Impact of Multi-Path Security in Wireless Ad Hoc Networks in Indoor Environments by using AOMDV Methods

Seyed Amin Ahmadi Olounabadi, Avula. Damodaram, V Kamakshi Prasad, Mahdi Hosseini

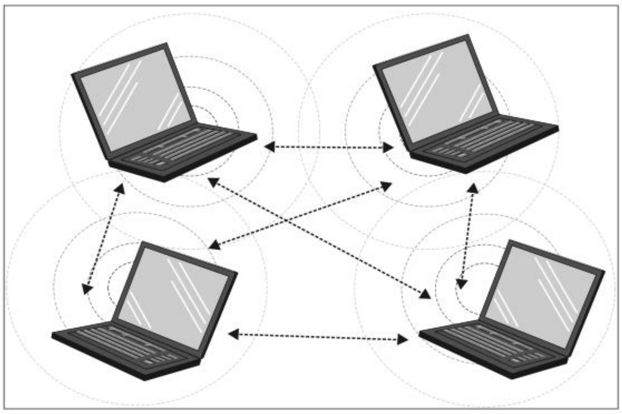
***Abstract:** Ad hoc Network is a decentralized type of wireless network and also is a local area network (LAN) that is built spontaneously as devices connect. , Instead of relying on a base station to coordinate the flow of messages to each node in the network, the individual network nodes forward packets to and from each other. Basically, an ad hoc network is a temporary network connection created for a specific purpose (such as transferring data from one computer to another). Multipath routing is the routing technique of using multiple alternative paths through a network, which can yield a variety of benefits such as fault tolerance, increased bandwidth, or improved security. Ad-hoc On-demand Multipath Distance Vector Routing (AOMDV) protocol is an extension to the AODV protocol for computing multiple loop-free and link disjoint paths and also increases the reliability through transmitting the messages in multiple paths with minimal redundancy, which used in present work. Simulations were conducted using the NS2 network simulator. In order to simulate most of the proposed Byzantine attacks in NS2, a protocol independent Byzantine attack simulation module was developed. This module provides the capability to simulate the black hole, Byzantine wormhole, and Byzantine overlay network wormhole attacks without modifying the routing protocol. We are considering our communication path is changeable even path or node is node failed. So data is sending through different paths, it provide high security than single path.*

***Key words:** wireless network, Ad hoc, AOMDV, Byzantine attacks*

I. INTRODUCTION

A. Ad hoc Network Definition

Ad hoc Network is a decentralized type of network and also is a local area network (LAN) that is built spontaneously as devices connect. The network is ad hoc because it does not rely on a pre-existing infrastructure, such as routers in wired networks or access points in managed (infrastructure) wireless networks, Instead of relying on a base station to coordinate the flow of messages to each node in the network, the individual network nodes forward packets to and from each other. Basically, an ad hoc network is a temporary network connection created for a specific purpose (such as transferring data from one computer to another). Each node participates in routing by forwarding data for other nodes, so the determination of which nodes forward data is made dynamically on the basis of network connectivity. In addition to the classic routing, ad hoc networks can use flooding for forwarding data. Wireless ad hoc networks are self-configuring, dynamic networks in which nodes are free to move. Wireless networks lack the complexities of infrastructure setup and administration, enabling devices to create and join networks "on the fly" – anywhere, anytime



B. Security

Most ad hoc networks do not implement any network access control, leaving these networks vulnerable to resource consumption attacks where a malicious node injects packets into the network with the goal of depleting the resources of the nodes relaying the packets. To thwart or prevent such attacks, it was necessary to employ authentication mechanisms that ensure that only authorized nodes can inject traffic into the network .Even with authentication, these networks are vulnerable to packet dropping or delaying attacks, whereby an intermediate node drops the packet or delays it, rather than promptly sending it to the next hop. Some behaviour-based detection techniques have been developed to counter such attacks in which a node overhears communication in the wireless neighbourhood and determines if a neighbour is behaving correctly, i.e., forwarding the packet toward the intended recipient promptly.

Manuscript published on 30 December 2016.
***** Correspondence Author (s)

Seyed Amin Ahmadi Olounabadi, Ph.D. scholar student, Department of Computer Science and Engineering, Jawaharlal Nehru Technological University Hyderabad (JNTUH), Hyderabad (Telangana) India.

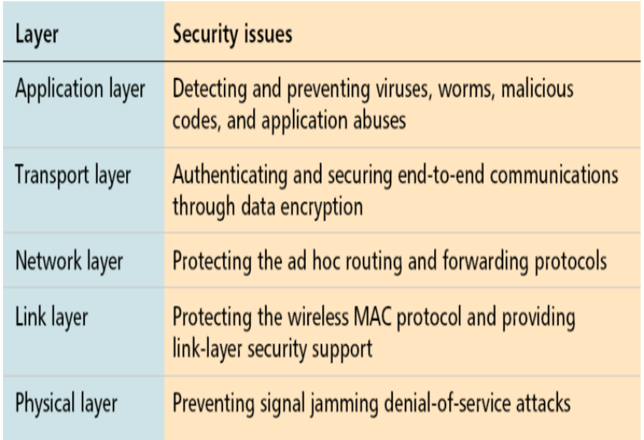
Prof. Avula. Damodaram, Faculty of Computer Science & Engineering, Vice-Chancellor Sri Venkateswara University, Tirupati (Andhra Pradesh) India.

Prof. V Kamakshi Prasad, Head of Department Computer Science and Engineering, Jawaharlal Nehru Technological University Hyderabad (JNTUH), Hyderabad (Telangana) India.

Mahdi Hosseini, Ph.D. Scholar Student, Dept. of Civil Engineering, Jawaharlal Nehru Technological University Hyderabad (JNTUH), Hyderabad (Telangana) India.

Security Requirements in Ad hoc Networks:
1- Availability
2- Authorization and Key Management
3- Data Confidentiality
4- Data Integrity
5- Non-repudiation

C. Security Issues:



Layer	Security issues
Application layer	Detecting and preventing viruses, worms, malicious codes, and application abuses
Transport layer	Authenticating and securing end-to-end communications through data encryption
Network layer	Protecting the ad hoc routing and forwarding protocols
Link layer	Protecting the wireless MAC protocol and providing link-layer security support
Physical layer	Preventing signal jamming denial-of-service attacks

D. Wireless Network

A wireless network is any type of computer network that uses wireless data connections for connecting network nodes (not connected by cables of any kind). Wireless networking is a method by which homes, telecommunications networks and enterprise (business) installations avoid the costly process of introducing cables into a building, or as a connection between various equipment locations. Wireless telecommunications networks are generally implemented and administered using radio communication (radio waves). This implementation takes place at the physical level (layer) of the OSI model network structure. Examples of wireless networks include cell phone networks, Wireless local networks, wireless sensor networks, satellite communication networks, and terrestrial microwave networks.

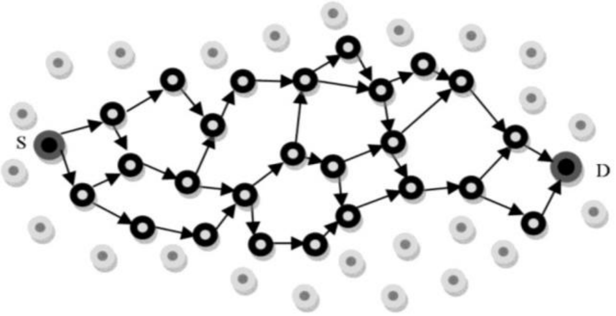
E. Multipath routing

Multipath routing is the routing technique of using multiple alternative paths through a network, which can yield a variety of benefits such as fault tolerance, increased bandwidth, or improved security. The multiple paths computed might be overlapped, edge-disjointed or node-disjointed with each other.

In this form, each stream is assigned a separate path, uniquely to the extent supported by the number of paths available. If there are more streams than available paths, some streams will share paths. This provides better utilization of available bandwidth by creating multiple active transmission queues. It also provides a measure of fault tolerance in that, should a path fail, only the traffic assigned to that path is affected, the other paths continuing to serve their stream flows; there is also, ideally, an alternative path immediately available upon which to continue or restart the interrupted stream.

This method provides better transmission performance and fault tolerance by providing:

- Simultaneous, parallel transport over multiple carriers.
- Load balancing over available assets.
- Avoidance of path discovery when reassigning an interrupted stream.



