International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 3 (2011) pp. 363-370

An Efficient Algorithm for Detection of Selfish Packet Dropping Nodes in Wireless Mesh Networks

Jaydip Sen

Innovation Labs, Tata Consultancy Services Ltd.,
Bengal Intelligent Park, Salt Lake Electronic Complex, Kolkata, India
Jaydip.sen@tcs.com

***Abstract*: In a wireless mesh network (WMN), high speed routers equipped with advanced antennas, communicate with each other in a multi-hop fashion over wireless channels and form a broadband backhaul. WMNs provide reliable connectivity and fault-tolerance, as each node is connected to several other nodes. If a node fails due to hardware problems, its neighbors can find another route. Extra capacity can be achieved by introducing additional nodes in the network. However, the throughput of a WMN may be severely degraded due to presence of some selfish routers that avoid forwarding packets for other nodes even as they send their own traffic through the network. This paper presents an algorithm for detection of selfish nodes in a WMN that uses statistical theory of inference for reliable clustering of the nodes. Simulation results show that the algorithm has a high detection rate and a low rate of false positives.**

***Keywords*:** Wireless mesh network (WMN), selfish node, finite state machine, AODV protocol, ANOVA, clustering.

I. Introduction

Wireless mesh networking has emerged as a promising concept to meet the challenges in next-generation networks such as providing flexible, adaptive, and reconfigurable architecture while offering cost-effective solutions to the service providers [1]. Unlike traditional Wi-Fi networks, where each access point (AP) is connected to a wired network, in WMNs only a subset of the APs are required to be connected to a wired network. As shown in Figure 1, the APs that are connected to the wired network are called the Internet gateways (IGWs), while the APs that do not have wired connections are called the mesh routers (MRs). The MRs are connected to the IGWs using multi-hop communication. Due to the recent research advances in WMNs, these networks have been used in numerous applications such as in home networking, community and neighborhood monitoring, security surveillance systems, disaster management and rescue operations etc [2].

In a community-based WMN, a group of MRs managed by different operators form an access network to provide last-mile connectivity to the Internet. As with any end-user supported infrastructure, ubiquitous cooperative behavior in these networks cannot be assumed a priori.

Preserving scarce access bandwidth and power, as well as security concerns may induce some selfish users to avoid forwarding data for other nodes, even as they send their own traffic through the network. The selfish behavior of an MR increases the latency in packet delivery and packet drops and decreases the network throughput in a WMN [2].

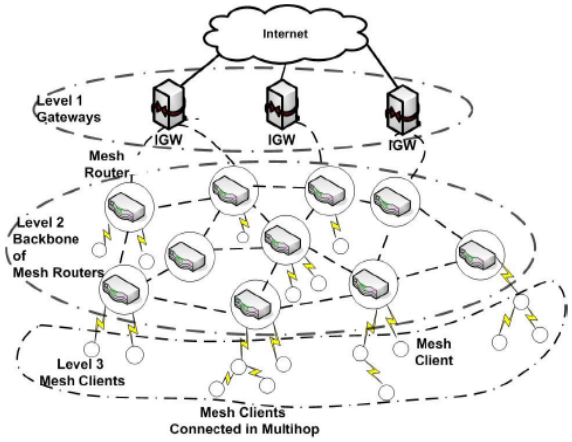

Figure 1. The hierarchical architecture of a WMN

To enforce cooperation among the nodes and for detection of selfish nodes in ad hoc wireless networks, various collaborative schemes have been proposed in the literature [3]. Majority of these proposals are based on trust and reputation frameworks which attempt to identify misbehaving nodes by suitable detection systems and then isolate these nodes from the network activities. The reputations of participating nodes are computed based on the local observations at the nodes and the second-hand observations at other nodes.

To address the issue of detection of selfish nodes in a WMN, this paper presents a scheme that is based on local observations in the nodes. The scheme is applicable for on-demand protocols like AODV, and uses statistical theory of inference and clustering techniques to make a robust and reliable classification of the nodes. To increase the detection accuracy, the scheme also introduces some additional fields in the headers of the AODV packets.

The rest of the paper is organized as follows. Section II presents some existing related work in the literature. Section III gives a brief background of the AODV protocol and a

finite state machine model of a node. Section IV discusses the proposed scheme. Section V presents simulations results. Section VI identifies some future work and concludes the paper.

II. Related Work

The concept of neighborhood monitoring to check the activities of other nodes has been proposed by researchers in a number of mechanisms especially in the context of wireless ad hoc networks. The idea of watchdog mechanism to monitor neighbors was first proposed by Marti et al. [4]. The authors have also proposed a scheme named *pathrater* to avoid misbehaving nodes in routing. Buchegger and Boudec have proposed the CONFIDANT protocol that is based on selective altruism and utilitarianism [5]. It is a distributed, symmetric reputation model that uses both first-hand and second-hand information for computation of reputation values. It uses dynamic source routing (DSR) protocol for routing and assume that promiscuous mode of operation is possible. The misbehaving nodes are punished by isolating them from accessing the network resources.

