Formal Modelling and Verification of Requirements of Adaptive Routing Protocol for Mobile Ad-Hoc Network

Bata Krishna Tripathy †, Ashray Sudhir †, and Padmalochan Bera
Indian Institute of Technology Bhubaneswar, India
Email: {bt10, sa17, and plb}@iitbbs.ac.in

Mohammad Ashiqur Rahman
Tennessee Tech University, USA
Email: marahman@tntech.edu

***Abstract*—A group of mobile nodes with limited capabilities sparsed in different clusters forms the backbone of Mobile Ad-Hoc Networks (MANET). In such situations, the requirements (mobility, performance, security, trust and timing constraints) vary with change in context, time, and geographic location of deployment. This leads to various performance and security challenges which necessitates a trade-off between them on the application of routing protocols in a specific context. The focus of our research is towards developing an adaptive and secure routing protocol for Mobile Ad-Hoc Networks, which dynamically configures the routing functions using varying contextual features with secure and real-time processing of traffic. In this paper, we propose a formal framework for modelling and verification of requirement constraints to be used in designing adaptive routing protocols for MANET. We formally represent the network topology, behaviour and functionalities of the network in SMT-LIB language. In addition, our framework verifies various functional, security, and Quality-of-Service (QoS) constraints. The verification engine is built using the Yices SMT Solver. The efficacy of the proposed requirement models is demonstrated with experimental results.**

***Index Terms*—Mobile Ad-Hoc Network (MANET), Routing protocol, Formal modelling, SMT-LIB, Formal verification, Yices SMT Solver.**

1. Introduction

Mobile Ad-Hoc Network (MANET) [1] has become a popular communication technology in many tactical and mission critical environments such as, military defence networks, disaster and rescue operational command centres, vehicular networks, etc. The popularity is due to the unique and distinctive features of MANET such as mobility, self-organization, rapid deployment without the need of a central administration, low cost infrastructure, improved reliability, robustness and multi-hop routing [2].

Due to frequent changes in network topology as the nodes are free to move randomly and organize themselves arbitrarily, use of open wireless medium, limited physical security and resource constraints, MANET may potentially suffer from various security threats. Therefore, security issues in MANET have attained significant interest in the research community. The security vulnerabilities in MANET may range from passive attacks like passive eavesdropping, passive impersonation, message replay and message distortion to active attacks like deleting messages, injecting erroneous messages, impersonating a node, denial of service.

An ad-hoc routing protocol is a convention, or standard, that controls how nodes decide which way to route packets between computing devices in a Mobile Ad-Hoc Network. In ad-hoc networks, nodes are not familiar with the topology of their network. Instead, they have to discover it. Typically, a new node announces its presence and listens for announcements broadcasted by its neighbours. Each node learns about others nearby and how to reach them, and may announce that it too can reach them.

A major class of research in MANETs is focused on developing several efficient routing protocols [3] which incur minimum costs in terms of security, bandwidth and battery power. These protocols are classified into two categories: Reactive and Proactive protocols. In reactive routing protocols like Ad-Hoc On-Demand Distance Vector (AODV) [4]-[5], the nodes find the routes only when required. On the other hand, in proactive routing protocols like Optimized Link State Routing (OLSR) [6]-[7], nodes obtain routes by periodic exchange of topology information.

It has been observed that, most of these routing protocols rely on cooperation between the nodes due to the lack of a centralized administration as in the case with ad-hoc infrastructure-less scenarios. In a hostile environment, a malicious node can launch routing attacks to disrupt routing operations. It has been reported that none of the existing routing protocols incorporates decision making depending on the changes in performance and security requirements, node behavioural dynamics, context and timing constraints which is evidentially essential for the deployment of MANET in tactical environments [2].

In the next subsection, we discuss the motivation behind our research work on developing an adaptive routing protocol for Mobile Ad-Hoc Network.

1.1. Motivation and Objective

A large amount of critical data along with control messages are exchanged between groups of mobile nodes in a typical MANET environment. Such communications must cope up with changes in context, time and geographic location of deployment and ensure satisfaction of various performance metrics, various levels of security requirements, and robustness in presence of environmental constraints.

- † *These authors have contributed equally to this work.*

0730-3157/17 $31.00 © 2017 IEEE
DOI 10.1109/COMPSAC.2017.132

The violations of security and performance parameters may introduce incorrectness in sensitive data, delayed delivery of critical message and also the exposure of resources to malicious entities which in turn may inflict damage upon the nation's resources. On the other hand, inefficient execution of routing function in the mobile nodes may introduce significant latency in delivering critical messages.

This motivates our proposal on designing an adaptive and secure routing protocol for Mobile Ad-Hoc Networks and finally developing an efficient execution platform for the routing function using network function virtualization (NFV).

In this paper, we have analysed various contextual requirements and presented various models based on them which will then be used to develop the adaptive routing protocol. In addition, various functional, security, and Quality-of-Service (QoS) properties with respect to the requirement models have also been formally encoded. The requirement models and the query specifications are then verified using the Yices SMT Solver with respect to varying contextual requirements and constraints.

The remainder of this paper is organized as follows. In Section 2, we discuss the relevant related works on routing protocols in MANET. In Section 3, we present an overview of our proposed adaptive routing protocol design framework. In Section 4, we define the various requirement models of our proposed framework. Then in Section 5, the formal verification of the proposed models and the query specifications are performed. In Section 6, we discuss the efficacy of our proposed models with a toy example. Finally, we conclude the paper in Section 7.

