Environment-Aware Trusted Data Delivery in Multipath Wireless Protocols

Mohit Virendra[1], Arunn Krishnamurthy[1], Krishnan Narayanan[1], Shambhu Upadhyaya[1] and Kevin Kwiat [2]

[1] Computer Science & Eng., State University of New York at Buffalo, Buffalo, NY USA
{virendra, ack6, kn38, shambhu}@cse.buffalo.edu
[2] US Air Force Research Laboratory, 525 Brooks Road, Rome, NY
kwiatk@rl.af.mil

Abstract. Current multipath protocols for Multi-Hop Wireless Networks (MWNs) use hop-count as the default route selection criteria. Route selection should also consider network and link conditions. We propose a network-environment-aware trust-based route selection framework for MWNs that makes informed and adaptive route-selection decisions. A node quantifies trust values for its neighboring nodes and for the routes that pass through it. The trust metric adjusts to varying network conditions and quick convergence of the protocol implies it works well in mobility scenarios. Glomosim simulations demonstrate throughput improvement over conventional multipath protocols under congestion, link failure and route unreliability scenarios.

Key words: AOMDV, Multipath, Routing, Security, Trust

1 Introduction

Ad-hoc On-demand Multipath Distance Vector (AOMDV) [1] and AODVM [3] routing are multipath variants of the well-known AODV protocol [2] for Mobile Multi-Hop Wireless Networks (MWNs). Route selection may be affected by congestion, presence of selfish (or malicious) nodes, and other adverse network or physical conditions. Additional route information may enhance probability of packets reaching destination. In existing multipath protocols, if nodes could evaluate confidence on available routes (estimate route reliability due to physical and network conditions), make *trusted* route selection decisions, and dynamically switch traffic across different available routes, data delivery robustness would be enhanced.

This paper presents a framework for MWN nodes to evaluate route conditions and provides metrics to make informed multipath routing decisions. Our goal is to quickly detect any *effects* on data transfer, attributable to malicious nodes or other adverse conditions in a route, and to take corrective actions (e.g., use alternate routes). In our model, a node quantifies trust values for its neighboring nodes and for the routes that pass through it. The trust metrics adjust to varying network conditions; quick protocol convergence implies it works well in mobility scenarios.

Research supported in part by U.S. Air Force Research Laboratory Grant No. 200821J

Recent trust models for node dependability, reliability and security in P2P systems are summarized by Li et al. [5]. Expensive peer-node promiscuous monitoring for behavior assessment, significant in all existing models [9, 6, 7, 8], is minimized in our framework by enhancing node accountability for data forwarding cooperation. Overall route-performance rather than individual node-misbehavior detection is our focus. No extra control overhead for trust computation is introduced; convergence time for our framework is the same as that of AOMDV. Since it assumes presence of cryptographic protocols, information will not be compromised when our protocol is executing in the presence of malicious nodes. We do not distinguish between packets dropped due to malicious or selfish behavior and due to congestion. All these are inhibitive towards efficient data transfer and worthy of loss of trust. Nodes have unique non-forgeable IDs. Links are bidirectional; link costs/capacities maybe directionally different and are estimated through well known techniques (e.g., [10]). Trust is non-transitive, i.e., Txy ≠ Tyx. Downstream is towards destination and upstream is towards source. Source and destination nodes are assumed to be non-malicious. Trust is computed on a continuous scale of 0 to 1.

2 Technique

In AODV, the source node broadcasts a *Route Request* (RREQ) packet which is in turn re-broadcasted by the nodes' neighbors until the sought route is discovered. Upon receiving an RREQ, the destination node or an intermediate node with a 'fresh enough' route to the destination, unicasts a *Route Reply* (RREP) packet back to the source node. AODV also uses *Route Error* (RERR) and *Route Reply Acknowledgement* (RREP-ACK) control packets for route management. AOMDV discovers link-disjoint or node-disjoint multi-paths between pairs of nodes.

In our model a node maintains two trust values, one for routes passing through it and another for its one-hop neighbors – *Route Trust:* measure of reliability of packets reaching the destination if forwarded on a particular route, computed by each node for all routes in routing table; *Node Trust:* measure of confidence on one-hop neighbors that they accurately assess and report downstream route conditions. Node trust is initialized at 1 for destination and 0.5 for all other participating nodes. Initial route trust is formalized by Effective Link Capacity (ELC) and Effective Route Capacity (ERC) values. ELC is an indicator of the traffic that can be scheduled on the link for a particular route/flow. ERC is the *effective* capacity of the route from an intermediate node to the destination. It factors in the ELCs computed at each intermediate node. An ERC value corresponding to a route at a node can thus be analogous to that node's trust on that route downstream. Subsequent updates to node and route trusts are interdependent. Route trust is recursively computed by each node starting at the destination and moving upstream, taking care of route divergences and convergences. At each hop, a node's assessment of its downstream reporting neighbor (i.e., node trust) is factored into the route trust. In turn, the difference between predicted route trust and the eventual route performance governs an upstream node's trust (node trust) on its downstream neighbor. Thus nodes are accountable for providing an accurate assessment of route conditions.

