Tolerance of Flooding Attacks in Delay Tolerant Networks

Submitted By

Maitri Shah

16mcec21

DEPARTMENT OF COMPUTER ENGINEERING

INSTITUTE OF TECHNOLOGY

NIRMA UNIVERSITY

AHMEDABAD-382481

May 2018

Tolerance of Flooding Attacks in Delay Tolerant Networks

Thesis

Submitted in partial fulfillment of the requirements

for the degree of

Master of Technology in Computer Engineering

Submitted By

Maitri Shah

(16mcec21)

Guided By

Prof. Pimal S Khanpara

DEPARTMENT OF COMPUTER ENGINEERING

INSTITUTE OF TECHNOLOGY

NIRMA UNIVERSITY

AHMEDABAD-382481

May 2018

Certificate

This is to certify that the thesis entitled **"Tolerance of Flooding Attacks in Delay Tolerant Networks"** submitted by **Maitri Shah (Roll No:16mcec21)**, towards the partial fulfillment of the requirements for the award of degree of Master of Technology in Computer Engineering of Nirma University, Ahmedabad, is the record of work carried out by her under my supervision and guidance. In my opinion, the submitted work has reached a level required for being accepted for examination. The results embodied in this thesis, to the best of my knowledge, haven't been submitted to any other university or institution for award of any degree or diploma.

Prof. Pimal S Khanpara
Guide & Assistant Professor,
Computer Engineering Department,
Institute of Technology,
Nirma University, Ahmedabad.

Dr. Priyanka Sharma
Professor,
Coordinator M.Tech - CSE ,
Institute of Technology,
Nirma University, Ahmedabad

Dr. Sanjay Garg
Professor and Head,
CE Department,
Institute of Technology,
Nirma University, Ahmedabad.

Dr Alka Mahajan
Director,
Institute of Technology,
Nirma University, Ahmedabad

Statement of Originality

I, **Maitri Shah**, **16mcec21**, give undertaking that the Thesis entitled "**Tolerance of Flooding Attacks in Delay Tolerant Networks**" submitted by me, towards the partial fulfillment of the requirements for the degree of Master of Technology in **Computer Engineering** of Institute of Technology, Nirma University, Ahmedabad, contains no material that has been awarded for any degree or diploma in any university or school in any territory to the best of my knowledge. It is the original work carried out by me and I give assurance that no attempt of plagiarism has been made.It contains no material that is previously published or written, except where reference has been made. I understand that in the event of any similarity found subsequently with any published work or any dissertation work elsewhere; it will result in severe disciplinary action.

_____

Signature of Student

Date:

Place:

Endorsed by

Guide Name

(Signature of Guide)

Acknowledgements

It gives me immense pleasure in expressing thanks and profound gratitude to **Prof. Pimal S Khanpara**, Assistant Professor, Computer Engineering Department, Institute of Technology, Nirma University, Ahmedabad for her valuable guidance and continual encouragement throughout this work. The appreciation and continual support she has imparted has been a great motivation to me in reaching a higher goal. Her guidance has triggered and nourished my intellectual maturity that I will benefit from, for a long time to come.

It gives me an immense pleasure to thank **Dr. Sanjay Garg**, Hon'ble Head of Computer Engineering Department, Institute of Technology, Nirma University, Ahmedabad for his kind support and providing basic infrastructure and healthy research environment.

A special thank you is expressed wholeheartedly to **Dr. Alka Mahajan**, Hon'ble Director, Institute of Technology, Nirma University, Ahmedabad for the unmentionable motivation he has extended throughout course of this work.

I would also thank the Institution, all faculty members of Computer Engineering Department, Nirma University, Ahmedabad for their special attention and suggestions towards the project work.

- Maitri Shah

16mcec21

Abstract

A delay-tolerant network is a network which is designed to operate efficiently over extreme distances such as those in space communications.In these environments, latency is the major factor affecting the overall quality of network. However, this problem can also occur over small distances where interference of external entities is extreme or resources are overburdened. Delay tolerant network is the network in which neither the nodes in the network are constantly connected to each other nor any specialized network infrastructure is available for managing the network.By not having a specialized communication infrastructure DTNs are already facing some major challenges like communication delay,data dissemination and routing but another major challenge for DTN is to protect the network nodes from the attackers.Existing mechanisms provide security to a good extent but they are using a complex hashing algorithm which takes significant amount of time ultimately affecting the limited bandwidth and limited battery life of mobile nodes.So the main aim of this project is to design such an algorithm which is less computationally complex and more effective for detecting and tolerating flooding attacks.So I have proposed an algorithm which maintains the information of a node in a map(i.e.HashMap) and monitors the map whenever a transmission occurs. Traversing a map requires less computation and traversal time.As a result of this algorithm, many parameter values like delivery probability, overhead ratio ,etc. show improvised values.

Abbreviations

DTN	Delay Tolerant Network.
ER	Encounter Record.
EMR	ER Manipulation Ratio.
BAB	Bundle Authentication Block.
PIB	Payload Integrity Block.
PCB	Payload Confidentiality Block.
ESB	Extension Security Block.
TTL	Time To Live.
LGL	Legitimate Gateway List.
BNL	Blocked Node List.
SAL	Spoofed Address List.
p-RNG	pseudo Random Number Generator.
NS	Network Simulator.
ONE	Opportunistic Network Environment.

Contents

List of Figures

Chapter 1

Introduction

1.1 Delay Tolerant Networks

Delay/Disruption Tolerant Network abbreviated as DTN, is designed to establish connection between two or more nodes which are mobile, being carried by humans or vehicles.DTN enables to establish communication in the most unstable and remote environments in which the nodes in the network would face frequent disconnections. The need of this kind of network architecture is because of Interplanetary Internet, in which deep space communication in high delay environment is a major challenge. This network architecture is useful for sensor based networks, terrestrial wireless networks in which end to end connectivity is difficult, satellite networks, underwater networks with delay and interruptions due to some environmental factors. Due to lack of constant connectivity and communication infrastructure two nodes of DTN can communicate with each other and transfer data only when they both move into transmission range of each other.[5]

A newly introduced layer called Bundle Layer above the transport layer enables the functionality similar to the internet layer but it focuses on virtual message forwarding.DTN enables data delivery using automatic store-and-forward mechanism. When a node receives one or more packets, it stores these packets into its buffer space and whenever it comes in the range of another node which can be redirect the packet to its destination, it forwards them. Since contact between nodes are opportunistic and duration of contact may be short because of mobile nodes, bandwidth available for transmission is limited. And due to mobility of nodes they may have limited buffer space also.

1.2 Security Issues in DTN

Having restricted buffer space and limited bandwidth, DTNs[6] are exposed to severe misuse of resources . To detect and tolerate this kind of malicious behavior there should be some form of authentication or access control mechanism required. It is least desirable that any malicious node will flood[7] the network with traffic which results into service denial of other nodes. As DTNs can be used for military applications or any mission critical application , unauthorized traffic forwarding over some channels can also be harmful. It still lacks on certain security components.

- Prevent unauthorized nodes flowing their data through the network or storing it in DTN.
- Prevent malicious applications from inserting control parameters inside the DTN infrastructure.
- Prevent traffic transfer of an authorized node for which they don't have permissions.
- Preventing the tampered bundle from transiting inside the network by removing it.
- Detecting malicious application or activities promptly.

Besides this there are many kinds of attacks that are possible in DTNs. i.e. Flooding attacks, Black hole attacks, Worm hole attacks, Grey hole attacks, etc. Let us take a brief of these kind of attacks.

1.2.1 Flooding Attack:

So there are two types of flooding attacks[7] possible in DTN i.e. 1)Packet Flood Attack 2)Replica Flood Attack. For the Internet and wireless sensor networks, many schemes have been proposed to prevent flood attacks. And those mechanisms cannot be directly applied to DTNs because they assume persistent connectivity which is not possible in DTNs. So there is a need of such mechanisms which can defend DTNs against flood attacks.

1.2.2 Worm hole Attacks

In a wormhole attack[8] using a low latency link, a malicious node establishes connection between two compromised nodes. Recorded data packets are tunneled to the another

compromised node by the first node. Attackers can tamper the network topology views of the normal node and severely damage the routing in the network by just creating these kind of wormhole links.

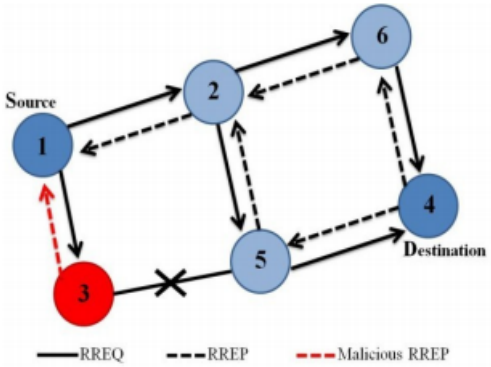


Figure 1.1: Black hole Attack[1]

The adversary transmits packets received from the one end of the wormhole to the other end of wormhole with the help of radio transceivers at the both end of wormhole. And re-instill the packets in the network at the other end of the wormhole. Any malicious node doesn't need to understand what it tunnels so it can also tunnel any encrypted packet. Wormhole attack can cause a severe damage to the routing in the network. In Wormhole attack if more than one node which have malicious purpose then this kind of collaborative attackers can perform a very effective attack. It is very dangerous attack for the network as the adversary nodes place themselves strategically in the network. The attackers keep on monitoring the data transfer inside the network.

1.2.3 Black hole Attacks

It is a major security threat for IC MANETS. In a network, sometimes it happens that incoming or outgoing traffic is silently dropped. The data never reaches to its destination or its intended node. This kind of behavior is caused by a black hole node which has malicious purpose. It is very difficult to track a black hole node inside a DTN by just checking the network topology. It can only be detected by monitoring dropped or lost packets. Such traffic loss is related to the likelihood of an illegitimate node is being on the path of the route to the destination. The adversary node selectively drops the packets which results in damaging the whole network.

In black hole attack[9], adversary nodes advertise that they are having the shortest path to the destination node. So malicious node will always reply positively to any route

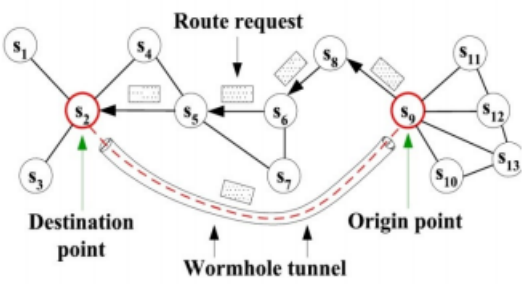


Figure 1.2: Black hole Attack[2]

request and can tamper any data packet on that route. Gray hole attack is similar to black hole attack to some extent. In gray hole attack attacker selectively transmit/drops the packets to cause the disturbance inside the network.

1.3 Routing in DTN

As in DTN there is no communication infrastructure, end to end packet delivery is never possible in it.So opportunistic contacts are made and data transfer happens.In DTN packets are routed hop by hop to the destination.So choosing the next hop is dynamically decided by the current hop. And for that many algorithms are designed to make routing happen in minimum cost.There are mainly two categories of routing protocols: Flooding Based And Forwarding Based[10].