2. Related Works

A number of research works have contributed in the area of routing protocol design for MANETs. Most of these works basically focus on designing efficient and secure routing protocols as discussed in this section.

The AODV algorithm [4]-[5] enables dynamic, self-starting, multihop routing between participating mobile nodes wishing to establish and maintain an ad-hoc network. Route Requests (RREQs), Route Replies (RREPs), and Route Errors(RERRs) are the message types defined by AODV.

OLSR is a proactive routing protocol [6]-[7] for Mobile Ad-Hoc Networks. The protocol inherits the stability of a link state algorithm and has the advantage of having routes immediately available when needed due to its proactive nature. It is an optimization over the classical link state protocol, tailored for Mobile Ad-Hoc Networks.

The routing protocols must be able to cope up with the high degree of node mobility that often changes the network topology drastically and unpredictably. As the mobility of the nodes is a factor of prime importance in a mission critical or tactical application context like the military scenario, the selection of the routing protocol should take into consideration the effect that the different mobility models have on the performance of the protocols.

A study [8] was conducted which implemented the three random based mobility models, Random way point, Random walk and Random directions and compared two different parameter constraints, packet-delivery fraction and end-to-end packet delivery delay with respect to mobility speed, traffic and network size. Based on the observations of the study it was concluded that AODV routing protocol can be used under high mobility since it outperforms Destination Sequenced Distance Vector (DSDV), Temporally Ordered Routing Algorithm (TORA) and Dynamic Source Routing (DSR) protocols. AODV uses fewer resources than OLSR, because the control messages size is kept small, requiring less bandwidth for maintaining the routes and the route table is also kept small, reducing the computational power. The AODV protocol can thus be used effectively in resource critical environments. Also, because AODV spends less resources, more cryptographically resource demanding solutions can be used for this protocol for enhancing the security [9].

Based on all the past experiments, studies and evidences [10]-[15], AODV protocol produces the best results and efficiency, especially considering the requirements of a military MANET setup and thus our prime focus will be on AODV while we try to design an adaptive and secure routing protocol for military MANETs.

Formal modelling and verification has played an important role in checking the correctness and consistency of different network functions with respect to the performance and security parameters in the last decade. The enterprise policy based security framework in [16] incorporates formal modelling of conflict free policy specifications in the network and finally deploys Boolean satisfiability (SAT) based verification procedure to check the conformation between the policy, constraints and implementation models. The policy conciliation framework for heterogeneous Mobile Ad-Hoc Networks in [17] uses Z Chaff SAT solver for checking the satisfiability of the combination of CNF (Conjunctive Normal Form) clauses. Another work in [18] uses the Yices SMT solver for finding the correct and optimal configuration.

Considering the significance of the current trends in use of formal methods in modelling and verification of security protocols in networking environments [16]-[18], we use the Yices SMT solver [19] as a formal verification tool as it provides all the functionalities and features required for designing the necessary adaptive routing protocol for MANET.

3. Proposed Adaptive Routing Protocol Design Framework

In this section, we present an overview of our proposed adaptive and secure routing protocol design framework for Mobile Ad-Hoc Networks which will dynamically configure the routing functions by taking even the contextual features into account with varying requirements for secure and real time processing of traffic.

There are two major modules in the design framework which are discussed in the following subsections.

3.1. Adaptive Routing Protocol Design

This module involves developing the novel routing algorithm based on various performance and security requirements, network topology and risk assessment with

node level behaviour and trust analysis. It consists of the following interactive tasks as shown in Figure 1:

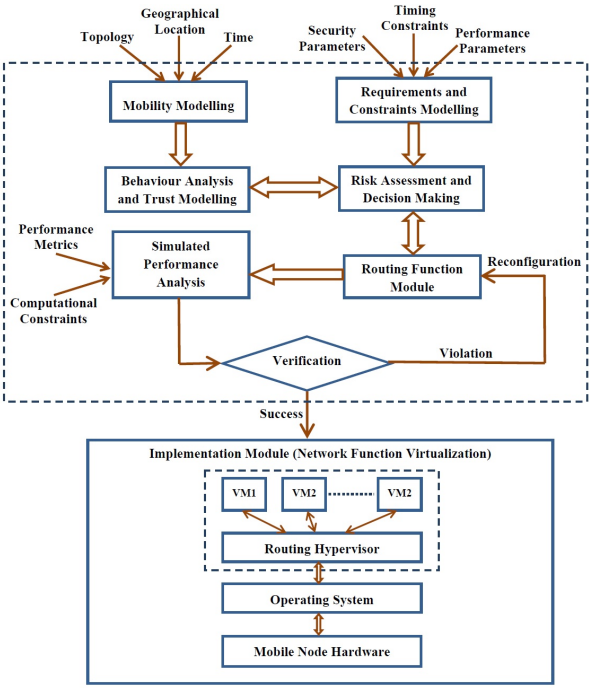

Figure 1. Adaptive Routing Protocol Design Framework

(i) ***Mobility Modelling***: This module extracts the mobility model of the Mobile Ad-Hoc Networks depending on the topology, geographic location and time information. The topology provides us with the properties of the mobile nodes, cluster information, intra and inter-cluster connectivity information.

