International Journal of Computational Research and Development (IJCRD)
Impact Factor: 4.775, ISSN (Online): 2456 - 3137
(www.dvpublication.com) Volume 1, Issue 2, 2016

ACTIVE TRUST: SECURE AND TRUSTABLE ROUTING IN WIRELESS SENSOR NETWORKS

T. Mashboob Abdul Karim* & S. Gowsalya**
* PG Scholar, Maharaja Prithvi Engineering College, Avinashi, Tamilnadu
** Assistant Professor & HOD, Department of Electronics and Communication Engineering, Maharaja Prithvi Engineering College, Avinashi, Tamilnadu

Cite This Article: T. Mashboob Abdul Karim & S. Gowsalya, "Active Trust: Secure and Trustable Routing in Wireless Sensor Networks", International Journal of Computational Research and Development, Volume 1, Issue 2, Page Number 111-119, 2016.

Abstract:

Wireless sensor networks (WSNs) are increasingly being deployed in security-critical applications. Because of their inherent resource-constrained characteristics, they are prone to various security attacks, and a black hole attack is a type of attack that seriously affects data collection. To conquer that challenge, an active detection-based security and trust routing scheme named Active Trust is proposed for WSNs. The most important innovation of Active Trust is that it avoids black holes through the active creation of several detection routes to quickly detect and obtain nodal trust and thus improve the data route security. More importantly, the generation and distribution of detection routes are given in the Active Trust scheme, which can fully use the energy in non-hotspots to create as many detection routes as needed to achieve the desired security and energy efficiency. Both comprehensive theoretical analysis and experimental results indicate that the performance of the Active Trust scheme is better than that of previous studies. Active Trust can significantly improve the data route success probability and ability against black hole attacks and can optimize network lifetime.

Introduction:

In many Wireless Sensor Network (WSN) applications, sensors are spatially distributed in a finite area so as to monitor physical or environmental conditions, such as pressure, humidity, temperature, etc. and also to transmit the sensed data to a base station cooperatively. In addition, at times, a set of target points has to be monitored in a given area. On the one hand, to provide a deterministic quality of service guarantees, every point of interest should be monitored by at least one sensor at all times. On the other hand, the energy consumption of sensors should be minimized since in most cases sensors are battery powered. Therefore sensors should have their power supplies turned off when they are not in use top reserve energy. Due to this limitation, a critical issue becomes how to prolong the lifetime of WSNs while also assuring the service quality of coverage. Thus, research on energy efficient sensor coverage problem has been extensively investigated in the literature. For a typical target coverage problem in WSNs, the network lifetime is defined as the time duration that all the target points are monitored. As pointed out in, network lifetime can be prolonged by alternating the working modes of sensors between settings of "on" and "off". In other words, schedule the entire time duration into a number of rounds and only turn-on the power supplies of a subset of sensors to monitor the target points in each round. Supposing that, all the sensors can work two time units, then by alternating the "on" and "off" modes, we can monitor all the target points for three time units.

Wireless Sensor Network:

The WSN is built of "nodes" – from a few to several hundreds or even thousands, where each node is connected to one (or sometimes several) sensors. Each such sensor network node has typically several parts: a radio transceiver with an internal antenna or connection to an external antenna, a microcontroller, an electronic circuit for interfacing with the sensors and an energy source, usually a battery or an embedded form of energy harvesting. A sensor node might vary in size from that of a shoebox down to the size of a grain of dust, although functioning "motes" of genuine microscopic dimensions have yet to be created. The cost of sensor nodes is similarly variable, ranging from a few to hundreds of dollars, depending on the complexity of the individual sensor nodes. Size and cost constraints on sensor nodes result in corresponding constraints on resources such as energy, memory, computational speed and communications bandwidth. The topology of the WSNs can vary from a simple star network to an advanced multi-hop wireless mesh network. The propagation technique between the hops of the network can be routing or flooding. In computer science and telecommunications, wireless sensor networks are an active research area with numerous workshops and conferences arranged each year, for example IPSN, Sense's, and EWSN.

Characteristics:

The main characteristics of a WSN include,

- ✓ Power consumption constraints for nodes using batteries or energy harvesting
- ✓ Ability to cope with node failures (resilience)
- ✓ Mobility of nodes
- ✓ Heterogeneity of nodes
- ✓ Scalability to large scale of deployment
- ✓ Ability to withstand harsh environmental conditions

- ✓ Ease of use
- ✓ Cross-layer design

Cross-layer is becoming an important studying area for wireless communications. In addition, the traditional layered approach presents three main problems:

- ✓ Traditional layered approach cannot share different information among different layers which leads to each layer not having complete information. The traditional layered approach cannot guarantee the optimization of the entire network.
- ✓ The traditional layered approach does not have the ability to adapt to the environmental change.
- ✓ Because of the interference between the different users, access conflicts, fading, and the change of environment in the wireless sensor networks, traditional layered approach for wired networks is not applicable to wireless networks.

