# AN EVALUATION ON SELFISH BEHAVIOUR ATTACK AND JELLYFISH ATTACKS UNDER AODV ROUTING PROTOCOL

Bhawna Singla[1], A.K.Verma[1]and L.R.Raheja[2]

[1]Thapar University, Patiala, India
[2]Indian Institute of Technology, Kharagpur, India

***ABSTRACT***

*The Mobile Adhoc network applications are increasing every day due to the properties like infrastructure less nature, dynamic topology and multihop network. A set of rule that govern which path is to be followed from the source node to destination node is called routing protocol which can be further divided into number of other categories like proactive and reactive. This paper discusses the AODV, a kind of reactive routing protocol, for its study. But MANETs is also vulnerable to number of attacks due to its nature. A lot of work has been done to make it more secure. The work that is addressed in this paper is the selfish node attack and jellyfish attack on AODV routing protocol. The selfish node attack is a kind passive attack in which node does not participate in routing process by not forwarding the packets. And jellyfish attack is the sub type of black hole attack that delays or drops the packets for certain amount of time. This paper studies all the three types of jellyfishattack: Jellyfishreorder attack, jellyfishperiodic dropping attack and jellyfishdelay variance attack. In this work we study the impacts of some of the attacks on network under a short term military rescue mission like scenario. We will do a comparative analysis of three kinds of JellyfishAttacks with selfish Behaviour Attack under AODV routing protocol. The analysis will be made with respect to different network sizes and under the presence of different number of attackers in the network. The impact on the performance will be measured with suitable metrics to understand the nature of different attacks.*

***KEYWORDS***

*MANETS Node , Attack , Jellyfish Attack, AODV, Performance , Energy Consumption, EED*

## 1. INTRODUCTION

Mobile Adhoc Network (MANET) is a wireless network of nodes that can freely organize it into arbitrarily and dynamically chosen topology [1]. It does not require any pre existing infrastructure. The main application of MANETS include the military or disaster sites as the soldiers of the military in battled or rescue teams must establish their own independent network. The routing protocols [2,3] of MANETs can be divided into two categories: proactive and reactive protocols. Proactive Protocols: In the Proactive routing proto-col up-to-date route information is maintained by every node, by exchanging topological information among the node, in form of routing tables. Route ta-ble updates are exchanged periodically. Reactive Protocol: In Reactive routing protocol, the route is discovered whenever there is requirement. It is also known as on demand routing protocol [4,5]. On demand routing protocols have two stages: a. Route Discovery: In route discovery phase source node checks to see whether it has path from source to destination, if it has, it uses that route otherwise it initiates route discovery process. b. Route Maintenance: The dy-namic topology of the network causes route failure due to link breakages etc. So, route maintenance is necessary.

### 1.1 THREATS ON MANET AND MANET ROUTING

Due to changing topology and infrastructure less nature MANET is vulnerable to number of attacks [6-8]. The attack can be passive or active. A passive attack does not cause any destruction to the network rather it listens to the tra c to access secret information, while active attack causes destruction by modifying the content of packet, injecting packet to invalid destination etc As in MANET, every node is a router but some nodes perform it in a negative way. The nodes performing adverse effects on MANETS are classified into two categories: malicious node and node. Malicious nodes are those nodes that perform active attack on MANETS and may be active in route establishment or data forwarding phase, while node perform passively by not forwarding the packet just for sake of saving battery energy etc [9,10].

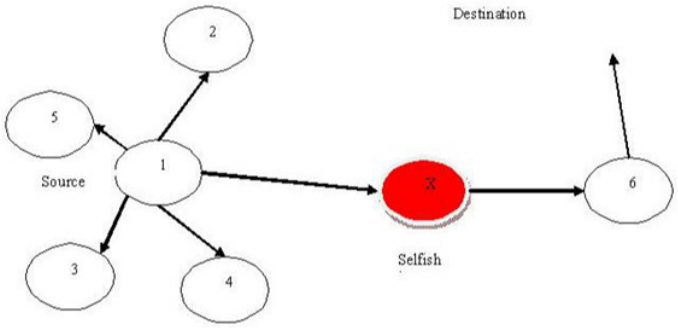


Fig. 1 Selfish Node Attack in AODV Routing Protocol

## 2. ATTACKS UNDER INVESTIGATION

### 2.1 SELFISH MISBEHAVIOR ATTACK

Whenever the selfish node feels that the packets requires lot of resources, the selfishnode does not forward it in the network [11-13].Node misbehaviors and failures causes node isolation problem. In node isolation phenomenon, nodes do not have active neighbors. Selfishnodes give rise to isolated nodes even if active nodes are still available. In figure 1, suppose node X is the selfish node. Suppose source node is node no. 1 and the destination node is 7, node 1 selects X as the next hop and sends data to it but X discards all the data forwarded to it or any other node at a distance of more than one hop away. In this way node 1 will become isolated. However, selfishnodes can still make the communication with all other nodes (via cooperative neighbors).Selfishnodes are of three types:

#### 2.1.1 NO PACKET FORWARDING

In this type, Selfishnode is does not forwards the packet. While it still works in route discovery or route maintenance phase of AODV protocol.

