ISSN 2277-3061

A Simple and Efficient Mechanism to Detect and Avoid Wormhole Attacks in IP network through LOLS

Anuja Divekar, Deepali Borade, Vivek Bugad

PG Fellow, Department of Computer Network Engineering, Flora Institute of Technology, Pune, Maharashtra, India.

anujadivekar13@gmail.com

Assistant Professor, Department of Computer Engineering, Flora Institute of Technology, Pune, Maharashtra, India

borade.deepali21@gmail.com

PG Fellow, Department of Electronics –VLSI Design, Bharati Vidyapeeth's College of Engineering ,Pune, Maharashtra, India.

vivekbugad@gmail.com

ABSTRACT

It has been observed that IP networks are vulnerable to many kinds of attacks. Among the various attacks possible in IP networks wormhole attack is one which is treated as a very severe attack. In LOLS, multiple failures are handled but network attack is not detected. LOLS cannot handle any kind of attack and this is the reason why we are working on wormhole attack and its detection by using AOMDV algorithm. In this 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 this paper 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 networks.

Indexing terms/Keywords

Fast Reroute; Failure Resilience; Local Rerouting; Wormhole attack; RTT; AOMDV; harmful node.

ISSN 2277-3061

I. INTRODUCTION

The internet plays an important role in our lives these days. Providing nonstop service availability even with attacks is the foremost challenge for the service providers. Unfortunately, service disturbances occur even in well managed networks due to attack on link or node failure or both. To upkeep evolving time-sensitive requests in today's Internet, these networks need to endure attacks. Hence, it is important to formulate schemes that protect the network not only multiple failures but also can handle wormhole attack in IP network**[3]**. 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 done 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 the next hop is determined by excluding the blacklisted links**[1]**. 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. The packet is then advanced to an alternate next hop. The packet's blacklist is return to empty when the next hop makes forward progress, 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 network attacks are not handled yet so we are implementing for the same**. In this We will be implementing AOMDV algorithm for wormhole attack detection in the developed system.[4]** 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.

II. RELATED WORK

Number of methods have been implemented in the past to make networks more robust to failures. Some of methods can deal with Localized Link State Updates , Forwarding over Diverse Set of Paths , Geographic Position based Routing , Protection Routing with Traffic Engineering . Here we describe a some systems fitting to each of these methods in the following Sections:

Localized Link State Updates: To make link state routing scalable for mobile ad hoc networks, limited dissemination based schemes have been proposed **[5]**. Fisheye state routing (FSR) **[6]** and hazy sighted link state (HSLS) **[5]** routing schemes update the nearby nodes at a higher frequency than the remote nodes that lie outside a certain scope. The drawback is that the chosen scope can be more than sufficient in some cases and less than necessary in other cases resulting in needless updates or forwarding loops. LOLS can be considered a form of limited dissemination based routing scheme that ensures loop-free forwarding while notifying only a small subset of nodes in the vicinity of a failure.

Forwarding over Diverse Set of Paths: RON **[7]** is a pioneering approach for recovering from path outages using a resilient overlay network. RON is primarily an attempt to overcome the slow convergence of BGP. A set of routing deflection rules are developed in that enable routers to independently deflect packets and thereby collectively construct a diverse set of paths. This approach amounts to a form of implicit source routing in which end-systems set a four byte tag in the packet header to select non-shortest path routes. Path splicing, which is proposed in **[8]**, perturbs link weights to produce multiple routing trees and allows traffic to switch trees at any hop route to the destination. While the path splicing can sustain connectivity in the face of multiple link and node failures, there is a small probability of forwarding loops and the number of splicing header bits needed is proportional to the number of hops. In contrast, LOLS is an intra-domain approach that is loop-free with less than one byte of overhead per packet and it deviates from shortest paths only near a failure.

Geographic Position based Routing: Greedy Perimeter Stateless Routing (GPSR) and similar schemes **[9]**, based on geographical positions forward in greedy mode, where each hop takes the packet closer (in terms of Euclidean distance) to the destination. However, when the packet reaches a deadend, i.e., when the forwarding node is closer to the destination than any of its adjacent nodes, the forwarding is switched to face mode. In face mode, a packet is forwarded along the boundaries of a planarized subgraph. A packet is forwarded back in greedy mode when it reaches a node closer to the destination that the node at which it entered the face mode. While position based routing is highly scalable, it yields suboptimal paths compared to topology-based routing. We will see in the following section that LOLS is similar in spirit to position-based routing and can be thought of as switching between greedy and recovery modes during a packet's flight .