Mahajan et al. have proposed a mechanism named CATCH [6], which consists of two modules: (i) *anonymous challenge message* (ACM), and (ii) *anonymous neighbor verification* (ANV). In the security scheme, first an ACM message from an unknown sender is sent to all its neighbors. As the sender is unknown, all the nodes further broadcast the ACM message. In the ANV phase, a tester node sends cryptographic hash of a random token for rebroadcast and also records other hashes sent by other nodes. The tester node releases the secret token to another node which successfully authenticates itself.

Vigna et al. have proposed an approach to detect intrusions in AODV that works by *stateful signature-based analysis* of the observed traffic [7]. Sensors are placed on selected nodes for promiscuous sensing of radio channels. Each sensor has database of attack signatures and looks for a signature match in the traffic. A match triggers a response, usually an alert.

Pirzada et al. have described a model of building trust relationship between nodes in an ad hoc network [8]. The nodes passively monitor the packets received and forwarded by other nodes and compute the trust values for their neighbors. The trust values are used for computing the trustworthiness of links. For routing, links with high trust values are chosen so as to avoid the malicious and selfish nodes.

Conti et al. have proposed a scheme in which a node exploits its local knowledge to estimate the reliability of a path [9]. Unlike the conventional method of denying selfish users, it provides a degraded service to these nodes by selective slow packet forwarding.

Patwardhan et al. have proposed a trust-based data management scheme in which mobile nodes access distributed information, storage, and sensory resources available in pervasive computing environment [10]. The authors have taken a holistic approach that considers data, trust, security, and privacy and utilizes a collaborative mechanism that provides trustworthy data management platform in an ad hoc network for secure authentication, data communication, data access and certificate and key management.

Santhanam et al. have presented a mechanism to judge a node's behavior based on observed traffic reports submitted to local sink agents, dispersed throughout the network [11]. The sink nodes apply a set of forwarding rules to isolate a selfish node based on the number of times it is caught in selfish acts. The scheme is independent of the routing protocol or network architecture, and is suitable for multi-channel wireless mesh network.

Baras et al. have proposed a trust management scheme for self-organized ad hoc networks, where the nodes share trust information only with their neighbors [12]. For establishing and maintaining trust among the neighbors authors have proposed a voting mechanism.

Repantis et al. have proposed a decentralized trust management middleware for ad hoc, peer-to-peer networks based on reputation [13]. The reputation information of each peer is stored in its neighborhood and piggybacked on its replies.

Tseng et al. have applied techniques based on finite state machines to detect misbehaving nodes in AODV routing protocol [14]. The approach involves monitoring nodes that cooperate with each other and aggregate their observations at different locations in the network.

Chang et al. have proposed a trust-based scheme for multicast communication in a MANET [15]. In a multicast MANET, a sender node sends packets to several receiving nodes in a multicast session. Since the membership in a multicast group in a MANET changes frequently, the issues of supporting secure authentication and authorization in a multicast MANET are very critical. The proposed scheme involves a two-step secure authentication method. First, an *ergodic* continuous Markov chain is used to determine the trust value of each one-hop neighbor. Second, a node with the highest trust value is selected as the certificate authority (CA) server. For the sake of reliability, the node with the second highest trust value is selected as the backup CA server. The analytical trust value of each mobile node is found to be very close to that observed in the simulation under various scenarios. The speed of the convergence of the analytical trust value shows that the analytical results are independent of the initial values and the trust classes.

Sun et al. have presented trust as a measure of uncertainty [16]. Using theory of entropy, the authors have developed a few techniques to compute trust values from certain observation. In addition, trust models – entropy-based and probability-based – are presented to solve the concatenation and multi-path trust propagation problems in a MANET.

Sen et al. have proposed a self-organized trust establishment scheme for nodes in a large-scale MANET in which a trust initiator is introduced during the network bootstrapping phase [17]. It has been proven theoretically and shown by simulation that the new nodes joining the network have high probability of successful authentication even when a large proportion of the existing nodes leave the network at any instant of time. A distributed detection mechanism of malicious packet dropping attack in MANETs has been proposed in [18], where local anomaly detection is utilized to make a more accurate network-wide (i.e. global) detection using a cooperative detection algorithm.

Yang et al. have proposed the SCAN protocol that addresses two issues simultaneously; (i) routing (control packets) misbehavior, and (ii) forwarding (data packets) misbehavior [19]. Each node monitors its neighbors independently and the nodes in a neighborhood collaborate with each other through a distributed consensus protocol.

Benjamin et al. have proposed that WMNs can be used to transmit vital information arising from the wireless body sensor network (WBSN) to a backbone network [20]. The integration of WBSN and WMN technologies results in wireless sensor mesh network (WSMN) and this type of network can be utilized for remote health monitoring of patients. The battery-powered, memory-constrained sensors transmit the sensed information to their nearest mesh nodes and the mesh nodes, in turn, use multi-hop routing to transmit the information to the backbone network devices like PDA or the servers for health monitoring applications. The authors have investigated performance of such a WSMN for patient health monitoring applications, in terms of parameters like delay, and throughput under varying number of patients and doctors.