(ii) ***Requirements and Constraints Modelling***: This module involves the modelling of various performance and security level requirements and timing constraints. The various requirements are categorized into three categories Performance (End-to-end delay, packet delivery ratio), Security (Confidentiality, integrity, availability, isolation, access control rules, device and software level vulnerability) and Timing (Deadlines, down-time, migration time). The building of the network topology, behavioural and the functional requirement models are also a part of this component.

(iii) ***Behaviour Analysis and Trust Modelling***: This module analyses the behaviour of the nodes based on the history as well as the recommendations from other nodes in the MANET cluster and procedurally derives a trust model for the MANET.

(iv) ***Risk Assessment and Decision Making***: This component takes input from behaviour analysis and trust modelling module and applies a systematic procedure for assessing the risk of various nodes and communication paths. Then, it applies decision making logic for generating the routing rules based on the requirement model and the trust model.

(v) ***Routing Protocol Building***: This module synthesizes the initial routing logic based on the routing rules from the risk assessment and decision making module. This routing logic can be modified depending on the real-time simulation and verification results from the successor modules or can be triggered by the changes in requirements and node behaviour.

(vi) ***Protocol Simulation***: This module simulates the routing logic on a real-time network simulation platform. The simulation platform can be configured based on the node level computational constraints (Processor clock, memory, battery life, etc.) and other real-time communication performance parameters (End-to-end delay, packet delivery ratio, bandwidth, etc.). The simulation generates results with various performance and security metrics.

(vii) ***Verification and Reconfiguration***: This module calculates the cumulative performance impact and security attack surface based on the simulation results and verifies these metrics with given performance thresholds (Throughput, end-to-end delay, etc.), security standards and attack scenarios (DoS, hidden tunnel, man-in-the-middle, false resource advertisement, etc.). In case of non-compliance with the established standards, the routing logic is reconfigured depending on the violation traces. The changes in requirements and node behaviours can be implemented using interrupts in the routing logic. In case of success, the corresponding routing logic is sent to the implementation module for realization.

3.2. Routing Logic Implementation Platform (Mobile Router Prototype)

This module involves developing an efficient and secure execution platform for implementing the routing logic. A novel routing task scheduling algorithm will be developed using network function virtualization (NFV), where various routing functions for different traffics with varying requirements can be intelligently scheduled into different virtual machines (VMs) that share the computational resources (processor, memory) of a mobile node. The task scheduling algorithm will be implemented as a hypervisor module that will run on the underlying mobile device operating system (Android OS). The routing functions will be redundantly implemented in different virtual machines for ensuring fault tolerance. The routing functions associated with different traffic will be randomly allocated to the virtual machines. This will improve the security perimeter over the execution platform. A mobile router prototype will be built based on the presented methodology.

In this paper, we focus on the most important module, i.e., “Requirements and Constraints Modelling” module, which is described in the following section in details.

4. Requirement Models of Proposed Adaptive Routing Protocol for MANET

This section illustrates the requirements specification model mathematically for building the routing protocol.

A Mobile Ad-Hoc Network is specified by a two-tuple $<N,G>$ where N is the node set and G is the set of groups (of nodes), each of which form a set of allied nodes. For simplicity, we divide the requirements specification model into three parts: (1) Network topology model, (2) Behavioural model, and (3) Functional model. Each of these models are mathematically described in the following subsections.

4.1. Network Topology Model

The Network topology model has the following entities.

(i) ***Node***: Every node n has a unique ID ($n.id \in N$) which is an address within a fixed global (within the MANET) address space and a group ($n.g \in G$). Node n_i can send a packet to n_j ($i \neq j$), if n_j is reachable from n_i. n_i refers to a node in N with ID i. Also, it follows that $\forall n_j \in N - n_j$, $j \neq i$.

(ii) ***Connectivity Matrix***: It is a $|N| \times |N|$ matrix, where the element a_{ij} is a real number $k \in (0, 1]$ ($i \neq j$), if n_i is in radio range of n_j (i.e., they can communicate directly). If they are not in each other's radio range, then the value of a_{ij} will be zero. If $a_{ij} > 0$, then nodes n_i and n_j are neighbours of each other and is denoted by $NB(n_i, n_j)$. The absolute value determines the strength of the connectivity between n_i and n_j.

4.2. Behavioural Model

The different entities of the Behavioural model are presented as follows.

(i) ***Trust Matrix***: It is a $|N| \times |N|$ matrix. An efficient trust model based on recommendations, central regulatory certifications, behavioural analysis, etc. assigns a trust value (between 0 and 1) to a node with respect to a particular other node. $T(n_i, n_j)$ represents the trust of a node n_j with respect to n_i, which is a real number within the range $[0, 1]$. $T(n_i, n_j) = 0$ denotes n_j is not trustworthy to n_i and $T(n_i, n_j) = 1$ denotes n_i trusts n_j completely. $U(n_i, n_j) = 1 - T(n_i, n_j)$ and it is the untrustworthiness of n_j with respect to n_i. On the basis of the trust values, the nodes take decisions or perform actions.

(ii) ***Network Instance***: A network instance is defined as a four tuple $<N,G,C,T>$ and is a function of time t, where N represents the set of nodes, G represents the set of groups, C represents the connectivity matrix of the nodes in N and T represents the trust matrix of the nodes in N.

4.3. Functional Model

The Functional model consists of the following entities.

(i) ***Services***: Each node can request for a variety of services from a set S. It can be a network service or a request for obtaining some information from some other node.

