Distributed Self-policing Architecture for Fostering Node Cooperation in Wireless Mesh Networks

Lakshmi Santhanam, Nagesh Nandiraju, Younghwan Yoo,
and Dharma P. Agrawal

OBR Center for Distributed and Mobile Computing, Department of ECECS
University of Cincinnati, Cincinnati, OH 45221-0030
Email: {santhal, nandirns, ymomo, dpa}@ececs.uc.edu

Abstract. Wireless Mesh Networks (WMNs) are evolving to be the key technology of the future. The self-configuring nature of WMNs and the ease, with which a mesh router/mesh point can be added, makes it pertinent to ensure their secure operation. All the routing protocols in WMNs naively assume the nodes to be co-operative in forwarding each other's packets. However, a node can behave selfishly by discretely dropping other's packets, in an attempt to maximize its throughput. In this paper, we present a distributed scheme called, Distributed Self-policing Architecture for Fostering Node Cooperation (D-SAFNC), for enforcing cooperation among the nodes in a WMN. We use a distributed approach in isolating any selfish node with the help of localized detection agents called sink nodes. We study the effectiveness of our scheme through simulations using ns-2 which reaffirm that D-SAFNC can successfully prevent any performance degradation due to the presence of selfish nodes.

Keywords: Free riders, Mesh networks, Node Misbehavior, Selfish Nodes.

1 Introduction

Recent years have witnessed a rapid evolution of Wireless Mesh Network (WMNs), as seen by the surge in its popularity surpassing well known peer technologies. Since its inception, it has become the limelight of all researchers. Nokia's Rooftop Mesh [1], MIT's roofnet [4], Radiant Networks [3] are some known efforts in this direction.

A WMN excels in performance by providing seamless broadband connectivity [12], when compared to other peer technologies such as cellular and WLAN. A cellular network offers wide area coverage, but provides low channel capacity (at best 3Mbps in 3-G and at best 100 Mbps in 4-G); while the WLANs 802.11 network has an attractive high bandwidth connectivity (802.11g currently in user at 54 Mbps and 802.11n with a theoretical throughput of 540Mbps) but with a very limited range.

A WMN is formed by a set of Access Points (a.k.a mesh routers) connected wirelessly, among which a small subset called the Internet Gateway (IGW), is directly connected to the internet. These mesh routers cooperatively forward each other's

packets with an underlying ideology of “using” and “providing” service. This kind of cooperative behavior helps in extending the network coverage without any additional infrastructure. The salient characteristics of WMN include: *scalability, self-healing, and self-configurable capability.*

Although, the notion of ad hoc networking facilitates the plug-and-play architecture there by increasing flexibility, it also increases the vulnerability of the network. A selfish or malicious user can add a rogue Mesh Router (MR) to the network and can start disrupting the network services. Such intermediate router can behave selfishly, by discreetly dropping other’s packets and forwarding only its own traffic. A selfish node might not forward another node’s traffic with an objective to maximize its throughput. We also call such a node “*free-rider*”, as it enjoys network resources without contributing to the community. It is even more precarious if a selfish node is located near the IGW as these nearby nodes are mainly in-charge of forwarding the bulk of traffic in a WMN. This would inordinately affect the multihop flows traversing from distant sources and result in wastage of network resources and cause total havoc to the system.

In order to maintain the system integrity, it is evident all the nodes should cooperatively forward each other’s traffic. Authenticating a node is not a complete solution as an intruder could still capture a legitimate node or a legitimate node could later on turn selfish. Hence, we propose a novel distributed self policing architecture to detect such selfishly behaving mesh routers in a WMN. We employ special agents called *sink nodes* that are delegated the duty of policing their local neighborhood to detect free-riders. On identifying *free-rider(s)*, *sink nodes* trigger a system wide alert, instructing rest of the nodes to take preventive measures by quarantining the defaulting nodes. It is quite possible that a free-rider might attempt to accuse an innocent node. Our system can elegantly detect such false accusations by observing the system behavior over a period of time and using an additive increase-multiplicative decrease scheme to relieve the innocent node. Simulation results show that D-SAFNC effectively discourages selfishness by taking timely action against free-riders and fosters cooperation.