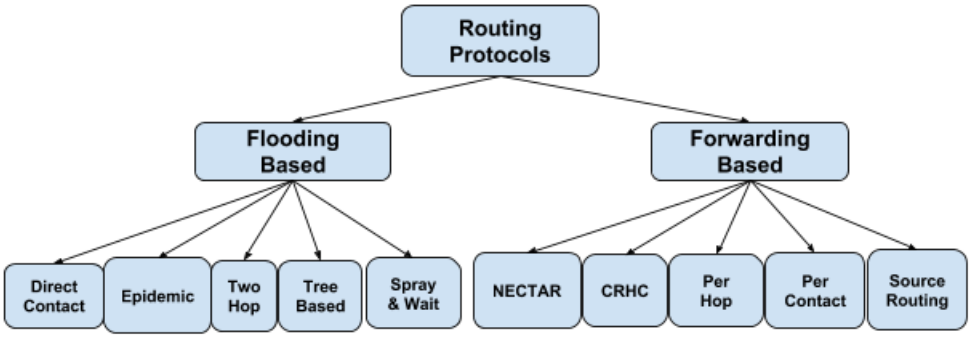


Figure 1.3: Routing Strategies In DTN [3]

1.3.1 Flooding Based Routing Strategies:

When a node in the network does not have any knowledge about another nodes in the network, then flooding based routing strategy is used.It is categorized into two categories further i.e. Replica based & Quota based.In Replica based routing number of replicas a

packet can have is fixed (i.e. no. of nodes-1).In quota based routing a packet can only have a fixed predefined number of copies.

Direct Contact[11]:

Bundle transfer in this algorithm is directly from source to destination. The source node will wait after creating the bundle until it encounters the destination. This algorithm does not require any knowledge about the network so it falls under the category of flooding based routing algorithm. In this type of algorithm delay for delivery of a bundle is very high and cost for that is too low.

Epidemic Routing[12]:

In this routing strategy ,source node of the data packet replicates the packet to each and every node it meets. A vector named Summary Vector is maintained at each node for capturing the information like the current messages in the buffer and all the packets that are passed by. Summary vector is checked before every replication of a packet.

Two-Hop Relay:

In Two-Hop relay routing[10], source node replicated the bundle to a large number of intermediate nodes. As the name suggests, the source node delivers the bundle to the destination either directly or within two hops. Intermediate nodes does not further replicate the bundle to any other nodes in the network.

Tree Based Flooding:

This binary tree based routing algorithm[13] works in such a way that source node of the bundle can only have N_c -1 number of replicas only. Each node in the binary tree can have maximum two child nodes so that replicas of the bundles are equally distributed among them.After receiving phase , to reach to the destination node transfers the load to collection station.

Spray and Wait:

Difference between epidemic routing and spray and wait[14] is only in the number of replicas source node generate. In this routing strategy optimal number is decided so to that number of nodes only , source send the replicas. So this routing algorithm is the advance version of epidemic routing algorithm. It consist of two phases: Spray & Wait. In the first phase , source node relays the bundles to the predefined number of nodes and

that nodes further relays the messages to the decided number of nodes. If in the spray phase destination is not found then all intermediate(relay) nodes will store the bundles and direct transmits them to the destination.

1.3.2 Forwarding Based Routing Strategies:

When nodes in the network have sufficient knowledge about the other nodes in the network then Forwarding based routing strategy is used.In this type of routing, a node will find a suitable node which has the highest possibility to encounter the destination and forwards the packet to that node.No replicas of a packet is generated so resource wastage is also reduced.So when resources are limited this routing strategy is used.

NECTAR:

In this routing strategy[15], each node maintains a table which stores the information about encounter frequency of a particular node with every other node of the network. The node which has the higher encounter frequency is assigned a higher index value. Whenever a node needs to forward a bundle to the destination it will choose the relay nodes with the highest encounter frequency with the destination.

Source Routing:

This routing strategy[16] consist of two phases route discovery and route maintenance. The route discovery phase consist of sending control packets towards the destination. Each relay nodes will append its address to the control packet it receives. Each node has its own buffer to store the routes it has learned so far. When the control packet encounters the destination ,the whole path to the destination is stored inside it. Route maintenance phase includes broadcasting an error message in the case of link failure.

Per-Hop Routing:

In this mechanism[17] for routing a packet to the destination, decision of the next node to which the packet is to be forwarded will be decided by relay node. This approaches performs better than Source Routing mechanism because in this mechanism more updated information is used to decide the route to the destination.

Per-Contact Routing:

In Per-Contact strategy[18] , any relay node receives the bundle for a particular destination the relay node checks for the current Up connections and choose an appropriate relay node for forwarding the bundle and then forward it to those Up contacts. Thus it uses the most updated information while routing.

Hierarchical Forwarding and Cluster Control Routing:

On the basis of link property and communication characteristics , this strategy[19] uses the concept of grouping(clustering) the nodes. Then cluster head is decided based on higher stability or the higher quality among all the nodes in one group. The cluster head is responsible for taking the routing decisions of that particular group.

Comparison Of Existing Routing Strategies:-

Routing strategies which are available for DTN have their own advantages and disadvantages.Some provide optimum resource utilization on the other side some consumes more resources but performs well.So which routing strategy should be used is totally depend on the resources one network may provide or the application for which the network is designed.Comparison of some existing routing approaches is shown in the tables[3] below.Where N is number of nodes in the network and K is optimal no. of nodes to guarantee delivery in Two hop routing.

Table 1.3.2(1) Comparison of Flooding Routing Techniques[3][20]

Protocol	**No Of messages generated**	**Message Delivery ratio**	**Avg Delay**	**Resource Consumption**
Direct Contact	Single	low	high	less
Epidemic	N-1	high	low	high
Two Hop	K	medium	medium	less
Tree based	$1 + log(N/2)$	medium	high	medium
Spray& Wait	$> K$	medium	medium	medium

Table 1.3.2(2) Comparison of Forwarding Routing Techniques [3][20]

Protocol	**Information Maintenance**	**Message Delivery ratio**	**Avg Delay**	**Resource Consumption**
NECTAR	medium	high	normal	less
Per Hop	medium	medium	medium	low
Per Contact	medium	high	low	medium
Source	normal	low	high	low
CRHC	high	high	normal	high

Chapter 2

Flood Attacks in DTNs

2.1 Effect of Flood Attacks

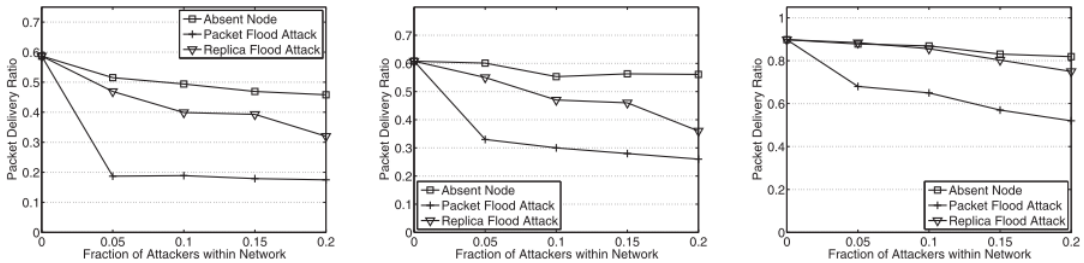
Any number of nodes may launch flood attacks for any malicious or selfish purpose.Malicious nodes,which can be the nodes that are deliberately deployed to bring down the performance of the network by wasting their limited resources or congest the network.And Selfish nodes,may launch flood attacks to congest the other network and increase its own communication throughput.A selfish node increases its throughput by flooding many replicas of its own packet.And in DTN contacts are opportunistic so probability of the packet delivery is less than 1.By flooding its own packet many times a selfish node can increase its own communication throughput.

2.2 Need of Tolerance Capabilities in DTN

To study the effect of flood attacks and the need of tolerance capabilities in DTN, let us understand three more used routing strategies in DTNs. 1)Single-Copy routing[21]:In this routing strategy,particular one node removes its own replica of a packet after sending it to other node in the network.That means a packet only have one copy in the whole network.2)Multi copy Routing[3]:The node which generate a packet sprays a predefined (maximum replicas a packet can have is fixed) number of copies to the other nodes during contact time and each copy is further routed individually by following single copy routing strategy.3)Propagation Routing[22]:when a node find an appropriate node to forward the packet according to the routing algorithm it simply copies the packet to the contacted node and keep its own copy.There is no predefined limit over the replicas of a packet to

be generated.

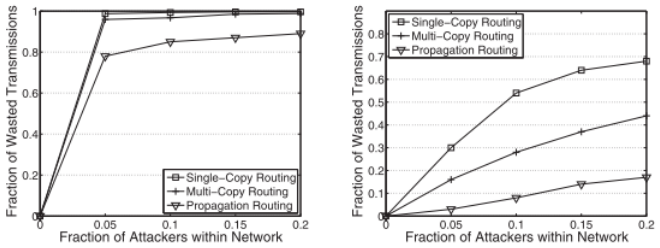
If we consider the simulation[7], two matrices are used.The fraction of packets delivered to their destinations out of all unique packet generated is defined as packet delivery ratio.And second metric is fraction of wasted transmission which is the transmissions made by good nodes for flooded packets.Higher the fraction of wasted transmission more network resources are wasted[23].In their simulations a packet flood attacker floods the packets destined to good nodes of the network in each contact or until the contacted node's buffer is full.Replica Flood Attacker replicates the packet it has generated to every node it contacts which does not have a copy of that packet.



(a) Single-copy Routing (b) Multi-copy Routing (c) Propagation Routing

Figure 2.1: Effect of flood attacks on packet delivery ratio.

By analyzing the simulation results[7], Fig 2.1[7] shows the effect of packet flood attacks on packet delivery ratio.Packet flood attack can reduce the packet delivery ratio to a great extent in all three types of routing.And Replica flood attack can significantly reduce the packet delivery ratio for single copy and multi copy routing. But it does not effect much on propagation routing.



(a) Packet Flood Attack (b) Replica Flood Attack

Figure 2.2: Effect of flood attacks on the fraction of wasted transmission.

Figure 2.2[7] shows the effect of flood attacks on wasted transmission.Packet flood attack can waste more than 80 percent of the transmissions made by good nodes in all routing strategies when the fraction of attackers is higher than 5 percent[7]. Replica

flood attack can waste 68 percent of transmissions in single copy routing and 44 percent of transmission in multi copy routing when 20 percent of the nodes are attackers. Replica flood attacks only wastes 17 percent of transmissions in propagation routing because in that scenario every good packet is also replicated.

Thus as all three routing strategies are vulnerable to packet flood attacks and replica flood attacks, security of DTNs is becoming a major issue.So there is a major need of developing such mechanisms which can prevent these type of attacks.