So the cross-layer can be used to make the optimal modulation to improve the transmission performance, such as data rate, energy efficiency, QoS (Quality of Service), etc.. Sensor nodes can be imagined as small computers which are extremely basic in terms of their interfaces and their components. They usually consist of a processing unit with limited computational power and limited memory, sensors or MEMS (including specific conditioning circuitry), a communication device (usually radio transceivers or alternatively optical), and a power source usually in the form of a battery. Other possible inclusions are energy harvesting modules, secondary ASICs, and possibly secondary communication interface (e.g. RS-232 or USB) I is shown in the Fig.1.1. The base stations are one or more components of the WSN with much more computational, energy and communication resources. They act as a gateway between sensor nodes and the end user as they typically forward data from the WSN on to a server. Other special components in routing based networks are routers, designed to compute, calculate and distribute the routing tables.

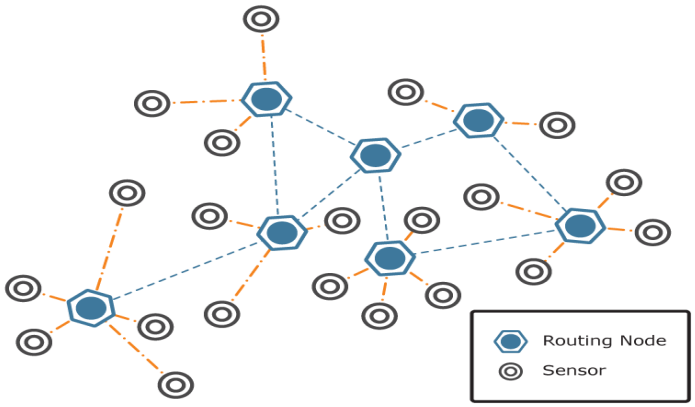

Figure 1.1: Sensor Network

Sensor Node:

A sensor node, also known as a mote (chiefly in North America), is a node in a sensor network that is capable of performing some processing, gathering sensory information and communicating with other connected nodes in the network. A mote is a node but a node is not always a mote. Although wireless sensor nodes have existed for decades and used for applications as diverse as earthquake measurements to warfare, the modern development of small sensor nodes dates back to the 1998 Smart dust project and the NASA Sensor Webs Project One of the objectives of the Smart dust project was to create autonomous sensing and communication within a cubic millimeter of space. Though this project ended early on, it led to many more research projects. They include major research centers in Berkeley NEST and CENS. The researchers involved in these projects coined the term mote to refer to a sensor node. The equivalent term in the NASA Sensor Webs Project for a physical sensor node is pod, although the sensor node in a Sensor Web can be another Sensor Web itself. Physical sensor nodes have been able to increase their capability in conjunction with Moore's Law. The chip footprint contains more complex and lower powered microcontrollers. Thus, for the same node footprint, more silicon capability can be packed into it. Nowadays, motes focus on providing the longest wireless range (dozens of km), the lowest energy consumption and the easiest development process for the user.

Sensors:

Sensors are hardware devices that produce a measurable response to a change in a physical condition like temperature or pressure. Sensors measure physical data of the parameter to be monitored. The continual analog signal produced by the sensors is digitized by an analog-to-digital converter and sent to controllers for further processing. A sensor node should be small in size, consume extremely low energy, operate in high volumetric densities, be autonomous and operate unattended, and be adaptive to the environment. As wireless sensor nodes are typically very small electronic devices, they can only be equipped with a limited power source of less than 0.5-2 ampere-hour and 1.2-3.7 volts. Sensors are classified into three categories: passive, omni-directional sensors; passive, narrow-beam sensors; and active sensors. Passive sensors

sense the data without actually manipulating the environment by active probing. They are self-powered; that is, energy is needed only to amplify their analog signal. Active sensors actively probe the environment, for example, a sonar or radar sensor, and they require continuous energy from a power source. Narrow-beam sensors have a well-defined notion of direction of measurement, similar to a camera. Omni-directional sensors have no notion of direction involved in their measurements. The overall theoretical work on WSNs works with passive, omni-directional sensors. Each sensor node has a certain area of coverage for which it can reliably and accurately report the particular quantity that it is observing. Several sources of power consumption in sensors are: signal sampling and conversion of physical signals to electrical ones, signal conditioning, and analog-to-digital conversion. Spatial density of sensor nodes in the field may be as high as 20 nodes per cubic meter.

