VERIFYING ROUTE REQUEST PROCEDURE OF AODV USING GRAPH THEORY AND FORMAL METHODS

Shakeel Ahmed[1], A. K. Ramani[1] and Nazir Ahmad Zafar[2]

[1]SCSIT, Devi Ahilya University, Indore, India
Email: {shakeel_rahi, ramaniak}@yahoo.com

[2]CCSIT, King Faisal University, Hofuf, Saudi Arabia,
Email: nazafar@kfu.edu.sa

ABSTRACT

Phenomenal growth both in wireless ad hoc networks and network-based real time applications has led to rapid change in the industry. For this purpose various protocols are designed for routing packets from source to destination to attain a better Quality of Service (QoS), which requires a mechanism that guarantees bounded delay and jitter. Finding the best route to destination satisfying the QoS for real-time interactive application has become a challenge for today's network services. The routing table of Ad hoc on-demand routing protocol (AODV) maintains only one route to the specified node without considering the other important parameters. This paper proposes a Formal framework to improve the QoS by searching the graph using formal techniques. The searching technique is based on considering packet type and battery life. Because the nodes of AODV protocol are not fixed hence the dynamic graphs are used to model the network topology. The Z notation is used to transform the graph into formal specification of the protocol. Finally, the formal specification is analyzed and validated using Z Eves tool.

KEYWORDS

AODV, Formal methods, Z notation, Graph theory, Route request procedure, QoS

1. INTRODUCTION

Ad hoc network is a collection of wireless nodes, which form a temporary network without relying on the existing network infrastructure or centralized administration [1]. A Mobile Ad Hoc Network (MANET) consists of a set of mobile hosts capable of communicating with each other without the assistance of base stations [2]. In MANETs mobile nodes are dynamic and self organize-able into random changing network, it helps people and devices to effortlessly work where there is no preexisting communication infrastructure such as, disaster recovery environments. People and vehicles can thus be internetworked in areas without a preexisting communication infrastructure. MANET objects must be able to communicate while moving and work together to provide routing services.

There are no dedicated routers in ad hoc networks, each individual node acts as a router and transmits packets from source to destination. If the source node does not have the destination node within the transmission range the intermediate nodes forward the packet to the destination. Routing protocol for Ad hoc wireless networks should have the special characteristics [3]. It must be fully distributed, adaptive to frequent changes in topology, involving a minimum number of nodes for route computation and maintenance, having minimum time for connection set-up, be localized, loop-free and independent of stale routes. The number of packets collisions must be kept to minimum and transmissions reliable to reduce message loss. It must converge to an optimal route once the network becomes stable. The convergence must be quick and use source resources such as bandwidth, battery life, computing and memory power optimally [3].

Many of the MANET routing protocols proposed are classified based on the routing strategy they follow to discover route to the destination. These protocols based on various factors including the issues about the power consumption, low bandwidth, and high error rates, in decision of routing. [4]. Routing protocols are classified into 3 categories. These are proactive reactive and hybrid protocols. Proactive protocols also known as Table-Driven protocols, maintain routes to all nodes, including nodes to which no packets are sent. These protocols adjust well in regard to change in the topology. Reactive protocols or On Demand protocols consider the demand for data transmission and determine the routes between hosts only when needed and can reduce routing overhead. Hybrid protocols combine both proactive and reactive protocol properties to come up with a better routing protocol for well-organized packet routing [5, 6]. Limited resources in MANETs make a challenging problem that is represented in designing of an efficient and reliable routing strategy [5]. Routing protocols are to be designed in order to use the limited resources efficiently. Further it should be time being adaptable to the changing network environment such as: packet type, network size, battery life, traffic density. Thus, the routing protocol may need to offer different ways or methods to select the path for example on the available QoS metrics on the discovered paths. Quality of Service (QoS) [7] is a set of service requirements to be met by the network while transporting a packet from the source to destination. It must use resources such as bandwidth, packet type and battery life optimally and effectively.

The issue of transferring real-time traffic over MANETs is a big challenge because as multimedia communications are designed to support both real and non-real time services. Real time applications, live voice and video conference are delay-sensitive as they have to be transmitted using minimum delay or a shortest routing path. QoS routing is an important building block for quality support in MANETs. Researchers have proposed many protocols and enhanced techniques for existing routing protocols to increase the QoS for MANETs [8], Most of the protocols are concerned with bandwidth reservations and are network-specific. Few of these focus on selecting the path depending on the QoS metrics, for example, delay, bandwidth requirement and battery power life.

