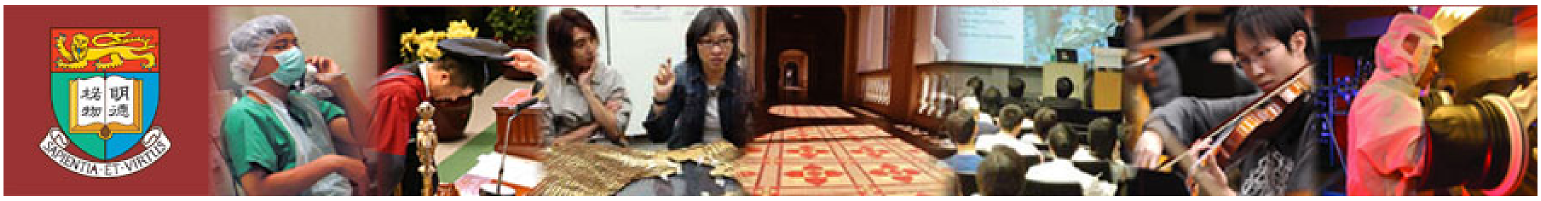

A New Multipath Routing Approach to Enhancing TCP Security in Ad Hoc Wireless Networks

Zhi Li and Yu-Kwong Kwok
Department of Electrical and Electronic Engineering
The University of Hong Kong, Pokfulam Road, Hong Kong
Corresponding Author: Yu-Kwong Kwok (Email: *ykwok@hku.hk*)

Abstract— **In a typical mobile *ad hoc* network, mobile computing devices wander autonomously and communicate via temporary links in a self-organized computing system without any central administrator or infrastructure support. To support truly ad hoc impromptu communication among such uncoordinated devices, a data multipath routing algorithm can be used because there is no need to rely on a centralized encryption facility (e.g., a PKI server) or complicated distributed keying protocols. In this paper, we propose a data multipath routing algorithm called *Multipath TCP Security* (MTS) to enhance the data security. In MTS, the source node adaptively chooses the available routes rather than exhaustively testing the "stored routes" one by one. Simulation results show that our algorithm provides a reasonably good level of security and performance.**

Index Terms— **TCP, wireless security, ad hoc networks, multipath routing, interception rates.**

I. Introduction

A mobile ad hoc network is a dynamically self-organizing network without any central administrator or infrastructure support. If two nodes are not within the transmission range of each other, other nodes are needed to serve as intermediate routers for the communication between the two nodes. Moreover, mobile devices wander autonomously and communicate via temporary links in such an ad hoc wireless network. Thus, frequent change of network topology is a tough challenge for many important issues, such as routing protocol robustness, performance degradation resiliency, and data security.

In ad hoc wireless networks, communicating data is vulnerable to lots of potential attacks due to their characteristics of having dynamic topology, limited bandwidth and energy constraints. Indeed, data security is an important issue in such environment. In general, the attacks are classified as *passive* and *active*. In passive attacks, the attackers attempt to discover valuable information within their transmission range. On the other hand, active attacks attempt to disrupt the operation of communication. In this paper, we focus on passive attacks, which are a much more serious problem in ad hoc wireless networks due to their stealthy nature. Specifically, since all nodes are sharing the same wireless medium, the nodes are exposed to a potentially insecure environment and can be easily attacked in a more complicated manner when compared to central wireless networks, e.g., cellular wireless networks. Consequently, it is usually infeasible to use techniques such as IPSec, TLS, etc. in ad hoc wireless networks.

Recently, multipath routing algorithms have been introduced to enhance data confidentiality in ad hoc wireless networks [1], [5], [8], [14]. The essence of these algorithms is that data is transmitted via multiple paths to prevent some fixed unauthorized nodes from intercepting useful information. Intuitively, multipath routing algorithms are simple and efficient because no encryption is needed and data is "split" among different routes to minimize or even disable potential captures by unauthorized users. In [1], existing multiple paths are used to increase the robustness of confidentiality. All data is transmitting along these paths concurrently. In [5], one path is used as the core path to deliver data and the other paths are alternatives (i.e., once the core path is broken down, an alternative path is nominated to deliver data). However, these two protocols ignore an important characteristic of ad hoc wireless networks: frequent topology change, which leads to one or more used paths to break down or the backup paths to be stale.