The proposed mechanism in this paper relies on local observation of each node in a WMN. Based on the local information in each node and using a finite state machine model of the AODV protocol, a robust statistical theory of estimation is applied to identify selfish nodes in the network. The proposed mechanism is a modification of the protocol proposed in [21]. Using statistical estimation technique, analysis of variance and some additional fields in the headers of the AODV packets, the proposed protocol is able to achieve a higher detection rate with a very low rate of false positives. In the next section, a finite state machine model of the AODV protocol is described which is used to design the proposed mechanism.

III. Modeling of the State Machine

Ad hoc on-demand distance vector (AODV) routing protocol uses an on-demand approach for finding routes to a destination node. It employs destination sequence numbers to identify the most recent path from a source node to a destination node [22]. The source node and the intermediate nodes store the next-hop information corresponding to each flow of data packet transmission. The source node floods the *route request* (RREQ) packet in the network when a route is not available for the desired destination. It may obtain multiple routes to different destinations from a single RREQ. A RREQ packet carries source identifier (*src_id*), destination identifier (*dest_id*), source sequence number (*src_seq_num*), destination sequence number (*dest_seq_num*), broadcast identifier (*bcast_id*), and time to live (TTL). When an intermediate node receives a RREQ, it either forwards the request further or prepares a *route reply* (RREP) if it has a valid route to the destination. Every intermediate node, while forwarding a RREQ, enters the previous node address and its *bcast_id*. A timer is maintained to keep track of the lifetime of this entry. In case a RREP is not received before the expiry of the timer, the record is deleted from the list. This helps in storing an active path at the intermediate node. AODV does not employ source routing of data packets. When a node receives a RREP packet, information of the previous node from which the packet was received is also stored, so that the data packets may be routed so that node as the next hop towards the destination.

A. Finite state machine model

In the proposed scheme, the set of all messages corresponding to a RREQ broadcast and the unicast RREP is referred to as a *message unit*. It is clear that no node can observe all the transmission in a message unit. The subset of a message unit that a node can observe is referred to as the *local message unit* (LMU). The LMU for a node consists of the messages transmitted by the node, the messages transmitted by its neighbors, and messages overheard by the node. The detection of selfish nodes is made based on the data collected by each node from its observed LMUs.

Corresponding to each message transmission in an LMU, a node maintains a record of its sender and the receiver in its neighborhood. It also keeps record of the neighbor nodes that receive the RREQ broadcast messages sent by the node itself. The messages follow the sequence of the standard AODV protocol.

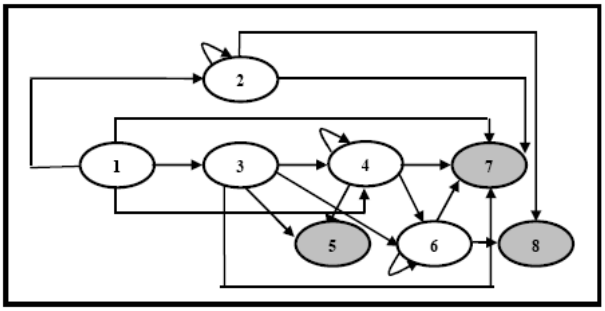

Figure 2. The finite state machine of a node

The finite state machine shown in Figure 2 depicts various states through which a neighbor node undergoes for each LMU [21]. The corresponding states for the numbers mentioned in Figure 2 are listed in Table 1. In Figure 2, the final sates are shaded. Every message transmission by a node causes a state transition in each of its neighbor's finite state machine. The finite state machine in one neighbor node gives only a local view of the activities of the node being monitored. It does not in any way, represents the actual behavior of the monitored node. The collaborative participation of each neighbor node makes it possible to get an overall idea about the behavior of the monitored node. In the rest of the paper, a node whose activity is being monitored is referred to as a *monitored node*, and each of its neighbors is referred to as a *monitor node*. Each node acts as a monitor node and a monitored node for each of its neighbors.

Each monitor node observes a series of interleaved LMUs for a routing session. Each LMU can be identified by the source-destination pair contained in a RREQ message. Let us denote the k^{th} LMU observed by a monitor node as (s_k, d_k). The tuple (s_k, d_k) does not uniquely identify a LMU as a source can issue multiple RREQs for the same destination. However, since the RREQs have some time intervals between them, it may be safely assumed that there is only one active LMU (s_k, d_k) in the network at an instant of time.

Table 1. The states of the finite state machine of LMU.

State	Interpretation
1: init	Initial phase; no RREQ is observed
2: unexp RREP	RREP received without any RREQ
3: rcvd RREQ	Receipt of a RREQ observed
4: fwd RREQ	Broadcast of a RREQ observed
5:timeout RREQ	Timeout after receipt of RREQ
6: rcvd RREP	Receipt of a RREP observed
7:LMU complete	A valid RREP forwarding observed
8: timeout RREP	Timeout after receipt of a RREP

At the beginning of a routing session, a monitored node starts with the state 1 in its finite state machine. The monitor node(s) observes the behavior of the monitored node, and records a sequence of transitions from state 1 to one of the possible final states: 5, 7 and 8.

