2008

Design & Evaluation of Path-based Reputation System for MANET Routing

Ji Li
San Jose State University

Design & Evaluation of Path-based Reputation System for MANET Routing

CS298 Report

Ji Li

Advisor: Dr. Melody Moh

Fall, 2008

APPROVED FOR Ji Li

THE DEPARTMENT OF COMPUTER SCIENCE

Prof. Melody Moh, Advisor,

Department of Computer Science

Prof. Teng Moh, Committee Member,

Department of Computer Science

Prof. Mark Stamp, Committee Member,

Department of Computer Science

Acknowledgements

To Professor Melody Moh, for guiding me on my research, and providing all the advice and encouragement during my Master studies,

to Professor Teng Moh and Professor Mark Stamp, for sitting in on my Master writing project committee, reading my report, and providing valuable advice and comments on my research,

to Mr. Thomas Belote, for useful discussions on research,

to my parents, Mr. Qinghui Li and Mrs. Guanglan Qian, for their love, encouragement and support,

to my wife, Mrs. Yan Li, for her love and endless support,

THANK YOU ALL VERY MUCH.

Abstract

Most of the existing reputation systems in mobile ad hoc networks (MANET) consider only node reputations when selecting routes. Reputation and trust are therefore generally ensured within a one-hop distance when routing decisions are made, which often fail to provide the most reliable, trusted route. In this report, we first summarize the background studies on the security of MANET. Then, we propose a system that is based on *path reputation*, which is computed from *reputation* and *trust* values of each and every node in the route. The use of *path reputation* greatly enhances the reliability of resulting routes. The detailed system architecture and components design of the proposed mechanism are carefully described on top of the AODV (Ad-hoc On-demand Distance Vector) routing protocol. We also evaluate the performance of the proposed system by simulating it on top of AODV. Simulation experiments show that the proposed scheme greatly improves network throughput in the midst of misbehavior nodes while requires very limited message overhead. To our knowledge, this is the first path-based reputation system proposal that may be implemented on top of a non-source based routing scheme such as AODV.

List of Acronyms

AODV	Ad-hoc On-demand Distance Vector
ARAN	Authenticated routing for ad hoc networks
CBR	Constant Bit Rate
CONFIDANT	Cooperation Of Nodes: Fairness In Dynamic Ad-hoc NeTworks
CORE	COllaborative REputation
DOS	Deny of Service
DSDV	Destination-Sequenced Distance-Vector
DSR	Dynamic Source Routing
MANET	Mobile Ad-hoc NETworks
SAFE	Securing pAcket Forwarding in ad hoc nEtworks
SAODV	Secure Ad-hoc On-demand Distance Vector
SEAD	Secure Efficient Ad hoc Distance vector
SMRTI	Secure MANET Routing with Trust Intrigue
SRP	Secure Routing Protocol
P2P	Peer to peer
PDR	Packet delivery ratio
PGP	Pretty Good Privacy
WD	Watch dog
WMN	Wireless Mesh Network

Table of Contents

1. Introduction

Mobile ad hoc networks (MANET) are communication networks in which nodes can dynamically establish and maintain connectivity with each other. Each node can also act as a router to forward packets on behalf of other nodes. Their major advantages include low cost, simple network maintenance, and convenient service coverage. These benefits, however, come with a cost. One of the main challenges is ensuring security and reliability in such networks that are dynamic and versatile in nature.

Reputation systems in MANET are intended to address the challenge. They are effectively in detecting nodes that are selfish or malicious. Most existing ones, however, consider only neighbors' reputations when selecting routes. In such systems, each node selects the next hop from its neighbors based on their reputation and trust values, thus trustworthiness is only ensured in a one-hop distance when making routing decisions. Since no node in the route has a complete view of the entire route, it is hard to select the best route when the path contains multiple hops. These greedy approaches usually do not provide most reliable routes [Buchegger and Le Boudec 2002; Michiardi and Molva 2002; Bansal and Baker 2003; Buchegger and Le Boudec 2003; Rebahi, Mujica et al. 2005; Hu and Burmester 2006]. Furthermore, almost all existing works built their reputation systems on top of DSR (Dynamic Source Routing) [Marti, Giuli et al. 2000; Buchegger and Le Boudec 2002; Michiardi and Molva 2002; Bansal and Baker 2003; Buchegger and Le Boudec 2003; Rebahi, Mujica et al. 2005; Hu and Burmester 2006].

In this report we propose a reputation system that maintains path reputation based on reputation and trust of every node in the path and each node evaluates the path reputation from itself to the destination node. In the system, nodes' reputations are calculated on direct observations and second hand information, and an ALARM message is introduced to exchange the second hand information between nodes. The path reputation is carried by control messages of the underline routing protocol. We use innovative ways to increment and decrement trust values corresponding to positive and negative observations, respectively. We also consider range of values (instead of absolute values) for trust to weigh second-hand information, thus avoid unnecessary transient fluctuations. As a result, the system is very effective in detecting misbehaviors and in ensuring efficient routing.

We believe that this is the first path-based reputation system proposed that is not for source-based MANET routing (such as DSR). AODV [Perkins and Royer 1999] has become a very important MANET routing protocol. It has been chosen as the default routing protocol for IEEE 802.11s, and recently has been implemented in most of new laptops. We believe that proposing a reputation system that can effectively equip AODV against selfish and malicious nodes contributes significantly to MANET research. A summary of major results has been submitted for publication [Li and Moh 2009].

2. Background and Related Studies

Security issues in MANET mostly result from the following attributes of the network:

1. Nodes communicate with each other on a wireless medium which is a shared resource.
2. The network is decentralized and distributed. Therefore there is no central agency to provide trustworthiness throughout the network.
3. Network topology could change dynamically since nodes may have high mobility and are free to join and leave the network. In such networks, to provide secure routing mechanism is difficult because routing updates happen frequently.

