A MULTI WAY ACKNOWLEDGMENT PROTOCOL TO DETECT MISBEHAVING NODES IN MANETS

S. Usha, Dr. S. Radha

ABSTRACT

Quality Function Deployment (QFD) is a product development process that encompasses a sheer amount of data gathered from customers through several market research techniques like personal interview, focus groups, surveys, video conferencing etc. This massive, unsorted and unstructured data is required to be transformed into a limited number of structured information to represent the actual 'Customer Needs'. However the process is tedious and time consuming and cannot be dealt with manually. In order to address these issues, this paper proposes a futuristic software framework based on an Affinity Process. The paper begins with the topic introduction and outlines the QFD process. The paper then describes the Affinity Process, builds the data structure and then makes an attempt to build the proposed framework using tools Visual Basic (VB) and MS-Access. The proposed framework is developed as a part of QFD software and it is anticipated that when completely developed, it would act as a vital component of QFD software.

Keywords: QFD, Affinity Process, Visual Basic, MS-Access, Software, Customer Needs.

1. INTRODUCTION

Routing the packets has been one of the most intricate things that have both been the primary instigator and defying aspect in the development of Mobile Ad hoc Networks (MANET). The problems associated with the routing techniques that could possibly be employed for MANETS are those which aim to keep up with the ultra dynamic topology of the nodes and the requirements of each node in order to behave as routers themselves. The difficulties arise when one or more of these nodes in a network link tend to misbehave. The tendency of a node to deviate from the accepted norm is classified into two categories, selfishness and malignance. The former being a node which deems it not necessary to forward those packets which are not destined to itself, chiefly because of the greed on the part of the node to conserve battery power(17). The second kind of misbehaving node is the one with the explicit aim to bluff the neighbors into thinking that it is behaving properly by even wasting some resources while actually misleading them. Several techniques (9) & (11) have been proposed which aims to mitigate the problems of detecting and isolating the misbehaving nodes. Some have used the optimistic strategy of encouraging the nodes to forward packets. These techniques have been drafted with some or most of the parameters viz the

routing overhead, packet delivery ratio and throughput in mind. The credit based schemes have been the ones which are optimistic by promoting the forwarding of packets by each node. This is done by awarding each node credits in the form of nuggets for having successfully forwarded a packet. Some of the widely acclaimed techniques include Packet Purse Model and the Packet Trade Model (1). An improvement of this protocol was made with the addition of a nugget counter (2). SPRITE (3) was another model which proposed to use the credit based scheme. The reputation based schemes involved having separate modules which would take care of the job of marking and detecting misbehaving nodes in the neighborhood and also informing other nodes about the suspected nodes based on pre specified parameters and thresholds. The techniques include watchdog, path rater (4) and the CONFIDANT (5) & (8) protocol. Finally, the end to end schemes were expected to be the panacea for all routing problems. They depended on explicit acknowledgements between nodes. A couple of those techniques were the 2ACK (6) scheme and SACK scheme. Some protocols aimed to detect even the malicious nodes. Secure Trace Route (7) is one such protocol.

While the protocols have the collective aim to mitigate the problems such as

1. Misbehaving node detection
2. Congestion avoidance
3. Avoid additional software modules
4. Detect colluding nodes (up to N-1 nodes colluding in a N nodes link)
5. Receiver collisions and
6. Ambiguous collisions none has been able to solve all the above said problems satisfactorily.

We begin by individually introducing each of these techniques by discussing their pros and cons and finally propose a theory which attempts to present an explication for all the problems.

2. RELATED WORK

PPM (1) is one of the credit based methods in which the forwarding of packets by intermediate nodes is encouraged by providing some resources other than physical, the presence of which is made indispensable for sending packets in the future. In the packet purse model the originator is charged for the message it wishes to send. The charges are handled in currencies called nuggets. In the PTM model each node has to buy the packets for a certain no of nuggets and can sell it to the next node for some amount of nuggets. This ensures that the packet purse which contains the nuggets need not be carried all along the path. Also, because of this scheme the source does not need to know the total amount of

