Received 04 August 2024, accepted 07 April 2025, date of publication 31 March 2026
www.computingonline.net / computing@computingonline.net

Print ISSN 1727-6209
Online ISSN 2312-5381
DOI 10.47839/ijc.25.1.4501

Early Rushing Attack Detection and Prevention in AODV MANETs

TEAMERAT M. ASRES[1], GETANEH A. ZIMBELE[2], TEWODROS M. KIFLE[3]

[1]Department of Computer Science, Haramaya University, Dire Dawa, Dire Dawa, Ethiopia
[2]Department of IT, Debre Berhan University, Debre Berhan, Amhara, Ethiopia
[3]Department of Physics, Dire Dawa University, Dire Dawa, Dire Dawa, Ethiopia

Corresponding author: G. A. Zimbele (e-mail: get.awulachew@gmail.com).

ABSTRACT Mobile Ad hoc Network (MANET) attacks can be classified into active and passive attacks. Among active attacks, the rushing attack is one of the basic network layer attacks. In MANET, it early exploits the duplicate suppression mechanism of Ad hoc on-demand distance vector (AODV) protocol by quickly forwarding the RREQ packet to neighboring nodes without processing it, to influence a source node to include the rushed node in its route, which leads to data loss when transmitting the data packet to the correct destination node. This Early Rushing Attack Detection and Prevention in AODV MANETs (E-RADP) paper is proposed to fill this gap.

To advance the security of our proposed algorithm, threshold value, ratio, and intermediate delay are introduced in rushing attack detection and prevention processes. For the performance analysis, the network simulator NS2.35 is used. The proposed protocol is compared with AODV, Prevention of Multiple Rushing Attacks Using AODV Routing Protocol (PMRA), and Rushing Attack Prevention with modified AODV (MAODV) based on true positive, true negative, false positive, and false negative values of normal and malicious nodes, and throughput, packet delivery ratio and end-to-end delay. E-RADP improves the rushing attack detection rate (DR), throughput, and packet loss rate (PLR) of existing protocols. It also improves the end-to-end delay (E2ED) of existing protocols when a rushing node is present in a MANET. Thus, the performance analysis shows that E-RADP is highly secure and faster than existing algorithms.

KEYWORDS AODV, Intrusion Prevention, MANET, Ratio, RREQ Time, Rushing Attack, Time Stamp.

I. INTRODUCTION

A preprint of this manuscript has previously been published at [1]. Wireless communication technology has grown fast in the past few decades. This rapid growth appears because people must communicate anywhere and anytime with their mobile devices. Wireless networks use radio waves in the air to interact with each other and to exchange data instead of using dedicated cables. Wireless networks are classified into two categories. One is infrastructure-based networks like GSM cellular networks and the second category is ad hoc wireless networks such as MANET [2]-[5].

Recently, MANETs have progressed as one of the crucial next-generation wireless network technologies like Vehicular Ad Hoc Networks (VANETs), Internet of Vehicles (IoV), unmanned aerial vehicles (UAVs), and Flying Ad Hoc Networks (FANETs) [2], [4], [6]. It is MANETs are characterized by (a) wireless transmission, (b) each node acting as a host and router, (c) decentralized administration and infrastructure-less network, (d) dynamical change of topology completed by regular route updates, (e) easy and low-cost deployment, (f) cooperative working and distributed network, (g) self-configuring, and self-managing, (h) scalability of a network, (i) minimum intervention of human for configuration of a network, (k) heterogeneity of nodes, (o) multi-hop mobile devices, (l) limited resources in a device, (m) more vulnerable to security issue than a wired network, and (n) bandwidth and energy constraints [4], [11]. MANET uses a routing protocol to establish, maintain, and repair the optimal routing paths between mobile nodes.

Routing is a mechanism that helps MANETs to move data from one mobile node to other nodes within the network. Previously, several competent routing protocols have been proposed. These protocols can be classified into proactive (table-driven), reactive (on-demand), and hybrid routing protocols [5], [8], [12], [13]. In proactive protocols, each node keeps routing information in its routing table and periodically updates its routing information by exchanging routing information with other nodes. Optimized Link State Routing Protocol (OLSR) and Destination Sequenced Distance Vector (DSDV) routing protocols are considered table-driven routing

protocols. In this protocol, there is a challenge of performance degeneration as the network size increases, which is due to increases in control overhead [5], [8], [12], [13]. In reactive protocols, each node will start the route, and update its routing information only when a new routing path is required instead of periodically updating. This includes dynamic source routing (DSR) and AODV routing protocols. The benefit of this protocol is that unused bandwidth brought from cyclically broadcast becomes decreases. This protocol may lead to packet loss [5], [8], [12], [13]. Hybrid routing protocols combine the advantages of both proactive and reactive routing protocols. Zone Routing Protocol (ZRP), core extraction distributed ad-hoc routing (CEDAR) protocol, and Temporally-ordered Routing Algorithm (TORA) protocols are examples of this type of protocol. This protocol uses the property of proactive protocol to gather the unidentified routing path, then it uses the property of reactive scheme to maintain the routing path when there exists a change in the topology [5], [8], [12], [13].

AODV is the prominently used reactive routing protocol designed for MANETs where nodes can enter and leave the network at will. It is both a unicast and multicast routing protocol that uses several control packets like route request packet (RREQ), routing reply message (RREP), route error message (RERR), HELLO messages, and Sequence numbers. In DSR nodes maintain their route cache from source to destination node. The performance of DSR decreases as the mobility of the network increases, with a lower packet delivery ratio within the higher network. In AODV, each node records the information of the next hop in its routing table. It finds a route to a destination when a source node likes to transfer a packet to that destination [5], [8], [12].

The route discovery process is executed when the destination node can't be reached from the source node. The source node broadcasts the route request (RREQ) packet across the network to start the route discovery process. Nodes receiving this packet update their information for the source node and set up backward pointers to the source node in the route tables. In addition to the source node's IP address, current sequence number, and broadcast ID, the RREQ also contains the most recent sequence number for the destination of which the source node is aware [5], [8], [12].

All the nodes receive the RREQ packets unicasts back the route reply (RREP) packet to the source node if it is either the destination or if it has a route to the destination with a corresponding sequence number greater than or equal to that contained in the RREQ. If this is the case, it unicasts an RREP back to the source. Otherwise, it rebroadcasts the RREQ. Nodes keep track of the RREQ's source IP address and broadcast ID. If they receive a RREQ that they have already processed, they discard the RREQ and do not forward it. As the RREP propagates back to the source, nodes set up forward pointers to the destination. Once the source node receives the RREP, it may begin to forward data packets to the destination. If the source later receives an RREP containing a greater sequence number or contains the same sequence number with a smaller hop count, it may update its routing information for that destination and begin using the better route. As long as the route remains active, it will continue to be maintained. A route is considered active as long as there are data packets periodically traveling from the source to the destination along that path. Once the source stops sending data packets, the links will time out and eventually be deleted from the intermediate node routing tables [5], [8], [12].

The Route Maintenance process is started when the network topology has changed or the connection has failed. If a link break occurs while the route is active, the node upstream of the break propagates a route error (RERR) message to the source node to inform it of the now unreachable destination(s). After receiving the RERR, if the source node still desires the route, it can reinitiate route discovery AODV protocol never produces routing loops by proving that a combination of sequence numbers and hop counts is monotonic along a route. HELLO messages are used to find active neighbors. Sequence numbers are used to find the freshness of routes toward the destination [5], [8], [12].

MANET security is the protection of mobile nodes and their communication system. It is a big challenge and highly vulnerable to different types of attacks as compared with wired and infrastructure-based wireless networks. It is due to a lack of central administration, use of unguided and shared radio waves as transmission medium, dynamic topology, limited resources, and other characteristics of MANETs [8]. Utmost applications wherever MANET is used are very serious and the data collected from them are confidential and valuable [9], [10], [12], [14]. To assure both confidentiality and data integrity, data security and route security are the two broad categories of MANET security. Due to a deficiency of security, there are numerous routing attacks [5], [8].

Attacks in MANET can be categorized based on various standards such as domain or source, the behavior or nature of the attack, the layer of the network an attack will be done, processing capacity, and the number of attackers. Based on the nature of the attack, it is classified into active attacks and passive attacks. In passive attacks, intruders having insufficient abilities perform traffic analysis, eavesdropping (reads the message), etc. without altering the original data. It includes attacks like Traffic Monitoring, Eavesdropping, Traffic Analysis, Sync flooding, snooping, and Jellyfish attacks. Active attacks on the other hand are characterized as very severe attack in which opponents have advanced attacking tools that gathers and modifies information transmitted over the network and sends the modified data to a destination. They may also corrupt the system functionality totally by altering links, routing, and topology. It includes attacks like Byzantine, Rushing, Black hole, Replay, Location disclosure, Flooding, Sink Hole, Spoofing, RERR Generation, Jamming, Sybil, Desynchronization, Overwhelm, Blackmail, Denial of service (DoS), Distributed DoS (DDoS), Selfish Nodes, Man-in-the-middle, Fabrication, Sleep Deprivation, Route Salvaging, Impersonation, Gray-hole, and wormhole attacks [6], [8], [9], [12], [15], [16]. Among these active attacks, a rushing attack is one of the most widely used effective Denial-Of-Service (DoS) attacks in which a malicious node early exploits the property of the normal routing process [3], [8], [9], [13].