F. Indoor Environments

Indoor environments, Set up by using a pair of wireless laptops to file share where there is human movement between the two nodes, Wi-Fi link throughput is measured in an obstructed office block, laboratory, library, and suburban residential home environments.

G. Mobile Ad-hoc Networks

An ad-hoc network is a collection of wireless mobile hosts forming a temporary network without the aid of any stand-alone infrastructure or centralized administration. Mobile Ad-hoc networks are self-organizing and self-re-configuring multi hop wireless networks where, the structure of the network changes dynamically. This is mainly due to the mobility of the nodes. Nodes in these networks utilize the same random access wireless channel, cooperating in a friendly manner to engaging themselves in multi hop forwarding. The nodes in the network not only act as hosts but also as routers that route data to/from other nodes in network. In mobile ad-hoc networks where there is no infrastructure support as is the case with wireless networks, and since a destination node might be out of range of a source node transmitting packets; a routing procedure is always needed to find a path so as to forward the packets appropriately between the source and the destination. Within a cell, a base station can reach all mobile nodes without routing via broadcast in common wireless networks. In the case of ad-hoc networks, each node must be able to forward data for other nodes. This creates additional problems along with the problems of dynamic topology which is unpredictable connectivity changes.

MANETS rely on wireless transmission, a secured way of message transmission is important to protect the privacy of the data. An insecure ad-hoc network at the edge of communication infrastructure may potentially cause the entire network to become vulnerable to security breaches. There is no central administration to take care of detection and prevention of anomalies in mobile ad hoc networks. Mobile devices identities or their intentions cannot be predetermined or verified. Therefore nodes have to cooperate for the integrity of the operation of the network. However, nodes may refuse to cooperate by not forwarding packets for others for selfish reasons and not want to exhaust their resources. Various other factors make the task of secure communication in ad hoc wireless networks difficult include the mobility of the nodes, a promiscuous mode of operation, limited processing power, and limited availability of resources such as battery power, bandwidth and memory. Therefore nodes have to cooperate for the integrity of the operation of the network. Nodes may refuse to cooperate by not forwarding packets for others for selfish reasons and not want to exhaust their resources.

Attacks on ad hoc are classified into non-disruptive passive attacks and disruptive active attacks. The active attacks are further classified into external attacks and internal attacks are carried out by nodes that do not belong to network and can be prevented by firewalls and encryption techniques. Internal attacks are from internal nodes which are actually authorized nodes and part of the network hence it is difficult to identify

Secure Message Transmission protocol provides security based on the security association between the end nodes. It is not able to overcome the compromised nodes attacks. The work presented in this paper has two phases. The first phase is to improve the security and reliability of data transmission in mobile ad hoc networks by providing secured routes.

The Byzantine faults are identified and those links will be avoided in the data transmission phase. The current topological information will be gathered based on the network behavior such as transmission time, Probability of lost packets and correctly received – acknowledged packets and a threshold is set which is used in binary search probing. The nodes may exchange the current velocity vectors such as speed and direction to predict the location of the nodes. The spatial and temporal mining can be used to find the relative appropriateness of the location. Mobile devices identities or their intentions cannot be predetermined or verified. Therefore nodes have to cooperate for the integrity of the operation of the network. However, nodes may refuse to cooperate by not forwarding packets for others for selfish reasons and not want to exhaust their resources. Various other factors make the task of secure communication in ad hoc wireless networks difficult include the mobility of the nodes, a promiscuous mode of operation, limited processing power, and limited availability of resources such as battery power, bandwidth and memory. Therefore nodes have to cooperate for the integrity of the operation of the network. Nodes may refuse to cooperate by not forwarding packets for others for selfish reasons and not want to exhaust their resources. In ad hoc networks devices (also called nodes) act both as computers and routers. Most routing protocols lead nodes to exchange network topology information in order to establish communication routes. This information is sensitive and may become a target for malicious adversaries who intend to attack the network or the applications running on it. There are two sources of threats to routing protocols. The first comes from external attackers. By injecting erroneous routing information, replaying old routing information, or distorting routing information, an attacker could successfully partition a network or introduce a traffic overload by causing retransmission and inefficient routing. The second and more severe kind of threat comes from compromised nodes, which might (i) misuse routing information to other nodes or (ii) act on applicative data in order to induce service failures. The provision of systematic approaches to evaluate the impact of such threats on particular routing protocols remains an open challenge today. Attacks on ad hoc are classified into non-disruptive passive attacks and disruptive active attacks. The active attacks are further classified into internal attacks and external attacks are carried out by nodes that do not belong to network and can be prevented by firewalls and encryption techniques. Internal attacks are from internal nodes which are actually authorized nodes and part of the network hence it is difficult to identify

H. Problems with Routing In Mobile Ad-Hoc Networks

There are 15 major issues and sub-issues involving in MANET such as routing, multicasting/broadcasting, location service, clustering, mobility management, TCP/UDP, IP addressing, multiple access, radio interface, bandwidth management, power management, security, fault tolerance, QoS/multimedia and standards/products. Currently, the routing, power management, bandwidth management, radio interface and security are hot topics in MANET research. Although in this study, the researchers only focus on the routing protocols and security issues in MANET. The routing protocols in MANET may generally be categorized as: table-driven/proactive and source-initiated (demand-driven)/reactive. In proactive routing protocols such as the Optimized Link State Routing (OLSR), nodes obtain routes by periodic exchange of topology information. In reactive routing protocols such as the Ad-hoc on demand Distance Vector (AODV) protocol nodes find routes only when required. The overall goal of the security solutions for MANET is to provide security services including authentication, confidentiality, integrity, anonymity and availability to the mobile users. In order to achieve to this goal, the security solution should provide complete protection spanning the entire protocol stack. We can categories MANET security in 5 layers such as application layer, transport layer, network layer, link layer and physical layer. However, we only focus on the network layer which is related to security issues to protect the ad-hoc routing and forwarding protocols. From the security design perspective, the MANETs have no clear line of defense. Unlike wired networks that have dedicated routers each mobile node in an ad-hoc network may function as a router and forward packets for other peer nodes.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)
© Copyright: All rights reserved.

Recently, several research efforts introduced to counter against these malicious attacks. Most of the previous research has focused mainly on providing preventive schemes to protect the routing protocol in a MANET. Most of these schemes are based on key management or encryption techniques to prevent unauthorized nodes from joining the network. In general, the main drawback of these approaches is that they introduce a heavy traffic load to exchange and verify keys which is very expensive in terms of the bandwidth constraint for MANET nodes with limited battery and limited computational capabilities.

The MANET protocols are facing different routing attacks such as flooding, black hole; link withholding, link spoofing, replay, wormhole and colluding miss relay attack (Ford and Fulkerson, 1962; Chiang et al., 1997; Clausen and Jacquet, 2003).

Asymmetric links: Most of the wired networks rely on the symmetric links which are always fixed. But this is not a case with ad-hoc networks as the nodes are mobile and constantly changing their position within network. For example consider a MANET (Mobile Ad-hoc Network) where node B sends a signal to node A but this does not tell anything about the quality of the connection in the reverse direction.

Routing Overhead: In wireless adhoc networks, nodes often change their location within network. So, some stale routes are generated in the routing table which leads to unnecessary routing overhead.

Interference: This is the major problem with mobile ad-hoc networks as links come and go depending on the transmission characteristics, one transmission might interfere with another one and node might overhear transmissions of other nodes and can corrupt the total transmission.

Dynamic Topology: This is also the major problem with ad-hoc routing since the topology is not constant. The mobile node might move or medium characteristics might change. In ad-hoc networks, routing tables must somehow reflect these changes in topology and routing algorithms have to be adapted. For example in a fixed network routing table updating takes place for every 30sec. This updating frequency might be very low for ad-hoc networks.

Routing Protocols

Routing Tables

Each routing table entry contains the following information

- Destination
- Next hop
- Number of hops
- Destination sequence number
- Active neighbors for this route
- Expiration time for this route table entry

Expiration time, also called lifetime, is reset each time the route has been used. The new. Expiration time is the sum of the current time and a parameter called active route timeout. This parameter, also called route caching timeout, is the time after which the route is considered as invalid, and so the nodes not lying on the route determined by RREPs delete their reverse entries. If active route timeout is big enough route repairs will maintain routes. RFC 3561 defines it to 3 seconds.

