ISSN:0975-9646

Handling multiple failures in IP Backbone network using LOLS along with AOMDV for detecting and avoiding wormhole attack

Anuja Sanjay Divekar[1] and Pankaj Chandre[2]

[1] PG Fellow, Department of Computer Network Engineering, Flora Institute of Technology ,Khopi,Pune,Maharashtra, India.

[2] Assistant Professor, Department of Computer Engineering, Flora Institute of Technology ,Khopi, Pune,Maharashtra, India

Abstract - *It has observed that IP networks are susceptible to many kinds of failures and attacks* but there are many solutions to provide another network or network path by rerouting method. But most of the solutions can handle only single failure and drops the packet or causes the loops. So we propose *Localized On-demand Link State* (LOLS) routing protocol. LOLS can surely handles the packet forwarding till the destination even in presence of multiple links. As LOLS can send a packet even with multiple failures but it fails when there is an presence of any kind of network attack. *Among the various attacks possible in IP networks wormhole attack is one which is treated as a very severe attack. LOLS can handle multiple failures but it cant detect the network attacks. As LOLS is unable to handle any kind of attack* and this is the reason why we are working on LOLS as well as AOMDV algorithm. *In Wormhole attack a harmful node records packets at one end in the network and tunnels them to another harmful node which is present in the other end of the network. In this paper, we have proposed an algorithm which detects and avoids the wormhole attack while data transfer. In AOMDV ,one mechanism is used, which is based on the total round trip time (RTT) of current route and the average round trip times. This mechanism works for both mobile ad hoc networks and wireless ad hoc network.* LOLS and AOMDV is going to hold the quality of LOLS of loop free forwarding even with wormhole attack.

***Keyword*-Failure Resilience, Local Rerouting ,Fast Reroute, Wormhole attack, AOMDV , harmful node , RTT**

1. INTRODUCTION

In our lives the internet plays an important role. Providing non-stop service availability even with attacks is the primary challenge for the service providers. Unfortunately, even in well managed networks, service disturbances occur due to attack on link or node failure or both. To overcome with these problems in today's Internet world, these networks need to handle multiple failures as well as network attack(Wormhole attack). Hence, it is important to formulate schemes that protect the network not only multiple failures but also can handle wormhole attack in IP network[2]. Therefore, providing uninterrupted service availability even in the presence of multiple failures and attacks is a major challenge for service providers. Hence, it is important to implement project with schemes that protect the network against not only single failures but also *multiple independent failures.*

The essential concept behind LOLS(Localized On-demand Link State Routing) is to have packets transmit a blacklist of degraded links come across along the path that are to be avoided in order to ensure loop-free forwarding. Forward progress is towards destination, packet's blacklist is reseted, limiting the spread of failure information to just a few hops. LOLS considers a link as degraded if its current state (say "down") is worse than its globally advertised state (say "up"). Under LOLS, each packet carries a blacklist (a minimal set of degraded links come across along its path), and by excluding the blacklisted links, the next hop is determined[8]. A packet's blacklist is initially void and remains blank **when there is no disagreement between the current and the advertised states of links *along its path.*** But when a packet reaches at a node with a tainted link neighboring to its next hop, that link is added to the packet's blacklist. Then the packet is advanced to an alternate next hop. When the next hop makes forward progress, the packet's blacklist is return to empty i.e., the next hop has a smaller path to the destination than any of the nodes navigated by the packet. With these simple steps, LOLS propagates the state of degraded links only when essential, and as far as necessary, and confirms loop-free delivery to all local destinations.

Through LOLS handles multiple failures but it cant handle network attacks, so we are implementing for the same. In this paper we are focusing on a particular kind of attack called wormhole attack which is considered as a severe attack.

For Wormhole attack detection and avoidance we will be implementing AOMDV algorithm [2]. In the wormhole attack, an attacker receives packets at one point in the network, tunnels them to another point in the network, and then replays them into the network from that point. When node is harmful node(attack by wormhole) then the node as a source node and they send to destination ,to handle this problem we use AOMDV algorithm**.**

We provide the details of this integration in next sections. The rest of the paper is structured as follows. Section II presents LOLS approach for handling multilink failures by greedy forwarding and blacklist based forwarding mechanism and Section III explains wormhole attack detection and avoidance by AOMDV algorithm. In Section IV, reports the results of the performance evaluation. and finally in Section V, we conclude the paper.

