ISSN No. 0976-5697

Volume 2, No. 1, Jan-Feb 2011

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Formal Verification of Route Request Procedure for AODV Routing Protocol

Shakeel Ahmed* and A. K. Ramani
SCSIT, Devi Ahaliya University
Indore, India
shakeel_rahi,ramaniak@yahoo.com

Nazir Ahmad Zafar
CCSIT, King Faisal University
Hofuf, Saudi Arabia
nazafar@kfu.edu.sa

Abstract: Many protocols have been designed for routing the packets from a source to destination. In Ad hoc on-demand routing protocol (AODV) the routing table maintains only one route to the specified node. The route is rediscovered by the source node when the earlier route fails. This paper aims to study the characteristics of Ad hoc networks and employ formal methods to model, investigate and analyze the routing protocol. The Z notation is used as a formal technique because of its abstract properties. In the proposed approach, it is specified how a source node can request for a route to the destination in AODV routing protocol. It is investigated how formal methods can be applied to the route discovery process in the AODV routing protocol. Finally, the formal specification is analyzed and validated using Z Eves tool.

Keywords: AODV, Formal methods, Z notation, Route request procedure, verification.

I. 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]. Mobile Ad hoc networks (MANETs) offer communication over a shared wireless channel and are extensively employed without any pre-existing infrastructure. Ad hoc networks form a multi hop network, where the communication is over the wireless channels, hopping over several mobile nodes. Efforts have been taken for achieving efficient and reliable routing procedures in mobile Ad hoc networks.

Ad hoc network is used in areas of sensor networks for environmental monitoring, rescue operations in remote areas, construction sites and personal area networking, emergency operations, military and civilian environments [2]. The scopes of the Ad hoc network are also associated with dynamic topology changes, bandwidth-constrained, energy constrained operation, limited physical security, mobility-induced packet losses, wireless transmission range, broadcast nature of the wireless medium, hidden terminal problem and packet losses due to transmission errors [2].

Due to the issues in an Ad hoc wireless network environment mentioned above, wired network routing protocols cannot be used in Ad hoc wireless networks. Hence, Ad hoc wireless networks require specialized routing protocols that address the challenges described above. A routing protocol for Ad hoc wireless networks should have the special characteristics [3]. It must be fully distributed, it must be adaptive to frequent topology changes, route computation and maintenance must involve a minimum number of nodes, minimum connection set up time, it must be localized, and it must be loop-free and independent of stale routes. The number of packets collisions must be kept to a minimum, the transmissions should be reliable to reduce message loss. It must converge to an optimal route once the network topology becomes stable. The convergence must be quick. It must optimally use source resources such as bandwidth, computing and memory power, and battery life [3].

However, most of the protocols are focused on simulation. Few implementations are proposed in which environments had no more than a dozen of nodes. The reasons are the difficulties to implement such routing protocols. The cost and material requirements to develop and insure that all functionalities presented in the IETF standards have been implemented [4].

Earlier works have been done on the verification of wired and wireless routing protocols. Bhargavan [5] presents an automated proof of AODV loop freedom using the SPIN model checker [6]. As model checkers can only deal with a network with a limited number of nodes. A Validation Model for the Dynamic Source Routing (DSR) Protocol [7] is proposed. In this research, the network diameter is collection of 35 hops. Moreover, the version of AODV is Internet-Draft. However, these approaches only handle very small number of nodes typically ranging from two to three nodes, and are extremely simple and fixed network topologies.

Graph theory is an effective tool for modeling and visualizing the communication networks because of its applications in the area of parallel and distributed algorithms. On the other hand, 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 [8] are used by linking with graph theory for describing the AODV route request procedure for mobile Ad hoc networks. The Z notation is used because of its abstraction and encapsulation of objects for further enhancement of the systems.

Rest of the paper is organized as follows: Section 2 provides an outline of the MANET and AODV routing protocol. Section 3 presents an introduction to formal methods. In section 4, formal specification of AODV routing protocol is given. Finally, conclusion and future work is presented in section 5.

II. MANET AND AODV

MANET is an efficient way of exchanging peer-to-peer information among devices such as fixed, portable and wireless nodes [9]. The network must be able to adaptively alter the routing paths to ease any of these effects. The important feature in MANET is to support robust and efficient operation by incorporating routing functionality [10]. AODV is a reactive routing protocol for the mobile ad hoc networks.