#### 2.1.2 NO PARTICIPATION

No Participation: In this type selfishnode does not participate in route dis-covery phase of AODV protocol. Due to this network maintenance becomes more signi cant as compared to route discovery phase. If node does not par-ticipating in Route Discovery Phase, then there is no route including selfishnode, as a result, packet forwarding function will never execute.

#### 2.1.3 PARTIAL PACKET FORWARDING WITH ENERGY SAVING

During normal operation, node simply consumes energy while performing net-work function like packet forwarding and routing.

### 2.2 JELLYFISHATTACK

Jellyfish attack is very prominent in the given scenario where the mobility is more and the route lifetime is short. Jellyfish attack by Aad et al.[14,15], is one of the denials of service attack and also a type of passive attack which is di cult to detect. It does not disobey the rules of the routing protocol. In case of distance vector routing protocol, malicious node will obey all the control messages. However, once the route is established, it will reduce the through-put of the network via jellyfish attack. It is also very di cult to distinguish the jellyfish attack from congestion and packet losses that occur naturally in a network, and therefore is hard and resource-consuming to detect. The goal of jellyfish attack is to reduce the performance of network to near about zero without dropping zero or negligible number of packets. Jellyfishattack can be classified into three categories [16,17]: 1) JF reorder attack 2)Periodic Dropping Attack 3)Delay variance attack.

#### 2.2.1 JF REORDER ATTACK

TCP has a vulnerability that the packets once transmitted sequentially may arrive at the destination in unordered sequence due to multipath ordering routing and route changes. No TCP variant is robust to malicious reordering of packets. Let ACK-N be the acknowledgement that all the segments from 1,,N have been received. Then receipt of duplicate ACK-N will show the out of order packets.

#### 2.2.2 PERIODIC DROPPING

In this malicious node drops the some percentage of packets for maliciously chosen period.TCP throughput can became equal to nearly zero even for the small values of x.

#### 2.2.3 DELAY VARIANCE ATTACK

In this the malicious node delays the packet while preserving the order in which packets are transmitted.

## 3. MODELING MISBEHAVIOR IN AODV ROUTING PROTOCOL

Generally, for implementing different types of attack, it is needed to change several sections of a routing protocol. But in our model, we tried to simplify that. We tried to minimize the additional lines count, line change count while implementing several types of attacks; so that one can easily understand the way in which these kinds of attacks are really working.

### 3.1 PSEUDO CODE OF DIFFERENT ATTACKS

The following Pseudo Code Explains the Changes needed in packet forwarding stage of AODV for simulating malicious behavior.

# 4.SIMULATIONS OF ATTACKS UNDER NS2 SIMULATOR

We used network simulator version NS2.35 under Ubuntu Linux operating system for achieving best performance in terms of speed. We have implemented the attacks on the AODV code of NS2.

## 4.1 THE CHANGES MADE IN NS2 AODV CODE

### 4.1.1 CHANGES MADE IN AODV.H

The additional function de nitions for simulating attacks and the variables that will be bound with TCL are declared in aodv.h. By using the variables from a tcl simulation code, we can control the behavior of the routing agent.

Fig. 2 Forward Packet Function in case of selfishnode attack and jellyfish attack

***1. forward(Pkt, Delay)***
***2. if ( ttl=0 )***

***2.1. drop(Pkt);***
***2.2. return;***

***3. if (pkt is addressed to this node)***
***3.1. recv(pkt);***

***3.2. return;***
***4. if (pkt is a AODV broadcast)***

***4.1. scheduleTransmission(pkt,delay) //The Attacks on Aodv pkt is not imple-mented here***

***4.2. else // here it is a data packet which needs to be forwarded***
***4.3. If (AttackMode= none)***

***4.4. scheduleTransmission(pkt,delay)***

***4.5. else if (AttackMode= Jelly shReorder)***

***//If dest is me then process the packet normally***
***4.6. if ( pkt addressed to this node)***

***4.7. scheduleTransmission(pkt,delay)***
***4.8. else***

***//here we are imposing reorder //by scheduling the packets at random time***
***4.9. delay= Jelly shReorderLimit * Rand() ;***

