magnitude of the vector \vec{v}_{n2} , and Φ is the angle between them. Even though the assumption of T_n is accurate only with constant relative velocity, it can show the effect of nodes' velocity on neighborhood time (in Fig. 4). From this figure, it is clear that the links' lifetime for fast moving nodes with their neighbors is very small due the expected links breaks.

In general, a rule can be defined as: when speed is high, the LL must be low and vice versa. Consequently the following rules are proposed:

R7: If speed is high then LL must be low

R8: If speed is medium then LL must be medium

R9: If speed is low then LL must be high

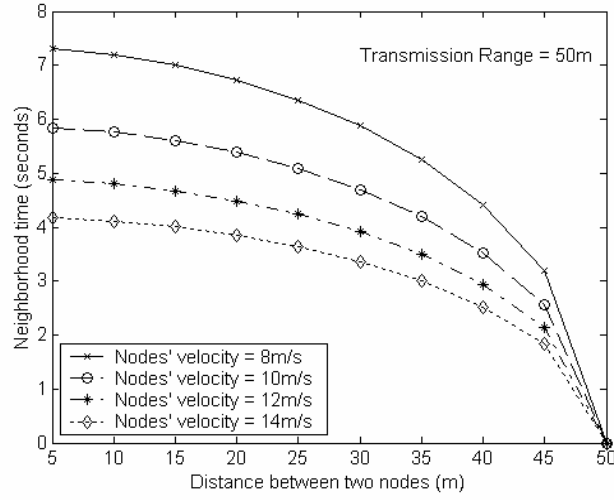


Fig. 4. The effect of nodes’ velocity on neighborhood time.

4.5. The rule-base for fuzzy links lifetime

To fulfill the fuzzy sets theory, the previous nine rules (R1 to R9) can be combined within one rule-base to control the LL adaptively as presented in Table 1. For example, according to Table 1 the first rule is: IF Δd is negative AND $DTTRR$ is low AND $speed$ is low THEN LL is high.

4.6. Membership functions for the fuzzy variables

After defining the fuzzy linguistic ‘if-then’ rules, the membership functions corresponding to each element in the linguistic set must be defined. For example, if the $TrPower$ equal to 7 mW, using conventional concept, it implies $TrPower$ is either ‘low’ or ‘medium’ but not both. In fuzzy logic, however, the concept of membership functions allows us to mention the $TrPower$ is ‘low’ with 80% membership degree and ‘medium’ with 20% membership degree.

The proposed membership functions are shown in Fig. 5 due to their economic nature of the parametric and functional descriptions. In these membership functions, the designer needs only to define two parameters; $midpoint$ and $maxpoint$. These membership functions mainly contain the *triangular* shaped membership function. This function is specified by three parameters (a, b, c) as follows:

$$triangle(x; a, b, c) = \begin{cases} (x - a)/(b - a) & \text{for } a \leq x \leq b \\ (c - x)/(c - b) & \text{for } b \leq x \leq c \\ 0 & \text{elsewhere} \end{cases} \quad (10)$$

where $a = midpoint/2$, $b = midpoint$, $c = 3 \times midpoint/2$ and x is the input to the fuzzy system. The remaining membership functions are as follows: Z-shaped membership to represent the whole set of low values and S-shaped membership to represent the whole set of high values.

$Midpoint$ is the value of fuzzy variable, which can be chosen from the real network measurements, simulation and analysis or from the default values of protocol specification as follows.

Normally the $TrPower$ of a node can be read from the properties of the network adapter. So, it is easy to expect the minimum and the maximum $TrPower$ for the nodes sharing the network. Also every node

Table 1
Rule-base for fuzzy links lifetime

no.	Δd	DTTRR	speed	LL
1	negative	low	low	high
2	negative	low	medium	high
3	negative	low	high	medium
4	negative	medium	low	high
5	negative	medium	medium	medium
6	negative	medium	high	low
7	negative	high	low	medium
8	negative	high	medium	low
9	negative	high	high	low
10	zero	low	low	high
11	zero	low	medium	high
12	zero	low	high	medium
13	zero	medium	low	high
14	zero	medium	medium	medium
15	zero	medium	high	low
16	zero	high	low	medium
17	zero	high	medium	low
18	zero	high	high	low
19	positive	low	low	high
20	positive	low	medium	high
21	positive	low	high	medium
22	positive	medium	low	high
23	positive	medium	medium	medium
24	positive	medium	high	low
25	positive	high	low	medium
26	positive	high	medium	low
27	positive	high	high	low

had its own speed range $[s_{min}, s_{max}]$. In the same way, the Δd range is $[TrRange_{min}, TrRange_{max}]$. Hence, *Midpoint* for those three variables is the average of their ranges. For example, if the TrPower of a node are within 8 mW to 14 mW then *midpoint* for its TrPower membership function is 11 mW.

AODV protocol specification [28] states the static value of LL is 3-second. Hence, for the LL membership function, *midpoint* should be equivalent to 3-second.

Since the values of input variables occur during the simulation run, exact knowledge of their values cannot be determined. The range of values (*maxpoint*) for these variables must be quite large. Hence, *Maxpoint* can be defined as follows:

For input variables: $maxpoint = 3 \times midpoint$.

For output variable: $maxpoint = 2 \times midpoint$.

4.7. Fuzzification, inference and defuzzification

The fundamental diagram of the fuzzy system is presented in Fig. 6. Formally, the rule-base (Table 1) of the fuzzy-LL method can be represented in the following format:

IF Δd is A_{i1} AND DTTRR is A_{i2} AND speed is A_{i3} THEN LL is B_i

where A_{i1} , A_{i2} , A_{i3} , and B_i are the linguistic labels *low*, *medium*, and *large* of the i th rule.

