

# TADL-V2: An Improved Trust-Aware Dynamic Location-based Adaptation Protocol For Discovering Multiple Paths in MANETs

Helen Bakhsh  
University of Manchester, UK  
helen.bakhsh@student.m-  
anchester.ac.uk

Dr. Ning Zhang  
University of Manchester, UK  
nzhang@cs.man.ac.uk

Dr. Andy Carpenter  
University of Manchester, UK  
andy@cs.man.ac.uk

## ABSTRACT

Mobile ad hoc networks (MANETs) have a number of features that make the provisioning of Quality of Services (QoS) particularly challenging. One of the most challenging issues is how to discover more reliable paths for high priority traffic delivery in a highly mobile network and do so with minimum overhead costs. This paper attempts to address this issue by designing and evaluating a multi-path discovery protocol. This protocol, named as Trust-Aware Dynamic Location-based protocol version 2 (TADL-V2), is an improved version of our earlier published protocol, TADL. TADL-V2 has improved TADL in two aspects: (1) it uses a hybrid approach to node-disjoint path discovery based on the network mobility level and (2) it uses a mobility based approach to search area resizing. These measures enable TADL-V2 to discover more paths with reduced number of control packets injected into the network. Our simulation study shows that TADL-V2 outperforms TADL in terms of reducing control overheads when the network is highly mobile. This overhead reduction can have a positive effect on QoS provisioning.

Keywords

## Keywords

MANETs, QoS, Trust, Multiple path, Node-disjoint

## 1. INTRODUCTION

A Mobile Ad hoc NETWORK (MANET) [1] is formed by a collection of wireless mobile nodes that are spontaneously inter-connected to communicate among themselves without having any infrastructural support. The nodes are mobile, and this node mobility causes link failures and paths breakage, resulting in frequent changes in the network topology, which, in turn, will cause packet losses, the lost packets will then be retransmitted and the repeated retransmissions may lead to network congestions thus further packet delivery delays and losses. All of these will have detrimental effects on Quality of Services (QoS). Providing QoS in MANETs

is a challenging issue. One of the challenging issues is how to find more reliable paths to deliver high priority traffic so as to satisfy their QoS requirements and how to do so with minimum overhead costs.

The *TADL-V2* protocol is an improved version of our early designed Protocol, *TADL* [2]. *TADL* was designed to discover the most reliable set of node-disjoint paths between a pair of source and destination nodes with minimum overhead costs. To achieve this goal, *TADL* has two novel measures embedded in its design. First, it uses an adaptive directional approach to path discoveries. That is, it defines a search area that covers the source and destination nodes and only broadcasts path discovery packets within this area. This can prevent unnecessary flooding of control packets across the entire network, but can also limit the number of paths that can be discovered. To maintain bandwidth cost low, while, at the same time, still being able to discover a sufficient number of paths, *TADL* adjusts the size of the search area dynamically in response to (a) the number of paths discovered in the previous discovery session, and (b) the underlying network conditions, namely neighbouring nodes average mobility level, attacker ratio, and the time of the previous search attempt. The second measure taken in the design is that it uses trust values (measuring link reliability) to govern the selection of neighbours during path formations, aiming at finding the most reliable set of path to serve high priority traffic. Reliability is measured in terms of the ratio of packets that are successfully delivered, as explained in [2].

Simulation results have shown that *TADL* outperforms other related protocols in a stationary network and a network with a low to medium level of mobility. However, when the network mobility level is high (e.g. when the network nodes move continuously, with 0 pause time), the performance of *TADL* decreases significantly, producing worse results than related protocols. This means that some of the design measures used in *TADL* is not cost-effective or even counterproductive when the mobility level is high.

In *TADL*, the discovering of a set of node-disjoint paths between a source and destination nodes is carried out using the method called the Source node Controlled Node-disjoint Path Discovery (*SC-MNPD*) method. This is a two-step process. In step-1, the source node initiates a path discovery process by multicasting a path discovery packet in a specified search area. Once step-1 is complete, i.e. once all the

paths in the searching area are discovered, the source node executes step-2 to select (a sub-set of) node-disjoint paths from the path set. In a highly dynamic network, the paths discovered in step-1 or selected in step-2 would be highly likely obsolete when they are used. If this is the case, packets transmitted along the paths will be lost, and the source node will be notified by the intermediate nodes involved, upon which the source node will initiate another path discovery process. Rediscovering the required number of paths will increase the number of control packets injected into the network, leading to degradation of protocol performance, as indicated in our simulation study.

To rectify this problem, this paper proposes an improved version of *TADL*, i. e. *TADL-V2*. The novelty of the *TADL-V2* design lies in three aspects. The first is to use an integrated approach to node-disjoint path discovery at a high network mobility level. Instead of first discovering all the available paths, and then selecting node-disjoint paths from the discovered path set, this approach (which is called the Intermediate node Controlled Node-disjoint Path Discovery (*IC-MNPD*)) performs the process in one step and is controlled by an intermediate node (rather than by the source node). That is, with this approach, the intermediate node will discover and determine node-disjoint paths during a path discovery process, thus preventing the transmissions of path discovery packets (i.e. control packets) along paths that are non-disjoint.

The second is to use the network mobility level to govern how node-disjoint paths should be discovered. In other words, *TADL-V2* switches between the two node-disjoint path discovery methods based on the network mobility level. When the network mobility level is low or medium, it uses the *SC-MNPD* method, and when the network mobility level is high, it switches to the *IC-MNPD* method. The reason for still using the *SC-MNPD* method when the network mobility level is at a low or medium level is that, under such mobility levels, the *SC-MNPD* method can discover more node-disjoint paths than the *IC-MNPD* method.

The third is to take a different measure in resizing the search area dictated by the mobility level. When the network mobility level is low or medium, it uses the same method used in *TADL*, and when the network mobility level is high, *TADL-V2* switches to a different method, where the source node starts the search from the largest searching area, Area-L, that covers the whole network.

We have used simulation studies to evaluate the effectiveness of the ideas used in *TADL-V2*. Simulation results show that, in a highly mobile network, *TADL-V2* outperforms *TADL* in terms of reducing the number of control packets injected into the underlying network, decreasing the average end-to-end packet delivery delays and increasing the packet delivery ratios. *TADL-V2* performs in exactly the same manner as *TADL* when the network mobility level is low or medium, so to avoid repetitions, in the remaining part of this paper, we only focus on evaluating the *TADL-V2* protocol under high network mobility setting.

