A Secure Core-Assisted Multicast Routing Protocol in Mobile Ad-Hoc Network

Faheem Khan[1], Abdul Wahid Khan[2], Samiullah Khan[3], Iqbal Qasim[2], Asad Habib[4]

[1] Department of Computer Science, The University of Lakki Marwat, Pakistan
[2] University of Science & Technology Bannu, Pakistan
[3] Agriculture University Peshawar, Pakistan
[4] Kohat University of Science & Technology, Pakistan
kfaheem81@gmail.com, wahidkn@gmail.com, samikhan@aup.edu.pk, r.rahatullah@gmail.com, asadhabib@kust.edu.pk

Abstract

In this paper, an Efficient and Reliable Core-Assisted Multicast Routing Protocol (ERASCA) is secured from a malicious/selfish receiver attack and fabrication attack. In former the ERASCA is secured from malicious/selfish attack through battery estimation technique. With battery estimation technique, a malicious or selfish receiver shows high or low battery capacity for the purpose to become a core or evade to become a core node. The malicious or selfish receiver is detected and removed from the mesh by comparing estimated value and claimed value. Similarly, malicious nodes may alter data or inject spoofed messages in the network. In addition, a packet authentication process is also used to prevent nodes from tampering with data and generating spoofed messages. At the end of paper, Network Simulator-2 is used to observe the performance of protocol and evaluate the conclusion based on results.

Keywords: Multicasting, MANET, Battery estimation technique, Malicious/Selfish receiver, Packet authentication process

1 Introduction

MANET can be deployed in scenarios where there is no infrastructure available like jungles, mountains, deserts, in earthquake scenario, special operation with a specific number of groups in battlefield, rescue operations, etc. [1-2]. In MANET, nodes must cooperate among each other to establish connectivity and routing in the network. Because of the cooperative environment, MANET is vulnerable to various attacks and the attackers can effortlessly disturb the correct network functioning through malicious activities [3-6].

To improve the efficiency of MANET and to discourage malicious/selfish receiver within the mesh, a Battery Estimation Technique (BET) is proposed. In this technique, a malicious receiver that shows high battery capacity instead of original battery capacity for the purpose to become a core or a selfish receiver by showing minimum battery capacity to evade as a core node are detected and discarded from the group. In such situations, claimed value and estimated values are compared with each other and if there is considerable deviation in the estimated value and the claimed value, then such a receiver is considered as a malicious/ selfish within the mesh.

In An Efficient and Reliable Core-Assisted Multicast Routing Protocol (ERASCA) [7-8], an election is conducted within the receiver group to elect a core. To know about the exact battery capacity of each receiver in the group, overhearing is used [9-11]. In overhearing, a receiver listens to the packet inside its broadcast range that is intended for other receivers in a group as shown in Figure 1, where all the neighbors overhear receiver A transmission.

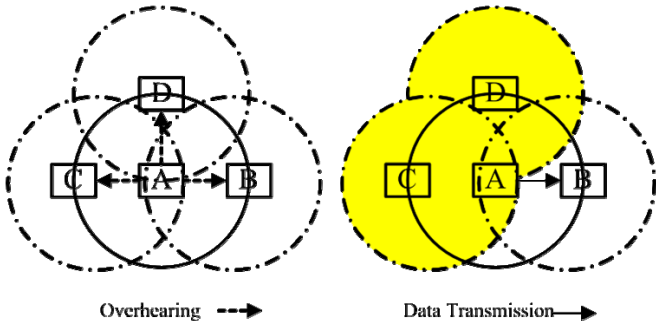

Figure 1. Overhearing in MANET [11]

A malicious, selfish and Fabricating attack can disrupt the core election process in ERASCA by fabricating Core Election Message (CEM) and disseminate false information in the group for malicious purposes.

The paper is organized as follow: Section 2 describes the literature review of malicious, selfish and

*Corresponding Author: Abdul Wahid Khan; E-mail: wahidkn@gmail.com
DOI: 10.3966/160792642020032102006

fabrication attack with related protocols. Section 3 describes the framework of BET in detail, in which a malicious and selfish attack is detected and isolated within the group. In Section 4 and 5 describes the detection of packet fabrication problem, in which an attacker as a malicious entity try to insert spoofed CEM to disturb the core election process. In Section 6 and 7, simulation parameters with results and discussion is presented in which internal attacks are detected efficiently with the improvement in the performance of ERASCA protocol. Finally, paper is concluded in Section 8.

2 Literature Review

Several multicast routing protocols like MAODV, ODMRP, PUMA, ERASCA and ERASCA-MC are still vulnerable to security threats. The analysis conducted previously on these protocols exposed flaws that are otherwise difficult to be detected [12-16].

Furthermore, the above-mentioned routing protocols concentrating on attacks like packet dropping, flood rushing and wormhole attacks. Detection based schemes are usually used against these attacks which depend on the transmission behavior of mesh routers to detect and remove selfish and malicious nodes during core election.

SODMRP is a multicast protocol based on a link layer metric [17]. SODMRP reports packet dropping attacks by detecting inconsistency between perceived packet delivery ratio (pPDR) and expected packet delivery ratio (ePDR). In SODMRP, if ePDR- pPDR for a path is greater than detection threshold, an attack is detected. This attack will affect the consistent data flow with its claimed quality.

