Bulletin of Electrical Engineering and Informatics
Vol. 12, No. 6, December 2023, pp. 3628~3637
ISSN: 2302-9285, DOI: 10.11591/eei.v12i6.4895

Develop a new handling method for selfish nodes in mobile ad-hoc networks

Sanaa Jafaar Hassan Al-Shakarchi[1], Raaid Alubady[1,2]
[1]Department of Information Networks, Collage of Information Technology, University of Babylon, Babylon, Iraq
[2]Engineering Technical College, Al-Ayen University, Thi-Qar, Iraq

Article Info

Article history:

Received Sep 27, 2022
Revised Nov 12, 2022
Accepted Dec 27, 2022

Keywords:

Ad-hoc on-demand distance vector
Communication ratio
Energy
Mobile ad-hoc networks
Selfish nodes

ABSTRACT

Mobile ad-hoc networks (MANETs) have been a crucial element of next-generation wireless networking technologies during the last decade. Because they allow users to access information and communicate with each other without infrastructure. Selfishness is one of the numerous undesirable behaviors that MANET network nodes may exhibit since this selfish node attempts to safeguard its own resources while accessing the services of other nodes and consuming their resources. Hence, a potential that the network's overall performance may degrade. This study developed a new method named detection, reintroduced, and collaborative of selfish node (DRCSN) that proposed detecting selfish nodes based on two factors: energy and the communication ratio (CR) and handling the rate of selfish nodes. Thus, selfish nodes were exploited to the maximum degree and significantly improve network performance. DRCSN was implemented inside ad-hoc on-demand distance vector (AODV) protocol. The test scenarios were implemented using the network simulator-2 (NS-2); many scenarios were created according to two important network parameters: the number of nodes and movement nodes. The proposed method improved the MANET's performance by increasing both the throughput and packet delivery ratio in the network in addition to that it reduced retransmission rate, delay, and power consumption compared to the related methods.

Corresponding Author:

Raaid Alubady
Department of Information Networks, Collage of Information Technology, University of Babylon
Babylon, Iraq
Email: alubadyraaid@itnet.uobabylon.edu.iq

1. INTRODUCTION

Over the past few decades, scientists and researchers have become very interested in wireless communication. With the fast improvements in wireless technology, ubiquitous computing, which maintains connectivity between mobile nodes regardless of their location, is becoming a reality more and more [1]. Mobile ad-hoc networks (MANETs) are wireless networks that are self-created, self-managed, and self-organized. They only work for a short time. These wireless nodes can move wherever they want and can act as a source, a destination, or an intermediate router in a network. This means that they can send and receive information. The network communication working is affected by the movement of nodes [2]. This is explained by the fact that short-range communication may not need the utilization of infrastructure. In contrast to cellular networks, there is no central controlling unit in a MANET [3], which distinguishes it from them. This unique characteristic has attracted its use in the fields of defense (military MANET) [4], emergency response [5], healthcare [6], and combined or collaborative networks [7].

Packet routing is an important part of MANETs. For each pair of nodes that are not next to each other, the intermediate nodes must send data packets to the destination nodes. Due to how dynamic and spread out the nodes are, energy use is one of the biggest problems in MANET. Energy use is especially important since all node area units run on batteries [8]. As a result, opportunistic routing algorithms make the assumption that each node will forward each packet it receives. This has not always been the case, though, because some nodes use the resources of other nodes to communicate and will not forward packets from other nodes within their radio spectrum. These nodes are said to be selfish or act badly in some way [9]. Even if only one node fails, the whole network can be affected [10].

Traditional MANET protocols presume that all mobile nodes must cooperate together in order for the network to work. Selfish node behavior might emerge if nodes refuse to cooperate since it is a costly activity. The selfish nodes do not need to use their energy, central processing unit (CPU), or bandwidth to send the data. In point of fact, each and every node that makes up a MANET has the potential to exhibit a selfish personality. In order to maximize revenues from network resources, but hesitant to share its resources with other nodes. In a situation where every node is required to send packets to its neighbors, a few selfish nodes deny doing so. Except for packets intended for them, these nodes block all traffic. For their own purposes, these nodes consume the network and its resources without providing service back [11]. Selfish nodes may negatively affect on the performance of the network in ways like network partitioning, less data availability, shorter network life, reduced throughput, and more packets being dropped [12], [13]. Selfish node identification is not a simple task, although various approaches such as [13]–[16] have been successful in preventing them from accessing any network resources. A MANET's performance may not be improved merely by identifying and isolating selfish nodes. There is currently no way to convince them to cooperate until they have expended their energy. Therefore, the proposed system is to resolve a solution to the problem of selfish nodes based on making them collaborative in the network. Hence, greatly improving the network performance.

The aim of this research is to develop a selfishness node handling method. The proposed method has the ability to detect selfish nodes and motivate them to cooperate in data delivery in order to increase the performance of the network. This aim can be reached even more with the following research objectives:

a. To investigate how selfish nodes can affect performance in a MANET scenario with different parameters.
b. To design a method for identifying selfish nodes early based on their energy and communicate rate to reduce the detection time of selfish nodes.
c. To build a model that can deal with the behaviour of selfish nodes by adjusting the threshold according to network status and motivating them to cooperate as much as possible.
d. To develop a selfishness node handling method for MANET based on standard ad-hoc on-demand distance vector (AODV) protocol and modify the required parameters.
e. To evaluate the performance of the proposed model in comparison with available solutions in a simulated network environment.