2. RELATED WORK

Numbers of methods have been implemented in the past to make networks more robust to failures. To overcome from single or multiple failures various methods have been implemented. The Not-via method [6] is used for packet redirection if packet fails . Failure Carrying Packets[1] implemented when multiple independent failures are present in network.

There have been several fast reroute proposals for handling transient failures in IP networks [5] by having the adjacent nodes perform local rerouting without notifying the whole network about a failure. However, most of these schemes are designed to deal with single or correlated failures only. Recently, proposed an approach to handle dual link, but only single node failures. On the other hand, failure carrying packets (FCP)[9] and packet recycle (PR) try to forward packets to reachable destinations even in case of arbitrary number of failures. The drawbacks, however, are that FCP carries failure information in each packet all the way to the destination whereas PR forwards packets along long detours. So , Authors Glenn Robertson, Srihari Nelakuditi proposed a scalable Localized On-demand Link State (LOLS) routing for protection against multiple failures.

V. Karthik Raju and K. Vinay Kumar have proposed an algorithm [2] which detects and avoids the wormhole attack in the routing phase itself. Their mechanism is based on the total round trip time (RTT) of the established route and the average round trip times of the sender one hop neighbors, which is considered as maximum one hop round trip time.

3. PROPOSED SYSTEM

3.1 Problem Definition:

Localized On-Demand Link State routing is proposed to handle multiple link failures. So that packet is forwarded to specified destination even if the link is failed. Fig. 1 shows the detailed architecture of proposed system. Blacklist is prepared after the forward progress which contains degraded links but it does not contain all such links which are failed for less duration less than the threshold.

The problem definition is to handling multiple link failures in IP network using Localized On-demand Link state routing by avoiding wormhole attack . A technique to identify wormhole attacks in network and a solution to discover a safe route avoiding wormhole attack. The problem with the LOLS is that, it does not detect and avoid any kind of attack at the time of data transfer to destination. The issue is handled by using simple and efficient mechanism i.e. by applying AOMDV algorithm to detect and avoid wormhole attack.**[2]** Here nodes are get created using JAVA programming and enable user of the system to send and receive the packets. To perform the forwarding of the packets, greedy forwarding algorithm is applied. While forwarding the packets using greedy forwarding, if the link gets degraded ,then apply the blacklist based forwarding algorithm.[8] convergence for this reason the use of AOMDV algorithm is implemented to avoid and detect wormhole attack. By this mechanism, the network will be sustainable for short-lived multiple failures without wormhole attack. This technique can be enhanced for the MANET system. The flow of the activities performed during packet transfer is shown in the to be performed is shown in the

Fig. 2.