Most of the proposed protocols are focused on simulation. Few implementations are proposed in which environments had no more than a dozen of nodes. Graph theory has much of its applications in the area of parallel and distributed algorithms and is an effective tool for modeling and visualizing the communication networks. Graph theory does not have much computer tool support for verifying and validating the systems. Formal techniques are best approaches for specification and proving the computerized models. In this research, formal methods in terms of Z notation [9] are used by linking with graph theory for describing the AODV [10] route request procedure. For routing procedure packet type and battery life of the node in selecting the nodes from source to destination are considered. Z notation is used because of abstraction and encapsulation of objects for further enhancement of the description of the system.

Rest of the paper is organized as follows: Section 2 provides an outline of the related work. Section 3 presents an introduction to formal methods. In section 4, formal specification of AODV routing protocol is given using Z notation and graph theory. Finally, conclusion and future work are discussed in section 5.

2. RELATED WORK

Several QoS models has been proposed for MANETs such as INSIGNIA [11], ASAP [12], SWAN [13, 14, 15], E-SWAN [16], FQMM [17] [18] and LWQ [19]. The major need of a QoS model is to define the methodology by which services e.g. per-flow or per-class could be provided in the network [20] along with differentiation [7] by considering multimedia flows of voice or video giving higher priority over best effort flows. In the bandwidth-constrained MANET environment shared resources must be allocated across traffic flows. Differing traffic flows have differentiated resource requirements and differentiated costs for not receiving desired resources.

AODV is a popular routing protocol in which when a source node wants to send a message to the destination node but it does not have a valid route yet, a path discovery process will be initiated to locate the destination. The source node broadcasts a route request (RREQ) packet to its neighbors which then forward the RREQ to their neighbors in turn. The forwarding process continues until either the destination or an intermediate node that has a route to the destination is located. When the RREQ reaches the destination, the destination node responds by unicasting RREP (route reply) back along the reverse path. As the RREP routes back to the source node, the route from the source node to the destination is established [21]. A first step in providing optimal resource allocation across all traffic flows is then to differentiate traffic flows according to requirements and priority. However, the AODV protocol neither provides the classification of RREQ based on the service differences nor provides a better QoS.

3. FORMAL METHODS

Formal methods are approaches which can be used for writing formal description, analysis and producing refinements [22]. A formal specification is a description that is abstract, precise and in a sense is complete. The abstraction allows a human reader to understand the big picture; the precision forces ambiguities to be questioned and removed; and the completeness means all aspects of behavior are described. Secondly, the formality of the description allows us to carry out rigorous analysis. By looking at a single description one can determine useful properties such as consistency or deadlock-freedom [23]. By writing different descriptions from different viewpoints one can determine important properties such as satisfaction of high level requirements or correctness of a proposed design [23].

Z notation is a model-based notation which is a strongly typed, mathematical specification language, not an executable notation and it cannot be interpreted or compiled into a running program. There are some tools for checking Z texts for syntax and type errors in much the same way that a compiler checks code in an executable programming language. In Z notation, schemas are used which are small pieces for decomposing a specification into manageable components. The schema is the feature that distinguishes Z from other formal notations. In Z, schemas are used to describe both static and dynamic aspects of a system. Z specification enables to produce a model that is unambiguous, verifiable and traceable. Z is more mature and has an ISO standard [24].

In Ad hoc networks nodes are free to move causing changes in the network topology and highly dynamic. This dynamic nature increases the complexity of the algorithms designed for Ad hoc networks and the verification of AODV algorithms is a difficult error-prone task that requires much effort. Formal methods have a lot to offer, AODV can be modeled from a complex system to mathematical entities resulting a rigorous model by using these techniques. It is possible to model and verify the AODV properties in a more thorough and detailed fashion than the empirical testing and simulation techniques.

4. FORMAL DESCRIPTION USING Z

Formal description of the route request procedure for Ad-hoc on-demand distance routing protocol using Z notation is presented in this section. Initially formal definitions of basic data types will be described then complex structures needed for Ad-hoc network will be defined. Finally, the routing request procedure for the network will be given.