Protection Routing with Traffic Engineering: Various traffic engineering approaches have been proposed in the past to distribute load across a network [**10]**. One of the recent proposals attempts to assign link weights such that the load remains balanced even after a failure, but it does not ensure forwarding continuity during convergence. Towards providing protection and balancing traffic, a version of MRC **[11]** has been suggested that optimizes link weights for the backup configurations based on the observed traffic patterns. However, this method does not scale well for all possible link failures, and does not address multiple failure scenarios. proposes an efficient algorithm for computing backup paths where spare capacity is shared between different backup paths but it focuses only on single link or node failures. Recently, a centralized routing approach has been proposed that incorporates both protection routing and traffic engineering. While this approach has a great feature of trading off protection and performance, it cannot guarantee 100% coverage even for single failures. Our focus is on ensuring guaranteed forwarding to all reachable destinations despite multiple failures by detecting wormhole attack, which in itself is a challenging problem even without traffic engineering..

ISSN 2277-3061

III. PROPOSED SYSTEM

3.1 Problem Description :

Using LOLS, the multiple link failure can be handled. This makes sure that the data is delivered to destination even if the link is failed. Figure 1 shows the detailed architecture of our system. Moreover, it does not transmit the facts of link failure to all the destinations if the link is failed for the 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 . We present 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.**[4]**

Here we will be creating nodes using JAVA programming and enable user of the system to send and receive the packets. To perform the forwarding of the packets, we shall apply greedy forwarding algorithm. While forwarding the packets using greedy forwarding, if we will come across the down link, we will apply the blacklist based forwarding algorithm.**[1]**

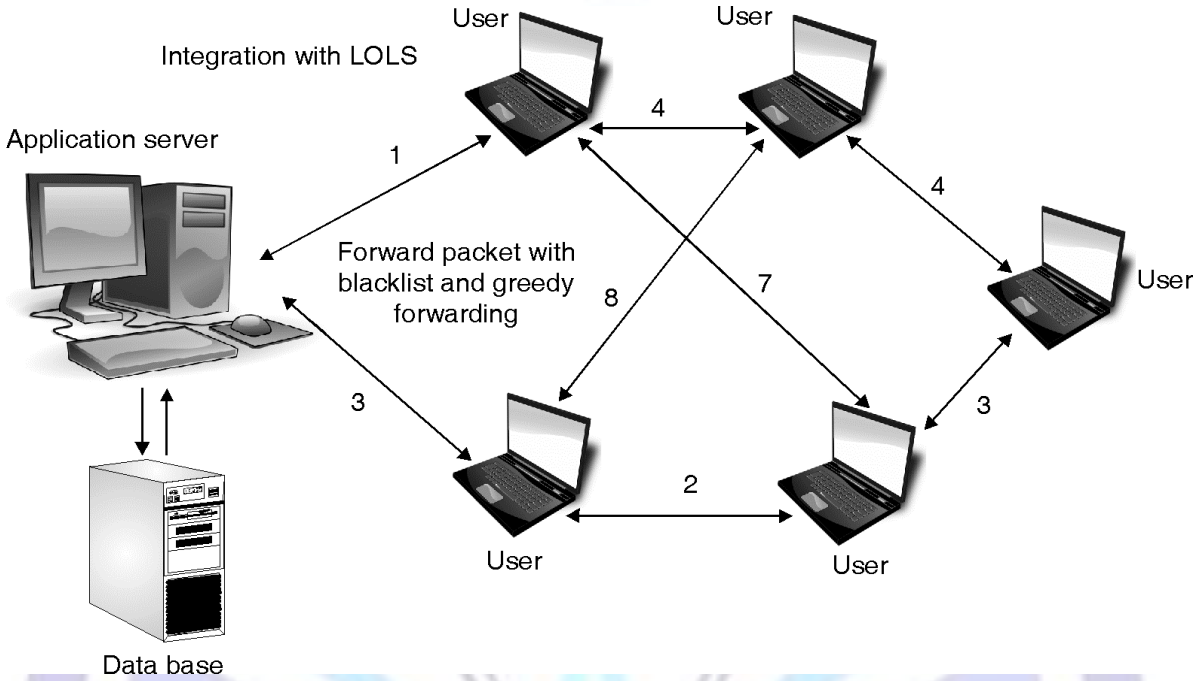


Figure 1 : System Architecture

3.2 Greedy Forwarding algorithm [12] :

We need to select a succeeding hop 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 reducing cost to the destination, i.e., every hop makes forward progress in the direction of the destination. It is crucial that the path cost is