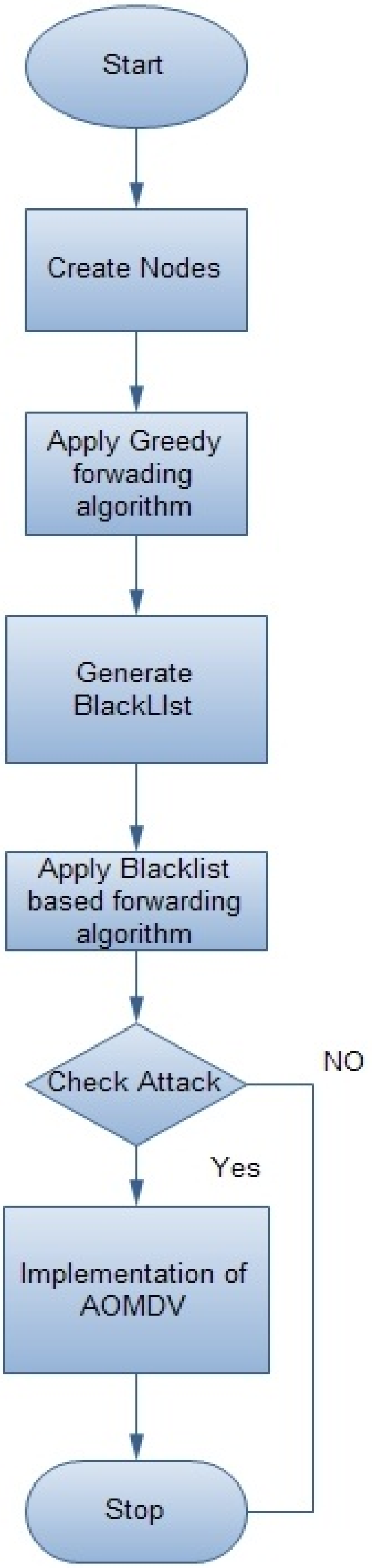

Fig. 2: Flow of activities to be performed while packet transfer

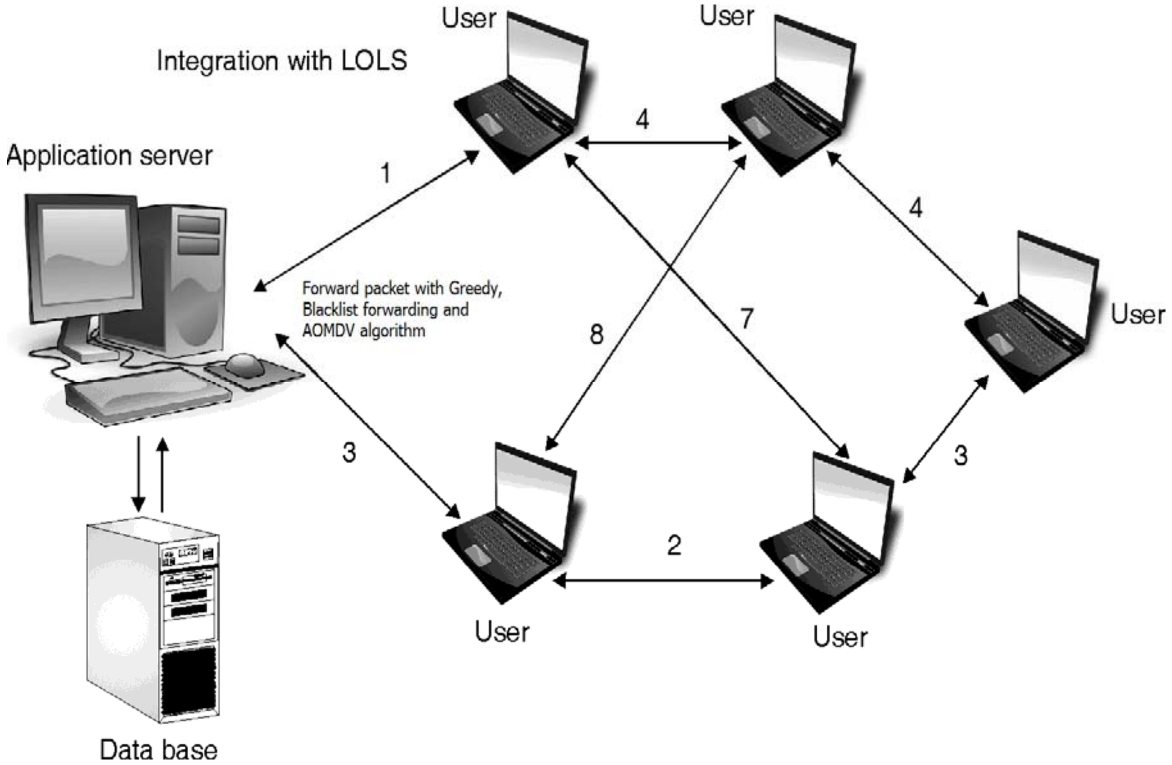

Fig. 1: Proposed System Architecture

A. *Greedy Forwarding algorithm [4]:*

Selection of a succeeding hops such that the packet does not get trapped in a forwarding loop. An approach to assure loop-freedom is to apply greedy forwarding that forwards the packet along a route with minimum cost to the destination, i.e., every hop makes forward progress in the direction of the destination. It is crucial that based on the broadcasted topology the path cost is determined regularly at all nodes and costs are advertised. A packet is usually advanced in greedy mode to a succeeding hop along the path with reducing cost (w.r.t. the announced topology) to the endpoint. When a packet come across a dead-end in greedy mode, it is advanced in recovery mode instead of dropping the packet. Packets carry a blacklist in recovery mode, which is a set of degraded links come across the route. A packet's subsequent hop is selected along a path that does not contain blacklisted links. The forwarding of a packet is swapped back to greedy mode, i.e., the blacklist is returned to empty, when it reaches at a node with lower cost (w.r.to the announced topology) to the destination than the node at which it moved in the recovery mode. Thus, LOLS successfully transmits link state on demand, and only to as several nodes as essential.

Algorithm:

Let d be the destination of the packet, j is the adjacent next feasible hop, i is the source node from which the route is to be calculated.