Fuzzification is a process where crisp input values are transformed into membership values of the fuzzy sets (as described in the previous subsection). After the process of fuzzification, the inference engine calculates the fuzzy output using the fuzzy rules described in Table 1. *Mamdani* method [21]

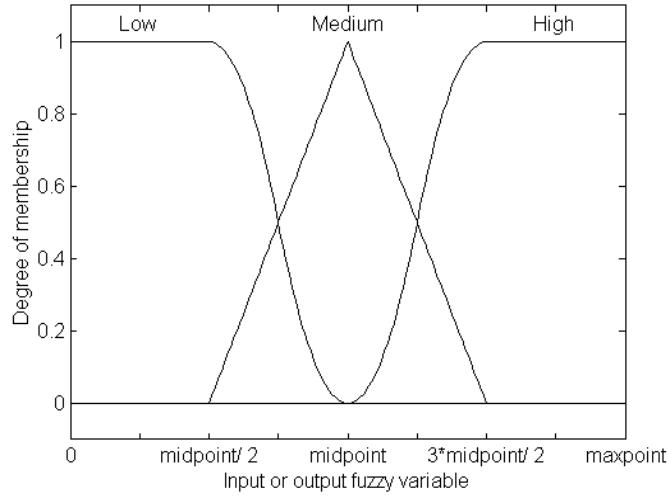


Fig. 5. Membership functions used for the fuzzy variables.

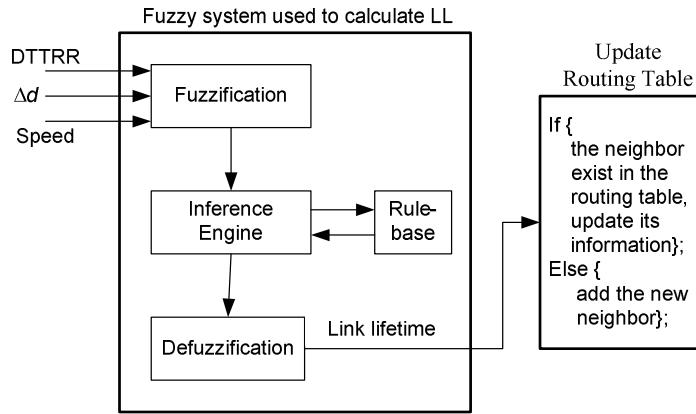


Fig. 6. Block-diagram for the basic elements of the fuzzy system.

is used as fuzzy inference engine, where Min (\wedge) operator is chosen as AND connective between the antecedents of the rules as follows:

$$\tau_i = A_{i1}(x_1) \wedge A_{i2}(x_2) \wedge A_{i3}(x_3) \tag{11}$$

where τ_i is called the *degree of firing* of the i th rule for the input values: $\Delta d = x_1$, DTTRR = x_2 and speed = x_3 . The next step is determining the individual rule output F_i (fuzzy set) which is obtained by:

$$F_i(y) = \tau_i \wedge B_i(y) \tag{12}$$

The third step is the aggregation of rules outputs to obtain the overall system output F (fuzzy set), where Max (\vee) operator is chosen as OR connective between the individual rules:

$$F(y) = \vee_i F_i(y) = \vee_i (\tau_i \wedge B_i(y)) \tag{13}$$

For use in the ad-hoc networks environment a fourth step must be added. A crisp single value for LL is needed. This process is called *defuzzification*. Center of area (COA) [21] is chosen as the defuzzification method given in the following:

$$LL = \frac{\sum_{j=1}^m F(y_j) \times y_j}{\sum_{j=1}^m F(y_j)} \quad (14)$$

here y_j is a sampling point in the discrete universe output F , and $F(y_j)$ is its membership degree in the membership function.

4.8. Compatibility between static and adaptive links lifetime

The proposed two adaptive LL methods are compatible with static-LL method in the sense that a node that uses adaptive LL (an “intelligent” node) may communicate with a node that uses static-LL (standard node), as there are no changes in the control messages format or in the other protocol parameters for adaptive LL methods. However, in a network mixed with intelligent and standard nodes, the intelligent nodes have changeable value of LL and hence provide more reliable links reflects updating topology changes.

5. Performance analysis of the proposed methods

5.1. Simulation environment

Simulation of the proposed AODV design was done using *OMNeT++* version 2.3 with *Ad-Hoc simulator* 1.0 [22]. *OMNeT++* is a powerful object-oriented modular discrete event simulator tool. Each mobile host is a compound module which encapsulates the following simple modules: an application layer, a routing layer, a MAC layer, a physical layer, and a mobility layer.

Application layer: This module produces the data traffic that triggers all the routing operations. In all scenarios, 7 nodes are enabled to transmit. The traffic was modeled by generating a packet burst of 64 packets sent to a randomly chosen destination that stays the same for all the burst length. The rate of transmission is 3 packets/s. The time elapsed between two application bursts is normally distributed in [0.1, 3] s. The packet size is 512 bytes.

Routing layer: The routing model is the heart of the simulator. This model depicts the AODV routing protocol, all of its functions, parameters and their implementation [1].

MAC layer: The simple implementation for this layer has been used. The outgoing messages are being let pass through. The incoming one instead is delivered to the higher levels with an *MMI queue* policy. When an incoming message arrives, the module checks a flag that advises if the higher level is busy. If so, the message will be saved in the buffer. If the buffer is full, it will be dropped. When the higher level is not busy, the MAC module picks the first message from the buffer and sends it upward.

Physical layer: It cares about the on-the-fly creation of links that allow the exchange of messages among the nodes. Every time a node moves from its position an interdistance check on each node is performed. If a node gets close enough (depending on the TrPower of the moving nodes) to a new neighbor, a link is created between the two nodes with the following properties: channel bandwidth is 11 Mb/s (IEEE 802.11a) and tolerable delay is 10 μ s. Each node has a defined transmission range chosen from an uniformly distributed number between [90, 120] m.

Table 2
control packets used by AODV

Message	Description
RREQ	a Route Request message
RREP	a Route Reply message
RERR	a Route Error containing a list of the invalid destinations
RREP_ACK	a RREP acknowledgment message

Mobility layer: The *random waypoint* model was adopted for the mobility layer. It is one of the most used mobility model in ad-hoc network simulations. In this model, a node randomly selects a destination. On reaching the destination, another random destination is targeted after 3 seconds *pause time*. The speed of movement of individual nodes is between [2, 12] m/s. The direction and magnitude of movement was chosen from a uniformly distributed random number.