4.1 Definitions

A communication network is a collection of objects interconnected by channels that help and allow the users to share the information and resources for communication. A mobile Ad-hoc network is a collection of self-configuring objects inter-connected by wireless devices which are free to move in any direction in the domain. These networks may or may not be part of another larger network. In this research, the communication network is defined as a graph in which the moving objects are assumed as nodes and communication links are supposed to be edges of the network. The graph is not static but is dynamic, i.e., its any two nodes may be connected at one time while are disconnected at the other time. The identifier of a moving object is denoted by *Node* as given below. Four types of nodes are assumed, i.e., source, destination, internal and nil. The power of battery is assumed dead, low or normal denoted by *Dead, Low and Normal* respectively.

$[\mathrm{Node}];\ \mathit{Power} ::= \mathit{Dead} \mid \mathit{Low} \mid \mathit{Normal}$

$\mathit{Type} ::= \mathit{Source} \mid \mathit{Destination} \mid \mathit{Internal} \mid \mathit{Nil}$

The above types are supposed as sets in Z. In modeling using sets in Z notation, we do not impose any restriction upon number of elements in a set and a high order of abstraction is supposed. Further, we do not insist upon any effective procedure for deciding whether an arbitrary element in a set is its member. As a consequent, the Node is a set over which we cannot define the operation of cardinality to know the number of elements. Similarly, the subset and complement operators are not well-defined in a set defined in Z notation. The set Power and Type defined above are assumed as free types in which at one time only one value of it is assumed.

4.2 Moving Objects and Possible Operators

The moving object of the network has four components, i.e., identification, type, battery and set of all the neighbors of it. The object is defined as a schema given below. The type of object is considered because it might be source, destination or an internal node. Further, the node is also given a *Nil* value if it is not part of any route stored in the routing table.

Object

$id: Node$
$type: Type$
$battery: Power$
$neighbours: \mathbb{F}\ Node$

The four possible operations to change the state of an object are defined in this section. The first one is used to change the type of an object and is defined as a schema denoted by *ChangeObjectType*. It takes two inputs, i.e., *ΔObject* and *type?* as given in the first part of the schema. The delta (Δ) symbol is used to show that the state of the object is changed. The symbol question mark (?) is used to represent that type is an input for this operation. In the second part of the schema, the old type is replaced by the new type *type?* of the object. And all other components are unchanged.

ChangeObjectType

$\Delta Object$
$type?: Type$

$type' = type?$
$id' = id$
$battery' = battery$
$neighbours' = neighbours$

The next operation is used to change state of the battery and is defined as a schema denoted by *UpdateObjectBattery*. It takes *ΔObject* and *battery?* as input. In the second part of the schema, the new status of battery is replaced with the old one. All other components of the schema are unchanged.

UpdateObjectBattery

$\Delta Object$
$battery?: Power$

$battery' = battery?$
$id' = id$
$type' = type$
$neighbours' = neighbours$

AddObjectNeighbour

$\Delta Object$
$node?: Node$

$neighbours' = neighbours \cup \{node?\}$
$id' = id$
$type' = type$
$battery' = battery$

```
┌─RemoveObjectNeighbour────────────────────
│ ΔObject
│ node?: Node
├──────────
│ neighbours' = neighbours \ {node?}
│ id' = id
│ type' = type
│ battery' = battery
└──────────────────────────────────────────
```

In route finding, neighboring nodes play an important role for a source node to be connected with destination. In definition of the object, a variable *neighbours* was used storing information about the neighbors of a node. The *AddObjectNeighbour* operation is used to add a node in the set of neighbors if a new node is connected to the given node using the union operator. On the other hand a node can be disconnected from the node as well. The schema *RemoveObjectNeighbour* is defined above to remove a node using the set difference operator.

4.3 Connectivity of Objects

The communication between two objects is defined by a link which is described by the schema *Connectivity* as given below. It has three components, i.e., *connection, status* and *weight.* The first variable is used to define the link (edge) based on two nodes of the graph. The second is used to represent its status that is the nodes are connected or disconnected represented by *Active* and *NotActive*. And the last one variable *weight* is used to represent the time needed a node to communicate with the other. Because an object cannot communicate to itself therefore it is verified in the second part of the schema that first element of the connection cannot be same as the second element of it.