1. If the cost of path from j to d is less than cost of path from i to d then, j is the next feasible hop for which node i has shortest path to destination d.
2. If no feasible hop is present then algorithm returns NULL and the packet is rejected. We want to point out that this algorithm is a variation of standard greedy forwarding as it does not continuously select a next hop with maximum forward advancement. Instead, it chooses a next hop such that it aggregates to shortest path forwarding when there are no down links, which is surely a desired.

B. *Blacklist based forwarding algorithm[8]:*

By this algorithm , every packet p carries a blacklist p.blist with it in its header while travelling through the network, and packet is to destination or next hops based on both p.dest and p.blist. The blist that is blacklist is initialized to NULL at the source and it is increases or shrinks as and when required during the whole forwarding process.

Algorithm:

1. Find the next hop with smallest path cost and which does not have links present in packet's blacklist.
2. If the links to the neighbor are down or degraded, add these links in the packet's blacklist.
3. Repeat the steps 1 and 2 until either we find the next hop which forwards the packet to the next hop and resets the packet's blacklist, or there is no feasible next hop this means that the destination is unreachable and the packet must be dropped. There are rules present for updating the packet's blacklist p.blist at node i, the rules are briefed here.

1. The link from i to j is added to blacklist if:
   a. Link is degraded
   b. No feasible next hop is present without the link i to j
   c. If the link i to j had not been down, then this link could have been the shortest path.
2. The blacklist is retuned to NULL when: a. The feasible next hop is present b. The cost from j to destination is less than that of any other node traversed by packet p.

D. Wormhole Attack by AOMDV algorithm:

In proposed system we are handling multiple failures even in presence of attack. As wormhole attack is very severe attack in all attack so we concentrate on wormhole attack in implementation also. In this attack the two harmful nodes resides in the two ends of the network and they form a link between them using an out-of-band hidden channel like wired link, packet encapsulation or high power radio transmission range[2]. After they form a tunnel between them as shown in Fig. 3, whenever a harmful node receives packets it tunnels them to the other harmful node and in turn it broadcasts the packet there. Since the packet is travelling through the tunnel it reaches the destination speeder than other route and moreover the hop count through this path is going to be less so this path is established between the source and the destination. Once the path is established between the source and the destination through wormhole link they can behave badly in many ways in the network like continuously dropping the packets, selective dropping the packets, analysing the traffic and performing Denial of Service attack. Wormhole attacks are divided into two types based on the behaviour of the harmful nodes; they are hidden attacks and exposed attacks. In the former one the malicious nodes do not update the packet header with their identities like MAC address, this keeps the harmful nodes invisible to the outside world but where as in the later one the harmful nodes update the packet header with their identities this makes them look like normal nodes in the network.

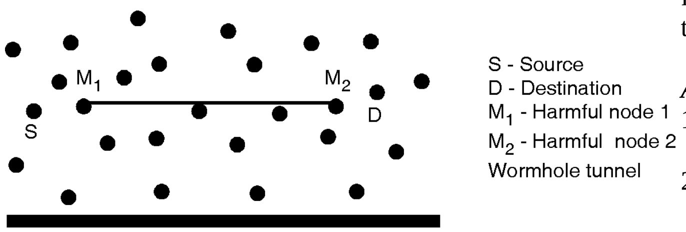

Fig.3: IP network with wormhole link.[2]

In the proposed mechanism the detection of wormhole attack is done in the routing phase itself, in the following way.