Three different network sizes are modeled: 700 m × 700 m map size with 25 nodes, 35 nodes and 800 m × 800 m map size with 45 nodes. Each simulation run takes 300 simulated seconds. Multiple runs were conducted for each scenario and collected data was averaged over those runs.

5.2. Validation of the simulator

Nicola Concer, the creator of *Ad-Hoc simulator*, proved the validity of the simulator by comparing AODV protocol performance with the performance of the same protocol generated by ns-2 simulator [34]. He used some performance metrics to compare between the two protocols performance and found that the difference between the results is less than 5%.

5.3. Performance metrics

The following metrics were used for measuring performance:

– Routing Overhead:

$$\text{Overhead} = \frac{\sum_{i=1}^n \text{Number of SentCtrlPkt by source}}{\sum_{i=1}^n \text{Number of received data by destination}} \quad (15)$$

where n is number of nodes in the network and *SentCtrlPkt* is control packets used by AODV and described in Table 2. This metric can be employed to estimate how many transmitted control packets are used for one successful data packet delivery to determine the efficiency and scalability of the protocol.

– *Average End-to-End Delay:* Average packet delivery time from a source to a destination. First, for each source-destination pair, average delay for packet delivery is calculated. Then the whole average delay is calculated from average delay of each pair. End-to-end delay includes the delay in the send buffer, the delay in the interface queue, the bandwidth contention delay at the MAC, and the propagation delay.

– *Sensitivity to mobility model:* In the performance evaluation of a protocol for an ad-hoc network, the protocol should be tested under realistic conditions especially realistic movements of the mobile nodes. We use the following mobility models that represent mobile nodes whose movements are independent of each other (*i.e.*, entity mobility models concept) [23]:

Random Walk (RW): A simple mobility model based on random directions and speeds.

Random Waypoint (RWP): A model that includes pause times between changes in destination and speed. This model has been described in subsection A.

Table 3
Percentage of decrement in routing overhead

	25 nodes	35 nodes	45 nodes
Adaptive-LL	15.5	3.4	10.9
Fuzzy-LL	29.5	21.8	18.1

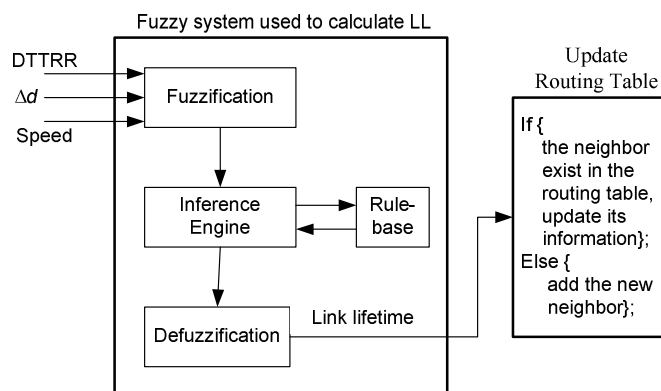


Fig. 7. Typical moving of a mobile node in (a) *Random Walk* and *Random Waypoint* models and (b) *Normal Walk* model.

Normal Walk (NW): The model is derived from Markovian models [23,24]. It chooses a new direction and speed picking a value from a fixed range of values surrounding the former speed and direction value. This model gives the nodes quite uniform motion that keeps on “shaking” around the initial direction angle. When a node reaches a boundary, it reflects the boundary by choosing a new random direction.

Figure 7 (a) and (b) illustrate the typical moving of a mobile node in RW and RWP models and NW model, respectively [25].

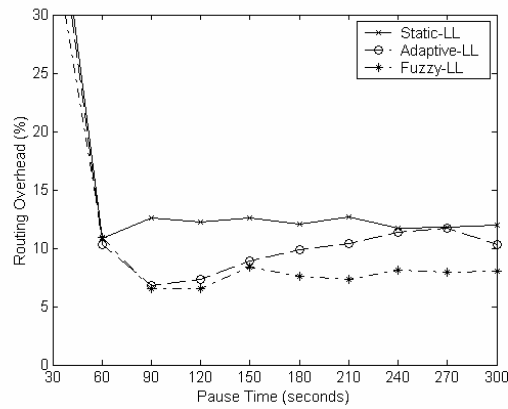
6. Simulation results and evaluations

6.1. Routing overhead details

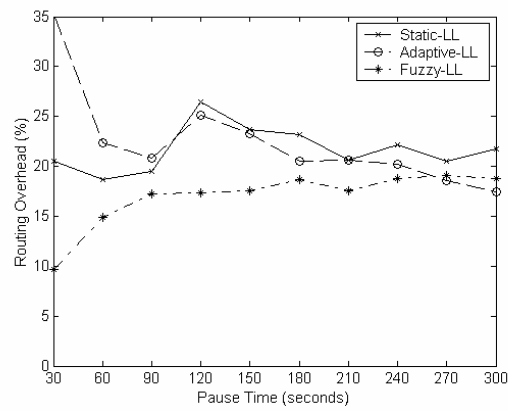
Comparison of routing overhead for AODV routing protocols are shown in Fig. 8. Using static-LL as a base system, Table 3 shows percentage of decrement in routing overhead for the proposed adaptive methods.

Significance of fuzzy-LL method is in decreasing data loss through broken paths, hence increasing the number of received data by destinations. This significance is due to adaptive values of LL to reflect the nodes’ mobility properties. These values of LL are more reliable than LL values used by adaptive-LL method which depends on linear function. Anyway, these two adaptive methods show effective decreasing in routing overhead than the original static-LL method, the reason for that can be described as follows.