nuggets required in advance. This also means that it is not necessary for the source to bear the entire cost of forwarding but the destination has to. Since the destination pays for the packet forwarding service, there is a scope for multicast packet transfer mode with this model. Then nugget model is also based on the credit based systems make use of simple counters for nuggets. The source node in order to send a packet calculates the no of hops to reach the destination and if the total no of nuggets it has is more than the no of hops required to reach the destination then the packet is sent else the packet is dropped. Also each node increases the total no of nuggets it has by one for forwarding each packet once. In the SPRITE model (3) there is a centrally located credit clearance system. A group of nodes which has access to the network interface via a wireless overlay are considered. Each node should possess a certificate provided by an authorized central authority. The SPRITE works above the DSR protocol. In general a node will gain more credit if it forwards a packet for some other node. The same node would lose a part of credit if its own message is to be forwarded. The WATCHDOG methodology is used to detect misbehaving nodes. The Watchdog method uses the passive method of overhearing the links of the next node to see whether they have forwarded the packet. This is possible as each node can listen to all the links of its next node. This way it can readily eavesdrop on the packets being forwarded by the next node. In cases where there is no link encryption, the nodes can even check for the integrity of the messages. PATHRATER model (4) proposes the use of link data as well as misbehaving node data to select a path. Each node maintains a metric for every node that it knows. And each node also maintains a metric for each path it knows. The path metric is calculated as an average of the metrics of the individual nodes in the path. So if a node finds that there are various paths that could be used to reach the destination, it chooses the one with the highest metric. In the CONFIDANT protocol the node has several modules whose aim is to monitor neighborhood links and attest ratings and provide alarms. The four modules used are the monitor, the trust manager, the reputation manager and the path manager. Each node has all the four modules. The monitor watches the neighborhood for any misbehavior. It does so by either overhearing the link to the next node or by observing the protocol behavior. By keeping a cache of recently forwarded messages, the monitor can also detect the changes in the content. If the monitor has sufficient reason to suspect any aberration, it apprises the Trust Manager about the event. The 2-ACK scheme (6) is a network layer technique to detect misbehaving links. It is implemented over DSR. It is used as an add-on over the DSR. It defines a packet (2-ACK packet), which has a fixed route of two hops in the direction opposite to the original packet flow.

3. AODV

The mobile ad hoc networks (MANETs) as well as other wireless ad hoc networks use a routing protocol called as ad hoc on-demand distance vector (AODV) routing. This routing protocol was developed in Nokia research center by C. Perkins and S. Das. AODV can be of both unicast as well as multicast routing. AODV starts routing only when there is a demand unlike other protocols which are proactive (finding routing paths independent of the paths). AODV is distance vector routing protocol. The disadvantage of other protocols that is the counting to infinity is not seen in AODV. It solves the problem by using the sequence numbers on the route updates. The main advantage of the AODV is that there is no extra traffic for communication. The Distance vector routing is simple and there is no need of much memory and calculation. But the AODV needs more time to establish a connection. AODV establishes the connection only when it is required by a source node for transmitting data packets and so it is said as on demand connection. It uses the concept of the destination sequence numbers to identify the most recent path. Usage of source routing is the major difference between the Dynamic source routing and AODV. The next hop information is stored by the source node and the intermediate node in the AODV. As AODV is an on demand routing protocol, the RouteRequest is flooded in the source node when a path is unavailable. . It may obtain multiple routes to different destinations from a single RouteRequest. Usage of the DestSeqNum is the major difference between the AODV and the other on –demand routing protocols. A node updates its path information only if the DestSeqNum of the current packet received is greater than the last DestSeqNum stored at the node. A RouteRequest carries the source identifier (SrcID), the destination identifier (DestID), the source sequence number (SrcSeqNum), the destination sequence number (DesSeqNum), the broadcast identifier (BcastID), and the time to link (TTL) field. The DestSeqNum shows the freshness of the route. When there is a RouteRequest in the intermediate node, it is either forwarded or a RouteReply is sent but only if there is a valid route. The validity is determined by the comparing the sequence number at the intermediate node with that of the destination sequence number. The duplicate copies are discarded when a RouteRequest is received multiple times when indicated by the BcastID-SrcID pair. Only the intermediate nodes that have valid routes can RouteReply to the source. When there is a RouteRequest sent the intermediate node enters the previous node address as well as the BcastID. When there is no RouteReply, a timer is used to delete the entry. So only the active paths through the intermediate nodes are stored. When a node receives a RouteReply packet, information about the previous node from which the packet was received is also stored in order to forward the data packet to this next node

as the next hop toward the destination. The main benefit is that the routes are created only on demand and also the destination sequence numbers are used in finding the latest route to the destination. The connection setup delay is lower. The main disadvantage is that there may be an inconsistent route in the intermediate nodes when the source sequence numbers are very old as the intermediate nodes do not have the latest destination sequence number which produced the stale entries. Another disadvantage is that there is a heavy control when there are multiple RouteReply packets to a single RouteRequest. There is also a periodic beaconing that leads to unnecessary bandwidth usage.