determined regularly at all nodes based on the broadcasted topology. 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(whose cost to the destination is lesser than any of the potential subsequent hops) in greedy mode, instead of dropping the packet, it is advanced in recovery mode. In recovery mode, packets carry a blacklist, 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 retuned to empty, when it reaches at a node with lower cost (w.r.t.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.

3.2.1 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.

3.3 Blacklist based forwarding algorithm[13]:

In Localized On-demand Link State routing, every packet p carries a blacklist p.blist with it while traveling through the network in its header, 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.

3.3.1 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.

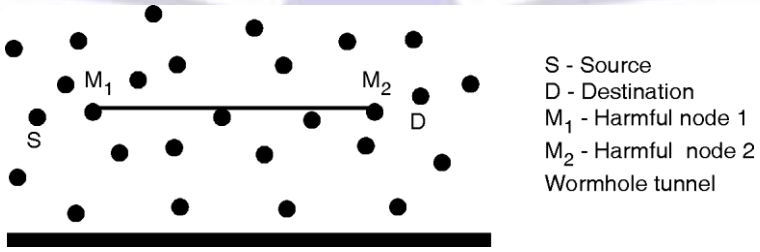
2. The blacklist is retuned to NULL when

a. The feasible next of is present

b. The cost from j to destination is less than that of any other node traversed by packet p.

3.4. Wormhole Attack

In this paper we are focusing on a particular kind of attack called wormhole attack which is considered as a severe attack in IP network. Minimum two harmful nodes are required to perform this attack; more than two harmful nodes are also used to perform this attack. 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[**14]**. After they form a tunnel between them as shown in Figure 2, 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 **[15]**. Once the path is established between the source and the destination through wormhole link they can misbehave in many ways in the network like continuously dropping the packets, selective dropping the packets, analyzing 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.



S - Source
D - Destination
M1- Harmful node 1
M2 - Harmful node 2
Wormhole tunnel

Figure 2: IP network with wormhole link.[4]

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 neighbor nodes and back to source node. Now calculate the average of all the times that noted in the above step. 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), if the total round trip time (t3_i) is less than or equal to estimated round trip time (te) then there is no wormhole link present in the established route 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 has less overhead and also does not require any additional hardware. 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.

3.4.1 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

3.5 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= Integrate system with FCFR} [2]

3.6 Data Independence and Data Flow Architecture

A diagram showing the course of information through the function and the change undertakes is presented in Figure 3