The remainder of this paper is organized as follows. We discuss the related work on detecting selfish nodes in multihop ad hoc networks in Section 2, followed by an outline of the assumptions, design goals and challenges in Section 3. We then describe the implementation of the proposed D-SAFNC scheme in Section 4 and present an analysis of its complexity in Section 5. Section 6 discusses the performance evaluation of our scheme. We finally conclude with a summary of the work in Section 7.

2 Related Work

Discouraging selfishness in MANETs (Mobile Ad Hoc Networks) has been widely studied. They adopt either credit-based or reputation-based or game theory based approaches. But, these schemes cannot be directly adopted for WMNs due to several differences in their design. First, WMNs are capable of employing multi-radio multi-channel for simultaneous transmission and reception as a result of which promiscuous

listening based reputation scheme cannot be applied. Second, WMNs are relatively static unlike MANETs and hence a credit based scheme fail. Third, the traffic in a WMN is oriented either to or away from the IGW.

In a credit-based scheme (like Nuglets [6], Sprite [16], and PIFA [15]), each node earns virtual currency by forwarding others packets so that they can originate their own packet. They require a tamper-resistant hardware for the authenticity of the currency or depend on a centralized credit agency to allocate wealth. A central authority is vulnerable to single point of failure as it is overloaded with report messages from all the nodes in the network.

In a reputation based approach, each node promiscuously eavesdrops on the neighboring node's transmission and assigns ratings to each other. The rating is then incorporated by other nodes during their route selection process. Watch-dog and Path-rater model [11] find the selfish nodes by a reputation mechanism. However, it does not take any action against the traffic of a selfish node. Such a neighborhood watch scheme is also prone to a replay attack. CONFIDANT [5] [14] uses a path manager that ranks the paths based on the intermediate nodes along the path, eschewing the selfish node. As the reputation spreads by global flooding, it faces scalability issues. Both schemes, fail to differentiate collision and misbehavior. Game theory approaches fix the forwarding rate of a node at certain Nash equilibrium for the network as in Generous Tit-for-Tat (GTFT) [14], but are realistically infeasible.

CATCH [9] is a distributed scheme for multi-hop wireless network that combines anonymity and Watch-dog approach in detecting free-riders. All nodes broadcast an anonymous message. As the selfish node is unaware of the sender's identity, it is forced to forward all of them dutifully to stay connected. If not, it would risk being isolated from the network. However, this scheme is inapplicable for a WMN employing multi-radio communication, as promiscuous eavesdropping would not be always possible. It also requires each node to possess large memory to store the unsent packets when a neighbor does not forward them. In contrast to all the above schemes, our proposed scheme entails lesser memory overhead due to 2-hop information sharing.

3 Assumptions, Design Goals & Challenges

In this section, we first outline our assumptions, enlist the envisioned goals of D-SAFNC and finally discuss the challenges involved in realizing our goals.

3.1 Assumptions

- We assume that a scheme like ingress filtering can be used to prevent source address spoofing.
- We host sink agents on certain trustworthy mesh routers such that each every node is within the 2-hop neighborhood of a sink agent.
- We assume there is no collusion among selfish nodes. A selfish node is different from a malicious node. A malicious node disrupts the network activity by

collusion. In contrast, a selfish node does not gain anything by disrupting the network (in fact by doing so it will defeat its purpose). Its greedy intention to devour all the network resources for itself results in its solitary operation. Hence, this is a safe assumption.