The rest of this paper is organised as follows. Section 2 critically analyses related works. Section 3 gives the design pre-

liminary for *TADL-V2*. Section 4 describes the *TADL-V2* design and its operations. Section 5 reports the simulation study of the protocol. Finally, Section 6 concludes the paper and outlines our future work.

## 2. RELATED WORK

Most existing works in the area of MANET QoS provisioning focus on the design of a multi-path routing approach [3]. That is, they discover multiple paths between a given source and destination pair in a single path discovery process. However, the protocols may differ in terms of how the discovered multiple paths are used. In some multi-path protocols, one of the discovered paths is used as a primary path, and the others are used as secondary or backup paths [4]. In this way, packets can be promptly delivered along the secondary paths if the primary path breaks, thus reducing end-to-end delivery delays. In other multi-path protocols, multiple discovered paths are used in parallel [5], each supporting the delivery of a fraction of the traffic flow to be delivered. Using multiple paths in parallel can provide load balancing. It can also harvest the bandwidths of multiple paths to speed up the traffic delivery.

The multiple path discovery protocols may also differ in terms of the types of paths they are designed to discover. Basically, there are three types of paths, classified based on the level of disjointedness among nodes or links. They are node-disjoint paths, link-disjoint paths and non-disjoint paths. Node-disjoint paths are paths that have no common nodes except the source and destination nodes. During a path discovery process by such a protocol, the intermediate nodes reject all the duplicated copies of a *RREQ* (Route REQuest) packet. AODVM [6] is a protocol designed to discover node-disjoint paths.

Link-disjoint paths are paths that have common nodes but no common links between a source and destination pair. During a path discovery process by such a protocol, the intermediate nodes typically accept some of the duplicated copies of a *RREQ*, which have better attribute values, e. g. lower hop counts. The accepted *RREQ* packets must come from different upstream nodes. The Ad hoc On-demand Multi-path Distance Vector (AODMV) routing protocol [7] is designed to discover link-disjoint paths.

Non-disjoint paths are paths that share both nodes and links. In protocols designed to discover non-disjoint paths, intermediate nodes only play the role of packet forwarding. They are not involved in any decision making process; they simply accept and pass on all the *RREQ* packets they receive. The Split Multipath Routing (SMR) proposed in [5] and the TOpology-Hiding multi-path Protocol (TOHIP) in [8] are designed to discover non-disjoint paths.

There are trade-offs in discovering, selecting and using different types of paths, in terms of packet delivery reliability, control overhead incurred in discovering them and the number of available paths that can be discovered. Using multiple node-disjoint paths offers the highest level of packet delivery reliability, as if one intermediate node fails, only one path, i. e. the path containing the failed node, will be affected. Discovering node-disjoint paths also imposes less control overhead than link-disjoint and non-disjoint, as, during a path

discovery process, an intermediate node will reject all the duplicated copies of the same *RREQ* packet. However, the number of node-disjoint paths that can be discovered during a path discovery process is also the least among the three path types.

Conversely, protocols designed to discover multiple link-disjoint or non-disjoint paths do not reject duplicated *RREQ* packets at intermediate nodes. This allows the discovery of more paths. However, this also means that more *RREQ* packets will be injected into the underlying network resulting in more control overheads.

With the above observations from the related works and from the simulation study of the *TADL* protocol [2], we have re-designed the *TADL* protocol, producing *TADL-V2* to further reduce control overheads in the discovery of multiple node-disjoint paths and to make the discovery process even more efficient.

### 3. TADL-V2: AN OVERVIEW

This section gives a high-level overview of *TADL-V2*. In the *TADL-V2* protocol, each node can know the mobility levels of its neighbouring nodes by using the HELLO packets. The *TADL-V2* classifies the mobility levels into two bands, low/medium mobility level and high mobility level.

At the low/medium mobility level, the *TADL-V2* protocol (run at a source node) invokes the Source node Controlled Node-disjoint Path Discovery (*SC-MNPD*) method to discover all available node-disjoint paths linking the source node and a destination node. This process is carried out in two phases. In the first phase, the source node initiates a path discovery process in which all the non-disjoint paths linking the source and the destination nodes in the specified search area are to be discovered. In this phase, no effort is made to differentiate or identify path types. Once this process is complete, i.e. once all the available paths are discovered, the source node identifies and selects all possible sets of node-disjoint paths from the discovered paths [5]. This process is solely carried out by the source node, and the intermediate nodes are not involved. In other words, with this multiple node-disjoint path discovery method, all an intermediate node needs to do is to receive a *RREQ* and forward it onto a set of selected neighbours. The set of neighbours are chosen based on their trust values. Any duplicated *RREQ* packets (i.e. multiple copies of the *RREQ* packet with the same sequence number) are also forwarded. For every *RREQ* packet received, the destination node would return a *RREP* packet to the source node. When the source node receives all the *RREP* packets from the destination node, it goes through all the paths discovered and identifies all the possible sets of node-disjoint paths and selects one set (which has the best trust values and has sufficient aggregated bandwidth as required by the traffic flow) for traffic delivery.

When the mobility level is high, the path information carried by the *RREP* packets will likely to be obsolete by the time when these packets are received, or paths selected, by the source node. If these paths are selected, packets received by the upstream intermediate node of a broken link will be discarded, and the node will return a route error (*RRER*)

packet to the source node. The source node, upon the receipt of *RRER*, would restart another path discovery session if the source node cannot find an alternative path(s) in its routing table. So the higher the network mobility, the more likely the links would break, the more *RRER* packets will be returned to the source node, and more route discovery sessions will be initiated. As a result, the more control overheads will be generated. A higher level of control overheads would be more likely to make the network more congested, causing more packets to be dropped.

To prevent the network get into such a state, *TADL-V2* uses a Mobility-based Adaptive Multiple Node-disjoint Path Discovery (*MA-MNPD*) method. With this method, *TADL-V2* uses the Source node Controlled Multiple Node-disjoint Path Discovery (*SC-MNPD*) method (i. e. the method described above and used in *TADL*) at a low or medium network mobility level, but switches to the Intermediate node Controlled Node-disjoint Path Discovery (*IC-MNPD*) method if the network mobility level is high. With this *IC-MNPD* method, it is not the source node, but every intermediate node, which filters out any duplicated *RREQ* packets and selects node-disjoint paths *RREQ* during the process of packet forwarding. In this way, the intermediate node will only need to forward *RREQ* packets along the selected node-disjoint paths, and as mentioned above, the transmission of the *RREQ* packets along non-disjoint paths can be prevented, thus reducing the number of control packet poured into the network.