Other multicast protocols like Hierarchical agent based secure multicast (HASM) [18] and Mesh certification authority (MeCA) [19] emphasis on secure group communication technique, key generation and key management to secure against external attacks. HASM is used for secure multicasting in mobile wireless network and dynamically organized a multicast group on mesh routers for multicast service management and integrated mobility. It decreases the whole cost of network communication produced by security key management, multicast packet delivery and mobility management.

In Authenticated Routing for Ad-hoc Network (ARAN) the malicious or selfish attacks are reduced due to authentication however still gaps of security are left due to core election [19].

All the existing mentioned schemes do not address the security threats of malicious or selfish attacks on multicast routing protocols in MANET. The present framework of unicast routing protocols cannot be implemented for multicast environment due to the vulnerabilities and core election process.

3 Battery Estimation Technique

In this section, a detection procedure i.e., BET is proposed to secure ERASCA protocol from the malicious and selfish attacks and presents a framework to detect and isolate these attackers within the group.

In ERASCA, when the core fails, an election is conducted to elect a resourceful core within the group. This core election depends on battery capacity and location (i.e. dense part of the network or maximum connectivity) of the receiver. To know the battery capacity and connected neighbors of each receiver, a Core Election Message (CEM) is flooded by receiver *n* to elect the best receiver in a group. In reply, all receivers flood a CEM within the group in which each receiver includes its BC and connected neighbors. Thus, all the receivers know the estimated battery capacity and connected neighbors of each other. After exchanging information through CEM, a core is elected in a group. In this paper, only battery capacity is considered for evaluation purposes. For malicious purposes, the malicious/selfish receiver sends inaccurate information related to battery capacity in CEM within the receiver group.

3.1 The Detection Rationale

Consider that the path between the source and destination is already established and the source broadcast CEM within its radio range. The in/out traffic is represented by T_{in} and T_{out}. The wrong claims about battery capacity are made at t_1 and t_2 and are represented as BC_{t_1} and BC_{t2}. From such wrong claims, a battery drainage rate is calculated which is represented as B_{dr}. B_{dr} is considered as a sum of claim value at t_1 and t_2. To verify the legitimacy of the claims, battery drainage at time t_3 is estimated as Bdr_{t_3} and is known as estimated value. The details of the process is as under.

Example: In Figure 2, receiver 2 is a source and receiver 8 is a destination, also receiver 5 is considered as a selfish/malicious receiver. In Figure 2, receiver 5 is a malicious receiver and it shows a wrong data about its battery capacity. Suppose at time t_1 and t_2 battery capacity claims by 5 is BC_{t_1} and BC_{t2}. The difference between two claimed values of receiver 5 is calculated through equation 1 by battery drainage i.e., the drainage rate between two claims at time t_1 and t_2.

$$B_{dr} = BC_{t_1} - BC_{t_2} \quad \textbf{(1)}$$

To verify whether the claim of 5 is legitimate or not, the neighbours find the battery drainage through the input and output of receiver 5 as shown in equation 2 by battery drainage at t_3, Bdr_{t_3}. When receiver 5 sends data packet of source 2 to destination 8, such information

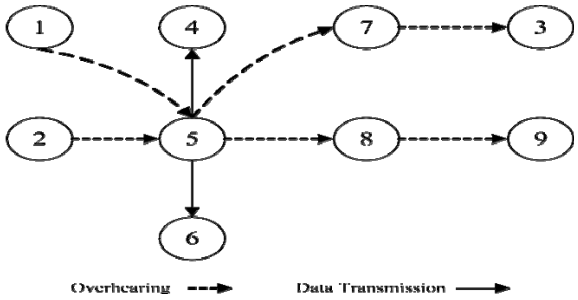

Figure 2. Overhearing in receiver group

is also received by others receivers because of the broadcast nature of the MANET. As a result, all the neighbours know about the T_{in} and T_{out} packet by receiver 5. On the basis of packet transmission and reception, neighbours can easily calculate the battery capacity of receiver 5. Bdr_{t_3} is the estimated battery drainage at t_3 and through Bdr_{t_3} the claim could be justified or rejected.

$$Bdr_{t3} = T_{in} + T_{out}/BC\ total \times 100 \quad \textbf{(2)}$$

Now if the value of battery drainage at t_3 is larger than the value of equation 1, such a receiver is considered as a malicious. It means that a receiver 5 claims a low battery drainage with a high battery to become as a core but in actual situation (equation 2) the battery capacity is low than it claims as shown in equation 2. It should be noted that drainage rate and battery capacity are inversely proportional to each other. Therefore, the estimated battery capacity is smaller than the claim battery capacity and receiver 5 is considered as a malicious attack.

Therefore, if

$$Bdr_{t_3} > B_{dr} \pm 10\% \quad \textbf{(3)}$$

Hence, malicious 5 is detected from the above equations and the core broadcasts the malicious identity in a group not to entertain its request and its candidature for core. A similar method is used if the selfish receiver claims low rating/weight about its BC and avoids himself as a candidate for core. According to equation 4, a selfish behaviour is found.

$$Bdr_{t3} < Bdr \pm 10\ \% \quad \textbf{(4)}$$

Equation 4 shows that the battery consumption is less but the selfish receiver shows high battery consumption, which makes it not a cooperative receiver and hence discarded from the group.

4 Algorithm for the Detection of Malicious/Selfish Behavior

In this algorithm, the core election is performed in the presence of selfish and malicious receivers in the mesh. A selfish and malicious receiver is detected and discarded with the help of estimated and claimed value. Likewise, the removal of selfish and malicious receiver from the group is also performed.