***4.10. scheduleTransmission(pkt,delay)***

***4.11. else if (AttackMode= Jelly shPeriodicDropping) //If dst is me then process the packet normally***

***4.12. if ( pkt addressed to this node)***
***4.13. scheduleTransmission(pkt,delay)***
***4.14. else // imposing periodic packet dropping //by scheduling the packets at random time***

***4.15. if (Jelly shAttackProbability > Rand())***
***4.16. scheduleTransmission(pkt,delay)***

***4.17 else //Malicious Dropping***
***4.18. drop(pkt)***

***4.19. else if (AttackMode= Jelly shDelayVariance) //If dest is me then process the packet normally***

***4.20. if ( pkt addressed to this node)***
***4.21. scheduleTransmission(pkt,delay)***

***4.22. else //scheduling the packets with high delay***
***4.23. delay= Jelly shAttackDelay+ Rand() ;***

***4.24. scheduleTransmission(pkt,delay)***
***4.25. else if (AttackMode= SelfishBehavior) //If dest is me then process the packet normally***

***4.26. if ( pkt addressed to this node)***
***4.27. scheduleTransmission(pkt,delay)***

***4.28. else //here we are behaving selfishly***
***4.29. MaliciousDrop(pkt);***

#### 4.1.2 CHANGES MADE IN AODV.CC

The actual code of the additional function de nitions for simulating attacks were implemented in aodv.cc. And here the new interfaces to the code through the control variables that will be bound with tcl are written here. By setting the variables from a tcl simulation code, we can control the behavior of the routing agent.

Table 1 Simulation Parameters

| | |
|---|---|
| *Topographical Area* | *1800 X 500* |
| *Mobility* | *20m/s* |
| *Pause Time* | *20s* |
| *Total Simulation Time* | *100s* |
| *Routing Protocol* | *AODV* |
| *Mobility Model* | *RandomWaypoint* |
| *Channel Model* | *WirelessChannel* |
| *Propagation Model* | *TwoRayGround* |
| *PhyModel* | *WirelessPhy* |
| *MacModel* | *802 11* |
| *AntennaModel* | *OmniAntenna* |
| *Queue* | *DropTail-PriQueue* |
| *Queue Length* | *50* |

## 4.2 THE FUNCTIONS MODIFIED FOR SIMULATING ATTACKS

### 4.2.1 THE FUNCTION AODV::COMMAND( .. )

Here the interface to the newly added functionalities are provided. It means, we can set some of the variables of C++ code from the tcl simulation script through the interfaces provided in this function.

### 4.2.2 THE FUNCTION AODV::AODV( .. )

In the constructor section of the aodv code, the code needed for binding of new control variables is added.

### 4.2.3 THE FUNCTION AODV::FORWARD( .. )

In this function, the code for different attacks such as JellyfishReorder Attack, JellyfishPeriodic Dropping Attack JellyfishDelay Variance Attack and SelfishBehavior Attack were implemented. With respect to the value of a control vari-able AttackType, the aodv routing agent will behave normal or do a particular attack.

After the modi cations on aodv.h and aodv.cc, the new version of ns2 is compiled to incorporate the modi ed version of AODV routing agent. Now the modi ed version of AODV routing agent can be used in a tcl simulation code. And the functionality of the aodv agent can be controlled by setting up the suitable value in control variable or a using appropriate aodv initialization function that is newly added in AODV::command( .. ) section.

## 5.RESULTS AND DISCUSSION

In our simulation, we used following common parameters of table 1 while setting up the network.

Table 2 Trace Parameters

| | |
|---|---|
| Transport Agent | TCP |
| No Flows | 10 |
| Trace Type | CBR |
| Packet Size | 1Kb |
| Interval | 100ms |
| Rate | 10kb |

### 5.1 TRACE PARAMETERS

The following parameters of table 2 are used to setting up the tcp ows with some periodic data.

