A Credit based Security Scheme (CBSS) for Multihop Routing in Wireless Sensor Networks

Mary Cherian
Associate professor
Dept. of CSE
Dr. Ambedkar Institute of Technology, Bengaluru, India

Anvitha A S
Dept. of CSE
Dr. Ambedkar Institute of TechnologyBengaluru, India

ABSTRACT
In most of the scenarios, communication in Wireless Sensor Network (WSN) takes place via multihop routing. Multihop routing does not provide much protection against identity deception, developed through the replay of routing information. This defect can be exploited by an opponent to launch several harmful attacks or even attacks that destroy the routing protocols, misdirecting the network traffic which results in disastrous consequences. In order to secure the multihop routing in WSNs from the adversaries, a Credit Based Security Scheme (CBSS) for WSN is proposed. CBSS, not only reduce the negative impacts from intruders but it also proves energy-efficient by incorporating the trustworthiness of the nodes into the routing decisions. The focus of this paper is to increase the Packet Delivery Ratio and offer an energy efficient route to the base station even in the presence of wormhole attack by incorporating Credit Based Security Scheme (CBSS). CBSS is proved effective through extensive evaluation with simulation using NS2.

Keywords
Wireless Sensor Networks, multihop routing, identity deception, credit value, energy rate.

1. INTRODUCTION
The fundamental component of a WSN [2] is a node which may range from hundreds to thousands in number, where each node is connected to one or several sensors. Since the sensor nodes are battery-powered, their processing capability is very limited. Messages to the base station will be sent wirelessly with a narrow radio communication range via a multihop route. Often, the multihop routing of WSNs becomes target area of malignant attacks. An assaulter may physically alter the nodes, create traffic collision from valid transmissions and even drop or misdirect the message in the routes.

In order to launch a simple type of attack which is easy to implement and harmful as well, a malignant node replays the routing packets from a valid node and forges the latter node's identity. A malignant node can now participate in the network routing to disorder the network traffic. Entire routing packets, including their original headers, are replayed as it is without any alteration. The only way for a node in a WSN to know about sender's identity is through the packets received. Hence, replaying the routing packets allows a malignant node to forge a valid node's identity. With this forged identity, a malignant node may disrupt the network traffic. For example, the packets may be dropped, form a network loop where in, the packets are passed through the malignant nodes infinite number of times, or even forward packets to a node which is not part of the routing path.

In this paper, Credit Based Security Scheme (CBSS) is proposed to protect the WSNs from the packet drop caused by wormhole attack. CBSS is proposed to offer secure routing solution in Wireless Sensor Networks (WSNs). The design of CBSS is mainly based on trustworthiness and energy efficiency. CBSS is designed to achieve high Packet Delivery Ratio even in the presence of wormhole attack [3] and offer an energy efficient route to base station via the intermediate nodes . CBSS is proved effective through simulation results.

2. RELATED WORK
Some more related work in addition to the introduction in section 1 is discussed in this section. Secure Routing in Wireless Sensor Networks [4], embodies threat models and security goals for secure routing in WSN. It introduces two novel classes of undocumented attacks against sensor networks such as sinkhole attacks and HELLO floods. It provides a way in which attacks against ad-hoc wireless networks and peer-to-peer networks can be incorporated into powerful attacks against sensor networks. Collaborative Trust-based Secure Routing in ad-hoc networks [5] incorporates a secure routing protocol in extension to earlier T-AODV routing protocol, where a secure end-to-end path is found free of malicious nodes. Any malicious entity trying to inject wrong routing information either independently or acting in collusion is effectively distinguished. Trust Aware Routing Protocol [6] is a routing protocol for sensor-actuator networks that keeps track of nodes' routing behavior and links' quality to determine efficient paths from SANET's nodes to its base station. Zahariadis [7] proposed another secure routing solution for WSN based on trust and reputation management. But both Trust Aware Routing Protocol and Along Track Scanning Radiometer don't address identity theft.

3. CREDIT BASED SECURITY SCHEME (CBSS)
It is necessary to make certain assumptions regarding Credit Based Security Scheme (CBSS) before getting into detailed design concept of the same.