Research by Al-Shakarchi and Alubady [17], made a survey that aims to overview and analyze many innovative methods for detecting selfish nodes in MANETs. In this section, the modern related studies are reviewed and criticized: Mubeen and Johar [14] identified a selfish node utilizing an energy credit based system (EBCS). The selfish node will be deleted from packet transmission in this scheme. The energy-based credit system recognizes cooperating nodes and awards them with energy. Checking the threshold energy level of selfish nodes is achieved by the energy-based credit system NS-2, which uses 10 nodes. Ad-hoc networks perform better with a payment system based on energy.

Ponnusamya *et al.* [15] proposed to selfish node removal using reputation model (SNRRM). Reputation is needed in order to exclude selfish nodes from routing. A node's reputation is based on its current energy level and communication ratio (CR). The sender node initiates communication when both the source and destination nodes are identified. If both 'S' and 'D' are within the communication range, the node will check 'S's reputation value; if it matches, the transmission operation will be finished, and the system will be updated. If S and D are out of range, S will send control packets to its neighbors and wait for reply messages. In this situation, reputation checks are problematic since selfish nodes don't respond to messages. The CR between nodes is computed using request and reply messages. This simulation uses NS-2.35 and 100 nodes. The simulation result shows that the reputation ratio and delivery rates are greater. This method detected and avoided selfish nodes it by choosing a reliable route.

Musthafa *et al.* [16] proposed a selfish node detection algorithm (SNDA). Where a node's desire to participate in routing operations is tested using the SNDA, based on how many routing packets the node in the network has lost over time, a threshold value ranging from 0 to *t* will be determined. A node is deemed gentle or somewhat selfish if the selfish threshold (*ST*) is lower than the threshold value (*t*). If the *ST* is greater than or equal to *t*, the node is deemed selfish. Nodes' reluctance to forward packets in the network is represented by the *t* value. Network simulator (NS-2) version 2.33 is used for simulation in the practical section. A flat area of 1,500×300 meters is used to simulate 50 to 100 nodes. With a 512-byte packet size and

a data rate of four packets per second, user datagram protocol (UDP) with a constant bit rate (CBR) traffic is employed. This approach is able to quickly and effectively identify and isolate the selfish node from the network.

Mangayarkarasi and Manikandan [13] developed a cost-effective collaborative anomaly detection system (CECAD) for MANET selfish node assaults. An anomaly detection module is first executed on a set of monitoring nodes that the system has selected. When a source node has to share information with a destination node, it depends on the monitoring nodes to do so cooperatively. The data collection component is in charge of gathering information about each node from packets of control and data. The fuzzy logic decision is used to identify the nodes that are either weakly or significantly suspicious. False positives and missed detections are less likely since the attacks are validated by a cooperative exchange of monitoring nodes. NS-2 and 60 to 140 nodes are used to replicate this system, with 10% of the nodes being attackers. According to the findings of the simulations, the CECAD system has reduced detection latency while also increasing detection precision.

On this basis, the present study is organized in the following way: started with illustrates an overview of the background and related works in section 1. Section 2 explains the proposed algorithm. While section 3 clarifies the research method including simulation setup and performance metrics. Section 4 discusses the findings. Finally, point out the conclusions of the paper in section 5.

2. PROPOSED ALGORITHM

All existing methods of manipulating selfish nodes merely detect and isolate them from network activities. In this study, we have developed a detection, reintroduced, and collaborative of selfish node (DRCSN) algorithm to deal with selfish nodes and exploit them to their fullest instead of early isolating them. Depending on the CR and energy, we managed to detect selfish nodes in the MANET because they have a significant and subtle impact on selfish node detection. The following equation are calculating energy and CR in detecting selfish nodes. The node's activity is monitored based on reply messages received from other nodes in the network. The node CR is established based on the nodes' behavior. If the available energy is lower than the energy threshold and the CR is lower than 30% (where a value of 30% has been imposed depending on [18], the node is normal and capable of sending and receiving data. Otherwise the node is considered selfish. The formulation of CR for each node according to [14], [18] was used as a base for our developed method, it presented based on (1) and (2):

$$unsentMassegenode = GRRnode - SRRnode \quad (1)$$

$$CR = ((GRR - unsentMassegenode)/GRR) * 100 \quad (2)$$

where GRR is get route request and SRR is send route reply. In order to detect the selfish nodes in the network, we relied on the residual energy as (3):

$$residualEnergynode = Initial\ Energy - Consumed\ Energy \quad (3)$$

where *residualEnergynode* represents the the current energy of the nodes.