Control messages

- *Routing request*

When a route is not available for the destination, a route request packet (RREQ) is flooded throughout the network. The RREQ contains the following fields: Source addresses, request ID, source sequence number, destination address, destination sequence number, hop count.

The request ID is incremented each time the source node sends a new RREQ, so the pair (source address, request ID) identifies a RREQ uniquely. On receiving a RREQ message each node checks the source address and the request ID. If the node has already received a RREQ with the same pair of parameters the new RREQ packet will be discarded. Otherwise the RREQ will be either forwarded (broadcast) or replied (unicast) with a RREP Message.

- ***Routing reply***

If a node is the destination, or has a valid route to the destination, it uncast a route. Reply message (RREP) back to the source. This message has the following format: **Source, destination address, destination sequence Number, hop count, life time.**

The reason one can unicast RREP back is that every node forwarding a RREQ message: Caches a route back to the source node.

- ***Route error***

All nodes monitor their own neighborhoods. When a node in an active route gets lost, a route error message (RERR) is generated to notify the other nodes on both sides of the link of the loss of this link.

- ***HELLO messages***

Each node can get to know its neighborhoods by using local broadcasts, so-called HELLO messages. Nodes neighbors' are all the nodes that it can directly communicate with. Each other. The HELLO messages will never be forwarded because they are broadcasted with TTL = 1. When a node receives a HELLO message it refreshes the corresponding lifetime of the neighbor information in the routing table. This local connectivity management should be distinguished from general topology management to optimize response time to local changes in the network.

I. Ad-hoc On-demand Multipath Distance Vector Routing (AOMDV)

Ad-hoc On-demand Multipath Distance Vector Routing (AOMDV) protocol is an extension to the AODV protocol for computing multiple loop-free and link disjoint paths. The routing entries for each destination contain a list of the next-hops along with the corresponding Hop counts. All the next hops have the same sequence number. This helps in keeping track of a route. For each destination, a node maintains the advertised hop count, which is defined as the maximum hop count for all the paths, which is used for sending route advertisements of the destination. Each duplicate route advertisement received by a node defines an alternate path to the destination. Loop freedom is assured for a node by accepting alternate paths to destination if it has a less hop count than the advertised hop count for that destination.

Because the maximum hop count is used, the advertised hop count therefore does not change for the same sequence number. When a route advertisement is received for a destination with a greater sequence number, the next-hop list and the advertised hop count are reinitialized.

AOMDV can be used to find node-disjoint or link-disjoint routes. To find node-disjoint routes, each node does not immediately reject duplicate RREQs. Each RREQs arriving via a different neighbor of the source defines a node-disjoint path. This is because nodes cannot be broadcast duplicate RREQs, so any two RREQs arriving at an intermediate node via a different neighbor of the source could not have traversed the same node. In an attempt to get multiple link-disjoint routes, the destination replies to duplicate RREQs, the destination only replies to RREQs arriving via unique neighbors. After the first hop, the RREPs follow the reverse paths, which are node disjoint and thus link-disjoint. The trajectories of each RREP may intersect at an intermediate node, but each takes a different reverse path to the source to ensure link disjointness. The advantage of using AOMDV is that it allows intermediate nodes to reply to RREQs, while still selecting disjoint paths. But, AOMDV has more message overheads during route discovery due to increased flooding and since it is a multipath routing protocol, the destination replies to the multiple RREQs those results are in longer overhead.

J. SYSTEM ANALYSIS

Proposed System

In this proposed system, a fixed threshold is used to identify the faults. Instead of fixed threshold, varying threshold considering dynamic changing networks can be set. The system can be compared with any of the multipath routing protocols. The additional delay due to probing might be reduced if the location of nodes after mobility especially destination node and adversaries can be predicted. This knowledge about nodes future location and behavior will be helpful in indoor environments and also in pervasive computing where mobile ad hoc networks plays a major role. Also this work with little variations along with service oriented architecture can be adapted for providing privacy and trust in pervasive computing.

SMT protocol provides security based on the security association between the end nodes. It is not able to overcome the compromised nodes attacks. The work presented in this paper has two phases. The first phase is to improve the security and reliability of data transmission in mobile ad hoc networks by providing secured routes.

The faults are identified and those links will be avoided in the data transmission phase. The current topological information will be gathered based on the network behavior such as transmission time, Probability of lost packets and correctly received – acknowledged packets and a threshold is set which is used in binary search probing. The nodes may exchange the current velocity vectors such as speed and direction to predict the location of the nodes. The spatial and temporal mining can be used to find the relative appropriateness of the location.

K. Problems with Routing In Mobile Ad-Hoc Networks

Asymmetric links: Most of the wired networks rely on the symmetric links which are always fixed. But this is not a case with ad-hoc networks as the nodes are mobile and constantly changing their position within network.

Routing Overhead: In wireless adhoc networks, nodes often change their location within network. So, some stale routes are generated in the routing table which leads to unnecessary routing overhead.

Interference: This is the major problem with mobile ad-hoc networks as links come and go depending on the transmission characteristics, one transmission might interfere with another one and node might overhear transmissions of other nodes and can corrupt the total transmission. Dynamic Topology: This is also the major problem with ad-hoc routing since the topology is not constant. The mobile node might move or medium characteristics might change. In ad-hoc networks, routing tables must somehow reflect these changes in topology and routing algorithms have to be adapted. For example in a fixed network routing table updating takes place for every 30sec. This updating frequency might be very low for ad-hoc networks.

II. METHODOLOGY

A. NS -2 INTRODUCTIONS

MOTIVATION FOR SIMULATION

- It is Cheap and does not require costly equipment
- The real thing isn't yet available
- Complex scenarios can be easily tested.
- Controlled experimental conditions
  - ➢ Reusability helps aid debugging
- Results can be quickly obtained
  - ➢ More ideas can be tested in a smaller timeframe.
- Disadvantages: Real systems are too complex to model

B. NS-2 FEATURES

- NS-2 is an object oriented discrete event simulator
  - ➢ Single thread of control: no locking or race conditions
  - ➢ Simulator maintains list of events and executes in sequence order i.e.) one event after another
- Back end is C++ event scheduler
  - ➢ Protocols mostly
  - ➢ Fast to run, more control
- Front end is OTCL
  - ➢ Creating scenarios, extensions to C++ protocols
  - ➢ Fast to write and change

C. NS-2 PROGRAMMING STRUCTURE

- Create the event scheduler
- Turn on tracing
- Create network topology

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

- Create transport connections
- Generate traffic
- Insert errors

D. PACKETS

It is the collection of data, whether header is called or not all header files where present in the stack registers.

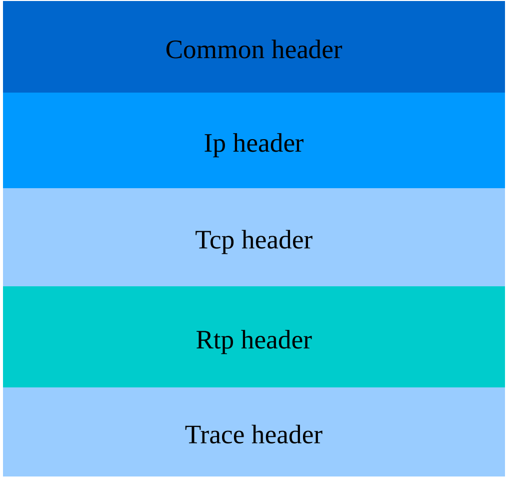


Figure 1: Packets Format

E. DESIGN

Modules Mobile Ad-hoc Networks

An ad-hoc network is a collection of wireless mobile hosts forming a temporary network without the aid of any stand-alone infrastructure or centralized administration. Mobile Ad-hoc networks are self-organizing and self-re-configuring multi-hop wireless networks where, the structure of the network changes dynamically. This is mainly due to the mobility of the nodes. Nodes in these networks utilize the same random access wireless channel, cooperating in a friendly manner to engaging themselves in multi-hop forwarding. The nodes in the network not only act as hosts but also as routers that route data to/from other nodes in network.

In mobile ad-hoc networks where there is no infrastructure support as is the case with wireless networks, and since a destination node might be out of range of a source node transmitting packets; a routing procedure is always needed to find a path so as to forward the packets appropriately between the source and the destination. Within a cell, a base station can reach all mobile nodes without routing via broadcast in common wireless networks. In the case of ad-hoc networks, each node must be able to forward data for other nodes. This creates additional problems along with the problems of dynamic topology which is unpredictable connectivity changes.

