Wireless Pers Commun (2015) 82:689–708
DOI 10.1007/s11277-014-2247-4

Opportunistic Routing in Presence of Selfish Nodes for MANET

Sandeep A. Thorat · P. J. Kulkarni

Published online: 6 January 2015
© Springer Science+Business Media New York 2015

Abstract Opportunistic Routing protocols use broadcast nature of wireless communication to improve packet delivery from source to destination in mobile ad hoc network (MANET). In traditional routing protocols for MANET, each node uses the best neighbor for forwarding packet to destination. In opportunistic routing, a node selects and prioritizes multiple nodes which can act as potential packet forwarders. Similar to traditional routing protocols, opportunistic routing assumes that all nodes participating in the network are honest and cooperative. However, this is practically difficult in an open MANET. Few nodes participating in the network may behave selfishly; these nodes drop packets and do not offer service to other nodes. Such behavior exhibited by the nodes may cause a collapse of MANET communication. The selfish behavior attack is addressed by researchers for many existing MANET routing protocols. There are just a few works, which address selfish nodes attack for opportunistic routing protocols. The paper proposes an opportunistic routing protocol which can overcome the presence of selfish nodes. The proposed protocol discovers reliable candidate nodes for packet forwarding using a new metric called 'path goodness'. The path goodness metric takes into account trustworthiness of the nodes on a path to the destination. The protocol decides trustworthiness of the nodes using packet forwarding behavior of the nodes. The work is a trust based extension of CORMAN opportunistic routing protocol. Simulation results show that, the proposed opportunistic routing protocol improves the packet delivery ratio by approximately 10 % in the presence of selfish nodes.

Keywords Mobile ad hoc network · Routing protocols · Opportunistic routing · Candidate selection algorithm · Selfish nodes · Trust based routing

S. A. Thorat (✉)
Rajarambapu Institute of Technology, Sangli, India
e-mail: sathorat2003@gmail.com

P. J. Kulkarni
Computer Science Department, Walchand College of Engineering, Sangli, India
e-mail: pjk_walchand@rediffmail.com

1 Introduction

Routing is a fundamental operation in mobile ad hoc network. Lack of infrastructure, dynamic links, and broadcast nature of the communication makes routing in MANET a challenging problem. Traditional routing protocols for MANET like AODV [1], DSR [2], AOMDV [3] perform best path routing, which work similar to routing in the wired networks. Best path routing selects the best neighbor from each hop to forward a packet. This strategy has limitations in dynamic wireless environment due to volatility of transmission links between the nodes. A distinctive feature of wireless communication is the broadcast nature. When a node sends a packet to its neighbor, each one-hop neighbor listens the packet. As it causes interference with other communications, this is considered as a disadvantage in traditional routing protocols. Opportunistic Routing (OR) uses the broadcast nature of the communication by dynamically selecting route to the destination. In OR, each transmission of a packet is for multiple neighboring nodes. OR considers multiple neighbors as potential candidate nodes, which can forward the packet further toward the destination. The candidate nodes attempt to forward the received packet as per the priority order decided by the previous sender. The OR causes decrease in the number of transmissions required to send a packet from source to the destination [4,5].

Du et al. [6] lists benefits of OR are as follows. OR combine multiple weak links into one strong link. As all possible links within one transmission are considered, OR may use the farthest hop which successfully receives the packet as the next packet forwarder. Hence it can take advantage of unexpectedly long transmissions. Hsu et al. [7] point out that, OR can use backup links and it minimizes transmission failure probability. This improves reliability of the communication. Experiment results in [4] and [8] show that OR has the potential to perform better than traditional routing protocols.

MANET routing protocols conventionally assume that, every node participating in the communication is honest and cooperative. This is also applicable to OR protocols. Hence routing is successful only if the participating nodes cooperate with each other. It is impractical to assume that, all nodes participating in the network are cooperative and honest every time. Few nodes participating in the network may be selfish nodes. Selfish nodes drop packets for conserving own battery and processing power, or these nodes are interested to disturb the communication. Few nodes participating in the network may be faulty, which again causes packet drops. Packet dropping attack is a serious issue in MANET routing, and it may result into the collapse of network. Hence designing a routing protocol which can overcome the presence of selfish nodes and improve network performance is important [9].

Using trustworthiness of nodes for decision making in routing has recently gained a large amount of attention. The [10–12] are few trust based routing protocols, which aims at identifying selfish nodes and neutralize their impact in the routing. These algorithms optimize the network performance by utilizing trustworthy nodes in an effective way. The presence of selfish nodes also impact on performance of the OR protocols. However, there is limited work done which addresses the presence of selfish nodes in the OR protocols.

The paper proposes a novel trusted OR protocol, which overcomes presence of selfish nodes in the network. The design of the proposed algorithm is inspired from CORMAN [8] and is a vital extension of CORMAN. The algorithm evaluates the path goodness value for each path towards the destination. The path goodness value is derived from trustworthiness of the nodes on the path and proximity of these nodes towards the destination. The algorithm uses a trust model based on packet forwarding behavior of the nodes. The path goodness value is used as metric for deciding and prioritizing candidate nodes participating in OR.

The proposed algorithm is termed as Opportunistic Routing in Presence of Selfish Nodes (ORPSN).

The main contributions of this paper are as follow:

1. ORPSN protocol is proposed, the protocol overcomes the presence of selfish nodes in OR.
2. A trust model suitable for OR is devised. Each node uses the trust model to calculate its neighbor's trustworthiness.
3. A new metric called 'path goodness' is devised for selecting and prioritizing candidate nodes participating in OR.
4. The work shows that, OR protocols have better immunity against selfish nodes as compared to traditional routing protocols.

The paper presents performance comparison of AODV [1], CORMAN [8], DSR [2], FTDSR [11] and ORPSN routing protocols in the presence of selfish nodes. The experimental results show that, ORPSN has better packet delivery ratio compared to CORMAN in the presence of selfish nodes. ORPSN enhances the reliability of packet forwarding and minimizes the threats from the selfish nodes.

The paper is organized as follows. Section 2 discusses the related work. Section 3 gives an overview of ORPSN and mathematical modeling of the protocol. Section 4 discusses algorithms and data structures used in ORPSN. In Sect. 5, simulation results are shown to evaluate the performance of proposed protocol. Section 6 concludes the paper.

2 Related Work

Extremely opportunistic routing (ExOR) [4] protocol is considered as the first OR protocol. In ExOR, a node broadcasts data packets to neighbors for packet delivery to the destination. Each node finds priorities of next potential forwarders and announces these priorities in the broadcasted packet. ExOR uses MAC and routing layer together to avoid transmissions of same packet from multiple nodes. The algorithm imposes strict timing constraints among the forwarders for coordination in the packet relay process. ExOR followed by a number of OR proposals by different researchers in the last decade. OR protocol design research is dominated by two issues. The first one is making proper choice of candidate nodes; and second is improving performance of OR protocols [7].

Candidate selection and ordering help to find the best route. The nodes use different metrics for candidate selection and prioritization. The traditional routing algorithm like AODV, DSR and AOMDV uses hop-count as metric for candidate selection. ExOR [4], SOAR [13] and MCOR [14] protocols use the Expected Transmission Count (ETX) as a metric for ordering candidate set. In these protocols, sender node prefers a node having minimum ETX towards the destination. The ETX parameter prefers the farthest node as a candidate node. Zhong et al. [15] uses EAX as parameter for choosing the best candidate node to the destination. EAX is required number of any-path transmissions needed to deliver a packet between two nodes. EAX is better than ETX for prioritizing candidate nodes, but it causes higher computational overhead.