AODV is a purely reactive routing protocol; it offers low network utilization and uses destination sequence number to ensure loop freedom [11]. In this protocol, a terminal does not need to keep a view of the whole network or a route to every other terminal. It also does not need to exchange route information periodically with the neighboring terminals. When a mobile terminal has packets to send to a destination it need to discover and maintain a route to that destination. In AODV, each node contains a route table for a destination. A route table stores the following information: destination address and its sequence number, active neighbors for the route, hop count to the destination, and expiration time for the table. An important feature of AODV is that it uses a destination sequence number, which corresponds to a destination node that was requested by a routing sender node. If there are multiple routes from a request sender to a destination, the sender takes the route with a higher sequence number. The expiration time is updated each time the route is used. If this route has not been used for a specified period of time, it is discarded. According to the specification of AODV it includes an optimization technique to control the Route Request (RREQ) for flooding in the route discovery process. It uses an expanding ring search initially to discover routes to an unknown destination. In the search, increasingly larger neighborhoods are searched to find the destination. The search is controlled by the RREQ packet. If the route to a known destination is needed, the prior hop-wise distance is used to optimize the search [11].

III. FORMAL METHODS

Formal methods are for writing formal description, analyzing the description and producing the refinements [12]. 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 that all aspects of behavior, for example, errors cases are described and understood. 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 [13]. 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 [13].

In Ad hoc networks, nodes are free to move, change in topology is 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. Thus, formal methods has a lot to offer and by using these techniques AODV can be modeled from a complex systems to mathematical entities by building a mathematically-based rigorous model of a complex system. It is possible to model and verify the AODV properties in a more thorough and detailed fashion than the empirical testing and simulation techniques.

IV. FORMAL SPECIFICATION USING Z NOTATION

In this section, we give the formal description of route request procedure for Ad hoc on-demand distance routing protocol using Z notation. For this purpose, at first, formal definition of communication network will be described. Then it will be refined to formalize an Ad hoc network by putting the constraints over it. Finally, the route request procedure will formalized.

A. *Moving Objects of Ad hoc Network*

As we now that a communication network is a collection of objects interconnected by channels that help and allow the users to share the information and resources. A mobile Ad hoc network is a self-configuring network of objects inter-connected by wireless devices which are free to move in any direction. These networks may operate by themselves or may be connected to the other larger networks. The communication network is defined as a graph in which the moving objects are assumed as nodes and communication links are supposed as edges. An id of a free moving object is denoted by Node as given below.

[*Node*]

In modeling using sets in Z, we do not impose any restriction upon number of elements in a set. 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 set 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 as well. The moving object is defined as a schema which consists of four components, i.e., identification, type, time to enter and set of all the neighbors of the object. The type of object is considered because it might be source, destination or an internal node.

∪_*Object* ______________________
→*id: Node; type: Type*
→*timestamp:* N
→*neighbours:* Φ *Node*
∠_________________________

Type ::= *Source* | *Destination* | *Internal* | *Nil*

The four possible operations to change the state of an object are: (i) to change type of an object, (ii) to update the time to access an object, (iii) to remove the neighbor if it is disconnected from it and (iv) to add a neighbor if it is connected to it. The first operation named ChangeObjectType takes two inputs, i.e., ∆Object and type? in the first part of the schema. The delta ∆ is used to show that state of the object is changed. The symbol ? is used to represent that type is an input for this operation. In the second part of the schema, the old type is replaced with the new type type? of the object. And all other components are unchanged.

∪_*ChangeObjectType* ________________
→∆*Object*
→*type?: Type*
∩______________
→*type' = type?*
→*id' = id*
→*timestamp' = timestamp*
→*neighbours' = neighbours*
∠_________________________

The second operation named *UpdateObjectTime* takes two inputs, i.e., ∆*Object* and *timestamp?*. The old time is replaced with the new time *timestamp?*. And all other components remain same.

∪_*UpdateObjectTime* ________________
→∆*Object*

→*timestamp?:* N
∩_______________
→*timestamp' = timestamp?*
→*id' = id*
→*type' = type*
→*neighbours' = neighbours*
∠___________________________

∪_*AddObjectNeighbour* ________________
→*ΔObject*
→*node?: Node*
∩_______________
→*neighbours' = neighbours* Y {*node?*}
→*id' = id*
→*type' = type*
→*timestamp' = timestamp*
∠___________________________

∪_*RemoveObjectNeighbour*______________
→*ΔObject*
→*node?: Node*
∩_______________
→*neighbours' = neighbours* \ {*node?*}
→*id' = id*
→*type' = type*
→*timestamp' = timestamp*
∠___________________________

B. 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, weight and status. The variable weight represents the time needs to send the data from one node to the other. The variable status is used to show if two nodes are connected or dead. Because an object cannot communicate with itself therefore it is verified that first element of the connection cannot be same as the second element of the connection.

