Enhanced Channel Estimation and Traffic Monitoring for Misbehaviour Nodes in Disruption Tolerant Networks

M. Rubini[1], N. Tajunisha[2]

[1]Research Scholar, Department of Computer Science, Sri Ramakrishna College of Arts and Science for Women, Bharathiar University, Coimbatore, Tamilnadu, India

[2]Associate Professor, Department of Computer Science, Sri Ramakrishna College of Arts and Science for Women, Bharathiar University, Coimbatore, Tamilnadu, India

Abstract: ***In disruption tolerant networks (DTNs), the malicious nodes can be detected by watchdog and pathrater solutions. To address the problem of noise between the nodes, a distributed scheme has been used to detect the packet dropping in DTNs, where a node is used to keep a few signed contact records of its previous contacts, based on it the next contacted node is detected whether the node has dropped any packet. Since misbehaving nodes may misreport the contact records in order to avoid being detected, a small part of each of the contact record is disseminated to a certain number of witness nodes. A scheme to mitigate routing misbehaviour by limiting the number of packets forwarded to the misbehaving nodes is used. Thus the misbehaving nodes ensure low packet delivery, throughput, end to end latency and more energy consumption. So a Channel Aware Detection (CAD) algorithm is used to enhance the above metrics and to limit the traffic flowing to the misbehaving nodes. The CAD algorithm is used based on two strategies, the channel based estimation and traffic monitoring. If the monitored loss rate at particular hops exceeds the estimated normal loss rate, those nodes identified will be taken as attackers. The NS2 simulation shows that the solutions are efficient and effectively enhance routing misbehaviour.***

Keywords: Detection of misbehaving nodes, network security, routing with peer id.

1. Introduction

1.1 Networks

A computer network or data network is a telecommunication of networks that allows computers to exchange the data. The connections (network links) between networked computing devices (network nodes) are established using either a cable media or wireless media. The best known computer network is the Internet. Network devices that originate, route and terminate the data are called as network nodes. Nodes can include hosts such as servers and personal computers, as well as networking hardware. Two devices are said to be networked when a device is able to exchange information with another device.

1.2 Wired Networks

A wired network is the one in which all the components are connected by network cables. The wires are used as medium for communication between two or more nodes.

1.3 Wireless Network

Wireless network refers to any type of computer network that utilizes some form of wireless network connection. Wireless telecommunications networks are generally implemented and administered using radio communication.

1.4 Wireless Mesh Networks

A wireless mesh network **(WMN)** is a communication network made up of radio nodes organized in a mesh topology. Wireless mesh networks often consist of mesh clients, mesh routers and gateways. The mesh clients are often laptops, cell phones and other wireless devices while the mesh routers forward traffic to and from the gateways which may, but need not, connect to the Internet. The coverage area of the radio nodes working as a single network is sometimes called a mesh cloud. Access to this mesh cloud is dependent on the radio nodes working in harmony with each other to create a radio network. A mesh network is reliable and offers redundancy. When one node can no longer operate, the rest of the nodes can still communicate with each other, directly or through one or more intermediate nodes. A mesh network whose nodes are all connected to each other is a fully connected network. Mesh networks can be seen as one type of ad hoc network. Mobile ad hoc networks (MANET) and mesh networks are therefore closely related.

1.4.1 Architecture

Wireless mesh architectures is, in effect, a router network without the cabling between nodes. It's built of peer a radio device that does not have to be cabled to a wired port like traditional WLAN Access Points (AP) do. Mesh architecture sustains signal strength by breaking long distances into a series of shorter hops. Intermediate nodes not only boost the signal, but cooperatively make forwarding decisions based on their knowledge of the network, i.e. by performing routing. Such architecture may with careful design provide high bandwidth, spectral efficiency, and economic advantage over the coverage area. Example of three types of wireless mesh network:

- Infrastructure wireless mesh networks: Mesh routers form an infrastructure for clients.

- Client wireless mesh networks: Client nodes constitute the actual network to perform routing and configuration functionalities.
- Hybrid wireless mesh networks: Mesh clients can perform mesh functions with other mesh clients as well as accessing the network.