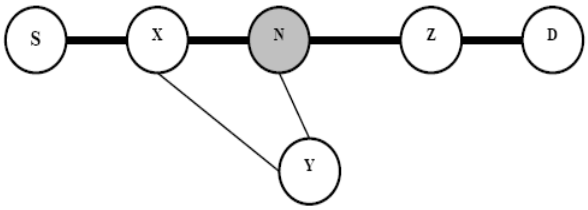

Figure 3. An LMU observed by node *N*

When a monitor node broadcasts a RREQ, it assumes that the monitored node has received it. The monitor node, therefore, records a state transition 1 → 3 for the monitored node's finite state machine. If a monitor node observes a monitored node to broadcast a RREQ, then a state transition of 3 → 4 is recorded if the RREQ message was previously sent by the monitor node to the monitored node; otherwise a transition of 1 → 4 is recorded to indicate that the RREQ was received by the monitored node from some other neighbor. The transition to a timeout state occurs when a monitor node finds no activity by the monitored node for the concerned LMU before the expiry of a timer. When a monitor node observes that a monitored node has forwarded a RREP, it records a transition to state no 7– *LMU complete*. At this state, the monitored node becomes a candidate for inclusion on a routing path.

Table 2. The state transitions of the neighbors of node *N*.

Node	Events	State changes
X	*X* broadcasts RREQ	1 → 4
	N broadcasts RREQ	4 → 4
	N sends RREP to *X*	4 → 6
	X sends RREP to *S* (overhears)	6 → 7
Y	*Y* broadcasts RREQ	1 → 4
	N broadcasts RREQ	4 → 4
	Timeout	4 → 5
Z	*N* broadcasts RREQ	1 → 3
	Z broadcasts RREQ	3 → 4
	Z sends RREP to N	4 → 7

Figure 3 depicts an example of LMU observed by the node *N* during the discovery of a route from the source node *S* to the destination node *D*. Table 2 shows the events observed by *N* and the state transitions for each of its neighbor *X*, *Y* and *Z*. When the final state is reached, the finite state machine terminates and the corresponding sequences of state transitions are stored by each node for each of its neighbors. When sufficient number of events is collected by a node, a statistical analysis is performed to detect the presence of any selfish nodes in the network.

IV. The Proposed Algorithm

As mentioned in the previous section, a monitor node keeps a record of state transitions in the finite state machine of a monitored node for each LMU. These sequences can be represented as a transition matrix $T = [T_{ij}]$, where T_{ij} is the number of times the transition $i \rightarrow j$ is found. The monitor node invokes a detection algorithm every W seconds using data from the most recent $D = d * W$ seconds of observations, where d is a small integer. The parameter D, called the *detection window*, should be such that it allows prompt identification of selfish nodes while maintaining a high level of accuracy. In Section 4.1, some of the issues in design of the proposed detection algorithm are discussed.

A. Design issues of the detection algorithm

While a transition matrix summarizes the local routing behavior of a monitored node, it is not possible to determine selfish behavior of a node based on its local transition probabilities only. By comparing the transition matrices of a group of nodes, it is possible to detect selfish nodes with higher confidence. Following this principle, the proposed algorithm initially maps the neighbors of a monitoring node into two clusters and then classifies the clusters into two types: *selfish* and *cooperative*.

Several issues are considered in clustering of the nodes. First, to make the clustering algorithm robust in presence of noise in the data, a statistical theory of inference-based approach is followed that takes into account the pair-wise comparisons of the transition matrices of each pair of nodes. Second, to reliably identify the cluster that contains the selfish nodes, an additional measure of cooperation, called *cooperation index*, for the nodes is computed. The cluster that has cooperation index less than a threshold value is identified as the one containing the selfish nodes. Finally, an *analysis of variance* (ANOVA)-based test is designed for the clusters to determine whether clustering is really informative. The proposed algorithm is described in Section 4.2.

B. The detection algorithm

As discussed earlier, each node monitors the activities of its R neighbors identified by the indices 1, 2,….R. Let $T^{(r)} = [f_{ij}^{(r)}]$ denote the observed transition matrix for the r^{th} neighbor, where $[f_{ij}^{(r)}]$ is the number of transitions from state i to state j observed in the previous detection window. If m is the number of states in the finite state machine, the size of $T^{(r)}$ is m x m. Let $T^{(r)} = [f_{i1}^{(r)}, \ldots f_{im}^{(r)}]$ denote the i^{th} row of the transition matrix $T^{(r)}$, which shows the transitions out of state i at the neighbor node r. If two neighbor nodes r and s have identical distributions corresponding to transitions from state i, then we write $T_i^{(r)} \equiv T_i^{(s)}$. To test the

hypothesis $T_i^{(r)} \equiv T_i^{(s)}$ the Pearson's χ^2 test is used as follows.

$$\chi^2(i) = \frac{\sum_{l \varepsilon (r,s)} \sum_{j=1}^{m} \left[f_{ij}^{(l)} - \bar{f}_{ij}^{(l)} \right]^2}{\bar{f}_{ij}^{(l)}} \quad (1)$$

where, $\bar{f}_{ij}^{(l)} = F_{ij}^{(l)} \frac{f_{ij}^{(r)} + f_{ij}^{(s)}}{F_i^{(r)} + F_i^{(s)}}$ and $F_i^{(r)}$ and $F_i^{(s)}$ denote total number of transitions for state *i* in $T^{(r)}$ and $T^{(s)}$ respectively.

