International Journal of Advanced Information Science and Technology (IJAIST) ISSN: 2319:268
Vol.2, No.3, March 2013 DOI:10.15693/ijaist/2013.v2i3.59-64

Ant Net based Efficient Load Balance-Aware Routing for Wireless Networks

K. Karthick[1] M. Thalaimalaichamy[2]

***Abstract*— Our propose works a load-aware routing scheme for wireless networks (WNs). In a WMN, the traffic load tends to be unevenly distributed over the network. In this situation, the load-aware routing scheme can balance the load, and consequently, enhance the overall network capacity. We design a routing scheme which maximizes the utility, i.e., the router is a primary component in the infrastructure of today's internet and are consequently are active target for the attackers. The Load balancing is difficult due to dropping, diverting, delaying or manipulating the packets and thereby mount denial of service, surveillance or man in the middle attach. In this paper. We detect the existence of compromised routers and isolate them from the routing fabric by the degree of user satisfaction, using mobile agents call the Ant-Nets. It is a distributed mobile agent that was inspired by the works on the ant colony metaphor for solving optimization problems. In this paper, when the packets moved from one AP to another AP using default path , if any problem occur in any AP, the AntNet is solved that existence problems to transmit the another shortest path . From that we can conclude avoid the packet loss and reduce the reception time to achieve the load optimization global load balancing. We present the graphical l results showing that the proposed scheme effectively balances the traffic load and outperforms the routing forms using the expected transmission time (ETT) as a routing metric.**

Keywords* – Ant Colony Optimization (ACO), *Expected transmission time (*ETT), *Quadratic assignment problem (QAP).

I INTRODUCTION

Sharing of files from source to the destination is often referred as "file sharing" in networking. The router is the primary component, which is used to transfer all such files. While transferring the files, these routers are compromised by the attackers and hence it becomes malicious in nature. Therefore there arrives a problem in the delivery of files, because of these malicious routers.
So we aim in detecting the existence of compromised routers and isolate them from the routing fabric by using the mobile agents called as Ant-Nets.

There are two main types of network categories which are:

- Server based
- Peer-to-peer

In a server based network, there are computers set up to be primary providers of services such as file service or mail service. The computers providing the service are called servers and the computers that request and use the service are called client computers.

In a peer-to-peer network, various computers on the network can act both as clients and servers. For instance, many Microsoft Windows based computers will allow file and print sharing. These computers can act both as a client and a server and are also referred to as peers. Networks are combination of peer-to-peer and server based networks. The network operating system uses a network data protocol to communicate on the network to other computers. The network operating system supports the applications on that computer. A Network Operating System (NOS) includes Windows NT, Novell Netware, Linux, UNIX and others.

Network Simulator is an open-source simulation tool that runs on Linux. It is a discreet event simulator targeted at networking research and provides substantial support for simulation of routing, multicast protocols and IP protocols, such as UDP, TCP, RTP and SRM over wired and wireless (local and satellite) networks. It has many advantages that make it a useful tool, such as support for multiple protocols and the capability of graphically detailing network traffic Network simulators, as the name suggests are used to simulate and then analyze the effect of various parameters on the network performance. Network simulator began as a variant of the REAL network simulator. The results of the simulation are viewed by Network Animator (Nam). Nam is a Tcl based animation tool for viewing network simulation traces and real world packet traces. It supports topology layout, packet level animation, and various data inspection tools.

II RELATED WORKS

The router is a primary component in the

infrastructure of today's Internet and are consequently are attractive targets for the attackers. A router can be compromised, making it malicious in nature.

Threats proposed by compromised routers:

1. Attack on control plane: The attacker may attack by means of the routing protocol (for example, by sending false advertisements).
2. Attack on data plane: The router violates the forwarding decisions it should make based on its routing tables.

It can be thought of as an instance of anomalous behavior-based intrusion detection. That is, a compromised router can potentially be identified by correct routers when it deviates from exhibiting expected behavior. This problem can be broken into three distinct sub problems.