When the source node broadcasts a RREQ packet note the time (t1) and when the corresponding RREP packet is received by the source, again note the received time of the packet, If there are multiple RREP packets received, that means there are more than one route available to the destination node then note the corresponding times (t2_i) of each RREP packet. By using the above two values one can calculate the total round trip time (t3_i) of the established route or routes. In the next step calculate the round trip times for all the one hop neighbors of the source, for this first fetch all the neighbors of the source node from the neighbor list of the source then broadcast hello packets to all the neighbors of the source. While broadcasting the hello packet note the time (ts) and similarly while receiving the reply for hello packets note the times (tr_i) for corresponding hello packets, from this calculate the round trip times (trt_i) taken for each hello packet to travel from source node to neighbour nodes and back to source node. Now as noted in the above step, calculate the average of all the times. Since all the round trip times of the one hop neighbors of the source are considered, by averaging them one can get the exact time taken for a packet to travel one hop distance, this time is considered as the maximum time taken for a packet to travel one hop distance and this time is noted as the maximum round trip time (tmax) for one hop. Now multiply the maximum round trip time (tmax) with the hop count (h) of the established route, this gives the maximum time taken for a packet to travel along the established route, this is considered as estimated round trip time (te). Now compare the total round trip time (t3_i) with the estimated round trip time (te), there is no wormhole link present in the established route if the total round trip time (t3_i) is less than or equal to estimated round trip time (te) and one can continue with that route else wormhole link is present in that route. Since wormhole link is detected in that route, that route is no more used and it is blocked and that route is kept in the blocking list at the source node. So that, from next time onwards whenever a source node needs a route to that destination, first