Routing Node:

Routing is the process of selecting best paths in a network. In the past, the term routing also meant forwarding network traffic among networks. However, that latter function is better described as forwarding. Routing is performed for many kinds of networks, including the telephone network (circuit switching), electronic data networks (such as the Internet), and transportation networks. This article is concerned primarily with routing in electronic data networks using packet switching technology. In packet switching networks, routing directs packet forwarding (the transit of logically addressed network packets from their source toward their ultimate destination) through intermediate nodes. Intermediate nodes are typically network hardware devices such as routers, bridges, gateways, firewalls, or switches. General-purpose computers can also forward packets and perform routing, though they are not specialized hardware and may suffer from limited performance. The routing process usually directs forwarding on the basis of routing tables, which maintain a record of the routes to various network destinations. Thus, constructing routing tables, which are held in the router's memory, is very important for efficient routing. Most routing algorithms use only one network path at a time. Multipath routing techniques enable the use of multiple alternative paths In case of overlapping/equal routes, algorithms consider the following elements to decide which routes to install into the routing table (sorted by priority):

- ✓ Prefix-Length: where longer subnet masks are preferred (independent of whether it is within a routing protocol or over different routing protocol)
- ✓ Metric: where a lower metric/cost is preferred (only valid within one and the same routing protocol)
- ✓ Administrative distance: where a route learned from a more reliable routing protocol is preferred (only valid between different routing protocols)

Routing, in a more narrow sense of the term, is often contrasted with bridging in its assumption that network addresses are structured and that similar addresses imply proximity within the network. Structured addresses allow a single routing table entry to represent the route to a group of devices. In large networks, structured addressing (routing, in the narrow sense) outperforms unstructured addressing (bridging). Routing has become the dominant form of addressing on the Internet. Bridging is still widely used within localized environments. The applications are provided below. NP-hard problems are often tackled with rules-based languages in areas such as:

- ✓ Configuration
- ✓ Data mining
- ✓ Selection
- ✓ Diagnosis
- ✓ Process monitoring and control
- ✓ Scheduling
- ✓ Planning
- ✓ Rosters or schedules
- ✓ Tutoring systems
- ✓ Decision support
- ✓ Phylogenetics
- ✓ Routing/vehicle routing

Need for the Study:

The Trusted Environment is a secure area of the main processor. It guarantees code and data loaded inside to be protected with respect to confidentiality and integrity. The trust as an isolated execution environment provides security features such as isolated execution, integrity of Trusted Applications along with confidentiality of their assets. In general terms, the trusted offers an execution space that provides a higher level of security than a rich mobility and more functionality than a secure element.

Objectives of the Study:

The adversary compromises a node and drops all packets that are routed via this node, resulting in sensitive data being discarded or unable to be forwarded to the sink. Because the network makes decisions depending on the nodes' sensed data, the consequence is that the network will completely fail and, more

seriously, make incorrect decisions. Therefore, how to detect and avoid BLA is of great significance for security in WSNs.

Literature Survey:

Wireless Sensor Networks (WSNs) are emerging as one of the prevailing technologies of the future due to their wide range of applications in military and civilian domains. Due to their operating nature, they are often unattended and hence prone to different types of novel attacks. For instance, an adversary could capture nodes, acquiring all the information stored there in sensors are commonly assumed to not be tamper-proof. Therefore, an adversary may replicate captured sensors and deploy them in the network to launch a variety of malicious activities.

Per-Hop Acknowledgement (Phack):