4. PROPOSED SCHEME

A. N-Ack scheme

The Nack scheme extends the 2 Ack scheme in trying to isolate misbehaving nodes in a MANETs. The Nack scheme requires an end to endAck packet to be sent between the source and the destination. The destination on receipt of the data packets sent by the source, responds with a Nack packet. Each node maintains a list of data packets sent and another list of data packets forwarded. As soon as a node initiates a data packet as a source, it adds the id of the packet to the list of data packet sent. As the node receives the Nack packet for the data packet it removes the corresponding data packet id from the data packet sent list. The data packet and the Nack packet keep track of the route they travel. The Nack would try to reach the source from the destination with the help of the path, which is found in the actual message packet, delivered to the destination. If a node is found to be misbehaving in the pre calculated path, the intermediate nodes are free to divert the Nack packet through alternative paths. But the new path will be stored in the Nack packet along with the older path, which is extracted from the original message. On receipt of the Nack packet, the source node compares the two paths that are in the Nack packet. If there is no variation in the paths, then the source node concludes that there are no potential misbehaving nodes in the path. In case the two paths vary, the node in the source to destination path, from where the path varies in the destination to source path is isolated. This node is marked as a potential misbehaving node by the source node. For each potential misbehaving node, a threshold is maintained. If the number of times a node is adjudged as a potential misbehaving node exceeds the threshold, then the node is flagged as misbehaving and information is sent to all the neighboring nodes advising them about the misbehaving node. Further each node must send back a normal Ack to its immediate source node after receipt of any kind of packet. This would help the intermediate node to judge about its immediate neighboring node and advice the other nodes about the credibility of the neighboring nodes. The process is similar to the protocol

followed by a source node to keep track of data packets initiated. Here the intermediate nodes keep track of the forwarded data packets and Nack packets in the forwarded message packets list. The judgment of a neighboring node as potentially misbehaving node is done when an Ack is not received within a pre set time out. As before, the number of times a neighboring is termed as potentially misbehaving node determines whether or not it should be termed as misbehaving nodes in the path. In case the two paths vary, the node in the source to destination path, from where the path varies in the destination to source path is isolated. This node is marked as a potential misbehaving node by the source node. For each potential misbehaving node, a threshold is maintained. If the number of times a node is adjudged as a potential misbehaving node exceeds the threshold, then the node is flagged as misbehaving and information is sent to all the neighboring nodes advising them about the misbehaving node. Further each node must send back a normal Ack to its immediate source node after receipt of any kind of packet. This would help the intermediate node to judge about its immediate neighboring node and advice the other nodes about the credibility of the neighboring nodes. The process is similar to the protocol followed by a source node to keep track of data packets initiated. Here the intermediate nodes keep track of the forwarded data packets and Nack packets in the forwarded message packets list. The judgment of neighboring node as potentially misbehaving node is done when an Ack is not received within a pre set time out. As before, the number of times a neighboring is termed as potentially misbehaving node determines whether or not it should be termed as a misbehaving node. To consider the case in which the Nack packets are lost, the source node will wait for a certain time out period and then re send the original data packets assuming the data packets were lost. If the N ack packet is lost either due to misbehaving nodes or some other reason, the destination would receive the same packet again. This should prompt them about the fact that the Nack it sent has not reached the source. Considering it as the work of misbehaving nodes the destination now should go for an alternating path. If the problem persists in multiple paths the common node in the path could be isolated as the misbehaving node. On the other hand if the data packets are lost in the first case, the destination would receive the data packets for the first time

B. Algorithm

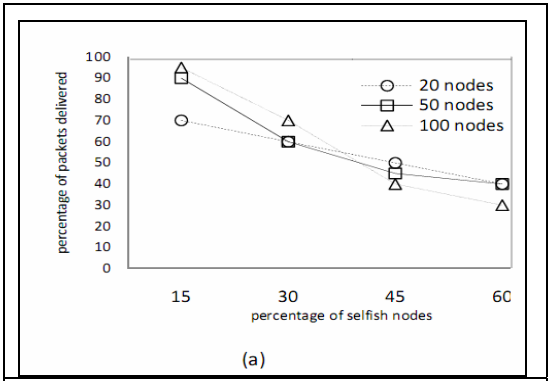
- N1 the source has to send a packet to N5 the destination via N2->N3->N4.
- N1 adds the id of the packet to a wait list.
- N1 forwards the packet to N2 and waits for ack.