it checks in the routing table in the route establishment phase for a route and it will come to know that that route is having wormhole link and it will not take that route instead it will take another route from the routing list of the source node which is free from wormhole link if available. The proposed mechanism does not require any additional hardware and also has less overhead. Since AOMDV routing protocol is used the end to end delay is less because even when a route is failed another route is fetched immediately from the routing table and the time taken for route establishment is saved by this.

Algorithm:

1. When the source node broadcasts RREQ packet note the time t1.
2. For each RREP packet received by the source node a. Note the time t2_i b. Calculate the round trip time for all routes using this formula t3_i = t2_i - t1.
3. End For
4. Fetch the neighbors from the neighbor list.
5. Broadcast the hello packet to neighbors of the source node and note the time ts
6. For each hello packet received by the source node. a. Note the time tr_i b. Calculate the round trip times (trt) using this formula trt_i=tr_i – ts
7. End For
8. Calculate the average round trip time for one hop neighbors, from the round trip times taken in the step 6.
9. Note this time as the Maximum round trip time (tmax) for one hop distance.
10. Fetch the hop count (h)
11. Calculate the estimated round trip time (te) using this formula te= tmax * h
12. If (t3_i <= te) then a. No wormhole link is present in that route b. Continue with that route
13. Else a. Wormhole link is present in that route b. Block that route and update it in the routing table c. Fetch another route from the routing table ri d. If (route is present && not in the wormhole blocked list) perform the process from step 10 Else Stop End If End If.

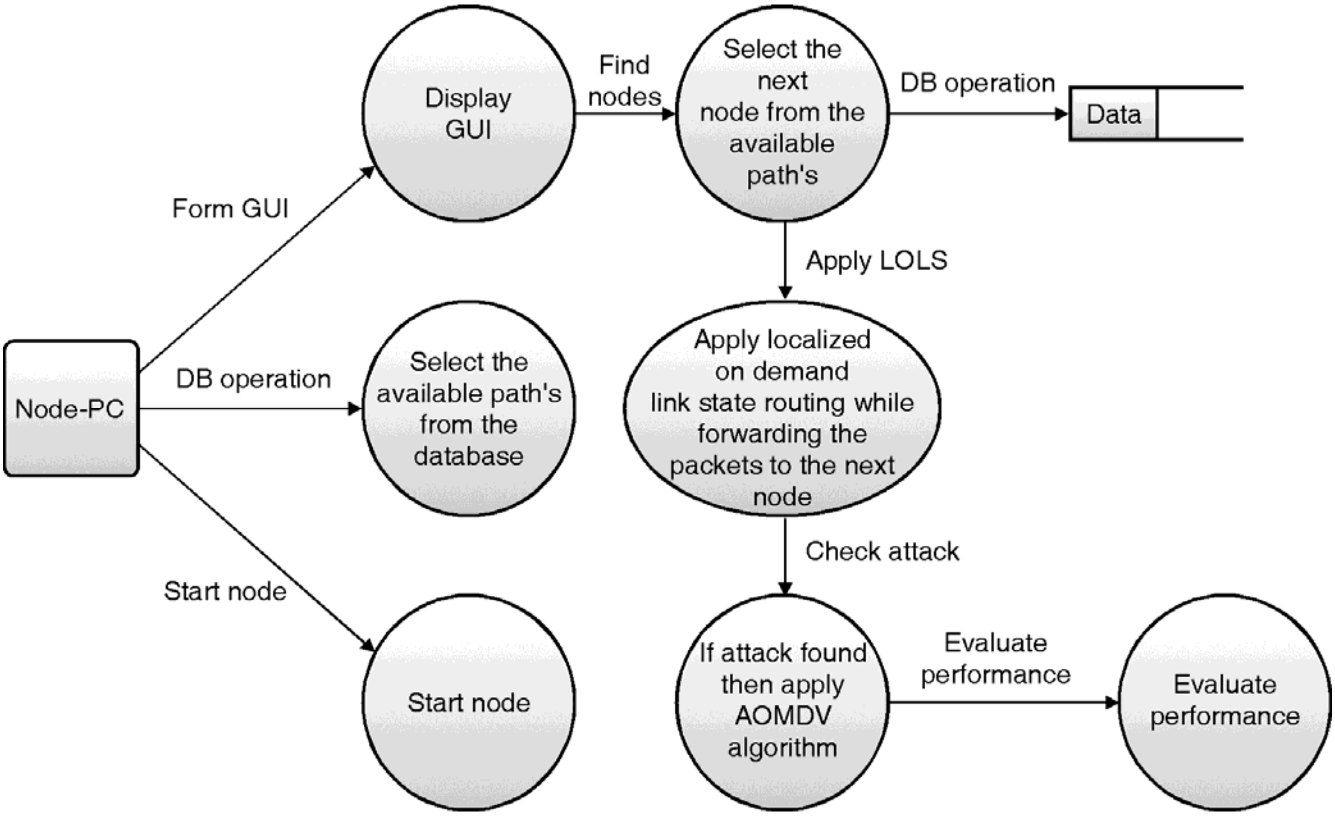
Node-PC
Form GUI
DB operation
Start node
Display GUI
Select the available path's from the database
Start node
Find nodes
Select the next node from the available path's
DB operation
Data
Apply LOLS
Apply localized on demand link state routing while forwarding the packets to the next node
Check attack
If attack found then apply AOMDV algorithm
Evaluate performance
Evaluate performance