As mentioned in the Introduction section, reducing the number of control packets in this way can be counterproductive to discovering a required set of node-disjoint paths between a pair of nodes. To offset any potential loss in the number of node-disjoint paths discovered, *TADL-V2* uses a different approach to search area resizing. It uses the search area resizing method used in *TADL* (i.e. the *AS2A* method) in conjunction with the *SC-MNPD* method when the mobility level is low or medium, but switches to the *MA-AS2S* method in conjunction with the *IC-MNPD* method when the network mobility level is high. In the *AS2A* method, the size of the search area is adjusted dynamically in response to the underlying network conditions. As shown in Figure 1, three distinctive sizes of the search area are specified, small (Area-*S*), medium (Area-*M*) and large (Area-*L*).

Area-*S* is the smallest one, which is defined by using the *DSA* method [2]. The *DSA* method defines the destination node searching area that used to forward control packets to the destination node. The searching area is a rectangular area that starts from the current location of *S* and includes the predicted location of *D* at the current time  $T_{cur}$ , as shown in Figure 2. Due to the frequent node mobility, *S* attempts to predict the location of *D* using the location information stored on its routing table.

Area-*M* is the next-level size (medium). The size is such that the size of the expected zone circle around the destination node double the one used in Area-*S*. Doubling the size of the expected zone increases the area (i.e. search area) in which *RREQ* packets are broadcast and this may lead to more paths be discovered.

Area- $L$  is the largest search area that covers the entire network. In other words, when Area- $L$  is used,  $RREQ$  packets will be broadcast to the entire network, defaulting to the use of the flooding path discovery protocol.

Each path search can start from any of the three search area sizes based on the number of discovered paths in the previous discovery session and observations on the channel conditions of neighbouring nodes.

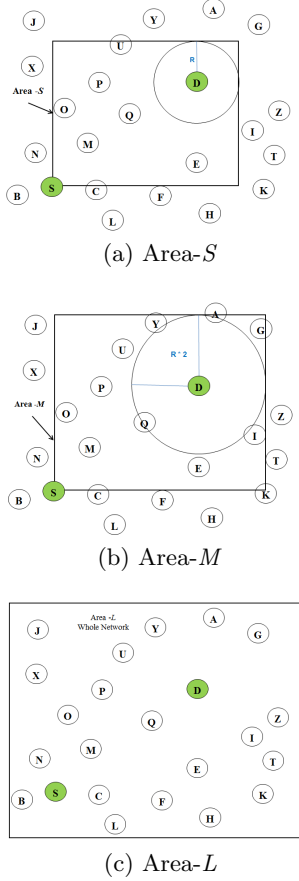


Figure 1: Searching Area Sizes

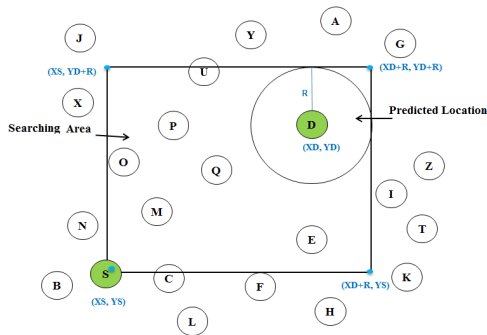


Figure 2: Predicted Location and Searching Area of D

In method  $AS2A$ , the discovery process starts with the smallest search area, i.e. Area- $S$ , and then progressively moves to larger sizes if the number of paths discovered is less than re-

quired. Using Area- $S$  will lead to the least control overhead being generated, but also the fewest paths being discovered. If the number of discovered paths is less than the required number, method  $AS2A$  will move to Area- $M$  in the next path discovery session, and so on.

With a higher mobility level, the discovered paths will be more likely to be void; in which case, another discovery session may have to be initiated causing more control overheads. So, under such cases, it makes more sense to discover as many paths as possible, and in the event when more paths are required, the source node will have more paths available in the routing table as backups. Based on this consideration, the Mobility-based Adaptive Adjusting Searching Area Algorithm ( $MA-AS2A$ ) is designed, i.e. it starts the search from the largest searching area, Area- $L$ , which covers the whole network, in a higher mobility level network.

In  $MA-AS2A$ ,  $AS2A$  is used when the network mobility level is at a low to medium level. When the network mobility level is high,  $TADL-V2$  switches to Area- $L$ . The probability of finding a large number of valid paths is high with a large search area. Thus, when one set of path is invalid, there will be alternative sets to be used immediately without the need of reconstructing paths and starting a new path discovery process. The  $MA-AS2A$  method is explained in detail in the next section.

#### 4. TADL-V2: LOW-LEVEL DESIGN

This section describes the  $TADL-V2$  protocol, its components and operations

##### 4.1 TADL-V2 Components

$TADL-V2$  consists of a number of components, which collectively perform the functions defined for  $TADL-V2$ . These components are executed on each node in the network. As shown in Figure 3, the components are three  $TADL-V2$  tables, the Neighbouring Node Trust Value Estimation ( $TVE$ ) method, the Mobility-based Adaptive Multiple Node-disjoint Path Discovery ( $MA-MNPD$ ) method and the Mobility-based Adaptive Adjusting Searching Area Algorithm ( $MA-AS2A$ ). This section describes these components in details.

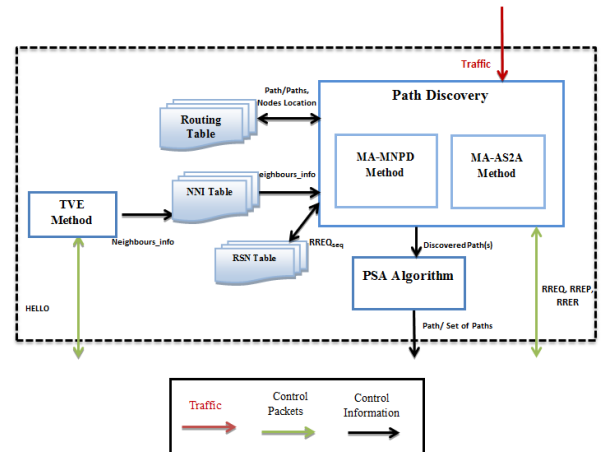


Figure 3: TADL-V2 Components

##### 4.1.1 TADL-V2 Tables

Each node maintains three tables used during route discovery. These are a Routing table, a Neighbouring Node Information table, and a *RREQ* Sequence number (*RSN*) table.