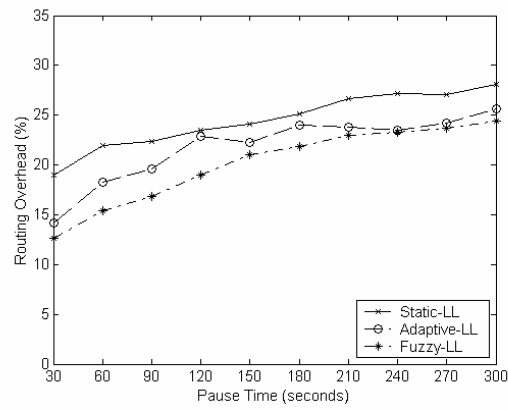
In adaptive methods, many neighbors are given a very short LL because of weak connectivity with them. Suppose a link with one of these neighbors is broken. The adaptive method discovers broken link and remarks it as expired while the static method still keeps it as active because of long link lifetime. Hence, if the node wishes to send data packets across that broken link, the adaptive method starts by



(a) 25 nodes



(b) 35 nodes



(c) 45 nodes

Fig. 8. Routing overhead comparison.

initiating a path discovery process, while using the static method the node sends the data directly using the old broken link. After sending the data, it discovers the broken link, and then the node starts initiating the path discovery process.

Table 4
Percentage of decrement in end-to-end delay

	25 nodes	35 nodes	45 nodes
Adaptive-LL	27.9	7.9	13.7
Fuzzy-LL	44.4	19.4	22.3

6.2. Average end-to-end delay details

Figure 9 indicates that the proposed adaptive methods have lower average end-to-end delay compared with static-LL method. The percentage of this decrement is summarized in Table 4.

The static-LL method needs more routing delay to recover from broken paths and discover new ones. To recover a broken path, an RERR message must first be launched from the intermediate nodes to tell the source node about the link break. The source node deletes the corresponding entry from its routing table. The RREQ must then be broadcast from the source to the destination, and an RREP consequently has to be transmitted back to the source. Data packets are buffered at the source node during this process and the duration of their buffering adds more delay time to the end-to-end delay. Adaptive methods, on the other hands, have reliable routes that minimize the need to this recovery process.

Again, significant improvements of fuzzy-LL than adaptive-LL can be observed. This is due to the higher number of SentCtrlPkt used by adaptive-LL method. This increment in SentCtrlPkt is used to recover from higher invalid routes in this method. Increased number of this packets increases buffer utilization and bandwidth consumption. In the other hand, intelligently optimization of LL in the fuzzy method is the reason for its significant improvements.

6.3. Sensitivity to mobility model details

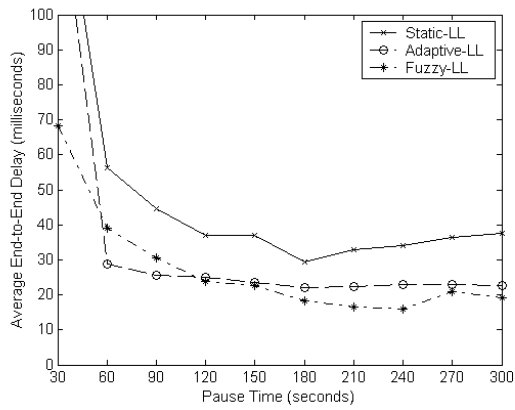
Figures 10 illustrate the performance of AODV with the three chosen mobility models and simulated with 35 nodes in the network. This figure illustrate that the RWP mobility model has less average routing overhead and less average end-to-end delay compared to RW mobility model. These results exist since mobile nodes using the RWP mobility model wait for 3 seconds pause time before moving to the next selected location. Using pause time in RWP model produces a more stable network (*i.e.*, few link changes per mobile node) than a RW mobility model.

The NW mobility model has the highest average routing overhead and the highest average end-to-end delay since each mobile node shaking in the same movement direction until reaching to the simulation area border before changing the direction. Thus, hop counts between the source node and the destination node are higher. Beside that, it has been observed in [26,27] that the nodes moving according to the RWP and RW models are often traveling through (or to) the center of the simulation area. This effect, known as border effect [26], increases the network connectivity due the high nodes density is in the center of the simulation area.

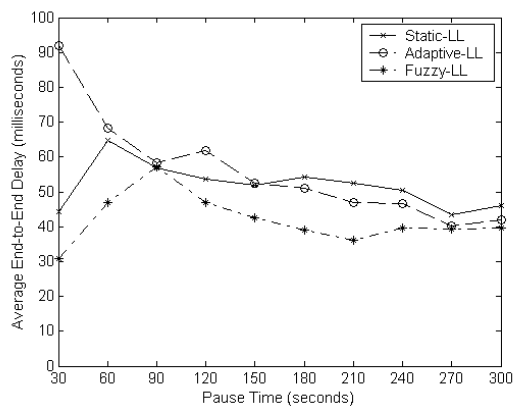
Lastly, it can be observed that the performance of AODV with the adaptive-LL method falls in-between the original (static-LL) and fuzzy-LL methods; more accurate links lifetime exist by using the fuzzy-LL method.

6.4. Links lifetime consistency details

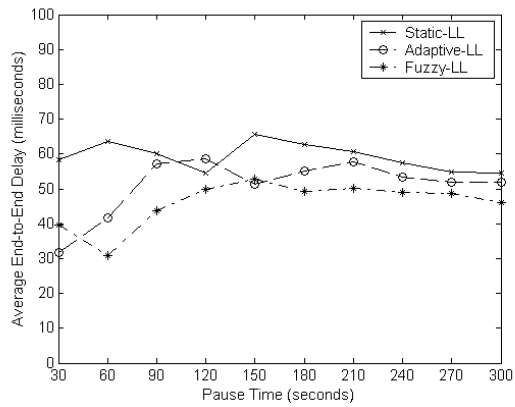
In the AODV parameters [28], LL always takes a static value of 3-seconds. Figures 11 and 12 show LL values used by proposed adaptive-LL and fuzzy-LL methods respectively for a randomly chosen node.



(a) 25 nodes



(b) 35 nodes



(c) 45 nodes

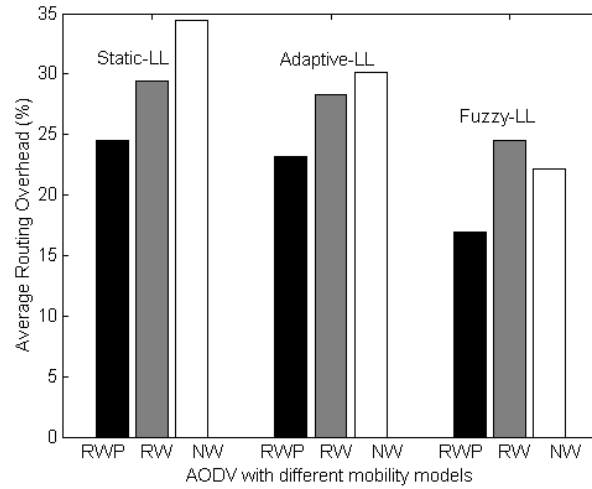
Fig. 9. End-to-end delay comparison.