After calculating the energy for each node included in the network, the following is the threshold (in (4)) that is used to detect the selfish nodes based on residual energy during the simulation time:

$$Threshold = ((IEnode - residualEnergynode)/residualEnergynode) * currentTime \quad (4)$$

where IE is initial energy and threshold is threshold of energy

If the remaining energy in the node becomes less than the threshold, this node will be identified as a selfish node. The reputation of each node is obtained by the current energy level of the node and its CR. Before the source node starts to send packets to a destination node, it lookups at its routing table for a routing route that leads to the destination node, the source node will send an overwhelming number of route request route request (RREQ) packets to its neighbors in the event that the route to the destination node cannot be located. The available threshold, residual energy, and CR are the three fields that are included in the RREQ packet that is part of the AODV protocol. According to the RREQ packet that AODV uses. If the available energy is less than the energy threshold and the CR is lower than 30%, then the mobile nodes in question are considered (detected) to be selfish.

On the other side, to make selfish nodes have to cooperate by controlling the packet rate, as long as nodes have enough power to send and receive packets. It will give them packets they can handle. Thus, reducing

the burden on selfish nodes to remain competent in the network and not cause damage to the network. Thus, the proposed method achieved its purpose and it will lead to greatly improved network performance. After the detection of selfish nodes depending on the remaining energy and CR, when the residual energy in the node is between threshold and 10% (where we assumed that the energy needed by the nodes to be effective in the network is 10% of its initial energy) and CR is less than 30%, it will classify the nodes as selfish but exploitable. In this case, it will be reduced the number of requests sent to it in order to be employed as much as possible. But, when the energy of the nodes is less than 10%, the isolation method will be called to isolate it from network activities and replacing with another node. Algorithm 1 is presented a DRCSN.

After defining the nodes to be isolated, these nodes are isolated through the selfish isolate scheme, which isolates the selfish nodes from the routing table. When the selfish node is detected, it will be deleted from the AODV routing table, thus reducing the number of contents of the routing table to minus one. When resending the hello messages, if the routing table is not full, a new ID will be entered for it, provided that the ID is not present in the routing table or is back to a selfish node that was previously deleted. Algorithm 2 is presented the steps involved in isolating selfish nodes.

Algorithm 1: detection, reintroduced and collaborative of selfish node (DRCSN)

```
Definitions:
      nNodes: Number of nodes
      GRR: Get Route Request
      SRR: Send Route Reply
      IE: Initial Energy
      CE: Consumed Energy
      CR: Communication Ratio
      residualEnergy: Current Energy
      Threshold: Threshold of Energy
Begin:
 1.   For each node_i ∈ nNodes do
 2.        Threshold ← 0
 3.        Calculate the residualEnergy ← IE_node_i - CE_node_i
 4.        Calculate the unsentMassegenode_i ← GRR_node_i – SRR_node_i
 5.        Calculate the CR_node_i ← ((GRR_node_i – unsentMassege_node_i) / GRR_node_i) * 100
 6.        Threshold ← ((IE_node_i – residualEnergy_node_i) / residualEnergy_node_i) *
           currentTime
 7.        if CR_node_i < 30% and residualEnergy_node_i < Threshold then
 8.             if residualEnergy_node_i > 10 then
 9.                  Update Rate ← Rate / 2
10.             Otherwise, Call isolateSelfishNodes (node_i)
11.             End_if
12.        Otherwise, node_i is cooperative
13.        End_if
14.   End_for
End_Algorithm
```

Algorithm 2: Isolate Selfish Node

```
Input:
    node_i: Selfish node
Output:
    Delete the route of the selfish node from node_i routing table
Begin
    1.  recordRouting Table Lookup (IDnode_i)
    2.  if recordRouting Table != 0 then
    3.      AODV_Delete (IDnode_i)
    4.      AODV_routing Table - 1
    5.  End_if
    6.  AODV broadcast Hello message
    7.  if AODV_routing Table is not full then
    8.      if new IDnode_i is not IDnodei and new IDnode_i is not included in AODV_routing
    Table then
    9.              AODV_Add (new IDnode)
    10.             AODV_routing Table + 1
    11.      End_if
    12.  End_if
End Algorithm
```

3. RESEARCH METHOD

The AODV protocol [19] is modified in order to implement the suggested approach. We will alter two files in the ns-2.35/aodv/ subdirectory, aodv.h and aodv.cc. To get the energy of nodes, include the #include <mobilenode.h> line at the beginning of the aodv.h file. In addition to declaring the variable

MobileNode*iNode; in the aodv class (in protected scope). We will make the following changes via adding and calling the required parameters of the proposed approach inside aodv.cc. To get the number of GRR and SRR for each node in the network, by AODV::recvRequest() function and AODV::sendReply() function. In AODV protocol, the run time information is required during a forward of a packet. So the function from the mobilenode.h will be called and the equations for detecting selfish nodes are written in Forward(aodv_rt_entry * rt, Packet * p, double delay). After the selfish nodes were discovered in the previous step, the rate of the selfish node will be reduced in AODV::sendRequest() function. In this part, the selfish nodes that were deleted from the routing tables will be replaced with normal nodes (where the entries of the routing tables will be checked are not selfish nodes). In AODV::recvReply(Packet *p) function.

NS-2.35 [20] is used extensively for simulation. The simulation settings that were configured for our research are outlined in Table 1. In this study also, the following performance measures have been utilized: packet delivery ratio (PDR), throughput, average end-to-end delay, packet retransmission, and power consumption.