Fig. 4: Data Flow Diagram

3.2 Mathematical Model

U is main set of users like u1, u2, u3….
U = {u1, u2, u3…….}
A is main set of Administrators like a1, a2, a3….
A = {a1, a2, a3…….}
P is main set of participating paths like p1, p2, p3…
P = {p1, p2, p3…….}
Identify the processes as P.
P = {Set of processes}
P = {P1, P2, P3……} & P1 = {e1, e2, e3, e4,e5}
Where
{e1=Find the nodes in the network}
{e2=Provide the weights to the each links in the network}
{e3=Perform Greedy forwarding algorithm}
{e4= generate blacklist for each packet and apply blacklist based forwarding algorithm}
{e5= Apply AOMDV algorithm if Wormhole attack is detected}

3.3 Data Independence and Data Flow Architecture

Data Flow Architecture represents flow of information and function flow during the execution and it is represented in Fig. 4.

3.4 Platform

The project is to be developed in JAVA using eclipse IDE, RMI and JAVAFx shall be also used. And MYSQL for storing the content of blacklist.

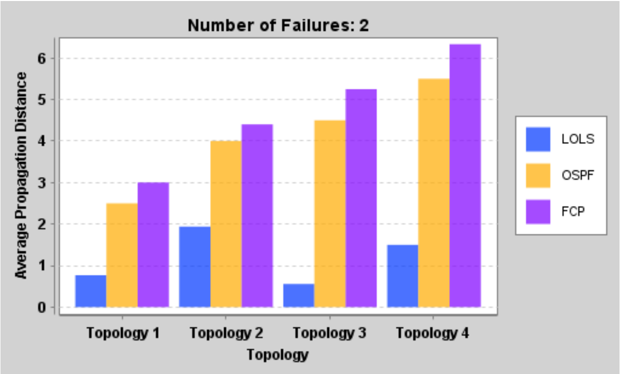

Fig.5(a) : Average Propagation Distance under LOLS,OSPF and FCP

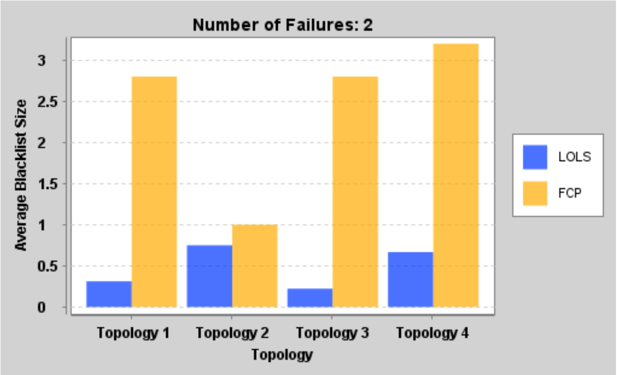

Fig.5(b) : Average blacklist size in packet under LOLS and FCP

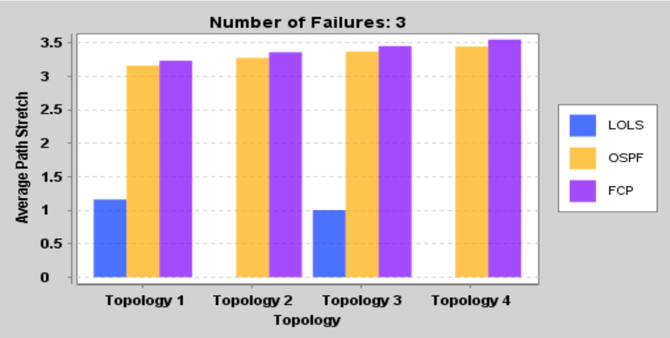

Fig.5(c) : Average path stretch under LOLS, OSPF and FCP

Fig. 5 : Results

3.5 Result

LOLS cannot guarantee the delivery of packet in presence of attack , it can handle multiple failures in IP network. To detect and avoid wormhole attack during data transfer AOMDV algorithm is used. Due to this simple and efficient mechanism , the network will be sustainable for multiple failures even in presence of severe attack. Fig 5 shows the results. Fig. 5(a) shows the average failure propagation distance of LOLS in comparison with FCP and OSPF. Fig. 5(b) shows the blacklist size of LOLS in comparison with FCP. Fig. 5(c) shows average path stretch of LOLS , OSPF and FCP.

4. CONCLUSION AND FUTURE SCOPE

In this paper, we presented an idea of LOLS and AOMDV algorithm, for handling multiple failures in IP backbone networks even in the presence of Wormhole attack. Blacklist is generated at every node which contains degraded links. Degraded links are excluded while packet transfer so even there are multiple failed links in network packet will get forwarded to destination. After the forward progress packet's blacklist is rested and this is one of the main feature of LOLS.

So failure information is forwarded to limited nodes. LOLS can handle multiple link failure but it cannot detect and avoid any type of network attack for this reason we use AOMDV algorithm. By applying these two different algorithms network will sustainable for multiple failures even in presence of attack. These two algorithms can be enhanced for MANET system.

REFERENCES

[1] Glenn Robertson and Srihari Nelakuditi "Handling Multiple Failures in IP Networks through Localized On-Demand Link State Routing" in IEEE TRANSACTIONS ON NETWORK AND SERVICE MANAGEMENT, VOL. 9, NO. 3, SEPTEMBER 2012.

[2] Marianne A. Azer, Sherif M. El-Kassas, Abdel Wahab F.Hassan, Magdy S. El-Soudani "Intrusion Detection for Wormhole Attacks in Ad hoc Networks a Survey and a Proposed Decentralized Scheme" IEEE 2008.

[3] Rama Gaikwad and S. P. Pingat" Idea of handling short-lived network failures using LOLS and loop free convergence using FCFR" at al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (2) , 2014, 2099-2102.

[4] B. Karp and H. T. Kung, "GPSR: greedy perimeter stateless routing for wireless networks," in Proc. 2000 ACM Mobicom, pp. 243– 254. Available:citeseer.ist.psu.edu/karp00gpsr.html

[5] K. Lakshminarayanan, M. Caesar, M. Rangan, T. Anderson, S. Shenker, and I. Stoica, "Achieving convergence-free routing using failure-carrying packets," in Proc. 2007 SIGCOMM, pp. 241–252.

[6] S. I. et al., "An approach to alleviate link overload as observed on an IP backbone," in Proc. 2003 IEEE Infocom.

[7] P. Francois, C. Filsfils, J. Evans, and O. Bonaventure, "Achieving subsecond IGP convergence in large IP networks," in ACM SIGCOMM Computer Commun. Rev., July 2005.

[8] S. N. et al, "Blacklist-aided forwarding in static multihop wireless networks," in 2005 SECON.

[9] K. Lakshminarayanan, M. Caesar, M. Rangan, T. Anderson, S. Shenker, and I. Stoica, "Achieving convergence-free routing using failure-carrying packets," in Proc. 2007 SIGCOMM, pp. 241–252.