### 5.2 VARIABLE PARAMETERS

The following parameters of table3 are used as variables for analyzing the impact of the different attacks on different condition.

| | |
|---|---|
| Attacking Nodes | 5, 10, 15 and 20 |
| Total Nodes | 40,50,60 |
| Simulated Attacks | a) Selfish Behavior attack<br>b) JellyfishReorderAttack<br>c) JellyfishPeriodicDroppingAttack<br>d) JellyfishDelayVarianceAttack |

### 5.3. METRICS CONSIDERED FOR EVALUATION

To calculate its performance the following metrics are considered:

#### 5.3.1 DATA PACKETS MALICIOUSLY DROPPED AT ROUTING LAYER

The count of data packets maliciously dropped at routing layer is the main metric which will help us to understand the malicious behavior of a attack at routing layer.

#### 5.3.2 TOTAL DATA PACKETS SENT

The count of data packets sent at source is a metric which will help us to under-stand impact of malicious behavior of a attack at routing layer on Application Layer.

#### 5.3.3 TOTAL DATA PACKETS RECEIVED

The count of data packets received at destination is a metric which will help us to understand impact of malicious behavior of a attack at routing layer on Application Layer.

#### 5.3.4 ACHIEVED THROUGHPUT

The throughput is a important metric which will show the impact of attack. Throughput is de ned as the number of bits or bytes sent by source to desti-nation per unit time. In this paper, it is measured in Kbps .

### 5.3.5 PACKET DELIVERY FRACTION (PDF)

Packet delivery fraction is the ratio of the number of packets received at des-tination to the total number of packets sent from the source.

### 5.3.6 END-TO-END DELAY

End-to-End delay(EED) is the average time interval between the generation of a packet at a source node and the successfully delivered at the destination node.

### 5.3.7 CONSUMED BATTERY ENERGY

The consumed energy per node is the metric which will show whether the at-tack caused energy loss. Battery energy consumption in this paper is measured in Joules.

### 5.3.8 DROPPED PACKETS AT SOURCE AND DESTINATION

We considered the packets dropped at source and destination as a metric to measure the impact of attack. Because, it an intermediate attacker causes destruction to a data ow, then it may induce packet loss at source and des-tination itself.

## 5.4 ANALYTIC RESULTS WITH RESPECT TO DIFFERENT NUMBER OF MALICIOUS NODES

In the following analysis the total number of nodes in the network is kept as 40 and among them the number of malicious nodes were varied as 10, 15 and 20. And the impact is measured using different metrics.

The following line graph shows the impact of different attacks in terms of total data packets sent at application source. As shown in the line graph in figure 3, under the presence of Attacks the application source itself can not able to send much. Selfish Behavior Attack seems to be causing little bit higher impact than all the Jellyfish Attacks. With respect to the increase of no of attackers, the performance decreases.

The following line graph in the figure 4 shows the impact of different attacks in terms of total data packets received at application destination. As shown in the line graph, under the presence of Attacks the application destination can not able to receive much. Selfish Behavior Attack seems to be causing higher impact than all the JellyfishAttacks. With respect to the increase of no of attackers, the performance decreases.

The following line graph in the figure 5 shows the impact of differentattacks in terms of data packets dropped at source and destination. It signi es the packets dropped at application layer. As shown in the line graph, SelfishBe-havior Attack Attack caused much packet dropping at application layer. The following line graph in the figure 6 shows the impact of differentattacks in terms of maliciously dropped data packets at routing layer. As shown in the line graph, except JellyfishReorder Attack and JellyfishDelay Variant attack, all the other attacks maliciously dropping packets at routing layer. Of course, conceptually, JellyfishReorder Attack and JellyfishDelay Variant attack will not drop any packet at routing layer; but only a ect the packet transmis-sion/forwarding in differentway. The selfishbehavior causes little bit of high data packet drop at routing layer. With respect to the increase of no of at-tackers, the malicious drops at routing layer is getting increase considerably.