3.1 Presumptions
Some of the assumptions made in the Credit Based Security Scheme (CBSS) are as follows: The aim is to provide secure routing for data collection task which is one of the most fundamental functions of WSNs. It is assumed that there is only one base station, though there could be more than one base station. An opponent may forge the identity of a valid node by replaying the outgoing routing packets of that valid node and spoof the acknowledgement packets, even remotely through a wormhole. It is assumed that a data packet has at

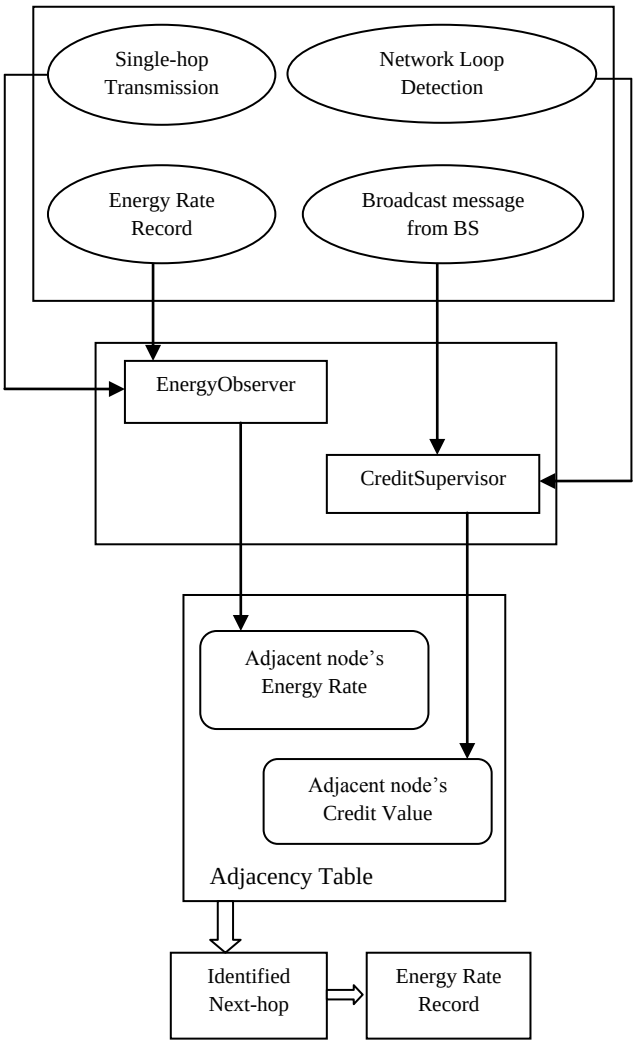
least the following fields: the transmitter id, the transmitter sequence number, the next node id, the root node id, and the root node's sequence number. Including root node's information helps the base station in tracing the successful delivery of data packet. And finally, it is assumed that the routing packet is sequenced.

3.2 Description of the scheme

CBSS evaluates the trustworthiness of adjacent nodes. It identifies intruders by low credit values and routes data through a path that bypasses those intruders to achieve satisfactory throughput, thus providing secure routing solution. In CBSS, for a node P to route a data packet to the base station, P has to select the adjacent node to which it forwards the data packet. P selects its adjacent node based on energy rate and credit value. After the data packet is forwarded to that adjacent node, the remaining task of delivering the data to the base station is completely assigned to it, and P is totally unaware of the routing decision that its adjacent node makes. An adjacency table is maintained by P with credit values and energy rate values for certain known adjacent nodes. In order to keep the table size acceptable, sometimes it is essential to delete some adjacent nodes' entries. The technique used to build an adjacency table is given in [8]. There are two types of routing information that need to be exchanged along with the data packet transmission: broadcast messages from the base station about data delivery and energy rate record messages from each node. None of the two messages needs acknowledgment. The base station floods the whole network with a broadcast message. The root node's sequence number field checks whether a broadcast message is new. The second type of routing information exchanged is the energy rate summary message from each node, which is broadcast to its neighbors only once. Such an Energy rate record message received by any node will not be forwarded.