4.1 Objectives and Assumptions

To devise an algorithm for the leader core election, the following conditions are required: (1) To protect all receivers in the group, all the receivers should be monitored by each other. (2) With battery estimation technique, a receiver that shows high or low battery capacity for the purpose to become a core or evade to become a core node is detected and removed from the mesh by comparing estimated value and claimed value. The algorithm is performed on each receiver with the following supposition about the receivers and group architecture:

>Every node in the mesh should aware about its 2-hop neighborhood through connectivity list.

>All the receiver in the mesh should aware about the entering of a new member or leaving of an existing member.

4.2 Core Election

To begin a core election in the group, four types of messages are used i.e., SD message, used by every receiver in the group to start the election process; Begin Election Request message, used to announce election by requesting the cost (battery capacity and position) of each receiver in the group; Acknowledge (r), reply by all receivers through Election Reply message; Send CEM, used to flood the cost of top most receiver in the group:

>*receiver-table(r)*: the list of all receivers in the group voted for the election of the core node *k.*

>*cost-table(r)*: the cost of every receiver in which each receiver keeps the cost of all receiver in the group.

>*neighbors(r)*: the set of receivers *k*…s neighbors.

>*corenode(r)*: The ID of receiver *k*…s core.

>*core(r)*: A Boolean variable that sets to TRUE if receiver *k* is a core and FALSE otherwise.

At the start of the communication, a node (Idn) is searching for the existence of any receiver group (group g). If it receives SD message from any receiver group, then it joins that receiver group and become a member of receiver group as Idr. On the other hand, if there is no receiver group then it announces itself as a core node and makes its own receiver group.

Algorithm 1. Before formation of receiver group (start of communication)

/* on receiving Status Declaration (SD) message, all the nodes will reply along with their cost */

1. **If***Idn* (received SD message from receiver group g) **then**
2. Include receiver-group(g);
3. **else if** (*Idn* = Φ) **then**
4. Send *SD* message
5. **end if**

With the help of algorithm 2, selfish and malicious receivers are detected and removed from the mesh.

Algorithm 2. For the Detection of Malicious/Selfish Behavior Parameters

Input: [Receiver, Ri, B_{dr} , Bdr_{t_3} Threshold limit 10 %]

Output: [Selfish Attack, Malicious Attack]

Begin

Consume battery capacity of Receiver Ri = B_{dr}

Estimated battery capacity of Receiver Ri = Bdr_{t3}

Let in a receiver group (Rgp) there is a Receiver (Ri), having Consume battery capacity (B_{dr})and estimated battery capacity (Bdr_{t3}).

If

$Bdr_{t3} < B_{dr} \pm 10\%$, then

Receiver Ri is a selfish attack and broadcast it

If

$Bdr_{t3} > B_{dr} \pm 10\%$, then

Receiver Ri is a malicious attack and broadcast it

Exit

After the detection of malicious and selfish receiver, a core is elected based on the remaining battery capacity and number of connected neighbors, a Cost i.e., Core Election Message (CEM) is flooded within receiver group to vote the topmost receiver as a core. In reply, all the receivers also flood the CEM to elect the best receiver in a group. Thus, all the receivers will receive a list of receivers from the neighboring nodes.

Algorithm 3. After detection of selfish and malicious receiver

/* After detection of selfish and malicious receiver, all receiver replies with their costs via SD message*/

1. **if** *Idr* (*detect selfish and malicious receiver*) **then**
2. Start *Begin-election Request* message (*Idr; Costr*);
3. Send Acknowledge (r);
4. Send *vote (Idr; Costr)*;
5. Corenode (r) = i;
6. **else if** (*neighbors(r)* = Φ) **then**
7. Send SD message
8. **end if**

After the information is shared between all the receivers through CEM, a core is elected. The core node floods the news of its selection through SD message in a group and updates the receiver group. All receivers will acknowledge core node by receiving it through SD message.

Algorithm 4. Execution by the Elected core node

/* Send an Ack to the receiver in a group */

1. **if** Core (i) = TRUE; **then**
2. Update *receiver-group(g)*;
3. Update *mesh-group(g)*;
4. Update *receiver-table(r);*
5. Update *cost-table(r);*
6. Select *mirror-core* (*Idr; Costr*);
7. Acknowledge (r);
8. Send SD message (i);
9. **end if**

5 Packet Authentication Process (PAP)

In this section, a packet fabrication problem is discussed. In fabrication attack, an attacker as a malicious entity try to insert spoofed Core Election Messages (CEM) to disturb the core election process. The fabrication attack is difficult to detect in MANET as discussed in [11]. Figure 3 explains a fabrication attack, where receiver D broadcasts about its own battery capacity and number of neighbors in CEM for core election process. Let consider that A is a selfish/malicious receiver. Therefore, when a packet is transferred from receiver D to C without proper protection, receiver A simply fabricates the packet and makes a fabricated CEM on behalf of D. Here, an authentication technique is required to protect a CEM of receiver D from being forged. To prevent a selfish/malicious behavior, a digital signature is used here. A digital signature is a small number of extra bits of information appended by D.