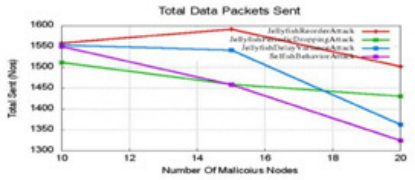


Fig. 3 Attackers vs Sent Data Packets

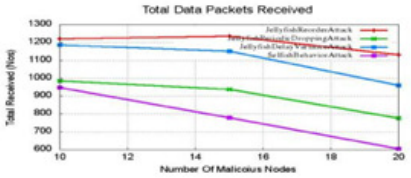


Fig. 4 Attackers vs Received Data Packets

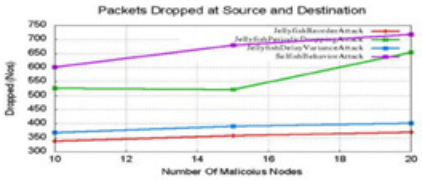


Fig. 5 Attackers vs Dropped at Application Layer

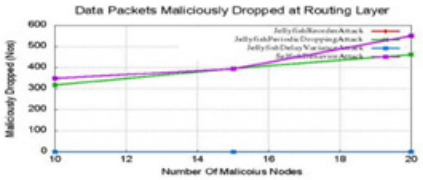


Fig. 6 Attackers vs Maliciously Dropped at Routing Layer

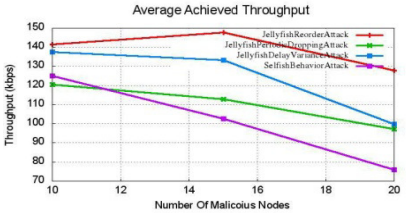


Fig. 7 Attackers vs Throughput

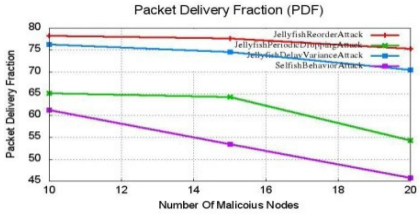


Fig. 8 Attackers vs Dropped PDF

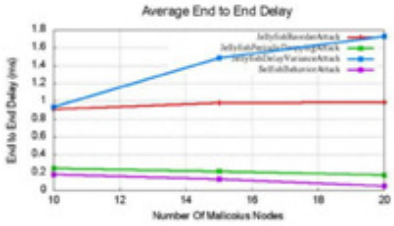


Fig. 10 Figure 10 Attackers vs Battery Energy

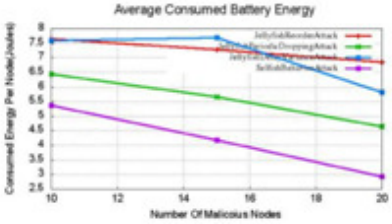


Fig. 9 Attackers vs End to End Delay

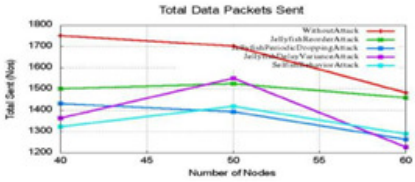


Fig. 11 Network Size vs Sent Packets

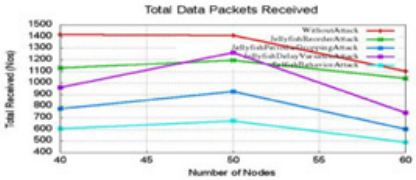


Fig. 12 Network Size vs Received Pack-packets

The following line graph in the figure 7 shows the impact of different attacks in terms of average achieved throughput of TCP ows. As shown in the line graph, Selfish Behavior Attack caused much packet loss so that the throughput was very lower than all other attacks. The following line graph in the figure 8 shows the impact of different attacks in terms of Packet Delivery Fraction (PDF) of tcp ows. As shown in the line graph, Selfish Behavior Attack seems to be causing little

bit higher impact than all the Jellyfish Attacks. With re-spect to the increase of no of attackers, the performance decreases. In most of the earlier papers, the metrics EED and Energy consumption were not stud-ied in detail or with proper comparision, because, the performance in terms of these two metrics will be somewhat strange to a researcher who always expect worst performance in the presence of an attack. Even in some previous papers, we may found some wrong interpretation for these graphs or even in correctly prepared graphs for these metrics.

The following line graph in the figure 9 shows the impact of different attacks in terms of End to End Delay (EED) of tcp ows. As shown in the line graph, Selfish Behavior Attack seems to be providing lower EED than all other Jel-ly sh Attacks but certainly it does not mean that Selfish Behavior Attack are improving the performance or the network. With respect to the increase of no of attackers, the performance getting a ected with respect to the na-ture of attack. The low end to end delay under this two attacks are due to a strange fact that these two attacks makes disconnection in tcp ows and since the packets are not at all forwarded to any further nodes, indirectly it is reduce the message overhead in the network and reduced bandwidth usage otherwise it will be consumed by the forwarded data packets. So, the ows that were unaffected by Selfish Behavior Attack (the connections where there is no neighboring attack nodes) utilizes that extra bandwidth and gains some performance in term of some metrics. Further, keep in mind that the end to end delay is only calculated based on the time in which a packet is sent and received. So if a packet is not received, in that case end to end delay can not be calculated. So this average EED is only the average EED of successfully delivered packets. Understanding these strange facts requires a better visual-ization of the whole network scenario. The following line graph in the figure 10 shows the impact of different attacks in terms of consumed battery energy. As shown in the line graph, in the presence of Selfish Behavior Attack the battery consumption is lesser than all other Jellyfish Attacks but certainly it does not mean that Selfish Behavior Attack is improving the performance in terms of energy consumption. The low energy consumption under this two attacks are due to a strange fact that these two attacks makes disconnection in tcp ows and since the packets are not at all forwarded to any further nodes, indirectly it is reduce the battery consumption at the other nodes otherwise it will be consumed for forwarding the data packets. So, the nodes that were unaffected by Selfish Behavior Attack (where there is no neighboring attack nodes) preserves some battery power. Understanding this strange fact requires a better visualization of the whole network scenario.