- PDR: the ratio of the number of packets received by the destination to the number of packets created by the source node is referred to as the PDR [21]. In (5) is used to calculate PDR:

$$PDR = (\frac{No.of\ the\ packet\ received}{No.of\ the\ packet\ \text{sent}}) * 100\% \quad (5)$$

- Throughput: it is one of the dimensional metrics of the network that determines the amount of the channel capacity that is really being utilized for productive transmission. Chooses a destination at the start of the simulation; this provides information on whether or not data packets were successfully delivered to their respective destinations [22]. In (6) is used to calculate throughput:

$$Throughput(kbps) = (\frac{\text{output data (byte)}}{Times}) * (\frac{8}{1024}) \quad (6)$$

- Average end-to-end delay: end-to-end packet delay may be described as the difference in time between the time moment at which the packet reaches the receiver and the time instant at which the packet is formed at the sender [23]. In (7) is used to calculate average end to end delay (AE2E):

$$AE2E = \frac{\text{sum of the time spent to deliver packets for each destination}}{\text{number of packets received by the all destination nodes}} \quad (7)$$

- Packets retransmission rate (PRR): resending data packets that could not be sent the first time successfully around due to corruption or loss is what is meant by the term "packet retransmission" [24]. In (8) is used to calculate PRR:

$$PRR = \frac{\text{No.of packets sent} - \text{No.of the packet received}}{\text{No.of packet sent}} * 100\% \quad (8)$$

- Power consumption: the term refers to the amount of energy that is consumed by each node [25]. In (9) is used to calculate power consumption:

$$Power\ Consumption = Initialenergy - Finalenergy \quad (9)$$

Table 1. Simulation parameters

Parameters	Values
Simulator	NS-2.35
Mac layer	Mac/802.11
Interface queue type	Queue/DropTail/PriQueue/MUP
Traffic source	CBR (4.0 packets/sec)
Terrain area	1,000×1,000 m
Packet size	512 bytes
Protocol	AODV
Propagation type	Two ray ground
Every mobile node contains energy	100 joules of energy
Required for each time slot of communication	10 joules of energy
Max packet in IFQ	50
Movement nodes	5, 10, 15, 20, and 25 m/s
No. of mobile nodes	20, 40, 60, 80, and 100
Simulation time	100 s

4. RESULTS AND DISCUSSION

In this section, we will discuss the results of the study, analyze quanitative data. The results are also addressed in the context of findings via implementing previous research and the accessible literature in order to highlight similarities and contrasts between the findings of this study and those of previous studies. DRCSN method was implemented in the NS-2 simulator and compared to prior work using standared AODV, SNRRM [11], and EBCS [10]. Two situations were studied in order to compare the performance of DRCSN, standard AODV, SNRRM, and EBCS under two conditions (impact of a number of nodes and impact of a variety of movement nodes).

4.1. Impact of the number of nodes

In this section, we will compare the results of the proposed method DRCSN with the standard AODV, SNRRM, and EBCS according to the impact of the parameter of a number of nodes. The maximum number of nodes was varied as 20, 40, 60, 80, and 100 nodes and the movement node are fixed as 10 m/s. Figure 1 illustrate the number of nodes impact for DRCSN, standard AODV, SNRRM, and EBCS methods regarding the various performance metrics.

As illustrated in Figure 1(a), when the number of nodes is increased, the throughput of the DRCSN method will be superior to that of SNRRM, EBCS, and standard AODV respectively. Except in some cases, the throughput of SNRRM is better. Since the natural variations in the intermediate network and devices has an effect on the packets. For the next simulation result (see Figure 1(b)), came to the realization that the proportion of DRCSN retransmissions steadily decreased as the number of nodes increased. When compared to previous related works, will find that DRCSN has the lowest PRR since where the lower the retransmission of the packet rate, the better performance can note that standard AODV has the highest retransmission rate, therefore the worst performance. As show in Figure 1(c), the performance comparison of DRCSN, standard AODV, SNRRM, and EBCS methods in terms of PDR. When the number of nodes increases, the recently introduced DRCSN method achieves a higher PDR than the models that are used in the older standard AODV, SNRRM, and EBCS. In the given figure (see Figure 1(d)), when the number of nodes rises, the value of the power consumption is almost never not more than 7% of its original value for DRCSN. When compared to other methods, because the standard AODV, SNRRM, and EBCS the selfish nodes cause a decrease in the reception of packets to the destination, and therefore the source will resend the packets. Therefore, the number of next hops in the transmission will increase, which will lead to large consumption of energy. So can clearly notice that the proposed DRCSN method is the best, as it consumes the least amount of energy, while both the standard AODV and SNRRM are the worst. As noted in Figure 1(e), the DECSN method has the lowest average E2E delay, which changes a little from 20 to 100 nodes. While in standard AODV, the average E2E delay is higher. This large difference in the average delays, due to the previous methods makes the source will keep sending packets to the destination, which will keep waiting for packets to arrive correctly. As shown in the previous results, the proposed DRCSN method significantly outperformed the current models when increasing the number of nodes from 20 to 100 nodes.

4.2. Impact of a variety of movement nodes

In this section, we will compare the results of the proposed method DRCSN with the standard AODV, SNRRM, and EBCS as well as analyze the impact of a variety of speed nodes on them. Where the maximum speed of nodes is varied as 5, 10, 15, 20, and 25 m/s, and the number of nodes was fixed at 100 nodes. Figure 2 show the impact of a variety of speeds of nodes for DRCSN, standard AODV, SNRRM, and EBCS regarding the various performance metric.