Zeng et al. [16] proposed opportunistic effective one-hop throughput (OEOT) metric for candidate selection in a geographic OR protocol. The OEOT implements one hop packet transfers with packet forwarding time using different data rates. Du et al. [6] proposed throughput oriented forwarder selection (TOFS) metric for candidate selection. TOFS works by considering packet forwarding behavior of the nodes. The algorithm discards certain nodes as being a candidate node by applying condition on throughput. It helps in achieving effective link stability using minimum candidate nodes. Zuo et al. [17] and Mao et al. [18] present an energy-efficient OR strategies. The EEOR [17] prioritizes the candidate nodes according to reduction achieved in total energy cost of forwarding data to the sink node. Shen et al. [19] applies OR for improving communication in Wireless Multimedia Sensor Networks (WMNS). The algorithm selects and prioritizes candidate nodes to achieve an energy-efficient delivery of video data under QoS constraints. Li et al. [20] compared different candidate selection strategies. The authors conclude that, if the candidate sets are updating frequently then fast and simple candidate selection algorithms are better.

Another motivation behind the design of OR protocol is to improve reliability or efficiency of routing. Duplicate packet forwarding in OR reduce efficiency as same packet is forwarded by many nodes. Myung et al. [21] propose a Duplicate-Free opportunistic packet forwarding protocol (DFOR). In DFOR forwarding nodes minimize packet broadcasts for each packet. Zuo et al. [17] uses energy-consumption based objective functions (OF) to calculate energy consumption by all nodes on the path to the destination. The [17] uses information exchange at physical, MAC and network layer for OF calculation. Each node trying to find the best node for packet forwarding causes heavy computational overhead in working of OR [22]. CORP-M [22] do not use the pre-selected list of potential forwarders. The protocol divides the network into different regions by allowing a node to calculate its own region. Then these regions are taken into consideration for packet forwarding.

The secure routing algorithms in MANET uses cryptography or trust based mechanisms for improving security of communication. Both approaches have their own merits and limitations. Few examples of MANET routing protocol using cryptography is SDMP [23], ALARM [24] and TEAP [25]. Cryptographic methods need pre-establishment of the keys between the nodes participating in communication. So it may require setting up a Certificate Authority or a Key Distribution Center for authentication and secure key distribution. This is practically difficult to implement in an open ad hoc network wherein single administrative control is not available on all participating nodes. Cordasco et al. [26] showed that, cryptographic methods used in MANET routing protocols have more computational overhead as compared to trust based approaches. In MANET, a node may start misbehaving after passing cryptographic security checks. The security attacks pertaining to node behaviors can be successfully detected and prevented using trust [26].

The trust management mechanism is useful for identifying selfish nodes and minimizing their impact on the communication. Few examples of such trust based routing protocols are: AOTDV [10], FTDSR [11], and CTrust [12]. AOTDV [10] is a trusted extension of ad hoc on-demand multipath distance vector (AOMDV) routing protocol. The protocol discovers multiple paths from source to destination on the basis of two parameters: hop counts and trust values. In FTDSR [11], authors have extended standard reactive routing protocol viz. Dynamic Source Routing (DSR) protocol. The algorithm isolates untrustworthy nodes from the network and finds reliable route. The authors have used fuzzy logic and analytic hierarchical processing for deriving trust values of the nodes. Zhao et al. [12] have devised trust model for cyclic MANET (cMANET) to handle trust establishment and aggregation. Trust model in cMANET considers the neighbor trust along with time and location.

MCOR [14] is the first scheme which addresses the presence of selfish nodes in OR. In MCOR, each node calculates trust for neighbor nodes according to interactions with that node in the past. MCOR uses trust degree as the parameter for filtering selfish nodes from being forwarder. The MCOR algorithm uses distance from a node to the destination as metric to determine the best candidate nodes. The ORPSN algorithm differs with MCOR on following aspects: calculation of trust degree of the neighbor nodes, using trust degree in decision making, strategy for choosing best candidate node and coordination amongst the nodes for deciding who will forward the packet. TMCOR [27] is inspired from MCOR. It applies the trust based OR for Vehicular ad hoc Network (VANET).

The design of ORPSN is extended from Cooperative Opportunistic Routing in Mobile ad hoc Networks (CORMAN) [8]. CORMAN is a network layer OR protocol for mobile ad hoc networks. The design of CORMAN is based on ExOR [4], but it extensively uses network layer operations rather than MAC layer.

3 ORPSN Routing Protocol

This section gives an overview of ORPSN routing protocol at the beginning; then mathematical modeling of the algorithm and protocol design is discussed in details.

3.1 Protocol Overview

ORPSN has two primary components viz. candidate selection and coordination method. Though OR protocol delays the final route selection, it is necessary to nominate proper candidate nodes in advance. Candidate selection component decides and arranges candidate nodes using path goodness metric. If the selected best candidate node does not respond to packet forwarding then coordination method is useful.

ORPSN mimics working of CORMAN as follows. ORPSN uses strategy similar to Proactive Source Routing (PSR) [28] to build routing tables on each node. The packets are forwarded in the form of batches from upstream nodes (which are closer to the source node) to downstream nodes (which are closer to the destination). The packet contains a forwarder list, which contains ID’s of nodes along the route to the destination. The packets from a batch use the same forwarder list. The forwarder list is initially prepared by the source node. When packets move toward the destination, the forwarder list may be modified by intermediate nodes. As the packets are forwarded as per updated route, this information is propagated to upstream nodes. The detailed discussion of ORPSN candidate selection and coordination method is as below.

The ORPSN’s candidate selection component is responsible for choosing candidates and ordering them as per priority according to the path goodness metric. The path goodness value is calculated for each path from source to destination using trustworthiness of nodes lying on the path and proximity of these nodes to the destination. ORPSN candidate order is global, it means that while choosing candidate order all intermediate nodes on the path to the destination are considered.

Each node calculates trustworthiness of its neighbor nodes depending on past behavior of the node. The nodes passively monitor behavior of their neighbors. The node records positive and negative observations of its neighbors and uses Bayesian inference to calculate the trust value. The path trust of each route toward the destination is calculated using trust values of nodes lying in the path to destination. ORPSN also considers proximity of the node to the destination for choosing it as a candidate node. ORPSN measures the proximity of the node to the destination in the form of estimated transmission count (ETX). Each node calculates ETX towards the destination and shares it with its neighbors.

The path goodness value is a weighted combination of path trust and ETX values. The path goodness value is used to decide the best path toward the destination. ORPSN builds routing table using path goodness values. Each node keeps information of two best paths to the destination in its routing table using two highest priority candidate nodes. Two highest priority candidate nodes are referred as the *next expected forwarder* and the *second best forwarder* respectively.

After data packet broadcasts, candidates will respond in the order, i.e. *next expected forwarder* followed by *second best forwarder*. The coordination method takes care about this. ORPSN uses timer based coordination method similar to ExOR [4] and CORMAN [8]. After receiving a packet, the selected *next expected forwarder* forwards the packet further. *Second best forwarder* responds only when it does not observe any response from *next expected forwarder* within a threshold time. In ORPSN, if *next expected forwarder* fails, then forwarding node declares who will act as *second best forwarder*. This strategy used by ORPSN is different from one used by CORMAN. If the *next expected forwarder* do not respond in time, then the nodes decide the *second best forwarder* at real time in CORMAN protocol.

3.2 Mathematical Modelling

The mathematical model for calculating path goodness and packet forwarding process is discussed in detail in this section.

3.2.1 Path Goodness Metric

ORPSN uses trust evaluation based on Bayesian network. A Bayesian network uses Beta distribution and Bayesian inference to determine the trust relationships among the nodes. As only two parameters viz. positive observations *u* and negative observations *v* are needed, we have used Beta distributions. The values of *u* and *v* are continuously updated as observations are made by the node [20].