(ii) ***Reachability***: n_i is reachable to n_j, if n_i can communicate with n_j either directly or indirectly. It is denoted as $RCH(n_i, n_j)$. Every node must be reachable from at least one other node, which also satisfies the same condition. $RCH(n_i, n_j) \Rightarrow NB(n_i, n_j) \vee (\exists n_x \in N$, such that $(NB(n_i, n_x) \wedge RCH(n_x, n_j))$. $NB(n_i, n_j)$ corresponds to direct reachability and if $NB(n_i, n_j)$ is false, $RCH(n_i, n_j)$ corresponds to indirect reachability and implies that a valid route exists.

(iii) ***Properties of Nodes***: Every node $n \in N$ has various properties like location, reachability, role, terrain, and mobility.

- **Location**: Every node $n \in N$ has a GPS coordinate which identifies its location or geographical position which may keep changing over time. The current location is given by $POS(n)$.
- **Role**: Every node $n \in N$ has a role index represented by $R(n)$, which takes values from the range of natural numbers up to a number N_r (number of types of roles), where each number represents a particular role in increasing value of priorities/powers depending on the MANET. The role may change depending on the activity, etc. assigned to the particular node. Each node needs to necessarily have a role assigned to it.
- **Terrain**: Every node $n \in N$ has a terrain index represented by $TR(n)$, which takes values from the range of natural numbers up to a number N_t (number of types of terrains), where each number represents a particular terrain. This may change depending on the geographical location of the node at a particular time. Each node needs to necessarily have a terrain index assigned to it.
- **Mobility**: Every node $n \in N$ has a mobility index represented by $MOB(n)$, which takes values from the range of natural numbers up to a number N_m (number of types of mobility models), where each number represents a particular mobility model (like random waypoint mobility model, etc). Every node needs to necessarily have a mobility model it follows.
- **Capacity**: Every node $n \in N$ has a capacity value represented by $CAP(n) \in [0, 1]$. This value is computed by taking into consideration the following parameters of the nodes:

– Remaining battery power
– Memory usage
– CPU utilization
– Network usage

Capacity $CAP(n)$ of a node n is mathematically defined as:

$$CAP(n) = w_E * Pow(n) + w_M * Mem(n) + w_C * CPU(n) + w_U * Net(n) \quad (1)$$

where, $Pow(n)$, $Mem(n)$, $CPU(n)$, and $Net(n)$ are the node parameters, i.e., remaining battery power, memory usage, CPU utilization, and network usage of a node n respectively at a particular time instant t. w_E, w_M, w_C, and w_U are the weights assigned to remaining battery power, memory usage, CPU utilization, and network usage respectively. The weights are statically assigned to these parameters such that

$$w_E + w_M + w_C + w_U = 1 \quad (2)$$

These parameters of node n are mathematically defined as follows.

- **Remaining Battery Power**: It is denoted by $Pow(n)$ and is normalized to be in the interval $[0, 1]$.

  $$Pow(n) = f(V(n), I(n), Bcap(n), \alpha(n)) \quad (3)$$

  where, f is defined by the particular battery decay model used and is a function of : (i) Voltage level $V(n)$, (ii) Current flow $I(n)$, (iii) Original battery capacity $Bcap(n)$, and (iv) Battery decay parameter $\alpha(n)$ specific to the type of battery used.

- **Memory Usage**: The memory usage $Mem(n)$ of a node n is mathematically expressed as:

  $$Mem(n) = \frac{\sum_{i=1}^{k} MemCons(i)}{TotalMem(n)} \quad (4)$$

  where, $MemCons(i)$ is the memory space consumed by application i either under execution or in suspended state within node n, $TotalMem(n)$ is the total memory space of the node n and k is the total number of applications within node n.

- **CPU Utilization**: The CPU utilization of a node n is denoted by $CPU(n)$ and is defined as:

  $$CPU(n) = 1 - \prod_{i=1}^{m} P(i) \quad (5)$$

  where, $P(i)$ is the probability of application i in node n either being idle or waiting for any I/O operation and m is the total number of applications within node n.

- **Network Usage**: The network usage $Net(n)$ of a node n is mathematically defined as:

  $$Net(n) = \frac{\sum_{i=1}^{p} BW(i)}{TotalBW(n)} \quad (6)$$

  where, $BW(i)$ is the bandwidth utilized on interface i of node n, $TotalBW(n)$ is the total bandwidth allocation for node n and p is the total number of interfaces of node n.

These node parameters always lie between 0 and 1.

(iv) ***Functions of Nodes***: The nodes in MANET may exhibit different functions during their life time. These functions are as follows.

- $SEND(n_i, n_j, rep, s)$: It denotes n_i sending a request or reply to n_j for a service $s \in S$ depending on the value of rep. If rep is NULL, it denotes a request from n_i to n_j for a service $s \in S$. If rep is not NULL, it denotes n_i sending a reply/response to n_j after it had earlier received a request from n_j for service s for which the reply rep is an answer. Reply might be encrypted or in plain-text depending on the type of reply and the policies of the replying node and network.

- $FWD(n_i, msg)$: It denotes node n_i forwarding the network packet msg. n_i forwards msg to a node $n_j \in RT(n_x, n_y)$ such that $NB(n_i, n_j)$ is true, msg is the network packet originated at n_x for n_y and $RT(n_x, n_y)$ represents the route from n_x to n_y. This decision is taken based on bandwidth constraints, maliciousness of nodes, contextual constraints, etc.