Various reputation systems and other approaches were proposed to resolve these issues. In this section, we present a summary of these approaches based on the published studies.

2.1. Reputation Systems for MANET

Reputation systems in MANET are intended to address the challenge of ensuring security and reliability in dynamic and versatile networks. They effectively avoid selfish or malicious nodes from routes based on neighbors' reputations. Many reputation systems have been proposed in MANET. Here, we discuss a few of the most important and popular ones.

2.1.1. CORE

CORE is a reputation system in MANET aiming to solve the selfish node problem [Michiardi and Molva 2002]. CORE is based on DSR and only evaluates the reputations in base system. For each node, routes are prioritized based on global reputations associated with neighbors. The global reputation is a combination of three kinds of reputation that are evaluated by the node. These three reputations are subjective, indirect and functional reputation. The subjective reputation is calculated based on node's direct observation while indirect reputation is the second hand information that sends to the node by the reply message (the reply message could be ROUTE REPLY for routing, or ACK packet for data forwarding). The subjective and indirect reputation is evaluated for each base system functions, such as routing and data forwarding. And the functional reputation is defined as the sum of the subjective and indirect reputation on a specific function. The global reputation is then calculated as a sum of functional reputations with a weight assigned to each function. CORE uses watchdog (WD) mechanism to detect node's misbehaver. For each node, there is a WD associated with each function, and the WD stores an expected result in the buffer for each request. If the expectation is met, the WD will delete the entry for the target node and reputations for all related nodes will be increased based on the list in reply message (reply message contains a list for all nodes successfully participated the service); if expectation is not met or time-out, the WD will decrease the subjective reputation for the target node in the

reputation table. In CORE system, only positive information is sent over network within reply messages, therefore CORE can eliminate the DOS attacks caused by spread negative information over network. And the past observations have more weight than the current observation in CORE.

The advantages of CORE system include that it is a simple scheme, easy to implement, and not sensitive to the resource. CORE uses reply message to transmit the second hand reputation information, thus no extra message is introduced by reputation system. When there is no interaction between nodes, the nodes' reputations are gradually decreased, which discourages the node to misbehave.

The disadvantages of CORE include that it is designed to solve selfish node problem thus it is not very efficient to other malicious problems. CORE is a single layer reputation system with first hand and second hand information having the same weight; it doesn't evaluate trustworthiness before accepting the second hand information, thus the system can not prevent risks on spreading incorrect second hand information. In CORE system, only positive information is exchanged between nodes, therefore another half capability carrying negative information is lost. Reputations are only evaluated between neighbors in one hop distance while the whole path usually contains multiple hops; in consequence, the result is not accurate/optimized for the whole path. Some simulations show that CORE is not very efficient when nodes have high mobility.

2.1.2. CONFIDANT

CONFIDANT, Cooperation Of Nodes: Fairness In Dynamic Ad-hoc NeTworks, provides more comprehensive protection to network than CORE, and it is designed to solve any detectable malicious behaviours [Buchegger and Le Boudec 2002; Buchegger and Le Boudec 2003]. CONFIDANT is based on DSR and uses two layers reputation system. It evaluates not only the node's reputation in the base system but also the trustworthiness for nodes in reputation system. Different from CORE, CONFIDANT allows both positive and negative information exchanged between nodes and introduces an ALARM message for carrying second hand information. When a node receives a second hand information by ALARM message, it first evaluates the trustworthiness of the source node, if it is trustable, the second hand information will be applied to the target node's reputation, if the source node is not trustable, a deviation test will be performed on the received information, if the information is matched the target node's reputation history, it will still be applied to the target node's reputation, if the deviation test is failed (the information is not matched the target's reputation history), the received information will be discard. In CONFIDANT, past observations have less weight than the current one, thus the node can recover from its accidently misbehaves by keep acting correctly in the system; this fading mechanism will encourage nodes to behave correctly. Like CORE, the reputation value will be gradually decreased when there is no interaction between nodes, which discourages misbehaviors. CONFIDANT also has the disadvantage that reputations are only evaluated between neighbors.

The authors also evaluate the performance of CONFIDANT by the comparison with standard DSR. They construct the simulation on GloMoSim [Zeng, Bagrodia et al. 1998] with variables in the total number of nodes in the network, the percentage of malicious nodes, the pause time, and the number of applications. Simulation results show that CONFIDANT performs better than DSR by introducing a small overhead for extra message exchanges (ratio of number of ALARM messages to number of other control messages is 1% -2%), but the performance drops with increasing in nodes' mobility. The results are reasonable since high node mobility indicates high chance to lose routes and high cost for routes reestablishing.

2.1.3. SAFE and Others

SAFE: Securing pAcket Forwarding in ad hoc nEtworks, is another reputation system in MANET [Rebahi, Mujica et al. 2005]. Same as CONFIDANT, SAFE gives more weight on recent observation than the previous ones, and allows positive and negative information exchanged among nodes. In SAFE, when a node detects misbehavior from another node through a direct monitoring, it has to broadcast this information to its neighborhood; and when a neighbor receives the second hand information, instead of directly updating the target node's reputation value with the information it received, it queries the rest nodes in the neighborhood for their opinions regarding the target node and then update the reputation value based on the received information. When a misbehave node recovers and joins again the network, SAFE assigns a "critical" reputation value to it, which makes the node easier to be discard from network when it misbehaves.