Additions/modifications to AOMDV are made for piggy-backing ELC and ERC values in the RREP packets upstream and maintaining trust details for all routes in each node's routing table. The details are as follows:

- Each node maintains an additional data structure called the Neighbors' Trust Table. It contains neighboring node IDs, and corresponding node trust values.
- RREP packets have an additional route trust field and routing tables have a route trust entry for every destination as well. Upon receiving an RREP, a node caches the route trust sent by the downstream node. The node then reevaluates its own trust on the route downstream, updates the corresponding route trust entry in its routing table, updates the route trust field in the RREP packet and forwards it upstream.
- A *Type* field in the packet header (values 0-3) in AOMDV identifies the control packet type (RREQ, RREP, RERR and RREP-ACK). We introduce two new control packets, the *Query* (QRY: *Type* value 4) and the *Query-Acknowledgement* (QRY-ACK: *Type* value 5) packets. Route tables have a *Query Flag* bit set when a QRY-ACK is expected in response to a QRY. These packets contain encrypted checksum, computed over the entire packet by the packet creator, ensuring tamper-detection.

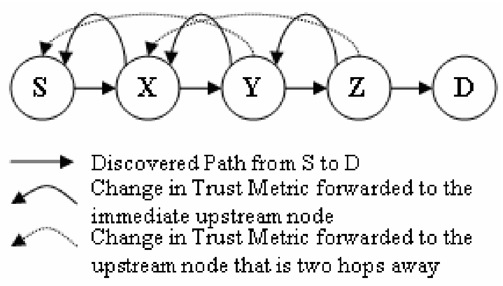
S
X
Y
Z
D
Discovered Path from S to D
Change in Trust Metric forwarded to the immediate upstream node
Change in Trust Metric forwarded to the upstream node that is two hops away

Fig. 1. Reporting of Route Trust Values

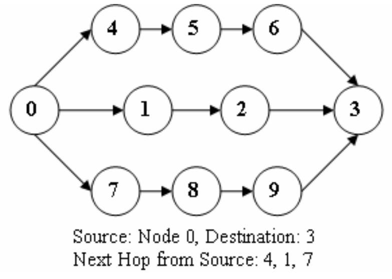

Fig. 2. Simulation Scenario

A node wanting to reassess its route trust whenever appropriate sends a QRY to the destination. The destination sends back a QRY-ACK containing the received packet count since transmission of the last QRY-ACK. The QRY initiator and the intermediate nodes forwarding the QRY-ACK re-compute route trusts using their ERC estimates and the ratio of data packets reaching the destination to data packets forwarded by them.

If multiple QRY or QRY-ACK packets are lost along a route, then the route trust would automatically decrease. We evaluate this in Sec. 3 through simulations. The node trust on the immediate downstream node is computed using the ratio of actual data rate achieved to the data rate promised by the downstream node.

Nodes recursively inform upstream neighbors of any changes in route trusts downstream, plugging in their own assessment of downstream-route-trust at each hop. Accuracy of such updates factors in re-evaluating node trusts on downstream neighbors in turn. For example, precise reporting of decreased route trusts due to congestion does not reduce node trust on the reporting downstream node. A Two-hop Reporting scheme employing AODV's *Localized Repair* feature is used for detecting bogus congestion reporting and silent discarding of QRY-ACK packets by malicious/selfish nodes. Assume that node *Y* (in Fig.1) is malicious. In the two-hop reporting scheme, *Z* sends QRY-ACK to both *Y* and *X* using the localized repair feature. Node *X* should thus receive two copies of the packet which it can compare.

If node *Y* claims route-congestion for a time more than a threshold or when there is ambiguity between reports sent by the two downstream nodes, then all the routes with next hop Y are invalidated and purged from the routing table and RERR messages are sent to the destination. The node trust on *Y* would be made 0.

3 Performance Evaluation and Discussion

Packet Delivery Ratio and Trust Convergence latency were evaluated through Glomosim-2.02 simulations using the topology of Fig. 2 over a field size of 100m X 100m. This is an enhanced version of the network topologies that were used by Das et al. [1] and Yuan et al. [11]. The simulation was run for 150 seconds. Each node has a transmission range of 30 Meters using a Free-Space propagation-path loss model. Constant Bit Rate (CBR) traffic at 512 Bytes per second (bps) with an inter-departure time of 5 ms was injected between the source node 0 and the destination node 3 from the beginning till the end of the simulation.