- $DR(n_i, msg)$: It denotes node n_i dropping the network packet msg, where msg is the network packet originated at some source node n_x.

- $ATK(n_i)$: It denotes node n_i demonstrating itself to be a malicious or an adversary node. In order to ensure a safe network state, it is assumed that a maximum of 5% of the total number of network nodes can become malicious at any point of time before the breakdown of the network occurs.

(v) ***Route***: It is denoted by $RT(n_i, n_j)$, which is an ordered set of nodes that would be able to forward the packets from n_i to n_j iff $RCH(n_i, n_j)$ is true. If $NB(n_i, n_j)$ is true, $RT(n_i, n_j) = \{n_i, n_j\}$. If $NB(n_i, n_j)$ is false, then if $(\exists n_x \in N$, such that $(NB(n_i, n_x) \wedge RCH(n_x, n_j))$ is true), then $RT(n_i, n_j) = \{n_i, n_x, ..., n_j\}$.

(vi) ***Trust of Route***: It is represented by $TRT(n_i, n_j)$. It depends on the untrustworthiness of the nodes in route with respect to their preceding nodes in the

route. It is defined as:

$$TRT(n_i, n_j) = 1 - \prod_{k=i+1}^{j} u_k \quad (7)$$

where u_x represents the untrustworthiness of node x with respect to its preceding node in $RT(n_i, n_j)$, where $RT(n_i, n_j)$ = $\{n_i, n_{i+1}, n_{i+2}..., n_j\}$.

In the next section, we formally define various queries related to different functional, security, and Quality-of-Service (QoS) properties with respect to the aforementioned requirement models. These queries are encoded using SMT-LIB language in Yices formal verification tool with which the properties are verified.

5. Formal Verification of the Proposed Model

Once the requirement models of a MANET is well-defined, it is important to ensure the correctness and consistency of the proposed routing protocol with respect to these contextual requirements. In the next subsection, we first define the types of queries necessary for verification of the proposed model. Then in the subsequent subsection, we formally encode these queries using the SMT-LIB language. Then, we verify the different network properties using the Yices formal verification tool.

5.1. Query Specification for Verification of Network Properties

The different network properties of a Mobile Ad-Hoc Network are verified through several queries. We define these queries using high level query specification terminology. The different queries are explained as follows.

(1) ***Functional Queries***:
The functional queries in the context of MANET are important in verifying the functionality of the proposed routing protocol. We define the following types of functional queries with respect to the aforementioned requirement models.

- ***Reachability***: To ensure successful data delivery, reachability must hold between the sender node and the receiver node. The reachability query is specified as follows:

  CHECK REACHABILITY
  FROM SOURCE x ∈ N
  TO DESTINATION y ∈ N
  AT TIME t

  The query is formalized as:

  $REACHABLE_{x,y,t} \Longleftrightarrow NEIGHBOUR_{x,y,t} \vee$
  $[\exists z \in N, (NEIGHBOUR_{x,z,t} \wedge$
  $REACHABLE_{z,y,t})]$.

- ***Route computation***: Optimal route computation between a pair of nodes is the key objective of our proposed routing protocol for MANET. The query for computing route is specified as follows.

  COMPUTE ROUTE RT
  FROM SOURCE x ∈ N
  TO DESTINATION y ∈ N
  AT TIME t

  The query is formalized as:

  $ROUTE_{x,y,t} \Longleftrightarrow REACHABLE_{x,y,t}$

- ***Consistency***: In addition to determining an optimal route, we need to check the consistency of different node properties over a period of time. This helps us in optimization of route computation. In this paper, we determine the consistency of trust levels of individual nodes during route computation and data transmission. The consistency query is specified as:

  CHECK CONSISTENCY
  OF TRUST LEVEL T
  OF NODE n ∈ ROUTE RT
  DURING TIME INTERVAL Δt

  The query is formalized as:

  $CONSISTENT_{n \in RT, T(n), \Delta t} \Longleftrightarrow \forall (t_i, t_{i+1})$,
  $i \in \{0, 1, ..., q\}$, Δt = $\{t_0, t_1, ..., t_q\} \Rightarrow$
  $|T_{t_i}(n) - T_{t_{i+1}}(n)| \longrightarrow 0$

(2) ***Security Queries***:
The main thrust of our proposed routing protocol is to determine a secure and adaptive routing path between a pair of nodes. The security related queries are helpful in ensuring security in route computation and data transmission, which are defined as follows.

- ***Trusted Route Computation***: It is important to ensure the trustworthiness of the route computed between the nodes. Thus, we incorporate trust values along with route computation in the context of MANET. The query is specified as follows.

  COMPUTE TRUSTED ROUTE TRT
  FROM SOURCE x ∈ N
  TO DESTINATION y ∈ N
  AT TIME t

  The query is formalized as:

  $TRUSTED\ \ ROUTE_{x,y,t} \Longleftrightarrow$
  $ROUTE_{x,y,t} \wedge (\forall n \in RT,\ T_t(n) \in [0.6,1])$

- ***Attack Surface***: It signifies the fraction of the network affected due to a security breach. For example, DoS attack surface is the portion of the network affected due to a DoS attack by some malicious

nodes. Attack surface in a network is defined as:

$$Attack\ surface = \frac{Number\ of\ affected\ nodes}{Network\ size} * 100 \quad (8)$$

We determine the attack surface of a Mobile Ad-Hoc Network with the query as stated below.