Watchdog and pathrater are methods proposed as extensions of DSR to improve the throughput in an ad-hoc network in the presence of nodes that misbehave [Marti, Giuli et al. 2000]. In the network, each node maintains the reputation of every other node it knows about. Different from any above reputation systems, a node calculates an average node reputation in the path and use pathrater to evaluate the reliability of the path by this average reputation instead of just base on neighbors' reputation. Since pathrater has more knowledge of the complete path, it is able to select more reliable route than the neighbor reputation based system. But since each node only maintains reputation for other nodes it knows about, only part of nodes in a path is counted when calculating the average reputation. Also, pathrater relies on the route information in DSR to determine nodes in the path, which make the method is impossible to work on other MANET routing protocols except DSR

2.2. Other Approaches for MANET Security

Security issues in MANET are mainly on AAA (authentication, authorization, and accounting), key management and secure routing. One of the previous studies suggested that the strategies to

enhance security of MANET should focus on embedding security mechanism into different network protocols such as routing and MAC protocols, developing intrusion detection systems to monitor service and respond attacks, and designing multi-protocol-layer security schemes to solve problems in different layers simultaneously [Akyildiz and Wang 2005].

Yih-Chun and Perrig studied on secure wireless Ad-hoc routing protocols [Yih-Chun and Perrig 2004]. They have presented different kinds of secure routing protocols which include:

1. Secure Efficient Ad hoc Distance vector routing protocol (SEAD) [Yih-Chun, Johnson et al. 2002] is a securing routing protocol based on Destination-Sequenced Distance-Vector routing protocol (DSDV) [Perkins and Bhagwat 1994]. Instead of using certificate-chain, this protocol utilizes the sequence number and one-way hash chain to authenticate lower bonder of metric, and then authenticates the routing update entry.
2. Secure Routing Protocol (SRP) [Papadimitratos and Haas 2002] requires a security association only between communicating nodes. This protocol uses this security association just to authenticate ROUTE REQUESTS and ROUTE REPLYS through the use of message authentication codes. However, the protocol doesn't prevent the unauthorized modification of the fields in packets.
3. Authenticated routing for ad hoc networks (ARAN) [Sanzgiri 2002] requires each node in the network has a certificate signed by a trusted authority, which associates its IP address with a public key. And each node in the path verifies the certificate and signature of the previous and the source hops and then forward the ROUTE REQUESTS and ROUTE REPLYS with its own certificate and signature.
4. Secure AODV (SAODV) [Zapata and Asokan 1999] is similar to ARAN. Both of them use a signature to authenticate most fields of a ROUTE REQUEST and ROUTE REPLY, but SAODV utilizes hash chains to authenticate the hop count instead of an extra signature to verify previous hop.

In this paper, they also pointed out that the secure routing problem isn't well modeled in MANET and optimistic approaches are needed to balance between security and performance.

2.2.1 Key Management

The key management is an important task for network security. However, it is difficult because the network has no central authority to manage security keys.

A self-organized public-key scheme is proposed to solve the key management problem in distributed networks [Hubaux, Batttan et al. 2001]. In the proposed scheme, each node in the network stores the certificates issued by itself as well as a set of selected certificates issued by other nodes. When any pair of nodes needs to communicate each other, they first merge the local certificate repositories and then find the appropriate certificate chains within the merged repositories that can pass this public keys verification, after that, they are able to trust each other

and exchange keys to secure their communicate channel. The proposed key management scheme is decentralized and has equivalent role for each node, which make it efficient in distributed network environment. However, this method only has probabilistic guarantees, thus appropriate certificate chains may not be found even if they exist.

In this paper, the authors also evaluate the performance of the algorithm to find appropriate certificate chains based on real Pretty Good Privacy (PGP) trust graphs with variables in trust graph size [Hubaux, Batttan et al. 2001]. The algorithm that they present is more efficient in small graph size but modifications are required for improving scalability.

2.2.2. Trust Enhanced Routing

One study has described the way to use fuzzy trust in P2P networks [Griffiths, Kuo-Ming et al. 2006]. Considering the similarities between P2P networks and Wireless Mesh Networks, this approach can also be applied to wireless mesh routing mechanisms. Fuzzy trust provides unclear border between trusted and untrusted nodes, and defines some middle areas in-between, which is more realistic than the one with the clear border. By using fuzzy trust, nodes' trust can be evaluated more accurately.

A trust enhanced Mobile Ad-hoc NETworks (MANET) routing protocol, Secure MANET Routing with Trust Intrigue (SMRTI), is proposed recently [Balakrishnan, Varadharajan et al. 2007]. SMRTI is based on DSR, and it selects routes by the trust of packet. Packet's trust depends on a few important nodes in the route, such as source node, destination node, previous node and next node. Since each packet caches the whole route in DSR, all necessary information needed to evaluate the trust can be derived from packet, thus, no extra message is introduced to exchange the trust information. Meanwhile, the nodes' mobility should have less impact to the protocol since all the necessary information is contained in the packet. However, SMRTI is specifically designed for DSR and uses some unique features in DSR, which make it hard to be applied in the other routing protocols.

This paper also evaluated the performance of SMRTI by packet delivery ratio (PDR) [Balakrishnan, Varadharajan et al. 2007]. The PDR is defined as the average ratio of total number of CBR data packets, which is received by the destination, to the total number of CBR packets, sent by source. They construct the simulation on NS-2 with variables in proportion of the malicious nodes and network connectivity. The simulation results show that SMRTI performs better than the standard DSR. The performance of SMRTI can keep flat with high node mobility (50m/s) even in the case that 80% nodes in the network are malicious nodes. These results are promising, but may depend on the simulation configuration that authors used. Simulation parameters indicate a density network with relatively long radio range. In this situation, the mobility and percentage of malicious nodes could have less impact to the system.