Each node P in a WSN, maintains an adjacency table with credit values and energy rate values for each known adjacent node by means of two components namely, EnergyObserver and CreditSupervisor that run on the node as shown in the Figure 1, in which each node selects an adjacent node based on its adjacency table and broadcasts its energy rate within its adjacent surrounding. To maintain this adjacency table, EnergyObserver and CreditSupervisor on the node keep track of related events to record the energy rate and credit values of its adjacent nodes.

EnergyObserver maintains a record of energy rate for each known adjacent node based on P's observation of single hop transmission to reach its adjacent nodes and the energy rate record from those adjacent nodes. An extremely low energy rate may be reported deceitfully by a compromised node to fake its adjacent nodes to select this compromised node as their next-hop node. CreditSupervisor tracks the credit level values of adjacent nodes based on network loop detection and broadcast messages from the base station about data delivery. After P decides its next-hop adjacent node according to its adjacency table, its energy rate record message is sent out: its energy rate to deliver a packet from the node to the base station is broadcast to all its adjacent nodes. Such an energy rate record also serves as the input of its receivers' EnergyObserver.



BS – Base Station

Figure 1: Architecture of CBSS

3.3 Routing technique

Credit Based Security Scheme runs as a periodic service. Length of the period determines frequency of the routing information exchanged and updated. A message about data delivery during the last period is broadcast by the base station into the whole network. This broadcast message consists of few contiguous packets, where each packet comprises of a field that indicates the number of packets still remaining to complete the broadcast of the current message. As soon as the base station completes the broadcast, exchange of energy record in new period is triggered. End of the most recent period and start of a new period is indicated when such a broadcast message from the base station is received by a node. The main advantage of CBSS is that it does not require rigorous synchronization to keep track of start or end of a period.

EnergyObserver and CreditSupervisor are the two components that run on each node. EnergyObserver on a node monitors the energy consumed to reach its adjacent node with single-hop transmission and maintains the energy rate entries in the adjacency table by processing the energy rate record from those adjacent nodes. The responsibility of CreditSupervisor is to keep track of network loops and

maintain the credit values in the adjacency table by processing the broadcast message about data delivery from base station.
If a node retains the same next-hop node until the occurrence of next fresh broadcast message from the base station, then stability of a routing path is said to be maintained and also guarantees that all paths are loop-free. Since it leads to slow improvement in routing path, a node is allowed to change its next-hop node when its current next-hop node is noticed to have poor performance in receiving and delivering data.

3.3.1 Routing information pattern and its interchange

A broadcast message from base station is sent in few contiguous packets. Broadcast message consists of some pairs of <node id of a root node, an interval of sequence not delivered [r, s] with a significant length>, <node id of a root node, minimal sequence number received in last period>, as well as several node id intervals of those without any delivery record in last period; only limited number of such pairs is selected in order to reduce overhead.
Each node in the network maintains a table of <node id of a root node, a forwarded sequence interval with significant length> about last period. Data packets that have root node and sequence number which falls in the forwarded sequence [r, s] have already been forwarded by this node. The reception of a broadcast message about data delivery allows the CreditSupervisor of a node to identify the packets that are not forwarded by this node to the base station. Once the table is full, old entries will be deleted to reduce the overhead in storing the table. As soon as a new broadcast message from the base station is received, a node invalidates all the existing node entries immediately. The node then receives new energy rate record from its adjacent nodes and selects its new next hop node either after a time out is reached or after it has received an energy rate record from highly trusted nodes with acceptable energy rate. Energy rate is calculated by EnergyObserver which is explained in the later section.

3.3.2 Route preference

This section illustrates how CBSS determines routes in WSN. A node P depends on the adjacency table for an ideal route selection. To select a route for delivering data to the base station, node P will select an ideal next-hop from the adjacency table based on the credit value and energy rate and then forwards the data to the chosen next-hop node. The adjacent nodes with credit value below certain threshold are not considered as candidates. However, evaluation of each adjacent node 'q' based on C_{Pq} and $\frac{Er_{Pq}}{C_{Pq}}$, allows node P to select its next-hop node among its remaining neighbors, where Er_{Pq} and C_{Pq} represent q's energy rate and credit value in the adjacency table respectively. Er_{Pq} represents the energy rate required to deliver a packet to the base station with an assumption that all the nodes in the route are sincere. $\frac{1}{C_{Pq}}$ reflects the number of attempts required to send a packet from node P to base station via multiple hops before considering credit level of q. Thus $\frac{Er_{Pq}}{C_{Pq}}$ combines trustworthiness and energy rate.