First the PHACK scheme proposed in this paper has better ability for detecting and identifying suspect nodes. In the PHACK scheme, each intermediate node along a forwarding path is responsible for generating acknowledgements (ACK) to the source node for each packet received. The difference from previous research is that each confirmation is routed to the sink along a different path. One benefit is that the probability of the confirmation information reaching the sink successfully can be improved, as this approach can avoid the risk of a single routing failure for the case in which all ACK packets are returned to the source node along the same data forwarding path as in previous work; additionally, because each acknowledgement is independently routed along different paths, this scheme has a higher ability for detecting and identifying suspect nodes. Though the acknowledgement can be returned by each intermediate node in this scheme, the network lifetime was not affected in comparison to other schemes. This is because, in the process of data collection in a wireless sensor network, the nodes near the sink consume more energy due to the increased amount of data that can be forwarded to the sink from the nodes far from the sink, called hotspots. After the premature death of nodes near the sink area forms an energy hole, the data from nodes in the distance cannot be routed to the sink, which causes the entire network to die in advance, with more than 90%of the total energy being unable to be used. This means that if the residual energy can be used effectively, it not only does not affect the network lifetime but also can improve the ability for detecting the selective forwarding attack. Theoretical analysis has proved that although the PHACK scheme increases the energy consumption in the peripheral area, it does not increase the energy consumption in hotspot areas, so the network lifetime in the PHACK scheme is not less than that of other acknowledgement-based schemes, but the performance can be improved significantly regarding the detection accuracy and effectiveness. Second, the PHACK scheme can not only effectively detect the selective forwarding attack, but it also can recover from routing failure, as the attacked data can be rerouted to the sink rapidly along an alternative routing path that excludes the suspect nodes. In the previous research, most of the selective forwarding attack detection schemes only have a detection function. In the PHACK scheme, suspect nodes can be accurately identified, and thus the dropped data can be rerouted from the nodes nearest to the sink to the sink along a routing path that bypasses the suspect nodes; this leads to the quick recovery of the routing data at the lowest cost.

The Network Model:

First we consider a wireless sensor network consisting of a large number of sensor nodes that are uniformly and randomly scattered in a circle network; the network radius is R, with the density of nodes equal to, and nodes do not move after being deployed. On detecting an event, a sensor node will generate messages, and those messages must be transmitted to the sink node. However, the routing method used for the data packets is determined based on the requirements of the application, such as the shortest routing approach. Second the attacker is considered to have strong intelligence. It obtains legal identification through compromising a sensor node. After that, the attacker can launch various attacks, such as dropping data packets or ACK messages or altering messages with a certain probability. The aim of attackers is to try not to expose themselves and to cause the greatest harm to the network. At the same time, the attackers can also collude to launch attacks. Third a message authentication code is adopted in the PHACK scheme, which provides assurance to the recipient that the message came from the expected sender and has not been altered in transit. Therefore, in this paper, if there are no special instructions, all packets or messages adopt the message authentication code technology.

Density Control:

Nodes close to the sink tend to deplete their energy budget faster than the other sensors. Based on this fact, suboptimal energy efficiency is possible if more nodes are deployed around the sink. In this way, more nodes participate in relaying data for the other part of the network and the lifetime of the network is prolonged. This is the so called non-uniform node distribution method. For example, in, the circular network is divided into M adjacent coronas with the same width of r, where r is the transmitting radius of sensor node.

Adjustable Transmission Ranges:

The transmission and sensing range is adjustable. For example, Berkeley Motes has 100 transmission power levels. And as the energy consumption is directly proportional to the a power of communication distance, using smaller transmission radius in the hotspots near the sink and larger transmission radius in the regions far away from the sink, a balanced energy consumption and longer network lifetime could be achieved.

System Analysis:

Existing System:

- ✓ Single-path routing is a simple routing protocol but is easily blocked by the attacker.
- ✓ Therefore, the most natural approach is via multi-path routing to the sink.
- ✓ Even if there is an attack in some route, the data can still safely reach the sink.
- ✓ Multi-path routing protocols can be classified into two classes depending on whether the data packet is divided. One is multi-path routing without share division.
- ✓ The other is multi-path routing with share division, i.e., the packet is divided into shares, and different shares reach the destination via different routes

Proposed System:

- ✓ In this paper, we have proposed a novel security and trust routing scheme based on active detection, and it has the following excellent properties High successful routing probability, security and scalability.
- ✓ The Active Trust scheme can quickly detect the nodal trust and then avoid suspicious nodes to quickly achieve a nearly 100% successful routing probability High energy efficiency.
- ✓ The Active Trust scheme fully uses residue energy to construct multiple detection routes.
- ✓ The theoretical analysis and experimental results have shown that our scheme improves the successful routing probability by more than 3 times, up to 10 times in some cases.