Let *x* and *y* be two neighboring nodes in the MANET and there are total *n* observations node *x* made about node *y*. If T_{new} is the probability of positive behavior by *y* at *n + 1* times, then posterior distribution of positive behavior of node *y* is a Beta distribution with the density function as below:

$$Beta(T_{old}|u, v) = \frac{\tau(u+v+2)}{\tau(u+1)\tau(v+1)} {T_{old}}^{u}(1 - T_{old})^{v} \tag{1}$$

$$T_{new} = E(Beta(T_{old}|u+1, v+1)) = \frac{u+1}{u+v+2}$$

$$\text{where } 0 < T_{old} < 1, \quad 0 < T_{new} < 1 \quad and \quad \text{u, v} > 0 \tag{2}$$

In ORPSN, the packet sent by a node contains a forwarder list and identity of *next expected forwarder* of the packet. Though the packet is meant for *next expected forwarder*, forwarding node broadcasts the packet. If the *next expected forwarder* do not respond for packet forwarding, the *second best forwarder* works as a forwarding node. This behavior is different from traditional routing protocols for MANET, wherein a node sends the packet targeted for a particular next hop node. Due to this, the interpretation of positive and negative observation in an OR protocol is different from traditional routing protocols. The detailed discussion of strategy used for updating *u* and *v* is given in Sect. 4.1.

The trust value of the path is computed using trust values of nodes along the path. ORPSN considers that, the path trust is not more than the trust value of most unreliable intermediate node on the path. ORPSN calculates the path trust denoted by MPT as below:

$$MPT = Min(\{T_{j,k}|n_j, n_k \; belong \; to \; path \; and \; n_j \rightarrow n_k \; and \; n_k \; \neq destination\}) \tag{3}$$

In above equation n_j and n_k are any two adjacent nodes on the path.

The ETX value is calculated for each neighbor with respect to the destination. The ETX represents node's proximity to the destination [4]. Each node shares ETX values with its neighbors. The path goodness value for node s to d using m as next hop is calculated as below:

$$\begin{aligned} &Path\ Goodness_{s,d,m} = \alpha * \left(\frac{1}{ETX_{s,d,m}}\right) + \beta * MPT_{s,d,m} \\ &here\ m\ \in Neighbor\ set\ of\ s; if\ m = d\ then\ MPT_{s,d,m} = 1 \end{aligned} \tag{4}$$

Here $MPT_{s,d,m}$ is minimum path trust of the path from node s to node d via node m and $ETX_{s,d,m}$ represent ETX value of path from s to d via m. The α and β are weighing factors such that, $\alpha + \beta = 1$. The values of α and β can be adjusted to give different weights to ETX value and trust value. Each node participating in the network uses path goodness to find the most reliable and optimal path toward the destination.

3.2.2 Markov Decision Process for Packet Forwarding

We formulate the packet forwarding process in OR as a Discrete-time Markov Chain. Markov decision process (MDP) is a discrete time stochastic control process consisting a set of states. We consider the network as a graph G(V,E), where V is the set of nodes, and E is the set of edges. We assume G is a connected graph, hence there exists at least one path between any pair of nodes in the network. Suppose a packet is transmitted; the packet is in the state *S* if it is currently carried by Node *S*. If node *S* meets node *S*' and *S* forwards the packet to Node *S*', then the state of the packet changes from *S* to *S*'. Thus, forwarding procedure on the packet can be regarded as a state transition process. The state transition function *P* determines the transition probabilities from one state to another state. If *S(t)* is denoted as the state of a packet at time t, the Markov property could be expressed as:

$$P[S(t+1)|S(t), S(t-1), \ldots, S(0)] = P[S(t+1)|S(t)] \tag{5}$$

Thus, the transition probability of packet is decided by its current state, and not by the previous states of the packet. We need to build the 1-step transition probability matrix *P*. The element $P_{S,S}{}'$ in the matrix *P* is the probability that node S chooses node S' as next expected forwarder. The components required in a MDP are defined as: $\{S, A, P\}$

Here S : state space of MDP viz. $\{S1, S2, S3, \ldots, Sn\}$

A: Action set of MDP, i.e. state transition decisions

Ps, s'*:* probability of changing state from *S* at time 't' to *S*' at time 't+1'

The state transition probability from state *S* to state *S*' to reach the destination *d* is computed as below.

$$P_{S,S'} = \frac{Path\ Goodness_{S',d}}{\Sigma_{m \in Neigbor\_Set(S)} Path\ Goodness_{m,d}} \tag{6}$$

The goal is to choose *S'* which maximizes $P_{S,S}{}'$ value.

4 ORPSN Routing Algorithm

This section discusses various algorithms used by ORPSN. Then data structures and packet format used by the ORPSN protocol is discussed in details. An illustrative example describing working of candidate selection in ORPSN is given at the end of the section.

4.1 Routing Algorithms

In Algorithm 1, node s wants to choose its candidate nodes viz. *next expected forwarder* and *second best forwarder* to reach the destination d. It starts by creating an initial candidate set. A neighbor m of s is included in the initial candidate set only if ETX(m,d) < ETX(s,d). Thus ORPSN filters out certain neighbors being potential candidates [29]. It ensures that, packet always moves in the forward direction toward the destination. After filtration newly generated candidate set is a subset of the initial candidate set. All nodes which are part of the candidate set must select their candidate sets first. This is done by recursively applying candidate selection algorithm. Finally, node s selects the best two candidate nodes from shortlisted candidate nodes using path goodness value. The candidate selection algorithm is continuously updating *next expected forwarder* and *second best forwarder* nodes with respect to each destination. The algorithm puts updated information in routing table of the node.

Algorithm 1 : Candidate node selection (s,d)

```
$CandidateSet_{s,d} \leftarrow \phi$
If s == d then
    $PathGoodness(s, d) \leftarrow 0$
    flag(s) = TRUE
    Return
End if
$InitialCandidateSet_{s,d} \leftarrow \phi$
For all $m \in NeighborSet(s)$ do
    If ETX(m,d) < ETX(s,d)
        $InitialCandidateSet_{s,d} \leftarrow InitialCandidateSet_{s,d} \cup \{m\}$
    End if
End for
For all $m \in InitialCandidateSet_{s,d}$ do
    If (flag($m$) == FALSE) then
        Call Candidate node selection($m$,d)
    End if
End for
For all $m \in InitialCandidateSet_{s,d}$ do
    Find $PathGoodness(s, d, CandidateSet_{s,d} \cup m)$
    flag(s) = TRUE
End for
Sort nodes from CandidateSet as per path goodness values.
Return first two candidates viz. next expected forwarder and
second best forwarder from Candidate Set.
```

Algorithm 2 updates ETX value of each node toward the destination. Each node periodically sends probe hello messages and determines link delivery probability (LDP) with each of its neighbor. The link ETX i.e. ETL is calculated using LDP. The nodes share ETX values for each destination after periodic intervals.

```
Algorithm 2: Update ETX (Estimated Transmission Count)
At a periodic interval update link delivery probabilities with all neighbor nodes
    using probe Hello packet
Update ETL i.e. link ETX value for each neighbor. ETL = 1/link delivery probability
Update ETX value for each destination. ETX = Sum of Link ETX values along
    lowest ETX path to the destination
```

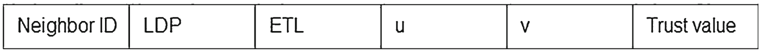
Algorithm 3 updates trust value of a node *i* on node *j*. In ORPSN, node uses ip_broadcasts to forward packet from an upstream node to a downstream node. The value *u* is incremented for a neighbor node when the node listens a packet broadcast made by the neighbor node. This essentially means that, the neighbor node is participating actively in the communication. If *next expected forwarder* node does not forward the packet within a threshold time, then value *v* corresponding to the *next expected forwarder* is incremented by 1. The possible reasons for not forwarding the packet are; the node is a selfish node or it moved away or it did not listen the packet correctly. The *u* and *v* are incremented cumulatively throughout the simulation time. Hence we expect that, if *v* is more the corresponding node is more likely to be selfish. The value of T_{new} is updated whenever there is a change in values of *u* or *v*.

```
Initialize: u = 1, v=1 for each neighbor of the node
Algorithm 3: Updating trust degree of node i on node j viz. $T_{i,j}$
For each ip_broadcast listened by the node
        Increase u for the node which made ip_broadcast
End for
If(next expected forwarder didn't make an ip_broadcast within the threshold time)
        Increase v for the next expected forwarder
End if
Update node trust value using equation 2 and store it in neighbor table
```

4.2 Data Structures and Packet Format

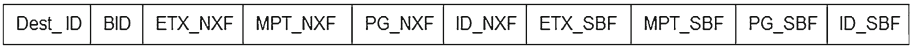
The neighbor table structure is shown in Fig. 1. The figure shows fields specific to ORPSN and does not show standard fields of a neighbor table. Here *u* and *v* represents the number of positive and negative observations recorded about the neighbor. The calculated trust value is stored in the neighbor table. Link Delivery Probability (LDP) stores the probability of successful packet delivery on the link to a particular neighbor. The neighbor table also stores link ETX i.e. ETL value.