The rushing attack, one of the basic active and network layer attacks in MANET, early exploits the duplicate suppression mechanism of AODV and consequences data loss while delivering the data packet to the correct destination. In this attack, when the malicious node receives the route request packet (RREQ), it quickly forwards the RREQ packet to its neighbor nodes without processing the RREQ packet to influence a source node to include the malicious node in its route. On the other hand, the normal nodes check the delay and send the packet to its neighbors [10]. The routing protocols depend on transmission between mobile nodes governed by the absence of a central administration and trust that no malicious

intruder i.e. every node is honest and well-behaved. A malicious node may begin routing attacks to disrupt routing processes, or a Denial-of-Service (DOS) attack to deteriorate services to normal nodes [6], [8], [10], [14], [17]. Rushing (forward motion) attacks are primarily against the Active (on-demand) routing protocols by changing the process of route discovery [12], [14].

In a rushing attack, the malicious node may be neighboring to the sender, neighboring to the receiver, or anywhere in the MANET [12], [14]-[16]. In the AODV routing protocol, when intermediate nodes receive RREQ packets, they should forward only the first received RREQ for route discovery to neighbor nodes and they consider all further received similar RREQ packets as duplicates and ignore them. Consequently, a Rushing node just adventures this property of the route discovery process by quickly forwarding RREQ packets to its neighbor nodes to be part of the route. It is extremely difficult to detect such malicious nodes. As a result, it will be difficult for a source node to discover any truthful routes without the Rushing node [5], [6], [8]. As shown in Figure 1, in rushing Attack, RREQ sequence numbers are multiplied by the malicious node. Then, the reactive protocols maintain the sequence numbers for suppressing replica packets at the nodes [6]. In this regard, this research is proposed to address the problem of rushing attacks against AODV-based MANETs.

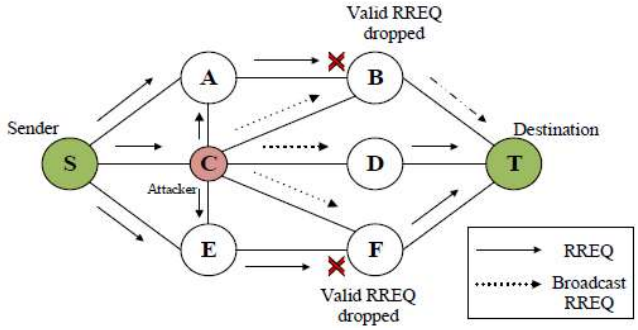

Figure 1. Rushing attack [6]

Depending on our review of the behavior of the Rushing attack and AODV protocol, the detection and prevention mechanism should be depending on the early RREQ packet and RREQ time stamp. However, there are some security countermeasures to these attacks, the existing countermeasures are usually too late or inefficient to take effect [10]. To address this shortcoming, our study effectively uses RREQ time and hop count (HC) as a ratio within the RREQ packet header.

The rest of the paper is organized as follows: Section II discusses important related works on rushing node detection and rushing attack prevention techniques. The proposed method, E-RADP, is introduced in Section III. In Section IV the performance of the proposed algorithm is compared with state-of-the-art related works. The contribution of this work is presented in Section V. Finally, Section VI presents a conclusion and recommendations for future work.

II. RELATED WORK

This section discusses the work done by various researchers on rushing attack prevention and mitigation techniques.

Authors in [18] (2014) proposed a method called "Rushing Attack Prevention with modified AODV in Mobile Ad hoc Network" that tries to reduce the effect of the Rushing attack by modifying some properties of AODV (MAODV). Instead of forwarding the first RREQ, the authors modify AODV to exploit the property of AODV, their work stores some RREQs at each node and forwards a randomly selected one from those stored RREQs as shown in Figure 2. Since their method uses waiting time for collecting some number of RREQs, it induces more delay. Another limitation of this approach is; due to any criterion that is not used for the selection of stored RREQs, their random selection method will select a rushing attack, and their modified method requires more storage and power to store those RREQs for future random path selection, which is not good in MANATET nodes.

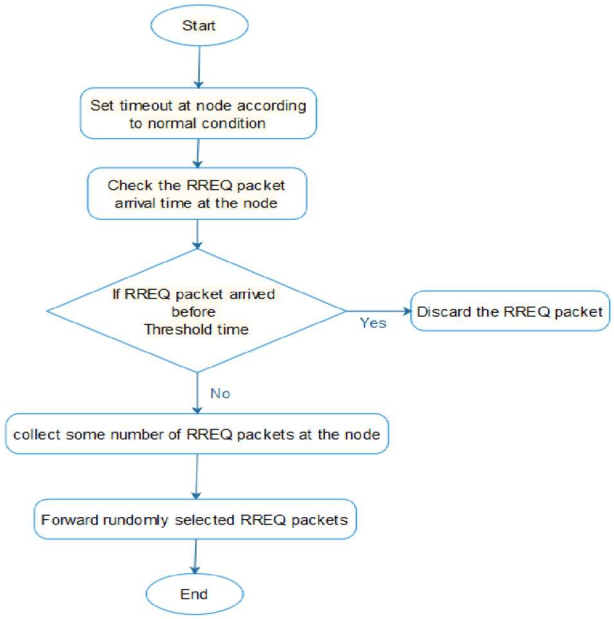

Figure 2. Prevention of rushing attack in MAODV [18].

Authors in [11] (2015) proposed an algorithm called "Prevention of Multiple Rushing Attack Nodes in Multicast MANET". This method tries to reduce the transmission cost while transmitting the same set of data packets to multiple receivers is an efficient mode of communication. The shortcoming of this method is that it does not properly address the rushing attack problem because a rushing node might be a member of the multicast group. Another limitation of this method is that its technique will work only for registered nodes, which is not always applicable in MANET.

Secure AODV Protocol (SAODV) is a modification of the AODV protocol [19] (2017). The authors advise using a pre-established public key infrastructure (PKI) in which every node in MANET would have public key pairs. The drawback of this method includes: it is more resource intensive for key generation and management. It is difficult to get and implement a trusted third party for MANET as the concept of MANET is broadcast based on the trust of every node. Since it assures confidentiality, it is possible to reject unwanted packets through hop count increment by an arbitrary number; since the rushing attack is an early problem during RREQ for route establishment, it does not prevent rushing attacks because this method has more focus on data packet protection of confidentiality and integrity.

Authors in [19] (2017) have proposed an enhanced algorithm named "Rushing Attack Prevention Algorithm for MANET using Random Route Selection" (RRS) based on SAODV to make AODV more efficient. This algorithm is

based on random route selection and time. First, this algorithm also uses a random path for every data transmission so that it reduces the malicious nodes from continuing to harm the data transmission. Second, this algorithm will compute the average transmission time to the adjacent node from the source. While the time taken by any packet is less than the average time, all packets will be discarded by the node. Therefore, the packets received by the adjacent nodes after taking at least an average time will only be acceptable by the node. Thus, this method as its bench SAODV does not well prevent rushing attacks because this method is also more effective during data packet transmission than route establishment stage. This method also transmits and processes each packet to all adjacent nodes, which requires more storage and processing costs, and reduces throughput. Another limitation is that; it also drops more packets which reduces the performance of the Packet Delivery Ratio (PDR).

Authors in [20] (2018) proposed a method called "Prevention of Multiple Rushing Attacks in MANET Using AODV Routing Protocol (PMRA)" only by setting a time threshold value (0.02) [21] using AODV routing protocol to improve the security of the existing standard AODV algorithms. In this method, if the RREQ packet arrival time is less than or equal to the time threshold value, it accepts the RREQ packet as the PMRA algorithm shown in Figure 3. This is contradictory to the property of a rushing attack, which is based on the advantage of the duplicate suppression method in which rushing nodes send packets quicker than normal nodes. Another limitation of this node is that it uses a constant threshold value of insecure MANET, which will not be appropriate for Secured MANET, which significantly takes more time to prevent security attacks. Therefore, the drawback of this method is that it will accept malicious nodes as normal nodes since they will take more time than the threshold value of normal MANET.

Authors in [22] (2020) proposed a node trust evaluation-based method called "Detection of Rushing Attack and Data Modification Attack in Mobile Ad Hoc Networks (DRRDMA)," to detect rushing attacks, route modification attacks, and data modification attacks. In this method, each node's trust is based on the number of packets sent, number of packets delivered, packets dropped, delayed, and mobility methodology. To achieve its objective, this method uses a trust table, which contains information about each node's trust value within the range of 0 to 1, and a table for network behavior analysis. This method decreases the trust value by 0.1 in every case of packets dropped and mobility change. It considers 0.8 to 1 as trusted and 0 to 0.7 as untrusted nodes. The drawback of this method includes; it requires more resources for storing and analyzing the two tables. The Rushing node will initialize its trust value as the most trusted value 1. This method is based on the packet transmission stage, but not on RREQ, which is not a good method to prevent an early rushing attack, which is based on the route discovery RREQ behavior of AODV protocol. However, the rushing attack is based on RREQ time advantage, this method does not consider time value, and it is not a good technique to prevent and mitigate the rushing attack in MANETs with more mobility.