System Architecture:

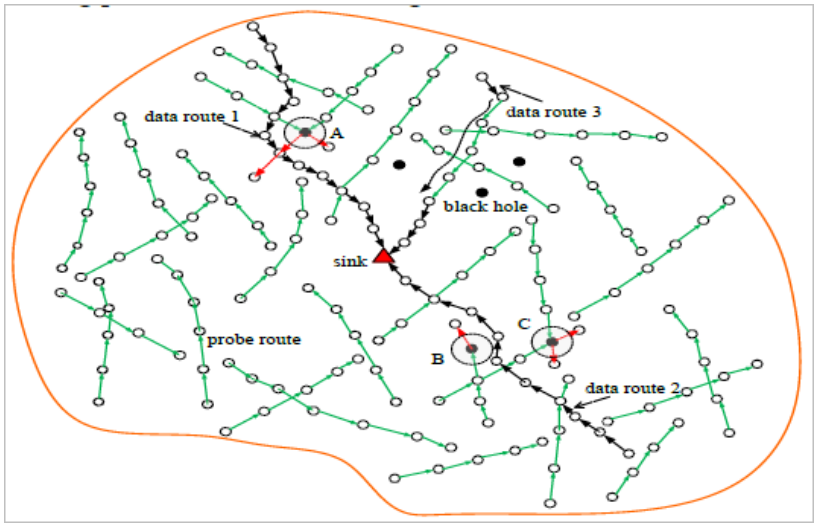

Figure 3.1: Illustration of the Active Trust scheme

However, the current trust-based route strategies face in obtaining trust. However, obtaining the trust of a node is very difficult, and how it can be done is still unclear. Energy efficiency. Because energy is very limited in WSNs, in most research, the trust acquisition and diffusion have high energy consumption, which seriously affects the network lifetime. (3) Security. Because it is difficult to locate malicious nodes, the security route is still a challenging issue. Thus, there are still issues worthy of further study. Security and trust routing through an active detection route protocol is proposed in this paper. The main innovations are as follows. The Active Trust scheme is shown in the Fig.3.1.is the first routing scheme that uses active detection routing to address BLA. The most significant difference between Active Trust and previous research is that we create multiple detection routes in regions with residue energy; because the attacker is not aware of detection routes, it will attack these routes and, in so doing, be exposed. In this way, the attacker's behavior and location, as well as nodal trust, can be obtained and used to avoid black holes when processing real data routes. To the best of our knowledge, this is the first proposed active detection mechanism in WSNs. The Active Trust route protocol has better energy efficiency. Energy is very precious in WSNs, and there will be more energy consumption if active detection is processed. Therefore, in previous research, it was impossible to imagine adopting such high-energy-consumption active detection routes. However, we find it possible after carefully analyzing the energy consumption in WSNs. Research has noted that there is still up to 90% residue energy in WSNs when the network has died due to the "energy hole" phenomenon. Therefore, the Active Trust scheme takes full advantage of the residue energy to create detection routes and attempts to decrease energy consumption in hotspots (to improve network lifetime). Those detection routes can detect the nodal trust without decreasing lifetime and thus improve the network security. According to theoretical analysis and experimental results, the energy efficiency of the Active Trust scheme is improved more than 2 times compared to previous routing schemes, including shortest routing, multi-path routing. The Active Trust scheme has better security performance. Compared with previous research, nodal trust can be obtained in Active Trust. The route

is created by the following principle. First, choose nodes with high trust to avoid potential attack, and then route along a successful detection route. Through the above approach, the network security can be improved. Through our extensive theoretical analysis and simulation study, the Active Trust routing scheme proposed in this paper can improve the success routing probability by 1.5 times to 6 times and the energy efficiency by more than 2 times compared with that of previous researches.

System Requirements:

Hardware Requirements:

- ✓ Processor - Pentium –III
- ✓ Speed - 1.1 Ghz
- ✓ RAM - 256 MB(min)
- ✓ Hard Disk - 20 GB
- ✓ Key Board - Standard Windows Keyboard
- ✓ Mouse - Two or Three Button Mouse
- ✓ Monitor - SVGA

Software Requirements:

- ✓ Operating System : LINUX
- ✓ Tool : Network Simulator-2
- ✓ Front End : O TCL (Object Oriented Tool Command Language)

Software Description:

NS2 Structure Introduction:

NS2 is an object-oriented simulator, written in C++, with a Tcl interpreter as a front-end. The simulator supports a class hierarchy in C++ (also called the compiled hierarchy), and a similar class hierarchy within the Tcl interpreter (also called the interpreted hierarchy). The two hierarchies are closely related to each other; from the user's perspective, there is a one-to-one correspondence between a class in the interpreted hierarchy and one in the compiled hierarchy. NS2 uses two languages because it has two different kinds of things it needs to do: Detailed simulations of protocols require a systems programming language which can efficiently manipulate bytes, packet headers, and implement algorithms that run over large data sets.