3.2 Design Goals and Challenges

The main design goal of our scheme is to accurately identify selfish nodes and give them a second chance to re-socialize in the network. We target to give each node a fair chance to originate and immediately transmit packet, irrespective of its geographic location. We stay clear of using a credit or reputation based scheme because of the following inherent flaws [8] [10]:

1. In a credit based scheme, a node positioned in the periphery of the network is handicapped and fails to earn credits as it is not used as an intermediate hop by other nodes. This is very much likely in a WMN wherein nodes located near the IGW forward more data than those at the periphery.
2. As the nodes are not allowed to send traffic until they earn enough credits, it is unsuitable for real-time voice and video applications like VoIP, video conferencing, and video surveillance. When the nodes have insufficient credit, they have to either buffer or drop the unsent traffic, until they earn sufficient credits. This causes undue latency in the packet delivery.
3. In a credit scheme, an egregious node might begin its selfish activity after accumulating enough credits which is counter-intuitive to the goal of the scheme.
4. Most of the credit/reputation schemes are applicable only to a source routing protocol as it needs to determine the credits to be loaded in the packet for transmission.
5. All reputation based schemes require a way to build a reliable mutual trust index by monitoring the network activity. This is in general accomplished by listening to neighboring node's transmissions. However this assumption does not hold well in asymmetrical link [7], and systems with directional antennae [13] or a WMN using multi-radio multi-channel capable nodes (if non-interfering channels are assigned to adjacent nodes).

The aforementioned disadvantages render these approaches impractical for promoting cooperation in WMNs. Thus, we focus on developing a distributed monitoring scheme. As there is a possibility to misclassify a genuine packet loss as misbehavior, it is important to monitor the node behavior over a significant period of time. Moreover the scheme should be resilient to member report losses.

4 Proposed D-SAFNC Scheme

From the discussions in the previous section, we realize that a pervasive solution is essential for monitoring WMN. We start with an overview of the system environment and architecture of D-SAFNC scheme and then proceed to the details of the scheme.

4.1 System Environment and System Architecture

We consider a static framework of interconnected nodes forming a mesh topology that provides wireless internet service in an office or a university as shown in **Fig. 1.** In order to facilitate simultaneous communication with the end users and other mesh routers, we assume that each mesh router has at least two interfaces operating on non-interfering channels.

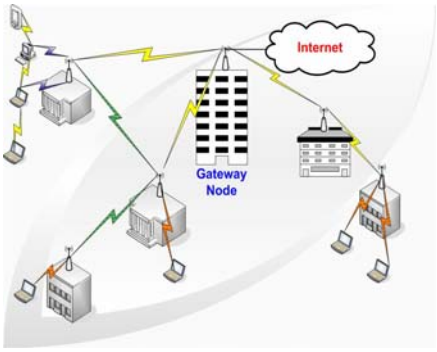

Fig. 1. A WMN in a University

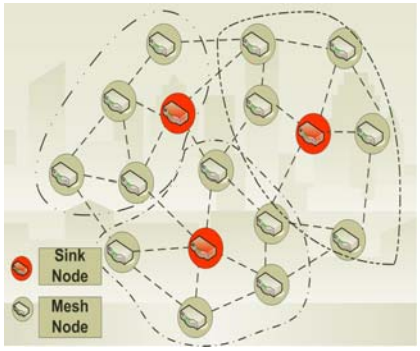

Fig. 2. Monitoring using Sink Nodes

We propose a distributed scheme, D-SAFNC; which helps in detecting free-riders by deploying sink agents at about 10% of the routers in the network. System monitoring is divided into two phases: Start-up phase and Monitoring phase. In the *start-up phase*, the sink agents as shown in **Fig. 2** advertise their presence by sending periodic beacons. Each mesh router upon receiving a beacon registers itself under the sink in two cases:

- If it has not already selected any sink or
- If the new sink is nearer than the previously registered sink.

To regulate the flooding of beacons each mesh node rebroadcasts the beacon only if the hop-count is less than BEACON_MAX_HOPS. In the *monitoring phase*, once all the mesh routers are aware of their respective sinks, they send periodic reports to the sinks every REPORT_ROUND time. Unlike SPRITE [16] that sends a report for every forwarding message, this aggregated scheme saves considerable overhead in terms of network bandwidth and payload. The periodic report consists of information on the number of packets received and forwarded by a node during a certain interval of time. Table 1 gives a brief definition of each field in the report message.