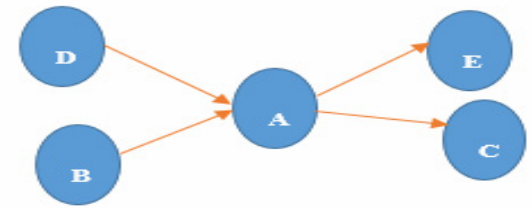

Figure 3. Fabrication attack

Therefore, a one-way hash chain is used, which is effective against selfish/malicious attacks as well as it is not considered as a resource constrained technique. A one-way hash chain is built on a one-way hash function, H. The input of hash function can be of any length, but the output must be of fixed length, i.e. H: {0, 1} * → {0, 1} ṗ, where ṗ is the length of the output of the hash function in number of bits. Likewise, H(x) is simple and can be calculated easily for any given input x.

To make a one-way hash chain, a receiver selects a random value x ε (0,1) ^{TM} and calculate the value of the hash. The first value in hash chain h_0 is represented as x. Hence, $h_i = H(h_{i-1})$, for 0 < i ≤ n, for some n, a chain of h_i is formed:

$$h_0, h_1, h_2, h_3, ..., h_n \quad \textbf{(5)}$$

The propose scheme uses the above equation to protect the packet against fabrication. For authenticated value of h_n , a receiver authenticates h_{n-3} by computing

$H(H(H(h_{n-3})))$ and comparing the result with h_n. For authentication purposes, D shares the h_n to C without informing A. In a wired network, information distribution is made through a trusted certificate authority. On the other hand, in MANET because of dynamic topology there is no centralised administration and hence no base station to perform as a trusted certificate authority. Because of the above reasons, two techniques are proposed to share the initial authentication element h_n of receiver D to receiver C.

The first method is termed as transmission extension technique. Using this method, D increases the transmission power to transmit the h_n directly to C without considering receiver A. This method bypassing the receiver A, and as a result, bypassing the potential threat to the distribution of h_n but it consumes the battery of receiver D quickly. Because of this reason the h_n is only shared, when the whole chain has been used as shown in equation 5. In the second method, h_n is shared through multipath transmission technique. In this technique, h_n is shared within the neighbourhood by receiver D through multiple path to receiver C. The shared h_n has a time to live (TTL) value of two or more hops. Here, C uses a majority vote technique to get h_n from the maximum numbers of receivers which are well behaved. On the contrary, a malicious/selfish receiver A is keen on forging h_n.

Once the h_n is shared from D to C, then D uses $h_i (0 \le i < n)$ consecutively to sign the transmitted packet to C. The h_i is revealed by D one at a time. Consider that h_{i+1} has been revealed initially, i.e. i = n-1. When receiver D transmits a packet to receiver C, it computes a Message Authentication Code (MAC) based on h_{i-1}, [A, C, ID] and appends the value of h_i and MAC with the transmitted packet as shown in Figure 4. The fields of Figure 4 are as follows:

A	*C*	*B*	t	*ID*	*ID*		h_i	h_i
Next hop Receiver NNext Receive	Destination Receiver			Sequence Number		*MAC* Signature	Hash Release	

MAC = [A, C, ID] $_{hi-1}$

Figure 4. Packet format for authentication+

C: the destination or observing receiver.
A: the receiver of the next hop or suspicious receiver.
ID: the corresponding data packet sequence number.
[A, C, ID] h_{i-1}: MAC signed with h_{i-1}.
h_i: the fresh shared element in one-way hash chain $(0 < i < n)$.

Receiver C always know about h_{i+1} and makes the comparison of $H(h_i)$ with h_{i+1}. If the comparison is equivalent, then h_i is accepted and saved. It should be noted that the transmission of message is always sent from D to C. However, the integrity of the packet is only established when the next packet reaches with h_{i-1}. When h_{i-1} is revealed to C, it confirms that the integrity of the packet is accepted previous time by measuring the MAC and matches it with the received packet.

The parameter timeout *t* is applied to place a timer for the packet transmission from D to C. If the timer expires before the packet transmission, then the lost counter of packet transmission increases. Therefore, a suitable value of *t* is crucial for the success of the operation. In this scheme the false alarm started, if the value of *t* is small. Alternatively, if the value is too large then the observing receiver uses a large list, for which a large memory is required. Hence, the value should be large enough that could handle the unsuccessful transmission because of congestion and dynamic topology. The value of *t* should be t > 4* [single hop transmission delay].

6 Performance Evaluation

Simulation Setup

It is implemented in NS-2 to evaluate the performance in the presence of selfish and malicious attacks. NS-2.35 is used on Ubuntu platform using Tcl/Otcl and C++ as a front and back-end languages respectively for implementing our proposed ideas. AWK script is developed and run on random seeds to collect data from NS-2 trace files. The simulation parameters are given in Table 1. In this paper, a random way-point mobility model is used for different mobile scenarios. Also, matrices like throughput, packet delivery fraction, overhead and energy is used to evaluate the performance in the experiments.

Table 1. Simulation parameters

Simulator	Network simulator (NS2)
Examined Protocol	ERASCA
Simulator time	450 Sec
Number of nodes	50
Maximum speed	10 m/s
Simulation area	1000m x 1000m
MAC type	802_11g
Type of attacks	Selfish and Malicious
Maximum Selfish receiver	25
Maximum Malicious receiver	25

7 Results and Discussion

Three scenarios have been simulated to determine the effect of selfish receiver, malicious receiver and both the selfish and the malicious receivers on the performance metrics of ERASCA protocol.
Scenarios 1: Varying the selfish receivers.

Scenarios 2: Varying the malicious receivers.
Scenarios 3: Varying both selfish and malicious receivers.