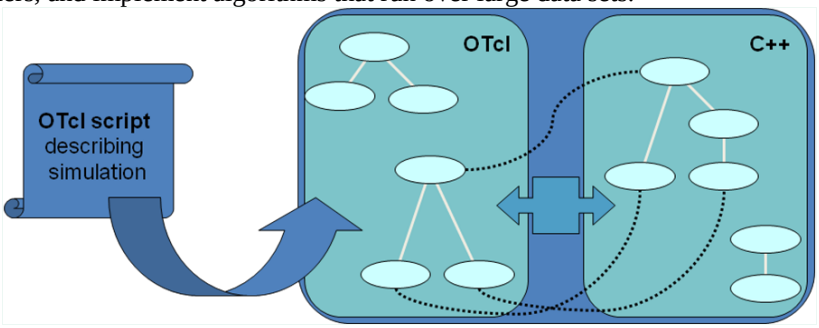

Figure 4.1: NS2 Internal Schematic Diagram

For these tasks run-time is important and turn-around time (run simulation, find bug, fix bug, recompile, re-run) is less important. C++ is fast to run but slower to change, making it suitable for detailed protocol implementation. A large part of network research involves slightly varying parameters or configurations, or quickly exploring several scenarios. In these cases, iteration time (change the model and re-run) is more important. Since configuration runs once (at the beginning of the simulation), run-time of this part of the task is less important. Tcl runs slower than C++ but can be changed very quickly (and interactively), making it ideal for simulation configuration. Users create new simulator objects through the Tcl interpreter. These objects are instantiated within the interpreter, and are closely mirrored by a corresponding object in the compiled hierarchy. Class TclObject is the base class for most of the other classes in the interpreted and compiled hierarchies. Every object in the class TclObject is created by the user from within the interpreter. An equivalent shadow object is created in the compiled hierarchy. The two objects are closely associated with each other. The interpreted class hierarchy is automatically established through methods defined in the class TclClass. User instantiated objects are mirrored through methods defined in the class TclObject.

Tcl / C++ Variable Binding:

Class InstVar defines the methods and mechanisms to bind a C++ member variable in the compiled shadow object to a specified Tcl instance variable in the equivalent interpreted object. The binding is set up such that the value of the variable can be set or accessed either from within the interpreter, or from within the compiled code always. Whenever the variable is read through the interpreter, the trap routine is invoked just prior to the occurrence of the read. The routine invokes the appropriate get function that returns the current value of the variable. This value is then used to set the value of the interpreted variable that is then read by the interpreter. Likewise, whenever the variable is set through the interpreter, the trap routine is invoked just after to

the write is completed. The routine gets the current value set by the interpreter, and invokes the appropriate set function that sets the value of the compiled member to the current value set within the interpreter.

Basic Primitive for Creating a Node: The basic primitive for creating a node is

- ✓ set ns [new Simulator]
- ✓ $ns node

The instance procedure node constructs a node out of simpler classifier objects (to be discussed later). The Node itself is a standalone class in Tcl. However, most of the components of the node are themselves Tcl Objects. Consider the Fig.4.2.,this simple structure consists of two Tcl Objects: an address classifier (classifier_) and a port classifier (dmux). The function of these classifiers is to distribute incoming packets to the correct agent or to correct outgoing link.

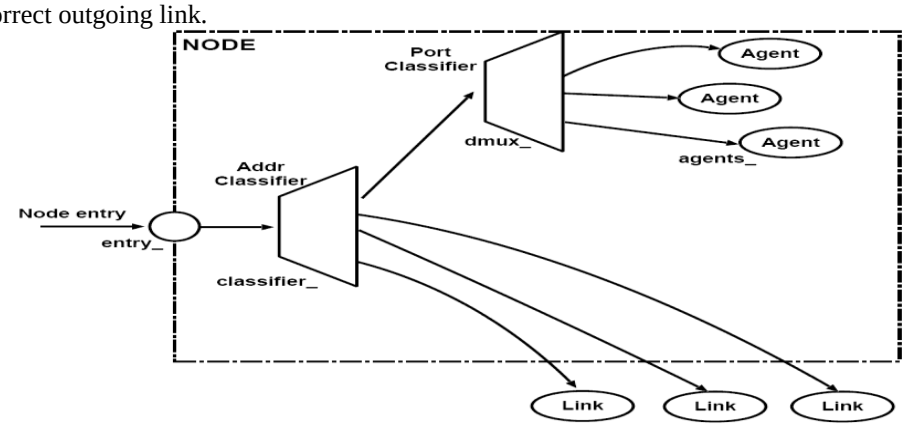

Figure 4.2: Node Structure.