Status ::= *Active* | *NotActive*

```
┌─Connectivity─────────────────────────────
│ connection: Object × Object
│ status: Status
│ weight: ℕ
├──────────
│ connection . 1 . id ≠ connection . 2 . id
└──────────────────────────────────────────
```

4.4 Mobile Ad hoc Network

Any object can communicate with another in communication network and hence is represented by a complete graph. On the other hand, a link is either active or dead which was considered in the definition of connectivity. The formal specification of the network is described by the schema *Network* which consists of two components *objects* and *connections*. The first one is a collection of objects (nodes) of the graph which is defined as a type of finite power set of *object*. And the second one is represented by *connections*, which is also a finite power set of *Connectivity* called the edges. In the predicate part of the schema, it is proved that for any two objects there must be an edge because any two nodes can communicate if the link is active. Similarly, for any edge there must be two nodes in the network which is a natural constraint.

```
┌─Network──────────────────────────────────────────────
│ objects: 𝔽 Object
│ connections: 𝔽 Connectivity
├──────────────
│ ∀o1, o2: Object | o1 ∈ objects ∧ o2 ∈ objects
│   • ∃con: Connectivity | con ∈ connections • con . connection = (o1, o2)
│ ∀con: Connectivity | con ∈ connections
│   • ∃o1, o2: Object | o1 ∈ objects ∧ o2 ∈ objects
│       • con . connection = (o1, o2)
│
└──────────────────────────────────────────────────────
```

In mobile Ad hoc network, if a node is connected with another node at one time it might be disconnected at the other time. It means the communication is possible only if the nodes are connected. In our model, we have supposed that communication is possible if the link between nodes is active. The formal specification the mobile Ad hoc network is described below based on the definition of network.

```
┌─AdhocNetwork─────────────────────────────────────────
│ adhoc: Network
├──────────────
│ ∀con: Connectivity | con ∈ adhoc . connections
│   • ∃o1, o2: Object | o1 ∈ adhoc . objects ∧ o2 ∈ adhoc . objects
│       • con . connection = (o1, o2)
│ ∀o1, o2: Object | o1 ∈ adhoc . objects ∧ o2 ∈ adhoc . objects
│   • ∃con: Connectivity | con ∈ adhoc . connections
│       • con . connection = (o1, o2) ⇒ con . status = Active
│ ∀o1, o2: Object | o1 ∈ adhoc . objects ∧ o2 ∈ adhoc . objects
│   • ∃con: Connectivity | con ∈ adhoc . connections
│       • con . connection = (o1, o2) ⇒ o1 . id ≠ o2 . id
│ ∀o1, o2: Object | o1 ∈ adhoc . objects ∧ o2 ∈ adhoc . objects
│   • ∃con1: Connectivity | con1 ∈ adhoc . connections
│       • con1 . connection = (o1, o2)
│         ⇒ (∃con2: Connectivity | con2 ∈ adhoc . connections
│             • con2 . connection = (o2, o1))
│
└──────────────────────────────────────────────────────
```

Invariants: (i) For any two objects in the graph there must be an edge connecting it. This edge may be active or passive (ii) For any link, there must be two objects of the network which can communication to each other. (iii) The identifiers of the any two objects in the network must be different. (iv) We have supposed that if an object A can communicate to object B then vice versa is possible that is the network is a symmetric relation.

4.5 Route Request Procedure

In this section, the data in terms of packet is defined before describing the route request procedure. The data is a set of packets, P, where each packet contains the information about the data itself, identifier, type of data and time allocated to the data packet for the successful delivery.

Three types of data, text, audio and video, are supposed. For each type of data different timing are supposed which are denoted by *tt, ta and tv*.

$[P]; PType ::= Text \mid Audio \mid Video$

```
Packet
  packet: P
  id: ℕ
  type: PType
  tt, ta, tv: ℕ
```

```
DataPacket
  data: 𝔽 P
```

In AODV routing protocol, the history of routes is stored in the routing table. A route is searched only if it does not exist in the routing table. If the source has already a route the data is transmitted, otherwise, the source node sends request to its neighbors. The routing table is defined by the schema *Routings* given below which consists of two variables, i.e., *AdhocNetwork* and *routes.* The variable *routes* is a collection of routes whereas each route is a sequence of nodes in the network. Two properties are defined in the predicate part of the schema. In the first property, it is stated that a route must be a path in the network whose all nodes are objects. In the second property, the connectivity of nodes in the route is checked and verified.

```
Routings
  AdhocNetwork
  routes: 𝔽 (seq Node)
  ----------------
  $\forall route: \text{seq } Node \mid route \in routes$
    $\bullet\ \text{ran } route \subseteq \{ o: Object \mid o \in adhoc . objects \bullet o . id \}$
  $\forall route: \text{seq } Node \mid route \in routes \wedge \# route > 1$
    $\bullet\ \forall i: \mathbb{N} \mid i \in 1 .. \# route - 1$
      $\bullet\ \exists con: Connectivity \mid con \in adhoc . connections$
        $\bullet\ (route\ i, route\ (i + 1)) = (con . connection . 1 . id, con . connection . 2 . id)$
```