Figure 2(a) depicts the performance comparison of DRCSN, standard AODV, SNRRM, and EBCS methods in terms of throughput when the speed increases from 5 to 25 m/s. In the given figure, when the speed is increased, the throughput of the DRCSN method will be better than SNRRM, EBCS, and standard AODV, respectively. However, in some cases, the throughput of SNRRM will be better. As show in Figure 2(b), the range of the PDRs varies based on the speed of the nodes that are being used. This is an issue which may be seen; in the case of its speed value being 10 m/s, the range is the greatest. This is in contrast to speed of 20 and 25 m/s, which have a range that gradually diminishes as they increase in speed. When comparing the DRCSN method of standard AODV, SNRRM, and EBCS, it found that the DRCSN method is the best in terms of delivery ratios, followed by EBCS, SNRRM, and then standard AODV. In Figure 2(c), the PRR value for DRCSN ranged from around 30 to 44% when the speed values were modified from 5 to 25 m/s. In contrast to the results obtained by previous methods, the DRCSN method produced much better results (30 to 44%) compared with the previous methods SNRRM, EBCS, and standard AODV which ranged between 51 to 77%, 49 to 70%, and 63 to 81% respectively. In terms of the power consumption in the network when the speed of nodes is increased from 5 to 25 m/s, as show in Figure 2(d). The performance of the DRCSN method is better compared to the EBCS, SNRRM, and the standard AODV, respectively. Due to the fact that the intermediate nodes on the way to the destination of the DRCSN will finish the requests

without necessitating a journey back to the source to select an alternative path when selfish nodes are present. Because of this, there will be fewer next hops, which means less energy will be required. The result is shown in Figure 2(e), the performance comparison of DRCSN, SNRRM, EBCS, and standard AODV methods in terms of average E2E delay over five different speeds from 5 to 25 m/s. According to the obtained results, it is noted that the SNRRM and EBCS methods are similar amounts when it comes to average E2E delay for all settings of the speeds of nodes, while the standard AODV has the worst results. On the other hand, the value of the average E2E delay for DRCSN is significantly different from theirs. That is because the source of standard AODV, SNRRM, and EBCS will continue sending packets to the destination, which will continue waiting for packets to arrive correctly. Therefore, it is indicated that the performance of DRCSN is much better than standard AODV, SNRRM, and EBCS.