Traffic information is the basis of detecting anomalous behavior: given traffic entering apart of the network, and an expected behavior for the routers in the network (i.e. a known routing configuration), anomalous behavior is detected when the monitored traffic leaving one part of the network differs significantly from what is expected. However, implementing such validation practically can be quite tricky and requires tradeoffs between the overhead of monitoring, communication and accuracy. The first problem we address is traffic validation: what information is kept about packet traffic and how it is used to determine that a router has been compromised. We assume that a compromised router can arbitrarily alter the forwarding behavior of that router. For example, a compromised router can drop or modify selected (or all) packets, or divert them to other routers. However, given the distributed nature of packet forwarding, such bad behavior can be detected. For example, suppose traffic traverses a path _r1, r2, r3_ and router r2 modifies this traffic. If routers r1 and r3 are not compromised, then one could simply compare what r1 sent to r2 and what r3 received from r2 to detect r2's behavior.

At an abstract level, we represent traffic validation mechanisms as a predicate TV (π, info (ri, π, τ), info (rj, π, τ)) where:

• π is a path segment _r1, r2, . . . rx_ whose traffic is to be validated between routers ri and rj ϵ π;
• Info(r, π, τ) is information about traffic that router r forwarded to along π over some time interval τ.

If routers ri and rj are not faulty, then TV(π,info(ri, π, τ), info(rj, π, τ)) evaluates to false if π contains a router that was faulty in forwarding traffic along π during τ. Implementing a traffic validation mechanism is a tricky engineering problem. The simplest, and most precise, representation of info(r, π, τ) is a complete copy of the packets sent, with each packet tagged with the time it was forwarded.

However, the storage requirements to buffer these packets and the bandwidth consumed by resending them, make this approach impractical at best. In practice, implementing traffic validation is a tradeoff between accuracy and overhead. For example, sampling can be used to decrease overhead, but depending on how sampling is done and the duration of the attack, accuracy may be reduced. The overhead-accuracy tradeoff also depends on what limits one might place on the kind of bad behavior a compromised router can exhibit.

Similarly, TV could be implemented simply using equality; info (ri, π, τ) = info (rj, π, τ). However, real networks occasionally lose packets due to congestion, reorder packets due to internal multiplexing, and corrupt packets due to interface errors. Consequently, TV needs to be somewhat more sophisticated to accommodate this abnormal, but non malicious behavior. Thus, implementing traffic validation involves striking a balance between the acceptable number of false positives and false negatives. Note that we have already made one engineering decision: we analyze aggregate traffic rather than individual packets. This is in contrast to some prior work. Doing so amortizes the monitoring overhead over many packets; if aggregation is not done then the computation and storage overhead is prohibitively large. It also makes it feasible to apply a threshold mechanism to distinguish between acceptable bad behavior (e.g. small amounts of packet loss and reordering) and malicious behavior.

Network routers occupy a key role in modern data transport and consequently are attractive targets for attackers. By manipulating, diverting or dropping packets arriving at a compromised router, an attacker can trivially mount denial-of-service, surveillance or man-in-the-middle attacks on end host systems.

We have proposed an approach to detect and isolate the compromised routers using the mobile agents called the AntNet. AntNet is a routing protocol for packet switched networks. It is an agent based routing algorithm based on Ant Colony Optimization (ACO).

III PROBLEM SOLUTIONS

Ants are able to find the best way to and from food source, in spite of being practically blind. Ants deposit a chemical substance called pheromone for marking path from food source to nest. By sensing pheromone trails, foragers can follow the path to food discovered by other ants. Shortest routes have high concentration of pheromone and almost all ants end up using this path. This behavior demonstrates collective and unsupervised learning without any centralized control. This simple but effective collective trail-laying

and trail-following behavior is the inspiring source of ACO. The idea behind ant algorithms is to simulate artificial stigmergy to coordinate societies of artificial ants. Initially, the ants choose one of the available paths randomly because of absence of pheromone trail on any of the paths. However, the ants that choose shorter paths, take less time to traverse it and hence pheromone deposition on the shorter paths occur earlier than the longer ones. Ants which arrive after pheromone deposition on shorter path has occurred and before ants on longer path have completed their journey prefer to choose shorter path because of higher pheromone concentration on it. More number of ants on shorter path further increases the rate of pheromone deposition on that path. Cumulatively, over time this results in highest pheromone concentration on best path and finally all the ants travel through that path only.