Lot of previous papers saying that the attacks will increase energy consump-tion. Of course, it also may be true but not in the same sense. For example if an application will continuously try to send data under attack, then the bat-tery of the sending node and some other nodes between sender and attacker nodes will get reduced rapidly. If the application will vigorously try to do re-transmission due to loss, then this will increase the energy consumption. But under tcp, it will handle lossy scenario and just reduce the sending rate to avoid further losses. That is why the average energy consumed in the network seems to be getting reduced. Understanding this strange fact requires a better visualization of the whole network scenario.

### 5.5 ANALYTIC RESULTS WITH RESPECT TO DIFFERENT NETWORK SIZE

Here we see the analytic results of comparision of different attacks with normal AODV (it means performance without any attack). And it is studied with Respect to Different Network Size.

In the following analysis the total number of nodes in the network is varied as 40, 50 and 60 and among them, the number of malicious nodes kept as 20. And the impact is measured using different metrics.

The following line graph in the figure 11 shows the impact of different attacks in terms of total data packets sent at application source. As shown in the line graph, under the presence of Attacks the application source itself can not able to send much. Selfish Behavior Attack seems to be causing almost equal impact like all the Jellyfish Attacks. But without the presence of any attack

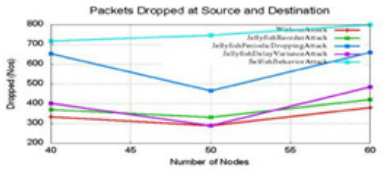


Fig 12 Dropped At Application Layer

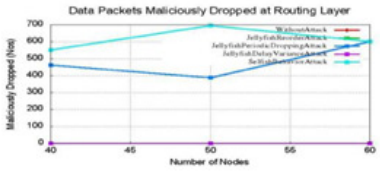


Fig 13 Dropped at Routing Layer

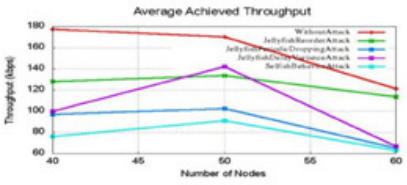


Fig. 15 Network Size vs Throughput

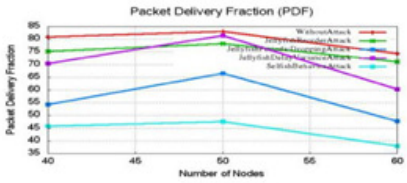


Fig. 16 Network Size vs PDF

AODV performs good and able to send much data packets. With respect to the increase of no of nodes in the network, the performance decreases in most of the cases. The following line graph in the figure 12 shows the impact of different attacks in terms of total data packets received at application destina-tion. As shown in the line graph, Selfish Behavior Attack seems to be causing much impact than all the Jellyfish Attacks. But even without the presence of any attack AODV performs good and able to send much data packets. With respect to the increase of no of nodes in the network, the performance de-creases in most of the cases.

The following line graph in the figure 13 shows the impact of different attacks in terms of data packets dropped at source and destination. It signi es the packets dropped at application layer. As shown in the line graph, Selfish Behavior Attack and JellyfishPeriodic Packet Dropping Attacks were causing much packet drop at application layer. The other two types of JellyfishAttacks also causing little packet drop at application layer. But without the presence of any attack, AODV performs good and dropping less packets at application layer. The following line graph in the figure 14 shows the impact of different attacks in terms of maliciously dropped data packets at routing layer. As shown in the line graph, except JellyfishReorder Attack and JellyfishDelay Variant attack, all the other attacks maliciously dropping packets at routing layer. Of course, conceptually, Jellyfish Reorder Attack and JellyfishDelay Variant attack will not drop any packet at routing layer; but only a ect the packet transmission/forwarding in different way. The selfish behavior causes little bit of high data packet drop at routing layer. With respect to the increase of no of nodes in the network, the malicious dropping increasing a little bit.