In routing table shown in Fig. 2, ORPSN keeps information of two best candidate nodes reaching a particular destination. Routing table stores ID's *next expected forwarder* (ID_NXF) and *second best forwarder* (ID_SBF). The routing table also stores the corresponding values of ETX (ETX_NXF and ETX_SBF), path trusts (MPT_NXF and MPT_SBF) and path goodness (PG_NXF and PG_SBF) of two nodes. The 'bid' is broadcast id field, it is used to track whether listened broadcast is the latest one and to decide whether to update the routing table.



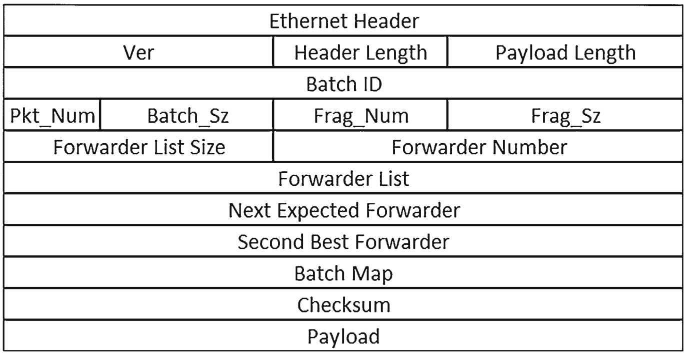
Neighbor ID	LDP	ETL	u	v	Trust value

Fig. 1 Neighbor table structure



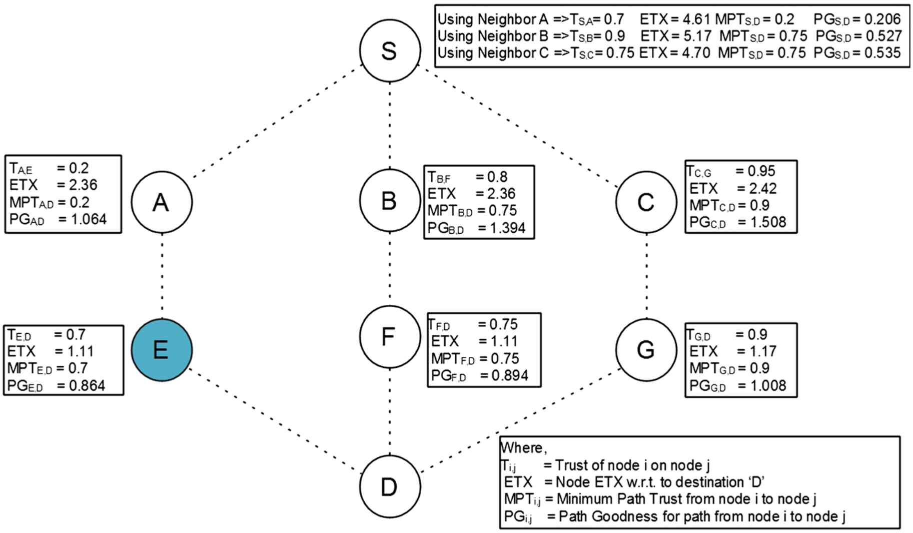
Dest_ ID	BID	ETX_NXF	MPT_NXF	PG_NXF	ID_NXF	ETX_SBF	MPT_SBF	PG_SBF	ID_SBF

Fig. 2 Routing table structure



<table>
<tr><td colspan="4">Ethernet Header</td></tr>
<tr><td colspan="2">Ver</td><td>Header Length</td><td>Payload Length</td></tr>
<tr><td colspan="4">Batch ID</td></tr>
<tr><td>Pkt_Num</td><td>Batch_Sz</td><td>Frag_Num</td><td>Frag_Sz</td></tr>
<tr><td colspan="2">Forwarder List Size</td><td colspan="2">Forwarder Number</td></tr>
<tr><td colspan="4">Forwarder List</td></tr>
<tr><td colspan="4">Next Expected Forwarder</td></tr>
<tr><td colspan="4">Second Best Forwarder</td></tr>
<tr><td colspan="4">Batch Map</td></tr>
<tr><td colspan="4">Checksum</td></tr>
<tr><td colspan="4">Payload</td></tr>
</table>

Fig. 3 ORPSN packet header format



Using Neighbor A => $T_{S,A}$ = 0.7 ETX = 4.61 $MPT_{S,D}$ = 0.2 $PG_{S,D}$ = 0.206
Using Neighbor B => $T_{S,B}$ = 0.9 ETX = 5.17 $MPT_{S,D}$ = 0.75 $PG_{S,D}$ = 0.527
Using Neighbor C => $T_{S,C}$ = 0.75 ETX = 4.70 $MPT_{S,D}$ = 0.75 $PG_{S,D}$ = 0.535

$T_{A,E}$ = 0.2
ETX = 2.36
$MPT_{A,D}$ = 0.2
$PG_{A,D}$ = 1.064

$T_{B,F}$ = 0.8
ETX = 2.36
$MPT_{B,D}$ = 0.75
$PG_{B,D}$ = 1.394

$T_{C,G}$ = 0.95
ETX = 2.42
$MPT_{C,D}$ = 0.9
$PG_{C,D}$ = 1.508

$T_{E,D}$ = 0.7
ETX = 1.11
$MPT_{E,D}$ = 0.7
$PG_{E,D}$ = 0.864

$T_{F,D}$ = 0.75
ETX = 1.11
$MPT_{F,D}$ = 0.75
$PG_{F,D}$ = 0.894

$T_{G,D}$ = 0.9
ETX = 1.17
$MPT_{G,D}$ = 0.9
$PG_{G,D}$ = 1.008

Where,
$T_{i,j}$ = Trust of node i on node j
ETX = Node ETX w.r.t. to destination 'D'
$MPT_{i,j}$ = Minimum Path Trust from node i to node j
$PG_{i,j}$ = Path Goodness for path from node i to node j

Fig. 4 Example of candidate selection in ORPSN

Figure 3 shows ORPSN's packet header format. The packet header format is similar to ExOR except the field *Second Best Forwarder*. The *Second Best Forwarder* is used to convey which node will take responsibility of packet forwarding if *next expected forwarder* fails to do so.

4.3 An Illustrative Example of Candidate Selection in ORPSN

The Fig. 4 gives a simplified scenario to understand candidate selection in ORPSN. In the figure, a node is shown by circle and dotted line between two circles indicates the wireless link.

Here a node S wants to send a packet to destination D, but it cannot deliver it directly to the destination. Node S has three neighbors viz. A, B and C which have a path to D. OR protocols like ExOR [4] or CORMAN [8], will make S to choose A as the *next expected forwarder* since the ETX value calculated using A is the lowest compared to other neighbors of S. However, on the path to D the node A meets E which is a selfish node. Thus, the packets forwarded by A is likely to be dropped by E and never reaches to the destination. Hence it is important to consider trust value of the nodes on the path to the destination before choosing *next expected forwarder*. Furthermore, it is important to know ETX value of the nodes on the path to the destination. The path goodness value for each neighbor is calculated to reach destination D. The node having maximum path goodness value becomes *next expected forwarder*; in this example it is node C. And the one which is having second best path goodness value becomes *second best forwarder* to reach destination D; in this case it is node B. These results are taken with $\alpha = 40\,\%$ and $\beta = 60\,\%$.