At end of REPORT_ROUND time, each sink applies the following three checkpoints. First, it computes a simple check between the output of a node and the input registered at its neighbor, as given in Equation (1). It checks for every link if the output from a node is same as the input at its neighbor. Let A_j and A_k denote the set of neighbors of node j and k respectively. If j and k are two neighboring nodes, then

$$O_{j,k} = I_{k,j} (j \in A_k, k \in A_j) \quad \textbf{(1)}$$

Where $O_{j,k}$ and $I_{k,j}$ denote the *O* and *I* fields of a message report whose IDR and IDN are *j* and *k*. This computation prevents any node from dropping packets.

Second, it checks if *S*, if the number of packets originating at current node *j* as reported by $node_j$ is equal to *NON* i.e. number of packets that originated at node *j* among input packets from node *j* to node *k* as reported by $node_k$. This check prevents a node from misreporting the number of packets that are originating from a given node.

$$S_{j,k} = NON_{k,j} (j \in A_k, k \in A_j) \quad \textbf{(2)}$$

The third and final check, finds the number of packets forwarded F_j by a node *j* using the two formulae and compares them. Equation (3) computes the total number of packets forwarded by a node *j* to all its neighbors (excluding its own packets). Equation (4) computes the packets forwarded by node *j* based on its neighboring node's (node *k*) reports i.e. the difference between total number of input packets to a node *j* and the total number of packets terminating at node *j*. Equation (3) and (4) should be equal to ensure that a selfish node does not manipulate the value of *S* or *O*.

$$F_j = \sum_{k \in A_j} O_{j,k} - \sum_{k \in A_j} S_{j,k} \quad \textbf{(3)}$$

$$F_j = \sum_{k \in A_j} I_{j,k} - \sum_{k \in A_j} NTC_{j,k} \quad \textbf{(4)}$$

There is a possibility of packet loss occurring due to interferences/channel degradation/queuing overflows in a wireless channel which should not be misinterpreted as selfish behavior. Hence, when the three checkpoints are applied, we always consider maximum permissible packet drop for a given network condition.

Table 1. Format of Report Messages

IDR	ID of the reporting node
IDN	ID of the neighboring node
IDS	ID of the node's registered local sink
SEQ	Sequence number of the node's report for synchronizing member reports at the sink
I	No. of input packets from the neighbor
O	No. of output packets to the neighbor
S	No. of packets originating at current node among the output packets to the neighbor
NON	No. of packets that originated at the neighbor among the input packets from the neighbor
NTN	No. of packets terminated at next hop (at IDN) among the packets sent from IDR to IDN.
NTC	No. of packets terminating at this current node (at IDR)

When a new node joins the network, it first registers itself to its nearest sink node and then places a request to the sink. The sink replies to the new node with the current sequence number being used, reply time and REPORT_ROUND time. The new node computes the new sequence number as given by Equation (5).

$$New\_SEQ = SEQ + [\frac{Current\_time - Reply\_time}{REPORT\_ROUND}] \quad \textbf{(5)}$$

4.2 Free Rider Detection Algorithm

The system runs a *free-rider detection algorithm* at every CHECK_ROUND time (= 4 * REPORT_ROUND time) that accurately identifies and punishes the free-rider. After applying the three checkpoints on its member reports, each sink checks if reports from two adjacent nodes do not accord with each other. If so, the node and the neighbor involved in the transaction is added to a NAM (Number of Alleged Manipulation) list maintained at the sink.

Looking at the inconsistencies at a single sink node, the identity of the free-rider is unclear, as member nodes involved in the alleged manipulations might belong to the same domain (intra) or a different domain (intra). Hence, one sink node is chosen as a sink manager (SM) to which all other sinks nodes unicast their NAM list. A master NAM list is created at the SM. As only the suspicious node list is passed to the SM, D-SAFNC when compared to a completely centralized scheme incurs lesser overhead in evaluating reports and lesser congestion, at each sink.