Ant Algorithm Characteristics

An ant algorithm presents the following characteristics:

• Natural algorithm since it is based on the behavior of real ants in establishing paths from their colony to source of food and back.
• Parallel and distributed since it concerns a population of agents moving simultaneously, independently and without a supervisor.
• Cooperative since each agent chooses a path on the basis of the information, pheromone trails laid by the other agents, which have previously selected the same path. This cooperative behavior is also autocatalytic, i.e., it provides a positive feedback, since the probability of choosing a path increases with the number of agents that previously chose that path.

•Versatile that it can be applied to similar versions of the same problem; for example, there is a straightforward extension from the traveling salesman problem (TSP) to the asymmetric traveling salesman problem (ATSP).
• Robust that it can be applied with minimal changes to other combinatorial optimization problems such as quadratic assignment problem (QAP) and the job-shop scheduling problem (JSP).

Positive feedback refers to a mechanism that allows colonies of ants to converge toward finding short paths from the anthill to a food source. It is created by stigmergic communication between the ants. Ants start moving randomly, at certain point an ant could find a food source and start to move back to the anthill (some ant species seems to always know the path back to the anthill). Other ants – still moving around randomly – smell the trail and find the food source and also return to the anthill, following almost the same path. This increase in pheromone value recruits even more ants that start to follow the path, thus the positive feedback mechanism ensures that almost all ants follow the path, and thus the ants have found a path to the anthill.

Examples of positive feedback include recruitment and reinforcement. For instance, recruitment to a food source is a positive feedback that relies on trail laying and trail following in some ant species. In the context of foraging, the problem is to find the shortest path from the anthill to the food source. In certain species, ants deposit pheromone trails while moving between a discovered food source and the anthill. Positive feedback is achieved because ants can smell deposited pheromone and have a natural tendency to follow the trail laid out by other ants. These ants again deposit pheromone on the same path, reinforcing the already existing pheromone trail, which again enhances the probability of other ants to discover the trail, thus yielding an ever-higher positive feedback as shown in following figure.

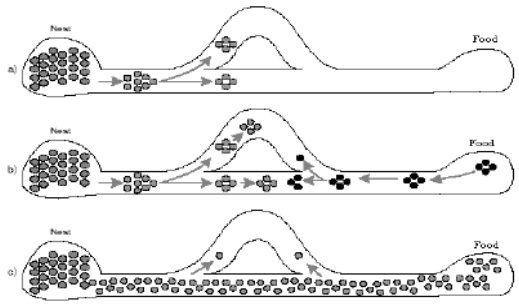

Fig.1 Positive Feedback Mechanism

Ant-Net Algorithms:

This section describes ant algorithm in detail. What has not been considered so far is how the probabilities in the routing tables are computed and updated. In the nature, ants lay pheromone and so they produce pheromone trails between the nest and a food source. On a computer, the pheromone has been replaced by artificial stigmergy, the probabilities in the routing tables. To compute and update the probabilities, intelligent agents are introduced to replace the ants.

There exist two kinds of agents, the forward agents and the backward agents. All forward and backward agents have the same structure. The agents move inside the network by hopping at every time step from a node to the next node along the existing links. The agents communicate with each other in an indirect way by concurrently reading and writing the routing tables on their way. The forward and backward agents receive percepts from the environment so they can do certain actions. Condition action rules are used to define what action should be chosen under which situation. The current situation is defined by the percept and the stored

internal state. A description of both agent types is given in Table 3-1. The condition action rules of the agents are defined in Table 3-2.

The internal state of the agents is a kind of memory which stores a list of (k,tk) pairs. Every such pair represents a node that has been visited by the forward agent. k is the identifier of the visited node and tk is the time a packet takes to travel the link from the last visited node to this node under the current traffic situation.

Updating Routing Tables

Updating in routing tables at each node will be done by backward ants using ants trip times. As shown in following figure, at every visited node k on the way back to the source node, a backward ant updates some of the probabilities in the routing table of node k by using the travel information in its memory which was collected by the forward ant.