Chapter 3

Literature Survey

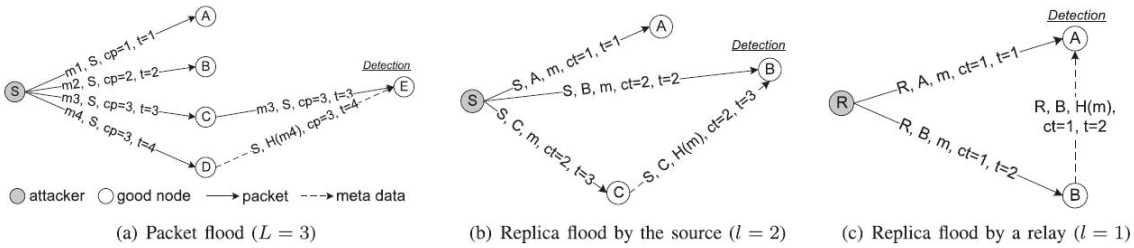
3.1 Existing Mechanism:-1

3.1.1 Claim-Carry And Check Using Rate Limiting Factor

In this paper[7][24], they have presented a rate limiting factor to defend against flood attacks using claim-carry and check in DTNs. In this mechanism each node has an upper limit for the number of packets it can generate in each time interval and a limit over the number of replicas that it can produce for each packet. They have proposed a distributed scheme to detect if a node has violated its rate limits.It is very difficult to count all incoming and outgoing packets because of absence of any communication infrastructure.In claim carry and check, each node monitors and maintains the incoming and outgoing packets from the node; the node which receives the claim,carries it with themselves .And whenever it makes contact with any other node,it checks for the inconsistency of claims. Pigeonhole principle will help us in detecting the inconsistent claims.

Claim-carry and check:

Any node in the network has to maintain the count for the number of packets it has sent out as a source into the network in a particular time interval to detect whether any node has violated the rate limit or not which leads to the detection of an attacker. Claim (up-to date packet count for a particular node with its ID and time stamp) is sent out with each packet. Along with claim the nodes rate limit certificate is also attached so that every receiving node can verify a nodes authenticity. An attacker will try to send packets more than its rate limit with dishonestly claiming a count smaller than its rate limit.



(a) Packet flood ($L = 3$) (b) Replica flood by the source ($l = 2$) (c) Replica flood by a relay ($l = 1$)

Figure 3.1: Basic idea of flood detection

That count must have been used before for another packet (pigeonhole principle) and thus this will be the clear indication of an attacker. When the two of the nodes having carrying the same claims contacts each other, inconsistency is found and the attacker is detected.Example is given in Figure3.1(a)for packet flood attacks.[24] This mechanism is also used for detecting an attacker which forwards the buffered packet more than one time for Replica flood attack. When a source node or an intermediate node transmits the packet to its next hop, it carries the claim which contains the number of time it has transmitted this packet including the current transmission. Thus if an attacker wants to transmit a packet more than its limit, it must use a count which has been used before .So the attacker can also be detected here by finding the inconsistent claims. Examples are given in Figure 3.1(b),(c) [24] for replica flood attack.

Claim Construction:

P-claim is used to detect packet flood attack and T-claim is used to detect replica flood attack.P-claim is generated by the source and sent to the next hop along with the original packet.On the other side, a source generates the T-claim and append it to the packet.Whenever that packet is passed to the next hop,that hop peels off T-claim and check for the inconsistency.And then it appends new T-claim to the packet.Generally P-claim of a source and T-claim of previous hop is used to detect attack by any hop.When a source node S sends a new packet which has been generated and not sent out before to a contacted node.It generates P-claim and T-claim as follows:

$$\text{P-claim:S,C}_p\text{,t,H(m),SIG}_s\text{(H(H(m)|S|C}_p\text{|t)).}$$

Here, t is the current time stamp.C_p is the packet count($1 < C_p <$L)of S where L is the rate limit,which means that this is C_p^{th} packet is generated by S and sent into the current time interval t.The P-claim is attached to the packet header and forwarded to next hops.Once a hop receive this type of packet it verifies signature and check the packet count claimed

in P-claim with its rate limit L.If C_p >L then it discards the packet otherwise it stores the packet into its buffered storage.

T-claim is appended to a packet whenever it is being transmitted to other node.Suppose, say that node A has transmitted a packet m to node B,T-claim which is appended to m includes the number of time this packet has been transmitted out by node A and current time stamp t.T-claim is:

$$\text{T-claim:A,B,H(m),}C_t\text{,t,SIG}_A\text{(A|B|H(H(m)|}C_t\text{|t)).}$$

On receiving any node will check C_t against the limit and take actions accordingly. Inconsistency at any node can be detected by verifying P-claim and T-claim.For example in Figure 3.1 the count value 3 is reused in the P-claims of packet m3 and m4. Similarly, count reuse is also caused by dishonest T-claims.

Algorithm Used[7][24]

1: P-claim and T-claim generation
2:if a node Have packets to send then
3 : P-Claim, T-Claim and sign are generated for each newly generated packet
4 : Send every P-claim and T-Claim attached to a packet
5:end if
6:if a node Receives a packet then
7 : if sign verification fails or the count of P-Claim/T-Claim is inconsistent
8 : Delete the packet;
9 : end if
10: Check local P-claim/T-Claim with newly arrived P-Claim/T-Claim for inconsistency;
11: if Inconsistency is detected then
12 : Mark the signer of that claim as an attacker and add it into a blacklist;
13 : Broadcast an alarm against the attacker to the network;
14: else
15 : Store the new P-claim and T-Claim;
16: end if
17: end if

3.1.2 Detection Strategy

Each node stores P-claim and T-claim in its local buffer whenever the node sends or receives any packets.Initially, full P-claim and T-claim are stored in the buffer.When any node removes a packet from its buffer (i.e., after a packet is delivered to the destination or dropped due to expiration), it stores compacted P-claim and T-claim to reduce storage cost.

Whenever a node receives a forwarded packet with a claim,it checks for inconsistency of it with a locally stored claim by verifying its signature.Forwarded packet's claim is a full claim .But locally stored claim may be stored as a full claim or compact claim.If locally store claim is a full claim ,that node can inform other node in the network via broadcasting a global alarm containing local claim and received claim.Upon receiving an alarm any node will check for the inconsistency between those claims.If found then it further broadcasts the alarm otherwise it discards the alarm.It does not broadcast the alarm further if it has already broadcast the alarm for the same claim before.If locally store claim is not a full claim,it can not broadcast a global alarm.Because a compacted claim does not include a node's signature in a claim.So any node can not be convinced upon receiving such claim about that node's authenticity.Since the attacker may have used that claim for another nodes also besides that locally stored claim,the detecting node can send a local alarm to the contacted nodes who have received that false claims.If any of those nodes is having a full claim , attacker can be detected and that detecting node generates global alarm to all other nodes.Upon receiving a global alarm a node removes its local alarm.

3.1.3 Advantages And Limitations

This scheme uses an efficient claim construction mechanism that keeps communication,computation and storage cost low.Even the attack detection probability is also high which makes this mechanism effective.It woks in a distributed manner such that no dependency on a single node which works as a central authority.But when the packet generation rate is too high some of the packets are dropped due to buffer overflow so the communication and computation time which was used to generate the signatures and constructing their claims are considered as wasted. So this is an overhead in major traffic scenario.

3.2 Existing Mechanism:-2

3.2.1 Using Encounter Records

In this scheme[25], they have proposed a detection scheme for flooding attack which piggybacks on an existing encounter record based scheme of detecting black-hole attack. To record the sent messages during their previous contacts , nodes required to exchange their ER(Encounter Record) history. Using this mechanism malicious node will be detected causing packet flooding or replica flooding attack. Adversary nodes may bluff the attack by removing or skipping unfavorable ERs. However, this also results in detection of attack due to inconsistency in time stamp and sequence number of ER.

How it works?

Suppose two nodes i and j with respected identifiers ID_i and ID_j contacted each other.After exchanging messages, each node generates an encounter record and stores it locally.The record ER_i^* stored by node i is as follows:

$$ER_i = <ID_i, ID_j, sn_i, sn_j, t, SL_i>$$
$$SL_i = \{MR_m | \text{ i send message m to node j}\}$$
$$ER_i^* = ER_i, sig_i, sig_j$$

where sn_i,sn_j are the sequence numbers for the nodes i and j respectively,t is the encounter time and sig_i,sig_j are the signatures for the of ER_i using their own private keys.SL_i indicates the messages that is sent by node i to j.If the node is contacting any other node ,it assigns new ER with sequence number incremented by 1 from its latest ER.Each message record is denoted as below:

$$MR_m = < ID_m > \text{ if } SRC_m \neq i$$
$$MR_m = < ID_m, REP_m, GEN_m > \text{ if } SRC_m = i$$

here, ID_m is the identifier of the message,REP_m is the number of replica of message m that node i has generated.GEN_i is the time stamp at which the node i has generated the message m.Due to limited buffer space any node keeps only a window of w latest ERs to show it to its neighbours.

For hiding an attack or malicious activity, an attacker may forge its own ER history to obtain a window which is advantageous for itself.It will present this forged window to its neighbour nodes.

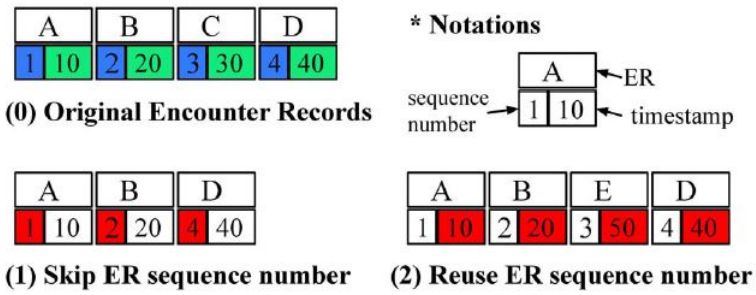


Figure 3.2: Encounter Record(ER) Manipulation Strategies

As shown in Figure 3.2 [25], for example an adversary node has an original series of encountered records A,B,C,D with their respective sequence number and time stamp ((1,10 min),(2,20 min),(3,30 min),(4,40 min)). If malicious node wants to delete the unfavourable encounter record for example C.It may not generate another ER using C's sequence number i.e. 3 or it may reuse the sequence number 3 to generate another ER i.e. E(3, 50 min).

3.2.2 Detection Strategy

Malicious nodes may have two types of misbehaviour.Either skip a sequence number or reusing the sequence number for any other node.Whenever any node manipulates the ER, its series of sequence number or time stamp becomes inconsistent.When any attacker floods messages, they have their rate limit for number of packets a node can send or number of replica of packet can be forwarded exceed the allowed limit. This leads to the detection of attack. When two nodes i.e. i and j are in contact, they exchange their ERs to examine behavior of each other . If any of the node finds that the other node is suspicious or malicious node, it will blacklist the node and also informs the network about the attacker if it has verified it properly. ER consist of consecutive sequence numbers. So higher sequence number has a bigger time stamp. But malicious nodes manipulate ERs using any of the strategy shown in Figure 3.2 [25]. If any of the attacker has skipped a sequence number it will have nonconsecutive sequence number i.e.1,2,4.And if an attacker has reused the sequence number, the series of time stamp may result in non decreasing order i.e. 10,20,50,40. Thus by examining the time stamp or sequence number of any of the nodes , one will be able to detect the attack.

