Object Deployment Using Data Trans-Reception in Military Applications

Subha J[1], Karthikeyan R[2]
[1](CSE, Bharath University, India)
[2](CSE, Bharath University , India)

ABSTRACT : *An mobile ad-hoc network (MANETs) is a set of limited range wireless nodes that function in a cooperative manner so as to increase the overall range of the network. In general, routing protocols for MANETs are designed based on the assumption that all participating nodes are fully cooperative. However, due to the open structure and scarcely available battery-based energy, node misbehaviors may exist. One such routing misbehavior is that some selfish nodes will participate in the route discovery and maintenance processes but refuse to forward data packets. In this paper, we propose the 2ACK scheme that serves as an add-on technique for routing schemes to detect routing misbehavior and to mitigate their adverse effect. The main idea of the 2ACK scheme is to send two-hop acknowledgment packets in the opposite direction of the routing path. In order to reduce additional routing overhead, only a fraction of the received data packets are acknowledged in the 2ACK scheme. Analytical and simulation results are presented to evaluate the performance of the proposed scheme.*

Keywords *- Battery-based energy, MANETs, Two-hop Routing Misbehavior, 2ACK .*

I. INTRODUCTION

Wireless networking is an emerging technology that allows users to access information and services electronically, regardless of their geographic position. Wireless networks can be classified in two types: -[1]. Infrastructure network consists of a network with fixed and wired gateways. A mobile host communicates with a bridge in the network (called base station) within its communication radius. The mobile unit can move geographically while it is communicating. When it goes out of range of one base station, it connects with new base station and starts communicating through it. This is called handoff. In this approach the base stations are fixed.

In ad hoc networks all nodes are mobile and can be connected dynamically in an arbitrary manner. All nodes of these networks behave as routers and take part in discovery and maintenance of routes to other nodes in the network. Ad hoc networks are very useful in emergency search-and-rescue operations, meetings or conventions in which persons wish to quickly share information, and data acquisition operations in inhospitable terrain.[1].

In traditional networks, most trust evidence is generated via potentially lengthy assurance processes, distributed o.-line, and assumed to be valid on long terms and certain at the time when trust relations derived from it are exercised. Authentication and access-control trust relations established as a consequence of supporting trust evidence are often cached as certificates and as trust links (e.g., hierarchical or peer links) among the principals included in these relations or among their “home domains.” Both certificates and trust relations are later used in authorizing client access to servers.

In contrast, few of these characteristics of trust relations and trust evidence are prevalent in mobile ad-hoc networks (MANETs). Lack of a fixed networking infrastructure, high mobility of the nodes, limited-range and unreliability of wireless links are some of the characteristics of MANET environments that constrain the design of a trust establishment scheme. In particular, trust relations may have to be established using only on-line-available evidence, may be short-term and largely peer-to-peer, where the peers may not necessarily have a relevant “home domain” that can be placed into a recognizable trust hierarchy, and may be uncertain.

In this work we argue that for trust establishment in MANETs a substantial body of trust evidence needs to be (1) generated, stored, and protected across network nodes, (2) routed dynamically where most needed, and (3) evaluated “on the fly” to substantiate dynamically formed trust relations. In particular, the management of trust evidence should allow alternate paths of trust relations to be formed and discovered using limited backtracking though the ad-hoc network, and should balance between the reinforcement of evidence that leads to ”high certainty” trust paths and the ability to discover alternate paths.[2]

II. THE 2ACK SCHEME

The 2ACK scheme is a network-layer technique to detect misbehaving links and to mitigate their effects. It can be implemented as an add-on to existing routing protocols for MANETs, such as DSR. The 2ACK scheme detects misbehavior through the use of a new type of acknowledgment packet , termed 2ACK.

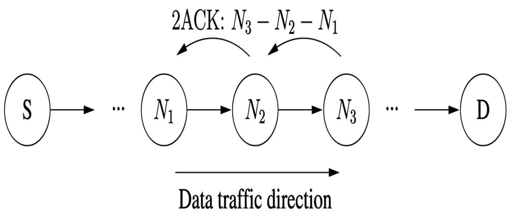


Fig 1…..2ACK SCHEME

Fig 2.1. illustrates the operation of the 2ACK scheme. Suppose that N1, N2, and N3 are three consecutive nodes (triplet) along a route. The route from a source node, S, to a destination node, D, is generated in the Route Discovery phase of the DSR protocol. When N1 sends a data packet to N2 and N2 forwards it to N3, it is unclear to N1 whether N3 receives the data packet successfully or not. Such an ambiguity exists even when there are no misbehaving nodes. The problem becomes much more severe in open MANETs with potential misbehaving nodes.

The 2ACK scheme requires an explicit acknowledgment to be sent by N3 to notify N1 of its successful reception of a data packet: When node N3 receives the data packet successfully, it sends out a 2ACK packet over two hops to N1 (i.e., the opposite direction of the routing path as shown), with the ID of the corresponding data packet. The triplet [N1 -> N2 -> N3] is derived from the route of the original data traffic. Such a triplet is used by N1 to monitor the link N2 -> N3. For convenience of presentation, we term N1 in the triplet [N1 -> N2 -> N3] the 2ACK packet receiver or the observing node and N3 the 2ACK packet sender.