Using the master NAM list, the SM then builds an Inconsistency Record Table (IRT) as shown in Table 2. Each entry $m_{a,b}$ in IRT represents the number of alleged manipulations in the packet transmission between node *a* and *b*. The last column denotes the total NAM values for each node. If this is greater than a certain threshold (UT_PERMISSIBLE_MANIPULATIONS- *for a node not in blacklisted history*) *and* (LT_PERMISSIBLE_MANIPULATIONS- *for a node in blacklisted history*), this node is blacklisted and added to a blacklisted node history. Each entry $m_{a,b}$ is incremented in the IRT by an additive increase and multiplicative decrease algorithm. This is done so that an innocent node is not unduly framed and punished. For example, if there is an inconsistency between node a's and b's report, we increase the NAM values given by Equation (6).

$$m_{a,b} = m_{a,b} + 1 \text{ and } m_{b,a} = m_{b,a} + 1 \quad \textbf{(6)}$$

As other nodes involved in a transaction with a or b might be penalized, the $m_{i,a}$ and $m_{i,b}$ values of other nodes are reduced by half given by Equation (7).

$$m_{i,a} = \frac{m_{i,a}}{2} \forall\ i \notin \{a, b\} \text{ and } m_{i,b} = \frac{m_{i,b}}{2} \forall\ i \notin \{a, b\} \quad \textbf{(7)}$$

Table 2. Inconsistency Record Table

	A	B	C	…	Total
A	-	$m_{a,b}$	$m_{a,c}$	…	$\sum m_{a,i}$
B	$m_{b,a}$	-	$m_{b,c}$	…	$\sum m_{b,i}$
C	$m_{c,a}$	$m_{c,b}$	-	…	$\sum m_{c,i}$
…	…	…	…	…	…

Once a blacklisted node is detected, the SM announces it to the entire network. Upon receiving this message, the nodes that have route through this blacklisted node invalidate their entries and take appropriate re-routing action. In AODV, this can be either performing a local repair or sending a route error to the source (RERR). Thus the affected nodes now reroute their traffic through alternate paths.

Selfish behavior is discouraged as all the legitimate nodes collectively refuse to forward any traffic originating from the blacklisted node. SM maintains the list of all previously blacklisted nodes along with the observed time of its misbehavior. A free-rider is not permanently blacklisted; instead its isolation is associated with a timer (FORGIVEN_TIME). On the expiration of this timer, the system temporarily pardons the isolated node to give it a second chance. The other nodes henceforth resume routing through this node. If the node begins its selfish activity at any time in the future and is found in the NAM list, its threshold for IRT table computation is lowered to LT_PERMISSIBLE_MANIPULATIONS as a precautionary measure. Using the IRT computation, if it is found to default again, it is *permanently blacklisted.* Other nodes permanently shun any traffic originating from this node and never consider routing through this blacklisted node. Thus transient liars that oscillate between good and bad behavior are successfully caught and punished.

5 Complexity Analysis

In this section, we analyze the message complexity of the proposed D-SAFNC. There are two kinds of messages: one is the report to the local sink from a WMN nodes and the other is the inconsistency information to the SM from local sinks. However, since the amount of the inconsistency information is just equal to the number of sink agents, this inconsistency information is not a large overhead if we assume optimal minimum number of sinks in a WMN. Thus, the analysis focuses on the member reports submitted periodically every REPORT_ROUND to the local sink agent.

The notations are as follows:
A: total area of a WMN
N: total number of WMN nodes
r: transmission range of each node
s: number of sink agents
c: number of nodes associated with one sink agent
b: number of neighboring nodes of a WMN node

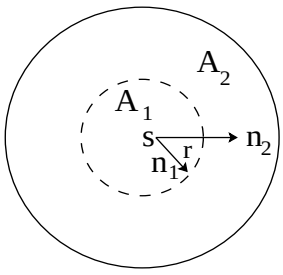

Fig. 3. Coverage of One Sink Node

We note that the number of sink agents, s, is determined so that every WMN node may reach at least one sink within two hops. Assuming all WMN nodes are uniformly distributed, the area a sink can maximally cover is $4 \pi r^2$ as shown in **Fig. 3.**