If the value of χ^2 exceeds the value of $\chi^2_{\text{m-1, }\alpha}$ then the hypothesis $T_i^{(r)} \equiv T_i^{(s)}$ is rejected at confidence interval α. If we write K_i^{rs} for the event that $\chi^2_{(i)} > \chi^2_{\text{m-1, }\alpha,}$ then $P\left(T_i^{(r)} \equiv T_i^{(s)} \mid B_i^{rs}\right)$, the conditional probability can be taken as a reasonable estimator of the similarity between *r* and *s* with respect to the state *i*. In absence of any prior information, it is reasonable to assume that *r* and *s* have no similarity in state *i* and the probability that the Pearson test rejects its hypothesis is 0.5 [20].

In order to evaluate the similarity between *r* and *s* for all the *m* states, (1) is applied to all rows of $T^{(r)}$ and $T^{(s)}$. This yields a vector $B^{(rs)} = [B_i^{(rs)}]$, $\{i = 1,2,\ldots,m\}$. From the standard Markovian principle we can write:

$$L_{rs} = P(T^{(r)} \equiv T^{(s)} \mid B^{(rs)}) \quad (2)$$

$$\alpha^{S^{(rs)}} (1-\alpha)^{m-S^{(rs)}} \approx \alpha^{S^{(rs)}} \quad (3)$$

where

$$S^{(rs)} = \sum_{i=1}^{m} B_i^{(rs)} \quad (4)$$

The lower-order terms in the right hand side of (3) are ignored since $\alpha << 1$. For small value of α, L_{rs} monotonically decreases in $S^{(rs)}$. As evident from (3), L_{rs} is the number of rejections in Pearson's test. Therefore, 1 - L_{rs} may be taken as the measure of the dissimilarity between the neighbor nodes *r* and *s*.

In presence of noise, however, it is found that for two nodes *r* and *s* which have $L_{rs} \approx 1$, a third node *t* may cause inconsistency such that $L_{rt} \not\approx L_{st}$. To avoid this inconsistency, in the proposed algorithm, clusters are not identified on the basis of pair-wise dissimilarity. To compute dissimilarity between *r* and *s*, the *L* values for all neighbors are computed with respect to *r* and *s* separately as follows:

$$d_{rs} = 1 - \frac{n_{rs}^2}{n_{r/s} * n_{s/r}} \quad (5)$$

where, $n_{rs} = \sum_{t \neq r,s} \min(L_{rt}, L_{st})$, $n_{r/s} = \sum_{t \neq r,s}^{K} L_{rt}$ and $n_{s/r} = \sum_{t \neq r,s}^{K} L_{st}$.

It may be observed that the computation of d_{rs} does not involve L_{rs} : the pair-wise similarity between nodes *r* and *s*. In fact, the metric d_{rs} measures the degree of inconsistency in similarity between *r* and *s* with all their neighbors. Since, in the computation, contribution of each neighbor plays its role, d_{rs} presents a robust indicator for dissimilarity between nodes and plays a crucial part in computing the clusters [21]. For clustering, an *agglomerative hierarchical clustering* technique is used. This is a single-linkage approach in which each cluster is represented by all of the objects in the cluster, and the similarity between two clusters is measured by the similarity of the closest pair of data points belonging to different clusters. The cluster merging process repeats until all the objects are merged into a single cluster [23].

After the nodes are clustered into similar sets, the sets are further classified into three groups: (i) a set (*G*) of cooperative nodes, (ii) a set (*B*) of selfish nodes, and (iii) a set of nodes whose behavior could not be ascertained. The cooperation score (C_r) of a node is computed as follows:

$$C_r = \frac{\sum_{i,j \varepsilon G}^{m} n_{ij}^{(r)}}{|G|} - \frac{\sum_{i,j \varepsilon B}^{m} n_{ij}^{(r)}}{|B|} \quad (6)$$

The set *B* is most likely to contain the selfish nodes. To reduce the rate of false positives (i.e. wrongly identifying a cooperative node as selfish), an ANOVA test is applied [20]. In this test, the probability P_k of the random variation among the mean cooperation scores of *k* clusters is computed. A lower value of P_k implies that the clusters represent distinct differences in their behavior. At each iteration, *k* clusters are formed and P_k is compared with a pre-defined level of significance β. If $P_k < \beta$, clusters are believed to be reliably reflecting the behavior of the nodes and their classifications are accepted. The cluster with lowest mean cooperation score is assumed to contain the selfish nodes. If $P_k > P_{k\text{-}1}$, the neighbor behavior is not properly reflected in the cluster formation, resulting in the higher value of P_k. In this case, all the nodes are classified as cooperative, and the next iteration of the algorithm is executed. The value of β is tuned to adjust the detection alacrity and the rate of false positives.