Wireless mesh networks have a relatively stable topology except for the occasional failure of nodes or addition of new nodes. The traffic, being aggregated from a large number of end users, changes infrequently. Practically all the traffic in an infrastructure mesh network is either forwarded to or from a gateway, while in ad hoc networks or client mesh networks the traffic flows between arbitrary pairs of nodes [1]

1.4.2 Management

This type of infrastructure can be decentralized (with no central server) or centrally managed (with a central server) both are relatively inexpensive, and very reliable and resilient, as each node needs only transmit as far as the next node. Nodes act as routers to transmit data from nearby nodes to peers that are too far away to reach in a single hop, resulting in a network that can span larger distances. The topology of a mesh network is also more reliable, as each node is connected to several other nodes. If one node drops out of the network, due to hardware failure or any other reason, its neighbours can find another route using a routing protocol.

1.4.3 Applications

Mesh networks may involve either fixed or mobile devices. The solutions are as diverse as communication needs, for example in difficult environments such as emergency situations, tunnels and oil rigs to battlefield surveillance and high speed mobile video applications on board public transport or real time racing car telemetry. A significant application for wireless mesh networks is VoIP. By using a Quality of Service scheme, the wireless mesh may support local telephone calls to be routed through the mesh. For example, minor safety has improved with VOIP phones communicating over a mesh network.

1.4.4 How Wireless Mesh Networks Works?

In a wireless mesh network, the network connection is spread out among dozens or even hundreds of wireless mesh nodes that "talk" to each other to share the network connection across a large area. Mesh nodes are small radio transmitters that function in the same way as a wireless router. Nodes use the common Wi-Fi standards known as 802.11 a, b and g to communicate wirelessly with users. Nodes are programmed with software that tells them how to interact within the larger network. Information travels across the network from point A to point B by hopping wirelessly from one mesh node to the next. The nodes automatically choose the quickest and safest path in a process known as dynamic routing**.**

In a wireless mesh network, only one node needs to be physically wired to a network connection like a DSL Internet modem. That one wired node then shares its Internet connection wirelessly with all other nodes in its vicinity. Those nodes then share the connection wirelessly with the nodes closest to them. The more nodes, further the connection spreads, creating a wireless "cloud of connectivity" that can serve a small office or a city of millions.

1.5 Disruption Tolerant Networks

Disruption Tolerant Networking (DTN) is a networking architecture that is designed to provide communications in the most unstable and stressed environments, where the network would normally be subject to frequent and long lasting disruptions and high bit error rates that could severely degrade normal communications. Disruption may occur because of the limits of wireless radio range, sparsity of mobile nodes, energy resources, attack, and noise.

DTN works using different kind of approach than TCP/IP for packet delivery that is more resilient to disruption than TCP/IP. DTN is based on a new experimental protocol called the Bundle Protocol (RFC 5050). BP sits at the application layer of some number of constituent internets, forming a store-and-forward overlay network where data is incrementally moved and stored throughout the network in hopes that it will eventually reach its destination.

The Bundle Protocol (BP) operates as an overlay protocol that links together multiple subnets (such as Ethernet-based LANs) into a single network. The basic idea behind DTN network is that endpoints aren't always continuously connected. In order to facilitate data transfer, DTN uses a store-and-forward approach across routers that are more disruption-tolerant than TCP/IP. However, the DTN approach doesn't necessarily mean that all DTN routers on a network would require large storage capacity in order to maintain end-to-end data integrity.

1.6 Problem Definition

To transfer the data packet from source to destination by which a user uses the TCP/IP protocol. In this process before transferring the data the connection has to be established where each node forwards the query to the next node. During the lookup process no information is sent back to the originator, resulting in fewer packets overhead. In order to enhance packet dropping detection and to limit the traffic flow to the misbehaving nodes, an efficient security routing algorithm is used.

1.7 Objective of the Research

The main objective of the research is to reduce the channel noise that is due to the misbehaving nodes. To enhance, the AODV routing protocol that is used to produce an efficient result based on channel estimation and traffic monitoring.

2. Related Work

In the last few researches, many methods has been summarized, J.Burgess et al [4] uses the MaxProp protocol for several mechanisms in concert to increase the delivery rate and lower latency of delivered packets. In addition, MaxProp uses acknowledgments sent to all peers to notify them of packet deliveries and assigns a higher priority to new packets, and it also attempts to prevent reception of the same packet twice. MaxProp unifies the problem of

scheduling packets for transmission to other peers and determining which packets should be deleted when buffers are low on space. The evaluations also examine MaxProp in simulated topologies shows that it performs well in a varied DTN environment.