5 Experimental Results

We used NS2 network simulator [30] to evaluate the performance of ORPSN protocol. All experiments are carried out on a PC machine with an Intel Core i5 (3.2 GHz) processor and 4 GB main memory. Section 5.1 describes the experimental setup in details, and Sect. 5.2 gives the information about results and observations.

5.1 Experimental Setup

The NS-2.34 [30] simulator is used to evaluate the performance of ORPSN, AODV [1], DSR [2], FTDSR [11] and CORMAN [8] routing protocols. The IEEE 802.11 extension distributed coordination function is used for wireless networks. We spread 50 nodes at random positions; each node having the transmission radius of 250m within a rectangular field. The node mobility uses the random waypoint model. The nodes use 10 s pause time. We evaluated performance of the protocols under various test conditions [31]. The test conditions used are varying grid size, varying node speed and varying number of selfish nodes. The simulation parameters in NS2 [30] are listed in Table 1. We used four metrics to evaluate the performance of the routing protocols viz. packet delivery ratio, average end to end delay, control overhead and throughput. The metrics are defined as follows. Packet delivery ratio (PDR) is defined as the ratio of data packets received by the destinations to those generated by the sources. Average end to end delay is defined as, average time it takes for a data packet to reach the destination. It includes all possible delays viz. buffer delays, queuing delays, packet re-transmission delays, and propagation delay. Control overhead is the fraction of control packets required to send given data packets from source to the destination. And throughput is the amount of data transmitted per unit time from source to the destination [11].

Each experiment was repeated 20 times, and average results computed to minimize the random error. The best results were observed for parameter setting as $\alpha = 40\,\%$ and $\beta = 60\,\%$.

5.2 Observations and Analysis

In first test condition, we compare the performance of ORPSN, CORMAN, DSR, FTDSR and AODV with different grid dimensions. We use various network dimensions as $l \times l$ (m^2), where $\mathrm{l} = 500, 600, 700, \ldots, 1{,}000$. We deploy 50 nodes in each of these network dimensions

Table 1 Simulation parameters

Parameter	Value
Simulation time	500 s
Number of nodes	50
Grid size (variable in test condition 1)	500, 600, 700, 800, 900, 1,000 in m^2
Mobility model	Random way point
Traffic type	Constant bit rate (CBR) UDP
Transmission radius	250 m
Packet size	512 bytes
Connection rate	50 packets/s
Connections	5
Pause time	10 s
Speed (variable in test condition 2)	0.5, 4, 8, 12, 16, 20 in m/s
Propagation model	Two ray ground
Number of selfish nodes (variable in test condition 3)	0, 3, 6, 9, 12

to test the protocols. These nodes move using random waypoint model with maximum speed of 10 m/s. We tested the protocols using five randomly selected source-destination pairs. The 10 selfish nodes are deployed in the scenario; these nodes are dropping packets. The Figs. 5, 6, 7 and 8 show observed performance of the routing protocols over four different parameters. The packet delivery ratio of all protocols decrease gradually as grid size increases. As grid size increases, distance between the nodes increases and density decreases. It minimizes the number of nodes which can work as packet forwarder. The AODV and DSR protocols do not consider packet dropping behavior of selfish nodes while making routing decisions. Due to this, the impact of selfish nodes on AODV and DSR protocols is high. When grid size is small (i.e. $\leq 600\,\mathrm{m}^2$) performance of CORMAN and ORPSN is almost comparable. This is due to higher density of nodes and transmission radius being 250 m, it's more likely that there is a direct link between many nodes. The ORPSN protocol PDR is approximately 10 % more than CORMAN for higher dimensions. The primary reason for this is, ORPSN considers trustworthiness of nodes for deciding candidate nodes. The FTDSR performs better than ORPSN due to more sophisticated trust model and use of recommendation trust. However complex trust evaluation model causes higher end to end delay in FTDSR.

With increase in grid size, average end to end delay increases. AODV does not have coordination overhead present in an OR protocol. In AODV, many packets dropped due to the presence of selfish nodes. The packets which did not meet any selfish nodes on the path reach to the destination in least time. CORMAN and ORPSN show resilience to packet drops and finds alternative candidate node if expected forwarder fails. This causes more coordination overhead and hence CORMAN and ORPSN have more average end to end delay. ORPSN needs global network state information for calculation of path goodness value. Hence ORPSN average end to end delay is slightly higher than CORMAN. The ORPSN does not use additional control packets for evaluating and distributing trust information amongst the neighbors. Hence control overhead of ORPSN and CORMAN is almost same. However, FTDSR uses new control messages for sharing trust information amongst the nodes which causes more control overhead as compared to DSR protocol. The performance for throughput shows a similar trend like the packet delivery ratio parameter.