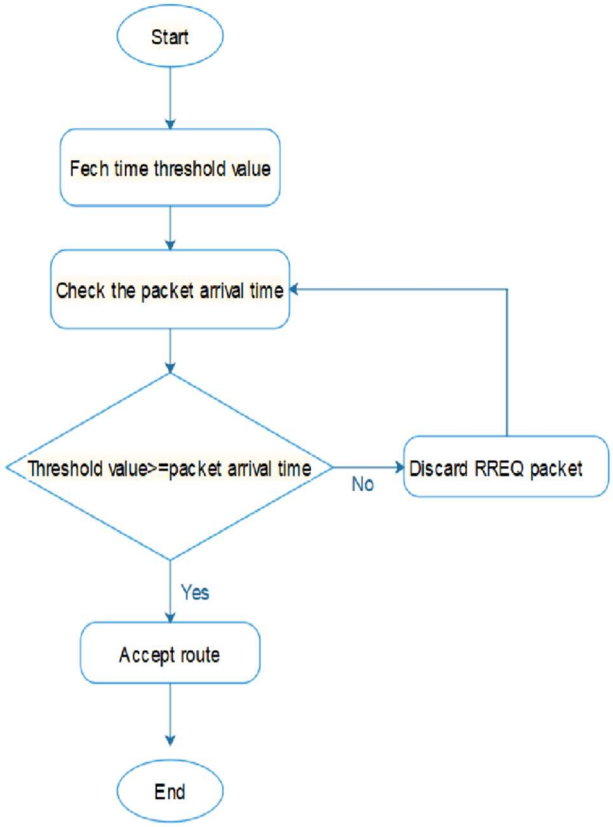

Figure 3. Prevention of multiple rushing attacks [20].

Authors in [17] (2020) proposed a method called "Prevention of rushing attack in MANET using the threshold-based approach for DSR". The protocol is based on accepting the entire RREQ packets instead of accepting the first RREQ packet, which reduces the network overhead. To overcome the DSR problem, this method changes the property of the DSR protocol by:

- First, instead of accepting only the first RREQ packet, the entire RREQ packets are accepted by the destination node.
- Second, the destination node stores the entire RREQ packets in its routing table.
- Then, the destination node will compute the delay of RREQ packets.

In this method, Delay is computed as the overall (source to destination) delay divided by the number of hops. If the delay value is less than the threshold value, this method assumes an attacker exists within the path. Then, all nodes identify the rushing node, and a warning sign is sent to all other intermediate nodes within the path. This method tries to improve the existing standard DSR protocols by using a threshold value, and overall delay of RREQ packets. This method uses a paper [23] that proposed a "threshold-based approach to prevent a rushing attack in MANET" as a benchmark. Based on this paper, every mobile node takes strictly five seconds to transmit a packet and as a result, five seconds is considered as a threshold value for this MANET. Any mobile node delivering packets before five seconds is assumed as a rushing node and the neighbor node will send an alert to all other nodes in the path [17]. The limitation of this approach includes; it is based on the reactive protocol DSR. It uses only a constant threshold value of MANET without security future that results in a rushing node to take more time

than the threshold value and is perceived as being a normal node by the destination node. Another limitation of this method is the overall delay time of the path containing a rushing node may be greater than a threshold value because of the security processing overhead induced by normal nodes, as a result, it can be considered as a correct path. Therefore, a single evaluation method based on threshold value only is not a good method for rushing node prevention techniques [17].

In general, the security of related work was dependent on secure neighbor detection (the normal direct wireless communication range), secure route delegation (neighbor verification), randomized message forwarding, secure route discovery (randomized selection of RREQ), integrating secure route discovery with secure AODV (set a flag to indicate that it wants to use rushing attack prevention), or trust mechanism [5], [22].

Secure neighbor and secure route delegation methods are not adequate to detect the rushing node, because an attacker can still acquire an advantage by sending route request packets very quickly [5], [22]. Based on these detection and mitigation techniques, it will be challenging to discover an optimal and accurate route to the destination node. Forwarding data packets through the non-optimal path or selectively dropping packets will not solve the rushing problem, because RREQ packets routed by the rushing node will be still delivered before any other normal node RREQ in the MANET [7], [9].

Since trust methodology evaluates the trust of each node based on the number of forwarded packets, the number of received packets, packet drop, packet delay, and mobility, it is too late to detect and mitigate rushing attacks. Because these are not based on RREQ time, rushing nodes compromises RREQ time. Hence, this method generates two tables namely the trust table and the network behavior analysis table, it is not a good method for MANET having storage limitations in its nodes [9], [22].

A review of related work shows that the existing rushing attack detection and mitigation methods tried to detect and reduce rushing attacks. Some of the existing work uses a single constant threshold value taken from standard AODV, which is not appropriate for security-based AODV for evaluating rushing node behavior [17]. Some of them need special devices and more resource costs, some of them do not consider high mobility nodes, some of them like SAODV require PKI where it is difficult to get a secured trusted third party in MANET, and most of them do not consider RREQ time, and some of them effective during data packet transmission than route establishment stage.

From our review of related works, we have found that MAODV [10], PMRA [20], and Modified DSR Protocol (MDSR) use a better method based on the property of rushing attack. However, MDSR is not based on the AODV routing protocol rather it is based on the DSR protocol. Therefore, we have selected MAODV, and PMRA as better attempts based on the properties of AODV and rushing attack. Nevertheless, even these algorithms have security, speed, and packet delivery performance drawbacks that could be addressed.

Hence, to address these shortcomings, this work attempts to propose a new security scheme called E-RADP, and its security, speed, and packet delivery performance are compared with state-of-the-art related works: MAODV, PMRA, and with common and popular protocol AODV. A preprint of this paper has previously been published in [1].

III. PROPOSED SYSTEM

In this section, we present the proposed early Ratio of RREQ Time and hop count-based rushing attack detection and prevention for AODV MANETs (E-RADP) protocol.

Generally, all existing works use insufficient parameters like single constant threshold value and hop count. These are more effective during data packet transmission than route establishment stage. None of them consider RREQ time parameters like previous node time stamp, packet arrival time, and their relationship which helps to detect and prevent rushing attacks from joining the network early during the route discovery stage. This makes existing works inefficient in detecting and preventing rushing attacks before being part of the routing path. To address these shortcomings, this work introduces new security parameters “Ratio of RREQ Time(R), Previous Node Time Stamp of RREQ packet (PTS) and HC” in addition to a constant threshold value 5 used by [17] and [23], and time out period introduced by existing methods. These new security parameters help our protocol for selecting a reliable routing path that is free from rushing node (RNode) and to improve the security, speed, and packet delivery performance of state-of-the-art related work.

Like AODV, the basic steps used in this proposed protocol are route discovery, rushing node prevention, route establishment, and data forwarding processes with our new modifications in the processes. Like other works, sub-processes performed by our protocol within these basic steps include: generating RREQ Packets, forwarding RREQ packets, detecting rushing nodes (RNode), selecting a reliable RNode free route, and replaying RREP packets.

In the Route discovery process, steps to be computed by the source node (SNode) are as follows: First, the RREQ packet is generated. Second, the two parameters ratio(R) = threshold value (TH) 5 seconds [17], [23], and previous node time stamp (PTS) = time stamp of RREQ packet (TS) has been computed. Third, R and PTS are added to the RREQ packet header. Fourth, broadcast the RREQ packet.

In the Route discovery process, steps to be computed by the intermediate node (INode) are as follows: First, INode receives RREQ packets from its predecessor neighbor node and fetches RREQ packet arrival time (PAT). Second, the intermediate delay of the RREQ Packet (IDelay) is computed as IDelay = (PAT – PTS). Third, IDelay is compared to the TH value to prevent RNode near the SNode. Then, if the condition is satisfied, the predecessor or source node is detected as RNode and the RREQ packet has been dropped. This prevents rushing nodes from joining the network. Fourth, if the condition in the third step is not satisfied, the Intermediate Ratio (IR) is computed as IR= (PAT-TS) / HC. Fifth, IR is evaluated if it is less than R fetched from the RREQ header. Then, if the condition is satisfied, the predecessor node is detected as RNode and the RREQ packet has been dropped. This prevents rushing attacks anywhere within the path before reaching the destination node. Sixth, if both conditions in the third and fifth step are not satisfied, INode will compute R=(R+IR)/2 and PTS=PAT, then it updates R and PTS in the RREQ packet header, and finally, broadcasts the RREQ packet to the successor neighbor or destination node.

In the Route discovery process, steps to be computed by the destination node (DNode) are as follows: First, the DNode receives RREQ packets from the INode and fetches PAT. Second, the IDelay of the RREQ Packet is computed. Third, Idelay has been evaluated if it is less than TH. Then, if the

condition is satisfied, the predecessor or source node is detected as RNode and the RREQ packet has been dropped. This prevents rushing attacks near the DNode. Fourth, if the condition at the third step is not satisfied, the IR is computed. Fifth, IR has been evaluated if it is less than R fetched from the RREQ header. Then, if the condition is satisfied, the predecessor neighbor node of DNode is detected as RNode and the RREQ packet has been dropped. This prevents rushing attacks from anywhere within the path before reaching the DNode. Six, if both the above conditions in the third and fifth steps are not satisfied, DNode selects the path through which the RREQ packet is received as the best route.

During the route establishment process, the steps to be computed by the DNode are as follows: First, the DNode fetches R from the RREQ packet header and computes a new R. Second, it updates R in the RREP packet header. Finally, it replays the RREP packet to SNode through the best route.

During the route establishment process, steps to be performed by the INode are as follows: First, all INodes in the RREP path added or updated a new best route in their routing table. Second, they update the R of their routing table with the R of RREP Packets. Finally, they forward RREP packets to their predecessor through the path to the SNode.

During the route establishment process, steps to be performed by the SNode are as follows: First, The SNode adds or updates a new best route in its routing table. Finally, it updates the R of its routing table to the R of the RREP packet.

During the data forwarding process, steps to be performed by the SNode, INode, and DNode are as follows: First, the SNode checks for the availability of the routing path. Then, if the path is available, it forwards the data packet to the neighbor INode or DNode. Then, INode searches for a route path from its routing table and forwards to the next neighbor or destination node. Finally, DNode captures a data packet. If a route is not found in any of the INode, it replies route error (RERR) packet back to the SNode, and the SNode starts the route discovery process.