7.1 Simulation Results for Selfish Receivers

In Table 2, the following parameters are used. The results in Figure 5 show the comparison of PUMA, MAODV and ERASCA in the presence and absence of Battery Estimation Technique (BET). BET highlighted the effectiveness in term of PDF, throughput, overhead and energy consumption. In the absence of BET, receivers do not detect the selfish receivers and, hence, select selfish receivers for data communication.

Table 2. Specific simulation parameters for selfish receivers

Number of nodes	50
Maximum speed	10 m/s
Min Minimum Selfish receivers	5
Maximum Selfish receivers	25

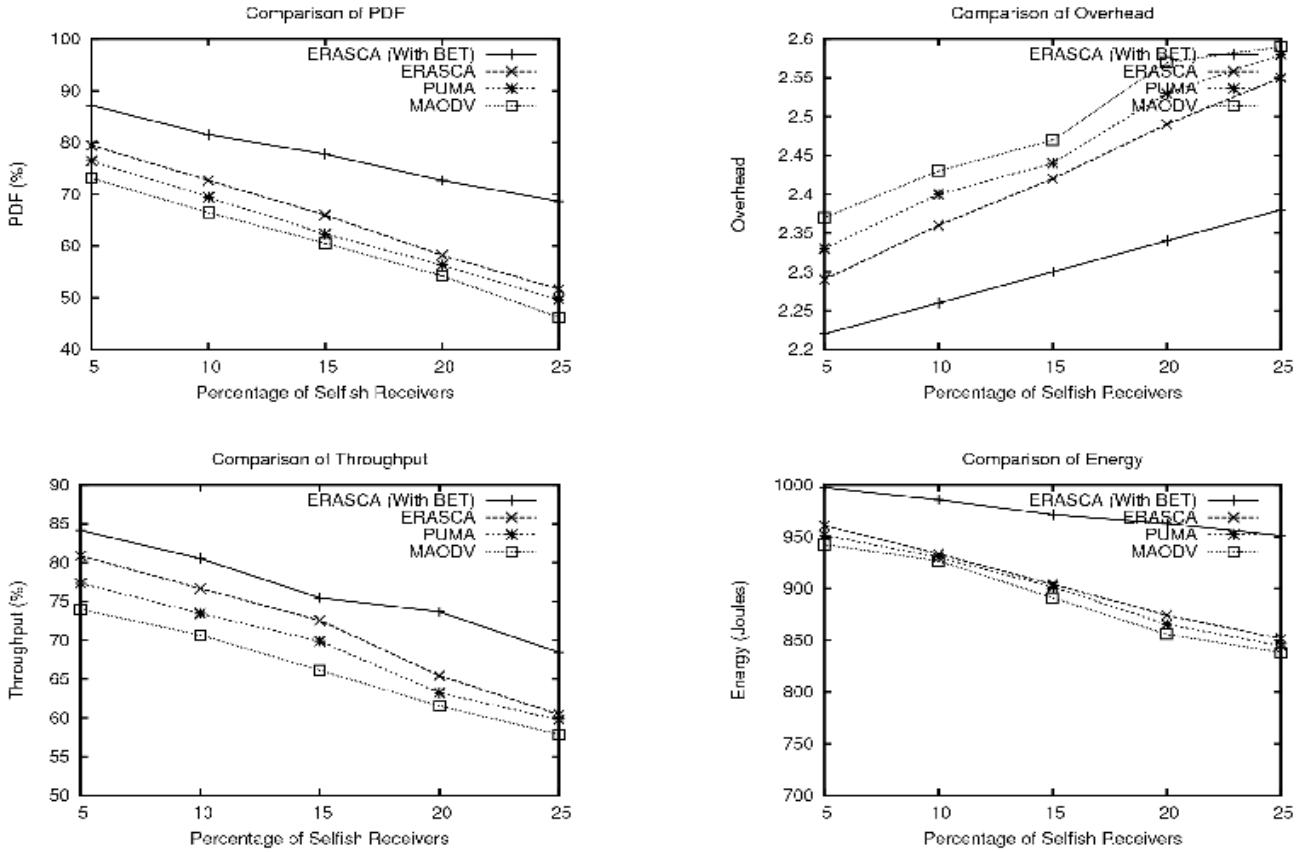

Figure 5. Simulation of varying number of selfish receivers with PDF, overhead, throughput and energy

These selfish receivers do not encourage other receivers to route the data through them by showing less battery capacity than the original battery capacity for energy saving. As we know the receiver in a group does not prefer to transfer the data through a receiver having minimum battery capacity and therefore could select a receiver with longer route having maximum battery capacity. Therefore, decreases the throughput with increase in overhead, delay and energy consumption. On the other hand, BET successfully detects selfish receivers in the routing process. The BET increases the throughput and PDF by selecting shorter paths. Also, the higher PDF keeps the routing overhead minimum, which is computed per received data packets. Figure 5 shows, that when the number of selfish receivers increases the performance does not deteriorate poorly in BET and shows its effectiveness in term of PDF, throughput, energy and overhead. In Figure 5, the performance of ERASCA is better than the PUMA and MAODV due to proper core Election. In PUMA and MAODV, the core is easily compromised to selfish receiver due to inappropriate core election process and as a result the performance is deteriorated. On the other hand, due to proper core election, the core is not easily exposed to selfish receiver and hence improve the security of ERASCA. At the end, after implementation of BET, the ERASCA shows more better performance as compared to PUMA and MAODV.