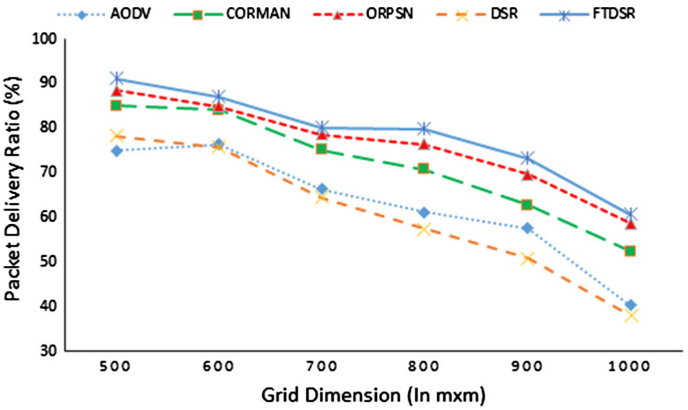


Fig. 5 Measurement of packet delivery ratio by varying grid dimensions

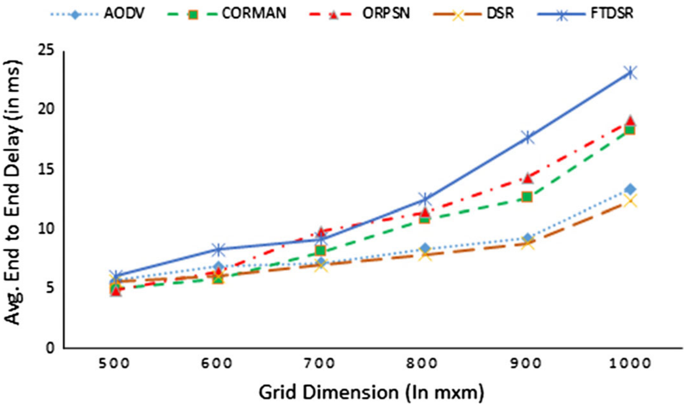


Fig. 6 Measurement of avg. end to end delay by varying grid dimensions

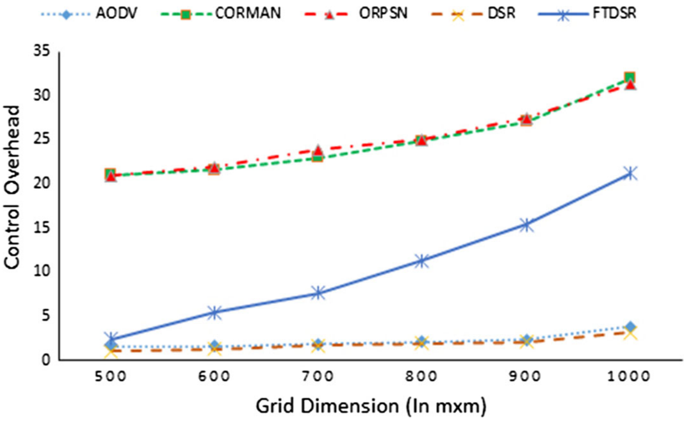


Fig. 7 Measurement of control overhead by varying grid dimensions

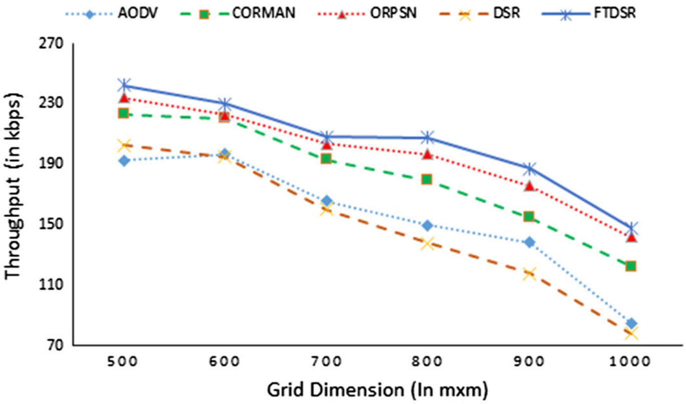


Fig. 8 Measurement of throughput by varying grid dimensions

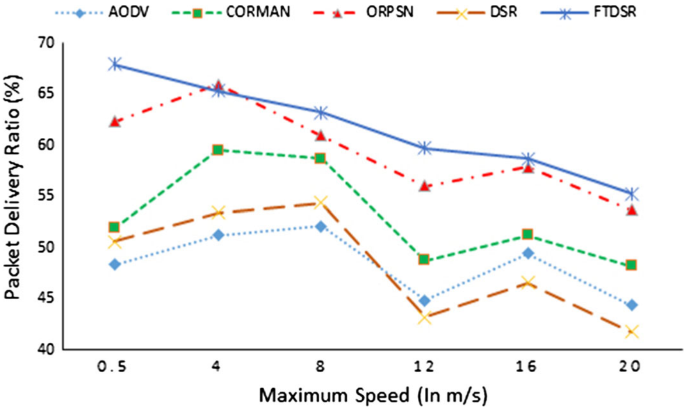


Fig. 9 Measurement of packet delivery ratio by varying node speed

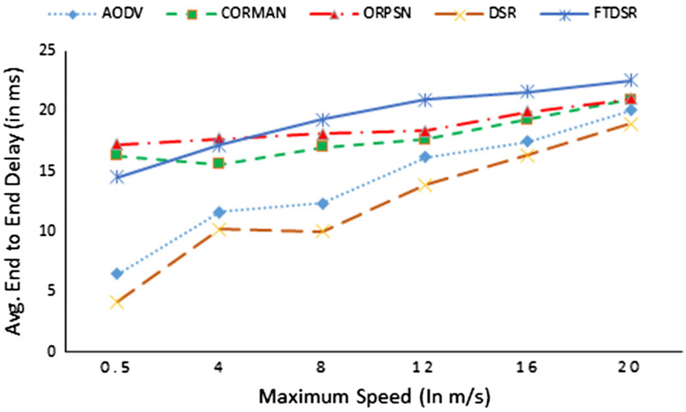


Fig. 10 Measurement of avg.end to end delay by varying node speed

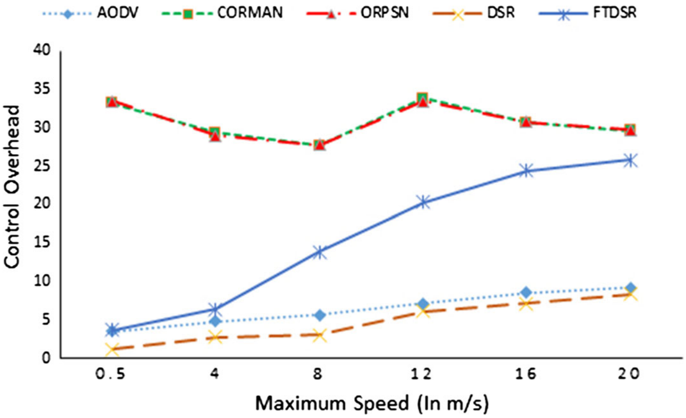


Fig. 11 Measurement of control overhead by varying node speed

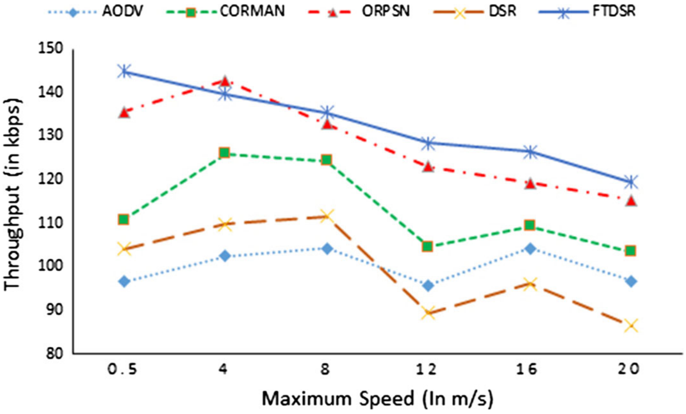


Fig. 12 Measurement of throughput by varying node speed

In the second test condition, we study the protocols performance using different speed of the nodes. Here we use 50 nodes deployed in a 1,000 × 1,000 m^2 space. The node maximum speed is varied to different values 0.5, 4, 8, 12, 16, 20 (m/s). We have deployed 10 selfish nodes in the network. We choose a fairly sparse network scenario in this test condition. The Figs. 9, 10, 11 and 12 shows observed performance of the routing protocols over four different parameters. The links between nodes become more volatile at higher speeds. Both OR protocols show better robustness against link variations as compared to AODV and DSR. ORPSN prefers more stable nodes as packet forwarders than one's which are frequently changing positions. This enables ORPSN to maintain higher packet delivery ratio as compared to other protocols. The FTDSR shows better packet delivery ratio than ORPSN due to similar reasons given in above paragraph. Due to similar reasons discussed in the first test condition, average end to end delay of AODV and DSR is less compared to CORMAN and ORPSN. The control overhead of DSR and AODV increases consistently as node's speed increases. This is due to need of more control packets for link repairs. OR protocols always consider multiple nodes as potential packet forwarders. Hence, though OR protocols control overhead is very high compared to standard routing protocols, it remains consistent with higher speed.