The Pseudo code for the proposed system is disclosed in Algorithm 1. Figure 4 further illustrates the route discovery and establishment architecture of the proposed protocol. As route maintenance and data packet forwarding processes are similar to the ordinary AODV, we did not include them in our protocol's pseudo code and architecture design.

Algorithm 1: E-RADP Pseudo Code for Best Route Establishment

Route Discovery Process Computed By SNode:

1. Source node (SNode) generates RREQ packet.
2. It computes Ratio(R) = Treshold Value (TH) = 5, and Previous node Time Stamp (PTS) =Time Stamp of RREQ packet (TS).
3. It adds R and PTS in the RREQ packet header.
4. It broadcasts the RREQ packet to its neighbor nodes

Route Discovery Process Computed By INode:

1. Intermediate node (INode) receives the RREQ packet and fetches the RREQ Packet Arrival Time (PAT) from the RREQ Packet.
2. It computes the Intermediate Delay (IDelay) of RREQ Packet= (PAT – PTS).
3. If (Idelay<TH) then RNode is detected and the RREQ packet from Rnode is dropped. //this prevents RNode near the SNode
4. If the condition in step 3 is not satisfied, it computes Intermediate Ratio (IR) = (PAT-TS) / HC. Then,
5. If (IR < R) then RNode is detected and the RREQ packet from Rnode is dropped //this prevents RNode from anywhere within the Path before reaching the DNode
6. If both conditions in steps 3 and 5 are not satisfied
   - It computes R=(R+IR)/2 and PTS=PAT.
   - It updates R and PTS in the RREQ packet header.
   - Finally, it broadcasts the RREQ packet to its neighbor nodes.

Route Discovery Process Computed By DNode:

1. DNode receives the RREQ packet and fetches PAT from the RREQ Packet
2. It computes IDelay = PAT – PTS
3. If (Idelay<TH) then RNode is detected and the RREQ packet from Rnode is dropped //this prevents RNode near the DNode
4. If the condition in step 3 is not satisfied, it computes IR = (PAT-TS) / HC
5. If (IR < R) then RNode is detected and RREQ packet from Rnode is dropped //this prevents RNode from anywhere in the path.
6. If both conditions in steps 3 and 5 are not satisfied, It selects the path as the best route path

Route Establishment Process Computed By DNode:

1. It fetches R from the RREQ packet header and computes new R=(R+IR)/2
2. It updates R in the RREP packet header, and unicast the RREP packet to SNode through the path.

Route Establishment Process Computed By INode:

1. All INodes in the RREP routing path add or update the new best route in their routing table and Update R of the routing table to R of the RREP Packet.
2. They update the R of their routing table with the R of RREP Packets.
3. They unicast RREP packets to their predecessor or SNode through the selected path.

Route Establishment Process Computed By SNode:

1. The SNode adds or updates a new best route in its routing table.
2. It updates the R of its routing table to the R of the RREP packet

Figure 4 shows a flow chart for route discovery and establishment steps. As shown in this figure, in addition to a constant threshold value of 5 used by [17] and [23], HC, TS, PAT, and time out period introduced by existing methods, the E-RADP protocol uses a new security parameters R, PTS, IDelay, and IR". Finally, it isolates RNodes from the network and selects RNode free paths as the best route to forward data packets.

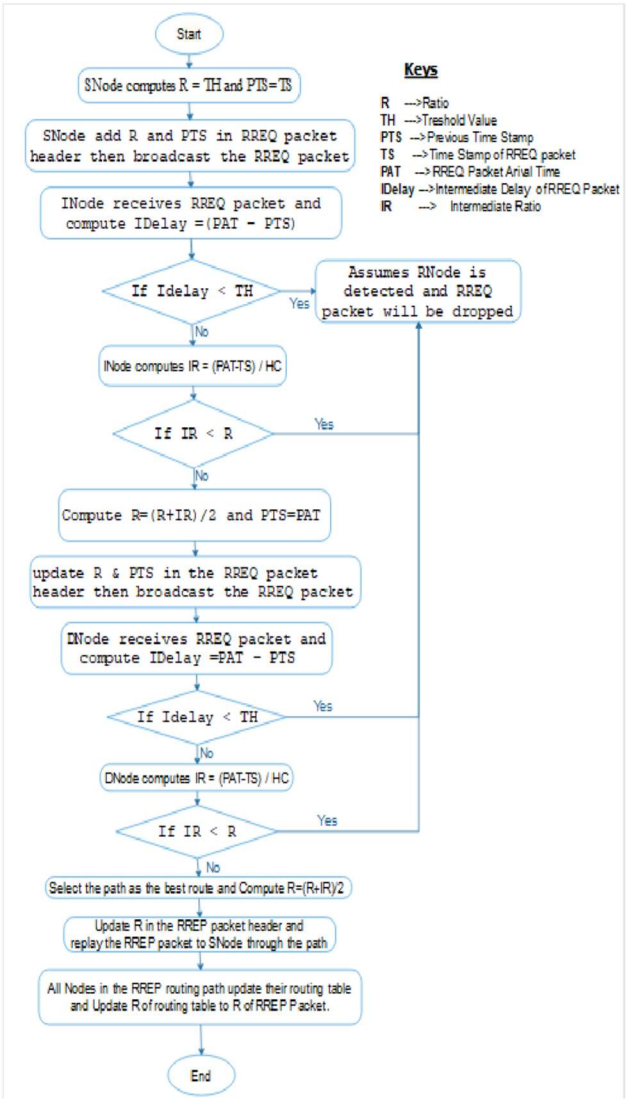

Figure 4. E-RADP Architecture.

IV. IMPLEMENTATION

A. SIMULATION SETUP

To conduct our experiment, we used NS-2.35 installed on Ubuntu 20.0.4 which was running on Intel(R) Core(TM) i5-6200U @ 2.30GHz (4 CPUs) and 4 GB RAM. To simulate the protocols, we have chosen the default transmission range 250m of NSG2.1 (number of nodes*25) as the simulation area, simulation time of 100 seconds, and 10, 30, 50, 80, and, 120 nodes in which among them 20% of nodes are rushing nodes. The queue for the interface has a maximum size of 512 packets and the IEEE 802.11 MAC layer protocol is used. As our research area is highly affected by the mobility factors and the location of rushing nodes, the random waypoint mobility model (RWP) is selected from various mobility models like the random walk mobility model, Gauss-Markov, and others [23]. Nodes linked from source to destination are randomly set up through a random waypoint mobility model. Both FTP and CBR flow rates are used as traffic with a packet size of 512, 1000, or 1500 bytes. AODV protocol is used for routing purposes. To generate a random waypoint mobility model, there is a node movement generator script which is found in ~ns/indep-utils/cmu-scen-gen/setdest directory [23].

The random flow of traffic is generated through the "cbrgen.tcl" script located in ~ns/indep-utils/cmu-scen-gen. To generate TCP/CBR traffic load on the network, we used cbrgen.tcl script: ~ns/indep-utils/cmu-scen-gen ns cbrgen.tcl [-type cbr|tcp] [-nn nodes] [-seed seed] [-mc connections] [rate rate] [23]. As shown in Figure 5, we have created a CBR connection for a random number of nodes with a data rate of 0.1Mb packets per second using cbrgen.tcl script: ns cbrgen.tcl -type cbr -nn 80 -seed 1.0 -mc 5 -rate 0.1Mb $ > cbr-80-nodes.

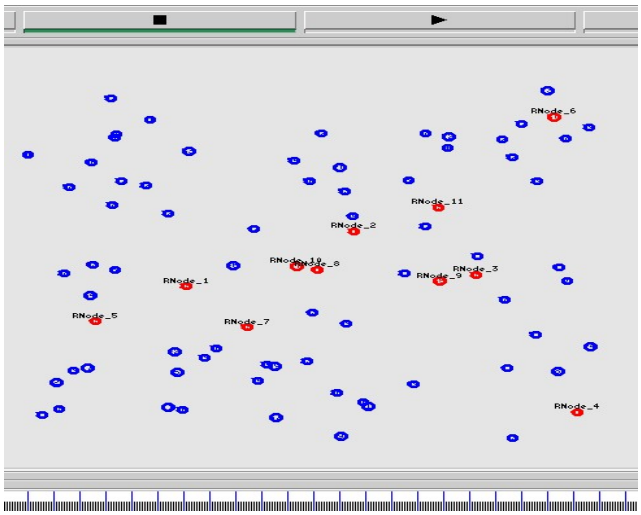

Figure 5. 80 Nodes simulation of MANET using NAM

As the two-ray ground propagation model is used for long distances, and the default ns2 model to predict the received signal power of the packet [24], we have selected it for our simulation [24], [25].

B. SIMULATION SCENARIO

To compare the performance of our proposed protocol, ordinary AODV, MAODV, and PMRA, our simulation is carried out in two different scenarios. In the first scenario, we make for security analysis with 10, 30, 50, 80, and 120 mobile nodes from which up to 20% were randomly selected as malicious nodes. Traffic sources and destinations, nodes between them, and their number vary and are randomly selected. In the second scenario, the network automatically generates a heterogeneous number of normal and malicious nodes from 1 up to 80 nodes. We used this scenario to analyze the impact of network density and simulation time on the protocol's packet delivery and speed performance. As shown in Table I, in both two scenarios, transmission range, simulation time, packet size, and packet rate values are constant.

Table 1. Parameters

Parameter	Value
Simulator	Network Simulator 2 (NS2)
Simulation Size	(# of nodes) m * (# of nodes) m

Simulation Time	150 seconds
Number of nodes	10,30,50,80,120
Propagation Model	Two-ray ground propagation
Node Movement	Random waypoint
Speed of Nodes (m/s)	5-20m/s
Transmission range	250m
Traffic type	CBR, FTP
Packet Size	512 bytes
Packet Rate	0.1Mb /sec
Transport protocol	UDP, TCP

C. NS2 SETUP

Based on the review in [26], NS2 is a popular open-source network simulator tool that is designed to simulate network protocols [27]. Therefore, we have used it for our simulation.