E.Daly et al [6] describes that due to the complexity of the centrality metrics in populated networks, the concept of ego networks is exploited where nodes are not required to exchange information about the entire network topology, but only locally available information is considered. So the SimBet Routing is proposed which exploits the exchange of pre-estimated 'betweenness' centrality metrics and locally determined social 'similarity' to the destination node. They present simulations using real trace data to demonstrate that SimBet Routing results in delivery performance close to Epidemic Routing but with significantly reduced overhead. Additionally, Sim- Bet Routing outperforms PRoPHET Routing is shown, particularly when the sending and receiving nodes have low connectivity.

W.Gao el al [7] has proposed a novel approach to improve the performance of data forwarding with a short time constraint in DTNs by exploiting the transient social contact patterns. These patterns represent the transient characteristics of contact distribution, network connectivity and social community structure in DTNs, and provide analytical formulations on these patterns based on experimental studies of realistic DTN traces. An appropriate forwarding metrics based on these patterns to improve the effectiveness of data forwarding is proposed. When applied to various data forwarding strategies, the proposed forwarding metrics achieve much better performance compared to existing schemes with similar forwarding cost.

V.Erramilli et al [8] analyse two variants of delegation forwarding and show that while naive forwarding to high contact rate nodes has cost linear in the population size, the cost of delegation forwarding is proportional to the square root of population size. Delegation forwarding with different metrics using real mobility traces and show that delegation forwarding performs as well as previously proposed algorithms at much lower cost is studied. In particular the delegation scheme based on destination contact rate is shown.

N.Eagle et al [10] describes the data collected from mobile phones can be used to uncover the regular rules and structure in the behaviour of both individuals and organizations and thus captures all the information to which the phone has access to (with the exception of content from phone calls or text messages) and describes how it can be used to provide insight into both the individual and the collective. The applications we have presented include ethnographic studies of device usage, relationship inference, individual behaviour modelling, and group behaviour analysis.

J.Burgess et al [11] uses the connectivity traces from our UMass Diesel- Net project and the Haggle project to quantify routing attack electiveness on a DTN that lacks security and has introduced plausible attackers and attack modalities and provide complexity results for the strongest of attackers. Thus concluded that disruption-tolerant networks are extremely robust to attack; in trace-driven evaluations, an attacker that has compromised 30% of all nodes reduces delivery rates from 70% to 55%, and to 20% with knowledge of future events. By comparison, contemporaneously connected networks are significantly more fragile. In this paper a routing protocol has been proposed for a variety of attack strategies with related complexity results and introduced attack modalities with a defence for the most powerful. Using a comprehensive set of experiments, we have demonstrated that even in the worst case, of a very powerful attacker that has corrupted 20% of the nodes, a replicative DTN routing protocol still delivers 45% of all packets successfully, compared with 70% when no attackers are present.

U.Shevade el al [9] propose a Tit For Tat(TFT) mechanism that incorporates generosity and contrition to address the issues about , lack of end-to-end paths, high variation in network conditions, and long feedback delay in DTNs. In this paper, an incentive-aware routing protocol that allows selfish nodes to maximize their own performance while conforming to TFT constraints is developed. For comparison, the techniques to optimize the system-wide performance when all nodes are cooperative are developed. Using both synthetic and real DTN traces, we show that without an incentive mechanism, the delivery ratio among selfish nodes can be as low as 20% as what is achieved under full cooperation; in contrast, with TFT as a basis of cooperation among selfish nodes, the delivery ratio increases to 60% or higher as under full cooperation but the control – plane exchanges cannot be made with this technique.

S.Marti el al [12] proposed watchdog-based solutions in which the sending node operates in promiscuous mode and overhears the medium to check if the packet is really sent out by its neighbour. But in this the neighbourhood monitoring relies on a connected link between the sender and its neighbor, which most likely will not exist in DTNs. This is the drawback of this method. The Dynamic Source Routing (DSR) protocol is a simple and efficient routing protocol designed specifically for the use in multi-hop wireless ad hoc networks of mobile nodes. DSR allows network to be completely self-organizing and self-configuring, without the need of any existing network infrastructure. DSR has been implemented by numerous groups, and deployed on several environments. Networks using the DSR protocol have been connected to the Internet. DSR can interoperate with the Mobile IP, and nodes that use Mobile IP and DSR have seamlessly migrated between the WLANs, cellular services, and DSR mobile ad hoc networks.