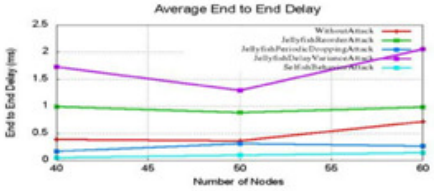


Fig. 17 Network Size vs End to End De lay

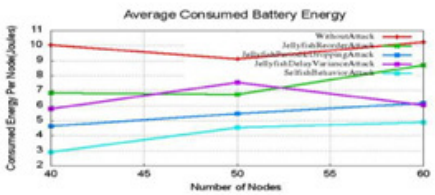


Fig. 18 Network Size vs Battery Energy

The following line graph in the figure 15 shows the impact of different attacks in terms of average achieved throughput of tcp ows. As shown in the line graph, Selfish Behavior Attack seems to be causing little bit higher im-pact than all other JellyfishAttacks. But without the presence of any attack AODV performs good and provided highest throughput. With respect to the increase of no of nodes in the network, the throughput decreases consider-ably. The following line graph in the figure 16 shows the impact of different attacks in terms of Packet Delivery Fraction (PDF) of tcp ows. As shown in the line graph, Selfish Behavior Attack seems to be causing little bit higher impact than all other JellyfishAttacks. But without the presence of any at-tack AODV performs good and provided highest PDF. With respect to the increase of no of nodes in the network, the performance getting decreased in most of the cases. In most of the earlier papers, the metrics EED and Energy consumption were not studied in detail or with proper comparison, because, the performance in terms of these two metrics will be somewhat strange to a researcher who always expect worst performance in the presence of an at-tack. Even in some previous papers, we may found some wrong interpretation for these graphs or even in correctly prepared graphs for these metrics. The following line graph in the figure 17 shows the impact of different attacks in terms of End to End Delay (EED) of tcp ows. With respect to the increase of no of nodes in the network, the performance getting decreased .As shown in the line graph, Selfish Behavior Attack seems to be providing lower EED normal AODV(without attack) but certainly it does not mean that SelfishBehavior Attack are improving the performance of the network. The low end to end delay under this two attacks are due to a strange fact that these two attacks makes disconnection in tcp ows and since the packets are not at all forwarded to any further nodes, indirectly it is reduce the message overhead in the network and reduced bandwidth usage otherwise it will be consumed by the forwarded data packets. So, the ows that were unaffected by Selfish Behavior Attack (the connections where there is no neighboring attack nodes) utilizes that extra bandwidth and gains some performance in term of some metrics.

Further, keep in mind that the end to end delay is only calculated based on the time in which a packet is sent and received. So if a packet is not received, in that case end to end delay can not be calculated. So this average EED is only the average EED of successfully delivered packets. The following line graph shows the impact of different attacks in terms of consumed battery energy. As shown in the line graph in the figure 18, in the presence of all the kinds of Attack the battery consumption is lesser than Normal AODV (without attack) but certainly it does not mean these Attacks are improving the performance in terms of energy consumption. The low energy consumption under attacks are due to a strange fact that these attacks makes disconnection in tcp ows and since the packets are not at all forwarded to any further nodes, indirectly it is reduce the battery consumption at the other nodes otherwise it will be con-sumed for forwarding the data packets. So, the nodes that were una ected by Attacks (where there is no neighboring attack nodes) preserves some battery power. Understanding this strange fact requires a better visualization of the whole network scenario. It is simple without any attack, AODV was able to send much and maximum nodes were able to participate in that communica-tion and utilized their energy for transmission/forwarding of packets so that the energy is consumed in most of the nodes. But in the presence of attack, the packets are getting dropped intermediately and the battery powers on other nodes that are not at all forwarding the packets gets preserved. With respect to the increase of no of nodes in the network, the performance seems to be getting decreasing.

Lot of previous papers saying that the attacks will increase energy consump-tion. Of course, it also may be true but not in the same sense. For example if an application will continuously try to send data under attack, then the bat-tery of the sending node and some other nodes between sender and attacker nodes will get reduced rapidly. If the application will vigorously try to do re-transmission due to loss, then this will increase the energy consumption. But under tcp, it will handle lossy scenario and just reduce the sending rate to avoid further losses. That is why the

average energy consumed in the network seems to be getting reduced under attack. Understanding this strange fact requires a better visualization of the whole network scenario.