F. Byzantine Fault Tolerance

A Byzantine fault is an arbitrary fault that occurs during the execution of an algorithm by a distributed system. It encompasses both omission failures (e.g., crash failures, failing to receive a request, or failing to send a response) and commission failures (e.g., processing a request incorrectly, corrupting local state, and/or sending an incorrect or inconsistent response to a request.) When a Byzantine failure has occurred, the system may respond in any unpredictable way, unless it is designed to have Byzantine fault tolerance. The object of Byzantine fault tolerance is to be able to defend against *Byzantine failures*, in which components of a system fail in arbitrary ways.

Byzantine Attacks

Byzantine Faults

Reliable computer systems must handle malfunctioning components that give conflicting information to different parts of the system. This situation can be expressed abstractly in terms of a group of generals of the Byzantine army camped with their troops around an enemy city. Communicating only by messenger, the generals must agree upon a common battle plan. However, one or more of them may be traitors who will try to confuse the others. The problem is to find an algorithm to ensure that the loyal generals will reach agreement. It is shown that, using only oral messages, this problem is solvable if and only if more than two-thirds of the generals are loyal; so a single traitor can confound two loyal generals. With unforgettable written messages, the problem is solvable for any number of generals and possible traitors.

1) Black Hole Attack

It is the basic Byzantine Attack where the adversaries stop forwarding the data packets but still participates in the routing protocol correctly.

2) Flood Rushing Attack

If the adversaries reach some of its neighbors with its version of the flood packet before they receive a version through a legitimate route, then those nodes will ignore the legitimate version and forwards the adversarial version. This may result in continual inability to establish an adversarial free route even if authentication mechanisms are used.

3) Byzantine Worm Hole Attack

It is a more effective attack. The adversaries collude with each other and establish a tunnel (worm hole) between them. The adversaries can use the low cost appearance of the wormhole links in order to increase the probability of being selected as part of the route, and then attempt to disrupt the network by dropping all of the data packets. The Byzantine wormhole attack is an extremely strong attack that can be performed even if only two nodes have been compromised.

4) Byzantine Overlay Network Worm Hole Attack

A more general variant of the previous attack occurs when several nodes are compromised and form an overlay network. By tunneling packets through the overlay network, the adversaries make it appear to the routing protocol that they are all neighbors, which considerably increases their chances of being selected on routes. This is the strongest attack considered in this work. By forming an overlay network they will attack the network severely.

G. Multipath Data Transmission

The application of multipath techniques in mobile ad hoc networks seems natural, as multipath routing allows diminishing the effect of unreliable wireless links and the constantly changing topology. The on-demand multipath routing scheme is presented in as a multipath extension of dynamic source routing (DSR) in which alternate routes are maintained, so that they can be utilized when the primary one fails.

Another extension of DSR, multiple sources routing (MSR), proposes a weighted round-robin heuristic-based scheduling strategy among multiple paths in order to distribute load, but provides no analytical modeling of its performance. The split multipath routing (SMR), focuses on building and maintaining maximally disjoint paths, however, the load is distributed only in two routes per session.

H. Secure Data Transmission

Networks of thousands tiny sensor devices, which have low processing power, limited memory and energy, play roles for an economical solution to some challenging problems, such as, traffic monitoring, building safety, border security, habitat monitoring, tsunami alarm, medical emergency response and so on.

Undoubtedly security is an integral part of these applications. Authenticity of message is more important than confidentiality of data in this case. Consequently, if application does not consider adequate security measure then the intruder could find possible backdoor to feed highly abnormal information into the sensing devices and gain advantage of its own choice.

If the data in WSN are made available directly to the external party, then authentication and authorization of the external party must be ensured before allowing him/her to access data. We design a protocol for WSN that provides mutual authentication and secure data transmission between communicating entities. Secure data communication are done using the following techniques in this paper. These are,

Secured Route Discovery by SMT

Secured routes are provided by establishing an End-to-End security association between the source and the destination. This scheme won't consider the intermediate nodes that may exhibit arbitrary and malicious behavior. The source node S and destination node T negotiate a shared secret key- KS, T with the knowledge of each other's public key.

Security Provided by SMT under Various Attacks

1) Fake Reply

If M1 receives the request by S and reply a fake route to S, that false reply will be discarded by the source since M1 doesn't know KS,T and not able to produce a valid MAC.

2) Tampering Route Reply

If the malicious nodes M1 or M2 changes the route reply send by T, S will discard it as the modified reply won't integrate with the expected MAC of T.

3) Resource Consumption Attack

If the adversaries want to exhaust the network resources then they will replay the requests. On receiving the replayed requests, the nodes will drop the requests based on query identifiers.

4) Fabricated Route Requests

Malicious nodes after observing for some time the requests generated by source it will fabricate several queries with subsequent query identifiers. The goal is the intermediate nodes will store this numbers and drop out the legitimate requests sent by the source. This type of attack cannot be prevented by SMT.

5) Spoofing Attack

The nodes M1 and M2 may spoof an IP address and participate in the route requests. This attack cannot be identified and they can hide their location by masking.

6) Colluding nodes Attack

If the nodes colluded during both the request and reply phase, the source will accept the false route information. For example in M1 tunnels the route request to M2.M2 will broadcast the route request with route segment between M1 and M2 falsified. In the reverse direction, T will consider this path and send the route reply back to the source through M2.Reply is reverse tunneled by M2 to M1.By this a false path will be included between S and T.

Secured Data Communication of SMT

1) Active Path Sets(APS) and Message Transmission

A set of diverse, node disjoint multiple paths are selected by applying secured route discovery protocol. The set of paths used for current data transmission are known as Active Path Sets. The message is dispersed based on Robin's algorithm and is transmitted in multiple paths by dispersing it into pieces and after encoding. Redundancy ensures successful reconstruction of data even if some loss occurs due to malicious nodes or breakage of routes.

2) Robust Feedback Mechanism

Each dispersed piece is transmitted in different route and carries a Message Authentication Code and by that the integrity of the message and authenticity of the source is verified. After validation, the destination acknowledges every successful receipt. The feedback mechanism is also cryptographically protected and dispersed.

3) APS Adaptation

Successful receipt of ACKS indicates operational routes while missing ACK implies that the route is either broken or compromised. The paths are rated based on short term and long term rating. The routes are selected or discarded based on their rates.

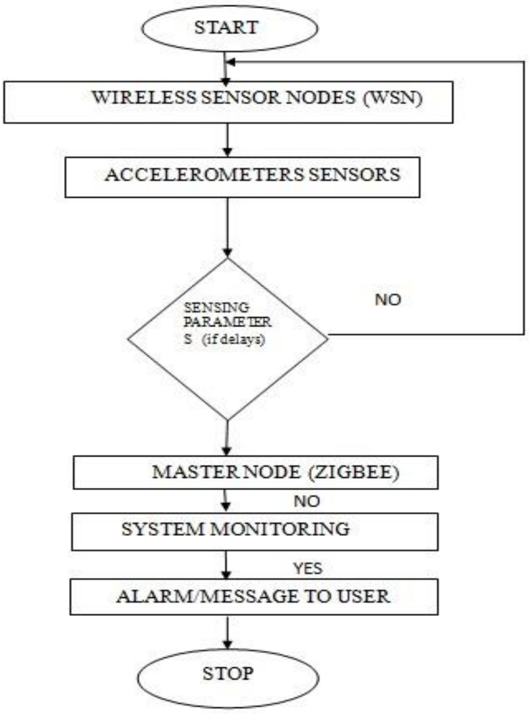


Figure 2: flowchart for wsn using in indoor environments

I. Reliability

The mesh network topology remedies this issue by using redundant communication paths to increase system reliability. In a mesh network, nodes maintain multiple communication paths back to the gateway, so that if one router node goes down, the network automatically reroutes the data through a different path. The mesh topology, while very reliable, does suffer from an increase in network latency because data must make multiple hops before arriving at the gateway.