In this paper, we propose a novel multipath routing algorithm called *Multipath TCP Security* (MTS) to combat the passive attacks in ad hoc wireless networks. Compared with other multipath routing protocols, MTS has two significant characteristics. First, on receiving the route check packets, the source node adaptively changes the current route to be the fastest one. That is, t

This research was supported by a grant from the Research Grants Council of the HKSAR under project number HKU 7162/03E.

of routes used is not fixed. It can be changed dynamically. By contrast, in other protocols, a new route is triggered to be active only when the old route could not work any more. Second, when the first route request (RREQ) packet reaches the destination node, the route reply (RREP) packet is immediately transmitted to the source node without any delay. On the other hand, the destination node receives other RREQ packets without triggering RREP packets. Under extensive NS-2 simulations with realistic TCP traffic, MTS demonstrates a remarkable improvement in the data security as well as TCP performance.

The rest of the paper is organized as follows. In the next section, we briefly outline the related work. In Section III, we describe the proposed algorithm—Multipath TCP Security (MTS). Simulation results are presented and discussed in Section IV. Finally, we conclude in Section V.

II. Related Work

Designed for ad hoc wireless networks, DSR [9] and AODV [15] are two well-known routing protocols. In [2], several routing protocols are compared using UDP traffic. In [3], [4], the TCP performance degradation due to link failures is investigated. To the best of our knowledge, there is little investigation on TCP performance over multipath routing. Thus, it is an interesting problem to study the TCP performance, especially the TCP security over multipath routing in ad hoc wireless networks.

Many multipath routing approaches [6], [7], [10], [11], [12], [16], [17] have been studied recently. Tsirgos and Haas [16], [17] have developed an analytical framework for evaluating multipath routing algorithms in ad hoc wireless networks. SMR [6], which is an extension of DSR and uses similar route discovery method as in DSR, builds multiple paths by forwarding duplicate RREQs having different incoming link than the first RREQ. In SMR, only the source node has alternative paths. Unfortunately, [7] shows that SMR behaves worse than using only single path with TCP traffic. The reason is that when TCP packets are being transmitted concurrently, TCP reacts to RTT sensitively and the out-of-order packets will lead to unnecessary congestion control.

To tackle this problem, Lim [7] proposes a backup path scheme based on SMR. The algorithmic improvement is that only one path is the current route and the other paths are for backup paths. Once the current route is broken down, the backup path is switched to be the primary route. The other multipath routing protocols in [11], [12] (DSR extensions) and [10] (AODV extension) provide all intermediate nodes in the primary route with alternative paths. Thus, when the route is broken, the intermediate nodes can be rescued by alternative paths. There is one common feature in most existing multipath routing protocols—among all routes, one is for use and the others are in the waiting list. When the current route is broken, another one is chosen to be the route from the waiting list.

Security over multipath protocols [1], [5], [8], [14] have also been studied recently. In [1], [14], existing multiple paths are already known and can be used immediately. The security is provided by introducing redundant information to the divided data. The enhanced security comes with a cost of received efficiency. In [5], the idea of multipath discovery is similar to [8], which applies to wired networks. The idea is that the intermediate nodes are not allowed to send RREPs to the source node and when the destination node receives the first RREQ, it waits for some time to receive other RREQs in order to compute disjoint paths within some hop difference. After finding multiple paths, one path is considered as important and the other paths are alternatives. However, the frequent topology change leads to the backup paths to be stale.

III. The Proposed Algorithm

In this section, we first present an overview of the main features of our proposed algorithm, called *Multipath TCP Security* (MTS). We then introduce the key steps in detail.

A. Protocol Overview

Compared with other multipath routing protocols, the main distinguishing feature of our proposed algorithm is that the source node adaptively chooses the available routes rather than exhaustively testing the “stored routes” one by one. That is, the current route for an active TCP session is not fixed but is dynamically changed over time.

Specifically, the proposed MTS algorithm finds routes on-demand using a route discovery procedure. The source node tries to discover disjoint and loop-free routes. The destination node keeps sending checking packets periodically to make sure if these paths are still alive. Thus, the source node can adaptively choose one of these legitimate paths to replace the current route, if any of the former is found to be better in enhancing the data security.