In AODV routing protocol, a route is established only on request of a source node for transmitting the data packets. If the source has already a route the data is transmitted, otherwise, the source node sends request to its neighbors. If any neighbor has a route to the destination, it is replied to the sender node otherwise it sends the request to the neighbors excluding the sender. This process is continued until the destination is found. The source node sends a request to the neighbors only once and it will resend if it does not get any response within the specified time. The route request procedure is divided into three procedures. The first one *RREQStoDest* is used to verify the route from source to the destination. It consists of seven components which are described above. The timestamp is used to record the time stamp of the object. The formal description of the procedure is given below.

RREQStoDest

$Routings$
$AdhocNetwork$
$DataPacket$
$timestamp?: \mathbb{N}$
$path: \text{seq}\ Node$
$source?, destination?: Object$

$destination?\ .\ battery \neq Dead$
$source?\ .\ type = Source$
$destination?\ .\ type = Destination$
$\exists o: Object \mid o \in adhoc\ .\ objects \bullet source?\ .\ id = o\ .\ id$
$\exists o: Object \mid o \in adhoc\ .\ objects \bullet destination?\ .\ id = o\ .\ id$
$source?\ .\ id \neq destination?\ .\ id$
$\forall o: Object \mid o\ .\ id \in source?\ .\ neighbours$
$\quad \bullet \exists con: Connectivity \mid con \in adhoc\ .\ connections$
$\qquad \bullet (source?, o) = con\ .\ connection$
$\forall p{:}Packet \mid p.packet \in data \bullet p\ .\ type = Text \Rightarrow timestamp? = p\ .\ tt$
$\forall p{:}Packet \mid p.packet \in data \bullet p.type = Audio \Rightarrow timestamp? = p\ .\ ta$
$\forall p{:}Packet \mid p.packet \in data \bullet p.\ type = Video \Rightarrow timestamp? = p\ .\ tv$
$\forall route: \text{seq}\ Node \mid route \in routes \wedge \#\ route \geqslant 1$
$\quad \bullet route\ 1 \neq source?\ .\ id \vee route\ (\#\ route) = destination?\ .\ id$
$\quad \Rightarrow path = \langle source?\ .\ id, destination?\ .\ id \rangle$
$\exists route: \text{seq}\ Node \mid route \in routes \wedge \#\ route \geqslant 1$
$\quad \bullet route\ 1 = source?\ .\ id \wedge route\ (\#\ route) = destination?\ .\ id$
$\quad \Rightarrow path = route$

Invariants: (i) The battery of destination node is not dead. (ii) The node sending request is a source type. (iii) The node receiving request is a destination type. (iv) The source node must be in the set of nodes of the network. (v) The destination node must be in the collection of nodes of the network. (vi) The source and destination nodes are distinct. (vii) The source node is connected to one of its neighbours. (viii) Different time stamps are given to different data types. (ix) The first and last elements in the route are source and destination respectively. (x) If there exists a route from source to destination in the table communication is possible.

RREQStoNeighs

$Routings$
$AdhocNetwork$
$DataPacket$
$timestamp?: \mathbb{N}$
$path: \text{seq}\ Node$
$source?, destination?, candidate!: Object$

$candidate!\ .\ battery = Normal$
$source?\ .\ type = Source$
$destination?\ .\ type = Destination$
$candidate!\ .\ type = Internal$
$\exists o: Object \mid o \in adhoc\ .\ objects \bullet source?\ .\ id = o\ .\ id$