2.3. AODV

AODV is an on-demand routing protocol [Perkins and Royer 1999]. The route is initiated by broadcasting the route requests message from the source, and the message is then propagated through the entire network. When the destination node or an intermediate node which has a route receives the route request message, it responds with a route reply message. Route request, route reply and route error are the control messages that are sent during the route establishing process for updating the route table of nodes in the route.

One previous study has classified misuses of the AODV protocol into two categories: atomic misuses and compound misuses [Ning and Sun 2003]. Atomic misuses are performed by manipulating a single routing message, while compound misuses are composed of multiple atomic misuses, and possibly normal uses of the routing protocol. They further divide the atomic misuse actions in AODV into the following four categories:

1. "Drop Reply. The attacker simply drops the received routing message."
2. "Modify and Forward. After receiving a routing message, the attacker modifies one or several fields in the message and then forwards the message to its neighbor(s) (via unicast or broadcast)."
3. "Forge Reply. The attacker sends a faked message in response to the received routing message. Forge Reply is mainly related to the misuse of ROUTE REPLY messages, which are in response of ROUTE REQUEST messages."
4. "Active Forge. The attacker sends a faked routing message without receiving any related message"

Similarly, another paper has classified main attacks to AODV as follows [Liu, Li et al. 2005]:

1. "Black Hole Attack - The attacker broadcasts some fraudulent messages to make others believe that data can be transmitted through itself with the shortest path or least cost, while this trickster never forwards these data packets, which forms a "black hole", that is, absorbing in everything but never giving out."
2. "Routing Table Overflow Attack - A malicious node keeps sending a large number of ROUTE REQUEST message for some node that don't exist, which consumes lots of computation and network bandwidth and causes failure to build normal route, even the paralysis of entire network."
3. "Network Segmentation Attack
   - Fabricating ROUTE ERROR Packet Attack: Malicious nodes broadcast fabricated ROUTE ERROR packets to destroy the route table of its neighbors, which causes network segmentation and lower performance.
   - Interrupt Routing Attack: The selfish node drops the received routing messages from its neighbors for limited power and computation ability, which also causes network segmentation."

3. Path-Based Reputation System

3.1. Main Ideas

A. Two Levels of Information: First-hand and Second-hand

The proposed system makes used of both first-hand and second-hand information. First-hand information is direct observations. Second-hand information arrives from *indirect observations*; i.e., a neighbor node's observations of other nodes. Note that RREQ and RREP (AODV control packets) are also considered as second-hand information. All second-hand information will go through a *trust* test, which is discussed next.

B. Two Levels of Credibility: Reputation and Trust

The proposed system contains two levels of credibility. *Reputation* tells whether the node behaves correctly in the base system (routing protocol) while *trust* indicates whether the node lies in the reputation system. Specifically, the trust system evaluates the credibility of second-hand information sent by a neighbor.

C. Two Levels of Reputation: Node-based and Path-based Reputation

In addition to node reputation that is formed by both first- and second-hand information, the proposed system computes path reputation based on node-reputation of all the nodes in the path. Routing decision is then based first on path reputation, then if there is a tie, on path distance. Note that routes that have very low path reputation are discarded even during route discovery phase (by discarding RREQ and RREP packets). This approach significantly improves throughput and reduces protocol message overhead.

D. Range-based vs. Value-based Reputation and Trust

Computing node and path reputation values need to factor in the trust of second-hand information. Instead of using an absolute trust value, the proposed system makes use of *ranges*. That is, if two trust values fall within the same range, they are assigned the same level of trust (same weight). This avoids unnecessary adjustments caused by transient fluctuations that are quite frequent in MANET.

3.2. System Design

In this subsection, we present a high-level view of the system design. Major components will be described in detail in the next subsection. Figure 1 shows the architecture of the proposed system. It includes five components described below. The arrows indicate interactions among these components. Solid arrows are specifically for exchanges of reputation system messages, including direct observations (such as suspicious events), indirect observations (also called ALARM), and trust values. Dotted arrows are for routing control messages RREQ and RREP (they also carry

path reputation values). Note that an extra field is needed on RREQ and RREP messages to hold path reputation values.

The proposed system constrains following components:

1. *Event monitor*: It detects suspicious events and feeds them to the Node reputation manager. It also receives second-hand information (i.e., ALARM messages) from neighbors and routing control messages by interacting with the underlying routing protocol (AODV). It then feeds these indirect observations to the Trust manager.
2. *Trust manager*: It evaluates trust of second-hand information provided by the Event monitor. It then feeds the evaluated information and trust values to either node or path reputation managers.
3. *Node reputation manager*: It computes neighboring node reputations. This is based both on direct observations passed by Event monitor and on evaluated indirect observations and trust values passed by Trust manager.
4. *Path reputation manager*: It handles path reputations based on neighbor node reputations (passed by Node reputation manager) and second-hand path reputations and trust values (passed by Trust manager).
5. *Path manager*: It makes the route decision based on path reputation value fed by Path reputation manager and passes the decision to the underlying routing protocol (AODV).

Among all the components, only event monitor and path manager have interfaces to communicate with the underline routing protocol. An extra field is added into both ROUTE REQUEST and ROUTE REPLY messages to store the path reputation. In our proposal, we defined that the observation is negative when misbehavior is detected by the event monitor, and the observation is positive when the detected behavior is correct.