J. Multicasting Delay

In multicast communication, there are two Quality of Service (QoS) parameters. The first is the end-to-end delay that is used to ensure that the messages transmitted by the source gateway can reach the destination gateways within a certain amount of time. The second is the multicast delay variation, defined as the difference between the maximum and the minimum multicast end-to-end delays on the multicast tree. It measures the consistency and fairness of receiving messages among all the destinations.

K. Binary search

The problem grerforms binary search between the smallest and largest weights. In each step, a breadth-first-search is used to check the existence of a path from the source point to the destination point using only line segments with weights that are larger than the search criterion. If a path exists, the criterion is increased to further restrict the lines considered in the next search iteration. To find the maximal support path, the weights of line segments of the Delaunay triangulation are assigned the lengths of the line segments.

L. Path selection

After a sender and a receiver start to exchange data packets, they build tables to keep traffic patterns. There is one table built by the sender and another one built by the receiver. The two tables have the same structure. Each table is composed of two fields: Packet identification number and time of action. Each time a packet is sent, the sender records the packet ID and the time. Each time a packet is received a receiver records the packet ID and the time.

M. Diagrams

Introduction to UML

The unified Modelling Language (UML) is a standard language for writing software blueprints. The UML may be used to visualize, specify, construct and document the artefacts of software-intensive system.

The goal of UML is to provide a standard notation that can be used by all object - oriented methods and to select and integrate the best elements .UML is itself does not prescribe or advice on how to use that notation in a software development process or as part of an object - design methodology. The UML is more than just bunch of graphical symbols. Rather, behind each symbol in the UML notation is well-defined semantics.

State Chart Diagram (Activity Diagram)

UML State chart is notation for describing the sequence of states an object goes through in response to external events. Objects have behaviour and state. The state of an object depends on its current activity or condition. A state chart diagram shows the possible states of the object ad the transitions that cause a change in state.

State chart describes the dynamic behaviour of an individual object as a number of states. A state is a condition satisfied by attributes of objects. Given a state, a transition represents a future state the object can move to and the conditions associated with the change of state.

A state is depicted by a rounded rectangle a transition is depicted by open arrows connecting two states. States are labelled with their names. A small solid black circle indicates the initial state and a circle surrounding the small solid circle indicates the final state.

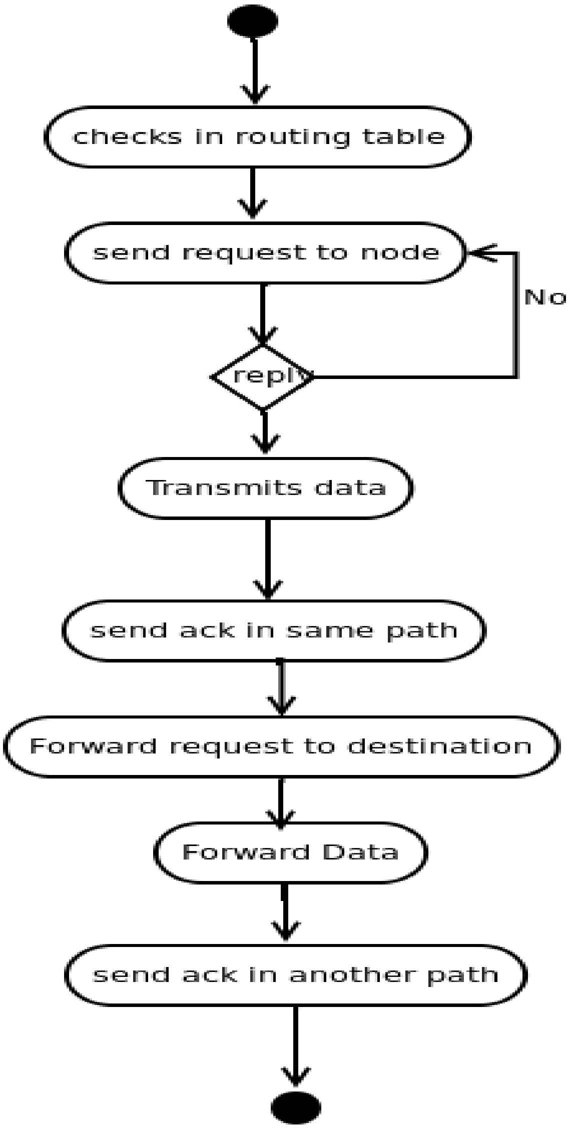


Figure 3: state chart diagram (activity diagram)

N. Feasibility Analysis

All projects are feasible which gives unlimited resources and infinite time. Before going further in to the steps of software development, the system analyst has to analyze whether the proposed system will be feasible for the organization and must identify the customer needs. The purpose of feasibility study is to determine whether the problem is worth solving.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

The success of a system is also lies in the amount of feasibility study done on it. Many feasibility studies have to be done on any system.

But there are three main feasibility tests to be performed. They are

- Operation feasibility
- Technical feasibility
- Economic feasibility

O. *Operational Feasibility*

During feasibility analysis operational feasibility study is a must. This is because; according to software engineering principles operational feasibility or in other words usability should be very high. A thorough analysis is done and found that the system is operational.

P. *Technical Feasibility*

The system analyst used to check the technical feasibility of the proposed system. Taking account of the hardware it is used for the system development, data storage, processing and output, makes the technical feasibility assessment. The system analyst has to check whether the company or user who is implementing the system has enough resource available for the smooth running of the application. Actually the requirements for this application are very less and thus it is technically feasible.

Economical Feasibility

Before we are going to further in to the development of the proposed system. The system analyst has to check the economic feasibility of the proposed system and the cost for running the system is composed with the cost benefit that can achieve by implementing the system. As in the case of Crypto Media development cost is not high, as it doesn't need any extra hardware and software. Thus the system is economically feasible.

System designs are the process of planning a new system document or altogether replace the old system. The purpose of the design phase is to plan a solution for the problem. The phase is the first step in moving from the problem domain to the solution domain. The design of the system is the critical aspect that affects the quality of the software. System design is also called top-level design. The design phase translates the logical aspects of the system into physical aspects of the system.

Implementation

System implementation is a stage in a stage in the project where the where the theoretical designs turned into working system. The most crucial stage is the user confidence that the new system will work effectively and efficiently.

The performance of reliability of the system was tested and it gained acceptance. The system was implemented successfully. Implementation is a process that means converting a new system into operation.

Proper implementation is essential to provide a reliable system to meet organization requirements. During the implementation stage a live demon was undertaken and made in front of end-users. Implementation is a stage of project when the system design is turned into a working system.

The stage consists of the following steps.

- ➢ Testing the developed program with sample data.
- ➢ Detection and correction of internal error.
- ➢ Testing the system to meet the user requirement.
- ➢ Feeding the real time data and retesting.
- ➢ Making necessary change as described by the user.

Input Design

Input Design is the part of overall system design that requires very careful attention. If the data going into the system is incorrect then the processing and the output will affect by these errors.

The inputs in the system are of three types:

- ➢ External : Prime inputs for the system
- ➢ Internal : User communication with the system
- ➢ Interactive : Inputs entered during a dialog with the computer

The above input types enrich the proposed system with numerous facilities that make it more advantageous in comparison with the exiting normal system. All the inputs entered are completely raw, initially, before being entered into a database, each of them available processing.

Output Design

Intelligent output design will improve systems relationships with the user and help in decision making. Outputs are also used to provide a permanent hardcopy of the results for latter consultations. The most important reason, which tempts the user to go for a new system is the output. The output generated by the system is often regarded as the criterion for evaluating the usefulness for the system.

Here the output requirements use to be predetermined before going to the actual system design.

The output design is based on the following:

- ➢ Determining the various outputs to be presented to the user.
- ➢ Differentiating between inputs to be displayed and those to be printed.
- ➢ The format for the presentation of the outputs.