The Routing table contains information to determine how to forward the data and control packets. For every discovered node in the network, there is an entry in the table, and the entry contains the Node Location Information and Path(s) to that node.

A neighbouring node information table stores attribute values for each of its directly connected neighbours. The attributes are the Location information of the Neighbouring Node, and the Trust Value of the link connecting this node to the neighbouring node.

The *RSN* table records the sequence number of all the *RREQ* packets that has passed this node during the current path discovery process for a given source and destination pair. The *RREQ* sequence number of a *RREQ* packet,  $RREQ_{seq}$ , *RREQ* is used to distinguish this *RREQ* or copies of this *RREQ* from other *RREQ*s.

#### 4.1.2 Neighbouring Node Trust Value Estimation (TVE)

##### Method

*TVE* estimates a trust value for each of the Neighbouring nodes by using a local trust model. The trust value of a neighbouring node measures the reliability of the link connecting this node to the neighbouring node. In other words, the trust value assigned to node  $N_i$  actually reflects the reliability of the link linking this node and node  $N_i$ .

#### 4.1.3 Paths Selection Algorithm (PSA)

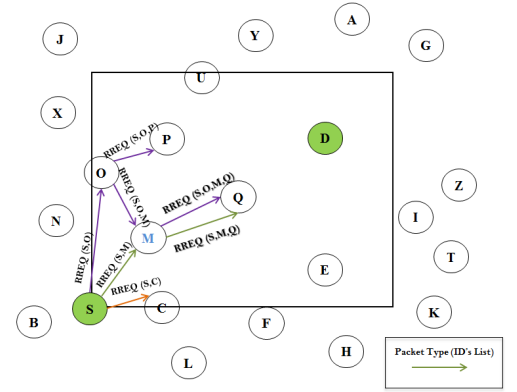
*PSA* is used to select a set of most trusted paths to forward traffic to the destination node. It is executed after the path discovery process.

#### 4.1.4 Mobility-based Adaptive Multiple Node-disjoint Path Discovery (MA-MNPD) Method

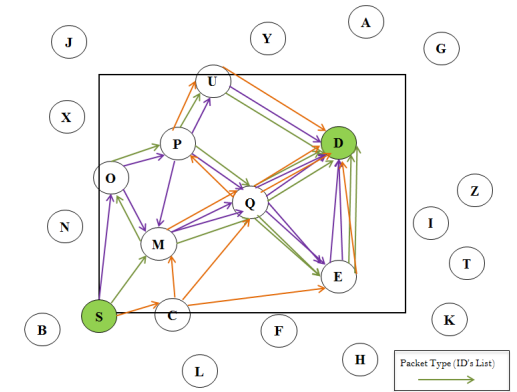
The *MA-MNPD* method is used for discovering as many paths as possible between a pair of source and destination nodes and do so with as less control overheads as possible under all network conditions, particularly when the network nodes are highly mobile. The method uses the Source node Controlled Multiple Node-disjoint Path Discovery (*SC-MNPD*) method (i.e. the method used in *TADL*) when the network mobility level is at a low to medium level, and *TADL-V2* switches to the Intermediate node Controlled Node-disjoint Path Discovery (*IC-MNPD*) method when the network mobility level is high.

*SC-MNPD* is a source node controlled path discovery method that is used by the source node to first discover all the available paths to a destination node, and then, the source node identifies a subset of paths that are node-disjoint from the set of paths discovered. An intermediate node, upon receiving a *RREQ* packet, simply forwards it onto a set of selected neighbours by using the neighbours selection algorithm (*NSA*). This set of neighbours is chosen based on their trust values. Multiple *RREQ*s with the same sequence number are also forwarded on. Figure 4 uses an example to

illustrate the multiple path discovery process using the *SC-MNPD* method.



(a) Intermediate Node Accepts Second *RREQ*



(b) The Discovered Paths

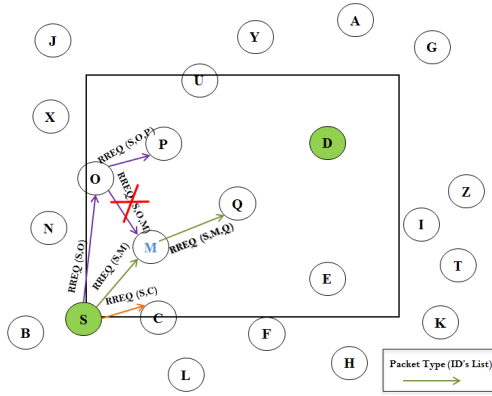
**Figure 4: *SC-MNPD* Method**

As shown in Figure 4(a), the source node, *S*, sends a *RREQ* packet to a selected set of neighbouring nodes, *O*, *M* and *C*. When node *M* receives the *RREQ* packet from node *S*, it accepts the packet and passes it to a set of selected neighbouring nodes. Next, *M* receives another copy of the same *RREQ* coming from a different path, i. e.  $S \rightarrow O$ , and it also accepts the packet and passes it on. As shown in Figure 4(b), for this single path discovery session, the destination node will eventually receives 14 *RREQ* packets through different paths, and, for each *RREQ* it receives, the destination node sends a *RREP* back to the source node.

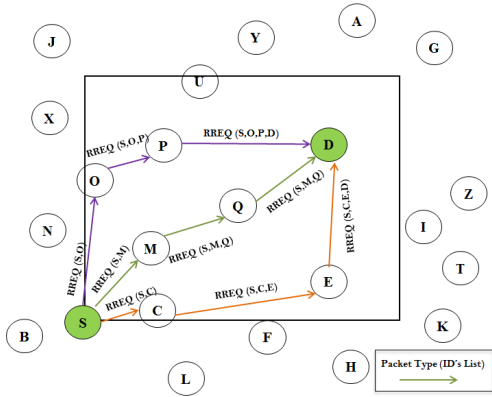
*IC-MNPD* discovers multiple node-disjoint paths with reduced control overhead. *IC-MNPD* is executed by intermediate nodes during a path discovery process. An intermediate node will only pass the *RREQ* once, filters out any duplicated *RREQ*s. Each node stores the sequence numbers carried in all the *RREQ*s that pass through the node in a *RREQ* Sequence Number (*RSN*) table.

As shown in Algorithm 1, upon the receipt of a *RREQ*, the intermediate node gets the *RREQ* sequence number of the packet,  $RREQ_{seq}$ , and search the *RSN* table for a match. If a *RREQ* match is found, the node will discard this *RREQ*. Otherwise, the node will accept the *RREQ*, adds the *RREQ* sequence number,  $RREQ_{seq}$ , into the *RSN* table. At the

end of the discovery process, only one set of node-disjoint paths will be discovered. Figure 5 uses an example to illustrate the working of the *IC-MNPD* method.