To create mobile nodes, we need to set the value of the parameters in the protocols TCL script [28]. Part of the script that shows the parameters set in our E-RADP.tcl file is:

```
set val(chan)   Channel/WirelessChannel;    # Channel Type
    set val(prop) Propagation/TwoRayGround;  # radio-
                  propagation model
    set val(netif)   Phy/WirelessPhy; #network interface
                  type
    set val(mac)    Mac/802_11;                 # MAC type
    set val(ifq)  Queue/DropTail/PriQueue; # interface
                  queue type
    set val(ll)       LL;     #          link layer type
    set val(ant)      Antenna/OmniAntenna; # antenna model
    set val(ifqlen)   512 ;  # max packet in ifq
    set val(nn)       80 ;     # number of mobile nodes
    set val(rp)       AODV;     # routing protocol
    #set val(rp)      DumbAgent ;
    set val(x)        [expr $val(nn)*25]; # X dimension of
                  topography
    set val(y)        [expr $val(nn)*25]; # Y dimension of
                  topography
    set val(stop)     $val(nn);        # time of simulation end
    set src-dest      2    # number of sources to destination
    pairs
```

Our TCL file defines wireless parameters listed in Table I. The flat topology having only x and y is created in our TCL file by defining the width and length of the topology [28].

In our E-RADP.tcl simulation, a random movement of the nodes is created with the help of a for-loop. Parts of the scripts in E-RADP.tcl are:

```
            # Define node initial position in nam
        for {set i 0} {$i < $val(nn) } { incr i } {
            set xx [expr rand()*$val(x)]
            set yy [expr rand()*$val(y)]
            $node_($i) set X_ $xx
            $node_($i) set Y_ $yy  }
    # dynamic destination setting procedure.
$ns at 0.0 "destination" proc destination {} {
                global ns val node_
                set time 1.0
                set now [$ns now]
            for {set i 0} {$i<$val(nn)} {incr i } {
                set xx [expr rand()*$val(x)]
                set yy [expr rand()*$val(y)]
            $ns at $now "$node_($i) setdest $xx $yy
                    20.0"}
            $ns at [expr ($val(stop) +$time)] "destination" }
```

A trace file is a text-based result, which records the actions and relevant information of the simulation. It is created by the "$ns use new trace" command [28], [29]. Part of the script to create a trace file in our E-RADP.tcl file is:

```
  #Creating nam and trace file:
 set tracefd       [open E-RADP.tr w]
 set namtrace      [open E-RADP.nam w]
 $ns trace-all $tracefd
 $ns use-newtrace
 $ns namtrace-all-wireless $namtrace $val(x) $val(y)
```

D. SIMULATION OF AODV PROTOCOL IN NS 2.35

NS2 implements the AODV routing protocol as an agent. Routing agents are used to create and receive control packets. As described in [29], NS2 declares a class AODV, and class AODV derives from the class agent. AODV files are stored in the directory of ~ns/aodv/. They include C++ and header files [30]. The main definitions of AODV routing protocols are found in the aodv.h and aodv.cc files. Whereas the aodv_packet.h file defines the AODV packet headers. The routing entries and routing tables of the AODV protocol are controlled in the aodv-rtable.cc and aodv_rtable.h files. During a route discovery procedure, a buffer is located in the aodv_rqueue.h and aodv_rqueue.cc files store data packets.

AODV uses RREQ and RREP messages for route discovery and RERR messages for route maintenance. To broadcast RREQ packets, AODV uses the sendRequest(nsaddr, t, dst) function and To unicast RREP packets it uses the sendReply(nsaddr, t, ipdst..) function. Whereas, the recvRequest(Packet *p) function is useful for receiving RREQ, the recvReply(Packet *p) function is responsible for receiving RREP.

E. E-RADP IMPLEMENTATION

First, we have modified the AODV protocol to our new E-RADP protocol. Second, we have created the E-RADP.tcl file using NSG2, and we have modified it. Then, E-RADP.tcl is executed using a network simulator (ns) command to generate an animator E-RADP.nam and a trace file E-RADP.tr.

Our protocol addresses the modification of different files and modules of the NS2 framework. All files are found in the directory of ~nsallinone-2.35/ns-2.35. Files we have modified include; aodv/aodv.h, aodv/aodv.cc, aodv/aodv_rtable.h, aodv/aodv_rtable.cc, aodv/aodv_packet.h, aodv/aodv_packet.cc, aodv/mobilenode.h/cc, mac/WirelessPhy.h, mac/WirelessPhy.cc, common/agent.h, common/agent.cc, and link/ hackloss.h.

To run the network performance analysis, first, we have created awk scripts for each performance metric. Second, the awk command is executed on each awk script for each protocol trace file to generate xgrah data for each protocol selected for performance analysis. Then, the xgraph command is executed on the xgraph data for each metric.

a) Modification in AODV_PACKET.H File

As shown in Table II and Table III, we have modified struct hdr_aodv_request, and struct hdr_aodv_reply methods to add new header fields in RREQ and RREP packets. Those newly added fields are used to capture Ratio(R) values and Previous Time Stamp (PTS) values.

Table 2. Modified RREQ message format

Type	R	A	Reserved	Prefix Sz	Hop Count
RREQ ID					
Destination IP Address					
Destination Sequence Number					
Originator IP Address					
Originator Sequence Number					
Ratio(R)					
Previous node Time Stamp (PTS)					

Table 3. Modified RREP message format

Type	R	A	Reserved	Prefix Sz	Hop Count
Destination IP Address					
Destination Sequence Number					
Originator IP address					
Lifetime					
Ratio(R)					

b) Modification in AODV_RTABLE.H/CC File

In the aodv-rtable.h/cc files, we have modified the route entry header format based on RFC 3561 [31]. As shown in Table IV, we added a new route entry information field called Ratio(R) inside the aodv route entry class "aodv_rt_entry".

Table IV: Modified Routing table format

Routing Table
Destination IP Address
Next Hop
Hop Count
Destination Sequence Number
Active Route Lifetime
Network Interface
Ratio(R)

c) Modification of AODV.H/CC Files

As shown in Algorithm 1, our E-RADP protocol modifies different methods in aodv.h/cc file such as rt_update, recvRequest, recvReply, sendRequest, and sendreply to select a reliable path by considering the ratio of RREQ time.

The following segment of code shows modification on a standard AODV protocol for the intermediate node route request (RREQ) process.

```
……. remaining codes ----------
IDelay =PAT – rq-> PTS
If (Idelay < TH) {
        Rnode=1; //RNode is detected and the RREQ packet
                  from Rnode is dropped
    }
Else {
      IR = IDelay) / rq->HC
   If (IR < rq->R) {
        Rnode=1; //RNode is detected and the RREQ
        packet from Rnode is dropped
      }
   Else {
      R=(R+IR)/2;
      PTS=PAT;
      rq-> PTS= PTS;
      rq-> R= R;
     }}
……. remaining codes ----------
```

The following segment of code shows modification on a standard AODV protocol for the destination node route request (RREQ) process.

```
……. remaining codes ----------
IDelay =PAT – rq-> PTS
If (Idelay < TH) {
        Rnode=1;  // RNode is detected and the RREQ
                  packet from Rnode is dropped
    }
Else {
      IR = (PAT - rq->PTS) / rq->HC
      If (IR < rq ->R) {
              Rnode=1; // RNode is detected and the
              RREQ packet from Rnode is dropped
              }
      Else    {
              R=(R+IR)/2;
              rt-> R=  R;
     }}
……. remaining codes ----------
```

The following segment of code shows modification on a standard AODV protocol for the destination node during the route replay (RREP) process.

```
……. remaining codes ----------
rp-> R= rt-> R.;
……. remaining codes ----------
```

The following segment of code shows modification on a standard AODV protocol for the intermediate node during the route replay (RREP) process.

```
……. remaining codes ----------
rt-> R= rp-> R;
…….…… remaining codes ----------
```

In our experiment, part of the E-RADP.tcl script on AODV with rushing node detection method is:

```
## Creating node objects and Setting color for node objects.
for {set i 0} {$i < $val(nn) } { incr i } {
      set node_($i) [$ns node]
      $node_($i) color blue
      $ns at 0.0 "$node_($i) color blue"
   }
#rushing attackers
set c 1
set n [expr round(rand()*$val(nn))]
for {set i $n} {$i < $val(nn) } {incr i 5 } {
      $ns at $i "$node_($i) color red"
    $ns at $i  "[$node_($i) set ragent_] rushing attack"
      $ns at $i "$node_($i) label RNode_$c"
      $ns at $i.1 "$ns trace-annotate \" Node_$i is
      Rushing node $c which drops the packet \""
        incr c 1
   }
```

F. SUMMARY

The implementation of the proposed algorithm is based on modifying the AODV protocol. The ratio of RREQ time (R) and hop count value are major metrics for route selection in our modified protocol. To implement our protocol we have created TCL and AWK scripts, and xgraph and trace files. We have also modified wireless component parameters and AODV files found in the NS2 framework. Then, we executed the proposed E-RADP protocol using the E-RADP.tcl script file to generate the E-RADP.tr trace file. Finally, we have executed the awk scripts we have created on the E-RADP.tr file to generate xgraph data.

V. DISCUSSION

Generally, all existing works use inadequate performance evaluation metrics like end-to-end delay, packet delivery ratio, packet loss rate, and packet throughput. These metrics will measure the effects of a rushing attack but cannot measure rushing node detection and prevention performance.