3.4 EnergyObserver

Before illustrating what EnergyObserver actually does, it is necessary to know the following notations.

Adjacent node: Adjacent node of a node P is a node that is reachable from node P with single hop wireless transmission.

Credit value: Credit value of a node, adjacent to node P is a decimal number in [0, 1], which represents P's opinion about the level of trustworthiness of that adjacent node. It is the probability that the adjacent node delivers the received data correctly to the base station. Credit value is represented by 'C' in this paper.

Energy rate: Energy rate of a node, adjacent to node P is defined as the average cost required in delivering a unit-sized data packet with this adjacent node as next-hop node, from P to the base station. Energy rate is represented by 'Er' in this paper.

This section illustrates how the EnergyObserver on node P computes Er_{Pq} and Er_P; where, former indicates the energy rate for its adjacent node q in its adjacency table and latter denotes node P's own energy rate. Er_{Pq} denotes the average cost required to deliver a unit-sized data packet successfully from node P to the base station with q being P's next-hop node, responsible for the remaining route. Single-hop transmissions occur until the acknowledgement is received or the number of retransmissions reaches a certain threshold. It is necessary to include the cost of single-hop retransmission when computing Er_{Pq}.Suppose P chooses Q to be its next-hop node then, P's energy cost is given by $Er_P = Er_{Pq}$. Also, $Er_{P\to q}$ denotes the average cost to deliver a data packet from P to its neighbor q with one hop. Considering that the retransmission cost should also be included, the following relationship can be established:

$$Er_{Pq} = Er_{P\to q} + Er_q$$

Although each neighbor q of P broadcasts its own energy rate Er_q to compute Er_{Pq}, N still has to know the value $Er_{P\to q}$. So, with an assumption that endings of single-hop transmissions from P to q are independent with same probability pr_{succ}, the average number of single-hop sending needed before the acknowledgement is received is computed as follows:

$$\sum_{k=1}^{\infty} k \cdot pr_{adj} \cdot \left(1 - pr_{adj}\right)^{k-1} = \frac{1}{pr_{adj}}$$

Er_{unit} denotes node P's energy rate to send a unit-sized data packet once, irrespective of whether it is received or not. Then,

$$Er_{Pq} = \frac{Er_{unit}}{pr_{adj}} + Er_q$$

The remaining job to compute Er_{Pq} is to obtain pr_{adj}, the probability that single-hop transmission is acknowledged. To compute pr_{adj}, P's EnergyObserver will update pr_{adj}, after each transmission from P to q , by using weighted averaging technique based on whether that transmission is acknowledged or not. A binary variable *Acknowledge* is used to record the result of current transmission; the value of *Acknowledge* is 1 if an acknowledgement is received; otherwise 0. Given *Acknowledge* and the pr_{old_adj}, the last probability value of an acknowledged transmission; a weighted average of *Acknowledge* and pr_{old_adj} is used as the

value of pr_{new_adj}. But, this method suffers from attacks that occur periodically. To solve this problem, pr_{adj} is updated with two different masses (weights): a relatively big $m_{lower} \in (0,1)$ and a relatively small $m_{upper} \in (0,1)$ as follows:

$$pr_{new\_adj} = \begin{cases} (1 - m_{lower}) \times pr_{old\_adj} + m_{lower} \times Acknowledge & \text{if } Acknowledge = 0 \\ (1 - m_{upper}) \times pr_{old\_adj} + m_{upper} \times Acknowledge & \text{if } Acknowledge = 1 \end{cases}$$

The two parameters m_{lower} and m_{upper} represents the extent to which upgraded and degraded performance are rewarded and penalized respectively. m_{lower} has to be assigned a relatively high value, if any fault and compromise is likely to be associated with high risk in order to penalize fault and compromise; m_{upper} has to be assigned a relatively low value, if few positive transactions can't make up proof of good connectivity which requires many more positive transactions.