7.2 Simulation Results for Malicious Receivers

The parameters in Table 3 show the presence of malicious node. Figure 6 shows the performance results in the presence of malicious receivers. In our protocol, the performance in the presence of malicious receivers are very poor as compared in the presence of selfish receivers.

Table 3. Specific simulation parameters for malicious receivers

Number of nodes	50
Maximum speed	10 m/s
Minimum Malicious receivers	5
Maximum Malicious receivers	25

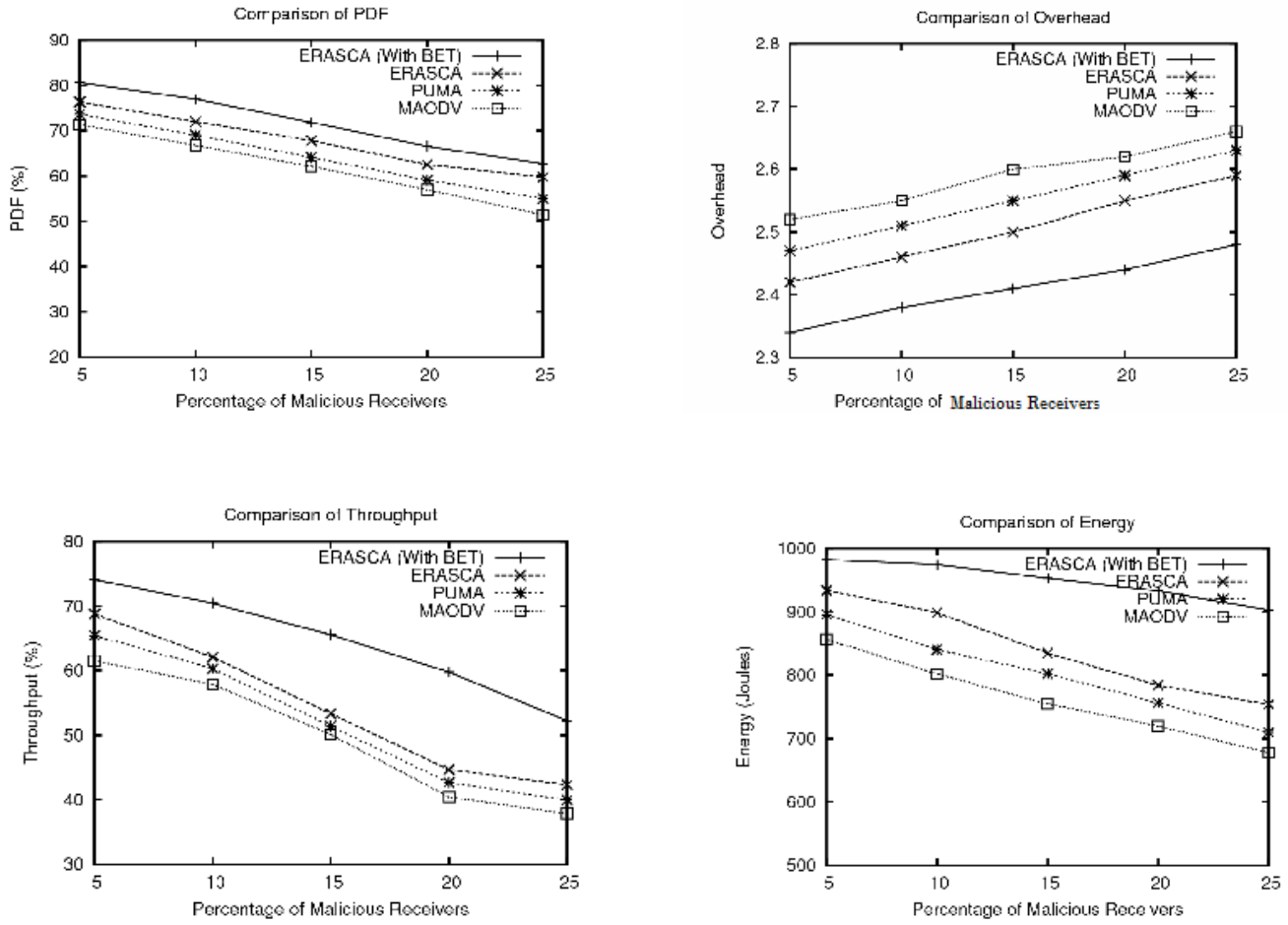

Figure 6. Simulation of varying number of malicious receivers with PDF, overhead, throughput and energy

As discussed earlier, selfish receiver only wants to become a member of the mesh group and try to avoid communication for saving energy and having no interest to become the core node. However, malicious receiver is interested to become a mesh member as well as interested to become a core node. If the malicious receiver becomes a core node, then it considerably deteriorates the performance by flooding false information within the network as shown in Figure 6 as compared in the presence of selfish receivers as shown in Figure 5.

7.3 Simulation Results for Malicious and Selfish Receivers

In Table 4, specific simulation parameters are given. As shown in Figure 7, both selfish and malicious receivers are used. The results in Figure 6 shows poor performance in the presence of malicious receiver, but shows improved performance in the presence of only selfish receiver (Figure 6).