- If ack fails to arrive within the stipulated time N1 retries for K times after which it announces N2 to be misbehaving.
- Then node N1 waits for the arrival of the N ack packet from the destination.
- It sets up a timer.
- Each intermediate node maintains a list of IDs for a data packet sent on a path.
- Each packet ID will stay for a time T.
- If Ack arrives within T, the ID is removed.
- Else the ID will be removed after the timeout.
- N5 has to send back the N ack packet to the source.
- Each intermediate node has to forward the N ack packet to the source in the same path in which the initial transmission took place.
- Each intermediate node also has to send to its immediate source node an ack packet.
- Each node excepting the source node maintains the list of sent N ack packets and sets up a timer.
- Each node maintains a black list of potential misbehaving nodes.
- If the ack is not received by a particular node then the node to which it has forwarded the packet and has failed to receive the ack is added to the list.
- After K failed attempts to send the packet without receiving the ack it is marked a as misbehaving node.
- And if the source node does not receive the N ack packet in a pre specified time it queries the intermediate nodes for a route update. By this protocol any arbitrary node is responsible for its immediate node.

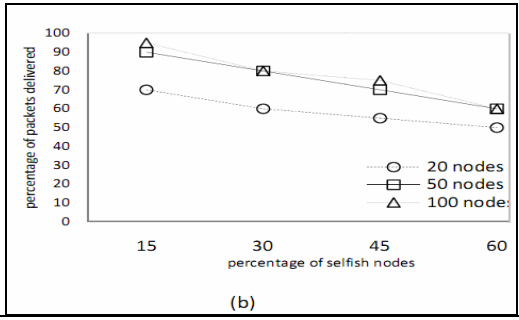
5. SIMULATION

To analyze the performance of our Nack algorithm in extenuating the problems due to misbehaving nodes, we made some modifications to the AODV implementation of the NS-2. The total no of nodes are varied from 20 to 100 nodes comprising a rectangular surface area of 1000X1000 m2. Further for the sake of simplicity we decided to use a constant bit rate source. To be better able to study the enormity of delays produced by the misbehaving nodes, especially in an exceedingly dense and mobile network, we consider the probability that a

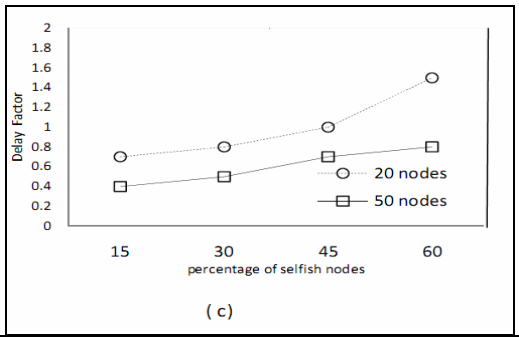
suspected misbehaving node will misbehave to be absolute at all times. But in reality the misbehaving nodes will not misbehave at all times and might behave normally (or along the expected lines of a normal node) when their resources are not under threat of depletion. This is one of the prime reasons which make the detection of the misbehaving nodes tougher. To put things into perspective we take into account two important metrics which form the basis of our simulation results. One of them, being the amount of packets that are successfully transmitted to the total amount of packets that were generated, while the other being the delay or the time taken for the packet to reach the destination from the source. Again in real time mobile network scenario the delays and most importantly the packet delivery ratio are not deterministic unless we take into consideration factors other than the misbehaving nodes. These might include the inherent network delays or problems due to collisions which might entirely be unrelated to the misbehaving nodes. Thus a protocol which aims to ameliorate the network throughput by eliminating misbehaving nodes alone will leave behind residual problems which are not caused due to the misbehaving nodes. Taking these factors into consideration Fig. 1