It is shown that these methods use a variety of values for LL from around 0-seconds to around 6-seconds. Table 5 shows the average values of these ranges. Every link in every node has its own values of LL.

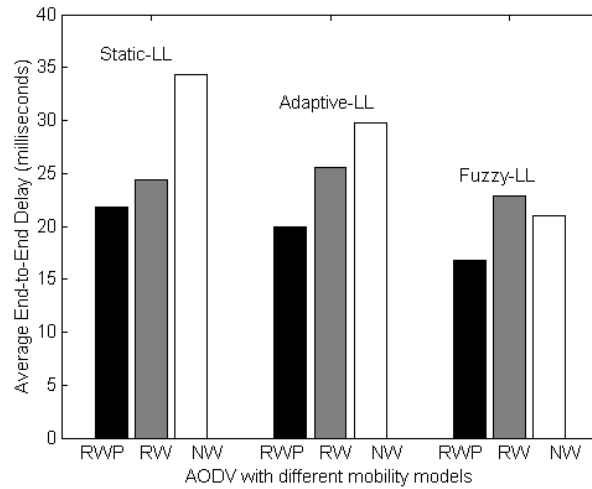
From Figs 11 and 12, we observe that the fluctuation range of adaptive-LL is far large than that of

Table 5
Average LL values used by a chosen node

	25 nodes	35 nodes	45 nodes
Adaptive-LL	2.6	2.6	2.8
Fuzzy-LL	3.0	2.6	2.6



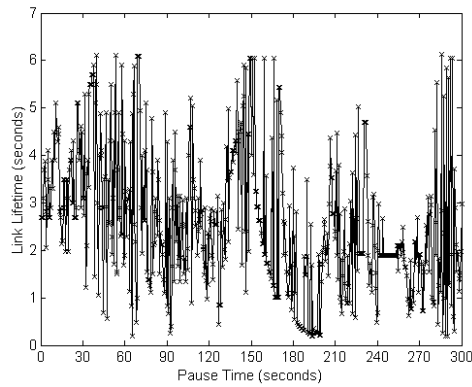
(a) Routing overhead comparison



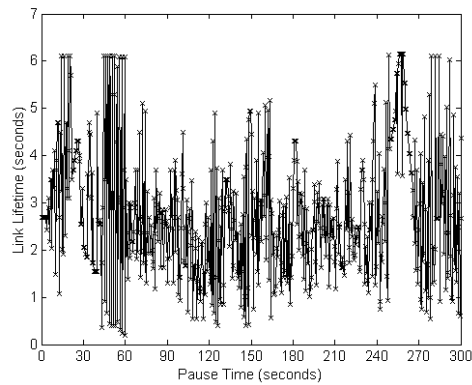
(b) End-to-end delay comparison

Fig. 10. Sensitivity to mobility model comparison.

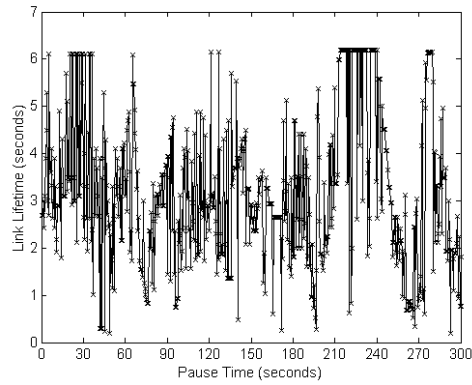
fuzzy-LL. This makes the fuzzy-LL method more consistent than adaptive-LL method. This is because it has a nonlinear relationship with the strength of the links as they are weaker or stronger. This consistency is the accuracy of the proposed mobility measure as the mobility measure reliably represents the links strength regardless of network scenarios.



(a) 25 nodes network



(b) 35 nodes network



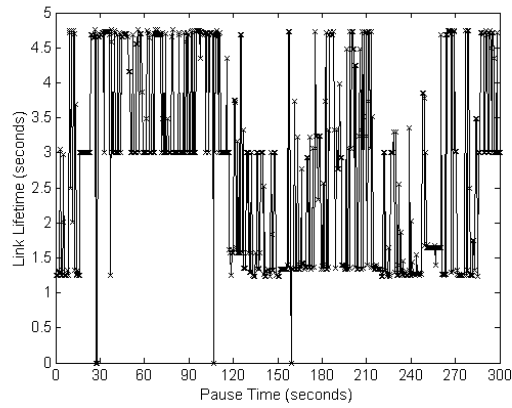
(c) 45 nodes network

Fig. 11. Links lifetime values used by a node in adaptive-LL method.

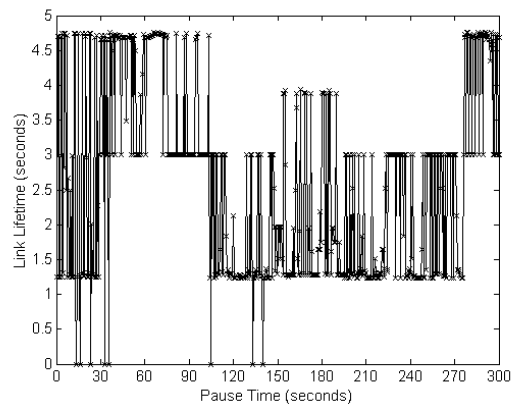
6.5. Illustrating links lifetime consistency

To illustrate LL consistency, we monitor two nodes in our 45 nodes network and record LL values and distance between them. Figure 13 shows LL values used by the two proposed methods.