B. Route Discovery in MTS

Since no permanent route is stored in any node in the network, the source node initiates a route discovery procedure by generating a route request (RREQ) packet when it has packets to send to the destination node. The RREQ includes the following information: packet type, source address, destination address, broadcast ID, hop coun the source, and list of intermediate nodes. Whenever the source node generates a RREQ, the broadcast ID is incremented by one. Thus, the source and destination addresses, together with the broadcast ID, uniquely identify this RREQ packet. The source node broadcasts the RREQ to all nodes within its transmission range. These neighboring nodes will then relay the RREQ to other nodes in the same fashion.

As the RREQ is broadcast in the whole network, some nodes may receive several copies of the same RREQ. An intermediate node will not relay the duplicate RREQ by checking the broadcast ID, source and destination addresses. If a RREQ has been received before, the intermediate node simply drops it; otherwise, the RREQ is stored in the intermediate node and then forwarded to the neighbors after several book-keeping tasks are done: caching the broadcast ID to make sure not to receive the copies of this RREQ; inserting its own address into the list of intermediate nodes; increasing the hop count by one; constructing the reverse path according to the information of the RREQ. The RREQ relaying process continues until the RREQ reaches the destination node. Even in the case where an intermediate node has a fresh route to the destination node, it has to relay the received RREQ.

Eventually, the destination node receives several copies of the RREQ from the same source via different routes. Upon receiving the first RREQ, the destination node generates a route reply (RREP) packet, which includes: packet type, source address, destination address, route reply ID, hop count, and list of intermediate nodes. The RREP is unicast to the source along the reverse path. When the destination node receives other copies of the RREQ, it firstly checks if they are disjoint paths (elaborated in the following section) and then stores the distinguishing information about the disjoint paths. In order to save space, the number of disjoint paths is not more than five in our current configuration.

Figures 1 and 2 illustrate how the RREQ is broadcast and the RREP is unicast in an ad hoc wireless network. It should be noted that several reverse paths are constructed leading toward the destination. Because the copies of RREQ are not simply discarded, the destination node can make full use of the information to construct the reverse paths.

C. Loop Freedom and Disjoint Paths

Sequence numbers are used for ensuring loop freedom. Every node maintains a monotonically increasing sequence number for itself. Furthermore, each node maintains the highest known sequence number for the destination in the routing table, i.e., the destination sequence

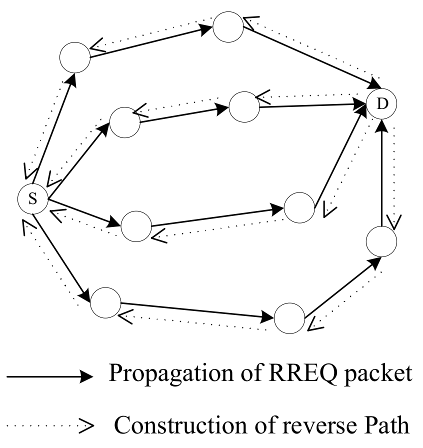

Fig. 1. Broadcasting of RREQ.

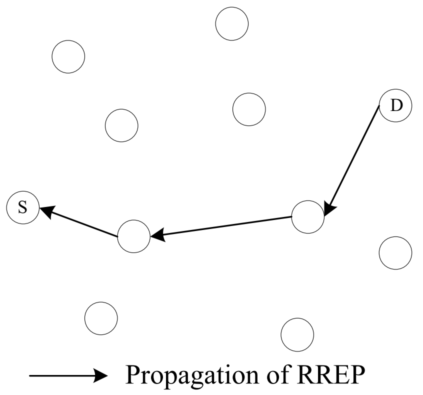

Fig. 2. Unicasting of RREP.

number, which is monotonically increasing too. A higher destination sequence number means more recent routing information. Every node keeps the latest route to the destination according to the information of the destination sequence number.