3.5 CreditSupervisor

The CreditSupervisor on node P decides the credit value of each adjacent node based on two events: network loop detection and broadcast message from base station about data delivery. T_{Pq} denotes credit value of each node q adjacent to P in P's adjacency table. Initially, each adjacent node is assigned a neutral credit value of 0.1. The occurrence of any of those two events allows the credit value of relevant adjacent nodes to be updated. Usually, existing routing protocols have their own mechanisms to detect routing loops. The following mechanism detects routing loops, when an existing routing protocol does not offer any anti-loop mechanism.

3.5.1 Loop detection

Node P's CreditSupervisor reuses the table of <node id of a root node, a forwarded sequence interval [r, s] with a significant length> in last period. CreditSupervisor on node P not only discards the packet but also degrades its next-hop node's credit value when P finds that a received packet is already in the record table. If that next-hop node is q, then C_{old_Pq} is the latest credit value of q. A binary variable '*loop_detect*' is used to record the result of loop discovery: 0 if loop is detected; 1 otherwise. The new credit value of q, with the update of energy rate is as follows:

$$C_{new\_Pq} = \begin{cases} (1 - m_{lower}) \times C_{old\_Pq} + m_{lower} \times loop\_detect & \text{if } loop\_detect = 0 \\ (1 - m_{upper}) \times C_{old\_Pq} + m_{upper} \times loop\_detect & \text{if } loop\_detect = 1 \end{cases}$$

When P has detected a loop such that the credit value of next hop node is too low, P will change its next-hop selection to break the loop. P cannot identify the node responsible for occurrence of loop and P degrades its next-hop node's credit value to break the loop.

3.5.2 Traffic misdirection detection

Node P's CreditSupervisor performs a comparison of P's stored table of <node id of a root node, forwarded sequence interval [r, s] with a significant length> recorded in last period with that of the broadcast messages from the base station about data delivery. It computes TransmitRatio, which is defined as the ratio of number of packets transmitted successfully which are forwarded by this node to the number of forwarded packets. Then, CreditSupervisor on node P updates the credit value of its next-hop node q as follows:

$$C_{new\_Pq} = \begin{cases} (1 - m_{lower}) \times C_{old\_Pq} + m_{lower} \times TransmitRatio & \text{if } TransmitRatio < C_{old\_Pq} \\ (1 - m_{upper}) \times C_{old\_Pq} + m_{upper} \times TransmitRatio & \text{if } TransmitRatio >= C_{old\_Pq} \end{cases}$$

The target of an assaulter is to prevent data delivery. It is important to note that recommendation from the CreditSupervisor of one node has no impact on CreditSupervisor of another node. If an assaulter tries to create a false route by forging false energy report, CreditSupervisor defeats such an attempt in the following manner: CreditSupervisor on a node degrades the credit value of its current next-hop node when it finds many delivery failures from the broadcast messages of the base station. When the credit value goes too low, it allows the node to select another next-hop node which is more trustworthy. It seems to be immoral when the CreditSupervisor degrades the credit value of a next-hop node which is honest when the attack takes place somewhere apart from that honest next-hop node in the route. The following example in Figure 2 shows how CreditSupervisor works.

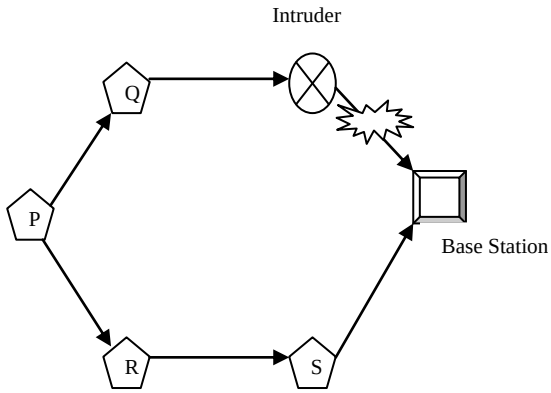


Figure 2: Working of CreditSupervisor

In the above depicted example, P, Q, R and S are honest nodes. Node Q is the current next-hop node of node P and an intruder is the next-hop node of node Q. The intruder drops every packet received and thus prevents data delivery to the base station. Thus the packets passing through node P seems to be undelivered. In effect, node P's CreditSupervisor degrades the credit value of its current next-hop node Qin spite of node Q being honest. Once the credit value becomes too low, node P selects node R as its new next-hop node. Thus P discovers a better route (P – R – S – base station). In spite of the sacrifice of node Q's credit value, the network performs better. Once a valid node finds an honest node as its next-hop node, it tends to retain the same next-hop node to maintain the stable routes.