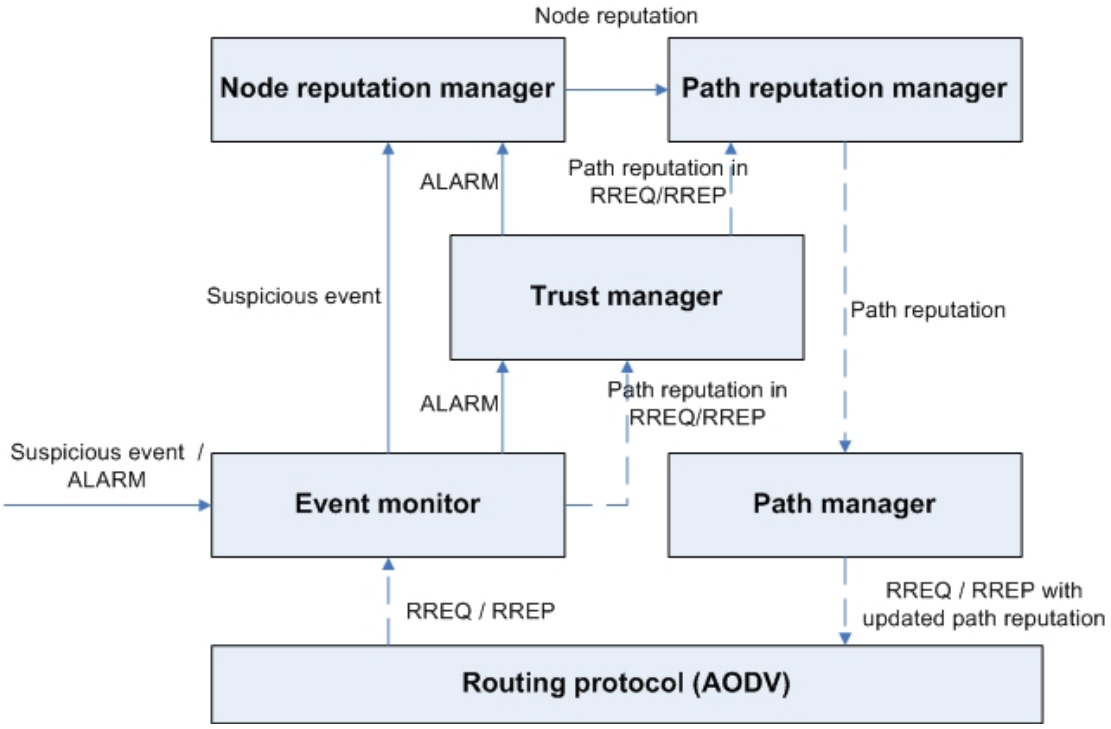


Figure 1. System architecture.

Below, we give the algorism of handling the direct and indirect observations as well as routing control messages. The algorithm runs on every individual node in the network. We list the main variables needed to execute the proposed scheme in a network node:

- D_n: **D**irect observation of node n.
- $I_{n,m}$: **I**ndirect observation of node *n* made by node *m*
- R_n: **R**eputation of node *n* based on direct observation
- S_n: Reputation of node *n* based on **S**econd-hand information
- T_n: **T**rust value of node *n*
- PR_p: **P**ath **R**eputation of route *p*
- $PS_{p,m}$: **P**ath reputation of route *p* received from node *m* (considered as **S**econd-hand information)

Note that all the reputation and trust values are normalized between 0 and 1;

Initialization: $T_m = R_n = 1$, $PR = \varnothing$
Waiting for the incoming event from the monitor module:

1. [EM] If a D_n is detected:

1.a [NRM] If D_n is negative:
    [NRM] Calculate the node reputation R_n by Equation (6)

1.b Else
    [NRM] Calculate the node reputation R_n by Equation (7)

1.c [NRM] If R_n changed more than a threshold (*Th_alarm*), send ALARM to all neighbors

2. [EM] If a $I_{n,m}$ is detected:

2.a [TM] Do the deviation test on $I_{n,m}$ by Equation (1)

2.b [TM] If T_m is trusted OR deviation test is passed:

2.c [NRM] If $I_{n,m}$ is negative:
    [NRM] Calculate the node reputation R_n by Equation (8) & (10)

2.d Else
    [NRM] Calculate the node reputation R_n by Equation (9) & (10)

2.e [TM] If deviation test is passed:
    [TM] Update T_m by Equation (5)

2.f Else
    [TM] Update T_m by Equation (3)

3. [EM] If a *RREQ* is detected:

3.a [PRM] Update the path reputation PR_p by Equation (11)

3.b [PRM] If the path reputation is below a threshold (*Th_route*), discard *RREQ* message

3.c Else
    [PM] Update the path reputation in *RREQ* and broadcast *RREQ* to all neighbors

4. [EM] If a *RREP* is detected:
4.a [PRM] Update the path reputation PR_p by Equation (11)
4.b [PRM] If the path reputation is below a threshold (*Th_route*), discard *RREP* message
4.c Else
4.d [PM] If it is current node requests this route:
[PM] Update PR_p and prioritize routes based on path reputations PR_p
4.e Else
[PM] Update the path reputation in *RREP* and send *RREP* to next hop

3.3. Major Components

In this section, three major components are described in detail.

3.3.1 Trust manager

Recall that it is responsible to evaluate the *trust* of second-hand observations as well as that of second-hand path reputation reported in routing control messages.

(A) Deviation Test and Decrement/Increment Functions
First, in order to prevent false alarms, we adopt the deviation test method [4, 5] to evaluate a new second-hand observation, as shown below. The idea is that a second-hand observation is trusted only if it does not differ too much from the node's own direct observation. In this case, the trust is increased. Otherwise, it is decreased. The increment and decrement are each further guided by a quadratic function to be described below.