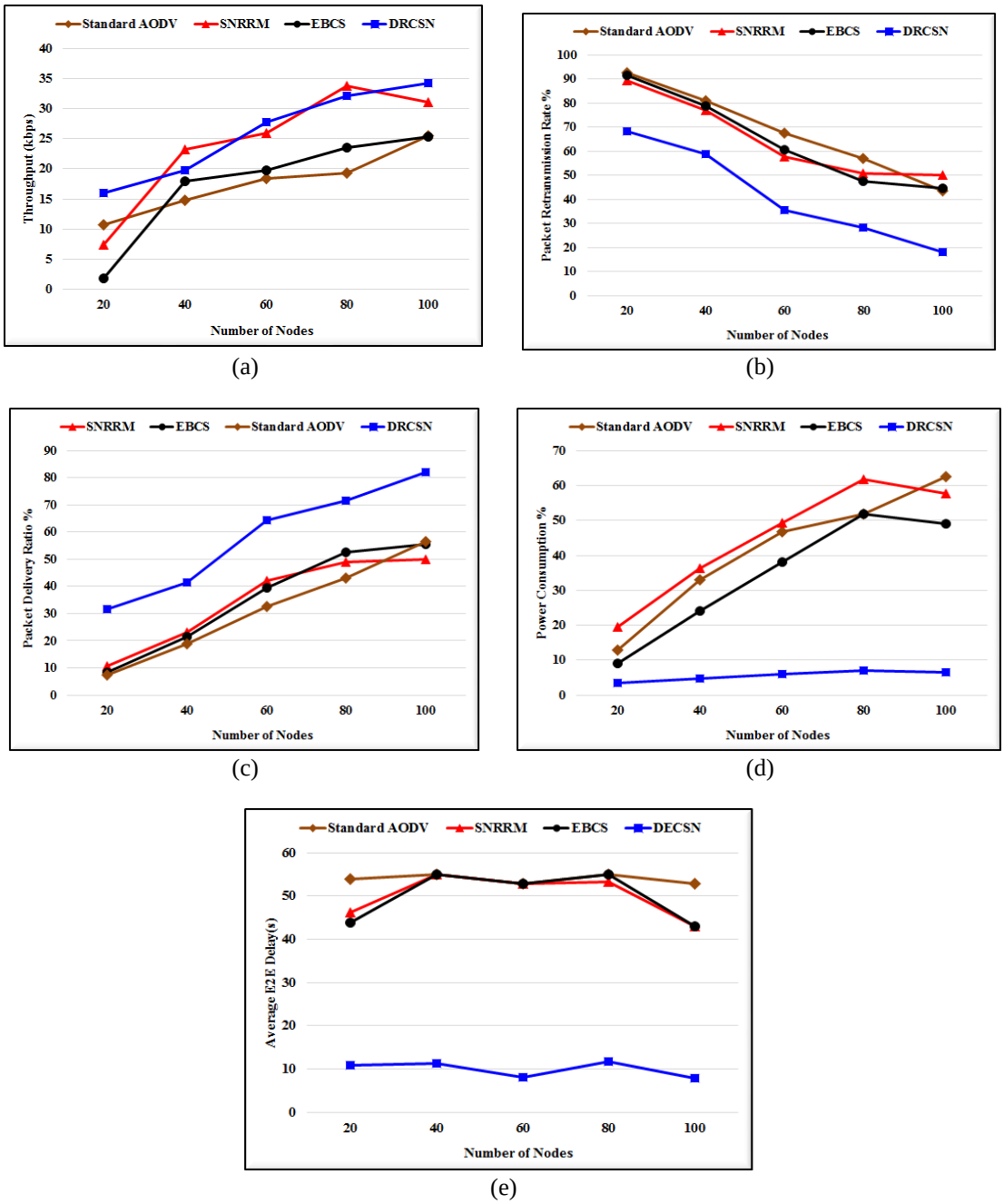

Figure 1. Impact of a number of nodes for DRCSN, standard AODV, SNRRM, and EBCS: (a) throughput, (b) PRR, (c) PDR, (d) power consumption, and (e) average E2E delay

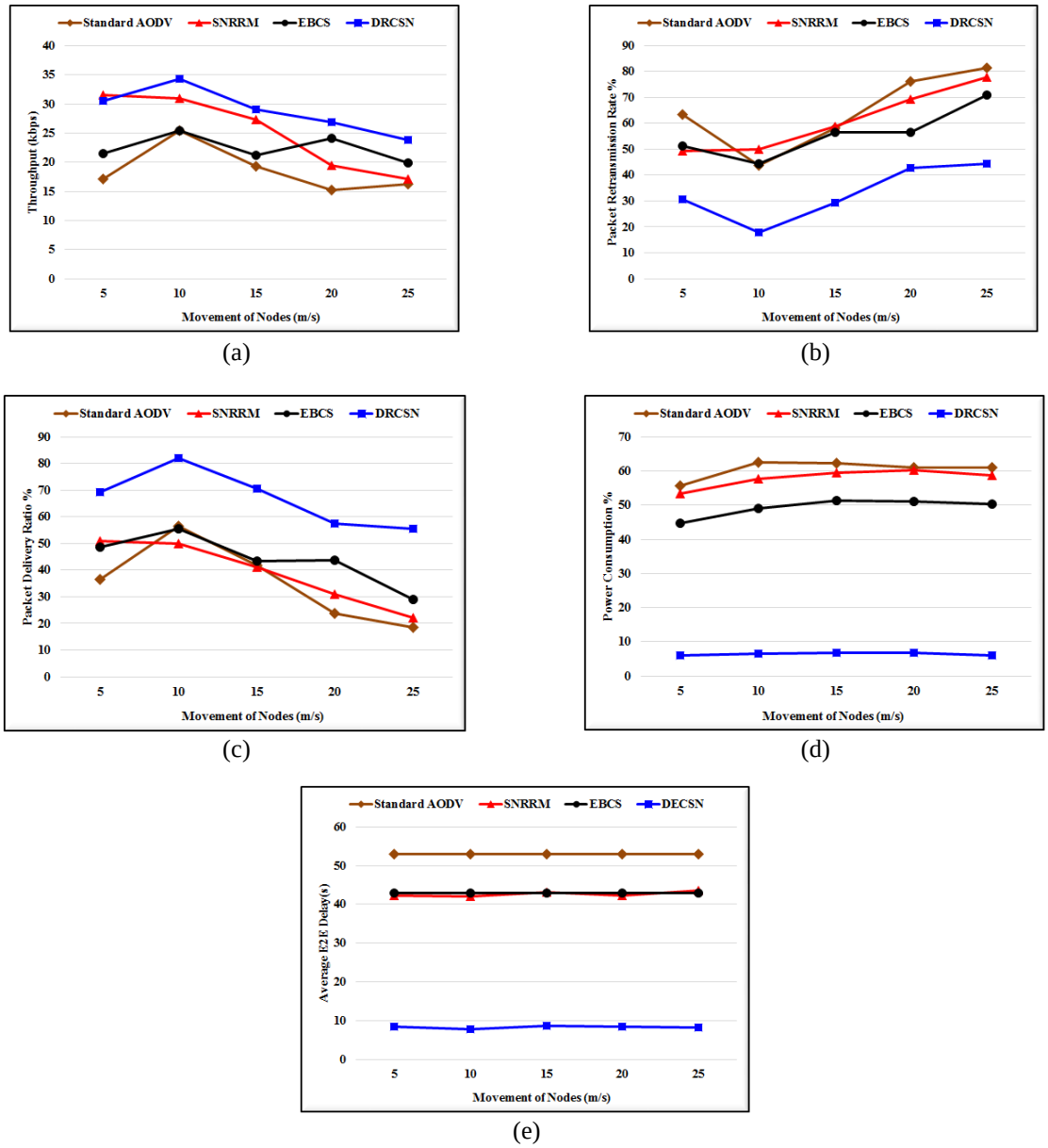

Figure 2. Impact of a variety of movement nodes of DRCSN, standard AODV, SNRRM, and EBCS: (a) throughput, (b) PRR, (c) PDR, (d) power consumption, and (e) average E2E delay