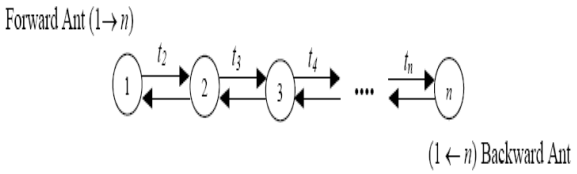

Fig. 2 Updating Routing Table by Backward Ants

The Ant-Net algorithm can be informally described as follows.

(i)At regular intervals, from every network node s, a forward ant Fs_d, is launched, with a randomly selected destination node d. Destinations are chosen to match the current traffic patterns.

(ii)Each forward ant selects the next hop node using the information stored in the routing table. The next node is selected, following a random scheme, proportionally to the goodness (probability) of each not still visited neighbor node and to the local queues status. If all neighbors have been already visited a uniform random selection is applied considering all the neighbors.

(iii)The identifier of every visited node k and the time elapsed since its launching time to arrive at this k-th node are pushed onto a memory Stack Ss_d(k) carried by the forward ant.

(iv)If a cycle is detected, that is, if an ant is forced to return to an already visited node, the cycle's nodes are popped from the ant's stack and all the memory about them is destroyed.

(v)When the ant Fs_d reaches the destination node d, it generates a backward ant Bd_s, transfers to it all of its memory, and then dies.

(vi)The backward ant makes the same path as that of its corresponding forward ant, but in the opposite direction. At each node k along the path it pops its stack Ss_d (k) to know the next hop node.

(vii) Arriving in a node k coming from a neighbor node f, the backward ant updates Mk and the routing table for all the entries corresponding to every node i on the path k_d, that is the path followed by ant Fk_d starting from the current node k. The sample means and variances of the model Mk (µi, _i2) are updated with the trip times Tk_i stored in the stack memory Ss_d(k).The routing table is changed by incrementing the probabilities associated with node f and the nodes i, and decreasing (by normalization) the probabilities Pin associated with the other neighbor nodes n. Trip times Tk_i experienced by the forward ant Fs_d are used to assign the probability increments.

IV RESULTS AND DISCUSSIONS

SIMULATION PARAMETERS

Table 1 Simulation Parameters and values

Parameter	Values
Scenario Size	500 m*500 m
Nodes	Generated 49 nodes randomly Generated 1 gateway on the corner
Transmission Range	250 m
Carrier Sense Rage	550 m
Flows	3 CBR flows
CBR Rate	1 Mbps
Packer Size	1500 bytes
δ	4 sec
Simulation Time	600 sec