$\exists o: Object \mid o \in adhoc . objects \bullet destination? . id = o . id$
$\exists o: Object \mid o \in adhoc . objects \wedge candidate! . battery \neq Dead$
$\quad \bullet candidate! . id = o . id$
$source? . id \neq destination? . id$
$destination? . id \notin source? . neighbours$
$candidate! . id \in source? . neighbours$
$\forall o: Object \mid o . id \in source? . neighbours$
$\quad \bullet \exists con: Connectivity \mid con \in adhoc . connections$
$\qquad \bullet (source?, o) = con . connection$
$\forall o: Object \mid o . id \in source? . neighbours \wedge o \neq candidate!$
$\quad \bullet \exists con1, con2: Connectivity$
$\qquad \mid con1 \in adhoc . connections \wedge con2 \in adhoc . connections$
$\qquad \bullet (source?, candidate!) = con1 . connection$
$\qquad \wedge (source?, o) = con2 . connection$
$\qquad \Rightarrow con1 . weight \leqslant con2 . weight$
$\forall p:Packet \mid p .packet \in data \bullet p . type = Text \Rightarrow timestamp? = p . tt$
$\forall p:Packet \mid p .packet \in data \bullet p.type = Audio \Rightarrow timestamp? = p . ta$
$\forall p:Packet \mid p.packet \in data \bullet p . type = Video \Rightarrow timestamp? = p . tv$
$path = \langle source? . id, candidate! . id \rangle$
$\forall route: \mathrm{seq}\ Node \mid route \in routes \wedge \# route \geqslant 1$
$\quad \bullet route\ 1 \neq source? . id \vee route\ (\# route) = destination? . id$
$\quad \Rightarrow path = \langle source? . id, candidate! . id \rangle$
$\exists route: \mathrm{seq}\ Node \mid route \in routes \wedge \# route \geqslant 1$
$\quad \bullet route\ 1 = source? . id \wedge route\ (\# route) = destination? . id$
$\quad \Rightarrow path = route$

If the source has no route to the destination then it sends the request to its neighbors. If any neighbor has a route to the destination, it is replied to the sender node otherwise it sends the request to the neighbors excluding the sender. This process is continued until the destination is found. The procedure of sending a request from source to neighbor is defined by the schema *RREQStoNeighs* given below. It consists of same number of components as defined above in addition to a candidate node. The candidate node is used to choose one of the several nodes as a part of the shortest path. The formal description of the procedure is given above.

RREQINtoNeighs

$Routings$
$AdhocNetwork$
$DataPacket$
$timestamp?: \mathbb{N}$
$pathold, pathnew: \mathrm{seq}\ Node$
$source?, destination?, current?, candidate!: Object$

$candidate! . battery = Normal$
$source? . type = Source$
$destination? . type = Destination$
$current? . type = Internal$
$candidate! . type = Internal$
$\exists o: Object \mid o \in adhoc . objects \bullet source? . id = o . id$
$\exists o: Object \mid o \in adhoc . objects \bullet destination? . id = o . id$

```
$\exists o: Object \mid o \in adhoc . objects \bullet current? . id = o . id$
$\exists o: Object \mid o \in adhoc . objects \bullet candidate! . id = o . id$
$source? . id \neq destination? . id$
$current? . id \neq destination? . id$
$candidate! . id \in current? . neighbours$
$\forall o: Object \mid o . id \in current? . neighbours$
  $\bullet \exists con: Connectivity \mid con \in adhoc . connections$
    $\bullet (current?, o) = con . connection$
$\forall o: Object \mid o . id \in current? . neighbours \wedge o \neq candidate!$
  $\bullet \exists con1, con2: Connectivity$
    $\mid con1 \in adhoc . connections \wedge con2 \in adhoc . connections$
    $\bullet (current?, candidate!) = con1 . connection$
      $\wedge (current?, o) = con2 . connection \Rightarrow con1 . weight \leqslant con2 . weight$
$\forall p:Packet \mid p.packet \in data \bullet p . type = Text \Rightarrow timestamp? = p . tt$
$\forall p:Packet \mid p.packet \in data \bullet p . type = Audio \Rightarrow timestamp? = p . ta$
$\forall p:Packet \mid p.packet \in data \bullet p.type = Video \Rightarrow timestamp? = p . tv$
$candidate! . type = Internal$
$candidate! . id \in Node$
$pathnew = pathold ^\frown \langle candidate! . id \rangle$
$\forall route: \mathrm{seq}\ Node \mid route \in routes \wedge \# route \geqslant 1$
  $\bullet route\ 1 \neq source? . id \vee route (\# route) = destination? . id$
    $\Rightarrow pathnew = pathold ^\frown \langle candidate! . id \rangle$
$\exists route: \mathrm{seq}\ Node \mid route \in routes \wedge \# route \geqslant 1$
  $\bullet route\ 1 = source? . id \wedge route (\# route) = destination? . id$
    $\Rightarrow pathnew = pathold ^\frown route$
```

After the source sends the route request to its neighbors the neighbor sends the same request to its neighbours if it has no route in the routing table. This process is continued until a route is found. The procedure of sending a request from internal node to its neighbors is defined by the schema *RREQINtoNeighs* given below. It consists of same number of components as defined above in addition to a current node and the new path to be updated. The current node is used to represent the current position. The formal description of the procedure is given above.