∪_*Connectivity* ____________________
→*connection: Object* ξ *Object*
→*weight:* N
→*status: Status*
∩_______________
→*connection* . 1 . *id* ⌊ *connection* . 2 . *id*
∠___________________________

C. Mobile Ad hoc Network

In communication network, any object can communicate with any other object and hence can be represented by a complete graph. The formal specification of the network is given in terms of the schema Network which consists of two components objects and connections. The objects is a collection of objects which is defined as a type of finite power set of Object and connections as the finite power set of edges called connections. 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:* Φ *Object*
→*connections:* Φ *Connectivity*
∩_______________
→A*o1, o2: Object* | *o1* ε *objects f o2* ε *objects*
→ ∞ E*con: Connectivity* |
→*con* ε *connections* ∞ *con. Connection* = (*o1*, *o2*)
→A*con: Connectivity* | *con* ε *connections*
→ ∞ E*o1, o2: Object* | *o1* ε *objects f o2* ε *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 given above.

∪_*AdhocNetwork* ___________________
→*adhoc: Network*
∩_______________
→A*con: Connectivity* | *con* ε *adhoc . connections*
→ ∞ E*o1, o2: Object* | *o1* ε *adhoc . objects f o2* ε *adhoc* .
→*objects* ∞ *con . connection* = (*o1*, *o2*)
→A*o1, o2: Object* | *o1* ε *adhoc . objects f o2* ε *adhoc* .
→*objects* ∞ E*con: Connectivity* | *con* ε *adhoc . connections*
→ ∞ *con . connection* = (*o1*, *o2*) ⇒ *con . status = Active*
→A*o1, o2: Object* | *o1* ε *adhoc . objects f o2* ε *adhoc* .
→*objects* ∞ E*con: Connectivity* | *con* ε *adhoc . connections*
→ ∞ *con . connection* = (*o1*, *o2*) ⇒ *o1 . id* ⌊ *o2 . id*
→A*o1, o2: Object* | *o1* ε *adhoc . objects f o2* ε *adhoc* .
→*objects* ∞E*con1: Connectivity* | *con1* ε *adhoc . connections*
→ ∞ *con1 . connection* = (*o1*, *o2*)
→ ⇒ (E*con2: Connectivity* | *con2* ε *adhoc . connections*
→ ∞ *con2 . connection* = (*o2*, *o1*))
∠___________________________

***Invariants**:* (i) For any communication there must be two objects in the network. (ii) For any two objects, the communication is possible only if the link is active. (iii) The ids of communicating objects must be different. (iv) If the communication is possible from object A to Object B then vice versa is possible.

D. Route Request Procedure

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 route request procedure is divided into to two procedures. In the first one, the source node sends request to the neighbors only once and it will resend if it does not get any response within the specified time. In the second procedure an intermediate node sends the request to its neighbors, which is a recursive call procedure. The first procedure is defined in terms of the schema RREQStoNeighs given below. It consists of six components, i.e., AdhocNetwork, source?, destination?, candidate!, timestamp? and path. The first variable is for network as defined above. The second and third nodes are used for source and destination. The variable candidate is used finding sequence number. The fifth is used to record the time stamp of the object. And last one is used for updating the path. The formal description of the procedure is given below. The variants are defined in the second part of the schema by relating input and output variables in addition to other constraints over the variables.

```
∪_RREQStoNeighs__________________
→AdhocNetwork
→source?: Object
→destination?: Object
→candidate!: Object
→timestamp?: N
→path: seq Node
∩______________
→Eo: Object |o ε adhoc . objects ∞ source? . id = o . id
→Eo: Object |o ε adhoc . objects ∞ destination? . id = o . id
→Eo: Object | o ε adhoc . objects ∞ candidate! . id = o . id
→source? . id⌊ destination? . id
→candidate! . id ε source? . neighbours
→Ao: Object | o . id ε source? . neighbours
→ ∞ Econ: Connectivity | con ε adhoc . connections
→ ∞ (source?, o) = con . connection
→Ao: Object |o . id ε source? . neighbours f o⌊ candidate!
→ ∞ Econ1, con2: Connectivity
→|con1 ε adhoc . connections f con2 ε adhoc . connections
→ ∞ (source?, candidate!) = con1 . connection
→f (source?, o) = con2 . connection
→        ⇒ con1 . weight⌡con2 . weight
→candidate!= destination?⇒candidate!. type = Destination
→candidate!⌊ destination? ⇒ candidate! . type = Internal
→candidate! . timestamp = timestamp?
→path = ©source? . id, candidate! . id⟩
∠________________________
```

Invariants: (i) The source node must be in the set of nodes of the entire network. (ii) The destination node must also be in the collection of nodes of the network. (iii) The candidate is also in the pre defined nodes (iv) The source and destination nodes are distinct. (v) The candidate node is an element of the neighbors of the source node. (vi) Every neighbor has an active edge with the source node. (vii) The candidate node has less weight and highest sequence number as compared to other neighbors of source node. (viii) If the candidate is destination node then its type is also destination. (ix) If the candidate is not a destination node then it is an internal node. (x) The candidate time is updated. (xi) Path is updated by inserting the candidate node.