Trust Derivation

Events gathered in passive mode (e.g.) frames received, data packets forwarded, control packets forwarded. The information from these events is classified into trust categories

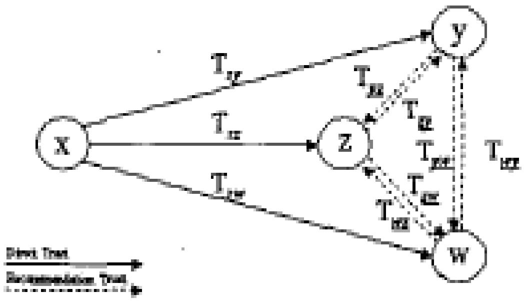


Fig 2 Trust Derivation

Trust Computation

$T_x(y)$: trust of node y by node x

$W_x(i)$: weight of i^{th} trust category to x

$T_x(i)$: situational trust of x in i^{th} trust category

n : number of trust categories

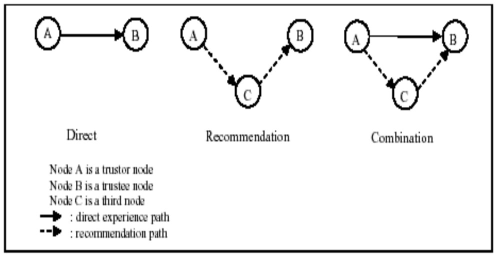


Fig 3 Trust Computation

III. PARAMETERS CONSIDERATION

- **Acknowledgements**

Passive acknowledgement method provides information about next hop (e.g.) it is acting like a black hole if the packet is not retransmitted. For every packet transmitted, the counter is incremented depending if the neighbor node has ***correctly forwarded it or not*** .

- **Packet Precision**

Accuracy of received data and routing packets offers a measure to compute trust level (e.g.) if routing packets received are correct, the originator can be allotted a higher trust value along with the set of nodes provided in that packet.

- **Gratuitous Route Replies**

Route shortening" to avoid unnecessary intermediate nodes by other overhearing nodes.

- **Blacklists**

AODV maintains blacklists for nodes displaying uni-directional behavior, i.e. if a neighbor node has received a packet and either due to a unidirectional link or selfish behavior the sender cannot hear it retransmitting.

- **BEACON/HELLO Packets**

AODV uses HELLO packets to maintain local neighborhood connectivity information. Each active node that has not sent any broadcast in a certain period, broadcasts a HELLO packet (ROUTE REPLY Packet with Hop-Count=0) with time-to-live set to 1. These packets ensure that all neighbors maintain active routes between each other at all times. All recipient nodes create forward routes to the transmitting node. The absence of a HELLO packet from a neighbor for certain duration makes the route to that node invalid.

IV. IMPLEMENTATION AND RESULT ANALYSIS

In Ad-hoc On-Demand Distance Vector (AODV), If a node S needs a route to a destination D and does not have one, it floods a route-request (RREQ) packet through the network. Each recipient R of this RREQ keeps a return pointer. R broadcasts the request to its neighbors if it is not D and does not have a route to D.If R is D, or has a route to D, it responds with a route-reply (RREP) packet using the return pointers for S.

Routing loops are undesirable.

AODV uses sequence numbers to indicate freshness of link information.

Key Invariant: If next(n) = n', then

seqno(n) $\leq$ seqno(n'), and

if seqno(n) = seqno(n'),

then hops(n) > hops(n').

The invariant ensures that there are no loops.

The Ad hoc On-Demand Distance Vector protocol is both an on-demand and a Table driven protocol. The packet size in AODV is uniform unlike DSR. Unlike DSDV, there is no need for system-wide broadcasts due to local changes. AODV supports multicasting and unicasting within an uniform framework. Each route has a lifetime after which the route expires if it is not used. A route is maintained only when it is used and hence old and expired routes are never used. Unlike DSR, AODV maintains only one route between a Source-destination pair.

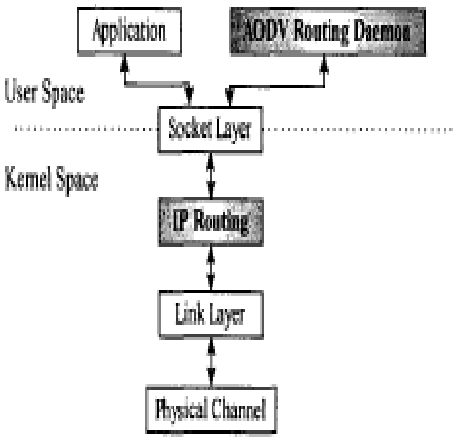


Table 4 AODV Implementation

Using NS2,the simulator itself, now version: ns-2.29. Simulates the program.NAM: Network animator. Visualized trace tool.