After the internal nodes send the route request to its neighbors until a destination is found. Finally, we need a different procedure to send the request from last one internal node to the destination. The procedure of sending a request from the internal node to destination is defined by the schema *RREQINtoDest* given below. It consists of same number of components as defined above except the candidate node. This is because the candidate node, destination node, is known to us. The formal description of the procedure is given below.

```
RREQINtoDest
  $Routings$
  $AdhocNetwork; DataPacket$
  $timestamp?: \mathbb{N}$
  $pathold, pathnew: \mathrm{seq}\ Node$
  $source?, destination?, current?: Object$
  ----------------
  $destination? . battery \neq Dead$
```

$source?\,.\,type = Source$

$destination?\,.\,type = Destination$

$current?\,.\,type = Internal$

$\exists o: Object \mid o \in adhoc\,.\,objects \bullet source?\,.\,id = o\,.\,id$

$\exists o: Object \mid o \in adhoc\,.\,objects \bullet destination?\,.\,id = o\,.\,id$

$\exists o: Object \mid o \in adhoc\,.\,objects \bullet current?\,.\,id = o\,.\,id$

$source?\,.\,id \neq destination?\,.\,id$

$current?\,.\,id \neq destination?\,.\,id$

$current?\,.\,id \neq source?\,.\,id$

$\forall o: Object \mid o\,.\,id \in current?\,.\,neighbours$

$\quad \bullet \exists con: Connectivity \mid con \in adhoc\,.\,connections$

$\quad\quad \bullet (current?, o) = con\,.\,connection$

$\forall p: Packet \mid p\,.\,packet \in data \bullet p\,.\,type = Text \Rightarrow \mid timestamp? = p\,.\,tt$

$\forall p: Packet \mid p\,.\,packet \in data \bullet p\,.\,type = Audio \Rightarrow \mid timestamp? = p\,.\,ta$

$\forall p: Packet \mid p\,.\,packet \in data \bullet p\,.\,type = Video \Rightarrow \mid timestamp? = p\,.\,tv$

$pathnew = pathold ^\frown \langle destination?\,.\,id\rangle$

$\forall route: \mathrm{seq}\ Node \mid route \in routes \wedge \#\, route \geqslant 1$

$\quad \bullet route\ 1 \neq source?\,.\,id \vee route\ (\#\, route) = destination?\,.\,id$

$\quad\ \Rightarrow pathnew = pathold ^\frown \langle destination?\,.\,id\rangle$

$\exists route: \mathrm{seq}\ Node \mid route \in routes \wedge \#\, route \geqslant 1$

$\quad \bullet route\ 1 = source?\,.\,id \wedge route\ (\#\, route) = destination?\,.\,id$

$\quad\ \Rightarrow pathnew = pathold ^\frown route$

5. CONCLUSION

In this paper, a Formal framework to improve the QoS by searching the graph using formal techniques is employed, investigated and analyzed for mobile ad hoc network. The Z notation is used as a formal technique because of its abstract characteristics and properties, and having a rigorous computer tool support. In the proposed approach, a formal procedure is specified how to send a request from a source node to the destination in the AODV routing protocol by considering the QoS parameters packet type and battery life. It is further investigated how formal methods can be applied to the route discovery process in the AODV routing protocol where QoS is not sacrificed. It was observed that inconsistencies and ambiguities were removed by the application of formal methods for the specification of the route request procedure. We believe that this integrated approach of graph theory and Z notation will be very effective tool for optimizing the route request and reply procedures in our future work.

REFERENCES

[1] S. A. Al-Omari and P. Sumari, “An overview of Mobile Ad Hoc Networks For the Existing Protocols and Applications”, The International Journal on Applications of Graph Theory in Wireless Ad hoc Networks and Sensor Networks, 1(1), 2010.

[2] Bindhu.R, “Mobile Agent Based Routing Protocol with Security for MANET”, Int’l Journal of Applied Engineering, 1(1), 2010.