NS2 CODE

TCL Script to Create Wireless Ad-hoc Network in NS2 Environment

```
set val(chan)       Channel/WirelessChannel  ;# channel type
set val(prop)       Propagation/TwoRayGround          ;#
radio-propagation model
set val(netif)      Phy/WirelessPhy/802_15_4          ;#
network interface type
set val(mac)          Mac/802_15_4            ;# MAC type
set val(ifq)         Queue/DropTail/PriQueue ;# interface
queue type
set val(ll)         ;# link layer type
set val(ant)         Antenna/Omni Antenna    ;# antenna
model
set val(ifqlen)     60             ;# max packet in ifq
set val(nn)         30           ;# number of mobilenodes
set val(rp)         AOMDV    ;# routing protocol
set val(a)          100                 ;#
set val(b)          100                 ;#
set val(nam)        out.nam             ;#
set val(traffic)    ftp                                    ;#
```

```
proc getCmdArgu {argc argv} {
        global val
        for {set c 0} {$c < $argc} {incr c} {
                set arg [lindex $argv $c]
                if {[string range $arg 0 0] != "-"} continue
                set name [string range $arg 1 end]
                set val($name) [lindex $argv [expr $c+1]]
        }
}
getCmdArgu $argc $argv
set applTime1           0.1    ;
set applTime2           0.4    ;
set applTime3           0.8    ;
set applTime4           12     ;
set sTime         16           ;

#---------------Initialize Global Variables-----------------#

set ns_              [new Simulator]
# Creating trace file and nam file

set tracefd     [open out.tr w]
$ns_ trace-all $tracefd
if { "$val(nam)" == "out.nam" } {
        set namtrace     [open ./$val(nam) w]
        $ns_ namtrace-all-wireless $namtrace $val(a) $val(b)
}
$ns_ puts-nam-traceall {# nam4wpan #}   ;# inform to
# For model 'TwoRayGround'
set dstnce(4m)  7.61113e-06
set dstnce(8m)  2.34381e-06
set dstnce(11m) 1.94278e-06
set dstnce(12m) 1.56908e-06
set dstnce(13m) 1.38527e-06
set dstnce(14m) 1.23774e-06
set dstnce(42m) 1.20574e-07
Phy/WirelessPhy set CSThresh_ $dstnce(16m)
Phy/WirelessPhy set RXThresh_ $dstnce(16m)
# set up topography object
set topo       [new Topography]
$topo load_flatgrid $val(a) $val(b)

# Create God
set god_ [create-god $val(nn)]
set chan_1_ [new $val(chan)]
# configure node

$ns_ node-config -adhocRouting $val(rp) \
                            -llType $val(ll) \
                            -macType $val(mac) \
                            -ifqType $val(ifq) \
                            -ifqLen $val(ifqlen) \
                            -antType $val(ant) \
                            -routerTrace OFF \
                            -macTrace ON \
                            -movementTrace OFF \
              -channel $chan_1_
for {set c 0} {$c < $val(nn) } {incr c} {
              set node_($c) [$ns_ node]
              $node_($c) random-motion 0                      ;
}
source Final.scn

# Setup traffic flow between nodes
```

```
proc cbrtraffic { srce dest interval strtTime } {
   global ns_ node_
   set udp($srce) [new Agent/UDP]
   eval $ns_ attach-agent \$node_($srce) \$udp($srce)
   set null($dest) [new Agent/Null]

   eval \$cbr($srce) attach-agent \$udp($srce)
   eval $ns_ connect \$udp($srce) \$null($dest)
   $ns_ at $strtTime "$cbr($srce) start"
}
proc poissontraffic { srce dest interval strtTime } {
   global ns_ node_
   set udp($srce) [new Agent/UDP]
   eval $ns_ attach-agent \$node_($srce) \$udp($srce)
   set null($dest) [new Agent/Null]
   eval $ns_ attach-agent \$node_($dest) \$null($dest)
   set expl($srce) [new Application/Traffic/Exponential]
   eval \$expl($srce) set packetSize_ 75
   eval \$expl($srce) set burst_time_ 0
   eval \$expl($srce) set idle_time_ [expr $interval*1000.0-
   70.0*8/260]ms ;# idle_time +   pkt_tx_time = interval
   eval \$expl($srce) set rate_ 260k
   eval \$expl($srce) attach-agent \$udp($srce)
   eval $ns_ connect \$udp($srce) \$null($dest)
   $ns_ at $strtTime "$expl($srce) start"
}
if { ("$val(traffic)" == "cbr") || ("$val(traffic)" == "poisson")
} {
   puts "\nTraffic: $val(traffic)"
   #Mac/802_15_4 wpanCmd ack4data on
   puts [format "Acknowledgement for data: %s"
[Mac/802_15_4 wpanCmd ack4data]]
   set lspeed 0.6ms
   set hspeed 1.6ms
   Mac/802_15_4 wpanNam PlaybackRate $lspeed
   $ns_ at [expr $applTime1+0.1] "Mac/802_15_4 wpanNam
PlaybackRate $hspeed"
   $ns_ at $applTime2 "Mac/802_15_4 wpanNam
PlaybackRate $lspeed"
      $ns_ at [expr $applTime3+0.1] "Mac/802_15_4
wpanNam PlaybackRate $hspeed"
   eval $val(traffic)traffic 18 7 0.1 $applTime1
   eval $val(traffic)traffic 11 5 0.1 $applTime2
   eval $val(traffic)traffic 4 3 0.1 $applTime3
   eval $val(traffic)traffic 29 26 0.1 $applTime4
   Mac/802_15_4 wpanNam FlowClr -p AOMDV -c tomato
   Mac/802_15_4 wpanNam FlowClr -p ARP -c aqua
   if { "$val(traffic)" == "cbr" } {
            set packetType cbr
   } else {
            set packetType exp
   }
   Mac/802_15_4 wpanNam FlowClr -p $packetType -s 18 -
d 6 -c red
   Mac/802_15_4 wpanNam FlowClr -p $packetType -s 11 -
d 5 -c aqua4
   Mac/802_15_4 wpanNam FlowClr -p $packetType -s 4 -d
3 -c cyan4
   Mac/802_15_4 wpanNam FlowClr -p $packetType -s 29 -
d 26 -c yellow4
```

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)
© Copyright: All rights reserved.

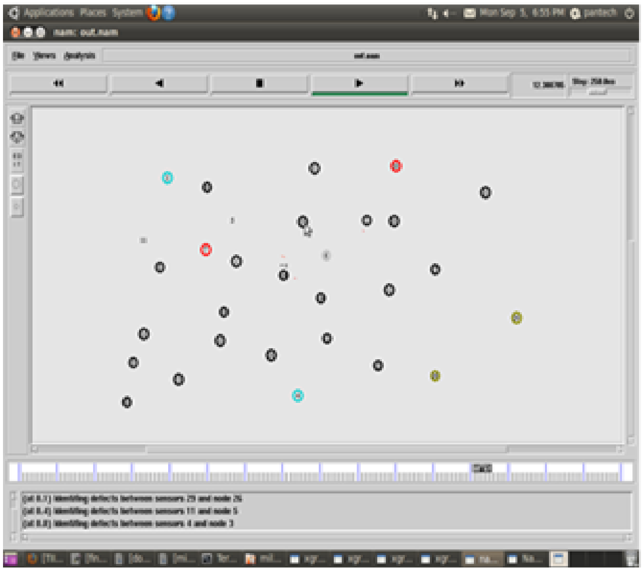
```
 $ns_ at $applTime4 "$node_(29) NodeClr yellow4"
   $ns_ at $applTime4 "$node_(26) NodeClr yellow4"
    $ns_ at $applTime4 "$ns_ trace-annotate \"(at
    $applTime1) $val(traffic) Identifing defects between
    sensors 29 and node 26\""
}
proc ftptraffic { srce dest strtTime } {
   global ns_ node_
   set tcp($srce) [new Agent/TCP]
   eval $ns_ connect \$tcp($srce) \$sink($dest)
   set ftp($srce) [new Application/FTP]
   eval \$ftp($srce) attach-agent \$tcp($srce)
   $ns_ at $strtTime "$ftp($srce) start"
}
if { "$val(traffic)" == "ftp" } {
   puts "\nTraffic: ftp"
   puts [format "Acknowledgement for data: %s"
[Mac/802_15_4 wpanCmd ack4data]]
   set lspeed 00.25ms
   set hspeed 01.6ms
   Mac/802_15_4 wpanNam PlaybackRate $lspeed

PlaybackRate $lspeed"
   $ns_ at [expr $applTime3+0.1] "Mac/802_15_4 wpanNam
PlaybackRate 6ms"
   $ns_ at $applTime4 "Mac/802_15_4 wpanNam
PlaybackRate $lspeed"
   $ns_ at [expr $applTime3+0.1] "Mac/802_15_4 wpanNam
PlaybackRate 11ms"
   ftptraffic 18 7 $applTime1
   ftptraffic 11 5 $applTime2
   ftptraffic 4 3 $applTime3
   ftptraffic 29 26 $applTime4
   Mac/802_15_4 wpanNam FlowClr -p AOMDV -c tomato
   Mac/802_15_4 wpanNam FlowClr -p ack -s 5 -d 11 -c
aqua4
   Mac/802_15_4 wpanNam FlowClr -p tcp -s 4 -d 3 -c
cyan4
   Mac/802_15_4 wpanNam FlowClr -p ack -s 3 -d 4 -c
cyan4
   Mac/802_15_4 wpanNam FlowClr -p tcp -s 29 -d 26 -c
yellow4
   Mac/802_15_4 wpanNam FlowClr -p ack -s 26 -d 29 -c
yellow4
   $applTime2) Identifing defects between sensors 11 and
node 5\""
   $ns_ at $applTime3 "$node_(4) NodeClr cyan3"
   $ns_ at $applTime3 "$node_(3) NodeClr cyan3"
    $ns_ at $applTime3 "$ns_ trace-annotate \"(at
    $applTime3) Identifing defects between sensors 4 and
    node 3\""
   $ns_ at $applTime1 "$node_(29) NodeClr yellow4"
   $ns_ at $applTime1 "$node_(26) NodeClr yellow4"
    $ns_ at $applTime1 "$ns_ trace-annotate \"(at
    $applTime1)  Identifing defects between sensors 29 and
    node 26\""
}
# defines the node size in nam

for {set c 0} {$c < $val(nn)} {incr c} {
             $ns_ initial_node_pos $node_($c) 2
}
# Tell nodes when the simulation ends
for {set c 0} {$c < $val(nn) } {incr c} {
    $ns_ at $sTime "$node_($c) reset";
}
$ns_ at $sTime "stop"
$ns_ at $sTime "puts \"\nNS EXITING...\""
$ns_ at $sTime "$ns_ halt"
BANDWIDTH
    exec xgraph BANDWIDTH &
    exec rm -f out-tcp.xgr
    exec awk -f graph2.awk out.tr > DELAY
    exec xgraph DELAY &
    exec rm -f out-tcp.xgr
    exec awk -f graph3.awk out.tr > LIFETIME
    exec xgraph LIFETIME &
    exec nam out.nam &
    }
}
puts "\nStarting Simulation..."
$ns_ run
```