Following is the condition to pass the deviation test:

$$|D_n - I_{n,m}| < d \qquad (1)$$

In the above, d is the threshold for the test ($d = 0.1$ is used in this work). If the difference between D_n (direct observation) and $I_{n,m}$ (new indirect observation give by node m) is greater than or equal to d, then $I_{n,m}$ fails the deviation test, and T_m, the trust value of node m that sent this second-hand observation, is decreased by a decrement function, f_D (refer to Figure 2a):

$$f_D(T_m) = \tfrac{1}{2}(T_m)^2 \qquad (2)$$

that is

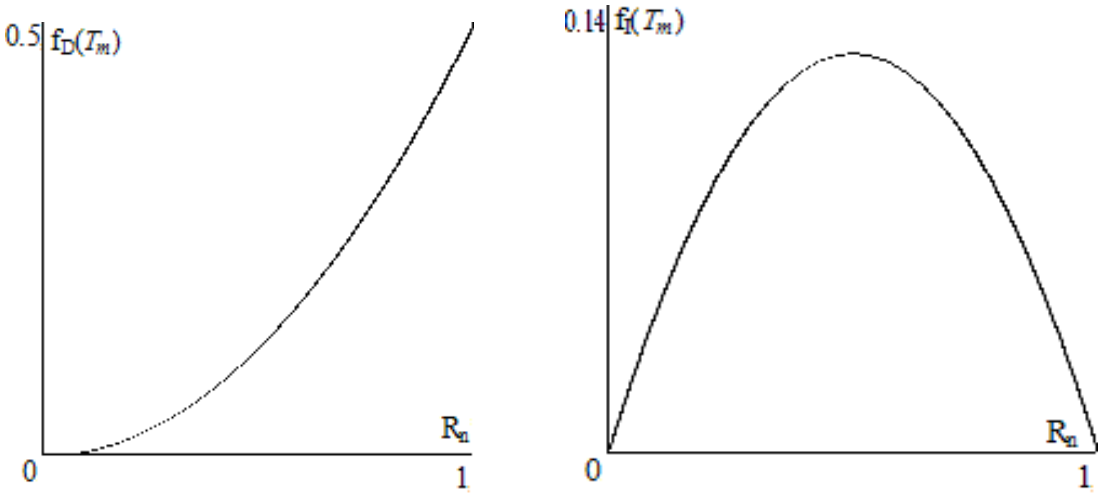
$$T_m = T_m - f_D(T_m) \qquad (3)$$

On the other hand, if the difference between D_n and $I_{n,m}$ is less than d, then the test is passed, and T_m is increased by an increment function, f_I (refer to Figure 2b):

$$f_I(T_m) = T_m - \tfrac{1}{2}(T_m)^2 \qquad (4)$$

that is

$$T_m = T_m + f_I(T_m) \qquad (5)$$



(a). $f_D(T_m)$ (b). $f_I(T_m)$

Figure 2. Decrement and increment functions.

The quadratic function f_D is used such that the decrement of trust is small when the trust value is still small. Yet, when the trust value is large, the corresponding decrement needs to be larger to "punish" more severely the bad behavior. The increment function f_I may be similarly explained. Note that these two functions are also used in node and path reputations.

(B) Ranged-based Trust Values

Second, to prevent transient fluctuations, categorized (or range) values are used for trust (instead of absolute values). In our implementation, the trust value is categorized into three ranges:

$$T_m \in \begin{cases} Trust, & 1 \geq T_m > 0.7 \\ Untrust, & 0.7 \geq T_m > 0.3 \\ Distrust, & 0.3 \geq T_m > 0 \end{cases}$$

This T_m value is then fed to either Node or Path reputation managers to aid their evaluations of node or path reputations.

3.3.2 Node Reputation Manager

The node reputation manager computes node reputation based on both direct observations from the Event monitor and the *trust-evaluated* indirect observations from the Trust manager. For each new direct and indirect observation concerning a node *n*, it computes the new reputation value of node *n* by making use of the decrement and increment functions introduced above (in Equations 1 and 2), and by using a *weighted* combination of direct and indirect observations, as follows:

1. If the new observation is a direct observation
    a. If the observation is negative
    Then $R_n = R_n - f_D(R_n)$ (6)
    b. Else (positive observation)
    $R_n = R_n + f_I(R_n)$ (7)
2. Else (indirect observation, say $I_{n,m}$)

If T_m is trusted **or** $I_{n,m}$ passed the deviation test at the Trust manager

a. If the observation is negative
   Then $S_n = R_n - f_D(R_n)$ (8)
b. Else (positive observation)
   $S_n = R_n + f_I(R_n)$ (9)

3. Update R_n by a weighted combination
   $R_n = (1 - w_1) R_n + w_1 S_n$, (10)
   where (in this work)

$$w_1 = f(T_m) = \begin{cases} 0.1, & T_m \in \{Trust\} \\ 0.05, & \text{if } T_m \in \{Untrust\} \\ 0.025, & T_m \in \{Distrust\} \end{cases}$$

3.3.3 Path Reputation Manager

When the Path reputation manager obtains a new path reputation value (say passed by node n) from the Trust manager, it also receives the trust level of T_n. In addition, it obtains R_n from the Node reputation manager. Based on these values, it computes the new path reputation value, PR_p, by taking the minimum of R_n and a *weighted* value of $PS_{p,n}$, as follows:

$$PR_p = \text{minimum}\,[R_n, w_2 \times PS_{p,n}] \quad (11)$$

where the weight w_2 is a function of T_n and is defined below in our implementation

$$w_2 = \begin{cases} 1, & T_n \in \{Trust\} \\ 0.5, & \text{if } T_n \in \{Untrust\} \\ 0.25, & T_n \in \{Distrust\} \end{cases}$$

If, however, the new PR_p falls below a pre-defined threshold value (*Th_route*), then the RREP or RREQ is dropped. This ensures that only trusted RREP and RREQ are forwarded.