The protocol is of two main mechanisms that are Route Discovery and Route Maintenance, which works together allowing the nodes to discover and maintain the routes to arbitrary destinations in the ad hoc networks. The protocol allows us with multiple routes to any of the destination and allows each of the senders to select and control the routes used in routing its packets. The other advantages of DSR protocol includes easy guaranteed loop-free routing, supports for use in networks containing unidirectional links, use of only "soft state" in the routing, and a very rapid recovery when routes in the network change by any cause. The DSR protocol is designed mainly for the mobile ad hoc networks for about two hundred nodes, and is designed that

it works well with even very high rates of mobility. Thus the results show that it can gain the benefits of an increased number of routing nodes while minimizing the effects of misbehaving nodes. It shows that this can be done without a prior trust or excessive overhead.

S.Buchegger el al [13] proposed a protocol, called CONFIDANT, for making misbehaviour unattractive; it is based on selective altruism and utilitarianism. It aims at detecting and isolating misbehaving nodes, thus making it unattractive to deny cooperation. Trust relationships and routing decisions are based on experienced, observed, or reported routing and forwarding behavior of other nodes. This paper recognizes the special requirements of mobile ad-hoc network in terms of cooperation, robustness, and fairness, and analyzes the performance of a scheme to cope with them by retaliating for malicious behaviour and warning affiliated nodes to avoid bad experiences. Nodes learn not only from their own experience, but also from observing their neighbourhood and from the experience of their friends. Observable attacks on forwarding and routing in mobile ad-hoc networks can be thwarted by the suggested CONFIDANT scheme of detection, alerting, and reaction. Performance analysis by means of simulation shows a significant improvement in terms of goodput when DSR is fortified with the CONFIDANT protocol extensions. The overhead for this increase is very low. The CONFIDANT protocol is scalable in terms of the total number of nodes in a network and performs well even with a fraction of malicious nodes as high as 60%.

K.Liu et al [14] proposed a 2ACK scheme in which the sending node waits for an ACK from the next hop of its neighbour to confirm that the neighbour has forwarded the data packet. However, this technique is vulnerable to collusions, i.e., the neighbour can forward the packet to a colluder which drops the packet. Although end-to-end ACK schemes are resistant to such colluding attacks, the ACK packets may be lost due to the opportunistic data delivery in DTNs. A scheme is used to detect packet dropping in DTNs. The detection scheme works in a distributed way; i.e., each node detects packet dropping locally based on the collected information. Moreover, the detection scheme can effectively detect misreporting even when some nodes collude. Analytical results on detection probability and detection delay were also presented. The proposed scheme is very generic and it does not rely on any specific routing algorithm. Trace-driven simulations show that our solutions are efficient and can effectively mitigate routing misbehaviour.

Y.Xue et al [15] propose a new routing service named best-effort fault-tolerant routing (BFTR). The design goal of BFTR is to provide packet routing service with high delivery ratio and low overhead in presence of misbehaving nodes. Instead of judging whether a path is good or bad, i.e., whether it contains any misbehaving node, BFTR evaluates the routing feasibility of a path by its end-to-end performance (e.g. packet delivery ratio and delay). By continuously observing the routing performance, BFTR dynamically routes packets via the most feasible path. BFTR provides an efficient and uniform solution for a broad range of node misbehaviours with very few security assumptions. The BFTR algorithm is evaluated through both analysis and extensive simulations. The results show that BFTR greatly improves the ad hoc routing performance in the presence of misbehaving nodes.

Q.Li et al [26] proposed a Social Selfishness Aware Routing (SSAR) algorithm to cope with user selfishness and provide good routing performance in an efficient way. To select an effective forwarding node, SSAR considers both users' willingness to forward and their contact opportunity, and derives a metric with mathematical modelling and machine learning techniques to measure the forwarding capability of the mobile nodes. Moreover, SSAR formulates the data forwarding process as a Multiple Knapsack Problem with Assignment Restrictions (MKPAR) to satisfy user demands for selfishness and performance. In the proposed work, the method used thus enhances the packet delivery, network throughput and controls traffic.

3. Simulation Environment

Network Simulator (Version 2), widely called as NS2, is simply an event driven simulation tool that has proved that it is useful in studying the dynamic nature of communication networks. Simulation of wired as well as wireless network functions and protocols (e.g., routing algorithms, TCP, UDP) can be done using NS2. In general, NS2 provides users with a way of specifying such network protocols and simulating their corresponding behaviours. NS2 provides users with executable command ns which take on input argument, the name of a Tcl simulation scripting file. Users are feeding the name of a Tcl simulation script (which sets up a simulation) as an input argument of an NS2 executable command ns. In most cases, a simulation trace file is created, and is used to plot graph and/or to create animation.