## 6. CONCLUSION

In this work we evaluated the impacts of some of the popular attacks on a short term military rescue mission like MANET scenario. We did a compara-tive analysis of three kinds of JellyfishAttacks with SelfishBehavior Attack under AODV routing protocol and presented our ndings. We did that analy-sis with respect to different network sizes and under the presence of different number of attackers in the network. We did lot of simulation and analysis and arrived at signi cant and interpretable results. We measured the impact of the attacks with suitable metrics and explained the nature of different attacks in the previous chapter. With respect to the increase of malicious nodes in the network, the performance is getting decreased with respect to the most of the metrics that we considered. Further, with respect to the increase in number of nodes in the network, the performance is getting a ected with respect to the nature of attack. Without any doubts, all the attacks a ect the performance of MANET and the tcp ows are very much a ected by all these attacks. The main scope of this paper is to compare the Selfish Behavior Attack with different JellyfishAttacks. We successfully did that and the results are more interesting. According to our observations and the arrived results, the Selfish Behavior Attack is much worst than all types of JellyfishAttacks with respect to most of the metrics.

## REFERENCES


[1] Imrich Chlamtac, Marco Conti, Jennifer J.-N. Liu, Mobile ad hoc networking: imperatives and challenges. Ad Hoc Networks, 1(1),13-64 (2003).

[2] Abolhasan, Mehran, Tadeusz Wysocki, and Eryk Dutkiewicz,A review of routing proto-cols for mobile ad hoc networks, Ad hoc networks, 2.1,1-22 (2004).

[3] Royer, Elizabeth M., and Chai-Keong Toh,A review of current routing protocols for ad hoc mobile wireless networks,Personal Communications, IEEE 6.2 , 46-55(1999).

[4] Perkins, Charles, Elizabeth Belding-Royer, and Samir Das. Ad hoc on-demand distance vector (AODV) routing. No. RFC 3561(2003).

[5] Lego, Kapang, and Dipankar Sutradhar,Comparative Study of Adhoc Routing Protocol AODV, DSR and DSDV in Mobile Adhoc NETwork, 1, (2011).

[6] Kumar, Sathish Alampalayam, Classi cation and Review of Security Schemes in Mobile Computing, Wireless Sensor Network, 2.06,419, (2010).

[7] Wu, Bing, et al,A survey of attacks and countermeasures in mobile ad hoc networks, Wireless Network Security. Springer US,103-135 (2007).

[8] Deng, Hongmei, Wei Li, and Dharma P. Aggrawal,Routing security in wireless ad hoc networks,Communications Magazine, IEEE 40.10,70-75 (2002).

[9] Desilva, Saman, and Rajendra V. Boppana,Mitigating malicious control packet oods in ad hoc networks, Wireless Communications and Networking Conference, 2005 IEEE. Vol. 4. IEEE, 2005.

[10] Marti, Sergio, et al.,Mitigating routing misbehavior in mobile ad hoc networks, Pro-ceedings of the 6th annual international conference on Mobile computing and networking, ACM (2000).

[11] Kargl, F., Klenk, A., Schlott, S., Weber, M., Advanced detection of selfishor malicious nodes in ad hoc networks. In Security in Ad-hoc and Sensor Networks, (pp. 152-165). Springer Berlin Heidelberg(2005).

[12] Mitra, P., Mukherjee, S., A review of trust based secure routing protocols in MANETs, In Computing and Communication (IEMCON), 2015 International Conference and Work-shop on (pp. 1-7), IEEE(2015).

[13] Subramaniyan, S., Johnson, W.,Subramaniyan, K. (2014). A distributed framework for detecting selfishnodes in MANET using Record-and Trust-Based Detection (RTBD) tech-nique. EURASIP Journal on Wireless Communications and Networking,1, 1-10(2014).

[14] Begum, Syed Atiya, L.Mohan and B.Ranjiha, Techniques for resilience of denial of service attacks in mobile Adhoc networks, Proceedings published by International Journal of Electronics communication and Computer Engineering ,3.1(2012).

[15] Aad, Imad, Jean-Pierre Hubaux, and Edward W. Knightly,Denial of service resilience in ad hoc networks, Proceedings of the 10th annual international conference on Mobile computing and networking, ACM, (2004).
[16] Mandala, Satria etal,Investigating severity of blackhole attack and its varianace in Wire-less Mobile Adhoc networks, International journal of embedded systems, 7.3-4,296-305 (2015).
[17] Azer, Marianne A., and Noha Gamal El-din Saad, Prevention of multiple coordinated Jellyfishattacks in Mobile Adhoc Networks, International Journal of Computer Applica-tions, 120.20(2015).