[3] C. Siva Ram Murthy and B. S. Manoj, “Adhoc Wireless Networks Architecture and Protocols”, Prentice Hall, 2004.

[4] S. Ahmed and A. K. Ramani, “Exploring the Requirements for QoS in Mobile Ad hoc Networks” Journal of Information & Communication Technology, Vol. 1 No. 2, 2007.

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

[6] Buruhanudeen, S., Othman, M., Othman, M., Mohd Ali, B "Existing MANET Routing Protocols and Metrics used Towards the Efficiency and Reliability-An Overview", IEEE-Malaysia International Conference on Communications, pp. 231-36, 2007.

[7] T. B. Reddy, I. Karthigeyan, B. Manoj, and C. S. R. Murthy, "Quality of service provisioning in ad hoc wireless networks: a survey of issues and solutions", http://www.sciencedirect.com.

[8] L. Chen andW. B. Heinzelman, "A survey of routing protocols that support QoS in mobile ad hoc networks," IEEE Network, vol. 21, no. 6, pp. 30–38, 2007.

[9] J. M. Spivey, "The Z Notation: A Reference Manual," Prentice Hall, 1989.

[10] Perkins, et. al. "Ad hoc On-Demand Distance Vector (AODV) Routing", RFC 3561, 2003.

[11] S. B. Lee, G. S. Ahn, X. Zhang, and A. T. Campbell. Insignia: an ip-based quality of service framework for mobile ad hoc networks. Journal of Parallel & Distributed Computing, 60(4):374.406, 2000.

[12] P. Stuedi, J. Xue and G. Alonso. Asap: an adaptive qos protocol for mobile ad hoc networks, PIMRC, 3(7):2616. 2620, 2003.

[13] G. S. Ahn, A.T. Campbell, A. Veres, and L. -H. Sun. Supporting service differentiation for real-time and best-effort traffic in stateless wireless ad hoc networks (swan). IEEE Transactions on Mobile Computing, 1(3):192.207, 2002.

[14] G. S. Ahn, A. T. Campbell, A. Veres and L. -H. Sun, SWAN: Service differentiation in stateless wireless ad hoc networks, Proc. IEEE INFOCOM'2002, New York, New York, June 2002.

[15] Veres, A. T. Campbell, M. Barry, and L. -H. Sun, "Supporting service differentiation in wireless packet networks using distributed control", IEEE Journal of Selected Areas in Communications, Special Issue on Mobility and Resource Management in Next-Generation Wireless Systems, 19(10), pp. 2094-2104, October 2001.

[16] Y. L. Morgan and T. Kunz. "Enhancing swan QoS model by adopting destination-based regulation (E-Swan)". WiOpt.04: Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks, 2004.

[17] Hannan, C. K. Chaing and S. K. G. Winston, "Quality of service models for ad hoc wireless networks", The handbook of ad hoc wireless networks, 2003, CRC Press, Inc. pp. 467- 482.

[18] X. Hannan, C. K. Chaing, W. Seah, A. Lo, "On service prioritization in mobile ad-hoc networks", ICC 2001, no. 1, June 2001, pp. 1900-1904.

[19] H. Arora and L. Greenwald. "Toward the use of local monitoring and network-wide correction to achieve QoS guarantees in mobile ad hoc networks", IEEE SECON 2004, 4-7 Oct. 2004, pp. 128 . 138.

[20] D. Perkins and H. D. Hughes. A survey on quality-of service support for mobile ad hoc networks. Wireless Communications and Mobile Computing, 2(5):503.513, 2002.

[21] S. Ahmed and A. K. Ramani, N.A.Zafar "Formal Verification of Route Request Procedure for AODV Routing Protocol" International Journal of Advanced research in computer science. Vol 2(1) 2011.

[22] S. Black, Paul P. Boca, Jonathan P. Bowen, J. Gorman and M. Hinchey, "Formal Versus Agile: Survival of the Fittest", Computer, Vol. 42(9), pp. 37–45, 2009.

[23] S. Chiyangwa and M. Kwiatkowska, "Modeling Ad hoc On-demand Distance Vector (AODV) Protocol with Time Automata", Proceedings of Third Workshop on Automated Verification of Critical Systems, 2003.

[24] ISO: Information technology – "Z formal specification notation, syntax, type system and semantics" (2002) ISO/IEC 13568:2002, International Standard