The second procedure is defined in terms of the schema RREQINtoNeighs to send the route request from an internal node to its neighbors. It consists of eight components, i.e., AdhocNetwork, source?, destination?, current?, candidate!, timestamp?, pathold and patnew. The first three variables are same as defined as above. The fourth variable current? is used to represent the node which is sending request to its neighbors. Of course it is different from the source node. The variable candidate has same meanings as defined above. The pathold is updated by the variable pathnew. The formal definition of this procedure is given below. The variants are defined in terms of constraints over these variables and defining relationship between it. The input and output variables are related to achieve the required objectives.

```
∪_RREQINtoNeighs __________________
→AdhocNetwork
→source?: Object
→destination?: Object
→current?: Object
→candidate!: Object
→timestamp?: N
→pathold: seq Node
→pathnew: seq Node
∩______________
→Eo: Object |o ε adhoc . objects ∞ source? . id = o . id
→Eo: Object |o ε adhoc . objects ∞ destination? . id = o . id
→Eo: Object |o ε adhoc . objects ∞ current? . id = o . id
→Eo: Object |o ε adhoc . objects ∞ candidate! . id = o . id
→source? . id⌊ destination? . id
→current? . id⌊ destination? . id
→candidate! . id ε current? . neighbours
→Ao: Object | o . id ε current? . neighbours
→ ∞ Econ: Connectivity | con ε adhoc . connections
→ ∞ (current?, o) = con . connection
→Ao:Object |o . id ε current? . neighbours f o⌊ candidate!
→ ∞ Econ1, con2: Connectivity
→| con1 ε adhoc.connections f con2 ε adhoc . connections
→ ∞ (current?, candidate!) = con1 . connection
→ f (current?, o) = con2 . connection
→        ⇒ con1 . weight⌡con2 . weight
→candidate!= destination?⇒candidate!. type = Destination
→candidate!⌊ destination? ⇒ candidate! . type = Internal
→candidate! . timestamp = timestamp?
→candidate! . id ε Node
→pathnew = pathold ⊥ ©candidate! . id⟩
∠__________________________
```

Invariants: (i) The source node must be in the set of nodes of the entire network. (ii) The destination node must also be in the collection of nodes of the network. (iii) The current node is in the existing set of nodes of the network. (iv) The candidate is also in the pre defined nodes. (v) The source and destination nodes are distinct. (vi) The current and destination nodes are also distinct. (vii) The candidate node is an element of the neighbors of the current node. (viii) Every neighbor has an active edge with the current node. (ix) The candidate node has less weight and highest sequence number as compared to other neighbors of current node. (x) If the candidate is destination node then its type is also destination. (xi) If the candidate is not a destination node then it is an internal node. (xii) The candidate time is updated. (xiii) Path is updated by inserting the candidate node.

V. CONCLUSION

In this paper, a study of characteristics of Ad hoc networks is done and formal methods are employed to model, investigate and analyze the routing protocol of mobile ad hoc networks. 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. It is further investigated how formal methods can be applied to the route discovery process in the AODV routing protocol. It was observed that ambiguities and inconsistencies 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.

VI. 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, Vol.1 (1), 2010.

[2] A. Rahman, S. Islam and A. Talevski, "Performance Measurement of Various Routing Protocol in Ad-Hoc Network", IMECS, Vol. 1, pp. 321-323, 2009.

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

[4] D. Maltz, J. Broch, and D. Jonhson, "Experiences designing and building a multi-hop wireless ad hoc network testbed," Carnegie Mellon University, Tech. Rep., 1999.

[5] K. Bhargavan, D. Obradovic and C. A. Gunter, "Formal Verification of Standards for Distance Vector Routing Protocols", Journal of the ACM, Vol. 49(4), pp. 538–576, 2002.

[6] SPIN: http://spinroot.com/spin/whatispin.html

[7] A. Cavalli, C. Grepet, S. Maag and V. Tortajada, "A Validation Model for the DSR Protocol", Proceedings of the 24th International Conference on Distributed Computing Systems, pp.768-773, 2004.

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

[9] Buruhanudeen, S., Othman, M., Othman, M., Mohd Ali, B.(2007), "Existing MANET Routing Protocols and Metrics used Towards the Efficiency and Reliability- An Overview", Proceedingsof the 2007 IEEE International Conference on Telecommunications, ICT-MIXCC,pp. 231-236, 2007

[10] Andreas Tonnesen, .Implementing and extending the Optimized Link State Routing Protocol,. Master Thesis at UniK, 2004.

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

[12] 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.

[13] 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.