FIND ATTACK SURFACE
RELATED TO NODE $n \in$ N
WHERE $ATK(n)$ IS TRUE
AT TIME t

The query is formalized as:

$ATTACK\ \ SURFACE_{n \in N,t} \Longleftrightarrow ATK(n)$=1 $\wedge$
$(\forall i \in NEIGHBOUR_{n,i,t},\ T_t(i) < 0.3)$

(3) ***Quality-of-Service Queries***:
The queries on Quality-of Service (QoS) in MANET are an inherent part of our proposed routing protocol. These queries are used in determining the capacity of the nodes in the network during route computation. This value is computed by taking into consideration the node's remaining battery power, memory usage, CPU allocation and network usage. We determine the capacity of a Mobile Ad-Hoc Network node with the query as stated below.

FIND CAPACITY
OF NODE n $\in$ ROUTE RT
AT TIME t

The query is formalized as:

$$CAPACITY_{n \in RT,t} = w_E * Pow(n) + w_M * Mem(n) + w_C * CPU(n) + w_U * Net(n)$$

5.2. SMT-LIB Reduction of Requirement Models

This subsection describes the reduction of the requirement models discussed in Section 4 into the SMT-LIB language which further is used for formal verification using the Yices SMT Solver. The SMT-LIB reductions of the network topology, behavioural and functional models are as presented in Table 1.

TABLE 1. SMT-LIB REDUCTION OF REQUIREMENT MODELS

Network topology model
(define-type n (subtype (list node::int)))
(define-type g (subtype (list group::int)))
(define-type c (subtype (list connectivity::real ($\geqslant 0$)($\leqslant 1$))))
Behavioural model
(define-type t (subtype (list array trust::real ($\geqslant 0$)($\leqslant 1$))))
(define-type net_inst::(tuple int int real real))
(define i :: net_inst)
Functional model
(define-type s (subtype (list array service::scalar)))
(define-type node_prop::(tuple (define l::(tuple int int)) int int real))
(define p ::node_prop)

5.3. SMT-LIB Reduction of Query Specifications

In this subsection, the reduction of the query specifications discussed in Section 5.1 into the SMT-LIB language is described. These are then used for the formal verification using the Yices SMT Solver. The SMT-LIB reductions of the functional, security and QoS queries are as presented in Table 2.

TABLE 2. SMT-LIB REDUCTION OF QUERY SPECIFICATION

Functional queries
(define reachability::($\rightarrow$ int int scalar bool))
(assert (reachability($\Rightarrow$ (>0 (select c (x::node) (y::node) (tm::scalar))) $\vee$ (exists (z::node) (>0 (select c x z tm)) $\wedge$ (reachability z y tm))))
(check)
(define route::($\rightarrow$ int int scalar list))
(assert (route($\Rightarrow$ ((reachability x y tm) (list r::subtype (n::node))))))
(check)
(define consistency::($\rightarrow$ int scalar scalar bool))
(assert (consistency($\Rightarrow$ (n::node) (($<$ (tm_1::scalar) (tm_2::scalar)) or ($\leqslant$ 0.1 (- (select t n tm_1 (a::trust)) (select t n tm_2 (b::trust)))) ($\leqslant$ 0.1 (- (select t n tm_2 (b::trust)) (select t n tm_1 (a::trust))))))
(check)
Security Queries
(define trusted_route::($\rightarrow$ int int scalar list))
(assert (trusted_route($\Rightarrow$ ((reachability x y tm) $\vee$ (forall (select (route x y tm) (z::node)) ($\geqslant$ 0.6 (select t z (t_z::trust))))(list r::subtype (n::node))))))
(check)
(define attack_surface::($\rightarrow$ int scalar real))
(define sum::int (0))
(assert (($\geqslant 0$)($\leqslant 1$) attack_surface(forall (y::node) (>0 (select c x y tm)) ($<$ 0.3 (select t y tm (b::trust))))) (= / ((define sum_y::int (= (if (exists y) (+ sum 1) (+ sum 0))) n))))
(check)
Quality-of-Service queries
(define capacity::($\rightarrow$ int real))
(define Pow::real ($\geqslant 0$)($\leqslant 1$))
(define Mem::real ($\geqslant 0$)($\leqslant 1$))
(define CPU::real ($\geqslant 0$)($\leqslant 1$))
(define Net::real ($\geqslant 0$)($\leqslant 1$))
(define w_1::real ($\geqslant 0$)($\leqslant 1$))
(define w_2::real ($\geqslant 0$)($\leqslant 1$))
(define w_3::real ($\geqslant 0$)($\leqslant 1$))
(define w_4::real ($\geqslant 0$)($\leqslant 1$))
(assert (= 1 (+ + + w_1 w_2 w_3 w_4))
(assert (($\geqslant 0$)($\leqslant 1$) capacity(= (+ + + (* w_1 Pow) (* w_2 Mem) (* w_3 CPU) (* w_4 Net))))
(check)

6. Verification Results and Discussion

Our proposed requirement modelling and verification framework has been implemented using the Yices SMT solver. The verification engine takes the network toplology, behavioural and functional models of the network as input and verifies the satisfiability of the requirements (using queries) in that model. It generates results, which is either SAT (satisfiable) or UNSAT (unsatisfiable). The SAT result indicates that there exists a satisfiable instance in the configuration against the verification query and otherwise the result is specified as UNSAT. These results help in assessing the correctness and consistency of the network model and thereby provides recommendations/inputs for designing adaptive routing functions for the network. In this section, we present the accuracy, and scalability of our framework supported by experimental results.