Since the intermediate node only receives the first copy of RREQ and drops other copies, this mechanism makes sure that the paths before the destination node are disjoint. However, the destination node receives all copies of RREQ, leading to paths that are possibly not disjoint. Figure 3 illustrates the route discovery process. As can be seen, nodes b and c are the neighbors of the destination node. The paths terminated at nodes b and c are disjoint. Unfortunately, the paths S-a-b-D and S-a-b-c-D are not disjoint. To tackle this problem, we use the following rule [10] to check disjoint paths at the destination: if every node on a path ensures that all paths to the destination from that node differ in their next and last hops, then the two paths are disjoint.

D. Route Checking in MTS

Because the wireless channel is time varying, the discovered routes may not be useful after a while. Ou

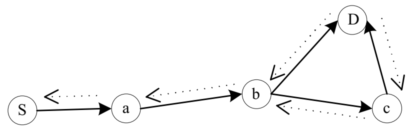

Fig. 3. Paths can possibly be not disjoint.

is to let the destination node periodically transmit checking packets to make sure that the discovered routes are still legitimate. The checking period depends on the coherence time of the fading/shadowing conditions and typically two to four seconds is acceptable [5]. During the life time of a communication session, the source node can receive several checking packets periodically from the destination node and thus, it can choose the best route available accordingly.

We choose at most five candidate paths in the destination for route checking. These checking packets are periodically transmitted to the source along the disjoint paths by the destination node. Each checking packet includes the following information: packet type, checking packet ID, hop count, and list of intermediate nodes. Whenever the five checking packets are sent out concurrently, the checking packet ID is increased by one. When the intermediate node receives the checking packets, it caches the checking packet ID as the entry ID to the destination. This entry ID is used to check if the route is fresh or not. Figure 4 shows how the checking packets are unicast to the source along the different disjoint paths and consequently, the forward paths are constructed.

If some checking packets cannot reach the source node, a checking error packet is sent to the destination (the same mechanism as route error (RERR) [15]). Then the destination node deletes the failed path and chooses another path if available. When a new RREQ packet (having larger broadcast ID) reaches the destination, all the existing legitimate paths are flushed.

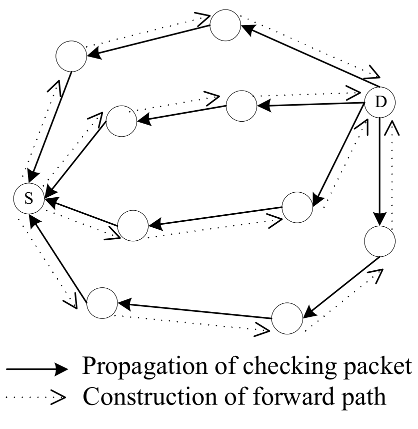

Fig. 4. Route checking.

E. Route Update and Maintenance

When the checking packets are received, the source node chooses the best path to update. We use a simple principle to judge which path is the best—the route of the first arrived checking packet used is considered the best. Since the wireless channel quality is time varying, the best path varies over time.

The feedback from the MAC layer can also be used to detect the connectivity of the link. When a node notifies that its downstream node is out of its transmission range, the node generates a route error (RERR) to its upstream node until it reaches the source node. The source node then triggers a new route discovery procedure.

IV. Performance Results

In this section, we first describe the simulation environment used in our study, followed by the definitions of the performance metrics. We then discuss the results in detail.

A. Simulation Environment

Our simulations are implemented in Network Simulator (NS-2) [13] from Lawrence Berkeley National Laboratory (LBNL) with extensions for wireless links form the Monarch project at Carnegie Mellon University. The simulation parameters are as follows:

- number of nodes: 50;
- testing field: $1000m \times 1000m$;
- mobile speed: uniformly distributed between 0 and MAXSPEED (we choose MAXSPEED $= 2, 5, 10, 15, 20 m/s$, respectively);
- mobility model: random way point model (when the node reaches its destination, it pauses for several seconds, e.g., $1s$, then randomly chooses another destination point within the field, with a randomly selected constant velocity);
- traffic load: TCP Reno traffic source;
- radio transmission range: 250 m; and
- MAC layer: IEEE 802.11b.

Each simulation is run for 200 seconds and repeated for 5 times. We compared three protocols in our simulations: DSR, AODV, and our proposed MTS.