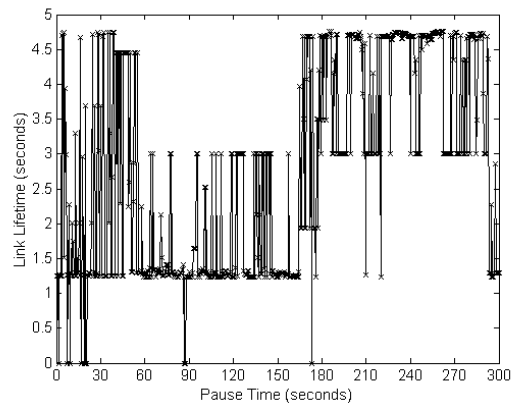
Figure 13 (a) shows LL values using adaptive-LL method. It shows when the two nodes are close



(a) 25 nodes network



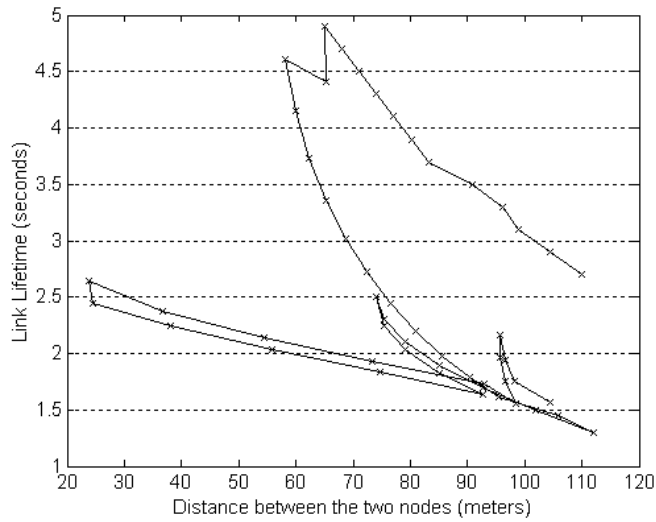
(b) 35 nodes network



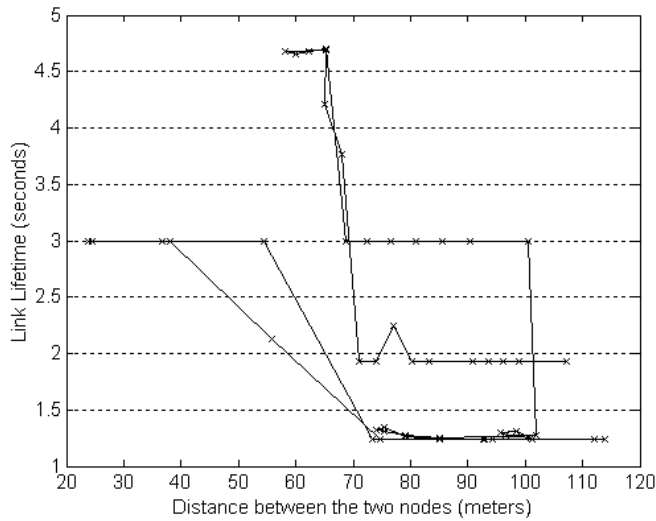
(c) 45 nodes network

Fig. 12. Links lifetime values used by a node in fuzzy-LL method.

enough (distance less than $TrRange$) a link between them will be created with 2.7 sec LL start value. As the two node moves closer, LL values increase linearly. After 60 m distance between the two nodes, they start to moves away from each other with linear decreasing of LL values.



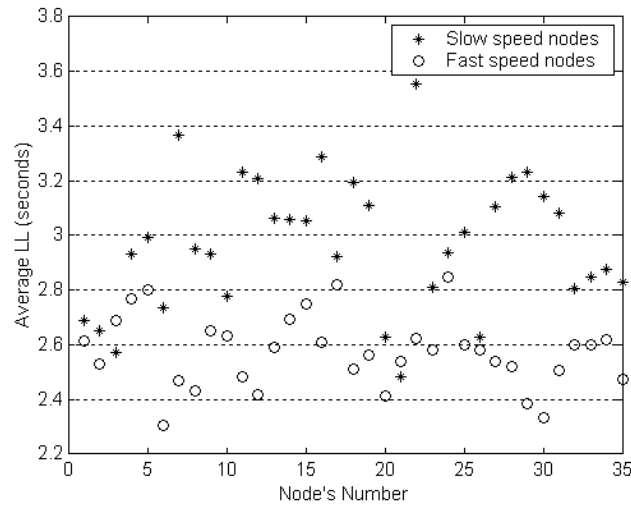
(a) Adaptive-LL method



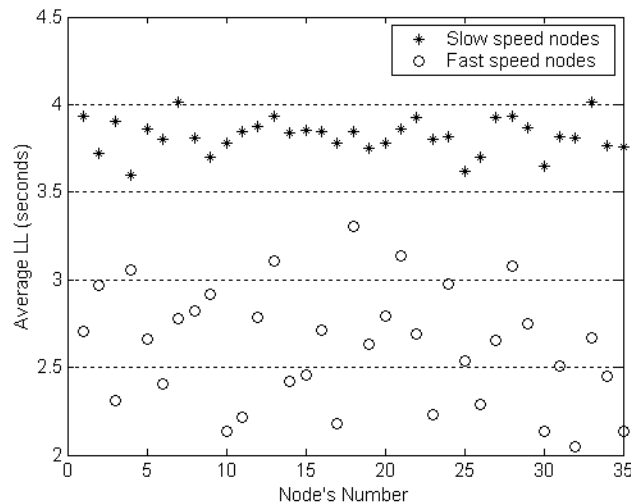
(b) Fuzzy-LL method

Fig. 13. Illustrating LL consistency for the two proposed methods.

Figure 13 (b) shows LL values using fuzzy-LL method. It shows when the two nodes are close enough the link will be created with 1.9 sec LL value. This low value is given to LL because nodes move to be closer that make Δd positive and rule R6 will applied. It's noticed that as nodes moves to be closer LL value not changed as long as other affinity parameters (TrPower and speed) not changed. According to our design, *DTTRR* is considered 'high' during the previous movement due far distance between the two nodes. As the distance between the two nodes comes close to the middle of TrRange, *DTTRR* be 'medium' and new values are given to LL. The new values of LL reflect that the two nodes are strongly connected. It's also clear from the figure that as the two nodes move away from each other, new nonlinear values are given to LL.