(a) Intermediate Node Rejects Second RREQ



(b) The Discovered Paths

**Figure 5: IC – MNPD Method**

As shown in Figure 5(a), the source node  $S$  sends a *RREQ* packet to a selected set of neighbouring nodes,  $O$ ,  $M$  and  $C$ . Node  $M$  filters out a duplicated *RREQ*. It receives and accepts the first *RREQ* from node  $S$ . When  $M$  receives another copy of the same *RREQ* coming from a different path, i. e.  $S \rightarrow O$ , it rejects it. As shown in Figure 5(b), in this way, only one set of node-disjoint paths is discovered, and the number of control packets propagated in the network is also reduced.

**Algorithm 1** The IC-MNPD method

---

```

procedure THE IC-MNPD METHOD(RREQ)
  Search  $RSN$  for  $RREQ_{seq}$ 
  if  $RREQ_{seq}$  is found then
    RREQ Rejected  $\triangleright$  Avoid RREQ Duplication
  else
    RREQ Accepted
    Add  $RREQ_{seq}$  to the  $RSN$  table
  end if
end procedure

```

---

In *MA-MNPD*, as shown in Algorithm 2, upon receiving

a *RREQ* packet, an intermediate node calculates the average neighbouring nodes mobility level,  $AvrMob_{neighbouring}$ . If the calculated value is high (above a predefined mobility level threshold,  $THR_{mob}$ , *TADL-V2* uses the *IC-MNPD* method for path discovery. Otherwise, i.e., when the calculated value is below this threshold value, it uses the *SC-MNPD* method. All the threshold values are proven experimentally.

**Algorithm 2** The MA-MNPD method

---

```

procedure THE MA-MNPD METHOD(RREQ)
  Calculate the  $AvrMob_{neighbouring}$ 
  if  $AvrMob_{neighbouring} \geq THR_{mob}$  then
    Use SC-MNPD method
  else
    Use IC-MNPD method
  end if
end procedure

```

---

#### 4.1.5 Mobility-based Adaptive Adjusting Searching Area Algorithm (*MA-AS2A*):

*MA-AS2A* adjusts the size of a searching area used in a path discovery process. It implements a set of rules to govern the selection of the start search area size for each route discovery process based on the neighbouring nodes mobility level.

As mention earlier, there are three searching area sizes (Area- $S$ , Area- $M$ , or Area- $L$ ). The path discovery protocol should start with any of these sizes in each path discovery session. Intuitively, this choice may be made based on the number of discovered paths in the previous search. For example, if the number of discovered paths in the previous search is equal to the required amount of paths, the current search may start from the same searching area size used in the previous search. If the number of paths discovered in the previous search is higher than the required number of paths, then the current search decrease the previous search area. If the number of paths discovered in the previous search is less than the current required number of paths, the current search area uses larger searching area than the previous searching area.

However, the above decision rules may be adequate if the underlying network is static, which is typically not true in a mobile MANET. In addition, there are three other factors which should be considered when choosing a searching area size to start a search. These factors are the neighbouring nodes attacker ratio, the neighbouring nodes average mobility level, and the time of the previous search attempt.

*Factor 1 - neighbouring nodes attacker ratio:* This is the ratio of the number of neighbouring nodes attackers over the total number of neighbouring nodes. When the neighbouring nodes attacker ratio is high, the possibility of finding trustworthy paths in a small searching area is low. Conversely, the probability of finding a sufficient number of trustworthy paths in a larger searching area is higher.

*Factor 2 - neighbouring nodes average mobility level:* This is the average mobility level of the neighbouring nodes. If the average mobility level is high, the network topology will change more frequently and the paths discovered in the pre-

vious occasion would be more likely to become obsolete and the probability of finding a sufficient number of trustworthy paths in a larger searching area is higher.

*Factor 3 - the time of the previous search attempt  $T_{prv}$ :* The topology is not only affected by the node mobility, but it also changes with time. The longer the time elapses since the last search operation, the more likely the topology will change and the paths discovered in the previous search operation is considered as obsolete. The above discussed rules have been implemented in Algorithm 3.

---

**Algorithm 3** MA-AS2A Algorithm

---

```

1: procedure MA-AS2A
2:   if AttackRatio  $\geq$   $THR_{att}$  then
3:      $SAS_{cur} \leftarrow Area - L$ 
4:   else
5:     if  $AvrMob_{neighbouring} \geq THR_{mob}$  then
6:        $SAS_{cur} \leftarrow Area - L$ 
7:     else
8:       if  $T_{cur} - T_{prv} \geq THR_T$  then
9:          $SAS_{cur} \leftarrow Area - S$ 
10:      else
11:        if  $NPS_{prv} = NPS_{req}$  then
12:           $SAS_{cur} \leftarrow SAS_{prv}$ ;
13:        else
14:          if  $NPS_{prv} > NPS_{req}$  then
15:             $SAS_{cur} \leftarrow SAS_{prv} - 1$ ;
16:          else
17:             $SAS_{cur} \leftarrow SAS_{prv} + 1$ ;
18:          end if
19:        end if
20:      end if
21:    end if
22:  end if
23:  return  $SAS_{cur}$ 
24: end procedure

```

---

As shown in the Algorithm 3, *MA-AS2A* starts by checking the attacker ratio of the neighbouring nodes. If the neighbouring nodes attacker ratio is above a predefined threshold value,  $THR_{att}$ , *MA-AS2A* starts the search from the largest searching area,  $Area-L$ , which covers the whole network. Otherwise, *MA-AS2A* checks the mobility level of the neighbouring nodes. If the average mobility level is above a predefined threshold value,  $THR_{mob}$ , *MA-AS2A* starts with the largest searching area,  $Area-L$ .

Otherwise, If the neighbouring nodes attacker ratio is less than  $THR_{att}$  and the average mobility level is less than  $THR_{mob}$ , *MA-AS2A* checks the time of the previous search attempt  $T_{prv}$ . If  $T_{prv}$  is above a specified threshold value for the time it elapses since the last search operation,  $THR_T$ , *MA-AS2A* starts from the smallest searching area,  $Area-S$ . Otherwise, the choice is made based on the number of discovered paths in the previous search,  $NPS_{prv}$ . If  $NPS_{prv}$  is equal to the required amount of paths,  $NPS_{req}$ , the current search may start from the same searching area size used in the previous search,  $NPS_{prv}$ . If  $NPS_{prv}$  is higher than  $NPS_{req}$ , then the current search will use a smaller search area than the previous search. If  $NPS_{prv}$  is less than

$NPS_{req}$  the current search will use a larger search area than the previous one.