6.1. Accuracy

Firstly, the accuracy of our framework is ensured by the use of formal constraint satisfaction checking method.

In addition, we have tested our framework on a small MANET testbed. The basic topology of our testbed is shown in Figure 2. For the purpose of simplicity, here we have not mentioned the various parameters used in the different nodes. We have varied the network size between 20 to 200 nodes with heterogeneous network properties. The results for some of the requirement constraints (queries) have been shown in Table 3. For the purpose of analyzing these constraints, we slide the values of different configuration parameters such as location, role, capacity, and mobility of nodes. We find some constraint violations that lead to untrusted route and reachability violations. Our verification engine also reports the attack surface with respect to a untrusted/attack-prone node. These results significantly support in verifying the accuracy of the framework.

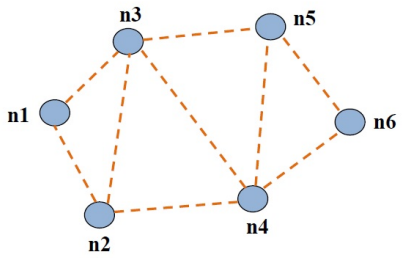

Figure 2. MANET Test Topology

TABLE 3. CONSTRAINT VERIFICATION FOR THE MANET EXAMPLE

Queries	Results
Functional queries	
(define n1::n)	→ sat
(define n2::n)	(=rt n1 n2 n4 n6)
.	(=rt n1 n3 n5 n6)
.	(=rt n1 n3 n4 n6)
(define n6::n)	(=rt n1 n2 n3 n5 n6)
(define c1 (n1 n2)::c)	(=rt n1 n2 n4 n5 n6)
(define c2 (n1 n3)::c)	.
.	.
.	.
(define c9 (n4 n6)::c)	(=rt n1 n2 n3 n4 n5 n6)
(define-type rt (subtype (list route::int)))	
(assert (⇒ (reachability n1 n6) rt))	
(check)	
Security Queries	
(define-type rt (subtype (list route::int)))	→ sat
(assert (⇒ (trusted_route n1 n6)) trt)	(=trt n1 n2 n4 n6)
(check)	
(assert (⇒ (trusted_route n2 n5) trt))	→ sat
(check)	unsat core ids: n3
Quality-of-Service queries	
(assert ((⩾ 0)(⩽ 1) capacity(= n1)	→ sat
(check)	(=capacity 0.8)

6.2. Scalability

We evaluate the scalability of our framework by analyzing the time and space required in constraint verification with varying network sizes.

***Impact of Network Size*:** Figure 3 and 4 illustrate the execution time of query verification with respect to network size. We report the verification time for functional and security queries. We have observed that the execution time for verifying reachability query is related quadratically with network size. This is because of the recursive implementation of the reachability function. We also observed the variation in execution time with different mobility models such as: Random waypoint model and Gauss-Markov model. On the other hand, the trusted route computation time is divided into two components namely route computation time and trust evaluation time. It has been observed that the variation of route computation time is cubic in nature with varying network size, while it is linear for trust calculation.

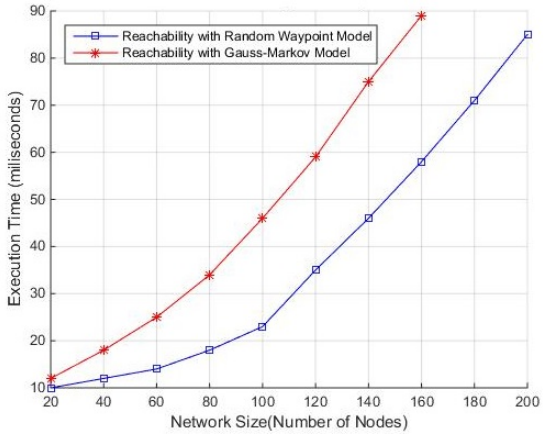

Figure 3. Execution time of reachability verification with respect to network size

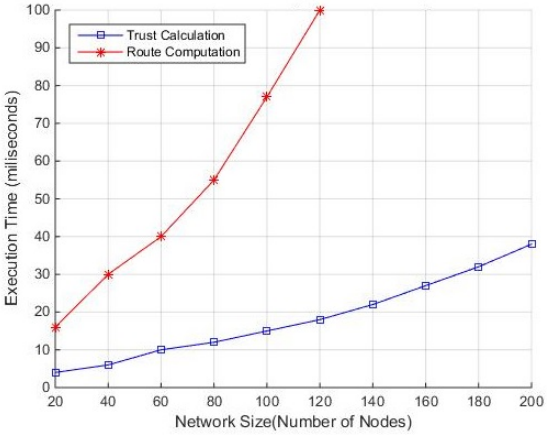

Figure 4. Execution time of trusted route computation with respect to network size

The capacity constraint verification time varies randomly due to the unpredictable nature of the components used in calculating the capacity of different nodes. For example, the unpredictability of the battery decay model being followed, CPU utilization at the time instant of calculation, etc. impact on this calculation.