3.4. Base Case: Node-based Reputation System

In order to understand the advantage of the proposed path-based reputation system, we also implemented a node-based reputation system in the simulation. It is identical to the proposed system except that Equation (11) is simply:

$$PR_p = R_n \quad (12)$$

Thus, each node in the network selected route based on the neighbor's (the potential next hop) reputation rather than a path-based reputation.

3.5. System Work Flows

In our proposal, there are four types of event that the system needs to handle. They are:

1. Direct observation of a suspicious event.
2. Second hand observation by an ALARM message.
3. ROUTE REQUEST message from underline routing protocol.
4. ROUTE REQUEST message from underline routing protocol.

We discuss system work flows for every event in detail in this section.

3.5.1. Direct Observation of a Suspicious Event

When a node detects a suspicious event by direct observation, the event monitor forwards the event to node reputation manager for evaluation, and then the node reputation manager updates the reputation by Equation 6 or 7 for the target node accordingly and sends out an ALARM message to the neighbors. Figure 3 illustrates how the system handles a direct observation.

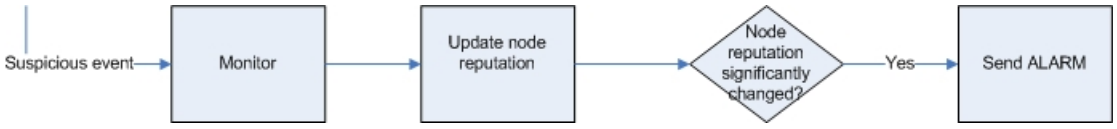


Figure 3. Flowchart for direct observation of a suspicious event.

3.5.2. Second Hand Observation by an ALARM Message

Figure 4 shows the flowchart when a node receives an ALARM message. The event monitor forwards the ALARM to trust manager, which evaluate whether the ALARM is sent by a trusted source. If the ALARM is from an untrusted source, rather than discard the message directly, trust manager performs the deviation test (Equation 1). If the ALARM message passes the deviation test or it is from a trusted source, node reputation manager updates the node reputation (Equation 10) accordingly, otherwise, it is discarded by system. For both cases, trust manager will update the trust of the source node by deviation test result (Equation 3 or 5).

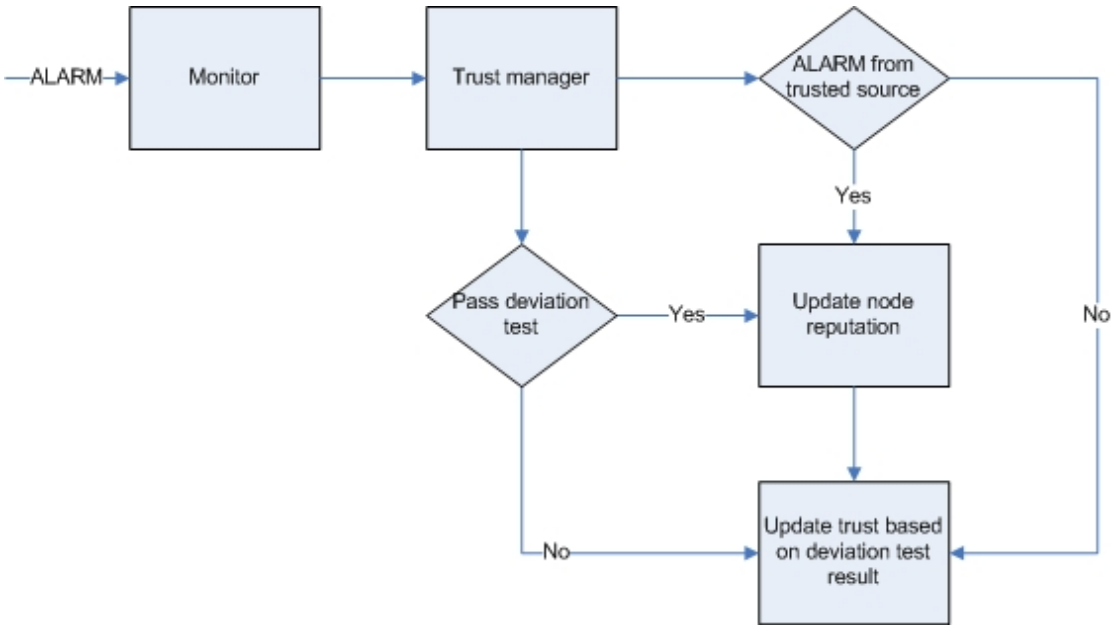


Figure 4. Flowchart for receiving an ALARM message.

3.5.3. ROUTE REQUEST Message from Routing Protocol

As shown in Figure 5, when a node receives an ROUTE REQUEST message, the event monitor forwards the message to path reputation manager. Path reputation manager then checks the path reputation in the message. If the path reputation is below the threshold, the message will be discarded, otherwise, the path reputation in the message will be updated (Equation 11) and the message will be forwarded neighbors.