To simulate general congestion in the network, we introduced an additional 1024 bps CBR traffic at the links [2, 3], [7, 8] and [4, 5] during the interval of 10-30S. Further, localized congestion was created through 2048 bps CBR traffic across the links [2, 3] and [4, 5] during 30-40S; across [2, 3] during 60-70S; across [7, 8] during 75-85S; and across [4, 5] during 85-99S. This emulated a variety of scenarios: simultaneous congestion on two routes, congestion on one route, different times the congestion eases, etc. Finally, nodes were failed during the following time intervals: Node 1:100-125S, Node 7:115-125S, Node 4:130-140S. This was done to study the adaptability of our protocol and achieve a fine grained comparison with AOMDV.

Route selection was weighted round robin. Source node reevaluated trust metrics by sending QRY packets at time intervals dictated by the already computed route trust values. For trust between 0.5-0.8, querying frequency was every 100 packets; for trust > 0.8, it was every 200 packets. The destination node sent back QRY-ACK packets to all the upstream nodes. Results were compared with native AOMDV. Trust values and Packet Delivery Ratio for each path (via nodes 1, 7 and 4) were evaluated. Results for path via Node 1 are reported; similar results were obtained for other paths.

As seen from the Figures 3a and 4, during the initial interval, 10-30S, local congestion simulated in all the available paths considerably affected the overall packet delivery ratio. As a consequence the route trust fluctuated during this time frame. Since there was no alternate path with better route trust, data packets were sent over all the paths and hence the overall protocol suffered due to this congestion. During the interval 60-70S, only the route via node 1 suffered congestion resulting in packet loss. The trust metric re-computation latency (time interval between the onset of congestion and the time at which the source obtains the QRY reply) was approximately 1.5 sec. This number was an averaged output of several test runs. Since the route trust on route via node 1 was greater than 0.8 before 60S, the QRY packets were sent out only after 200 data packets, and hence the trust convergence interval was large as indicated by pointer 1 in Fig. 3b. Once the congestion was realized at the source, the route trust on node 1 was decreased and the traffic was diverted through alternate paths via nodes 4 and 7. Thus, the route trust follows the packet delivery ratio computed using the QRY-ACK packet(s) from the destination. During this 60-70S interval, 5 data packets were sent through the congested route periodically to check if the congestion got cleared. Thus when the localized congestion between nodes 2 and 3 subsided after 70S, the source was able to reassess the trust within the next 0.5 sec. This was because of the reduced QRY request frequency that was set to be every 20 data packets. Thus, the trust convergence interval was less as indicated by pointer 2 in Fig. 3b. In this duration, traffic was redirected through alternate paths and hence the overall packet delivery ratio was not affected as can be seen from the overall packet delivery ratio in Fig. 4. Likewise, when the trust metrics were maintained in between 0.5-0.8, the trust convergence interval was approximately 1 sec. This could be visualized during the 30-31st sec in Fig. 3b. Similar localized congestion, reduction in route trust and diversion of data through highly trusted routes were monitored during 75-85S and 85-99S for the alternate paths and were found to show strict resemblance to the trust convergence latencies observed in Fig. 3b.

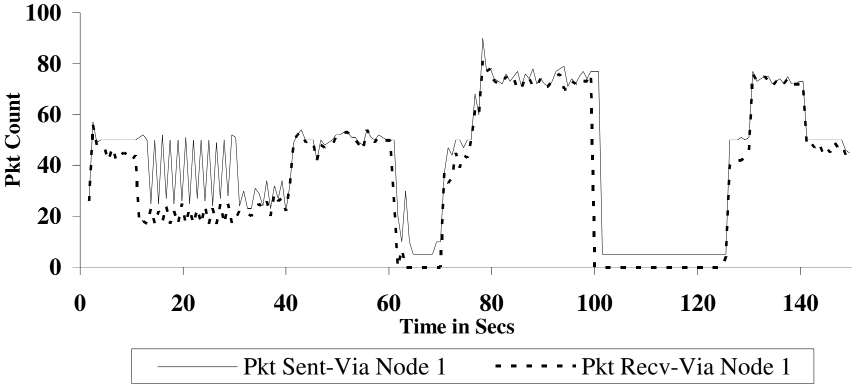

Fig. 3a. Packets Sent/Received Vs Time (For Next Hop Node 1)