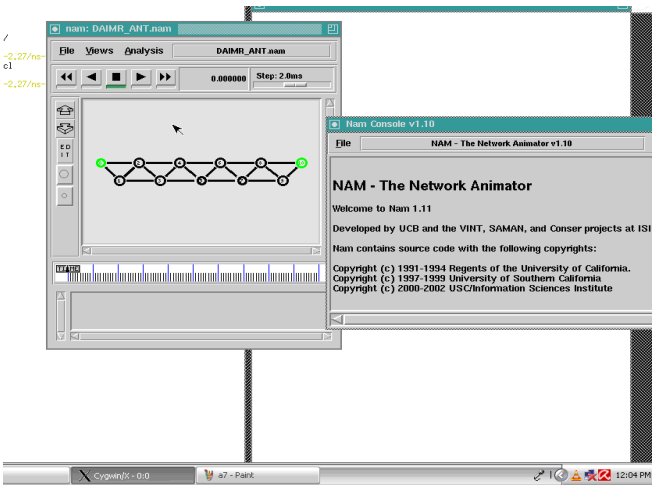

Fig. A1 Wireless Network creation with topology

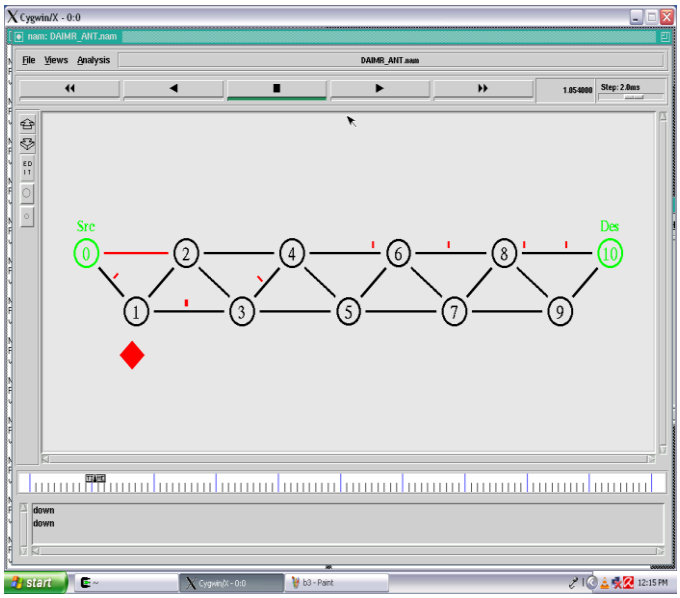

Fig. A2 Packet Transition due to load balancing in Ant-net approaches in router fabric

The backward ant makes the same path as that of its corresponding forward ant, but in the opposite direction. At each node k along the path it pops its stack Ss_d (k) to know the next hop node