Hence, at least $\left\lceil \frac{A}{4\pi r^2} \right\rceil$ sink agents are needed. Considering minimum number of sink nodes, the number of WMN nodes a sink should manage is $c = \frac{4\pi r^2 N}{A}$. Each WMN node sends reports for every neighbor, and the average number of neighbors of one given node in a uniformly distributed network is $b = \frac{\pi r^2 N}{A} - 1$. Hence, each sink agent receives as many reports as cb every REPORT_ROUND, and the total number of reports in a WMN using D-SAFNC is given by following Equation (8):

$$scb \geq \left\lceil \frac{A}{4\pi r^2} \right\rceil \cdot \frac{4\pi r^2 N}{A} \left(\frac{\pi r^2 N}{A} - 1 \right) \quad \textbf{(8)}$$

Meanwhile, the total hop count of report messages can be computed as follows. In **Fig. 3**, as A_2 is three times wider than A_1, we can safely assume that A_2 includes three times as many nodes as A_1. Since a node in A_1 and A_2 can reach the sink with one hop and two hops respectively, the average hop count, is $h = (1 \times 1 + 2 \times 3)/4 = 1.75$. Thus, the total hop count required for report messages is $1.75scb$.

6 Performance Analysis

In this section, we evaluate the performance of D-SAFNC using ns-2 simulator [1]. Although D-SAFNC can be run on top of any underlying routing protocol, we choose AODV as the routing protocol. We consider a network of 25 mesh points in a 5x5 grid (shown in **Fig 4(a)**) spread over an area of 1500m x 1500m. IEEE 802.11 is used for channel arbitration with the transmission range and channel capacity set to 250 m and 11 Mbps respectively. The total simulation time is set to 200 seconds. We set D-SAFNC specific parameters as follows: CHECK_ROUND (28 sec), REPORT_ROUND (7 sec), LT_PERMISSIBLE_MANIPULATIONS (1 sec), FORGIVEN_TIME (14 sec), BEACON_MAX_HOPS (2), and UT_PERMISSIBLE_MANIPULATIONS (3 sec).

6.1 Instantaneous Throughput

To evaluate the effectiveness of our scheme in the presence of selfish nodes, we study the fluctuations in the instantaneous throughput of the flows. We start two flows (Flow 1 and 2) in both directions between mesh routers MR 0 and MR 20 (which is an Internet GW) as shown in **Fig. 4(a)** at time equal to 1 second. We place a selfish

node (MR 10 in **Fig. 4(a))** in the shortest path between the two nodes. At time 10 seconds, we start a traffic flow from this misbehaving node to the IGW. As seen from the **Fig. 4(b)** and 4**(c)** during the time period 1-30 seconds, both the flows from the good nodes (MR 0 & MR 20) suffer from 100% packet loss as they choose their routes through the selfish node (MR 10).

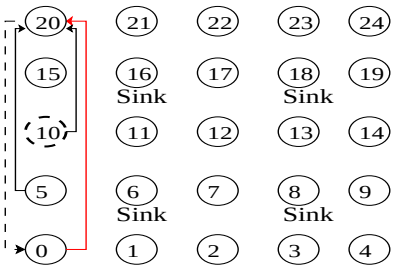

Fig. 4(a) Grid Network

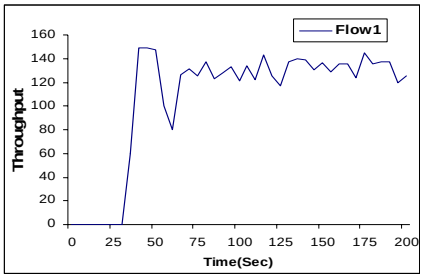

Fig. 4(b) Flow 1 (between 0-20)