ER of node j will be used by node i to obtain how many packets j generates per time interval and how many replicas j forwards for each packet. Then i compares these limits

with its predefined limits(L).If j violates the limits , i blacklists j. From the ER history of node j, node i can extract the set of distinct messages that node j has recently sent to the network. Suppose n messages are transferred and their respective time stamps are $t_1, t_2, , t_n$. It means that node j has generated n messages during $(t_n - t_1)$ time period. On an average node j has generated n^* messages in each time interval T i.e. $n^* = n/ceil((t_n - t_1)/T)$. If $n^* > L$, node j is detected as flood attacker. Node i can obtain replica numbers which are appended to each packet that node j has created and transmitted by checking ER history. Replica count is defined as $rep_1, rep_2, , rep_k$ for such m packets. And their encountered time stamps are also managed. If node j is not an attacker, its series of replica count should be sequential i.e. $rep_l = rep_l - 1 + 1 for l = 1, 2,$ .The highest replica count rep_k should not cross replica count limit L. If any of these conditions are violated, node j is detected as replica flood attacker.

3.2.3 Advantages And Limitations

By analyzing this scheme, detection delay decreases and the detection rate increases as EMR increases. Detection rate of a flooding attack is very high (almost 1). It does not incur any false positive. Storage requirement for each encountered record is affordable enough for mobile nodes. But each node has to be capable enough to decode the ER every time it contacts. And if the node which is attacker comes in the contact with such node which have the ER history of that particular node after long time then the effect of attack will slow down the network due to resource over utilization.

3.3 Existing Mechanism:-3

3.3.1 Using Stream Check Method

This scheme [26]has an intrusion detection mechanism that uses streaming node (a node with monitoring capabilities) to monitor the network environment.Monitor node has to maintain three tables.First table is Rate Limit table which include rate limits of all the nodes in the network, second table is of Delivery probability table which contains probability of delivery of each node in the network.And a table for blacklisted nodes which are attackers. Streaming node compares estimated probability of delivery and actual probability of delivery of packets.If difference greater than assigned limit value,that

node will be listed as malicious node by the streaming node.

3.3.2 Stream Check Detection Method

Streaming node[26] is used to detect malicious node inside a DTN.At every time interval, whenever two nodes of the network contacts streaming node travels along the path of the packet to check the authenticity of the communicating nodes.Rate limit is used to restrict the packet rate and it is performed in a request response manner.It is assigned by some trusted authority based on your traffic demand. During the transmission packet is transmitted into small data blocks.Rate limit table contains approved rate limit for each node,node details, starting and ending sequence number. The node can send packet to the node to which it contacts. Streaming node is not monitor these activities so node itself monitors these activities.Node itself manages the updated packet count and updated claims.And node also verifies the claims against its rate limit certificates which are attached to the packet.If an attacker node is flooding network with new packets or with the replicas of the same packet by claiming false count,streaming node detect this node as an attacker because it has violated its rate limit and list it into blacklist table and inform all the nodes in the network.

For example[26],an attacker in the network knows that two nodes(i.e. A and B) never communicate to each other.Then attacker can send some packets to one of the nodes i.e. node A and invalidly replicates that packet and send it to the other node B.Since A and B never communicate, in this case the attacker cannot be detected.In this case as streaming node contains three tables which have all the information about all the nodes.It compares these tables with all the nodes which participate in the communication.It first check for the rate limit then check the black listed table if any of the nodes are already added into that table or not.And then check probability of the delivery of the packets that the node has estimated.Streaming node compares the actual value of probability (calculated from the rate limit table with starting and ending sequence number) with the estimated value.If it is greater than the threshold value,streaming node will list that node as an attacker and add it into the black listed table.

3.3.3 Advantages And Limitations

This approach can find malicious node very efficiently and effectively in such conditions where attacker node deploys attacks in such a way that no distributed mechanism is able

to find it.Packet delivery ratio is increased and propagation delay is decreased using this mechanism.But using this mechanism whenever the streaming node fails or corrupts, the network will collapse.

3.4 Existing Mechanism:-4

3.4.1 Using DTNCookie

As DTN is vulnerable to resource misuse and flooding due to DoS attacks, several DoS mitigation schemes for wired and wireless networks has been introduces but those cannot be directly applied to DTNs. To tolerate the effect of flooding and resource exhaustion they[27]have proposed scheme which includes ingress filtering, limiting rate and some security mechanism to monitor , detect and filter the traffic from the attacker. They have proposed a concept of DTNCookies. They have proposed three variants of the light-weight packet authenticators (DTNCookies).To make DTNCookies random in nature and hard to forge by the attacker nodes , they have made an assumption that DTN nodes are loosely time synchronized to generate a number that is unique in different time slots for the verification and computation. The proposed scheme encompasses packet authentication, rate limiting mechanism and ingress filtering for gateways. For both the scenarios (i.e. intra-region and inter region DoS) they have proposed three DTNCookie variants. Figure3.4 Fields for security authentication are added to a DTN bundle like Bundle Authentication Block(BAB), Payload Integrity Block(PIB), Payload Confidentiality Block(PCB) and DTNCookie block. This fields are useful for providing protection to bundle from attacks like modification attacks, replay attacks, flooding attacks, resource exhaustion.

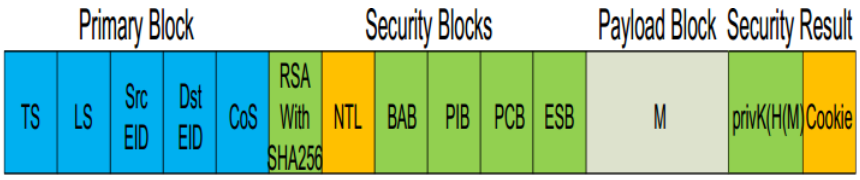


Figure 3.3: A Generic bundle with security block

Table3.4.1provides a description of bundle fields shown in Figure 3.3.

Table 3.1 Bundle Fields and their meanings

Symbol	**Description**
TS	Bundle Time stamp
LS	Bundle Life Time
SrcEID	Identifier of the Source End point
DstEID	Identifier of the Destination End point
RSA-SHA256	Cipher suite for digital signature
NTL	Network Threat Level Indicator
BAB	Bundle Authentication Block
PIB	Payload Integrity Block
PCB	Payload Confidentiality Block
ESB	Extension Security Block
M	Message Payload
H(M)	Hash of the message payload
privK(H(M))	Digital signature
Cookie	DTN Cookie Block

How it works?

The three DTNCookie Variants are defined as below:

DTNCookie1= H (((TS | SrcEID) | LS | CoS | NTL) | p- RNG (IV))

DTNCookie2= H(((TS | SrcEID) | LS | CoS | NTL)xor p - RNG (IV))

DTNCookie3= Hmac(((TS | $SrcEID$) | LS | CoS | NTL)$xor p - RNG(IV)), K_{RS}$)

The proposed light-weight (DTNCookies) are derived from the fields which are specific to each bundle. DTNCookie1 is generated from an IV(Initialization Vector) which is known to only legitimate and registered nodes. Which is further used as base to pseudo-Random Number Generator(p-RNG). The resulting value is a big integer which is used as nonce. To uniquely identify each packet, combination of packet's source address and its generated time stamp is used. And further the nonce is combined with it. The result of it is hashed using SHA-256 denoted here using the notation H. And the resultant fixed length hash h will be the light-weight DTNCookie which is appended to every packet. DTNCookie2 is formed same way as DTNCookie1.The only difference is that in this the concatenation operation is replaced by XOR(Exclusive-OR) operation. The

XOR operation enables more randomness in the DTNCookie generation. Same way as the DTNCookie2, DTNCookie3 is generated. The only difference is that the result is hashed with a secret key (K_RS) using SHA-256 to produce a fixed length MAC(Message Authentication Code) which is appended to every packet. The reason of DTNCookie's randomness and secrecy is because of the pseudo-Random Number Generator, bit wise XOR operation and it's length. So because of this DTNCookies are hard to tamper. DTNCookie1 and DTNCookie2 are used as lightweight bundle authenticator in the intra region scenario shown in the figure 3.4. DTNCookie3 is used for the same purpose as shown in the figure 3.5 for inter region scenario. DTNCookie3 encompasses more complex computation than DTNCookie2 and DTNCookie3. Here, they have assumed that DTN gateways are loosely time synchronized for deriving DTNCookie3.

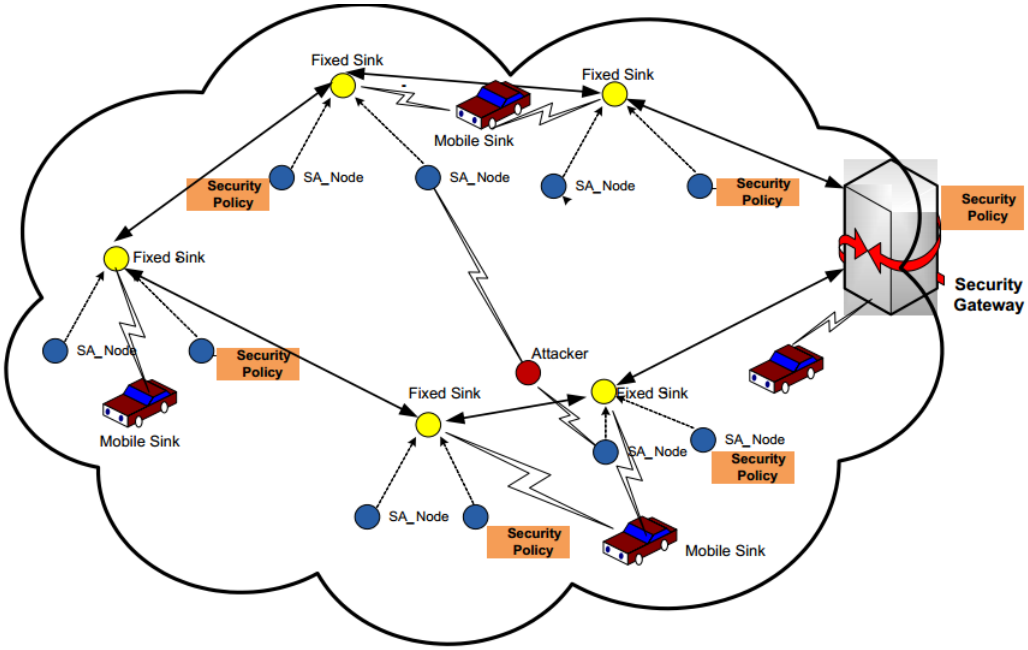


Figure 3.4: A DTN Region: Intra Region Scenario

3.4.2 Detection Strategy

For differentiating between legitimate and illegitimate traffic of the network, they have proposed a scheme of analyzing the address of the source packets and some specific fields of that packet. A regional gateway is shown in the figure3.6 with two interfaces in and out . The bundles which pass through the gateway, it has to pass through the ingress filter for confirming that the bundle is originated from a trusted source. The rate limiting filter ensures that each traffic flow should not exceed its predefined threshold value. Rate