## 4.2 The TADL-V2 Protocol Operation

This section describes the operation of the *TADL-V2* protocol and how it is used to discover node-disjoint multiple paths. The protocol consists of a *TADL-V2* source node operations, a *TADL-V2* intermediate node operations and a *TADL-V2* destination node operations. We only focus on the source and intermediate node components, as the destination component is identical to the one used in *TADL* [2] *TADL-V2*.

### 4.2.1 The TADL-V2 Source Node Operations:

This component searches the network to find as many number of paths linking a source node  $S$  to a destination node  $D$ . As shown in Algorithm 4, *TADL-V2* the source node,  $S$ , first searches its routing table for a valid set of paths to the destination node,  $D$ . If  $S$  finds a valid set,  $S$  it uses these paths to transmit the high priority traffic. Otherwise,  $S$  reads  $D$  physical location from its routing table. If  $S$  does not know the location of  $D$ ,  $S$  defaults to use the basic broadcast method to discovery paths. In this case, the source node  $S$  and all the intermediate nodes that receive the *RREQ* packet will broadcast the *RREQ* packet to all the neighbouring nodes, until the destination node is reached.

---

**Algorithm 4** The TADL-V2 source node Operation

---

```

1: procedure TADL-V2 SOURCE NODE(Traffic)
2:   Search for valid set of paths to  $D$ 
3:   if Paths to  $D$  is found then
4:      $SelectedPaths \leftarrow PSA(DiscoveredPaths)$ 
5:     Transmit traffic over  $SelectedPaths$ 
6:   else
7:     Search for  $D$ 's physical location
8:     if  $D$ 's physical location is not found then
9:        $SelectedNeighbours \leftarrow Allneighbours$ 
10:    else
11:      do
12:         $SAS_{cur} \leftarrow MA-AS2A()$ 
13:         $SelectedNeighbours \leftarrow NSA(SAS_{cur})$ 
14:        Send RREQ to the  $SelectedNeighbours$ 
15:         $S$  initiates a RDTimer timer
16:        Wait for a RREP(s) from  $D$ 
17:        if RREP(s) arrived then
18:           $SelectedPaths \leftarrow$  ←
19:             $PSA(DiscoveredPaths)$ 
20:          Transmit Traffic over  $SelectedPaths$ 
21:        else
22:          if  $SAS_{cur} == Area - L$  then
23:             $D$  is an unreachable destination
24:            Exit
25:          end if
26:           $SAS_{cur} \leftarrow SAS_{cur} + 1$ 
27:        end if
28:      while  $SAS_{cur} != Area - L$ 
29:    end if
30:  end if
31: end procedure

```

---

If  $S$  knows the location of  $D$ ,  $S$  uses the *MA-AS2A* algo-

rithm to identify the starting search area size (Section 4.1.5). In this case, the *RREQ* will be transmitted to a set of most trusted neighbours selected using the algorithm *NSA* (described in [2]).

Upon the transmission of the *RREQ* packet, node *S* initiates a Route Discovery Timer (*RDTimer*). By the expiration of this timeout interval, if *S* does not receive a sufficient number of Route REPLY (*RREP*) packets, a new *RREQ* will be transmitted in a larger search area than the one just used (as explained in the *MA-AS2A* algorithm). If, by the expiration of *RDTimer*, *S* receives a sufficient number of *RREP* packets, *S* will invoke *PSA* algorithm (explained in Section 4.1.3) to select a set of node-disjoint paths from the discovered paths. *S* then sends the data traffic via the selected paths to the destination node.

#### 4.2.2 The TADL-V2 Intermediate Node Operations:

This component implements the novel solution, the *MA-MNPD* method, described in Section 3. *MA-MNPD* is executed by intermediate nodes when the neighbouring nodes mobility level is high. Intermediate nodes are responsible for forwarding the *RREQ* packet towards the direction of the destination node until the destination node is reached. They do not reply to the *RREQ* packet back to the source node. They are also responsible for avoiding *RREQ* looping. As shown in Algorithm 5, *TADL-V2* intermediate node component operates as follows.

Upon receiving a *RREQ* packet, the intermediate node checks the path list in the *RREQ*, *RREQ<sub>path-list</sub>*. If the intermediate node ID, *Node<sub>ID</sub>*, is listed in the *RREQ<sub>path-list</sub>*, the intermediate node discards the *RREQ* packet to avoid *RREQ* looping, as this means that this copy of the *RREQ* has already passed through this intermediate node before. Then the intermediate node invokes the *MA-MNPD* method. Upon the receipt of an *RREQ* packet, the method checks the neighboring nodes mobility level. If the network mobility level is high, the *MA-MNPD* method executes the *IC-MNPD* method which rejects any duplicated *RREQ* packets and selects node-disjoint paths as it passes *RREQ* packets on. It accepts the *RREQ* only once. Otherwise, if the mobility level is medium to low, the *MA-MNPD* method executes the *SC-MNPD* method which accepts all the *RREQ*s arrived. No effort is made to differentiate or identify path types. All the available paths are discovered.

---

#### Algorithm 5 TADL-V2 Intermediate node Operation

---

```

procedure TADL-V2 INTERMEDIATE NODE(RREQ)
  Search RREQpath-list for NodeID
  if NodeID is found then
    RREQ Rejected ▷ Avoid Looping
  else
    MA-MNPD(RREQ)
  end if
  if The RREQ is accepted then
    SelectedNeighbours ← NSA(SAScur)
    Add node's information to RREQpath-list
    Send RREQ to the SelectedNeighbours
  end if
end procedure

```

---

Upon accepting the *RREQ* packet, the intermediate node adds the values of its attributes onto the *RREQ<sub>path-list</sub>* in the *RREQ* header. These attributes are the node ID and location information, and the trust values of the selected down-stream neighbours. It then forwards the *RREQ* packet on to the chosen neighbours which have been selected using the *NSA* algorithm (described in [2]).

## 5. SIMULATION STUDY

This section presents the results of our simulation study of the *TADL-V2* protocol and compare the results against those from the *TADL* protocol.

### 5.1 Performance Metrics

The simulation study is carried out using four performance metrics, i. e. control overhead, packet delivery ratio, average end-to-end packet deliver delay and discovered paths.