Even with all these statistical approaches, there is still a possibility of misclassification. The probability of misclassification is further reduced by a cross-checking mechanism. For this purpose, a minor modification is made in the packet header of the AODV protocol by inserting two additional fields in the header of a RREQ packet. These additional fields, *next_to_source* and *duplicate_flag,* indicate respectively the address of the node that is next hop to the source, and whether the packet is a duplicate packet which has already been broadcasted by some other nodes. In the header of a RREP packet, another field called *next_to_destination* is added to indicate the address of the node to which the packet must be forwarded in the reverse path. With these additional

fields, it is possible to detect every instance of selfish behavior in a wireless network, if the following conditions are satisfied: (i) no packet loss lost due to interference, (ii) links are bi-directional, (iii) the nodes are stationary, and (iv) the queuing delays are bounded [24]. Since all these conditions cannot be guaranteed in a real-world deployment, there will be always some detection inaccuracy. However, with the combination of the robust clustering and monitoring of packets with additional fields, it is possible to substantially increase the detection efficiency and reduce the false positives as evident from the experimental results presented in Section V.

V. Simulation Results

The proposed protocol is evaluated with network simulator *ns-2* (version 2.29) [25] with parameters presented in Table 3. The objective is to evaluate the efficiency of the algorithm and compare its performance with the protocol proposed by Wang et al. in [21].

Table 3. Simulation parameters

Parameter	Value
Simulation area	900 m * 900 m
Simulation duration	1600 sec
No. of nodes in the network	50
MAC protocol	802.11b
Routing protocol	AODV
Raw channel bandwidth	11 Mbps
Traffic type	CBR UDP
Network traffic volume	60 packets / sec
Packet size	512 bytes
Time-out for RREQ broadcast	0.5 sec
Time-out for receiving RREP	3 sec
Pearson confidence (α)	0.1
Observation window (W)	100 sec
Detection window (D)	400 sec
Session arrival distribution	Poisson
Session duration distribution	Exponential

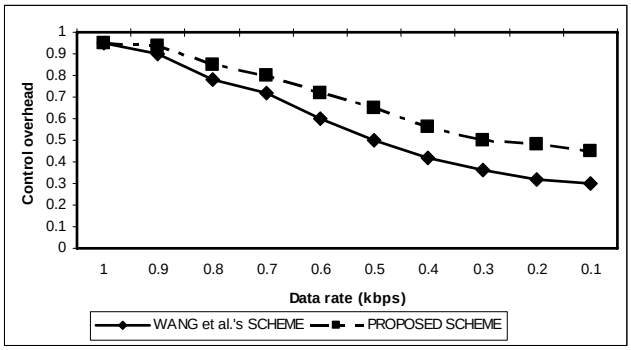

Figure 4. Detection rate in DROP_REQ case

At the start of the simulation, a fraction of nodes are chosen randomly as the selfish nodes. A selfish node adopts either of the two strategies: (i) dropping RREQs (DROP_REQ) and (ii) dropping RREPs (DROP_REP). In both cases, control packets are dropped with a constant probability. For DROP_REP, a selfish node always rebroadcasts RREQs even if it has a route in its cache. To evaluate the detection efficiency and speed, the packet dropping probability is varied from 1.0 to 0.1. The value of the parameter β is chosen as 0.4 to achieve the best tradeoff between detection rate and false positive rate.

Figure 4 and Figure 5 represent respectively the detection rate and the false alarm rate with 50% nodes in the network configured as selfish and dropping RREQs (i.e. DROP_REQ). The results are the average of 10 runs of the simulation. The proposed algorithm performs better than Wang's algorithm since it doubly checks the detection results- one from the clustering and the other from the routing header information to make a more reliable detection.