III. DISCUSSION OF RESULTS

A. Nam

NAM provides a visual interpretation of the network topology created. Nam can be executed directly from Tcl script and it presents information such as throughput,

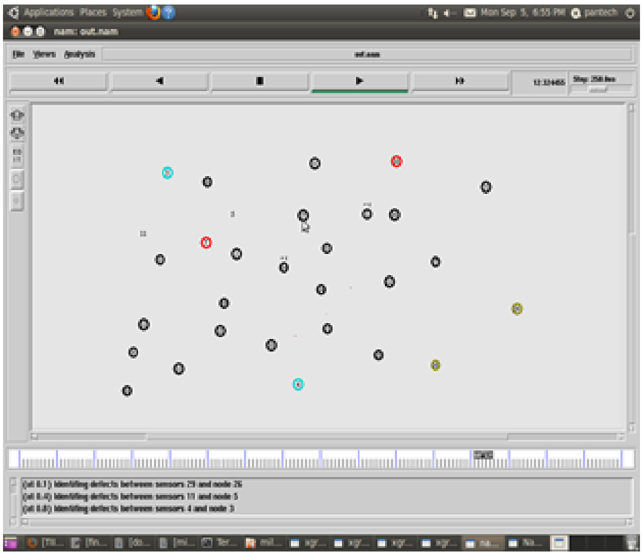
B. System Testing



(at 0.1) Identifing defects between sensors 29 and node 26
(at 0.4) Identifing defects between sensors 11 and node 5
(at 0.8) Identifing defects between sensors 4 and node 3

Figure 4: AOMDV WITHOUT MALICIOUS

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)
© Copyright: All rights reserved.



(at 0.11) Identifying defects between sensors 29 and node 26
(at 0.4) Identifying defects between sensors 11 and node 5
(at 0.8) Identifying defects between sensors 4 and node 3

Figure 5: AOMDV WITH MALICIOUS

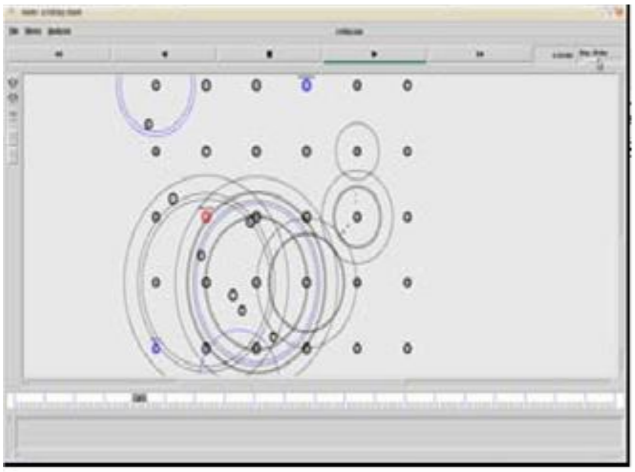


Figure 6: Multipath Routing

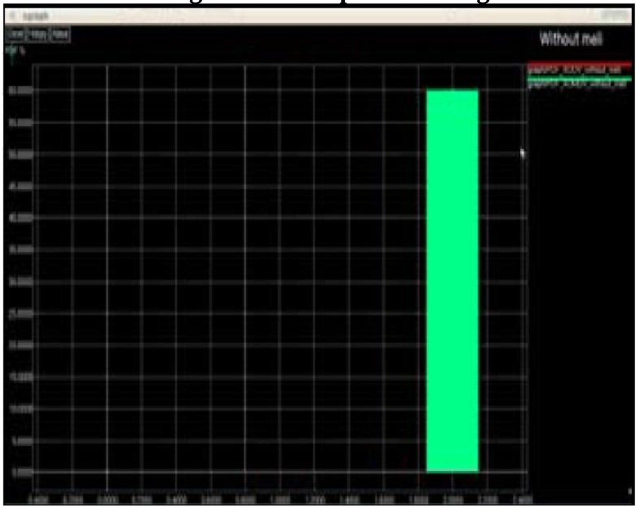


Figure 7: Without Malicious Environment

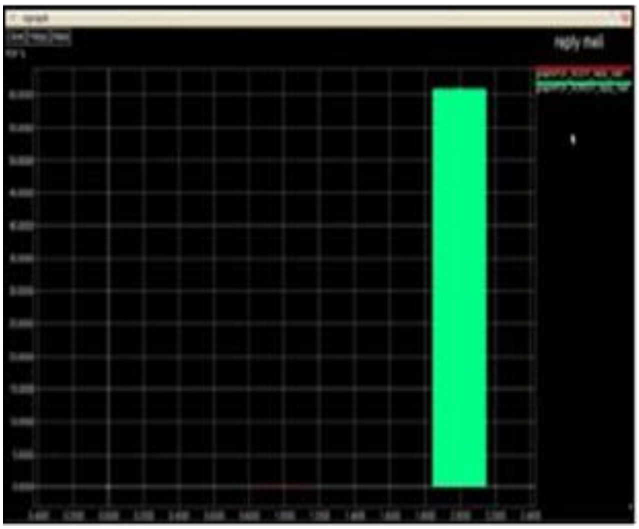


Figure 8: With Reply Malicious

X-Graph

X-Graph is an X-Windows application that includes:

- Interactive plotting and graphing
- Animation and derivatives

To use X-Graph in NS2 the executable can be called within a tcl script. Then It loads a graph displaying the information visually of the trace file produced from the simulation.