Although different security performance metrics can be used to evaluate the performance of routing protocols, to evaluate our proposed protocol with the existing ordinary AODV, MAODV, and PMRA routing protocols, we have used the performance metrics; attack detection rate, true positive, false positive, false negative, throughput, packet delivery ratio, and end to end delay. Because these metrics are more affected by rushing attacks. In this study, the metrics are calculated using different AWK scripts which are used to process trace files and produce xgraph data for analysis. The awk command to generate xgraph data is "Awk –f filename.awk tracefile.tr >filename.xgr", whereas the xgraph command to compare four protocols is "xgraph E-RADP.xgr MAODV.xgr PMRA.xgr rushedAODV.xgr".

A. ATTACK DETECTION RATE

In this research work, the detection rate, which is also called as true negative rate can be mathematically expressed as:

$$\text{DR} = \frac{\text{number of truly detected rushing nodes}}{\text{number of input rushing nodes}} \text{ X } 100. \qquad (1)$$

From Figure 6, we have analyzed that when the number of rushing nodes increases, the DR value of all protocols slightly decreases. We have also observed that the DR of all protocols is greater than 80%, while our protocol detection performance is better than existing MAODV and PMRA protocols. This is because, in addition to the threshold value used in existing protocols, our protocol uses security parameters intermediate delay, and the ratio of RREQ time to better detect and prevent rushing nodes from joining the MANET.

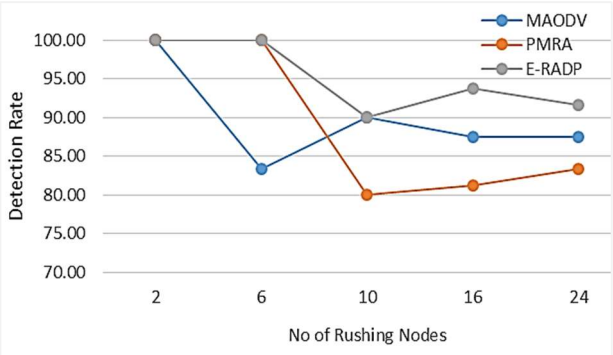

Figure 6. Attack Detection Rate

B. TRUE POSITIVE RATE

In this research work, the true positive rate is mathematically expressed as:

$$\text{TPR} = \frac{\text{number of positively detected normal nodes}}{\text{number of input normal nodes}} \qquad (2)$$

From Figure 7, we have analyzed that when the number of normal nodes increases, the TPR value of our protocol slightly decreases. We have also observed that the TPR of our protocol is greater than 0.978(97.8%), nearest to the detection rate value 1 (100%), and better than the existing MAODV and PMRA protocols. This is because, in addition to a constant threshold value used by existing protocols, our protocol uses a security parameter called ratio. This ratio parameter learns normal nodes from the network and updates its value based on the number of nodes (hop count), and RREQ packet arrival time to detect and prevent rushing nodes early during the route discovery stage.

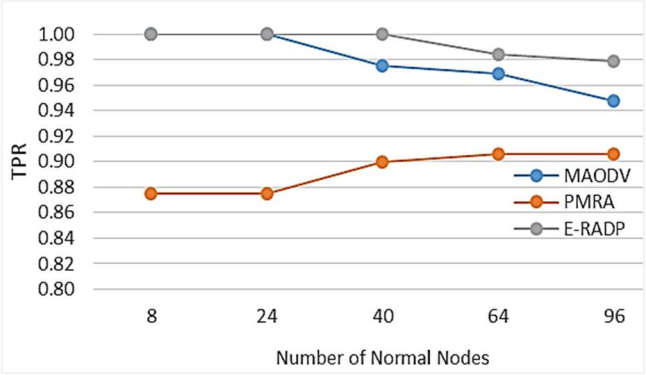

Figure 7. True Positive Rate

C. FALSE POSITIVE RATE

In this research work, the false positive rate is mathematically expressed as:

$$\text{FPR} = \frac{\text{No of rushing nodes detected as normal nodes}}{\text{No of detected normal nodes}}. \qquad (3)$$

where, the number of rushing nodes detected as normal nodes = number of detected normal nodes minus the number of actual normal nodes in a network.

From Figure 8, we have analyzed that the FPR value of our protocol is less than 0.25, the nearest to 0, MAODV is between 0.25 and 0.45, and PMRA is above 0.45, which indicates that our protocol outperforms MAODV and PMRA protocols. This is due to our protocol uses a larger initial threshold value of "5" than PMRA with threshold value of "0.02". In addition to this initial constant threshold value, our protocol computes a ratio parameter as a threshold value which continuously learns normal nodes behavior from the network and updates its value based on HC, and RREQ PAT early during the route discovery stage. Therefore, our protocol minimizes rushing nodes malfunctioning to being as normal nodes from joining a network.

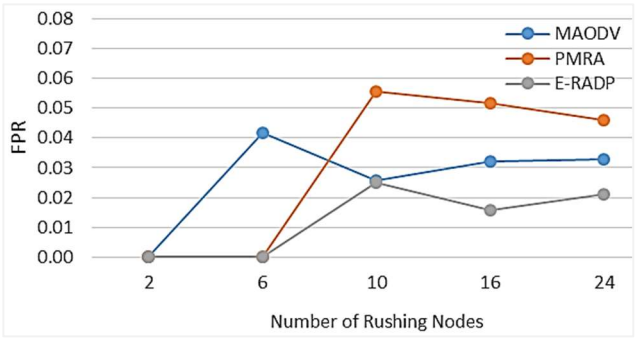

Figure 8. False Positive Rate

D. FALSE NEGATIVE RATE

In this research work, the false negative rate is mathematically expressed as:

$$\text{FNR} = \frac{\text{No of normal nodes detected as rushing nodes}}{\text{number of input normal nodes}}. \quad (4)$$

where, the number of normal nodes detected as rushing nodes= number of detected rushing nodes minus the number of actual rushing nodes.

From Figure 9, we have analyzed that the FNR value of our protocol is less than MAODV while greater than PMRA, which indicates that our protocol's FNR performance is better than MAODV while worse than PMRA protocol. This is due to our protocol mainly focusing on security using a larger threshold value than PMRA to protect false positive issues. Therefore, in our protocol, any normal nodes malfunctioning as rushing nodes are considered as rushing nodes and are isolated from a network.

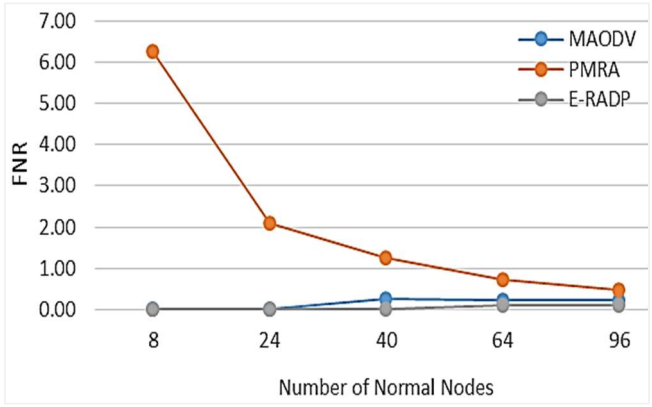

Figure 9. False Negative Rate

E. THROUGHPUT

In this research work, throughput is the number of data packets sent successfully per unit time. It is expressed in bits per second (bps). The higher the throughput is, the better the protocol security efficiency, and the smaller the packet loss rate. It is mathematically expressed as:

$$\text{Throughput} = \frac{\text{number of bytes received} \times 8}{\text{Simulation Time}} \times 100 \quad (5)$$

From Figure 10, we have analyzed that when the simulation time increases, without change in the size of the network and the behavior of nodes, the throughput value of our protocol also increases; in this case, we have used 16 rushing and 64 normal nodes. From this investigation, we found that the throughput of our protocol highly outperforms other existing protocols. This is because our protocol detects and prevents rushing nodes early during the route discovery stage.

Therefore, in our protocol, there will be no rushing nodes in any newly discovered route. This relieves our protocol from frequent route discovery, update and maintenance, and dropping and retransmitting data packets during the data transmission stage. Existing protocols require route discovery, route update, and retransmitting of data packets during the data transmission stage if a rushed node is detected in the path. These processes require additional time to deliver data packets to destination nodes.

Figure 10. Throughput Based on Simulation Time

From Figure 11, we have analyzed that when the number of both normal and rushing nodes increases instantaneously in the network, the value of the throughput of all protocols decreases. We have also observed that the throughput of our protocol is better than state-of-the-art protocols. From this investigation, we can conclude that our algorithm is preferable for throughput when there are rushing nodes in a network. This is because our protocol uses Ratio, and IDelay early during the route discovery process to prevent rushing nodes from joining the network. This minimizes data loss, and time for retransmission of data.