NS2 consists of two key languages: C++ and Object-oriented Tool Command Language (OTcl). While the C++ defines the internal mechanism (i.e., a backend) of the simulation objects, the OTcl sets up simulation by assembling and Configuring the objects as well as scheduling discrete events (i.e., a frontend). The C++ and the OTcl are linked together using TclCL. Mapped to a C++ object, variables in the OTcl domains are sometimes referred to as handles. Conceptually, a handle (e.g., n as a Node handle) is just a string (e.g., _o10) in the OTcl domain, and does not contain any functionality. Instead, the functionality (e.g., receiving a packet) is defined in the mapped C++ object (e.g., of class Connector). In the OTcl domain, a handle acts as a frontend which interacts with users and other OTcl objects. It may defines its own procedures and variables to facilitate the interaction.

4. Methodology

4.1 Channel Estimation and Traffic Monitoring Based On Channel Aware Detection (CAD) Algorithm

A Channel Aware Detection (CAD) algorithm can effectively identify the selective forwarding attackers by filtering out the normal channel losses. The CAD approach is based on two procedures, channel estimation and traffic monitoring. The procedure of channel estimation is to estimate the normal loss rate due to bad channel quality or medium access collision. The procedure of traffic

monitoring is to monitor the actual loss rate; if the monitored loss rate at certain hops exceeds the estimated loss rate, those nodes involved will be identified as attackers.

Specifically, the traffic monitoring procedure at each intermediary node along a path monitors the behaviours of both its upstream and downstream neighbours, termed as upstream monitoring and downstream monitoring, respectively. The channel estimation procedure at each node correspondingly sets an upstream detection threshold and downstream detection threshold.

Each node judges the behaviour of its neighbours by comparing the upstream/downstream observations against the detection thresholds to identify the misbehaving nodes. In particular, the thresholds will be dynamically adjusted with the normal loss rates to maintain the detection accuracy when network status changes. The contribution of CAD is:

- The channel estimation is integrated with traffic monitoring to achieve channel-aware detection of gray hole attack, which can effectively identifies selective forwarding misbehaviour hidden in the normal loss events due to bad channel quality or medium access collisions.
- In CAD, upstream and downstream traffic monitoring are combined to achieve a versatile detection method.
- Based on the analytical model, the optimal upstream/downstream detection thresholds can be computed to minimize the summation of false alarm and missed detection probabilities.
- The thresholds are dynamically adjusted with the channel status to maintain the efficiency of Channel Estimation under varying network condition.
- A path with trustworthy source and destination nodes should always be considered. It is also assumed that the communication on every link between the mesh nodes is *bidirectional*.
- Buffer of infinite size and a packet can be dropped due to bad channel quality, medium access collision, or presence of an attacker.

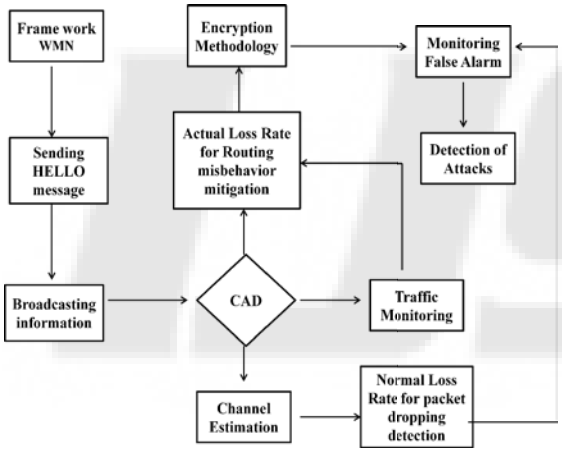

Figure 1: Flow Diagram of the process of Detection

To enhance the process of detection, a part of the CAD algorithm is introduced (i.e. channel estimation and traffic monitoring). The process of estimation is as follows:

- When a node receives a packet from the upstream, it updates the packet count history with the corresponding packet sequence number.
- Each witness node is given a sequence number and it will be recorded in the contact record.
- When an unauthorized node enters with a new sequence number that is not in the contact record, then the node is recognized as misbehaving node and checks for the conditions.
- When a router forwards a packet to the downstream node, it performs two operations: (i) For each packet relayed to the downstream, it buffers the link layer acknowledgments. (ii) It also overhears downstream traffic and determines whether the node forwarded or tampered the packet.
- For instance, when node A forwards a packet to B, it maintains the acknowledgment returned by B and overhears whether B tampered or forwarded the packet. Based on these observations, each node maintains a **probability of distrust (P_{dt})**

$$P_{dt} = \frac{N_t + N_d}{N_f}$$

N_t - Number of packets tampered.
N_d - Number of packets dropped.
N_f- Total number of packets delivered to downstream node