*Control Overhead* [5] refers to the total number of control packets transmitted by all the nodes in the network divided by the total number of data packets successfully received by all the destination nodes.

*Packet Delivery Ratio* is the ratio of the total number of data packets successfully received by the destination nodes to the total number of data packets transmitted by the source nodes.

*Average End-to-End Packet Delivery Delay* is the average time difference between when a data packet is transmitted by the source node and when the data packet arrives at the destination node.

*Discovered Paths* is the total number of the discovered paths during a path discovery process by all the source nodes in the network.

### 5.2 Simulation Model

The study is carried out using the network simulator NS-2 (version 2. 27. 2) [9]. The network consists of 50 nodes located in a 1000m x 1000m area. Nodes move according to the Random Waypoint Mobility model. They move with a minimum speed of 1 m/s, and a maximum speed of 19 m/s. The duration of each simulation run is 900 seconds. Constant bit rate (CBR) traffic is transmitted at a rate of 4 packets per second. The packet size is 512 bytes. The traffic load used is transmitted by 40% of the nodes. 30% of the source nodes transmits high priority traffic, and the remaining 70% of source nodes transmits low priority traffic. These parameter settings are commonly used in the relevant studies published in literature [10].

### 5.3 Simulation Results

This section describes the simulation results and compares the results of the *TADL-V2* protocol with those from the *TADL* protocol. *TADL-V2* performs in exactly the same manner as *TADL* when the network mobility level is low or medium, they differs only in how node-disjoint paths are discovered when the network mobility level is high. There are two foundational differences.

Firstly, to discover all available node-disjoint paths linking



the source node and a destination node, *TADL* uses *SC-MNPD* method that discovers all the non-disjoint paths linking the source and the destination nodes in the specified search area. Then the source node identifies and selects all possible sets of node-disjoint paths from the discovered paths. Whereas, *TADL-V2* uses the *MA-MNPD* method that uses the *SC-MNPD* method at a low or medium network mobility level, but switches to the *IC-MNPD* method if the network mobility level is high. In *IC-MNPD*, intermediate node are involved during the process of packet forwarding. They filters out any duplicated *RREQ* packets and selects node-disjoint paths *RREQ*.

Secondly, *TADL* uses *AS2A* method that starts the discovery process with the smallest search area and then progressively moves to larger sizes if the number of paths discovered is less than required. Whereas, to optimize the trade-off between the number of discovered paths, and the level of traffic poured into the network, *TADL-V2* uses the *MA-AS2A* method that uses *AS2A* when the network mobility level is at a low to medium level and when the network mobility level is high, *TADL-V2* starts with the largest search area.

The study shows the effects of varying network node mobility levels on control overheads, packet delivery ratios, average end-to-end packet delivery delays and the number of discovered paths (Figure 6). Different pause times are used to reflect the mobility levels. These are 0, 300, 600, and 900 seconds. The 0 second pause time indicates that the network nodes move continuously, whereas the 900 second pause time indicates the network nodes are stationary. Results are averaged over 30 simulation runs, with a 2.86 confidence level.

The simulation results from Figure 6 show that the *TADL* and *TADL-V2* protocols are almost the same in a stationary network or a network with a low to medium level of mobility. Whereas, *TADL-V2* significantly outperforms *TADL* in a high mobility network. When the mobility level increases, links break very frequently causing packet delivery to fail, and in this case, the sending node will return a route error (*RREER*) packet to the source node. For each *RREER* received, the source node restarts a new path discovery process if the source node did not find an alternative path(s). The new path discovery process increase the number of control packets, thus increases the control overhead. From Figure 6(a), it can be seen that the number of control overhead increases slightly in *TADL-V2*, but in *TADL*, it increases sharply. The different levels in the control overhead increase can be contributed to the fact that, in *TADL*, more *RREQ* will be passing the intermediate nodes for each search leading to a higher level of control overhead. Whereas, in *TADL-V2*, when the network mobility level is high, the intermediate node only passes the *RREQ* once, filters out any duplicated *RREQ*. Otherwise, when the mobility level is low to medium level, *TADL-V2* acts the same as *TADL*, they pass all the *RREQ* to allow discovering more paths. The control overhead reduction indicates that, to discover multiple paths in a high mobility network, adjusting the injection of control packet in the network dynamically based on the node mobility level, can significantly reduce the control overhead in the network.

Figure 6(b) shows the effect of varying network node mobil-

ity levels on the packet delivery ratio. From the figure, it can be seen that, the PDR ratios decreases as the network mobility level increases. This observation is in line with our expectation - high mobility level means more links break and more path discovery needed, which requires more control packets to discover alternative paths. This increase the chance of the network being congested, which can lead to PDR drops. From the figure, we can see that the PDR of *TADL-V2* is better than *TADL* in a high mobility network as the first produce less control overhead than the second. Less congested networks allow more packets to be delivered.

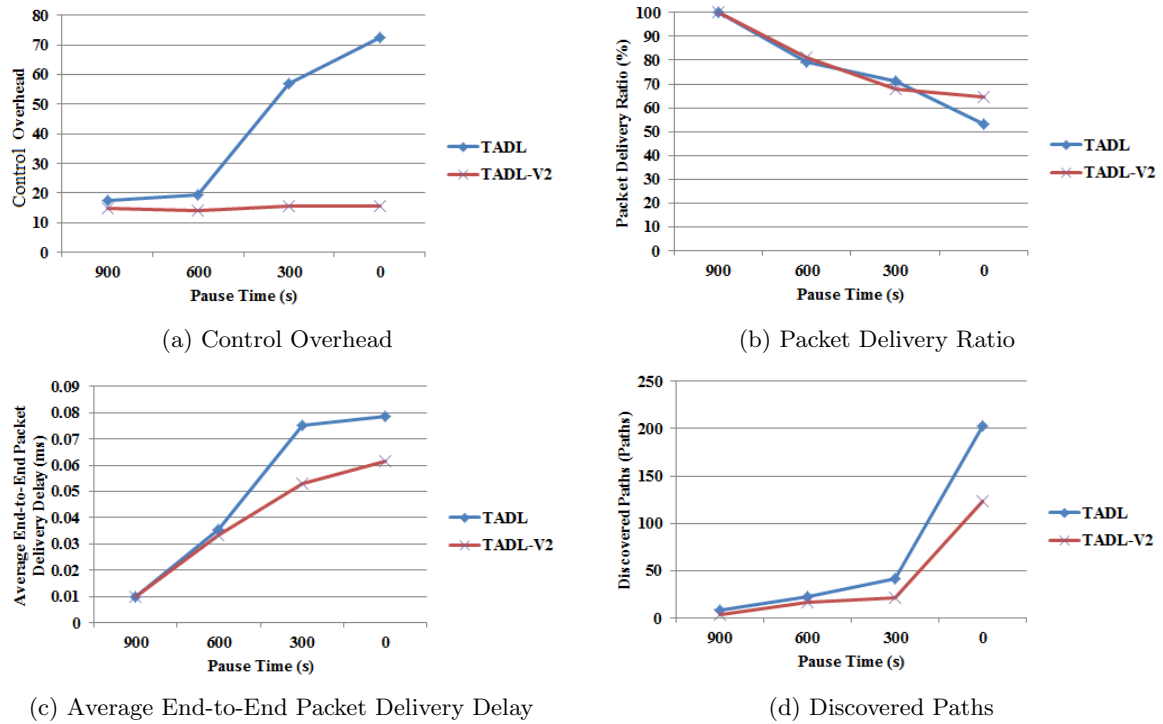
Figure 6(c) shows the effect of varying network node mobility levels on the average end-to-end packet delivery delay. The figure shows that *TADL-V2* performs better than *TADL* in a high mobility network with a greater difference in delay between *TADL-V2* and *TADL* is 42.3% at 0s pause time. This is a consequence of the reduced control overhead of *TADL-V2*; In *TADL*, the high control overhead congests the network; the data packets take a longer time at the congested intermediate nodes to be served.