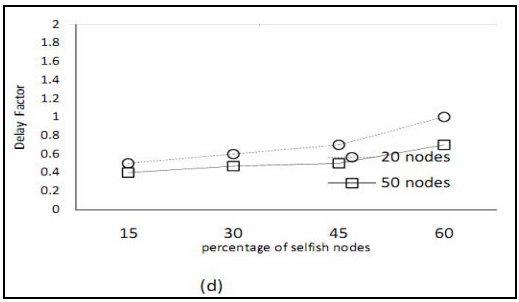
(a)



(b)



(c)



(d)

Fig. 1, 2, 3, 4

depicts the decrease in the packet delivery ratio with increase in the incidence of misbehaving nodes in the network under AODV without our N ack implementation. On the other hand Fig. 2 portrays a marked reduction in the rate of decrease of packet delivery ratio even as the misbehaving nodes increase due to the implementation of Nack over the AODV. This gradual increase in the dropped packets is attributed to the mechanism which aims to pluck out the nodes which are aberrant. The Fig. 3 plots the delay due to misbehaving nodes under normal AODV. Fig. 4 once again shows a much smoother increase of delay as the number of misbehaving nodes increase, when Nack is implemented over the AODV.

6. CONCLUSION

The schemes that have been developed so far while fulfilling some of the requirements of an ideal MANET environment have fallen short in locating the misbehaving nodes. The situation where the nodes collude with each other is still an area of concern where no satisfactory solution has been proposed. Our Nack scheme though, proposes to overcome most of the problems.

REFERENCES:

1. L. Buttyan and J.-P. Hubaux,(2000) "Enforcing Service Availability in Mobile Ad-Hoc WANs," Proc. MobiHoc.
2. L. Buttyan and J.-P. Hubaux, "Stimulating Cooperation in Self-Organizing Mobile Ad Hoc Networks," ACM .
3. S. Zhong, J. Chen, and Y.R. Yang, (2003)"Sprite: A Simple, Cheat-Proof, Credit-Based System for Mobile Ad-Hoc Networks," Proc.INFOCOM.
4. S. Marti, T. Giuli, K. Lai, and M. Baker,(2000) "Mitigating Routing Misbehavior in Mobile Ad Hoc Networks," Proc. MobiCom.
5. S. Buchegger and J.-Y. Le Boudec,(2002) "Performance Analysis of the CONFIDANT Protocol: Cooperation of Nodes, Fairness in Dynamic Ad-Hoc Networks," Proc. MobiHoc.
6. K. Balakrishnan, J. Deng, and P.K. Varshney, (2005) "TWOACK: Preventing Selfishness in Mobile Ad Hoc Networks," Proc. IEEE Wireless Comm. and Networking Conf. (WCNC '05).
7. S. Zhong, J. Chen, and Y.R. Yang,(2005) "Sprite: A Simple, Cheat-Proof, Proc. IEEE Wireless Comm. and Networking Conf.
8. Y. Hu, D.B. Johnson, and A. Perrig, (2003)"SEAD: Secure Efficient Distance Vector Routing for Mobile Wireless Ad Hoc Networks," Ad Hoc Networks, vol. 1, no. 1, pp. 175-192.
9. D. Johnson, D. Maltz, Y.C. Hu, and J. Jetcheva, (2002) "The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR)," Internet draft.
10. B. Awerbuch, D. Holmer, C.-N. Rotaru, and H. Rubens,(2002) "An On-Demand Secure Routing Protocol Resilient to Byzantine Failures," Proc. ACM Workshop Wireless Security (WiSe).

11. L. Zhou and Z.J. Haas, "Securing Ad Hoc Networks, (2003)" IEEE Network Magazine, vol. 13, no. 6, Nov./Dec. 1999. Proc. INFOCOM.
12. H. Miranda and L. Rodrigues, (2002) "Preventing Selfishness in OpenMobile Ad Hoc Networks," Proc. Seventh CaberNet RadicalsWorkshop.
13. (13)L. Buttyan and J.-P. Hubaux, (2006) "Security and Cooperation in Wireless Networks," http://secowinet.epfl.ch,
14. S. Buchegger and J.-Y. L. Boudec, (2002)"Performance analysis of the CONFIDANT protocol: Cooperation of nodes -fairness in dynamic ad-hoc networks," in Proceedings ofIEEE/ACM Workshop on Mobile Ad Hoc Networking and Computing (MobiHOC). IEEE.
15. The network simulator http://www.isi.edu/nsnam/ns