B. Performance Metrics

We designate one randomly selected intermediate node as an eavesdropping node in our simulations. This node performs the same procedures as other legitimate nodes to relay packets but it also collects unauthorized data within its radio range. The interception ratio [5] is used main performance metric to measure the confidentiality. The interception ratio is defined as follows:

$$R_i = \frac{P_e}{P_r} \tag{1}$$

where R_i is the interception ratio, P_e is the total number of packets successfully eavesdropped, and P_r is the total number of packets arrived at the destination.

Intuitively, if more nodes relay packets during the TCP session, less packets are eavesdropped by the single malicious node. Thus, the number of participating nodes is also an important metric. We define a "participating node" as any intermediate node which has relayed at least one packet during the session.

Figure 5 demonstrates the situation where a number of participating nodes in the communication have different speeds. As can be seen, the proposed MTS algorithm has the largest number of participating nodes. The reason is that several routes can be chosen by the source node in our algorithm. The source node chooses different routes when the route checking packets are received. Thus, the security level provided is higher.

Another useful metric is the standard deviation of number of relayed packets. This metric is used to measure the difference among the numbers of received packets in all participating nodes. However, the result from the absolute value of the number of received packets is in no way comparable between different routing protocols. It is because the total received packets at destination and the received packets by the intermediate nodes are different in every run. Even in the same routing protocol, the absolute values are not the same in different scenarios. Thus, we firstly get the sum of all received packets by participating nodes, denoted by α:

$$\alpha = \sum_{i=1}^{N} \beta_i \tag{2}$$

where N is the number of participating nodes and β_i is the number of received packets of node i. Although a packet should be relayed several times before reaching the destination, we focus on not the packet itself but the times. Normalization of these values so as to obtain the percentage γ of received packets is the second step, i.e., the number of received packets divides the sum in each participating node:

$$\gamma_i = \frac{\beta_i}{\alpha} \tag{3}$$

After normalization, the values are between 0 and 1. They are percentage in the total times of relay. Thirdly, the standard deviation is calculated by the percentage values:

TABLE I

NORMALIZATION OF THE RECEIVED PACKETS IN THE PARTICIPATING NODES

Node ID	β	γ
2	10581	34.70%
3	283	0.93%
17	1	0.00328%
21	3886	12.75%
23	1	0.00328%
28	15458	50.70%
36	275	0.90%
45	1	0.00328%
α	Standard Deviation	
30486	19.60%	

$$\sigma = \sqrt{\frac{\sum_{i=1}^{N} (\gamma_i - \overline{\gamma_i})^2}{N}} \tag{4}$$

The last step is to calculate the average of standard deviation in different scenarios. Table I shows how to calculate the standard deviation in one scenario with DSR. As can be seen, eight nodes participate in the communication. In the upper part of the table, the first column shows the name of the node; the second column shows the number of received packets by each node; after calculating the sum of the second column, the third column can be calculated. The last column is the normalized values to calculate the standard deviation. In the lower part, the sum and standard deviation results are shown. Figure 6 shows that our algorithm has the lowest standard deviation, which means that the relay process does not rely on some participating node.

The last performance metric is the highest interception ratio. The largest number of received packets should be P_e in Equation 1. This metric is to demonstrate the worst case, i.e., the most dependent node is the eavesdropper. Figure 7 shows that the highest interception ratio is the lowest in our algorithm, when compared to DSR and AODV.

C. Results

As mentioned before, when compared with DSR and AODV, our algorithm has the highest number of participating nodes, lowest standard deviation of number of relayed packets and lowest highest interception ratio. Thus, the eavesdropper has relatively low possibility to get more information in our algorithm. At the same time, our algorithm does not much depend on one or several nodes. This also ensures the confidentiality.

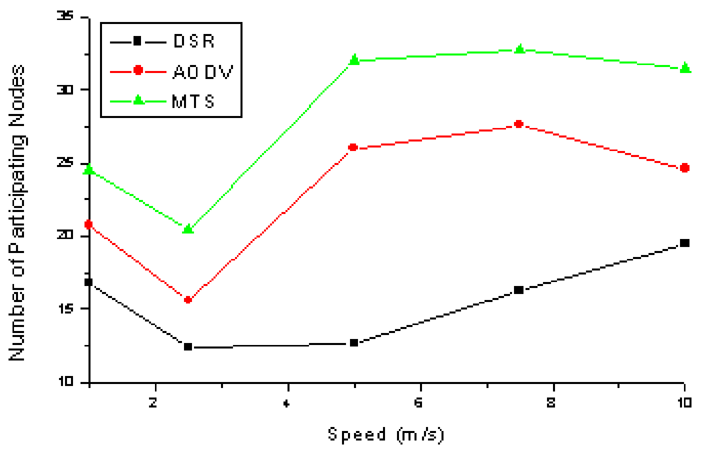

Fig. 5. The number of participating nodes under different speeds.