Figure 9: AOMDV-LIFETIME

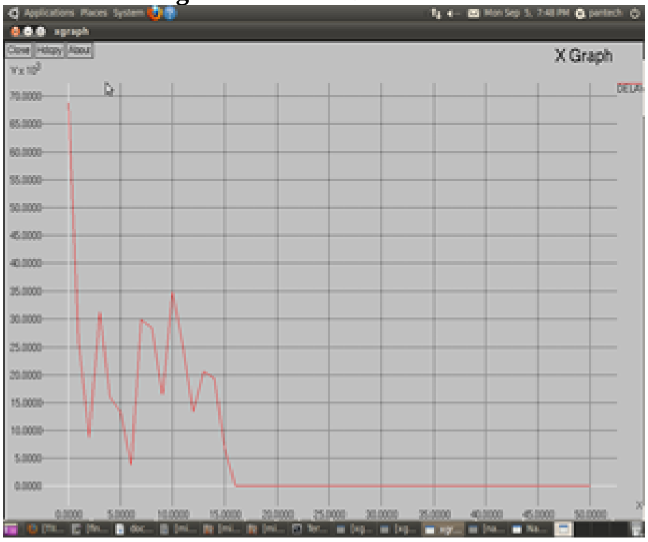


Figure 10: AOMDV-DELAY

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

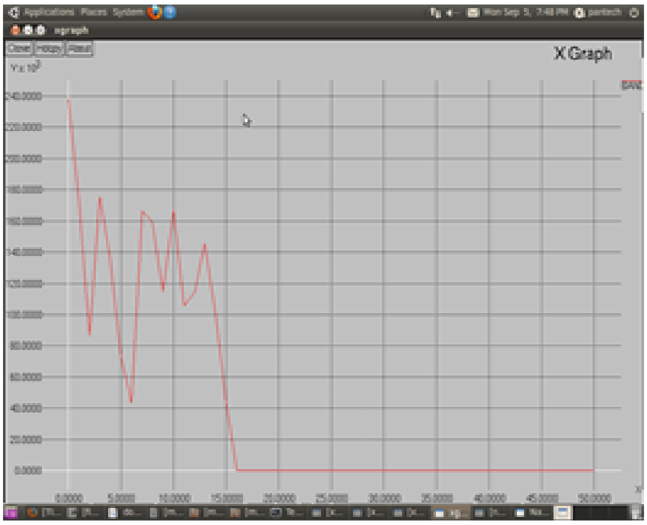
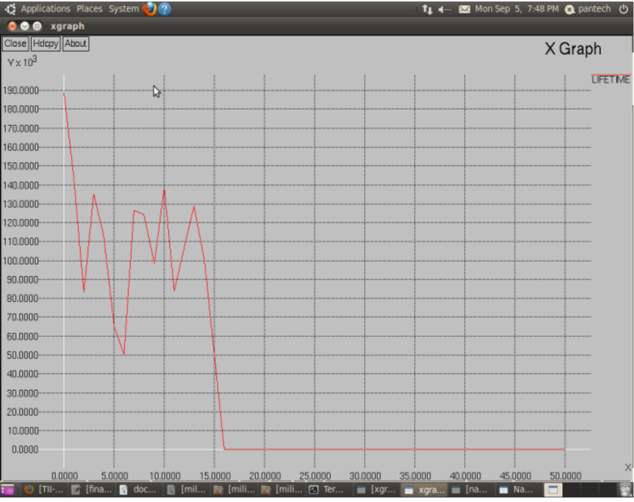


Figure 11: AOMDV- BANDWIDTH

IV. CONCLUSIONS

Simulations were conducted using the NS2 network simulator. Nodes in the network were configured to use 802.11 radios with a bandwidth of 2 Mbps and a nominal range of 250 m. In order to simulate most of the proposed Byzantine attacks in NS2, a protocol independent Byzantine attack simulation module was developed. This module provides the capability to simulate the black hole, Byzantine wormhole, and Byzantine overlay network wormhole attacks without modifying the routing protocol. It was not possible to simulate the flood rushing attack using this technique because it requires timing changes in the routing protocol code. The module is implemented as part of the NS2 Link Layer (LL) object which lies directly below the Routing Agent and directly above the MAC layer.



In our network all wireless nodes are deployed in random manner. And we are creating one source and destination. To avoid data hacking in our network we are going to implement multipath communication in our network. Here we are considering our communication path is changeable even path or node is node failed. So data is sending through different paths, it provide high security than single path, this result is shown in same Nam window in the period of 23-30 second. This model is proving security but not Qos, so we have to improve this model to provide high quality of service. Due to need of quality of service we are going to implement energy efficient transmission in our network model, due to some technical reason we can't so that QoS parameter in NAM window, but we can show in X-graph.

Another type of result is X-graph, using x-graph we can compare different networks. Our graph shows multipath communication which is marked as green colour. Each Graph have same x and y axis value parameter time in seconds and throughput in Kbits respectively.

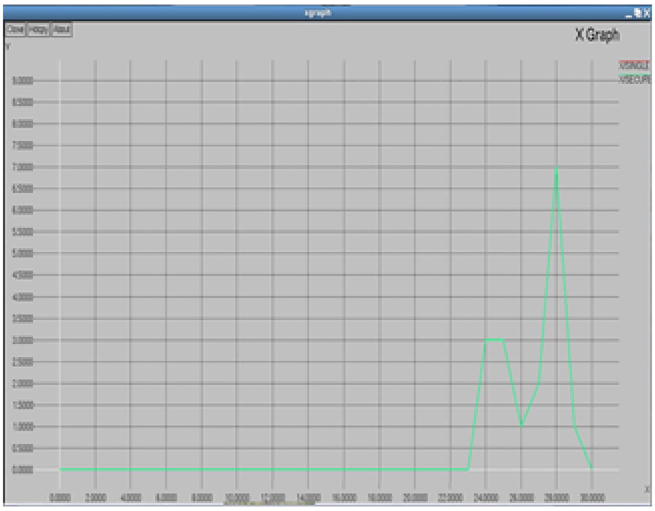


Figure 12: Graph 1

Our basic multipath communication is started at the time of 23rd sec is marked as green colour line.

REFRENCES

1. Reza Curtmola Cristina Nita-Rotaru, "BSMR: Byzantine- Resilient Secure Multicast Routing in Multi-hop Wireless Networks", IEEE Transactions on Mobile Computing, vol. 8, Issue. 4, pp. 445 - 459, February 2009.
2. A.Tsirigos and Z.J.Hass (2004), "Analysis of multi path routing, Part 1: The effects on the packet delivery ratio" IEEE Transactions on Wireless Communication., vol.3, no.2, pp: 500- 511.
3. Banner, R. Orda, A, "Multipath Routing Algorithms for Congestion Minimization". This paper appears in: Networking, IEEE/ACM Transactions on Publication Date: April 2007 Volume: 15, Issue: 2, on page(s): 413-424.
4. Jun Peng, Biplab Sikdar and Liang Cheng (2009) "Multicasting with Localized Control in Wireless Ad Hoc Networks" IEEE Transaction on Mobile Computing.
5. Papadimitratos, P. Haas, Z.J, "Secure data communication in mobile ad-hoc networks" , This paper appears in: Selected Areas in Communications, IEEE Journal on Publication Date: Feb. 2006,Volume: 24, Issue: 2,On page(s): 343- 356.
6. Banner , R. Orda, A. "Multipath Routing Algorithms for Congestion Minimization"Conference version in Proc. IFIP Networking 2005.
7. Papadimitratos, P. Haas, Z.J, Sirer, E, G."Path Set Selection in Mobile Ad Hoc Networks". June 09 - 11, 2002. Pages 1 - 11.

Author's Profile

Seyed Amin Ahmadi Olounabadi, Ph.D. scholar student in Computer Science and Engineering, Dept. of Computer Science and Engineering, Jawaharlal Nehru Technological University Hyderabad (JNTUH), Hyderabad, Telangana , India, ,Research interest: Network and Network Security, IT, Network Management.

Prof. Avula. Damodaram, Vice-Chancellor Sri Venkateswara University, Tirupati, Andhra Pradesh, India. Faculty of Computer Science & Engineering at JNTU, Hyderabad, Research interests: include Image Processing, Pattern Recognition, Network Security, Steganography and Digital Watermarking.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

Prof. V Kamakshi Prasad, Head of Computer Science and Engineering Department, Jawaharlal Nehru Technological University Hyderabad (JNTUH), Hyderabad, Telangana, India. Research interests: Speech Recognition and Processing, Image processing, Pattern Recognition, Data Mining, Ad-hoc networks, Computer Graphics.

Mahdi Hosseini, Ph.D. scholar student in Structural Engineering, Dept. of Civil Engineering, Jawaharlal Nehru Technological University Hyderabad (JNTUH), Hyderabad, Telangana ,India, Research interest: Structural Engineering, Structural Dynamics ,Structural Optimization, Structural design, Reinforced Concrete Structures, Earthquake Engineering.