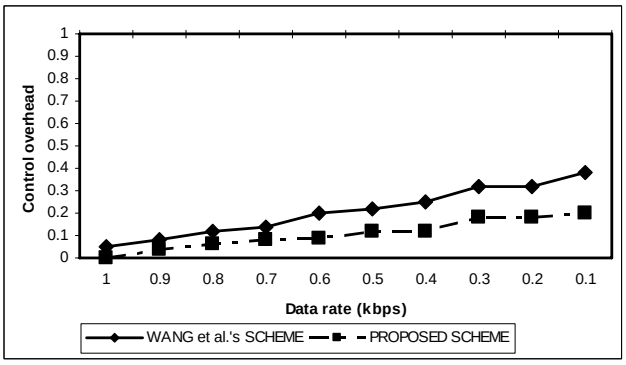

Figure 5. False alarm rate in DROP_REQ case

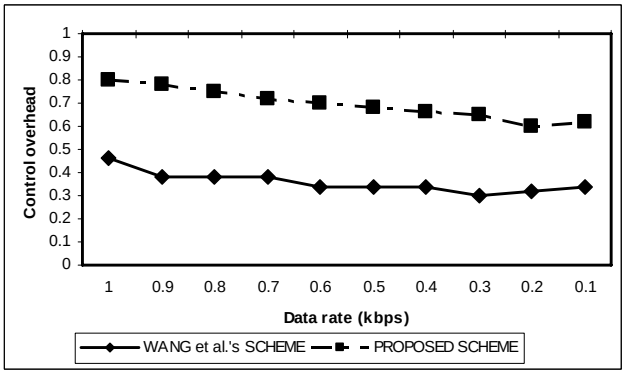

Figure 6. Detection rate in DROP_REP case

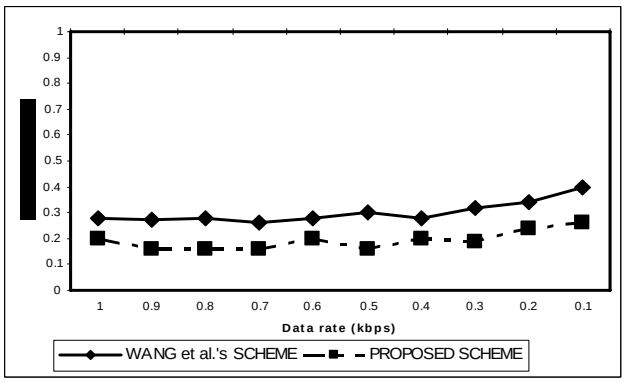

Figure 7**.** False alarm rate in DROP_REP case

Figure 6 and Figure 7 show that the packet dropping (DROP_REP) has no impact on the detection rate and the false positive rate when 50% nodes in the network are acting as selfish nodes. This difference in behavior in case of DROP_REQ and DROP_REP lies in the fact that while RREQ is a broadcast message sent from the source, the RREP is sent in a single path by the destination in a unicast manner.

Since RREP involves very few number of nodes, for majority of the nodes the state machine will terminate in state 5, instead of states 7 and 8. It is evident from Figure 6 and Figure 7 that the proposed algorithm has an average 80% increase in detection rate and 50 % reduction in false positives compared with the corresponding figures for the algorithm proposed by Wang et al. in [20].

VI. Conclusion and Future Work

Detection of selfish nodes is crucial in WMNs since these nodes don't forward packets for other nodes and thereby degrade the performance of the networks. This paper has presented a statistical theory of inference-based clustering algorithm for detection of selfish nodes in a WMN. A finite state machine model is developed on the AODV routing protocol based on the local observation in each node. To increase the reliability of clustering, an ANOVA test is applied and finally a new cross-checking mechanism is used by inserting extra fields in the AODV packet headers. Simulation results show that the algorithm has better detection efficiency and reduced false alarm rates when compared to a well-known existing algorithm. As a future scope of work, it is planned to design an efficient and secure routing protocol for WMNs using the selfish node detection algorithm described in this paper.

References

[1] I.F. Akyildiz, X. Wang, and W. Wang, "Wireless mesh networks: a survey", *Computer Networks*, Vol 47, pp. 445-487, 2005.

[2] A.A. Franklin and C. S.R Murthy, "An introduction to wireless mesh networks," in *Security in Wireless Mesh Networks* Y. Zhang, J. Zheng, and H. Hu, (Eds). CRC Press, pp. 3-44, 2007.

[3] L. Santhanam, B. Xie, and D.P. Agrawal, "Selfishness in mesh networks: wired multihop MANETs," *IEEE Wireless Communication Magazine*, Vol 15, No 4, pp. 16-23, 2008.

[4] S. Marti, T.J. Giuli, K. Lai, and M. Baker, "Mitigating routing misbehavior in mobile ad hoc networks", in *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (MobiCom '00)*, pp. 255 – 265, 2000.

[5] S. Buchegger and J.-Y. L. Boudec, "Performance analysis of the CONFIDANT protocol: cooperation of nodes- fairness in dynamic ad-hoc networks," in *Proceedings of the 3rd ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc'02)*, pp. 226-236, Lausanne, Switzerland, 2002.

[6] R. Mahajan, M. Rodrig, D. Wetherall, and John Zahorjan, "Sustaining cooperation in multihop wireless networks," in *Proceedings of the 2nd Conference on Symposium on Networked Systems Design and Implementation (NSDI'05)*, Vol 2, pp. 231-244, 2005.

[7] R. Mahajan, M. Rodrig, D. Wetherall, and John Zahorjan, "Sustaining cooperation in multihop wireless networks," in *Proceedings of the 2nd Conference on Symposium on Networked Systems Design and Implementation (NSDI'05)*, Vol 2, pp. 231-244, 2005.

[8] A. Pirzada and C. McDonald, "Establishing trust in pure ad hoc networks", in *Proceedings of the 27th Australasian Conference on Computer Science*, Vol. 26, pp. 181-199, 2004.

[9] M. Conti, E. Gregori, and G. Maselli, "Reliable and efficient forwarding in MANETs," *Ad Hoc Networks Journal*, Vol 4, No 3, pp. 398-415, 2006.

[10] A. Patwardhan, F. Perich, A. Joshi, T. Finn, and Y. Yesha, "Querying in packs: trustworthy data management in ad hoc networks", *International Journal of Wireless Information Networks*, Vol. 13, No. 4, pp. 263 – 274, October 2006.

[11] L. Santhanam, N. Nagesh, Y. Yoo, and D.P. Agrawal, "Distributed self-policing architecture for fostering node cooperation in wireless mesh networks," in *Proceedings of the 11th IFIP International Conference on Personal Wireless Communications (PWC'06),* LNCS Vol 4217, pp. 147-158, Springer-Verlag, Heidelberg, Germany, 2006.