Table 4. Specific simulation parameters for malicious and selfish receivers

Number of nodes	50
Maximum speed	10 m/s
Minimum Malicious and Selfish receivers	5
Maximum Malicious and Selfish receivers	25

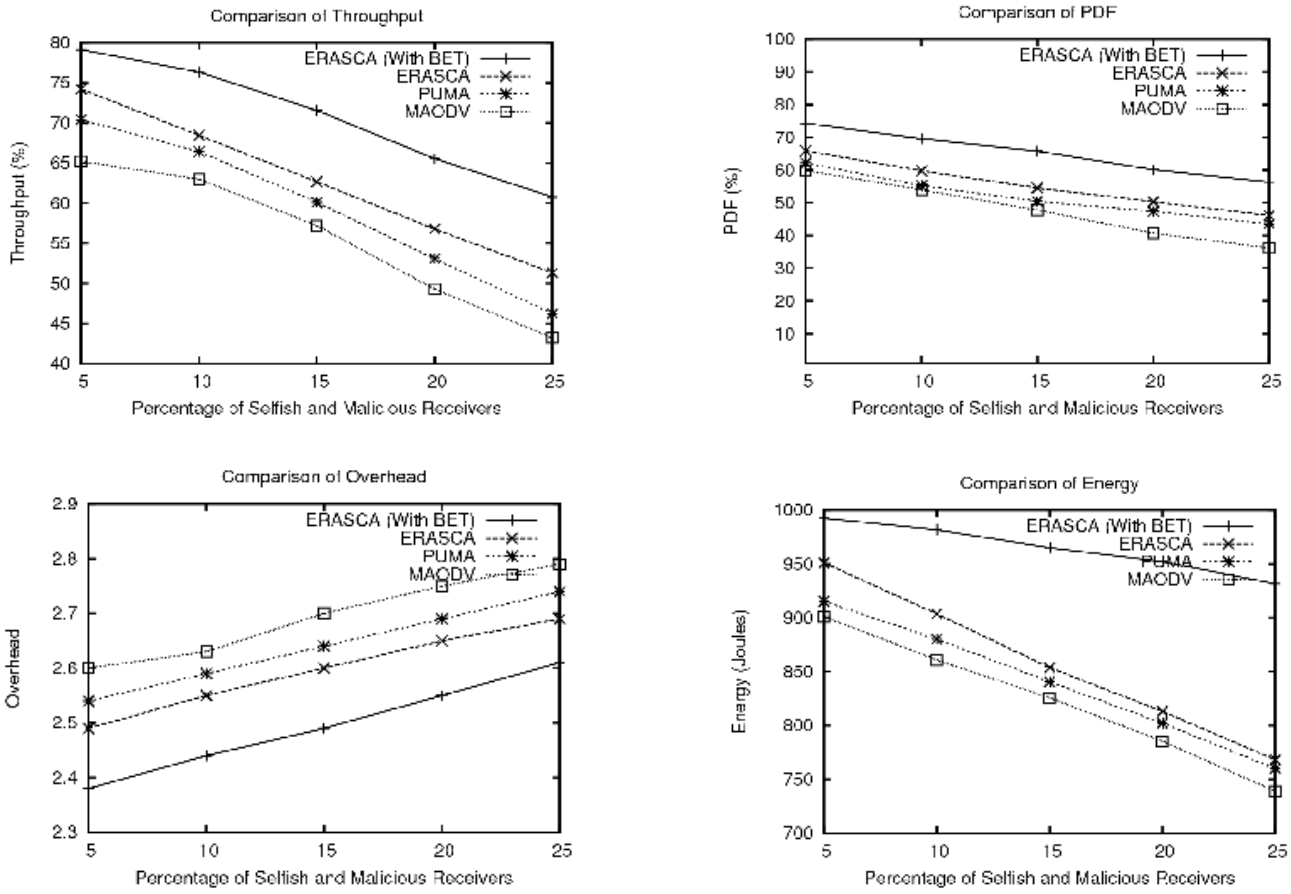

Figure 7. Simulation of selfish and malicious receivers with PDF, overhead, throughput and energy

Figure 7 shows that in the presence of selfish and malicious receivers the performance decreases, however, in the presence of BET malicious and selfish receivers are detected and hence the performance increases. By increasing the number of selfish and malicious receivers, PDF and throughput decreases, however, this decrease in

PDF and throughput will not occur in the presence of BET. Likewise, the non-cooperation and malicious behaviour in the receiver group decreases the packet transmission to the destination, which increases the resending of data and control overhead, hence increases the energy consumption. However, BET detects the malicious and selfish receivers and allows the member receivers to cooperate with each other and, hence, increases the performance.

8 Conclusion

Two types of attacks have been discussed in this paper i.e. selfish and malicious attack. In selfish attack, a receiver does not cooperate with the group members to save its battery capacity and in malicious attacks, a receiver tries to become a core node and disrupt the maintenance and update process. To tackle such attacks, a BET was introduced to detect and discard the selfish and malicious receivers. The results showed that in the presence of BET, the performance in term of throughput, PDF, overhead and energy utilization are better as compared in the absence of BET. Also, a packet authentication process was discussed to further increase the integrity of packets.

References

[1] R. Hemangini, M. Nirupama, Study of Routing Protocols in Mobile Ad Hoc Network, *SSRG International Journal of Mobile Computing and Application (SSRG-IJMCA)*, Vol. 2, No. 2, pp. 10-14, May-August, 2015.

[2] A. Kodole, P. Agarkar, A Survey of Routing Protocols in Mobile Ad Hoc Networks, *Multidisciplinary Journal of Research in Engineering and Technology*, Vol. 2, No. 1, pp. 336-41, January, 2015.