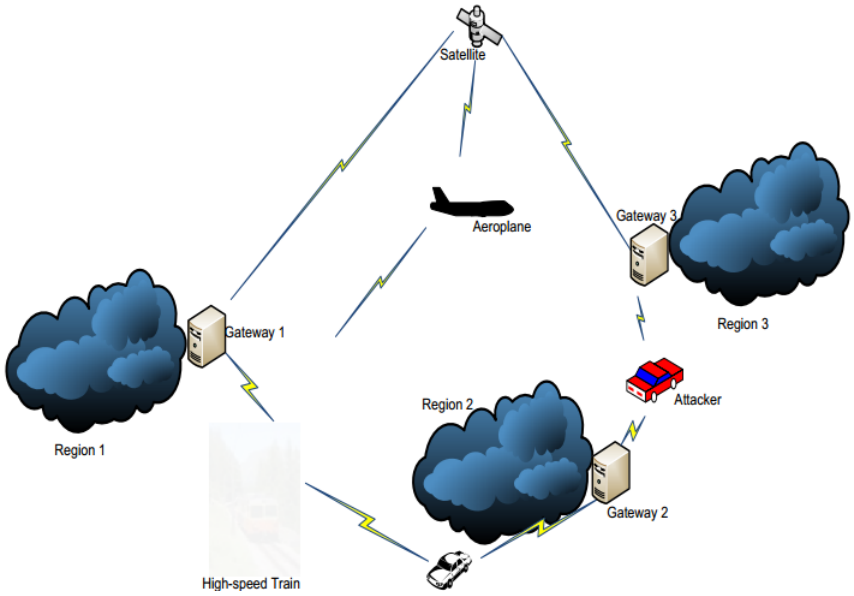


Figure 3.5: DTN Regions connected via Gateways and Data Mules : Inter Region Scenario

limiting filters reduce the effect of the DoS attack .The bundles that do not fulfill the specified security policy requirements are discarded and the address of the node that has originated the packet is kept with the gateway to take future decisions wisely. Thus light weight filter authentication helps to ensure that only trusted and legitimate bundles are allowed to use the DTN network resources. Steps taken by a gateway to handle DoS attacks caused by flooding shown in figure 3.7 .

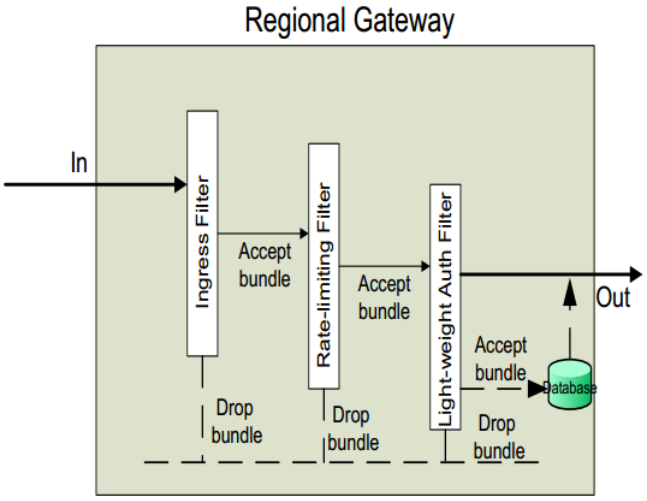


Figure 3.6: A DTN Gateway Block Diagram with DoS Filters

They have incorporated a rate limiting filter which helps to identify the illegitimate nodes that send bundles at a high rate than specified. Such nodes are identified and penalized for their behavior. The size of the bundle, TTL and Class-of-Service rights

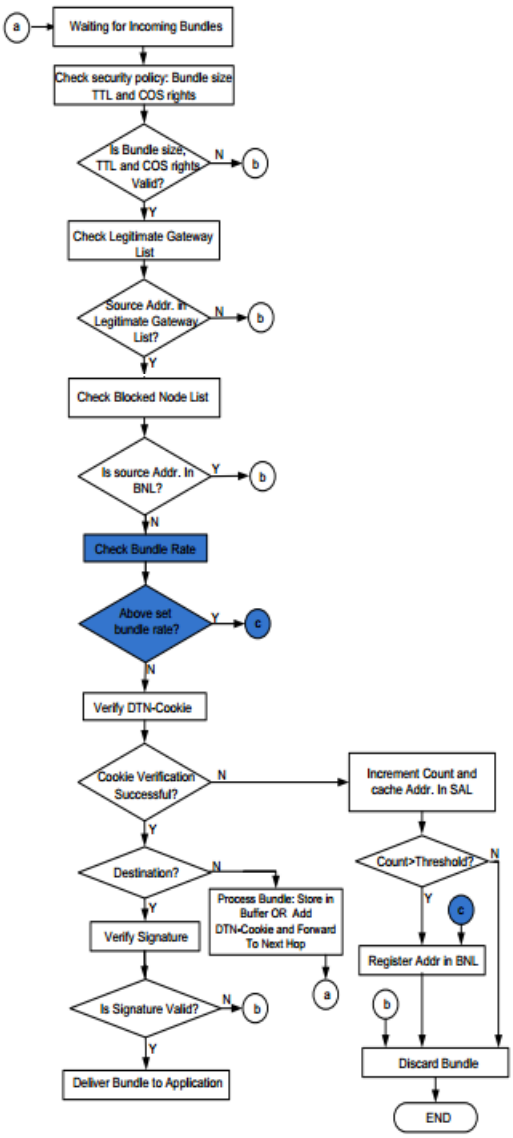


Figure 3.7: Flooding Attack Mitigation in DTN Gateways

of the user are checked whenever a bundle passes through the gateway. Those bundles who do not match the security requirements are simply discarded. On the other side if size of the bundle, TTL and Class-of-Service rights meet the security requirements of the gateway then the address of the source node of the bundle is stored in the LGL(Legitimate Gateway List) of the gateway. If an address of a source node of a bundle is not stored in LGL then it is to be considered as an illegitimate and malicious node. On the other side, if an address of a source node of a bundle is present in LGL then it is to be again checked in the BNL(Blocked Node List). The gateways which have failed DTNCookie verification or crosses the predefined rate limit are considered as malicious gateways and the addresses of these kind of spoofed gateways are listed in the BNL. If a gateway which is listed

in BNL and crosses the rate limit by creating more bundles then subsequent bundles from these kind of gateways are dropped until time period of a configured block expires. This results in more free bandwidth for the trustworthy/legitimate node which improves the network performance. However, if required security configuration are valid then the gateway proceeds to validate the DTNCookie. The gateway verify the DTNCookie by checking the timestamp to pick the right seed for creating the unique nonce for processing the DTNCookie. The computed DTNCookie and received DTNCookie must match. If they are not matched then the packet must be from the unauthorized source and it must be dropped. The bundles for which the DTNCookie verification fail , source addresses of those bundle are kept in a list called Spoofed Address List (SAL). The nodes which are in SAL automatically entered in BNL when a COUNT exceeds a predefined threshold. Until the configured block period expires, packets from unauthorized/illegitimate gateways are dropped. If the DTNCookie verification is successful then the gateway verify the digital signature which is protecting the payload of the packet. Bundle is delivered to the application if the digital signature verification is successful. The bundle is either stored in the buffer of the gateway or another DTNCookie is computed and attached to the bundle and then forwarded the bundle to the next hop if the gateway is not the destination for the bundle. On the other hand if digital signature verification is failed then the bundle is discarded.

Table 3.4.2 describes the use of variable seeds in computation of unique nonce value in different time period. It ensures that DTNCookie is hard to forge and random. As the nonce variable is generated in different timeslots so time synchronization is an important factor for the proposed scheme. We assume that the security gateways gave a common view of time irrespective of their time zones. The pseudo random number generator(p-RNG) functions on a security gateways have uniform initial seed value(No). DTN owner or the Administrator (one of the gateways let's say Gateway 2) send two different seed values to other two gateways Gateway1 and Gateway3. The seed is encrypted using the public key of Gateway1 and Gateway3 and it contains bundle payload. Gateway 2 sends the bundle to the another gateways by appending singed bundle using its own private key and calculated DTNCookie. Based on the prior shared symmetric keys between all the gateways, timestamp and the sender EID are captured at the gateways.

After the computation of DTNCookie , the computed DTNCookie is compared with

the received bundle. If the DTNCookie verification fails then the bundle is dropped. If verification is successful then to test the integrity of the digital signature, each gateways verifies the digital signature using public key of Gateway2. Packet is drooped for modifies content on transit if the digital signature verification fails. On the other hand, if the digital signature verification is successful then they proceed to decrypt the bundle payload and each gateway decrypt the bundle payload using its own private key which is now the new reference seed for generating the nonce values. Attackers which are present inside the coverage of data mules are able to monitor each and every communication channel if the channel is a broadcast channel as in the satellite communication. For protecting the seed against eavesdropping they encrypt the payload. If the TTL of the bundle is not expired it can still be processed if they arrive after the creation timeslot. Bundle timestamp is captured at a receiving node for determining the seed it should use for nonce creation for the verification of the bundle.

Table 3.2 Creation and use of Nonce variables during different Time slots[27]

Timeslots(sec)	**Nonce Variables**	**Description**
0-7200	N_0	When a reference seed S_0 is given as input in p-RNG, nonce N_0 (256 bit random bigInteger) value is generated.
7200-14400	N_1	256 bit random number,Derived using seed S_1 ,seed S_1=S_0 + [counter] when seed S_1 is input into a p-RNG.
14400-21000	N_2	Derived using seed S_2 ,seed S_2=S_1 + [counter] and derived same way as N_1.
21000-28200	N_3	Derived using seed S_3 ,seed S_3=S_2 + [counter] and derived same way as N_2.
28200-36000	N_4	Derived using seed S_4 ,seed S_4=S_3 + [counter] and derived same way as N_3.
36000-43200	N_5	Derived using seed S_5 ,seed S_5=S_4 + [counter] and derived same way as N_4.

3.4.3 Advantages And Limitations

By analyzing this scheme, delivery ratio and average latency does not affect much as the number of attackers increases in the network. Even if in the high traffic scenario, delivery ratio does not affect much(negligible). By generating random number using cryptography and using the nonce variables for generating the DTNCookie , made DTNCookie hard to forge and very secure. This scheme is also capable of protecting the limited resources like energy for battery, buffer storage of the nodes and bandwidth. But providing such efficient and computationally powerful resources for deploying this algorithm might not be cost effective otherwise this scheme is suitable for all the scenarios in which tolerance/mitigation of DoS attacks are concerned.

Chapter 4

Simulator Study

4.1 Introduction

Through mobile communication devices, communication of voice and data is possible which enables global connectivity using infrastructure networks i.e. cellular, WLAN. Local connectivity among the mobile devices can be obtained by constructing an ad-hoc network as the mobile devices are capable enough to act as routers. These types of networks do not face frequent topology changes or disruption so TCP/IP based communication can work in these scenarios. But for intermittently connected mobile ad-hoc networks end-to-end connectivity is not possible and it faces sudden topology change or disruption because of some unwanted parameters so we can not directly use the conventional TCP/IP based communication in this kind of network scenarios.