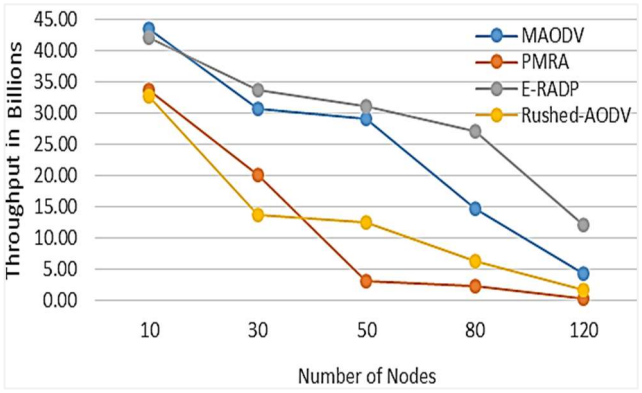

Figure 11. Throughput Based on No of Normal and RNodes

F. PACKET DELIVERY RATIO

In this research work, the packet delivery ratio is mathematically expressed as:

$$\text{PDR} = \frac{\text{sum of data packets received by destination}}{\text{sum of data packets generated by sources}} \times 100 \quad (6)$$

From Figure 12, we have analyzed that when the simulation time increases, without a change in the size of the network and the behavior of nodes, the packet delivery ratio of our protocol highly increases. In this case, we have used 16 rushing and 64 normal nodes the packet delivery ratio of our protocol is far better than other existing protocols. Figure 12 shows that the packet delivery ratio of our protocol highly outperforms others. This is because our protocol prevents rushing nodes from joining the network using Ratio, and IDelay parameters early during the route discovery process. This minimizes data loss.

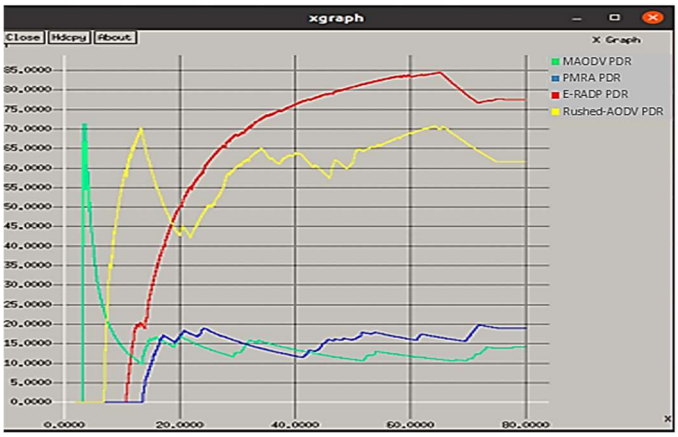

Figure 12. PDR Based on Simulation Time

From Figure 13, we have analyzed that when the number of both normal and rushing nodes increases instantaneously in the network, the packet delivery ratio of all protocols decreases. From this investigation, we have also observed that the PDR of our protocol is better than state-of-the-art protocols as the number of nodes increases with the presence of rushing nodes.

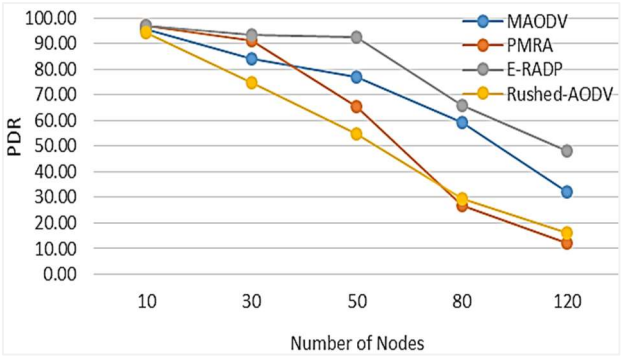

Figure 13. PDR Based on Normal and Rushing Nodes

G. PACKET LOSS RATE

In this research work, the packet loss rate is mathematically expressed as:

$$\text{PLR} = \frac{\text{sum of dropped data packets}}{\text{Simulation Time}} \times 100 \quad (7)$$

From Figure 14, we have analyzed that when the simulation time increases, with a constant network size and node behavior, the PLR of our protocol also increases; in this case, we have used 16 rushing and 64 normal nodes. Figure 14 also shows that the packet loss rate of our protocol highly outperforms state-of-the-art protocols. This is because our protocol detects and prevents rushing nodes early during the route discovery stage. Therefore, there will be no rushing nodes in any route.

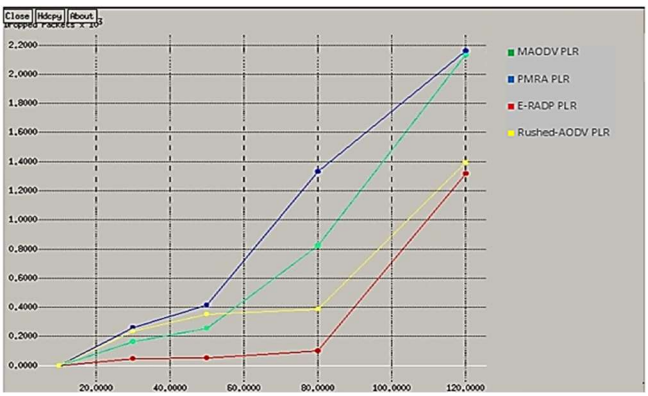

Figure 14. PLR Based on Simulation Time

From Figure 15, we have analyzed that when the number of both normal and rushing nodes increases instantaneously in the network, the PLR value of all protocols increases. From this investigation, we have also observed that the PLR performance of our protocol is essentially better than state-of-the-art protocols.

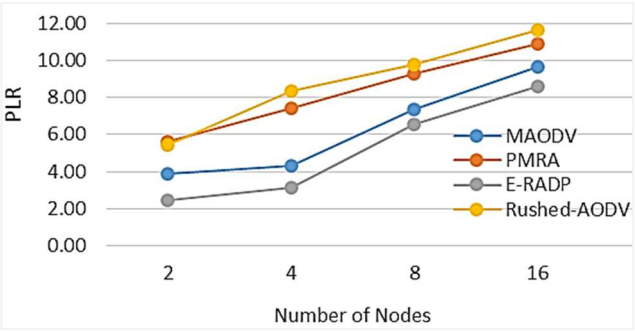

Figure 15. PLR Based on No of Normal and Rushing Nodes

H. END-TO-END DELAY

In this research work, end-to-end delay (E2ED) is mathematically expressed as:

$$\text{Delay}[i] = (\text{receiving time}[i] - \text{sending time}[i]) \quad (8)$$

$$\text{Total Delay} = \text{Total Delay} + \text{Delay}[i] \quad (9)$$

$$\text{Average Delay} = \frac{\text{Total Delay}}{\text{count}} \quad (10)$$

where i = packet sequence number and count = Total packet count.

From Figure 16, we have analyzed that the simulation time and the number of nodes have less effect on the end-to-end delay performance of our protocol. From this figure, observed that the end-to-end delay performance of our protocol is better than state-of-the-art protocols. Our protocol's average end-to-end delay is less than 0.3 seconds (300 ms), therefore its performance fulfills the required end-to-end delay standard.

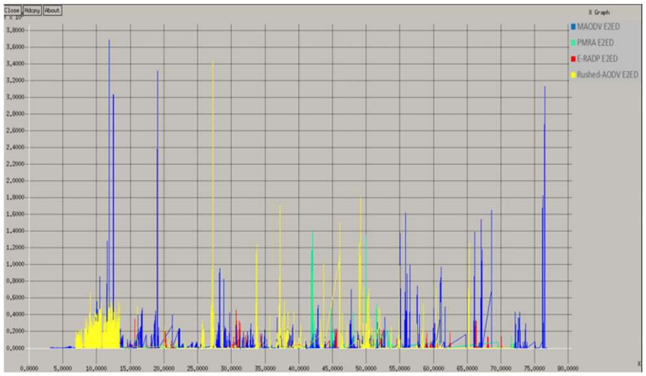

Figure 16. E2ED Based on Simulation Time

From Figure 17, we have analyzed that when the number of both normal and rushing nodes increases instantaneously in the network, the E2ED value of all protocols also increases. Figure 17 also shows that the E2ED performance of our protocol is essentially better than state-of-the-art protocols.

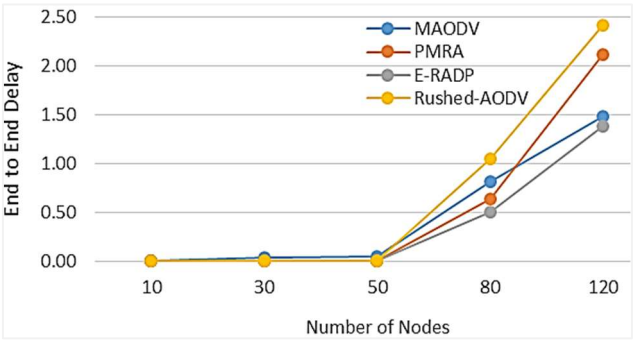

Figure 17. E2ED Based on No of Normal and Rushing Nodes

VI. CONCLUSIONS

This paper proposes the E-RADP protocol, a novel approach for early detection and prevention of rushing attacks in AODV-based MANETs, introducing RREQ packet arrival time (PAT), intermediate delay (IDelay), and ratio-based metrics (IR, R) alongside previous timestamp (PTS) and threshold (TH) analyses. Unlike existing systems reliant on static thresholds (TH) or hop count (HC) methods which are effective during data packet transmission, E-RADP addresses vulnerabilities during route discovery by integrating dynamic temporal and ratio evaluations, enabling faster, more secure node authentication. These evaluation methods prevent, detect, and isolate rushing nodes from joining the network early at the route discovery stage.

Results demonstrate our protocol's superior performance in detection rate, throughput, packet delivery ratio, and end-to-end delay reduction, with scalability across network sizes. These attributes make E-RADP ideal for security-critical MANET applications, including VANETs, military networks, UAVs, and emergency operations.

Future work will focus on optimizing false negative rates, incorporating game theory and smart contract vulnerability detection, and validating the protocol in real-world deployments such as IoV, FANET, and smart home systems.

DATA AVAILABILITY

Simulation data used in this work is submitted as part of the paper.

CODE AVAILABILITY

Complete step-by-step NS2 code to implement the protocols, complete tcl files, trace files, nam files, and awk scripts to simulate the protocols, and our complete E-RADP protocol that modifies standard AODV protocol can be requested from the corresponding author.