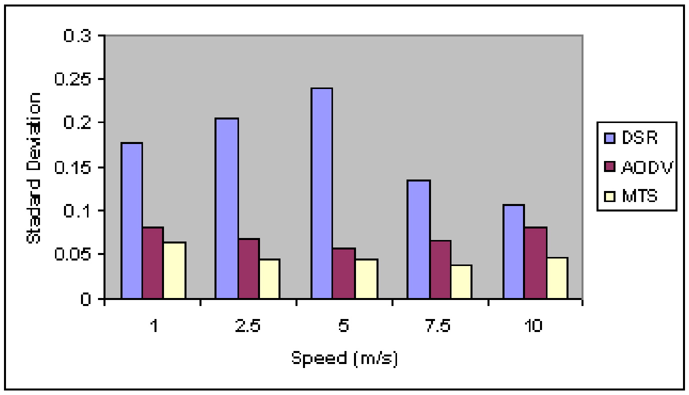

Fig. 6. The standard deviation of number of relayed packets.

To evaluate TCP performance in different routing protocols, we compare them using four metrics:

- Average End-to-End Delay: the interval between sending by the source node and receiving by the destination node, which includes the processing time and queuing time.
- Throughput: the successfully received TCP packets at the destination node.
- Delivery Rate: the ratio of packets reaching the destination node to the total packets generated at the source node.
- Control Overhead: the total routing packets.

Figure 8 shows the average end-to-end delay of each protocol. It should be noted that the delay is effective delay of the packets that actually arrive at the destinations. We can see that DSR has lower delay than AODV. The reason is that DSR caches recently used routes and AODV is an on-demand protocol. As MTS always chooses the best route, it has the lowest delay.

The TCP throughput of each protocol over the simulation time is shown in Figure 9. Obviously, since DSR has some idle time, it has lower throughput, especially when

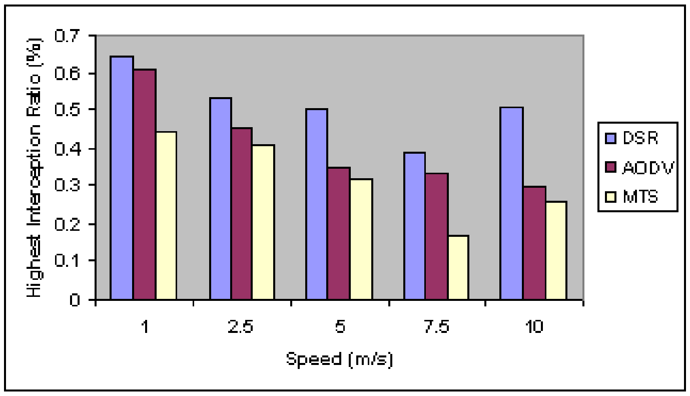

Fig. 7. The highest interception ratio.

the speed is large. On the other hand, MTS has higher throughput than AODV due to the best route chosen. Although our algorithm has much more number of participating nodes in the communication, the reason for the higher throughput is not the increased hop count, but the changing to the best route upon receiving route checking packets. Thus the throughput of our algorithm is the best.

Figure 10 shows the delivery rate. DSR's delivery rate decreases dramatically with increasing device speed. This is because when the speed increases, cached routes becoming stale is more common. In other routing protocols, the delivery rate has little change with increasing speed.

Figure 11 shows the control overhead required by the transportation of the routing packets. Again, because of idleness, DSR has lower control overhead. Our algorithm has the highest control overhead. The reason is that our algorithm transmits route checking packets to assist finding the best path.

In summary, our algorithm has good security and throughput with the cost of increased routing overhead.