- **Downstream Detection Threshold** is taken as $\mathbf{T_d}$ for comparing the distrust value with the threshold.

$$Downstream\ Detection = \begin{cases} 1, if\ P_{dt} > T_d\ (Misbehaving\ node) \\ 0, if\ P_{dt} \le T_d\ (Normal\ node) \end{cases}$$

- **Loss Rate over Link (LRL)** is calculated as,

$$LRL = 1 - \frac{Packets\ received\ by\ C\ from\ B}{Packets\ received\ by\ B\ from\ A}$$

- **Upstream Detection Threshold** is taken as T_u,

$$Upstream\ Detection = \begin{cases} 1, if\ LRL > T_u\ (Misbehaving\ node) \\ 0, if\ LRL \le T_u\ (Normal\ node) \end{cases}$$

Thus the above conditions are checked for the detection of malicious nodes during the process of forwarding messages to the authorized users.

4.2 Elliptic Curve Digital Signature Algorithm

The Elliptic Curve Digital Signature Algorithm (ECDSA) is the elliptic curve analogue of the DSA. At each node, the message is attached with a 56-bytes ECDSA signature generated with a 28-bytes (224-bits) key. The ECDSA signature can protect the message from being tampered. Moreover, in order to prevent the replay attack, each source node further incorporates a nonce random number ηS to generate the signature for the first message $M1$ attached with a PROBE packet, and the corresponding destination node stores the nonce number having been used. When the destination (or gateway) receives the PROBE message, it first retrieves the ID of the last hop node, say, vn and uses the corresponding public key to verify $SIGNvn$. If $SIGNvn$ is correct, it then retrieves the ID of the upstream node of $vn-1$ and verifies the $SIGNvn-1$. The destination node continues this process until it verifies all the signatures or it finds an incorrect signature. Once all the signatures are verified, the destination node D builds a list of suspicious nodes based on the downstream/upstream opinions, a kind of reputation, marked by each node in the forwarding path. By

using ECDSA the message that are forwarded from source to destination are being encrypted / decrypted and verified.

5. Evaluation Results

This section focuses on the results and its analysis based on the simulation performed in NS2. The various analysis of performance parameters are given based on a dense wireless mesh network of 99 nodes which is simulated in a field with 1000m X 1000m area with a duration of 100s. During each simulation 12 Constant Bit Rates (CBR) connections are generated, producing 4 packets per second with packet size of 512 bytes.

5.1 Packet delivery Ratio

Packet delivery ratio (PDR) measures the percentage of data packets generated by nodes that are successfully delivered, expressed as

$$\text{PDR} = \frac{\text{TOTAL NUMBER OF PACKETS SUCCESFULLY DELIVERED}}{\text{TOTAL NUMBER OF PACKETS SENT}} \text{X100\%}$$

Figure 2: Packet delivery with increased ratio

5.2 Energy Consumption

Energy Consumption (EC) measures the energy expended per delivered data packet. It is expressed as

$$\text{EC} = \frac{\sum \text{ENERGY EXPENDED BY EACH NODE}}{\text{TOTAL NUMBER OF PACKETS DELIVERED}}$$

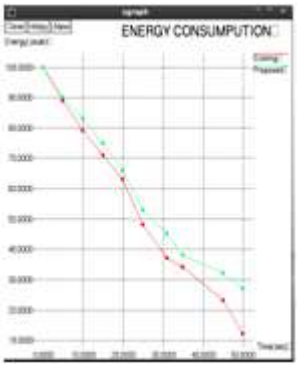

Figure 3: Energy Consumption with increased time

5.3 Network Throughput

Network Throughput (NT) is defined as the number of packets received at destination side at a particular time. It means

$$\text{NT} = \frac{\text{NUMBER OF PACKET RECEIVED}}{\text{TIME(Sec)}}$$

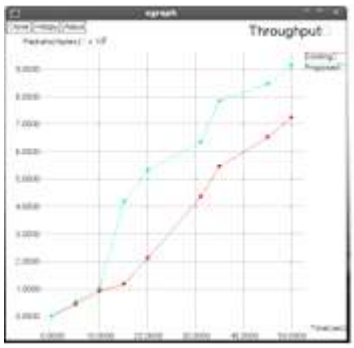

Figure 4: Network Throughput at a particular time

5.4 End-to-End Latency

End-End Latency (EEL) measures the average time it takes to route a data packet from the source node to the hub. It is expressed as

$$\text{EEL} = \frac{\sum \text{INDIVIDUAL DATA PACKET LATENCY}}{\text{TOTAL NUMBER OF PACKETS DELIVERED}}$$