As in Delay Tolerant Networks, no end-to-end connectivity is available so it enables communication in challenged environments like satellite communication, underwater networks, etc. So conventional routing strategies would not work for this kind of network so new routing protocols have been developed to provide support for Delay Tolerant Network. For analyzing the behavior of the DTN routing and protocols, simulation is necessary. DTN simulation work with an assumption that two nodes can communicate only when they move into the range of each other. Simulation helps to extract information about the link characteristics, bundle transfer, etc. Many new simulators have been developed for DTNs and extensions for many existing simulators are also available for DTNs like Opportunistic Network Environment (ONE) Simulator, Network Simulator (NS).

4.2 Network Simulator

NS is the most popular open source tool for computer networks related simulation work. It is a discrete event simulator. One of the main advantages of using simulations compared to real networks for testing is that a simulation environment is much cheaper to set up, and is also usually faster to set up because the simulator only needs a topology description. It has mainly two known versions NS2 & NS3.

4.2.1 Network Simulator 2 (NS2)

Network Simulator 2[28] is a discrete event and open source network simulator used for simulating different network topologies of the different network protocols of wired as well as wireless networks. It is the most commonly used network simulator tool for research and development work. NS2 is built in C++. And by using oTCL (Object TCL), the simulation interface can be specified. User will have to give the input to the simulator by writing oTCL script for describing a network topology. And then this network topology will further simulated by the main NS program with specified parameters. Network Animator (NAM) is used for the graphical view of the NS2. The GUI for the NS2 has the control functions which allows user to play, pause, stop or forward the simulation. Arbitrary topologies which are composed of links, routers and shared media can be defined. There is an event scheduler queue which contains all the physical activities in the form of events. And these events are scheduled as the simulated time increases and the scheduled event time is encountered. This simulator does not work with the real time. The simulation time is considered to be virtual.

4.2.2 Network Simulator 3 (NS3)

As NS2, NS3[29] is also an open source and discrete event simulator. It was initiated in 2006 and it is still under major development. It is not to be considered as an extension to NS2 but it is to be considered as a replacement to NS2. NS3 is also written in C++ and there is limited support for python also. But for simulating any network topology, oTCl script is not required. The latest working version of NS3 is 3.28 that support IPV6 for LTE, extended addressing, FIFO queue disc for traffic control modules. In NS3, network topology specification can be done in C++ and some parts of the topology simulation can also be done in Python. It supports emulation and simulation using sockets. It

generates the trace file in the form of pcap (Packet Capture) traces. And network tools like Wireshark can be used for reading those traces files to analyze the network traffic.

4.3 Opportunistic Network Environment Simulator (ONE)

ONE is a simulator tool which is developed in JAVA offering a broader set of DTN protocol simulation capabilities in a single framework[30]. It is an agent-based discrete event simulation engine. It is specifically designed for evaluating DTN routing and application protocols. The main functions of the ONE simulator are the modeling of node movement, inter-node contacts, routing and message handling. Data flow between different modules of the ONE simulator is shown in the figure4.1 below.

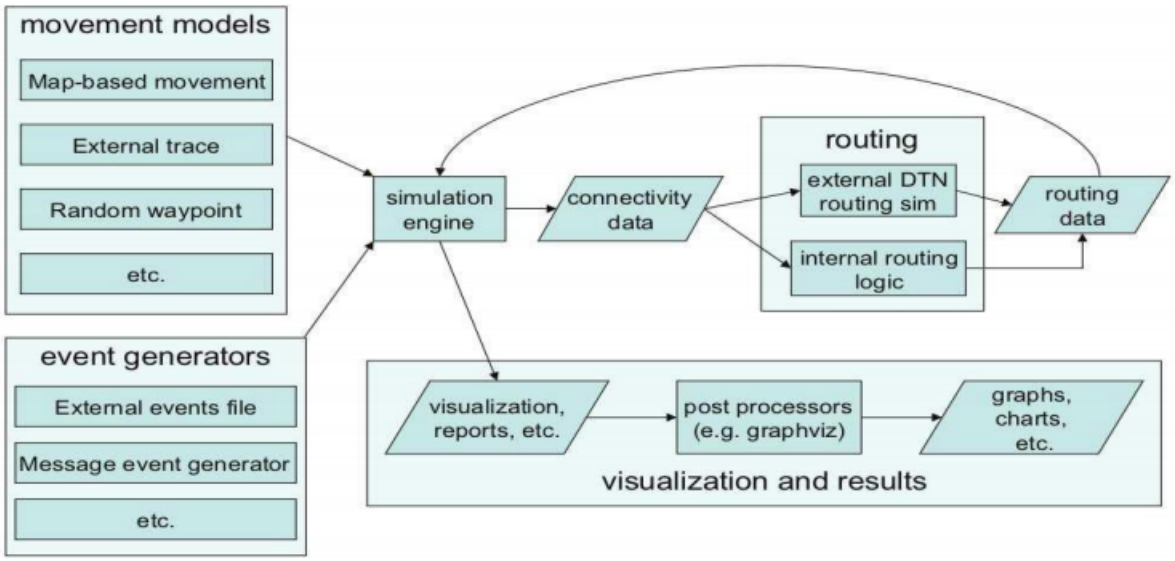


Figure 4.1: Control Flow inside ONE Simulator[4]

Outcome of a scenario simulated in ONE and its analysis can be done using visualization, reports, and many post processing tools. Communication between the nodes is dependent on each node's location, its bandwidth and range of communication. There are many routing protocols implemented in ONE. The routing function which is implemented by the routing protocol decides which message to forward or replicate over the existing contacts. The messages in the simulator are generated through event generators and having single source and single destination nodes in the simulator that means they are always uni-cast.

The outcome of the simulation is collected through reports generated after the com-

pletion of the simulation. Report module of the simulator generates the reports based on the received message or connectivity events. Those reports contain different parameters concerned with the specific report type and can be post processed using external tools available for result processing. The GUI of the simulator depicts the visualization of different nodes, their location, active links between the nodes, message transfer activities. Graphical User Interface for ONE is shown in the figure 4.2.

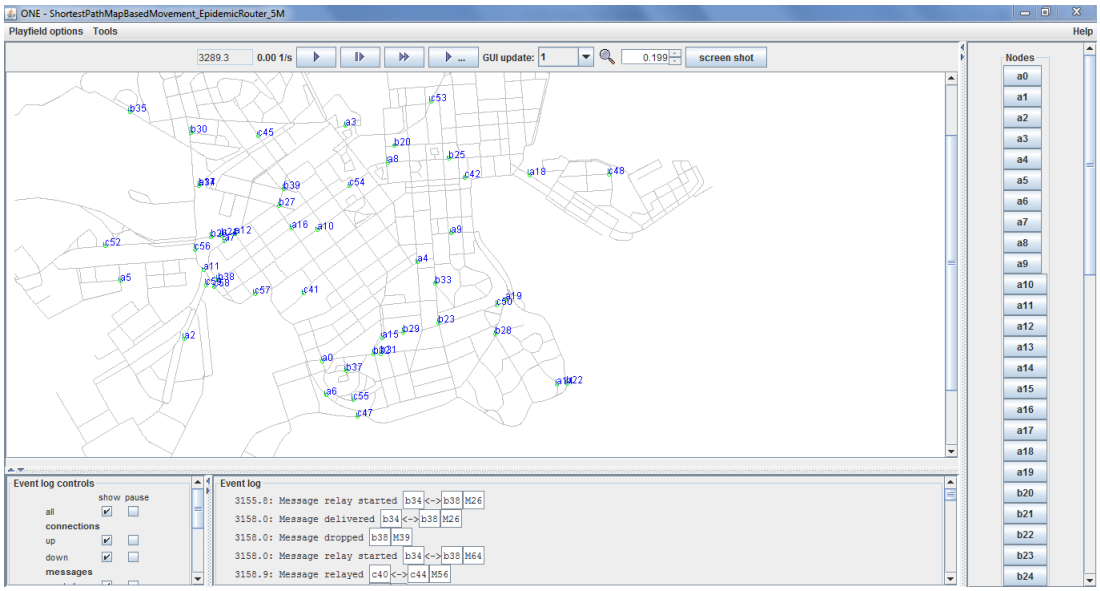


Figure 4.2: GUI Mode of ONE Simulator

Main modules for the ONE simulator tool are described as below:

1. **Node Characteristics:**

   Nodes are the most basic agents in the simulator. Each and every node in the network is capable of fulfilling the duties of a store-carry-forward router. Simulation scenarios can be simulated using the groups of nodes. Each group can be configured with different set of capabilities to simulate the scenario. And the parameters that are changeable are interface, buffer storage, movement models, energy consumption, message routing, no of nodes in the group and routing protocol. Some of the parameters can be easily configured using parameterization but more complex parameters like routing and movement modeling can only be done by using dedicated modules which implement the behavior of the particular parameter.

2. **Mobility Modelling:**

   Mobility models are used for implementing node movement capabilities. They define

algorithm and rules for a node to move in a particular region. There are mainly three type of synthetic movement models are included in the simulator.

1. Random Movement
2. Map Constrained Random Movement
3. Human behavior Based Movement

We can also create our own movement models in the simulator. And any external movement data or map (.wkt file) can be used for loading external movement data. Some popular movement models like Random Way point (RWP), Random Walk (RW), Shortest Path Map Based Movements, etc. are also included in the simulator.

3. **Routing:**

This module of the simulator includes rules and algorithms used for routing inside the network. There are mainly six well known routing protocols which are already implemented in the simulator. Those implemented routing protocols cover mainly all the scenarios of routing in the simulator i.e. single copy routing, multicopy routing, n-copy routing and estimation based routing protocols. And those are:

1.Direct Delivery (DD)

2.First Contact (FC)

3.Spray-and-Wait

4.PRoPHET

5.Max-Prop

6.Epidemic

These routing protocols have been described in detail in section 2.Other simulators like NS2s routing capabilities can also be used as an extension to the existing routing capabilities of the ONE simulator.Report Module can be used to model the results generated by the simulator. Many post processing tools and external scripts can be used to compare the results or for performing analysis on the outcomes generated from the simulated scenario.

4.4 Comparison of Simulators in the context of DTN

Comparison of both the simulator in the context of Delay Tolerant Networks is shown in the table4.4.