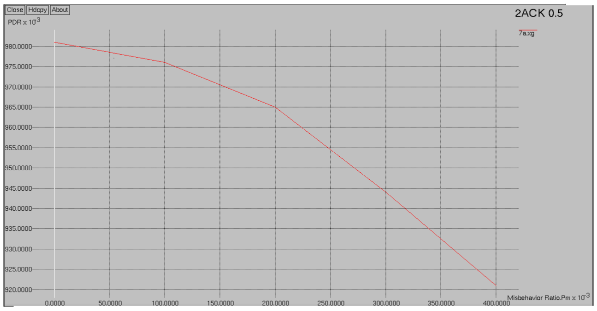
Displays the transmission of the nodes. Pre-processing: Traffic and topology generators. Post-processing: Simple trace analysis, often in Awk, Perl(mostly), or Tcl.

TRACEGRAPH: Graphs can be plotted by loading the trace file that is generated when we run the program. The graph is plotted for various parameters.Both 2D and 3D graphs can be plotted.

The below trace graph X-axis is Misbehavior Ratio and Y-axis is Packet Delivery Ratio with constant of Acknowledgment Ratio and Maximum Mobile speed were plotted.

Packet Delivery Ratio (PDR) Vs Misbehavior Ratio(Pm):

Acknowledgment Ratio(Rack)= 0.5



. Fig 5. graph 1

Packet Delivery Ratio (PDR) Vs Misbehavior Ratio(Pm):

Acknowledgment Ratio(Rack)= 0.2

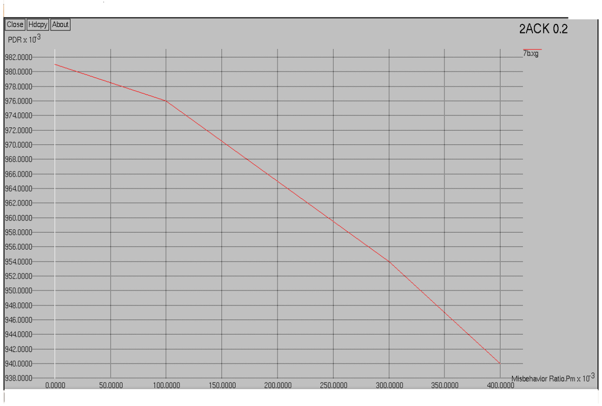


Fig 5. graph 2

Packet Delivery Ratio (PDR) Vs Misbehavior Ratio(Pm):

Acknowledgment Ratio(Rack)= 0.05

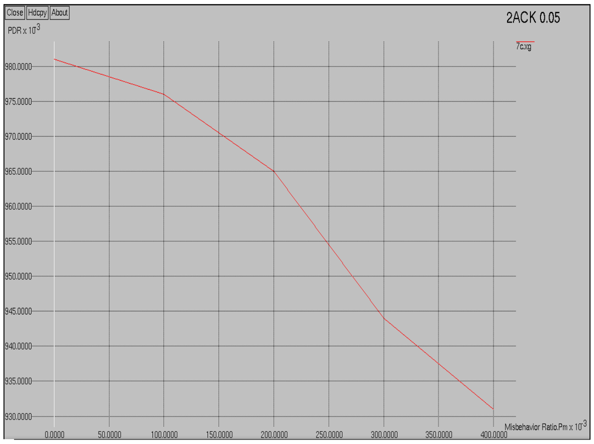


Fig 5. graph 3

V. CONCLUSION

We have investigated the performance degradation caused by such selfish (misbehaving) nodes in MANETs. We have proposed and evaluated a technique , termed 2ACK, to detect and mitigate the effect of such routing misbehavior. Aims to build confidence measures regarding route trustworthiness .Builds confidence regarding route trustworthiness that is computed and modified based on effort expended and passively observed. In an ad-hoc network where doubt and uncertainty are inherent, our trust model helps to create and maintain trust levels based on an effort/return mechanism. Establishes relative levels of trustworthiness in the routes selected using our model. From the simulation we check the route which has least number of packet drops and route disruptions and select it as the most reliable path for transmission and establish relative levels of trustworthiness in the routes selected using our model.

REFERENCES

[1] H. Miranda and L. Rodrigues, “ Preventing **Selfishness in Open Mobile Ad Hoc Networks** ,” Proc. Seventh Caber Net Radicals Workshop, Oct. 2002.

[2] L.M. Feeney and M. Nilsson, “ **Investigating the Energy Consumption of a Wireless Network Interface in an Ad Hoc Networking Environment**,” Proc. IEEE INFOCOM, 2001.

[3] S. Marti, T. Giuli, K. Lai, and M. Baker, “ **Mitigating Routing Misbehavior in Mobile Ad Hoc Networks**,” Proc. MobiCom, Aug.2000

.

[4] Y. Hu, A. Perrig, and D.B. Johnson, “ **Ariadne: A Secure On-Demand Routing Protocol for Ad Hoc Networks**,” Proc. MobiCom, Sept. 2002.

[5] L. Zhou and Z.J. Haas, “ **Securing Ad Hoc Networks**,” IEEE Network Magazine, vol.13, no. 6, Nov./Dec. 1999.