***SMT Space Requirement*:** The space requirement (memory used) of the SMT Solver is evaluated by changing the network size. We observe that the space requirement increases linearly with the network size. The constraints involving more quantifiers require larger memory space for encoding.

7. Conclusion and Future Work

In various critical and tactical applications, a large amount of critical data along with control messages is being exchanged through various nodes in MANET. In such situations, the requirements (mobility, performance, security, trust, and timing constraints) vary with change in context, time, and geographic location of deployment. The violations of these requirements may introduce incorrectness to sensitive data, incorrect routing decisions, delayed delivery of critical messages, and exposure and tampering of resources. We have presented a framework for modelling the heterogeneous requirements in Mobile Ad-Hoc Networks and verification of various constraints with respect to the requirement models. The verification engine is built using Yices SMT Solver, which is capable of solving a large number of clauses. The efficacy of our framework has been demonstrated in terms of accuracy and scalability supported by experimental results. The modelling and verification time, space complexity for different constraints have been reported. The verification results will help in automatic assessment of the dynamic network configuration with change in context and thereby can be used in designing algorithms for adaptive routing in MANET. In future, we will use the proposed modelling and verification framework for developing a context-aware adaptive and secure routing protocol for MANET. We will also work on designing protocols for ensuring end-to-end security in MANET.

References

[1] J. Loo, J. Lloret, J. H. Ortiz. *Mobile Ad Hoc Networks: CurrentStatus and Future Trends*, Boca Raton, FL, USA. In CRC, 2011.

[2] B. K. Tripathy, P. Bera, M. A. Rahman. *Analysis of trust models in Mobile Ad Hoc Networks: A simulation based study*. In IEEE COMSNETS, pages 1-8, January 2016.

[3] S. Corson, J. Macker. *Mobile Ad hoc Networking(MANET): Routing Protocol Performance Issues and Evaluation Considerations*. In RFC 2501, IETF Network Working Group, January 1999.

[4] E. Belding-Royer, C. Perkins, S. Das. *Ad Hoc On-Demand Distance Vector (AODV)Routing*. In RFC 3561, IETF Network Working Group, July 2003.

[5] I. D. Chakeres, E. Belding-Royer. *AODV routing protocol implementation design*. In IEEE Proceedings, 24th International Conference on Distributed Computing Systems Workshops, pages 698-703, March 2004.

[6] T. Clausen, P. Jacquet. *Optimized Link State Routing Protocol (OLSR)*. In RFC 3626, IETF Network Working Group, October 2003.

[7] P. Jacquet et al.. *Optimized link state routing protocol for ad hoc networks*. In IEEE INMIC, pages 62-68, 2001.

[8] M. K. Jeya Kumar, R. S. Rajesh. *Performance Analysis of MANET Routing Protocols in Different Mobility Models*. In IJCSNS International Journal of Computer Science and Network Security, VOL.9 No.2, pages 22-29, February 2009.

[9] A. Huhtonen. *Comparing AODV and OLSR routing protocols*. In HUT T-110.551 Seminar on Internetworking, pages 1-9, Sjokulla, Finland, 2004.

[10] S. A. Ade, P. A. Tijare. *Performance comparison of AODV, DSDV, OLSR and DSR routing protocols in mobile ad hoc networks*. In International Journal of Information Technology and Knowledge Management 2.2, pages 545-548, 2010.

[11] Q. Razouqi, A. Boushehri, M. Gaballah, L. Alsaleh. *Extensive simulation performance analysis for DSDV, DSR and AODV MANET routing protocols*. In 27th International Conference on Advanced Information Networking and Applications Workshops (WAINA), pages 335-342, March 2013.

[12] M. S. Islam, M. N. Hider, M. T. Letonmiah. *An Extensive Comparison among DSDV, DSR and AODV Protocols in MANET*. In International Journal of Computer Applications, Volume 15 No.2: pages 22-24, February 2011.

[13] R. Jain, N. Khairnar, L. Shrivastava. *Comparative study of three mobile ad-hoc network routing protocols under different traffic source*. In International Conference on communication Systems and Network technologies, IEEE, pages 104-107, June 2011.

[14] M. Arefin, M. Tawhiddul, I. Toyoda. *Performance Analysis of Mobile Ad-hoc Networks routing protocols*. In International conference on Informatics and Vision, IEEE, pages 535-539, May 2012.

[15] N. Naqvi, U. Gupta, S. Kochhar, R. Agarwal. *Mobile Ad Hoc Network Routing Protocols on the Basis of Energy Consumption*. In Proceedings of the 5th National Conference, INDIACom, March 2011.

[16] P. Bera, S. K. Ghosh, P. Dasgupta. *Policy based security analysis in enterprise networks: A formal approach*. In IEEE Transactions on Network and Service Management, 7(4), pages 231-243, December 2010.

[17] S. Maity, S. K. Ghosh, E. AlShaer. *PoliCon: a policy conciliation framework for heterogeneous mobile ad hoc networks*. In Security and Communication Networks 8.3, Wiley, pages 418-430, February 2015.

[18] S. K. Majhi, P. Bera, S. Kumar, E. Al-Shaer. *Synthesizing optimal security configurations for enterprise networks: a formal approach*. In 9th IET International Conference on System Safety and Cyber Security, IET, pages 1-7, October 2014.

[19] B. Dutertre and L. De Moura, *The Yices SMT Solver 2006, Technical Report [Online]*. Available: http://yices.csl.sri.com/tool-paper.pdf.