Table 4.1 : Simulator Comparison []

Comparison Parameter	**Network Simulator(NS)**	**Opportunistic Network Environment(ONE)**
Platform	Linux only	java,jdk installed any platform(windows,linux,Mac)
Programming Language	oTCL,C++	JAVA
Change compatibility	complex to build a new feature	easy to program new features because of less configurations
Real time map support	not supported	ability to load Map data and run simulations using that
Simulation output	it generates a NAM file or python based real time visualization package	generate reports which have different calculated parameter values.useful for post processing
DTN support	partial DTN support as it is general network simulator	full DTN support as it is used only for simulating DTNs

Chapter 5

Proposed Approach & Implementation

5.1 Problem Statement

Let us consider there are n number of mobile nodes in the network.The nodes will communicate with each other whenever they come in the range of the other nodes.So the contacts between the nodes are opportunistic in delay tolerant networks.If there are some nodes in the network,which have malicious or selfish purpose.So these kind of nodes can originate flooding attack to reduce the throughput or the delivery probability of the network.They can do so by intentionally instilling as many unnecessary packets as possible into the network.Which will ultimately waste the resources of the network and lower the throughput.As this kind of flood attack is most basic attack in resource target attacks, so we will focus only onto this attack.

5.2 Algorithm

5.2.1 Detection Strategy

For detecting the nodes which try to instill more packets than a decided limit, we use the rate limiting mechanism. In this mechanism if a node try to send more packets/bundles than a specified limit (threshold) that means it is trying to flood the network with more unnecessary packets. Which results into resource overuse.So those nodes should be detected and marked as illegitimate nodes. For defining the rate limit(threshold) value, we run the simulator under the normal scenario where there is no flooding in the network.

And then we will observe the number of messages relayed in the network. Average of those observation will result into the rate limiting threshold.

$$Threshold = \sum_{i=1}^{n} No.of relayed messages/n \tag{5.1}$$

Any node tries to send the packets more than the predefined threshold will be considered as an attacker.

5.2.2 Tolerance/Mitigation Scheme

Methodology:

Each and every node in the network maintains a collection map which stores (key,value) pair in it and a list which stores the black listed node's ids.Each node will store node's id as key and the packet count as value in the collection.

If any node A come in the range of the node B in the network at a particular time.Then they both exchanges their collections(Map) with each other.On every node encounter collection map is exchanged.And a node will never store its own information inside its own collection map. So the problem that a node will forge a count and claim a false count can never happen using this scheme.As on the exchange of collection map, each node increase the count for that particular encountered node and compares the received collection map with its own map and updates its own knowledge base of the network.

If the count of the encountered node is less than the predefined threshold value then it is considered to be a legitimate node. But if the count exceeds than the predefined threshold value then the encountered node will be considered as an illegitimate node. The node which has discovered such an illegitimate node, make an entry in a list of blacklisted nodes' table. Here we are also maintaining list of blacklisted node or bad nodes so that no bad node can falsely claim to be a good or legitimate node in the network.

Algorithm:

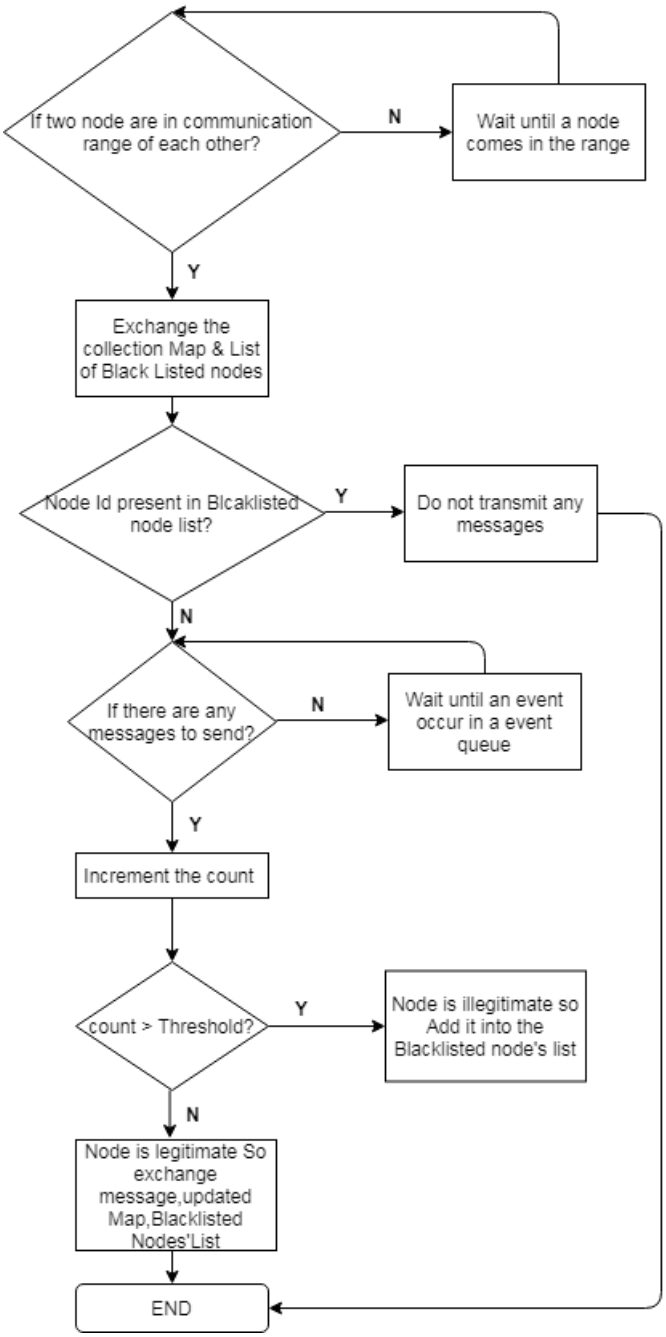


Figure 5.1: Flow chart of the proposed scheme

5.3 Performance Evaluation And Simulation Results

5.3.1 Simulation Setup & Evaluation Metrics

While providing security to DTN nodes, it affects bandwidth utilization cost and computational cost of the DTN nodes. The amount of bandwidth consumed and the amount of

computation required is depended on how we adjust the different affecting parameters.We implement our tolerance mechanism for flooding attacks in ONE simulator and evaluate its performance.In simulating an ideal scenario in the simulator we used 3 different node group containing 20 nodes per group resulting in 60 nodes in total. They are having transmission speed of 2Mbps. And the transmission range is 10 meters.These nodes are uniformly deployed in an area of 4500 meters by 3400 meters. All nodes are having Bluetooth interface.Then we induce the flood attack in the network to see the effect on the performance parameters. Initially we try to analyze the performance parameters by introducing 10-30% of the nodes as attackers.

After implementing the proposed algorithm, we try to test it under certain scenarios to test its flexibility under such situations.In the first scenario we compare the changes made by introducing our scheme in the performance parameters when the network was under the flooding attack.In the second scenario, we try to increase the number of attackers in the network from 30% to 60-70% of the total nodes.In the third scenario,we vary the number of hosts from 10 to 60 having same number of groups(i.e. 3). We implement our mechanism using Epidemic Routing protocol.The simulator parameters used in the evaluation are described in the table5.3.1for ONE simulator.

Table 5.1 Simulator Parameters Used in ONE

Parameter	**Value**
Simulation Time	43200 seconds
No of hosts	20
No of groups	3
Buffer Size	5M
Message TTL	300 minutes
Routing Protocol	Epidemic Routing
Message generation interval	25-35 seconds
Movement Model	Shortest Path Map Based Movement

Evaluation Parameters:

- **Delivery Probability:** Delivery probability is the fraction of generated messages

that are correctly delivered to the final destination.

$$DeliveryProbability = \frac{noofBundleDelivered}{no.ofBundlecreated} \tag{5.2}$$

- **Overhead Ratio:**The overhead ratio measures how many transfers were needed for each message delivery.

$$OverheadRatio = \frac{no.ofBundleRelayed - no.ofBundleDelivered}{no.ofBundleDelivered} \tag{5.3}$$

5.4 Simulation Results

5.4.1 Scenario 1:Effect of Proposed Scheme

If we compare the results of the scenario when there is no flooding present in the network with flooding present in the network, it is but obvious that the performance parameters will be affected.It is shown in the figures 5.2 & 5.3 below. Here we have used shortest path map based movement model with epidemic routing strategy in which each an every node has buffer capacity of 5M.As we can see in the figures that flooding attacks can degrade the network performance. In normal ideal scenario when there was no flooding in the network delivery probability is higher than the case when we induce flooding attacks in the network.

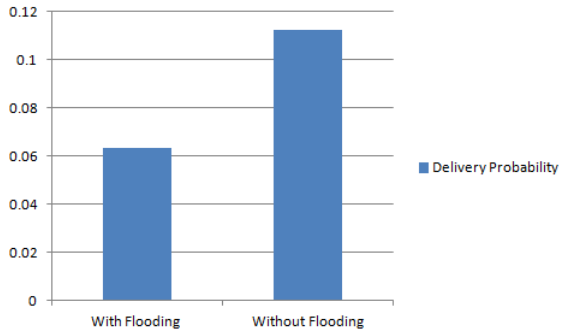


Figure 5.2: Comparison of Delivery Probability

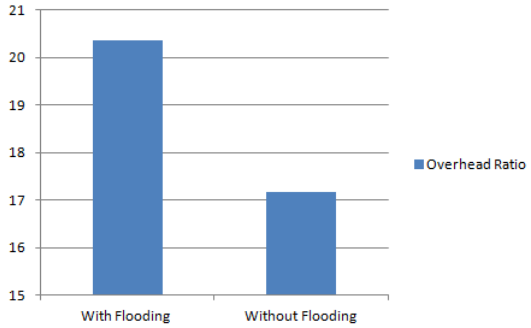


Figure 5.3: Comparison of Overhead Ratio

5.4.2 Scenario 2:Effect of Increasing Number Of Attacker Node

In this scenario of implementation, we have kept the number of host and number of groups as it was in the ideal scenario. We test the robustness of the proposed algorithm in presence of increased number of attackers.We vary the number of attackers from 10 to 40 and examine how this affects the network performance.

Figure 5.4 & 5.5 shows the comparison of the performance parameters like delivery probability and overhead ratio.As we can see from the graph that as the number of attackers increase delivery probability decreases for the case in which flooding attack is present in the network.And same way overhead ratio also increases which degrade the network performance. But if we see the effect of our approach implemented when the flooding was present in the network, there are slighter improvements in terms of delivery probability and overhead ratio.Thus in slighter high traffic scenario proposed approach shows the improvisation in performance parameters.