5. CONCLUSION

This study aims to find a approach that improves the selfish nodes in the MANET and makes them cooperative to the maximum extent by developing a new handling method for selfish nodes. The proposed method is named a DRCSN to detect the selfish nodes depending on two factors which are the energy and CR, then make the selfish nodes cooperative and exploit them to the fullest extent by reducing the rate of a packet for the selfish nodes. Also, we adopted a selfish isolate as a specific scheme that aims to delete the ID of the detected selfish node from the AODV routing table. The obtained results demonstrate that the proposed method obtained the best results for the two scenarios (the number of nodes and speed of nodes). When compared with current methods (standard AODV, SNRRM, and EBCS), the proposed method (DRCSN) significantly outperformed when increasing the number of nodes from 20-100 nodes. DRCSN had increased the throughput by 8%, 44%, and 44% from the SNRRM, EBCS, and the standard AODV, respectively. Furthermore, it improved the PDR by 66%, 63%, and 87% from the SNRRM, EBCS, and the standard AODV, respectively. On the other hand, DRCSN decreased the PRR by 35%, 35%, and 38% from the SNRRM, EBCS, and the standard AODV, respectively. While in the case of average E2E delay, it decreased by 80%,79%, and 81% from the SNRRM, EBCS, and the standard AODV, respectively. It is

observed also, the proposed method decreased the power consumption by 86%, 82%, and 85% compared with the SNRRM, EBCS, and the standard AODV, respectively. DRCSN method significantly outperformed the SNRRM and EBCS with a variety of speeds of nodes from 5-25 m/s. It increased the throughput by 16%, 31%, and 61% from the SNRRM, EBCS, and the standard AODV, respectively. Also, it increased the PDR by 71%, 52%, and 91% from the SNRRM, EBCS, and the standard AODV, respectively. On the other hand, it decreased the PRR by 46%, 45%, and 65% from the SNRRM, EBCS, and the standard AODV, respectively. While in the case of average E2E delay, DRCSN decreased by 81%, 80%, and 85% from the SNRRM, EBCS, and the standard AODV, respectively. Finally, the proposed method decreased power consumption by 89%, 86%, and 88% from the SNRRM, EBCS, and the standard AODV, respectively. As a final result, the proposed method achieved its purpose, which is to detect, and reintroduce the selfish node to the network and force it to collaborate. Hence, it enhanced the performance of MANET. Our proposed approach can be extended as future work by taking into consideration MANET parameters that may affect network behavior such as the number of connection and different terrain areas.

REFERENCES

[1] R. Kaur, R. Singla, B. Kaur, and S. Singh, "MANETs: overview, tools, security and applications in health care," *Australian Journal of Basic and Applied Sciences*, vol. 11, no. 8, pp. 1–6, 2017.

[2] K. Susan, K. C., J.-O. A. M., and M. E. S., "An improved token-based umpiring technique for detecting and eliminating selfish nodes in mobile ad-hoc networks," *Egyptian Computer Science Journal*, vol. 44, no. 2, pp. 74–85, 2017.

[3] R. Alubady and H. A. Marhoon, "Enhancing transmission control protocol performance for mobile ad-hoc network," in *AIP Conference Proceeding*, 2019, pp. 1-11, doi: 10.1063/1.5123125.

[4] R. Sivakami and G. M. K. Nawaz, "Secured communication for MANETS in military," in *2011 International Conference on Computer, Communication and Electrical Technology (ICCCET)*, Mar. 2011, pp. 146–151, doi: 10.1109/ICCCET.2011.5762456.

[5] S. Manaseer and A. Alawneh, "Emergency centers set-up in the existence of ad hoc networks in disaster recovery areas," *International Journal of Recent Contributions from Engineering, Science & IT (iJES)*, vol. 7, no. 1, pp. 59–66, 2019, doi: 10.3991/ijes.v7i1.10319.

[6] H. E. D. Mohamed, H. Azmi, and S. Taha, "Using MANET in IoT healthcare applications: a survey," *International Journal of Computing and Digital System*, pp. 1–15, 2021.

[7] S. P. Kumar, D. Magesh, P. N, G. A., and S. Uma, "Collaborate framework based on software defined network in MANET," *Journal of Science Technology and Research (JSTAR)*, vol. 10, no. 6, pp. 818–828, 2022.

[8] S. Dodke, P. B. Mane, and M. S. Vanjale, "A survey on energy efficient routing protocol for MANET," in *2016 2nd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT)*, 2016, pp. 160–164, doi: 10.1109/ICATCCT.2016.7911984.

[9] R.-I. Ciobanu, C. Dobre, M. Dascălu, Ș. Trăușan-Matu, and V. Cristea, "SENSE: A collaborative selfish node detection and incentive mechanism for opportunistic networks," *Journal of Network and Computer Applications*, vol. 41, pp. 240–249, May 2014, doi: 10.1016/j.jnca.2014.01.009.