4. PERFORMANCE AND RESULTS

In this section the experiments conducted with NS2 indicate that the performance of CBSS in the presence of wormhole attack offer better Packet Delivery Ratio. Here we consider two scenarios: Sensor networks with (i) 10 sensor nodes deployed in an area of 800*800. (ii) 20 sensor nodes deployed in an area of 800*800.

Initial energy is set to 100J and initial credit value has been set to 10.We have done simulation for the following scenarios (i) when there is no wormhole attack in the network (ii) when there is wormhole attack in the network, and (iii) when a wormhole attack takes place in a CBSS implemented sensor network.

(i) No wormhole attack: In this scenario, there is no attack and the CBSS is not incorporated. The packet delivery takes place normally. (ii) **Wormhole Attack:** In this scenario, the assaulter node drops every incoming packet and the Packet Delivery Ratio is considerably decreased. CBSS is not incorporated.

(iii) Wormhole attack in a CBSS implemented sensor network: In this scenario, we are considering a CBSS implemented sensor network in the presence of wormhole attack. A different route is constructed to the destination node in order to bypass the assaulter node so that packets will be routed through the alternate path and the Packet Delivery Ratio is improved.

Packet Delivery Ratio for 10 nodes and 20 nodes is recorded for simulation times: 5 ms, 10 ms, 15 ms, and 20 ms as shown in Table 1 and Table 2 respectively. Thus by incorporating CBSS in the presence of wormhole attack achieves more number of packets delivered to the destination with low overhead.

Table 1: Packet Delivery Ratio for 10 nodes

Scenario	Packet Delivery Ratio [No. of nodes =10]			
	5ms	10ms	15ms	20ms
No Attack [No CBSS]	36.96	32.37	31.07	30.26
Wormhole Attack [No CBSS]	7.36	10.47	10.28	10.55
Wormhole Attack [With CBSS implemented]	36.64	32.13	30.97	30.3

Table 2: Packet Delivery Ratio for 20 nodes

Scenario	Packet Delivery Ratio [No. of nodes = 20]			
	5ms	10ms	15ms	20ms
No Attack [No CBSS]	8	8.15	8.15	7.23
Wormhole Attack [No CBSS]	1.12	3.83	3.99	3.75
Wormhole Attack [With CBSS implemented]	2.88	7.51	5.01	4.31

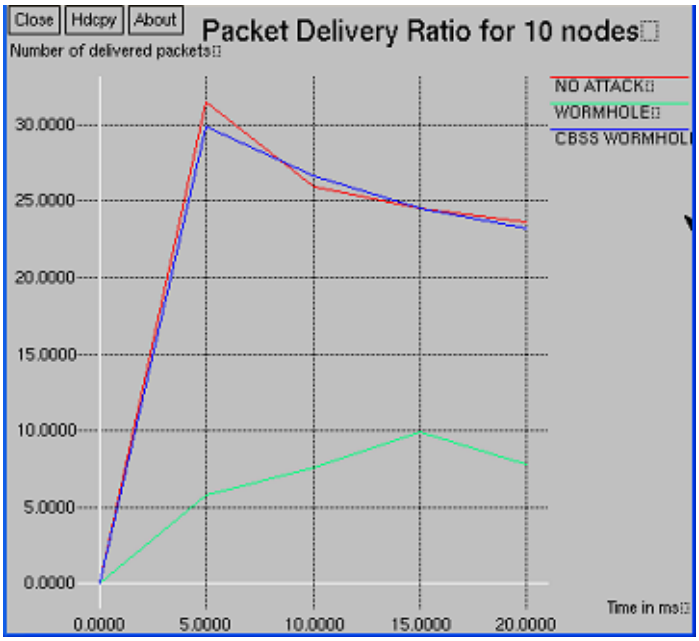


Figure 3: Packet Delivery Ratio [10 nodes]