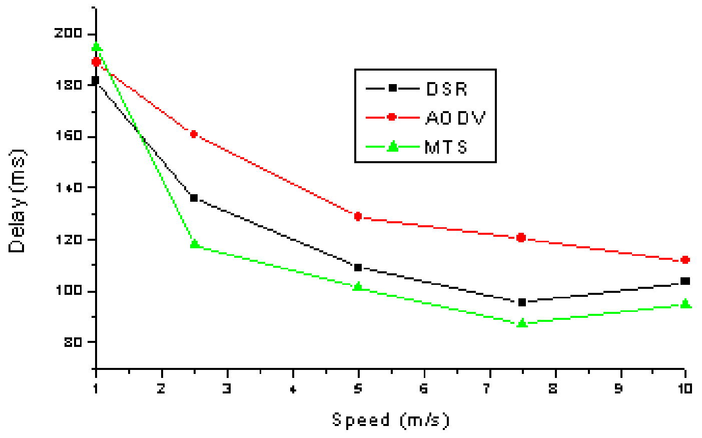

Fig. 8. Average delay.

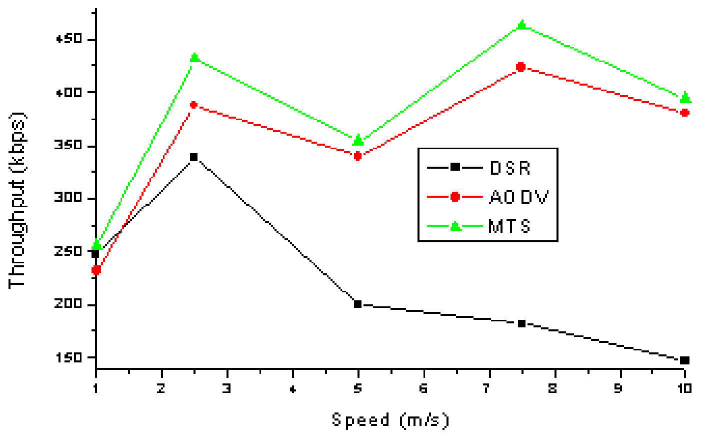

Fig. 9. Average throughput.

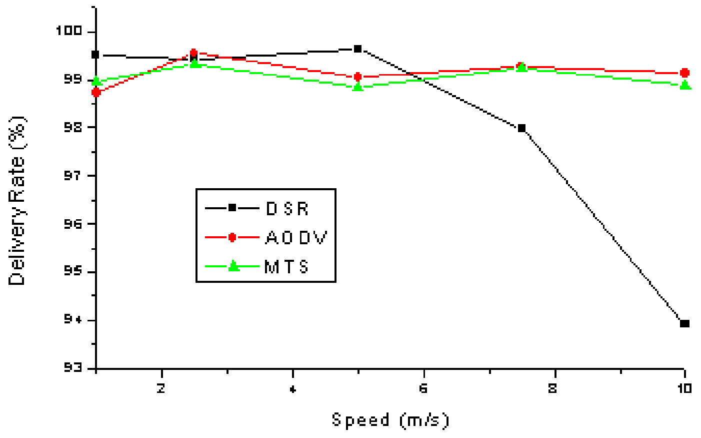

Fig. 10. Average rate of successful delivery of packets.

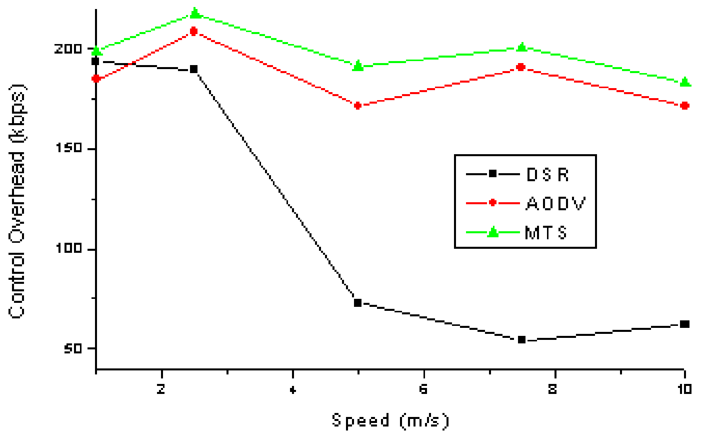

Fig. 11. Average control overhead.