(a) Adaptive-LL method



(b) Fuzzy-LL method

Fig. 14. Sensitivity to nodes speed comparison.

6.6. Sensitivity to nodes speed details

Here, the effect of fast and slow topology changes due to nodes speed especially with the proposed adaptive methods is investigated. As mentioned earlier, all the previous study done with nodes speed range between [2, 12] m/s, those nodes can be considered as fast speed nodes. To simulate slow speed nodes, the simulation is done with nodes given speed between [0, 2] m/s. The results of the simulation using 35 nodes network are shown in Fig. 14.

It can be observed in fuzzy-LL method that due to few topology changes, slow speed nodes have longer LL values than fast speed nodes. LL of slow speed nodes are quite stable and have closer values while the reverse behavior is observed for fast moving nodes. This phenomenon is not occurred in

adaptive-LL method. This proves that the signal strength metrics used by fuzzy-LL method reflect an accurate measure for topology changes. Beside that, the fuzzy engine used in fuzzy-LL method use the strength of reasoning more optimally than linear functionality used by adaptive-LL method.

7. Conclusions

This paper has shown how adaptive LL can be used to effectively reduce routing overhead and end-to-end delay in ad-hoc networks. To accomplish this, two methods to adaptively optimize LL has been proposed. The first method (adaptive-LL) used linear function while the second method (fuzzy-LL) used fuzzy reasoning. These two methods maps the relationship between nodes affinity and their LL. Fuzzy-LL method showed more consistency to LL fluctuation and more robust to topology changes than adaptive-LL method. Experimental results using a small number of nodes have shown the efficacy of the two proposed methods. Since LL optimization become more effective as number of nodes increase, it is expected that the performance will improve significantly as the network grows.

We are currently examining ways to methodically improve the adaptiveness of the ad-hoc routing protocols. This paper presents adaptation for one specific parameter (LL). There are many other parameters that can be optimized. For example, route lifetime can actually keep track of the number of nodes in the path and their affinity changes. In such cases, the protocol should monitor nodes' affinity to trigger link break notification in time to prevent path loss.

Acknowledgment

This work is partially supported by USM's project: "Performance Study on Mobile Ad Hoc Networks Routing Protocols in Heterogeneous Density Environment."

References

- [1] C. Perkins, E.M. Royer and S.R. Das, *Ad Hoc On-demand Distance Vector (AODV) routing*, Internet-Draft, draft-ietf-manet-aodv-13.txt (Work in progress), Feb 2003.
- [2] R. Dube, C.D. Rais, K. Wang and S.K. Tripathi, Signal Stability-Based Adaptive Routing (SSA) for Ad Hoc Mobile Networks, *IEEE Personal Communications* **4**(1) (Feb 1997), 36–45.
- [3] K. Paul, S. Bandyopadhyay, A. Mukherjee and D. Saha, Communication-Aware Mobile Hosts in Ad-Hoc Wireless Network, *Proc of IEEE Int'l Conf. on Personal Wireless Comm (ICPWC)* Jaipur, India, (Feb 17–19 1999), 83–87.
- [4] S. Agarwal, A. Ahuja, J. Singh and R. Shorey, Route-Lifetime Assessment Based Routing (RABR) Protocol for Mobile Ad-Hoc Networks, *Proc of IEEE Int'l Conf on Comm* **3** (June 2000), 1697–1701.
- [5] Z. Cheng and W.B. Heinzelman, Exploring Long Lifetime Routing (LLR) in ad hoc Networks, *Proc of the 7th ACM Int'l Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems* Italy, (Oct 4–6 2004), 203–210.
- [6] S. Bandyopadhyay and K. Hasuike, A Connection Management Protocol to Support Multimedia Traffic in Ad Hoc Wireless Networks with Directional Antenna, *Proc of IEEE Int'l Conf on Multimedia and Expo (ICME2001)* Tokyo, Japan, (Aug 2001), 1037–1040.
- [7] S. Bandyopadhyay, K. Hasuike, S. Horisawa and S. Tawara, An Adaptive MAC and Directional Routing Protocol for Ad Hoc Wireless Network Using ESPAR Antenna, *Proc of the ACM Symposium on Mobile Ad Hoc Networking & Computing* USA, (Oct 2001), 243–246.
- [8] K.W. Chin, J. Judge, A. Williams and R. Kermode, Implementation Experience with MANET Routing Protocols, *ACM SIGCOMM Computer Comm. Review* **32**(5) (Nov 2002), 49–59.
- [9] J. Singh, N. Bambos, B. Srinivasan and D. Clawin, Proposal and Demonstration of Link Connectivity Assessment based Applications to Routing in Mobile Ad-hoc Networks, *IEEE Vehicular Technology Conf* **5** Florida, (Oct 2003), 2834–2838.