[3] N. Lal, S. Kumar, A. Saxena, V. K. Chaurasiya, Detection of Malicious Node Behaviour via I-watchdog Protocol in Mobile Ad Hoc Network with DSDV Routing Scheme, *Procedia Computer Science*, Vol. 49, pp. 264-273, 2015, DOI: 10.1016/j.procs.2015.04.252.

[4] R. F. Olanrewaju, B. U. I. Khan, R. N. Mir, B. W. Adebayo, Behaviour Visualization for Malicious-Attacker Node Collusion in MANET Based on Probabilistic Approach, *American Journal of Computer Science and Engineering*, Vol. 2, No. 3, pp. 10-19, May, 2015.

[5] S. Kriplani, R. Kesharwani, A Survey on Malicious and Selfish Nodes in Mobile Ad Hoc Networks, *International*

Journal of Scientific Research in Science, Engineering and Technology, Vol. 1, No. 5, pp. 244-251, September-October, 2015.

[6] R. Sruthi, R. Vijayakumar, Prevention of MANETS from Malicious Node Attacks, *International Journal of Computer Applications*, Vol. 112, No. 14, pp. 23-25, February, 2015.

[7] F. Khan, S. Abbas, S. Khan, An Efficient and Reliable Core-Assisted Multicast Routing Protocol in Mobile Ad-Hoc Network, *International Journal of Advanced Computer Science and Applications*, Vol. 7, No. 5, pp. 231-242, May, 2016.

[8] F. Khan, A. W. Khan, K. Shah, I. Qasim, A. Habib, An Algorithmic Approach for Core Election in Mobile Ad-hoc Network, *Journal of Internet Technology*, Vol. 20, No. 4, pp. 1099-1111, July, 2019.

[9] S. Bohra, N. Choudhary, An Efficient Misbehaving Node Detection Algorithm in Manet, *Global Journal of Computer Science and Technology: H Information & Technology*, Vol. 15, No. 2, pp. 17-23, 2015.

[10] E. Zamani, M. Soltanaghaei, The Improved Overhearing backup AODV Protocol in MANET, *Journal of Computer Networks and Communications*, Vol. 2016, Article ID 6463157, 2016.

[11] J. M. S. P. J. Kumar, A. Kathirvel, N. Kirubakaran, P. Sivaraman, M. Subramaniam, A Unified Approach for Detecting and Eliminating Selfish Nodes in MANETs Using TBUT, *EURASIP Journal on Wireless Communications and Networking*, Vol. 2015, No. 1, pp. 1-11, 2015.

[12] P. Mohapatra, S. Krishnamurthy, *AD HOC NETWORKS: Technologies and Protocols*, Springer Science & Business Media, 2004.

[13] G. Acs, L. Buttyan, I. Vajda, Provably Secure On-demand Source Routing in Mobile Ad Hoc Networks, *IEEE Transactions on Mobile Computing*, Vol. 5, No. 11, pp. 1533-1546, November, 2006.

[14] M. Burmester, B. De Medeiros, Towards Provable Security for Route Discovery Protocols in Mobile Ad Hoc Networks, *IACR Cryptology ePrint Archive*, Vol. 2007, pp. 324, 2007.

[15] G. Ács, L. Buttyán, I. Vajda, Provable Security of On-demand Distance Vector Routing in Wireless Ad Hoc Networks, in *Security and Privacy in Ad-hoc and Sensor Networks (ESAS)*, Visegrad, Hungary, 2005, pp. 113-127.

[16] F. Javed, S. Khan, A. Khan, A. Javed, R. Tariq, Matiullah, F. Khan, On Precise Path Planning Algorithm in Wireless Sensor Network, *International Journal of Distributed Sensor Networks*, Vol. 14, No. 7, pp. 1-12, July, 2018.

[17] L. Buttyán, T. V. Thong, Formal Verification of Secure Ad-hoc Network Routing Protocols Using Deductive Model-checking, *Third Joint IFIP in Wireless and Mobile Networking Conference (WMNC)*, Budapest, Hungary, 2010, pp. 1-6.

[18] S. Roy, D. Koutsonikolas, S. Das, Y. C. Hu, High-throughput Multicast Routing Metrics in Wireless Mesh Networks, *Ad Hoc Networks*, Vol. 6, No. 6, pp. 878-899, August, 2008.

[19] Y. Li, I. R. Chen, Hierarchical Agent-based Secure Multicast for Wireless Mesh Networks, *International Conference on Communications (ICC)*, Kyoto, Japan, 2011, pp. 1-6.

Biographies

Faheem Khan did his Ph.D. from University of Malakand, Pakistan. He is currently working as Assistant Professor/Director IT at the Department of Computer Science, The University of Lakki Marwat, Pakistan.

Abdul Wahid Khan did his Ph.D. in Global Software Engineering in 2016 from Pakistan. At Present he is working as Assistant Professor at Department of Computer Science, University of Science & Technology Bannu, KP, Pakistan.

Samiullah Khan is currently Assistant Professor at The University of Agriculture Peshawar, Pakistan. He has completed his Ph.D. in Computer Science at Capital University of Science and Technology, Islamabad-Pakistan.

Iqbal Qasim did his Ph.D. is Web mining from South Korea. At Present he is working as Assistant Professor at Department of Computer Science, University of Science & Technology Bannu, KP, Pakistan.

Asad Habib earned his Doctor of Engineering degree from the Graduate School of Information Science, NAIST (Nara Institute of Science and Technology) Japan. At present, he is supervising university teachers and graduate students in research and development at the Institute of Computing, Kohat University of Science and Technology, Pakistan.