FUNDING STATEMENT

This research work is not funded by any organization.

CONFLICT OF INTEREST

We declare that we have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] G. A. Zimbele, T. M. Asres, and T. M. Kifle, "Early rushing attack detection and prevention in AODV MANETs," *Preprint*, 8 Oct. 2024. https://doi.org/10.21203/rs.3.rs-5219038/v1.

[2] A. Malik, M. Z. Khan, and S. M. Qaisar, "An efficient approach for the detection and prevention of Gray-Hole attacks in VANETs," *IEEE Access*, vol. 11, pp. 46691–46706, 2023. https://doi.org/10.1109/ACCESS.2023.3274650.

[3] S. Vijayalakshmi, S. Bose, G. Logeswari, and T. Anitha, "Hybrid defense mechanism against malicious packet dropping attack for MANET using game theory," *Cyber Security and Applications*, vol. 1, p. 100011, 2023. https://doi.org/10.1016/j.csa.2022.100011.

[4] S. Khan, M. Z. Khan, P. Khan, G. Mehmood, A. Khan, and M. Fayaz, "An ant-hocnet routing protocol based on optimized fuzzy logic for swarm of UAVs in FANET," *Wireless Communications and Mobile Computing*, vol. 2022, 2022. https://doi.org/10.1155/2022/6783777.

[5] R. A. Jothi, A. L. Jeeva, and V. Palanisamy, "Various attacks and countermeasures in mobile ad hoc networks: A survey," *International Journal of Engineering Research & Technology (IJERT)*, vol. 3, no. 33, pp. 1–8, 2015.

[6] R. F. Olanrewaju, B. U. I. Khan, F. Anwar, and B. R. Pampori, "MANET security appraisal: Challenges, essentials, attacks, countermeasures & future directions," *International Journal of Recent Technology and Engineering (IJRTE)*, vol. 8, no. 6, pp. 3013–3024, 2020. https://doi.org/10.35940/ijrte.E6537.038620.

[7] S. Sivanesh, and V. R. S. Dhulipala, "Comparative analysis of blackhole and rushing attack in MANET," *Proceedings of the 2019 TEQIP III Sponsored International Conference on Microwave Integrated Circuits, Photonics and Wireless Networks (IMICPW)*, 2019, pp. 495–499. https://doi.org/10.1109/IMICPW.2019.8933192.

[8] M. Goyal, S. K. Poonia, and D. Goyal, "Attacks finding and prevention techniques in MANET: A survey," *Advances in Wireless and Mobile Communications*, vol. 10, no. 5, pp. 1185–1195, 2017.

[9] M. Sharma, and M. Rashid, "Security attacks in MANET – A comprehensive study," *Proceedings of the International Conference on Intelligent Communication and Computational Research (ICICCR-2020)*, 2020, pp. 1–6. https://doi.org/10.2139/ssrn.3565860.

[10] A. Aranganathan, and C. D. Suriyakala, "Agent based secure intrusion detection and prevention for rushing attacks in clustering MANETs," *International Journal of Engineering & Technology*, vol. 7, pp. 22–25, 2018. https://doi.org/10.14419/ijet.v7i2.20.11736.

[11] R. Thilagarasi, and D. Geetha, "Prevention of multiple rushing attack nodes in multicast MANET," *International Journal of Computer Trends and Technology (IJCTT)*, vol. 29, no. 2, pp. 64–68, 2015. https://doi.org/10.14445/22312803/IJCTT-V29P112.

[12] E. K. Narang, and Sonal, "A study of different attacks in MANET and discussion about solutions of black hole attack on AODV protocol," *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, vol. 2, no. 4, 2013.

[13] M. Ahmed, A. Abdullah and A. El-Sayed, "A survey of MANET survivability routing techniques," *International Journal of Communications, Network and System Sciences*, vol. 6, no. 4, pp. 176-185, 2013. doi: 10.4236/ijcns.2013.64021. https://doi.org/10.4236/ijcns.2013.64021.

[14] R. Thilagarasi, and D. Geetha, "Review on rushing attack and its prevention techniques in MANET," *Proceedings of the UGC Sponsored National Conference on Advanced Networking and Applications, Int. Jnl. Of Advanced Networking and Applications*, March 2015.
[15] S. Ghoreishi, S. A. Razak, I. F. Isnin, and H. Chizari, "Rushing attack against routing protocols in Mobile Ad-Hoc Networks," *Proceedings of the 2014 IEEE International Symposium on Biometrics and Security Technologies (ISBAST)*, 2014, IEEE, pp. 220–224. https://doi.org/10.1109/ISBAST.2014.7013125.
[16] D. Anitha, and B. V. Priya, "Survey on detecting rushing attack by using routing protocol," *I. J. of Research in Computer Applications and Robotics*, vol. 3, no. 9, pp. 70–73, 2015.
[17] S. S. Narayanan, and G. Murugaboopathi, "Prevention of rushing attack in MANET using threshold-based approach," *Int. J. Internet Technology and Secured Transactions*, vol. 10, no. 5, pp. 576–584, 2020. https://doi.org/10.1504/IJITST.2020.109536.
[18] C. Suthar, and B. Panchal, "Rushing attack prevention with modified AODV in Mobile Ad hoc Network," *International Journal of Engineering Development and Research*, vol. 2, no. 4, pp. 3489–3493, 2014.
[19] D. Shankari, S. Sudhalakshmi, and V.P. Saranya, "Random route selection rushing route algorithm for MANET," IJARIIE, vol. 3, no. 4, pp. 423–430, 2017.
[20] W. Junaid, and A. Iqbal, "Prevention of multiple rushing attacks in Mobile Ad Hoc Network using AODV routing protocol," *Sci.Int. (Lahore)*, vol. 30, no. 1, pp. 173–177, 2018.
[21] S. Shrivastava, and D. Mangal, "A new technique to prevent MANET against rushing attack," *International Journal of Computer Science and Information Technologies*, vol. 5, no. 3, pp. 3460–3464, 2014.
[22] S. Joshi, and D.K. Mishra, "Detection of rushing attack and data modification attack in Mobile Ad Hoc Networks," *Journal of Critical Reviews*, vol. 7, no. 19, pp. 9486–9498, 2020.
[23] S. Shrivastava, and D. Mangal, "A new technique to prevent MANET against rushing attack," *International Journal of Computer Science and Information Technologies*, vol. 5, no. 3, pp. 3460–3464, 2014.
[24] K. Gangadhara Rao, Ch. Suresh Babu, B. Basaveswara Rao, and D. Venkatesulu, "Simulation based performance evaluation of various routing protocols in MANETs", Journal of Mobile Computing & Application, vol. 3, no. 4, pp. 23-39, 2016. https://doi.org/10.9790/0050-03042339.
[25] N. Nissar, N. Naja and A. Jamali, "A neighbor signal strength based coverage for reducing routing overhead in Mobile Ad hoc Networks," *Proceedings of the 2016 5th International Conference on Multimedia Computing and Systems (ICMCS)*, Marrakech, Morocco, 2016, pp. 321-326. https://doi.org/10.1109/ICMCS.2016.7905527.
[26] A.A. Salem, and H. Awwad, "Mobile ad-hoc network simulators, a survey and comparisons," *International Journal of P2P Network Trends and Technology*, vol. 4, no. 3, 2014.
[27] M. H. Kabir, S. Islam, Md. J. Hossain, and S. Hossai, "Detail comparison of network simulators," *International Journal of Scientific & Engineering Research*, vol. 5, no. 10, 2014.
[28] K. Fall, and K. Varadhan, "The ns Manual," The VINT Project - A collaboration between researchers at UC Berkeley, LBL, USC/ISI, and Xerox PARC, February 2011. [Online]. Available at https://www.isi.edu/nsnam/ns/doc/ns_doc.pdf.
[29] T. Issariyakul, and E. Hossain, "An introduction to network simulator NS2," Springer, Berlin, 2012. https://doi.org/10.1007/978-1-4614-1406-3.
[30] M. H. Rehmani, S. Doria, and M. R. Senouci "Tutorial on the Implementation of Ad-hoc OnDemand Distance Vector (AODV) Protocol in Network Simulator (NS-2)", NS-2, June 2009.
[31] C. Perkins, E. Belding-Royer, and S. Das, "Ad hoc on-demand distance vector (AODV) routing," *No. RFC* 3561, pp. 1–37, 2003. https://doi.org/10.17487/rfc3561.

TEAMERAT MEKONNEN ASRES, *received the B.S. and M.S. degrees in computer science from Haramaya University in 2016 and 2022, respectively. Since 2016, he has been working as a lecturer with the Computer Science Department at Haramaya University. His research interests are in the area of network security and natural language processing (NLP).*

GETANEH AWULACHEW ZIMBELE, *graduated from Adama Science and Technology University in 2013 with a B.S. in information technology and from Debre Berhan University in 2019 with an M.S. in computer networks and security. Since 2017, Debre Berhan University's Information Technology Department has employed him as a lecturer. He is currently researching AI security after publishing research papers in the field of network security. Since 2024, he has currently been a PhD candidate at Tianjin University. His current areas of focus in study are resource-efficient algorithms and AI security, specifically LLM edit security.*

TEWODROS MEKURIA KIFLE, *graduated from Debre Berhan University with a B.S. in computer science in 2016 and an M.S. in computer networks and security in 2020. From 2016 up to 2024, he served as the chief technical assistant in the physics department at Dire Dawa University. Since 2025, Akufada Akufada Microfinance has employed him in the digital and agent banking division. His areas of interest in research are sensor networks, energy-efficient MANET routing, and MANET security.*