[10] S. Kumar, K. Dutta, and G. Sharma, "A detailed survey on selfish node detection techniques for mobile ad hoc networks," in *2016 Fourth International Conference on Parallel, Distributed and Grid Computing (PDGC)*, 2016, pp. 122–127, doi: 10.1109/PDGC.2016.7913128.

[11] H. Yadav and H. K. Pati, "A survey on selfish node detection in MANET," in *2018 International Conference on Advances in Computing, Communication Control and Networking (ICACCCN)*, Oct. 2018, pp. 217–221, doi: 10.1109/ICACCCN.2018.8748420.

[12] S. Senthilkumar and J. William, "A survey on reputation based selfish node detection techniques in mobile ad hoc network," *Journal of Theoretical and Applied Information Technology*, vol. 60, no. 2, pp. 208–215, 2014.

[13] R. Mangayarkarasi and R. Manikandan, "Cost effective collaborative anomaly detection system for selfish node attacks in MANET," *Journal of critical reviews*, vol. 7, no. 13, pp. 148–154, Jun. 2020, doi: 10.31838/jcr.07.13.26.

[14] S. Mubeen and S. Johar, "Detection and elimination of the selfish node in ad-hoc network using energy credit based system," *Journal of Network and Information Security*, vol. 7, no. 2, pp. 18–22, 2019.

[15] M. Ponnusamya, A. Senthilkumar, and R. Manikandan, "Detection of selfish nodes through reputation model in mobile adhoc network - MANET," *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, vol. 12, no. 9, pp. 2404– 2410, 2021.

[16] M. M. Musthafa, K. Vanitha, A. M. J. M. Z. Rahman, and K. Anitha, "An efficient approach to identify selfish node in MANET," in *2020 International Conference on Computer Communication and Informatics (ICCCI)*, Jan. 2020, pp. 1–3, doi: 10.1109/ICCCI48352.2020.9104076.

[17] S. J. H. Al-Shakarchi and R. Alubady, "A survey of selfish nodes detection in MANET: solutions and opportunities of research," in *2021 1st Babylon International Conference on Information Technology and Science (BICITS)*, Apr. 2021, pp. 178–184, doi: 10.1109/BICITS51482.2021.9509889.

[18] D. S. Kumari and K. T. Sikamani, "Communication based clustering to detect selfish nodes in MANET," *Indian Journal of Science and Technology*, vol. 8, no. 20, pp. 1–6, Aug. 2015, doi: 10.17485/ijst/2015/v8i20/59236.

[19] S. Deepak and H. Anandakumar, "AODV route discovery and route maintenance in MANETs," in *2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS)*, Mar. 2019, pp. 1187–1191, doi: 10.1109/ICACCS.2019.8728456.

[20] G. Borboruah and G. Nandi, "A study on large scale network simulators," *International Journal of Computer Science and Information Technologies*, vol. 5, no. 6, pp. 7318–7322, 2014.

[21] L. E. Jim, N. Islam, and M. A. Gregory, "Enhanced MANET security using artificial immune system based danger theory to detect selfish nodes," *Computers & Security* , vol. 113, 2022, doi: 10.1016/j.cose.2021.102538.

[22] K. R. Abirami and M. G. Sumithra, "Evaluation of neighbor credit value based AODV routing algorithms for selfish node behavior detection," *Cluster Computing*, vol. 22, pp. 13307–13316, Nov. 2019, doi: 10.1007/s10586-018-1851-6.

[23] S. Sayyar, A. Khan, F. Ullah, H. Anwar, and Z. Kaleem, "Enhanced TWOACK based AODV protocol for intrusion detection

system," in *2018 International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*, Mar. 2018, pp. 1–4, doi: 10.1109/ICOMET.2018.8346444.
[24] N. I. Sarkar and W. G. Lol, "A study of MANET routing protocols: joint node density, packet length and mobility," in *The IEEE symposium on Computers and Communications*, Jun. 2010, pp. 515–520, doi: 10.1109/ISCC.2010.5546763.
[25] A. P. N. S. Lingareddy, M. T, and A. S, "Enhancedsecure sensor protocol with information via negotiation (SSPIN)," *International Journal of Engineering Trends and Technology*, vol. 43, no. 5, pp. 268–273, Jan. 2017, doi: 10.14445/22315381/IJETT-V43P245.

BIOGRAPHIES OF AUTHORS

Sanaa Jafaar Hassan Al-Shakarchi was born in Benghazi, Libya in 1993. She received a Bachelor's degree in Information Technology/Information Networks from the University of Babylon, Iraq in 2015. She is currently studying toward a Master's degree of Information Networks at the University of Babylon, Iraq. She can be contacted at email: sanaa.hasan@student.uobabylon.edu.iq.

Raaid Alubady received his Ph.D. degrees in Information Technology/ Computer Networks from the Universiti Utara Malaysia, in 2017. He is currently an assistant professor in College of Information Technology at University of Babylon-Iraq. Alubady's research and development experience includes over 18 years in the Academia. He is a member of societies; an editor/reviewer of several international academic journals and conferences. Raaid current area of research focuses on the future internet (ICN and NDN), wireless networking/MANET, VANET, internet of things, blockchain technology, fog and cloud computing, and routing protocol. He can be contacted at email: alubadyraaid@itnet.uobabylon.edu.iq.