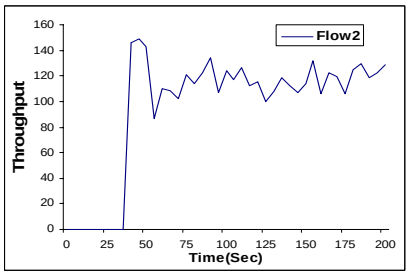

Fig. 4(c) Flow 2 (between 20-0)

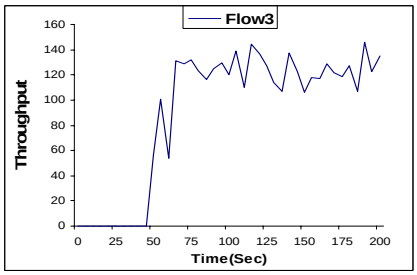

Fig. 4(d) Flow 3 (between 5-20)

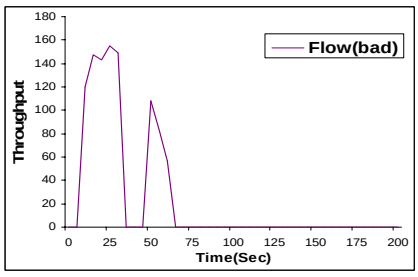

Fig. 4(e) Bad Flow (between 10-20)

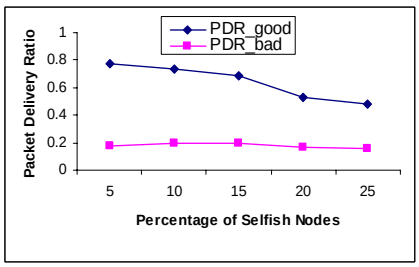

Fig. 5. PDR vs. % of selfish nodes

On the other hand, the flow from MR 10 enjoys good throughput, **Fig. 4(e)**. However this free-riding does not continue for a long period of time. After four

rounds (nearly 30 seconds) of continued misbehavior by MR 10, the SM confirms MR 10 as a free-rider and broadcasts this information to the entire network. Thus, MR 5 and MR 15 which have active routes through MR 10 purge their routing entry and re-route their traffic. This is clearly illustrated from the **Fig. 4(b)** & **(c)** during the time period 40-200 seconds. At the same time, neighbors of the selfish node (MR 5 and MR 15) stop forwarding any traffic originating from MR 10 and thus the flows from MR 10 are shut albeit for a short period of time (FORGIVEN_TIME).

In order to illustrate the punishment for prolonged misbehavior, we start another flow from MR 5 towards the IGW (Flow 3 in Fig. 4(d)) shortly after the selfish node is forgiven. MR 5 will now consider routing its traffic through MR 10 as it is on its shortest path to MR 20. However as MR 10 continues its misbehavior, flow from MR 5 suffers 100% packet loss. This can be seen during the time period 45-65 seconds in **Fig. 4(d)**. The SM quickly identifies the misbehavior of MR 10, permanently blacklists it and then notifies to the entire network. MR 5 now tries to reroute its traffic through an alternate route. From this point on, MR 10 which is permanently blacklisted will not be able to route any traffic in the network. This can be seen from the **Fig. 4(e)** during the time period 75-200 seconds.

6.2 Packet Delivery Ratio

We now evaluate the effectiveness of D-SAFNC in the presence of multiple selfish nodes. We measure the Packet Delivery Ratio (PDR), which is the ratio of the number of packets received at the destination to the number of packets generated at the source.

We randomly pick different source MRs and IGWs and start traffic from these nodes. **Fig. 5** shows the PDR of the good and bad nodes for varying percentage of selfish nodes. As can be seen from **Fig. 5**, D-SAFNC ensures that PDR of good nodes is well maintained while considerably throttling the PDR of bad nodes. Even though good nodes may occasionally loose packets because of the presence of selfish nodes in their path, they quickly recover and try to reroute the traffic, consequently maintaining a steady PDR. As D-SAFNC gives a second chance to the misbehaving node, the PDR of free-riders is low but non zero as indicated by the PDR_bad plot in **Fig. 5**.