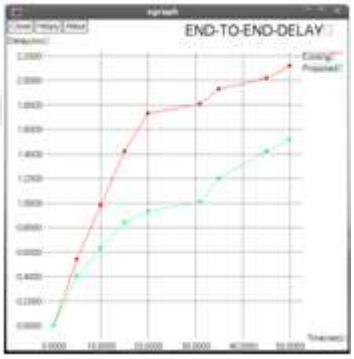

Figure 5: End to End Latency with Time

6. Conclusion

In Disruption Tolerant Networks (DTNs), the malicious node tampers original messages and misreports the destination. To detect the misbehaving nodes, the CAD based schemes are being introduced and enhanced the way of detection. Thus providing security and improving the quality of channels other than normal losses. NS2 simulation shows that our proposed work more efficient for the cause of attackers in daily process transmitting messages with improved network throughput, packet delivery, and reduces end to end delay, energy consumption.

7. Future Enhancements

In future work, with the increasing rate of threshold values one can detect the misbehaving nodes by using the technique of anomaly based intrusion detection algorithm. Thus it can be applied for the metrics of detection probability, false alarm rate, detection delay and their robustness to different detection threshold values.

References

[1] J. Jun, M.L. Sichitiu, "The nominal capacity of wireless mesh networks", in IEEE Wireless Communications, vol 10, 5 pp 8-14. October 2003.