V. Conclusions

In this paper, we propose a multipath routing algorithm to enhance the TCP security in ad hoc wireless networks. The simulation results show that MTS outperforms DSR and AODV not only in data security but also TCP performance.

References

[1] S. Bouam and J. Ben-Othman, "Data Security in Ad Hoc Networks Using Multipath Routing," *Proc. PIMRC 2003*, vol. 2, pp. 1331–1335, Sept. 2003.

[2] J. Broch, D. A. Maltz, D. B. Johnson, Y.-C. Hu, and J. Jetcheva, "A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols," *Proc. MOBICOM 1998*, pp. 85–97, Oct. 1998.

[3] K. Chandran, S. Raghunathan, S. Venkatesan, and R. Prakash, "A Feedback-Based Scheme for Improving TCP Performance in Ad Hoc Wireless Networks," *IEEE Personal Communications*, vol. 8, no. 1, pp. 34–39, Feb. 2001.

[4] G. Holland and N. H. Vaidya, "Analysis of TCP Performance over Mobile Ad Hoc Networks," *Proc. MOBICOM 1999*, pp. 219–230, Feb. 1999.

[5] C. K.-L. Lee, X.-H. Lin, and Y.-K. Kwok, "A Multipath Ad Hoc Routing Approach to Combat Wireless Link Insecurity," *Proc. ICC 2003*, vol. 1, pp. 448–452, May 2003.

[6] S.-J. Lee and M. Gerla, "Split Multipath Routing with Maximally Disjoint Paths in Ad Hoc Networks," *Proc. ICC 2001*, vol. 10, pp. 3201–3205, June 2001.

[7] H. Lim, K. Xu, and M. Gerla, "TCP Performance over Multipath Routing in Mobile Ad Hoc Networks," *Proc. ICC 2003*, vol. 2, pp. 1064–1068, May 2003.

[8] W. Lou and Y. Fang, "A Multipath Routing Approach for Secure Data Delivery," *Proc. MILCOM 2001*, vol. 2, pp. 1467–1473, Oct. 2001.

[9] D. A. Maltz, J. Borch, J. Jetcheva, and D. B. Johnson, "The Effects of On-Demand Behavior in Routing Protocols for Multihop Wireless Ad Hoc Netwroks," *IEEE J. Selected Areas in Communications*, vol. 17, no. 8, pp. 1439–1453, Aug. 1999.

[10] M. K. Marina and S. R. Das "On-Demand Multi Path Distance Vector Routing in Ad Hoc Networks," *Proc. ICNP 2001*, pp. 14–23, Nov. 2001.

[11] A. Nasipuri and S. R. Das, "On-Demand Multipath Routing for Mobile Ad Hoc Networks," *Proc. ICCN 1999*, pp. 64–70, Oct. 1999.

[12] A. Nasipuri, R. Castaneda, and S. R. Das, "Performance of Multipath Routing for On-Demand Protocols in Mobile Ad Hoc Networks," *Mobile Networks and Applications*, vol. 6, no. 4, pp. 339–349, Aug. 2001.

[13] NS, The UCB/LBNL/VINT Network Simulator (NS), http://www.isi.edu/nsnam/ns/, 2004.

[14] P. Papadimitratos and Z. J. Haas, "Secure Data Transmission in Mobile Ad Hoc Networks," *Proc. ACM WiSe 2003*, pp. 41–50, Sept. 2003.

[15] C. E. Perkins, E. M. Royer, and S. R. Das, "Ad Hoc On-Demand Distance Vector(AODV) Routing," *IETF Internet Draft*, draft-ietf-manet-aodv-10.txt, 2002.

[16] A. Tsirigos and Z. J. Haas, " Multipath Routing in the Presence of Frequent Topological Changes," *IEEE Communications Magazine*, vol. 39, no. 11, pp. 132–138, Nov. 2001.

[17] A. Tsirigos and Z. J. Haas, “Analysis of Multipath Routing—Part I: The Effect on the Packet Delivery Ratio,” *IEEE Trans. Wireless Communications*, vol. 3, no. 1, pp. 138–146, Jan. 2004.