PDR of good nodes decreases as we increase percentage of selfish nodes. This is because as we increase the number of selfish nodes we also increase the traffic flows as a result increasing the load on the network. Also, packets from good nodes experience some loss during re-routing process, as now they take longer hops to reach their destination. However, D-SAFNC prevents the PDR of good nodes from dropping below 50% even when 25% of the nodes are selfish.

7 Conclusion

Mesh networks are continuously gathering momentum in its evolution in the wireless industry which also raises several security concerns. We highlighted the

inadequacy of credit/reputation based schemes in promoting cooperation in a WMN and presented a distributed policing architecture. As the information sharing of member reports is restricted to a two hop neighborhood, it has considerably less overhead as compared to a centralized scheme. These are fortified by the simulation results which indicate that D-SAFNC increases the throughput of the system. The system tries as much as possible to re-accommodate even the past misbehaving nodes and this way fosters cooperation among the mesh routers. In our future work, we plan to implement the scheme using multi-channel multiple interface architecture such that backhaul links of different frequency are for sending reports to the sink.

Acknowledgement. This work has been partially supported by the Ohio Board of Regents, Doctoral Enhancement Funds.

References

[1] Network Simulator (NS-2), http://www.isi.edu/nsnam/ns/index.html.
[2] Nokia RoofTop Wireless Routing. White paper.
[3] Radiant Networks Website – www.radiantnetworks.co.uk
[4] Aguayo, D., Bicket, J., Biswas, S., Judd, G., Morris, R.: Link-level Measurements from an 802.11b Mesh Network. In: the Proc. of SIGCOMM. (2004)
[5] Buchegger, S., Boudec, J.-Y. L.: Performance analysis of the CONFIDANT protocol: Cooperation of nodes- fairness in dynamic ad-hoc networks. In: the Proc of Mobi-HOC. (2002)
[6] Buttyan, L., Hubaux, J.-P.: Enforcing Service Availability in Mobile Ad-Hoc WANs. In: the Proc of IEEE/ACM MobiHOC Workshop. (2000)
[7] De Couto, D., Aguayo, D., Bicket, J., Morris, R.: A High-Throughput Path Metric for Multi-Hop Wireless Routing. In: the Proc. of ACM MobiCom. (2003)
[8] Huang, E., Crowcroft, J., Wassell, I.: Rethinking incentives for mobile ad hoc networks. In: the Proc. of ACM SIGCOMM PINS. (2004)
[9] Mahajan, R., Rodrig, M., Wetherall, D., Zahorjan, J.: Sustaining Cooperation in Multi-Hop Wireless Networks. In: the Proc. of NSDI. (2005)
[10] Mahajan, R., Rodrig, M., Wetherall, D., Zahorjan, J.: Experiences Applying Game Theory to System Design. In: the Proc. of ACM SIGCOMM. (2004)
[11] Marti, S., Giuli, T.J., Lai, K., Baker, M.: Mitigating router misbehavior in mobile ad-hoc networks. In: the Proc. of Mobi-Com. (2000)
[12] Poor, R., Corp, E.: Wireless MESH Network. In: Intelligent System – Wireless. (2003)
[13] Saha, A.K., Johnson, D.B.: Routing improvement using directional antennas in mobile ad hoc networks**.** In: the Proc. of IEEE GlobeCom, Vol.5. (2004) 2902 – 2908
[14] Srinivasan, V., Nuggehalli, P., Chiasserini, C.F., Rao, R.R.: Cooperation in wireless ad hoc networks. In: the Proc. of IEEE INFOCOM. (2003)
[15] Yoo, Y., Ahn, S., Agrawal, D.P.: A Credit-Payment Scheme for Packet Forwarding Fairness in Mobile Ad hoc Networks. In: the Proc. of IEEE ICC. (2005)
[16] Zhong, S., Yang, Y., Chen, J.: Sprite: A simple, cheatproof, credit-based system for mobile ad hoc networks. In: the Proc. of IEEE INFOCOM. (2003)