[12] J. S. Baras and T. Jiang, "Managing trust in self-organized mobile ad hoc networks", in *Proceedings of the 12th Annual Network and Distributed System Security Symposium (NDSS '05), Invited Talk in the Wireless and Mobile Security Workshop*, February 2005, San Diego, California, USA.

[13] T. Repantis and V. Kalogeraki, "Decentralized trust management for ad hoc peer-to-peer networks", in *Proceedings of the 4th International Workshop on Middleware for Pervasive and Ad Hoc Computing (MPAC '06)*, p. 6, April 2006, Melbourne, Australia.

[14] C.Y. Tseng, P. Balasubramanyam, C. Ko, R. Limprasittiporn, J. Rowe, and K. Levitt, "A specification-based intrusion detection system for AODV, in *Proceedings of the 1st ACM Workshop on Security of Ad Hoc and Sensor Networks*, pp. 125-134, 2003.

[15] B-J. Chang, S-L. Kuo, Y-H. Linag, and D-Y. Wang, "Markov chain-based trust model for analyzing trust value in distributed multicasting mobile ad hoc networks", in *Proceedings of the IEEE Asia-Pacific Services Computing Conference (APSCC'08)*, pp. 156 – 161, 2008.

[16] Y.L. Sun, W. Yu, Z. Han, and K.J.R. Liu, "Information theoretic framework of trust modeling and evaluation for ad hoc networks", *IEEE Journal on Selected Areas in Communications*, Vol. 24, pp. 305 – 317, 2006.

[17] J. Sen, P. Roychowdhury, and I. Sengupta, "A distributed trust establishment scheme for mobile ad hoc networks", in *Proceedings of the International Conference on Computing: Theory and Applications*, pp. 51 – 58, March 2007, Kolkata, India.

[18] J. Sen, M. G. Chandra, P. Balamuralidhar, S.G. Harihara, and H. Reddy, "A distributed protocol for detection of packet dropping attack in mobile ad hoc networks", in *Proceedings of the International Conference on Telecommunications (ICT'07)*, Paper ID: 74, Track: Ad Routing and Protocol, Penang, Malaysia.

[19] H. Yang, J. Shu, X. Meng, and S. Lu, "SCAN: self-organized network-layer security in mobile ad hoc networks," *IEEE Journal on Selected Areas in Communications*, Vol 24, pp. 261-273, 2006.

[20] N. A. Benjamin and S. Sankaranarayanan, "Performance of wireless body sensor based mesh

network for health application", International Journal of Computer Information Systems and Industrial Management Applications (IJCISIM), Vol 2, pp. 20 – 28, 2010.

[21] B. Wang, S. Soltani, J.K. Shaprio, P-N. Tan, and M. Mutka, "Distributed detection of selfish routing in wireless mesh networks," *Technical Report – MSU-CSE-06-19*, Department of Computer Science and Engineering, Michigan State University, 2008.

[22] C.E. Perkins and E.M. Royer, "Ad hoc on-demand distance vector routing," in *Proceedings of IEEE Workshop on Mobile Computing Systems and Applications*, pp. 90-100, 1999.

[23] W.F. Eddy, A. Mockus, and S. Oue, "Approximate single linkage cluster analysis of large datasets in high dimensional spaces, *Computational Statistics and Data Analysis*, Vol 23, 29-43, 1996.

[24] H.J. Kim and J.M. Peha, "Detecting selfish behavior in a cooperative commons," in *Proceedings of IEEE DySPAN*, pp. 1-12, 2008.

[25] Network simulator: http://www.isi.edu/nsnam/ns.

Author Biography

Jaydip Sen is currently associated with the Innovation Lab of Tata Consultancy Services Ltd. In Kolkata, India, where he is leading the research and development activities in security and privacy in ubiquitous computing for the last four years. He has more than 15 years of experience in the field of networking, communication and security. Prior to joining TCS, he has worked with reputed organizations like Oil and Natural Gas Corporation Ltd., India, Oracle India Pvt. Ltd., and Akamai Technology Pvt. Ltd. His research areas include security in wired and wireless networks, intrusion detection systems, secure routing protocols in wireless ad hoc and sensor networks, secure multicast and broadcast communication in next-generation broadband wireless networks, trust and reputation-based systems, quality of service in multimedia communication in wireless networks and corss-layer optimization-based resource allocation algorithms in next-generation wireless networks, sensor networks and privacy issues in ubiquitous and pervasive communication. He has more than 80 publications in reputed international journals and referred conference proceedings and 10 book chapters in books published by internationally renowned publishing houses e.g., Springer, CRC press, IGI-Global etc. He has also delivered expert talks and keynote lectures in various international conferences and symposia. He is a member of the ACM and IEEE. He was also an active member of the security group of IEEE 802.16 standard body where he proposed a few standards contributions. His biography has been listed in Marquis Who's Who in the World, 2009 and 2010 editions. Dr. Sen obtained his Bachelor of Engineering (B.E.) in Electrical Engineering with honours from Jadavpur University, Kolkata, India 9n 1993, Master of Technology (M.Tech) in Computer Science with honours from Indian Statistical Institute, Kolkata, India in 2001, and PhD from Indian Institute of Technology, Kharagpur, India in 2007.