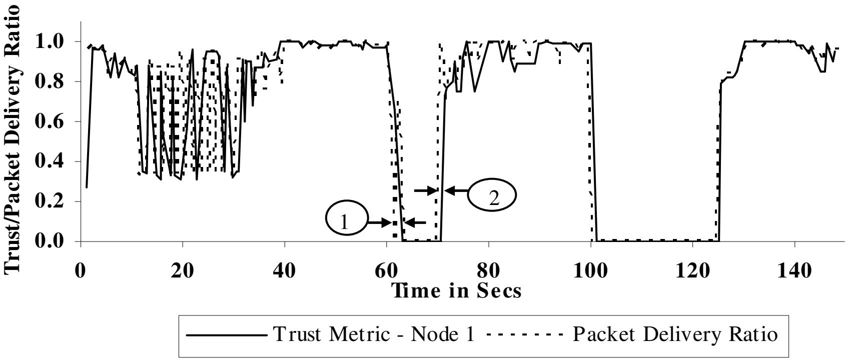

Fig. 3b. Trust & Packet Delivery Ratio Vs Time (For Next Hop Node 1)

The same simulation setup was also used to run two variants of AOMDV: round robin route selection and using a single route. Comparison of our scheme's overall packet delivery ratio with AOMDV variants (Fig. 4) shows that AOMDV (round robin) suffered approximately 50% throughput decline with downstream route congestion; single route AOMDV was even worse. Additionally, our protocol quickly sensed node failures and diverted traffic via alternate paths as against AOMDV that kept attempting to send traffic via routes with failed nodes. The results assure the effectiveness of our proposal when adapted to multipath protocols. It is a self learning scheme which adapts to environment conditions.

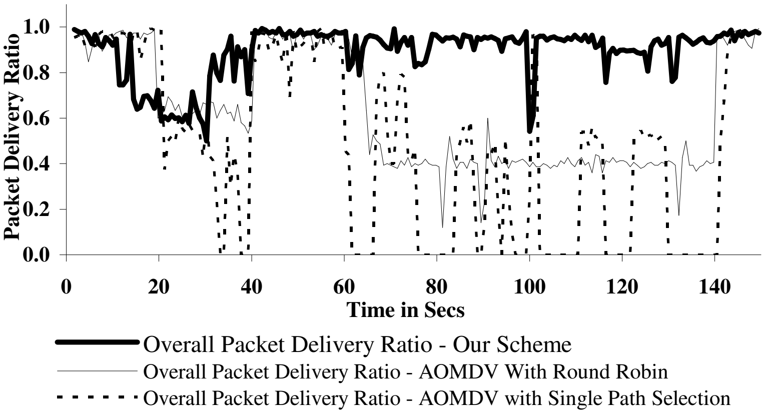

Fig. 4. Throughput Comparison

References

1. Marina, M. K., Das, S. R.: Ad-hoc On-demand Multipath Distance Vector Routing. Proceedings of the IEEE International Conference on Network Protocols (ICNP) (2001) 14-23
2. Perkins, C., Royer, E., Das, S.: Ad hoc On-Demand Distance Vector Routing. RFC-3651 (2003)
3. Ye, Z., Krishnamurthy, S., Tripathi, S.: A Framework for Reliable Routing in Mobile Ad Hoc Networks. Proceedings of the IEEE Conference on Computer Communications (INFOCOM) (2003)
4. Zeng, X., Bagrodia, R., Gerla, M.: GloMoSim: A Library for Parallel Simulation of Large-Scale Wireless Networks. Proceedings of the 12th Workshop on Parallel and Distributed Simulations (1998)
5. Li H., Singhal, M.: Trust Management in Distributed Systems. Computer, Vol. 40. (2007) 45-53
6. Sun, Y., Han, Z., Yu, W., Liu, K.J.: A Trust Evaluation Framework in Distributed Networks: Vulnerability Analysis and Defense against Attacks. Proceedings of the IEEE Conference on Computer Communications INFOCOM (2006)
7. Li, X., Lyu, M.R., Liu, J.: A Trust Model Based Routing Protocol for Secure Ad Hoc Networks. Proceedings of the IEEE Aerospace Conference (IEEEAC) (2004) 1286-1295
8. Zouridaki, C., Mark, B., Hejmo, M., Thomas, R.: A Quantitative Trust Establishment Framework or Reliable Data Packet Delivery in MANETs. Proceedings of Workshop on Security of Ad-Hoc and Sensor Networks (SASN), Alexandria, VA (2005)
9. Marti, S., Giuli, T., Lai, K., Baker, M.: Mitigating Routing Misbehavior in Mobile Ad Hoc Network. Proceedings of 6th Annual Conference on Mobile Computing and Networking (2000) 255-265
10. Li, J., Blake, C., De-Couto, D., Lee, H., Morris, R.: Capacity of Ad Hoc Wireless Networks. Proceedings of International Conference on Mobile Computing and Networking (MOBICOM) (2001)
11. Yuan, Y., Chen, H., Jia, M.: An Optimized Ad-hoc On-demand Multipath Distance Vector (AOMDV) Routing Protocol. Proceeding of the Asia-Pacific Conference on Communications, Australia (2005)