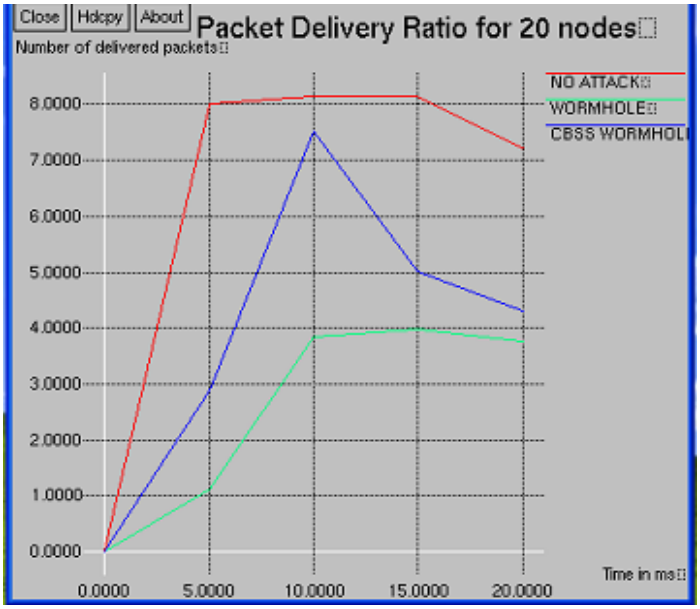


Figure 4: Packet Delivery Ratio [20 nodes]

5. CONCLUSION

A Credit Based Security Scheme (CBSS) is designed and implemented to secure the multihop routing in WSN against one of the harmful attacks such as wormhole attack. As stated, in wormhole attack, the assaulter node drops every incoming packet; thus preventing the successful delivery of packets to the destination. Thus CBSS selects an energy efficient route based on the energy efficiency and trustworthiness of the adjacent nodes in the adjacency table. With this phenomenon, number of packets delivered by source to the destination is increased.

6. REFERENCES

[1] Guoxing Zhan, WeisongShi, and Julia Deng, "Design and Implementation of TARF: A Trust-Aware Routing Framework for WSNs", IEEE Transactions on

Dependable and Secure Computing, vol. 9, No. 2, March/April 2012.

[2] F. Zhao and L. Guibas , Wireless Sensor Networks: An Information Processing Approach. Morgan Kaufmann, 2004.

[3] M. Jain and H. Kandwal, "A Survey on Complex Wormhole Attack in Wireless Ad Hoc Networks" Proc. International Conference Advances in Computing, Control, and Telecomm. Technologies (ACT '09), pp. 555-558, 2009.

[4] C. Karlof and D. Wagner, "Secure Routing in Wireless Sensor Networks: Attacks and Countermeasures", Proc. First IEEE International Workshop Sensor Network Protocols and Applications, 2003.

[5] T. Ghosh, N. Pissinou, and K. Makki, "Collaborative Trust-Based Secure Routing against Colluding Malicious Nodes in Multi-Hop Ad Hoc Networks", Proc. 29th IEEE International Conference Local Computer Networks, pp. 224-231, Nov. 2004.

[6] A. Rezgui and M. Eltoweissy, "Tarp: A Trust-Aware Routing Protocol for Sensor-Actuator Networks", Proc. IEEE International Conference Mobile AdHoc and Sensor Systems (MASS '07), 2007.

[7] T. Zahariadis, H. Leligou, P. Karkazis, P. Trakadas, I. Papaefstathiou, C. Vangelatos, and L. Besson, "Design and Implementation of a Trust-Aware Routing Protocol for Large WSNs", International Network Security and its Applications, vol. 2, No. 3, pp. 52-68, July 2010.

[8] A. Woo, T. Tong, and D. Culler, "Taming the Underlying Challenges of Reliable Multihop Routing in Sensor Networks", Proc. First ACM International Conference Embedded Networked Sensor Systems (SenSys '03), Nov. 2003.