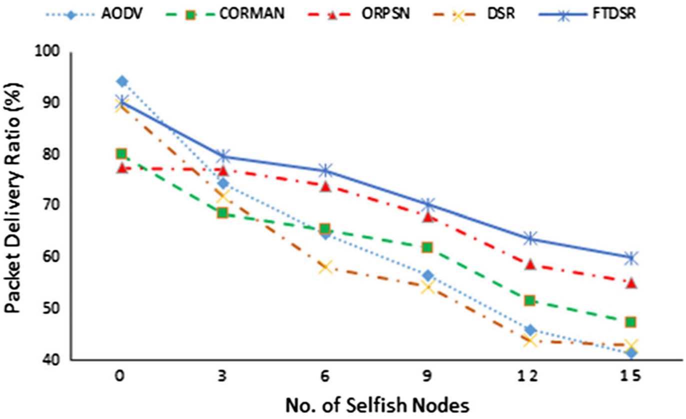


Fig. 13 Measurement of packet delivery ratio by varying no. of malicious nodes

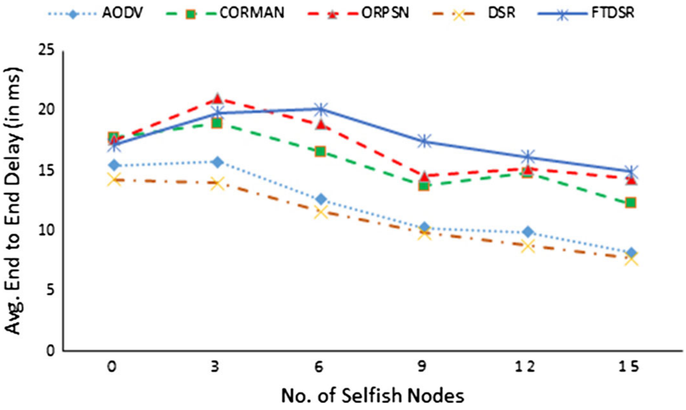


Fig. 14 Measurement of avg.end to end delay by varying no. of malicious nodes

In test condition 3, we evaluate the protocols by varying number of selfish nodes. Here we use a grid of size 1,000 × 1,000 m^2 with 50 nodes and node speed of 50 m/s. The remaining parameters are as given in Table 1. The Figs. 13, 14, 15 and 16 show observed performance of the routing protocols over four different parameters. When there are no selfish nodes, the AODV and DSR performed better than ORPSN and CORMAN in the terms of PDR and average end to end delay. The selfish nodes minimize the communication between the nodes. The delivery ratios of AODV, DSR and CORMAN degrade sharply as the number of selfish nodes increases. In ORPSN and FTDSR after initial few interactions between the nodes, each node gets a fair idea about behavior of other nodes. Then these protocols uses past behavior of the nodes to take further routing decisions. Hence FTDSR and ORPSN succeed to have approximately 10 % higher PDR compared to DSR, AODV and CORMAN. The difference between FTDSR and ORPSN performance slightly increases as the number of selfish nodes in the network increases. The average latency of all protocols declines slowly with the increase in the number of selfish nodes. However, it causes a rise in control overhead for all protocols. ORPSN gives higher packet delivery ratio at the cost of a slight increase in average end to end delay as compared to CORMAN. Also ORPSN has less control overhead compared to CORMAN when more selfish nodes are present in the network. This is due to

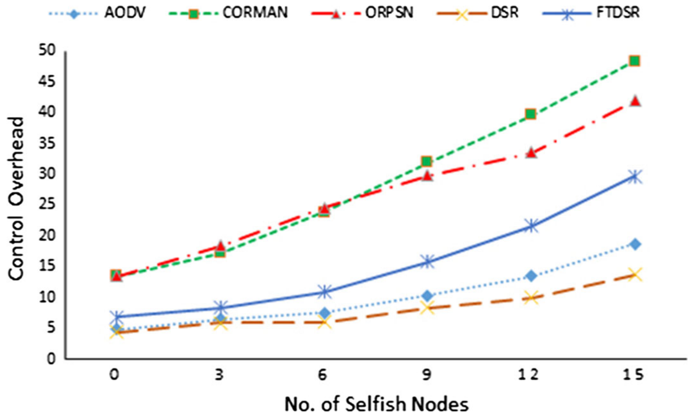


Fig. 15 Measurement of control overhead by varying no. of malicious nodes

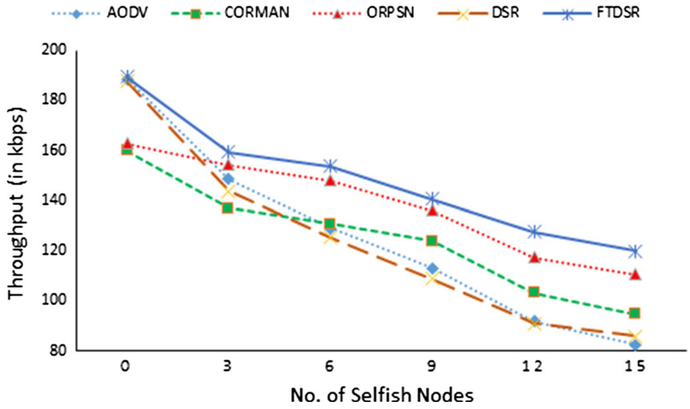


Fig. 16 Measurement of throughput by varying grid no. of malicious nodes

ORPSN's forwarder selection strategy, which takes into account packet forwarding behavior of the nodes.

6 Discussions

6.1 Immunity Against Selfish Nodes

Experimental results show that, OR protocols viz. ORPSN and CORMAN have better immunity against selfish nodes attack as compared with traditional reactive routing protocols like AODV and DSR. In OR, a set of candidate nodes can potentially act as packet forwarders. The candidate nodes respond to the packet forwarding as per priority decided by the original sender. If the high priority candidate does not forward the packet, next candidate node attempts packet forwarding. This continues until at least one candidate forward the packet. This improves reliability of packet delivery in OR. CORMAN takes care of packet drops by selfish nodes up to a certain extent and it has a better resilience to presence of selfish nodes in the network as compared to AODV and DSR. However, CORMAN chooses the candidate nodes depending on hop count and ETX, which may result into choosing a selfish node as *next expected forwarder* or *second best forwarder*. To sum up, PDR of CORMAN is more than AODV and DSR but ORPSN performs better.

6.2 Second Best Forwarder Strategy

The size of the candidate set affects the performance of an OR protocol. A larger candidate set size can increase packet delivery probability, but it causes more coordination overhead. It also causes more memory overhead [32]. Hence both CORMAN and ORPSN uses candidate set of size two. In ORPSN, forwarding node itself declares *next expected forwarder* and *second best forwarder*. ORPSN places this information in each packet being broadcast. This minimizes computational overhead at receiving nodes at great extend, which CORMAN fails to do. CORMAN chooses *second best forwarder* at real time by using heavy computations at each node. CORMAN and ORPSN uses timer-based coordination method. So in both algorithms *second best forwarder* waits until a threshold time gets expired. Hence though computational overhead in ORPSN is small, it does not reflect into reduction of average end to end delay for packet transfer.

6.3 Attack and Trust Model

ORPSN assumes simple attack model; wherein selfish nodes simply drop the packets. The trust model adopted in this paper uses packet forwarding behavior of the nodes to decide trustworthiness. However, there are several ways through which malicious nodes can disrupt the network operations. Though trust management systems are useful for improving security in the MANET, it can be attacked. Sun et al. [33] have discussed possible attacks on the trust schemes in MANETs. The [33] gives hints about the secure and robust design of a trust management system. A more complex trust model can be developed, which addresses sophisticated attacks on network operation and attack against the trust management systems.

7 Conclusions and Future Work

OR makes ingenious use of the broadcast nature of wireless communications for routing. The existing research work on OR protocols demonstrates its potential benefits, still many challenges are open. The issues like security and presence of selfish nodes are not yet addressed much in the context of OR protocols. This paper introduced a novel trusted OR protocol, which selects the candidate nodes using trust and path optimality. ORPSN extends CORMAN functionality to neutralize the presence of selfish nodes in the network. ORPSN uses a new metric viz. path goodness to decide optimal forwarder for delivering the packet to the destination.

In ORPSN, source node establishes a trustworthy and optimal path to a destination by selecting the best candidate nodes at each stage in route discovery. ORPSN also replaces the strategy used by CORMAN for selecting second best forwarder. Experimental results show that, ORPSN achieves a approximately 10 % improvement in packet delivery ratio in the presence of selfish nodes.

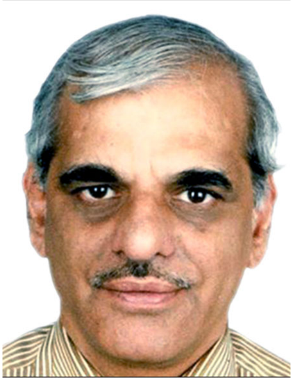
The future extensions to this work may address malicious nodes, which are capable of making complex attacks in the network operation. Our forthcoming plan includes refining the trust model to make it robust against attacks on itself. We believe that, trust may be useful for decision making at various stages in OR. In the future, we plan to design an OR protocol which makes better use of node trustworthiness to improve efficiency and reliability of the protocol.