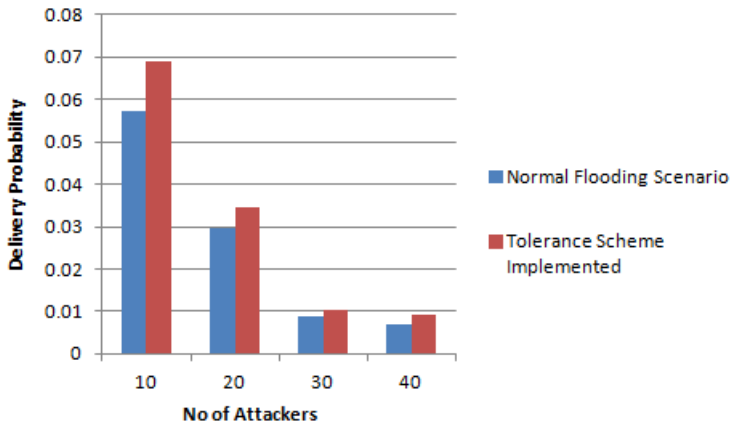


Figure 5.4: Comparison of Delivery Probability with increased number of attackers

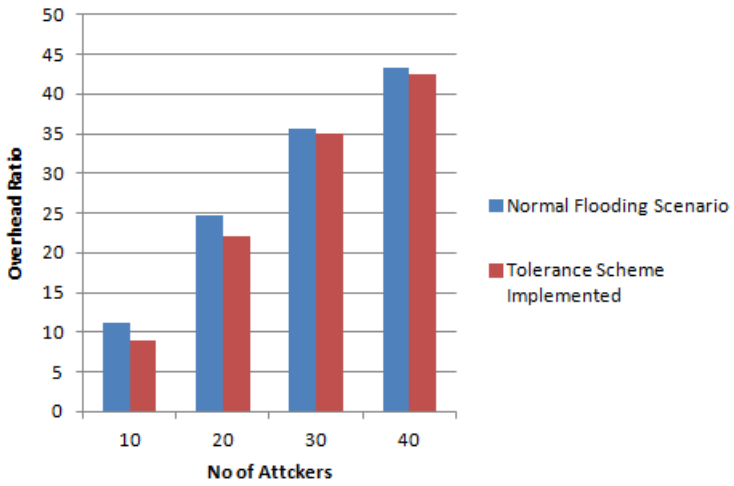


Figure 5.5: Comparison of Overhead Ratio with increased number of attackers

5.4.3 Scenario 3:Effect of Increasing Number Of Hosts

In this scenario of implementation, we have kept the number of attackers and number of groups as it is. We test the robustness of the proposed algorithm in presence of increased number of hosts.We vary the number of hosts from 15 to 60 per group and examine how this affects the network performance.

Figure 5.4 & 5.5 shows the comparison of the performance parameters like delivery probability and overhead ratio.If the number of hosts increases, traffic of the network also increases and thus in the scenario where flooding is present in the network, delivery probability decreases and overhead increases. As we can see from the graph that as the number of hosts increases delivery probability decreases for the case in which flooding attack is present in the network.And same way overhead ratio also increases which degrade the network performance. But if we see the effect of our approach implemented when the flooding was present in the network, there are noticeable improvements in terms of delivery probability and overhead ratio.

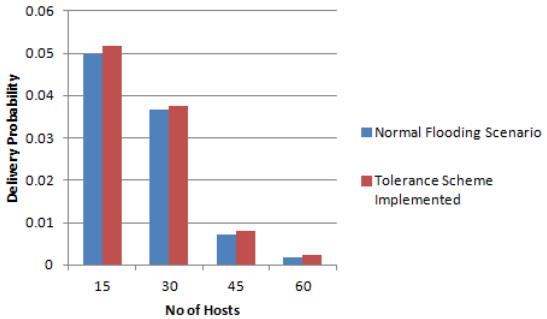


Figure 5.6: Comparison of Delivery Probability with increased number of hosts

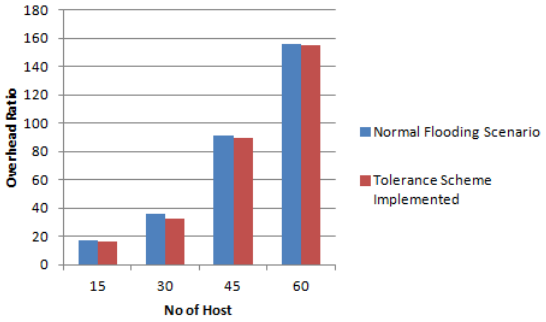


Figure 5.7: Comparison of Overhead Ratio with increased number of hosts

Chapter 6

Conclusion & Future Work

Security is the major issue in Delay Tolerant Network.Particularly flooding attacks cause network slow down and resources of the network are not utilized properly. Thus flooding attacks are a threat to network availability in DTN. The use of strong and complex security algorithms against flooding attacks exposes DTN nodes to a new threat called resource exhaustion attack. This type of attack depletes the scarce resources of the network and degrades performance.The proposed scheme provides DoS resilience against flood based DoS attacks.The proposed scheme is lightweight tolerance mechanism which is less computationally complex. We adopted the Epidemic routing in our approach.The main aim of the approach is to detect the illegitimate nodes in the network and tolerate the effect of the flood attack in the network.Tolerance of the effect of the flood attacks includes the improvisation of the performance parameters.We have shown through the results that the proposed approach is scalable as the number of attackers and the number of nodes increases in the network.We can extend this approach by providing support for tolerance of collaborative attacks.For collaborative attack, our scheme detect this attack after some part of the network has been affected by it.For which tolerance mechanism can not optimize the affected performance parameters.So we can work on that aspect in future.

Bibliography

[1] Y. Ren, M. C. Chuah, J. Yang, and Y. Chen, "Detecting wormhole attacks in delay-tolerant networks," *Wireless Commun.*, vol. 17, pp. 36–42, Oct. 2010.

[2] F. Li, J. Wu, and A. Srinivasan, "Thwarting blackhole attacks in disruption-tolerant networks using encounter tickets," in *INFOCOM 2009, IEEE*, pp. 2428–2436, IEEE, 2009.

[3] Q. Li, S. Zhu, and G. Cao, "Routing in socially selfish delay tolerant networks," in *INFOCOM, 2010 Proceedings IEEE*, pp. 1–9, IEEE, 2010.

[4] A. Haris, "A dtn study: Analysis of implementations and tools," Master's thesis, Technical University of Denmark, DTU, DK-2800 Kgs. Lyngby, Denmark, 2010.

[5] "Delay-tolerant networking." https://en.wikipedia.org/wiki/Delay-tolerant_networking.

[6] K. Fall, "A delay-tolerant network architecture for challenged internets," in *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, pp. 27–34, ACM, 2003.

[7] Q. Li, W. Gao, S. Zhu, and G. Cao, "To lie or to comply: Defending against flood attacks in disruption tolerant networks," *IEEE Transactions on Dependable and Secure Computing*, vol. 10, no. 3, pp. 168–182, 2013.

[8] Y. Ren, M. C. Chuah, J. Yang, and Y. Chen, "Detecting wormhole attacks in delay-tolerant networks [security and privacy in emerging wireless networks]," *IEEE Wireless communications*, vol. 17, no. 5, 2010.

[9] Y. Guo, S. Schildt, and L. Wolf, "Detecting blackhole and greyhole attacks in vehicular delay tolerant networks," in *Communication Systems and Networks (COMSNETS), 2013 Fifth International Conference on*, pp. 1–7, IEEE, 2013.

[10] Z. Zhang, "Routing in intermittently connected mobile ad hoc networks and delay tolerant networks: overview and challenges," *IEEE Communications Surveys & Tutorials*, vol. 8, no. 1, pp. 24–37, 2006.

[11] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Single-copy routing in intermittently connected mobile networks," in *Sensor and Ad Hoc Communications and Networks, 2004. IEEE SECON 2004. 2004 First Annual IEEE Communications Society Conference on*, pp. 235–244, IEEE, 2004.

[12] A. Vahdat, D. Becker, *et al.*, "Epidemic routing for partially connected ad hoc networks," 2000.

[13] T. Small and Z. J. Haas, "Resource and performance tradeoffs in delay-tolerant wireless networks," in *Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*, pp. 260–267, ACM, 2005.

[14] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Efficient routing in intermittently connected mobile networks: The multiple-copy case," *IEEE/ACM Transactions on Networking (ToN)*, vol. 16, no. 1, pp. 77–90, 2008.

[15] E. C. De Oliveira and C. V. De Albuquerque, "Nectar: a dtn routing protocol based on neighborhood contact history," in *Proceedings of the 2009 ACM symposium on Applied Computing*, pp. 40–46, ACM, 2009.

[16] L. R. Reddy and S. Raghavan, "Smort: Scalable multipath on-demand routing for mobile ad hoc networks," *Ad Hoc Networks*, vol. 5, no. 2, pp. 162–188, 2007.

[17] S. Biswas and R. Morris, "Exor: opportunistic multi-hop routing for wireless networks," *ACM SIGCOMM Computer Communication Review*, vol. 35, no. 4, pp. 133–144, 2005.

[18] E. P. Jones, L. Li, J. K. Schmidtke, and P. A. Ward, "Practical routing in delay-tolerant networks," *IEEE Transactions on Mobile Computing*, vol. 6, no. 8, pp. 943–959, 2007.

[19] D. Hua, X. Du, Y. Qian, and S. Yan, "A dtn routing protocol based on hierarchy forwarding and cluster control," in *Computational Intelligence and Security, 2009. CIS'09. International Conference on*, vol. 2, pp. 397–401, IEEE, 2009.

[20] J. Shen, S. Moh, and I. Chung, "Routing protocols in delay tolerant networks: A comparative survey," in *The 23rd International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC 2008)*, pp. 6–9, 2008.

[21] E. M. Daly and M. Haahr, "Social network analysis for routing in disconnected delay-tolerant manets," in *Proceedings of the 8th ACM international symposium on Mobile ad hoc networking and computing*, pp. 32–40, ACM, 2007.

[22] A. Lindgren, A. Doria, and O. Schelén, "Probabilistic routing in intermittently connected networks," *ACM SIGMOBILE mobile computing and communications review*, vol. 7, no. 3, pp. 19–20, 2003.

[23] V. Natarajan, Y. Yang, and S. Zhu, "Resource-misuse attack detection in delay-tolerant networks," in *Performance Computing and Communications Conference (IPCCC), 2011 IEEE 30th International*, pp. 1–8, IEEE, 2011.

[24] K. Ramaraj, J. Vellingiri, C. Saravanabhavan, and A. Illayarajaa, "Denial of service flood attacks in disruption tolerant networks,"

[25] P. T. N. Diep and C. K. Yeo, "Detecting flooding attack in delay tolerant networks by piggybacking encounter records," in *Information Science and Security (ICISS), 2015 2nd International Conference on*, pp. 1–4, IEEE, 2015.

[26] D. Kuriakose and D. Daniel, "Effective defending against flood attack using stream-check method in tolerant network," in *Green Computing Communication and Electrical Engineering (ICGCCEE), 2014 International Conference on*, pp. 1–4, IEEE, 2014.

[27] j. v. n. p. y. u. i. d. p. Ansa, Godwin and Cruickshank, Haitham and Sun, Zhili and Alshamrani, Mazin, title=A Security Scheme to Mitigate Denial of Service Attacks in Delay Tolerant Networks

[28] "ns2vsns3." http://wrc-ejust.org/crn/images/Tutorials/ns2vsns3.pdf.

[29] A. ur Rehman Khan, S. M. Bilal, and M. Othman, "A performance comparison of network simulators for wireless networks," *CoRR*, vol. abs/1307.4129, 2013.

[30] A. Keränen, J. Ott, and T. Kärkkäinen, "The one simulator for dtn protocol evaluation," in *Proceedings of the 2nd international conference on simulation tools and techniques*, p. 55, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2009.