Figure 6(d) shows the effect of varying network node mobility levels on the number of discovered paths. When the mobility level increases more paths are being discovered. This is due to more frequent link breaks, and more paths are required to be discovered as mentioned earlier. It also can be shown from the figure that, in *TADL-V2*, decreasing the *RREQ* packets during the path discovery process in a high mobility level, has no significant effects on the number of discovered paths. In *TADL*, searching for paths in a small searching area will cause fewer paths to be discovered; and in the high mobility level, more paths needed to be reconstructed, as the discovered paths will be no longer available and no alternative paths can be used immediately. Whereas, in *TADL-V2*, involving the intermediate node in the node-disjoint path decision in *TADL-V2* reduces the number of control packet poured into the network (as shown in Figure 6(a)). However, reducing the number of control packet in this way reduces the number of discovered paths. As, at the end of the discovery process, only one set of node-disjoint paths will be discovered. On the other hand, increasing the searching area, increases the probability of finding a large number of valid paths.

## 6. CONCLUSIONS

This paper has presented the design and evaluation study of a novel multiple node-disjoint path discovery protocol, *TADL-V2*. It overcomes the limitations identified in an earlier published protocol, the *TADL* protocol. *TADL-V2* integrates two novel methods, an intermediate node controlled node-disjoint path discovery method and a mobility-based adaptive search area adjustment algorithm, into a trust-aware dynamic location-based multi-path discovery protocol for MANETs. This integration enables multiple node-disjoint paths to be discovered with less control overheads in a highly mobile network. Simulation results show that *TADL-V2* provides a better performance than *TADL* in a high mobility network in terms of reducing control overheads. This overhead reduction has a positive effect on packet delivery ratio and end-to-end packet delivery delay.

There are pros and cons with the use of the two different



**Figure 6: Network Nodes Mobility Levels Vs control overhead, Packet Delivery Ratio, Average End-to-End Packet Delivery Delay and Discovered Paths for *TADL-V2* and *TADL***

approaches, *SC-MNPD* and *IC-MNPD*. The former places less trust on intermediate nodes, but introduces more control overheads into the underlying network, whereas the latter *TADL-V2* assumes more trust on the intermediate nodes, but can reduce control overhead by 76.4% at 0s pause time. As our future work, we shall investigate how to identify and counter any misbehavior by the intermediate node during a path discovery process, and try to find out a solution to lower the level of trust on the intermediate node.

## Acknowledgments

Helen Bakhsh gratefully acknowledges King Abdulaziz University (KAU) and Saudi Ministry of Higher Education for their financial supports.

## 7. REFERENCES

- [1] M. Wang, "Manet global connectivity and mobility management using hmip6 and olsr," Ph.D. dissertation, Carleton University, 2003.
- [2] H. Bakhsh, N. Zhang, and A. Carpenter, "Tadl: A trust-aware dynamic location-based protocol for discovering multiple paths in manets," in *Proceedings of the 16th ACM International Conference on Distributed Computing and Networking*, ser. ICDCN '15. Goa, India: ACM, 2015.
- [3] S. Mueller, R. Tsang, and D. Ghosal, "Multipath routing in mobile ad hoc networks: Issues and challenges," *Performance Tools and Applications to Networked Systems*, pp. 209–234, 2004.
- [4] M. Gerharz, C. Vogt, and C. De Waal, "Current approaches towards improved quality-of-service provision in mobile ad-hoc networks," *Computer Science Department, Communications Systems*, 2003.
- [5] S. Lee and M. Gerla, "Split multipath routing with maximally disjoint paths in ad hoc networks," in *Communications, 2001. ICC 2001. IEEE International Conference on*, vol. 10. IEEE, 2001, pp. 3201–3205.
- [6] Z. Ye, S. V. Krishnamurthy, and S. K. Tripathi, "A framework for reliable routing in mobile ad hoc networks," in *IEEE INFOCOM*, 2003, pp. 270–280.
- [7] M. Marina and S. Das, "On-demand multipath distance vector routing in ad hoc networks," in *Network Protocols, 2001. Ninth International Conference on*, Nov 2001, pp. 14–23.
- [8] Y. Zhang, T. Yan, J. Tian, Q. Hu, G. Wang, and Z. Li, "Tohip: A topology-hiding multipath routing protocol in mobile ad hoc networks," *Ad Hoc Networks*, vol. 21, no. 0, pp. 109 – 122, 2014.
- [9] T. N. Simulator, "ns-2. <http://nsnam.isi.edu/nsnam/index>. retrieved;," May 2014. [Online]. Available: <http://nsnam.isi.edu/nsnam/index>
- [10] P. J. McNerney and N. Zhang, "A 2-dimensional approach to qos provisioning in adversarial mobile ad hoc network environments," in *Proceedings of the 15th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, ser. MSWiM '12. New York, NY, USA: ACM, 2012, pp. 143–150. [Online]. Available: <http://doi.acm.org/10.1145/2387238.2387264>