Trace and Monitoring Support:

There are several ways of collecting output or trace data on a simulation. Generally, trace data is either displayed directly during execution of the simulation, or (more commonly) stored in a file to be post-processed and analyzed. There are two primary but distinct types of monitoring capabilities currently supported by the simulator. The first, called traces, record each individual packet as it arrives, departs, or is dropped at a link or queue. Trace objects are configured into a simulation as nodes in the network topology, usually with a Tcl "Channel" object hooked to them, representing the destination of collected data (typically a trace file in the current directory). The other types of objects, called monitors, record counts of various interesting quantities such as packet and byte arrivals, departures, etc.

Motivation for Simulations:

- ✓ Cheap does not require costly equipment
- ✓ Complex scenarios can be easily tested
- ✓ Results can be quickly obtained more ideas can be tested in a smaller time frame
- ✓ The real thing isn't yet available
- ✓ Controlled experimental conditions

– Repeatability helps aid debugging

Disadvantages are the follows

- ✓ Real systems too complex to model Features of NS-2
- ✓ Protocols: TCP, UDP, HTTP, Routing algorithms, MAC etc
- ✓ Traffic Models: CBR, VBR, Web etc
- ✓ Error Models: Uniform, bursty etc
- ✓ Misc: Radio propagation, Mobility models, Energy

The models are

- ✓ Topology Generation tools
- ✓ Visualization tools (NAM), Tracing NS Structure
- ✓ NS is an object oriented discrete-event simulator

– Simulator maintains list of events and executes one event after another

Single thread of control: no locking or race conditions

- 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

Basics of NS Programming:

Variables: set x 1

Arrays: set y $x
Printing: set a(0) 1
Arithmetic Expression: puts "$a(0) \n"
Control Structures: set z = [expr $y + 5]

Procedures:

```
if {$z == 6} then { puts "Correct!"}
for {set i =0} {$i < 5} {incr i }{
puts "$i * $i equals [expr $i * $i]"
}
proc sum {a b} {
return [expr $a + $b] }
```

NS Programming Structure:

- ✓ Create the event scheduler
- ✓ Turn on tracing
- ✓ Create network topology
- ✓ Create transport connections
- ✓ Generate traffic
- ✓ Insert errors Creating Event Scheduler
- ✓ Create event scheduler: set ns [new simulator]
- ✓ Schedule an event: $ns at <time> <event> event is any legitimate ns/tcl function

Conclusion:

In this paper, I have proposed a novel security and trust routing scheme based on active detection, andit has the following excellent properties: High successful routing probability, security and scalability. The Active Trust scheme can quickly detect the nodal trust and then avoid suspicious nodes to quickly achieve a nearly 100% successful routing probability. High energy efficiency. The Active Trust scheme fully uses residue energy to construct multiple detection routes. The theoretical analysis and experimental results have shown that our scheme improves the successful routing probability by more than 3 times, up to 10 times in some cases. Further, our scheme improves both the energy efficiency and the network security performance. It has important significance for wireless sensor network security.

Future Scope:

The further progress is based on the simulation of the appropriate progaramme. For that the analysis of Simulator is mandatory. The analysis includes the scope as well as criticisms of simulator which will be followed by tracing. NS programming is the next step which required for the simulation which leads to a study on creation topology helpful for tracing and animation. After that the data transmission and its related observations have to be done. By means of that the packet loss and other errors have to be verified and thus hopefully the topic will reach the assumed conclusion with final output

References:

1. Aad I. Hubaux P. J. and. Knightly W. E, 2008."Impact of Denial-of-Service Attacks on Ad-Hoc Networks," IEEE-ACM Transactions on Networking, vol. 16, no. 4, pp. 791- 802,
2. Dong M., Ota K., Liu A., et al. 2016. "Joint Optimization of Lifetime and Transport Delay under Reliability Constraint Wireless Sensor Networks," IEEE Transactions on Parallel and Distributed Systems, vol. 27, no. 1, pp. 225-236, 2016.
3. He D. Chen C. , Chan S. Bu J. Vasilakos A. V. 2012. "ReTrust: Attack-resistant and lightweight trust management for medical sensor networks," IEEE Transactions on Information Technology in Biomedicine, vol. 16, no. 4, pp. 623-632, 33. Gómez F. Mármol, Martínez Pérez G. 2012"TRIP, a trust and reputation infrastructure-based proposal for vehicular ad hoc networks," Journal of Network and Computer Applications, vol. 35, no. 3, pp. 934-941..
4. He Q. , Wu D. , . Sori P. K, 2004."a secure and objective reputation-based incentive scheme for ad hoc networks," IEEE Wireless Communications and Networking Conference, pp. 825–830,
5. Hu .Y, Dong M., Ota K., et al.Doi: 10.1109/JSYST.2014.2308391, 2014. "Mobile Target Detection in Wireless Sensor Networks with Adjustable Sensing Frequency," IEEE System Journal,
6. He S., Chen J., Jiang F., et al. (2013) "Energy provisioning in wireless rechargeable sensor networks," IEEE transactions on mobile computing, vol. 12, no. 10, pp. 1931-1942, 2013.
7. He S.,Chen J., Li X, (2015). et al. "Mobility and intruder prior information improving the barrier coverage of sparse sensor networks," IEEE transactions on mobile computing, vol. 13, no. 6, pp.1268-1282,
8. Hsieh M. Y. Huang Y. M, Chao H. C. 2007. "Adaptive security design with malicious node detection in cluster-based sensor networks," Computer Communications, vol. 30, no. 1, pp. 2385-2400,
9. Hu Y., Liu A. 2015.. "An efficient heuristic subtraction deployment strategy to guarantee quality of event detection for WSNs," The Computer Journal, vol. 58, no. 8, pp. 1747-1762,

10. Kamvar S., Schlosser M.. Garcia-Molina H, (2003). "The eigentrust algorithm for reputation management in P2P networks," in: Proceedings of the 12th International Conference on World Wide Web, pp. 640–651,
11. Liu Y., Zhu Y., Ni L. M., ,(2006)et al. 2011 "A reliability-oriented transmission service in wireless sensor networks," IEEE Transactions on Parallel and Distributed Systems, vol. 22, no. 12, pp. 2100-2107
12. Lee S. J., Gerla M. 2011., "Split Multipath Routing with Maximally Disjoint Paths in Ad Hoc Networks," IEEE ICC, pp. 3201-3205,
13. Leligou H. C., Trakadas P., Maniatis S., Karkazis P., Zahariadis T. 2012."Combining trust with location information for routing in wireless sensor networks," Wireless Communications and Mobile Computing, vol. 12, no. 12, pp. 1091-1103,
14. Lou W., Kwon Y., "H-Spread: A Hybrid Multipath Scheme for Secure and Reliable Data Collection in Wireless Sensor Networks," IEEE Transaction on vehicular technology, vol. 55, no. 4, pp. 1320-1330
15. Nghiem T. P., Cho T. H. 2010 , "A multi-path interleaved hop-by-hop en-route filtering scheme in wireless sensor networks," Computer Communications, vol. 33, no. 10, pp. 1202-1209,
16. Stinson. D. R. Cryptography, Theory and Practice. CRC Press, 200013. J. Long, A. Liu, M. Dong, et al.(2015) "An energy-efficient and sink-location privacy enhanced scheme for WSNs through ring based routing," Journal of Parallel and Distributed Computing, vol. 81, pp. 47-65,.
17. Wang J, Liu, Jiao Y., "Building a trusted route in a mobile ad hoc network considering communication reliability and path length," Journal of Network and Computer Applications, vol. 34, no. 4, pp. 1138-1149,.
18. Zhan G. X. Shi W. S., Deng J L, 2012 "SensorTrust: A resilient trust model for wireless sensing systems," Pervasive and Mobile Computing, vol. 7, no. 4, pp. 509-522,.
19. Zhan G. X., Shi W. S., Deng J. L. 2012. "Design and implementation of TARF: A trust-aware routing framework for WSNs," IEEE Transactions on Dependable and Secure Computing, vol. 9, no. 2, pp. 184-197.
20. Zhang Y., He S., Chen J. 2015. "Data Gathering Optimization by Dynamic Sensing and Routing in Rechargeable Sensor Networks," IEEE/ACM Transactions on network, doi:10.1109/TNET.2015.2425146
21. Zheng Z., Liu A., Cai L. 2016. et al. "Energy and Memory Efficient Clone Detection in Wireless Sensor Networks," IEEE Transactions on Mobile Computing.vol. 15, no. 5, pp. 1130-1143,
22. Zhou P., Jiang S., Irissappane A. 2015. et al. "Toward Energy-Efficient Trust System Through Watchdog Optimization for WSNs," IEEE Transactions on Information Forensics and Security, vol. 10, no. 3, pp. 613-625,