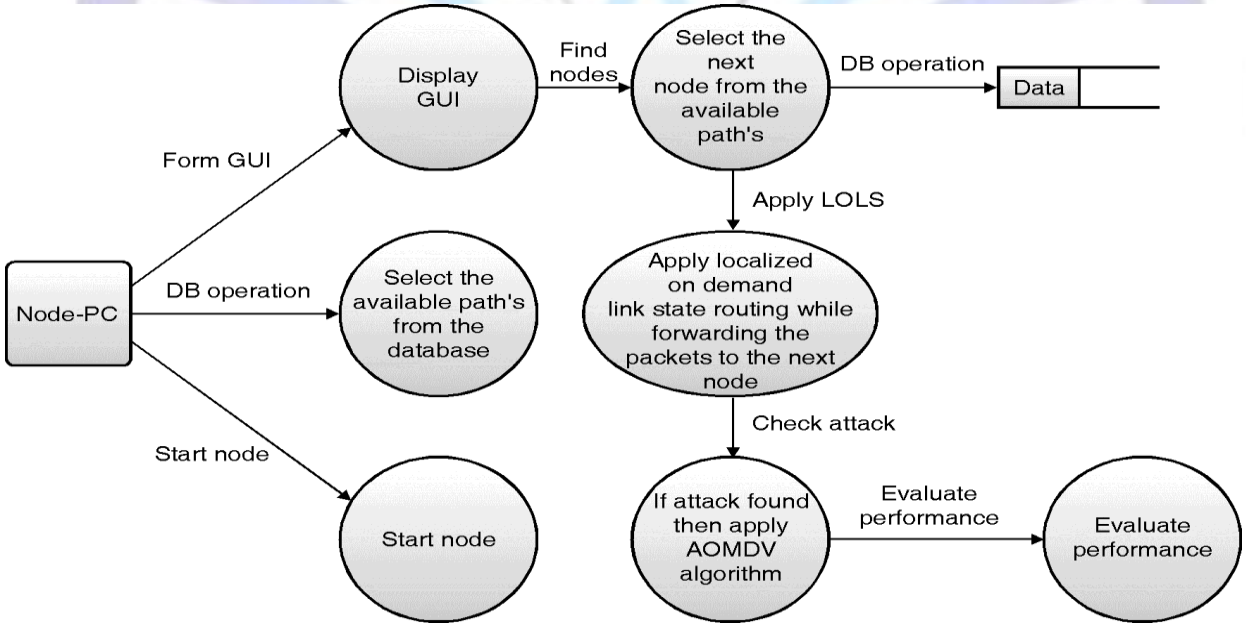


Figure 3: Data Flow Architecture

3.7 Platform

The project is to be developed in JAVA using eclipse IDE,RMI and JAVAFx shall be also used.

3.8 Expected Result

LOLS cannot detect and avoid wormhole attack during data transfer. By applying AOMDV algorithm attack is detected. Due to this simple and efficient mechanism , the network will be sustainable for short-lived multiple failures and will be

attack free in case of convergence. The total propagation distance of failures in case of LOLs is found less than that of the other multiple failure handling techniques.[2].

IV. CONCLUSION AND FUTURE SCOPE

In this paper, we presented an idea of LOLS integrating, for controlling multiple failures in IP backbone networks and simple and efficient mechanism to detect and avoid wormhole attack by using AOMDV algorithm. The fundamental notion behind LOLS is to have packets carry a blacklist of degraded links came across the path that are to be escaped in order to guarantee loop-free forwarding. The significant feature of LOLS is that a packet's blacklist is reset to null as soon as it makes forward movement in the direction of the destination, restricting the propagation of failure information to limited hops. LOLS cannot detect any kind of attack during route convergence for this reason the use of AOMDV algorithm is done 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

V. 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]Glenn Robertson, James Bedenbaugh, SrihariNelakuditi "Fast Convergence with Fast Reroute in IP Networks" in Proc. IEEE Infocom, Mar. 2007.

[3]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.

[4] V. Karthik Raju and K. Vinay Kumar,"A Simple and Efficient Mechanism to Detect and Avoid Wormhole Attacks In Mobile Ad Hoc Networks" in 2012 International Conference on Computing Sciences , 978-0-7695-4817-3/12 $26.00 © 2012 IEEE

[5] C. Santivanez, R. Ramanathan, and I. Stavrakakis, "Making link-state routing scale for ad hoc networks," in Proc. 2001 ACM MobiHOC.

[6] M. Gerla, X. Hong, and G. Pei, "Fisheye state routing protocol for ad hoc networks," IETF Internet Draft, June 2002, draft-ietf-manet-fsr-03.txt.

[7] D. Anderson, H. Balakrishnan, F. Kaashoek, and R. Morris, "Resilient overlay networks," in 2001 SOSP.

[8] M. Motiwala, N. Feamster, and S. Vempala, "Path splicing: reliable connectivity with rapid recovery," in 2008 SIGCOMM.

[9] 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

[10] N. Wang, K. H. Ho, G. Pavlou, and M. Howarath, "An overview of routing optimization for Internet traffic engineering," IEEE Commun. Surveys & Tutorials, vol. 10, no. 1, pp. 36–56, 2008.

[11] A. Kvalbein, T. Cicic, and S. Gjessing, "Post-failure routing performance with multiple routing configurations," in 2007 IEEE INFOCOM.

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

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

[14] Azer, M.A., El-Kassas S.M., Hassan, A.W.F., El-Soudani M.S., "Intrusion Detection for Wormhole Attacks in Ad hoc Networks a Survey and a proposed Decentralized Scheme Marianne " IEEE Third International conference on Availability, Reliability and Security, 2008.

[15]Reshmi Maulik, Nabendu Chaki "A comprehensive review on wormhole attacks in MANET" International Conference on Computer Information Systems and Industrail Management Applications(CISIM) 2010.