[2] S.M. Chen, P, Lin, D-W Huang, S-R Yang, "A study on distributed/centralized scheduling for wireless mesh network" in Proceedings of the 2006 International Conference on Wireless Communications and Mobile Computing, pp 599 - 604. Vancouver, British Columbia, Canada. 2006.
[3] K. Fall, "A delay-tolerant network architecture for challenged internets," in *Proc. SIGCOMM*, 2003, pp.27–34.
[4] J. Burgess, B. Gallagher, D. Jensen, and B. Levine, " Maxprop: Routing for vehicle-based disruption-tolerant networks," in *Proc.IEEE INFOCOM*, 2006, pp. 1–11.
[5] W. Gao and G. Cao, "User-centric data dissemination in disruption tolerant networks," in *Proc. IEEE INFOCOM*, 2011, pp. 3119–3127.
[6] E. Daly and M. Haahr, "Social network analysis for routing in dis- connected delay-tolerant manets," in Proc. ACM MobiHoc, 2007, pp. 32–40.
[7] W. Gao and G. Cao, "On exploiting transient contact patterns for data forwarding in delay tolerant networks,"in Proc.IEEEICNP,2010,pp. 193–202.
[8] V. Erramilli, A. Chaintreau,M. Crovella, and C. Diot, "Delegation forwarding," in *Proc. ACM MobiHoc*, 2008, pp. 251–260.
[9] U. Shevade, H. Song, L. Qiu, and Y. Zhang, "Incentive-aware routing in dtns," in Proc. IEEE ICNP, 2008, pp. 238–247.
[10] N. Eagle and A. Pentland, "Reality mining: Sensing complex social systems," *Pers. Ubiquitous Comput.*, vol. 10, no. 4, pp. 255–268, 2006.
[11] J. Burgess, G. D. Bissias, M. Corner, and B. N. Levine, "Surviving attacks on disruption-tolerant networks without authentication,"in Proc. ACM MobiHoc, 2007, pp. 61–70.
[12] S. Marti, T. J. Giuli, K. Lai, and M. Baker, "Mitigating routing misbehavior in mobile ad hoc networks," in *Proc. ACM MobiCom*, 2000, pp. 255–265.
[13] S. Buchegger and Y. L. Boudec, "Performance analysis of the confidant protocol (cooperation of nodes: Fairness in dynamic ad-hoc networks),"in *Proc. MobiHoc*, 2002, pp. 226–236.
[14] K. Liu, J. Deng, P. K. Varshney, and K. Balakrishnan, "An acknowledgment-based approach for the detection of routing misbehavior in MANETs," *IEEE Trans. Mobile Comput.*, vol. 6, no. 5, pp. 536–550, May 2007.
[15] Y. Xue and K. Nahrstedt, "Providing fault-tolerant ad-hoc routing service in adversarial environments," *Wireless Pers. Commun.*, vol. 29, no. 3-4, pp. 367–388, 2004.
[16] C.M Barushimana, A.Shahrabi, "Comparative Study of Reactive and Proactive Routing Protocols Performance in Mobile Ad-Hoc Networks," Workshop on Advance Information Networking and Application, Vol. 2, pp. 679-684, May, 2003.
[17] M.Abolhasan, T.Wysocki, E.Dutkiewicz, "A Review of Routing Protocols for Mobile Ad-Hoc Networks," Telecommunication and Information Research Institute University of Wollongong, Australia, June, 2003.
[18] B. Awerbuch, D. Holmer, C.N. Rotaru, and H. Rubens, "An on-de- mand secure routing protocol resilient to byzantine failures," in Proc. ACM WiSe, 2002, pp. 21–30.
[19] P. Hui, J. Crowcroft, and E. Yoneki, "Bubble rap: Social-based forwarding in delay tolerant networks," in *Proc. ACM MobiHoc*, 2008, pp. 241–250.
[20] http://www.netmeister.org/misc/zrp/zrp.html#SECTION 00041000000000000000,
[21] http://www.faqs.org/rfcs/rfc3561.html
[22] http://www.faqs.org/rfcs/rfc3626.html
[23] F. Li, A. Srinivasan, and J. Wu, "Thwarting blackhole attacks in distruption-tolerant networks using encounter tickets," in *Proc. IEEE INFOCOM*, 2009, pp. 2428–2436.
[24] C.E.Perkins and E.M.Royer, "Ad-Hoc On Demand Distance Vector Routing," Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applictions, pp.90-100, Feb, 1999.
[25] Y. Ren, M. C. Chuah, J. Yang, and Y. Chen, "Detecting wormhole attacks in delay tolerant networks," *IEEE Wireless Commun.Mag.*, vol.
[26] Q. Li, W. Gao, S. Zhu, and G. Cao, "A routing protocol for socially selfish delay tolerant networks," in Ad Hoc Networks ,Aug.2011 ,DOI: 10.1016/j.adhoc. 2011.07.007
[27] Zhu, C. Lee, M.J.Saadawi, T., "RTT-Based Optimal Waiting time for Best Route Selection in Ad-Hoc Routing Protocols," IEEE Military Communications Conference, Vol. 2, pp. 1054-1059, Oct, 2003.
[28] Devu Manikantan Shila, *Student Member, IEEE*,Yu Cheng, *Senior Member,IEEE,* and Tricha Anjali,*Senior Member,IEEE* " Mitigating Selective Forwarding Attacks with a Channel-Aware Approach in WMNs" *IEEE* transactions on wireless communications, vol. 9, no. 5, may 2010.
[29] D. Johnson, A. Menezes, and S. Vanstone, "The Elliptic Curve Digital Signature Algorithm (ECDSA)," International J. Inf. Security, vol. 1, no. 1, pp. 36-63, Aug. 2001
[30] D.Manikantan Shila, Y. Cheng, and T. Anjali, "Channel-aware detection of gray hole attacks in wireless mesh networks," in Proc. GLOBECOM, 2009, submitted.
[31] ANSI X9.62 Public Key Cryptography for the Financial Services Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA), 1999.
[32] M.Bellare, R.Canetti and H. Krawczyk, "A modular approach to the design and analysis of authentication and key exchange protocols", Proceedings of the 30th Annual ACM Symposium on the Theory of Computing, 1998.
[33] I.D.Chakeres and E.M.Belding-Royer, "AODV routing protocol implementation design," in Proc. International Workshop on Wireless Ad Hoc Networking (WWAN), Tokyo, Japan, Mar. 2004.

Author Profile

Rubini Muthukrishnan received the Bachelor's degree in Computer Application from Bharathiar University in 2010.She received the Master's degree in Computer Science from Bharathiar University in 2012.

Dr. N. Tajunisha working as Associate Professor in Sri Ramakrishna College of Arts and Science for Women, Bharathiar University, Coimbatore, Tamilnadu. She has guided several PG and Research projects. She has presented her papers in International Conferences and has published papers in International Journals.