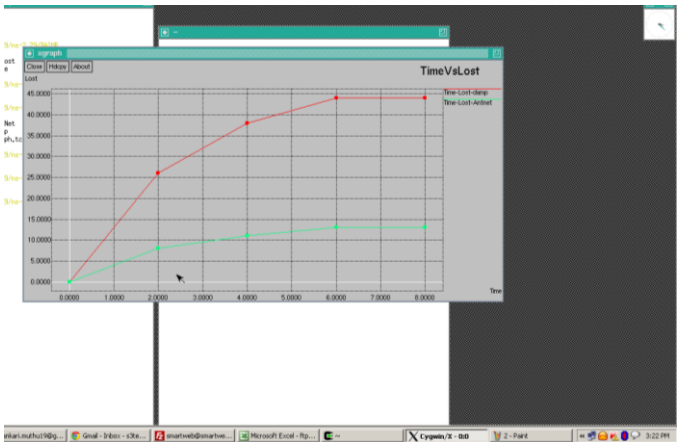

Fig. A3 Performance result time vs lost

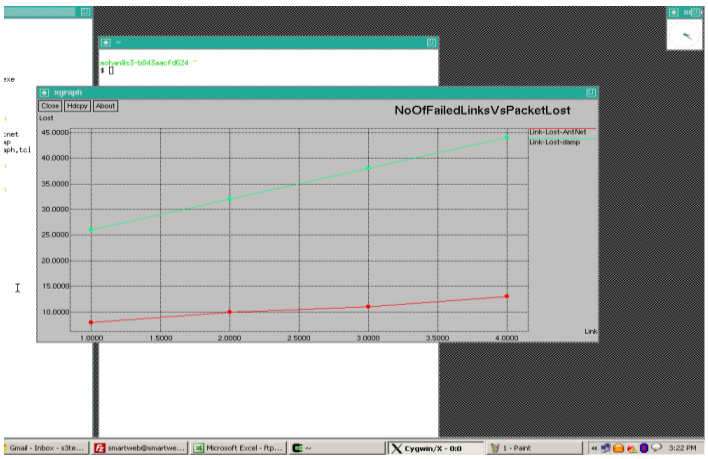

Fig. A4 Performance graph Number of Failed Link Vs

Packet Lost

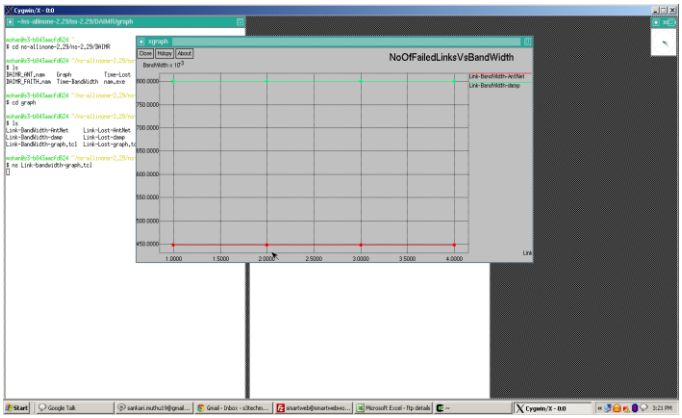

Fig. A5 Number of failed link Vs Bandwidth

V CONCLUSION

This paper proposes an adaptive situation-aware load balance routing Scheme that suit for MWMNs[11]. In order to maintain the optimal path during the transmissions, we also propose a routing metric for judging the situation of routing paths. The simulation results indicated that our proposed Antnet Scheme with routing metric not only achieves the load balance but also improve the total throughput and end-to-end delay.

REFERENCES

[1] L. Ma, and M.K. Denko, "A Routing Metric for Load-Balancing in Wireless Mesh Networks", 21st Int. Conf. Advanced Information Networking and Applications Workshops, vol. 2, pp. 409 - 414, 2007.

[2] I.F. Akyildiz, and X. Wang, "A survey on wireless mesh networks", IEEE Communications Magazine, vol. 43, Issue 9, pp. S23 - S30, 2005.

[3] P.-H. Hsiao, A. Hwang; H.T. Kung and D. Vlah, " Load-balancing routing for wireless access networks" IEEE INFOCOM, vol. 2, pp. 986- 995, 2001.

[4] Y.Bejerano, S.J.Han, and A.Kumar, "Efficient load-balancing routing for wireless mesh networks", Int. Journal Computer and Telecommunications Networking, vol. 51, Issue 10, 2007.

[5] R. Pal, "A Lexicographically Optimal Load Balanced Routing Scheme for Wireless Mesh Networks", IEEE Int. Conf. Communications(ICC), pp. 2393-2397, 2008.

[6] D. De Couto, D. Aguayo, J. Bicket, and R. Morris, "A high-throughput path metric for multi-hop wireless routing", 9th Annual Int. Conf. Mobile Computing and Networking (MobiCom), 2003.

[7] R. Draves, J. Padhye, and B. Zill, "Routing in multiradio, multi-hopwireless mesh networks", 10th annual Int. Conf. Mobile computing and networking, pp. 114-128, 2004.

[8] S. Waharte, B. Ishibashi, R. Boutaba, and D. Meddour, "Interference- Aware Routing Metric for Improved Load Balancing in Wireless Mesh Networks", IEEE Int. Conf. Communications(ICC), pp. 2979-2983, 2008.

[9] IEEE 802.11 WG TGs, "Draft Amendment to Standard IEEE 802.11TM: ESS Mesh Networking", P802.11sTM /D1.08, Jan. 2008.

[10] R. Langar, N. Bouabdallah, R. Boutaba and G. Pujolle, "Interferer Link- Aware Routing in Wireless Mesh Networks", IEEE Int. Conf. C Communications (ICC), pp. 1-6, 2010.
[11]Guan-Lun Liao, Chi-Yuan Chen, Shih-Wen Hsu, Tin-Yu Wu ,HanChiehChao,"Adaptive situationaware load balance scheme for Mobile Wireless Mesh Networks",IEEE Int. Conf. Communications(ICC), pp. 2393-2397, 2011

Author Profile

K.Karthick received his B.E degree in Electronics and Communication Engineering from Scad college of Engineering and Technology, Cheranmahadevi,Tirunelvali Dist., Tamil Nadu, India. Currently he pursuing his M.E (Communition Systems) from Sethu Institute of Technology, Kariyapatti, Virudhunagar Dist.,Tamil Nadu, India.

His research interests include image processing, and Wireless Sensor Networks.

M.Thalaimalaichamy received his B.E. Electronics and communication Engineering from Kamaraj college of engineering and Technology, Virudhunagar and M.E (Communication Systems) from Sudharsan engineering college Pudukottai in the year 2006 and 2011 respectively. He is presently working as Assistant Professor, Department of ME (communication Systems) at Sethu Institute of Technology India. He has published many research papers in the national and international journals, conferences.

His research interests are Wireless Sensor Networks, Digital Communication and Embedded Systems.