References

1. Perkins, C. E., & Royer, E. M. (1999). Ad hoc on-demand distance vector routing. In *Proceedings of workshop mobile computing systems and applications*, pp. 90–100.
2. Johnson, D. B., Maltz, D. A., & Broch, J. (2001). DSR: The dynamic source routing protocol for multihop wireless ad hoc networks. In *Ad hoc networking*, (pp. 139–172). Reading: Addison-Wesley.
3. Marina, M. K., & Das, S. R. (2001). On-demand multipath distance vector routing in ad hoc networks. In *Ninth international conference on network protocols*, pp. 14–23.
4. Biswas, S., & Morris, R. (2005). ExOR: Opportunistic multi-hop routing for wireless networks. In *Proceedings of conference on applications, technologies, architectures, and protocols for computer communications (SIGCOMM '05), ACM, New York, NY, USA*, pp. 133–144.
5. Liu, H., Zhang, B., Mouftah, H. T., Shen, X., & Ma, J. (2009). Opportunistic routing for wireless ad hoc and sensor networks: Present and future directions. *IEEE Communications Magazine*, *47*(12), 103–109.
6. Du, X., Liu, Y.-A., Liu, K.-M., Tang, B.-H., & Chen, X. (2013). Throughput oriented forwarders selection analysis for opportunistic routing in wireless mesh network. *The Journal of China Universities of Posts and Telecommunications*, ISSN 1005–8885. *20*(2), 73–78.
7. Hsu, C.-J., Liu, H.-I., & Seah, W. K. G. (2011). Opportunistic routing: A review and the challenges ahead. *Computer Networks*, ISSN 1389–1286, *55*(15), 3592–3603.
8. Wang, Z., Chen, Y., & Li, C. (2012). CORMAN: A novel cooperative opportunistic routing scheme in mobile ad hoc networks. *IEEE Journal on Selected Areas in Communications*, *30*(2), 289–296.
9. Djahel, S., Nait-abdesselam, F., & Zhang, Z. (2011). Mitigating packet dropping problem in mobile ad hoc networks: Proposals and challenges. *IEEE Communications Surveys & Tutorials*, *13*(4), 658–672.
10. Li, X., Jia, Z., Zhang, P., Zhang, R., & Wang, H. (2010). Trust-based on-demand multipath routing in mobile ad hoc networks. *IET Information Security*, *4*(4), 212–232.
11. Xia, H., Jia, Z., Ju, L., & Zhu, Y. (2011). Trust management model for mobile ad hoc network based on analytic hierarchy process and fuzzy theory. *IET Wireless Sensor Systems*, *1*(4), 248–266.
12. Zhao, H., Yang, X., & Li, X. (2013). Trust management in cyclic mobile ad hoc networks. *IEEE Transactions on Vehicular Technology*, *62*(6), 2792–2806.
13. Rozner, E., Seshadri, J., Mehta, Y., & Qiu, L. (2009). SOAR: Simple opportunistic adaptive routing protocol for wireless mesh networks. *IEEE Transactions on Mobile Computing*, *8*(12), 1622–1635.
14. Bo, W., Chuanhe, H., Layuan, L., & Wenzhong, Y. (2011). Trust-based minimum cost opportunistic routing for ad hoc networks. *Journal of Systems and Software, 84*(12), 2107–2122, ISSN 0164-1212.
15. Zhong, Z., & Nelakuditi, S. (2007, June). On the efficacy of opportunistic routing. In *4th annual IEEE communications society conference on sensor, mesh and ad hoc communications and networks, SECON '07*, pp. 441–450, 18–21.
16. Zeng, K., Yang, Z., & Lou, W. (2009). Location-aided opportunistic forwarding in multirate and multihop wireless networks. *IEEE Transactions on Vehicular Technology*, *58*(6), 3032–3040.
17. Zuo, J., Dong, C., Nguyen, H. V., Ng, S. X., Yang, L.-L., & Hanzo, L. (2014). Cross-layer aided energy-efficient opportunistic routing in ad hoc networks. *IEEE Transactions on Communications*, *62*(2), 522–535.
18. Mao, X., Tang, S., Li, X., & Ma, H. (2011). Energy-efficient opportunistic routing in wireless sensor networks. *IEEE Transactions on Parallel and Distributed Systems*, *22*(11), 1934–1942.
19. Shen, H., Bai, G., Tang, Z., & Zhao, L. (2014). QMOR: QoS-aware multi-sink opportunistic routing for wireless multimedia sensor networks. *Wireless Personal Communications Journal*, *75*, 1307–1330.
20. Li, F., & Jie, W. (2010). Uncertainty modeling and reduction in MANETs. *IEEE Transactions on Mobile Computing*, *9*(7), 1035–1048.
21. Myung, J., & Lee, W. (2012). Eliminating duplicate forwarding in wireless opportunistic routing. *IEEE Communications Letters*, *16*(4), 510–513.
22. Ajmal, M. M., Madani, S. A., Maqsood, T., Bilal, K., Nazir, B., & Hayat, K. (2013). Coordinated opportunistic routing protocol for wireless mesh networks. *Computers & Electrical Engineering*, ISSN 0045–7906, *39*(8), 2442–2453.
23. Othmana, J. B., & Mokdadb, L. (2010). Enhancing data security in ad hoc networks based on multipath routing. *Elsevier Journal of Parallel Distributed Computing*, *70*, 309–316.
24. El Defrawy, K., & Tsudik, G. (2011). ALARM: Anonymous location-aided routing in suspicious MANETs. *IEEE Transactions on Mobile Computing*, *10*(9), 1345–1358.

25. Gunasekaran, M., & Premalatha, K. (2013). TEAP: Trust-enhanced anonymous on-demand routing protocol for mobile ad hoc networks. *IET Information Security*, *7*(3), 203–211.
26. Cordasco, J., & Wetzel, S. (2008). Cryptographic versus trust-based methods for manet routing security. *Electronic Notes in Theoretical Computer Science*, *197*(2), 131–140.
27. Zhizhong, J., Chuanhe, H., Liya, X., Bo, W., Xi, C., & Xiying, F. (2012, October). A trusted opportunistic routing algorithm for VANET. In *Third International Conference on Networking and Distributed Computing (ICNDC)*, pp. 86–90.
28. Wang, Z., Chen, Y., & Li, C. (2014). PSR: A lightweight proactive source routing protocol for mobile ad hoc networks. *IEEE Transactions on Vehicular Technology*, *63*(2), 859–868.
29. Li, Y., Chen, W., & Zhang, Z. L. (2009). Optimal forwarder list selection in opportunistic routing. In *IEEE 6th international conference on mobile adhoc and sensor systems, MASS'09*, pp. 670–675.
30. http://www.isi.edu/nsnam/ns/.
31. Fouchal, H., Hunel, P., & Ramassamy, C. (2014). *Towards efficient deployment of wireless sensor networks*. Special issue paper in Security and Communication Networks. New York: Wiley.
32. Darehshoorzadeh, A., Cerd-Alabern, L., & Pla, V. (2011, September). Modeling and comparison of candidate selection algorithms in opportunistic routing. *Computer Networks*, ISSN 1389–1286, *55*(13), 2886–2898.
33. Sun, Y., Han, Z., & Liu, K. J. R. (2008). Defense of trust management vulnerabilities in distributed networks. *IEEE Communications Magazine*, *46*(2), 112–119.

Sandeep A. Thorat completed B.E. from Shivaji University Kolhapur. He completed M.Tech. from IIIT Hyderabad with specialization in Information Security. His area of interest is Wireless Networks and Security. He has authored a book on C programming. Presently he is Ph.D. scholor in Walchand College of Engineering, Sangli India.



P. J. Kulkarni is working as Professor in Computer Science Department in Walchand College of Engineering, Sangli India. He has more than 25 years of experience in teaching. His area of interest is Machine Learning, Pattern Recognition and Wireless Networks. He has guided number of post graduate and Ph.D. students. He has received a distinguished position in Asia-Pacific's Who's Who 2004, for his contribution to excel academics. P. J. Kulkarni has been instrumental on various bodies of All india council for Technical Education, (New Delhi), Directorate of Technical Educational (Maharashtra State, Mumbai) and Shivaji University, Kolhapur.