- [10] J. Singh, N. Bambos, B. Srinivasan, D. Clawin and Y. Yan, Empirical Observations on Wireless LAN Performance in Vehicular Traffic Scenarios and Link Connectivity based Enhancements for Multihop Routing, *IEEE Wireless Comm. and Networking Conf* **3** (March 2005), 1676–1682.
- [11] J. Liu and V. Issarny, Signal Strength based Service Discovery (S3D) in Mobile Ad Hoc Networks, *Proc of the 16th Annual IEEE Int'l Symposium on Personal Indoor and Mobile Radio Communications (PIMRC)*, Germany, Sep 2005.
- [12] S. Kuo, Y. Tseng, F. Wu and C. Lin, A Probabilistic Signal-strength-based Evaluation Methodology for Sensor Network Deployment, *Int Journal of Ad Hoc and Ubiquitous Computing* **1**(1/2) (2005), 3–12.
- [13] L. Qin and T. Kunz, On-demand Routing in MANETs: The Impact of a Realistic Physical Layer Model, *Proc of the 2nd Int'l Conf Ad Hoc Networks & Wireless* (2003), 37–48.
- [14] A. Gupta, I. Wormsbecker and C. Williamson, Experimental Evaluation of TCP Performance in Multi-hop Wireless Ad Hoc Networks, *Proc of the IEEE Computer Society's 12th Annual Int'l Symposium on Modeling, Analysis and Simulation of Computer and Telecommunications Systems* Netherlands, (Oct 4–8 2004), 3–11.
- [15] J. Lee, S. Singh and Y. Roh, *Interlayer Interactions and Performance in Wireless Ad Hoc Network*, Internet-Draft, draft-irtf-ans-interlayer-performance-00.txt (work in progress), Sep 2003.
- [16] G. Lim, K. Shin, S. Lee, H. Yoon and J. S. Ma, Link Stability and Route Lifetime in Ad-hoc Wireless Networks, *Proc of the Int'l Conf on Parallel Processing Workshops (ICPPW'02)* (Aug 18–21 2002), 116–123.
- [17] T. Tien and S.J. Upadhyaya, A Local/Global Strategy Based on Signal Strength for Message Routing in Wireless Mobile Ad-Hoc Networks, *IEEE Academic/Industry Working Conf on Research Challenges* **3** (Apr 2000), 227–232.
- [18] T. Goff, N.B. Abu-Ghuzaleh, D.S. Phatak and R. Kahvecioglu, Preemptive Routing in Ad Hoc Networks, *Proc of the 7th Annual Int'l Conf on Mobile Computing and Networking* Italy, (2001), 43–52.
- [19] D.P. Agrawal and Q. Zeng, *Introduction to Wireless and Mobile Systems* **3** Thomson, (2003), 59–79.
- [20] L. Galluccio, A. Leonardi, G. Morabito and S. Palazzo, Tradeoff between Energy-Efficiency and Timeliness of Neighbor Discovery in Self-Organizing Ad Hoc and Sensor Networks, *Proc of the 38th Annual Hawaii Int'l Conf on System Sciences (HICSS'05)* **9**(9) (2005), 286.1.
- [21] R.R. Yager and D.P. Filev, *Essentials of Fuzzy Modeling and Control* **4** John Wiley & Sons, (1994), 109–153.
- [22] Available at: <http://www.omnetpp.org/>.
- [23] T. Camp, J. Boleng and V. Davies, A Survey of Mobility Models for Ad Hoc Network Research, *Wireless Communication & Mobile Computing (WCMC): Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications* **2**(5) (2002), 483–502.
- [24] D. Shukla, Mobility models in ad hoc networks, Master's Thesis, KReSIT-ITT, Bombay, Nov 29, 2001.
- [25] B. Kwak, N. Song and L.E. Miller, A Standard Measure of Mobility for Evaluating Mobile Ad Hoc Network Performance, *IEICE Trans. on Communication* **E86-B**(11) (Nov 2003), 3236–3243.
- [26] C. Bettstetter, Mobility Modeling in Wireless Networks: Categorization, Smooth Movement, and Border Effects, *ACM Mobile Computing and Communications Review* **5**(3) (2001), 55–67.
- [27] C. Bettstetter, G. Resta and P. Santi, The Node Distribution of the Random Waypoint Mobility Model for Wireless Ad Hoc Networks, *IEEE Trans on Mobile Computing* **2**(3) (2003), 257–269.
- [28] J. Broch, D.A. Maltz, D.B. Johnson, Y. Hu and J. Jetcheva. A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols, *Proc of the 4th Annual ACM/IEEE Int'l Conf on Mobile Computing and Networking (MOBICOM-98)* (Oct 1998), 85–97.
- [29] C. Wang, S. Chen, X. Yang and Y. Gao, Fuzzy Logic-Based Dynamic Routing Management Policies for Mobile Ad Hoc Networks, *Proc of the IEEE Workshop on High Performance Switching and Routing* (May 12–14 2005), 341–345.
- [30] E. Natsheh, A Survey on Fuzzy Reasoning Applications for Routing Protocols in Wireless Ad-Hoc Networks, *International Journal of Business Data Communications and Networking (IJBDCN)* **4**(2) (April–June 2008), 19–31.
- [31] E. Natsheh, S. Khatun, A.B. Jantan and S. Subramaniam, Fuzzy Metric Approach for Route Lifetime Determination in Wireless Ad-hoc Networks, *International Journal of Ad Hoc and Ubiquitous Computing (IJAHUC)* **3**(1) (2008), 1–9.
- [32] E. Natsheh, A.B. Jantan, S. Khatun and S. Subramaniam, Fuzzy Reasoning Approach for Local Connectivity Management in Mobile Ad-hoc Networks, *International Journal of Business Data Communications and Networking (IJBDCN)* **2**(3) (July–September 2006), 1–18.
- [33] E. Natsheh, A.B. Jantan, S. Khatun and S. Subramaniam, Intelligent Reasoning Approach for Active Queue Management in Wireless Ad-hoc Networks, *International Journal of Business Data Communications and Networking (IJBDCN)* **3**(1) (January–March 2007), 16–33.
- [34] Available at: <http://www.cs.unibo.it/~concer/>.

Essam Natsheh obtained his PhD in Communications and Networks Engineering from University Putra Malaysia in 2006. Currently, he is an Assistant Professor at the Computer Information Systems Department, College of Applied Studies, King Faisal University (Saudi Arabia). Beside that, he is doing his post-doctorate research at School of Computer Sciences, Universiti Sains Malaysia, in mobile ad-hoc networks, in particular, the development of a new routing algorithm for heterogeneous

density environment. Essam has more than ten years of teaching and research experiences in Malaysia and Saudi Arabia. Also, he has published 18 articles in journals and books at international level and over 10 papers in refereed conference proceedings.

Tat-Chee Wan received his BSEE and MSECE from University of Miami, FL in 1990 and 1993 respectively; and his PhD from Universiti Sains Malaysia, Penang, Malaysia, in 2005. He is currently Program Chairman for Computer Systems at the School of Computer Sciences, Universiti Sains Malaysia. His research interests include Wireless and Sensor Networks, Multicast protocols, QoS and embedded real-time systems.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