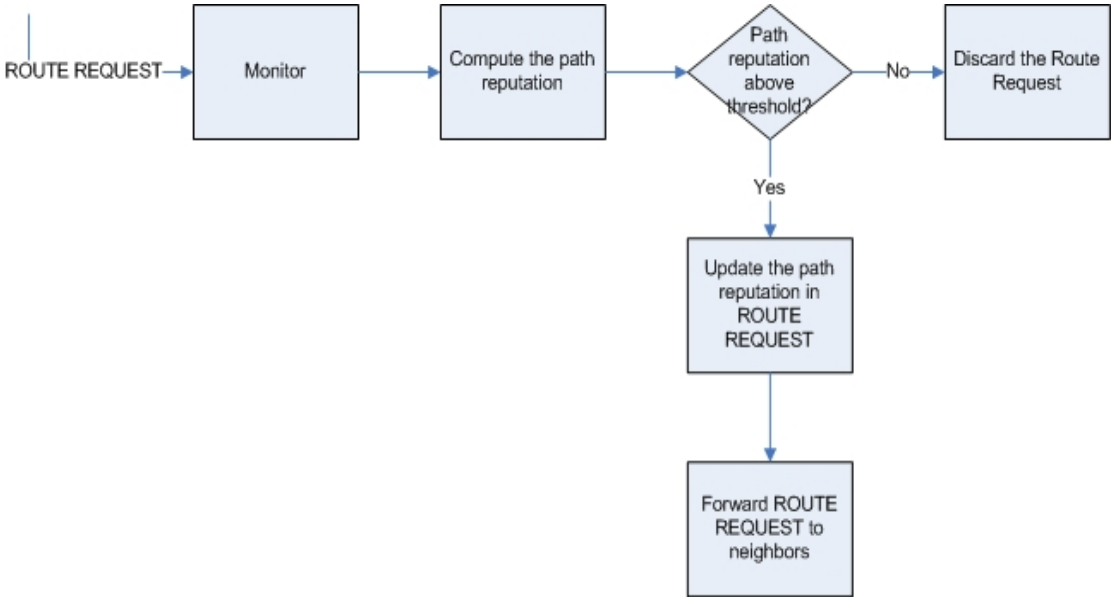


Figure 5. Flowchart for receiving an ROUTE REQUEST message.

3.5.4. ROUTE REPLY Message from Routing Protocol

As shown in Figure 6, when a node receives an ROUTE REPLY message, the event monitor forwards the message to path reputation manager. Path reputation manager then checks the path reputation in the message. If the path reputation is below the threshold, the message will be discarded. Otherwise, the message will be forwarded to the path manager. If it is the current node that requests the route, path manager will prioritize routes based on their path reputation; or path manager will update the path reputation value in the message and then send the message to the next hop.

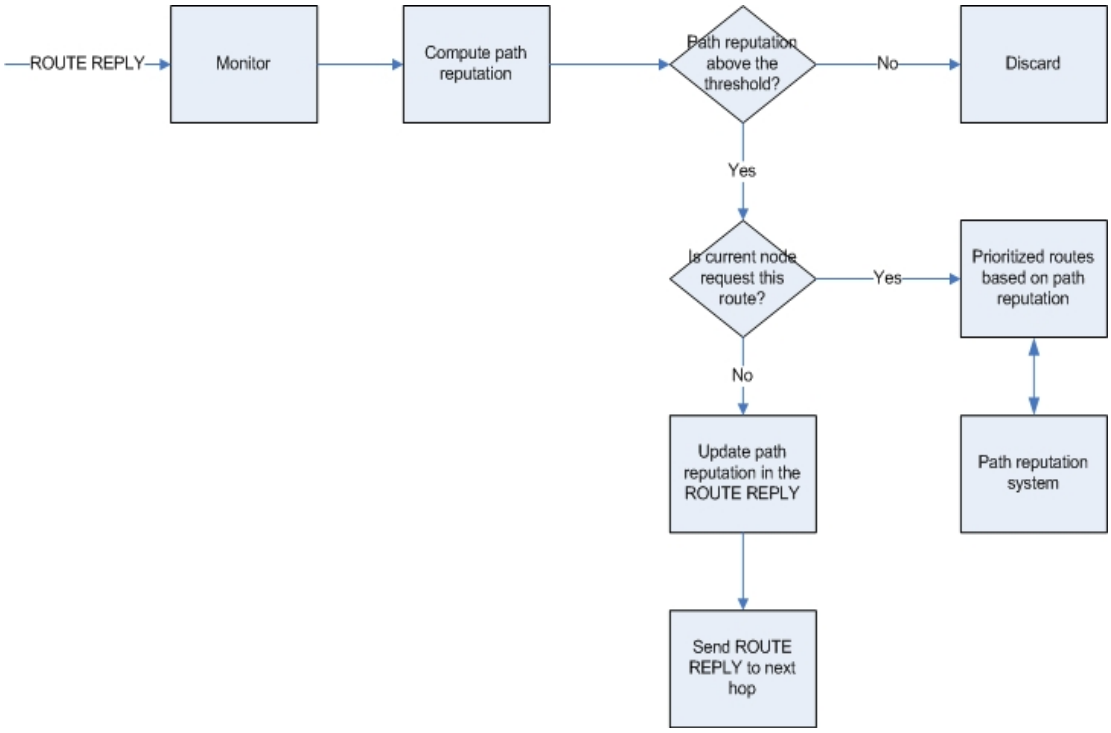


Figure 6. Flowchart for receiving an ROUTE REPLY message.

3.6. Protocol Overhead and Complexity Analysis

In this section we note that the additional overhead required by the proposed system has the same complexity as AODV; this includes time (for computation), space (for storage), and message (for extra control packets) complexities. For each node, reputation and trust values are stored in hash tables, thus the computation of reputation and trust has time complexity O(1) based on equations in section 3.3. The space complexity is O(N) as the maximum table size can be up to N entries for the reputation and trust tables, which has the same space complexity as the AODV when these values are stored in neighbors table. Since path reputation values are carried by the AODV routing control messages, the only extra message we introduced is the ALARM message, which could be

sent to up to N neighbors from each node, thus the message complexity is $O(N^2)$ as there are up to N nodes to send alarm messages in the network, which is the same message complexity as the AODV routing control messages.

4. Performance Evaluation

4.1. Simulation Settings

The simulation was developed on GloMoSim 2.03 [Zeng, Bagrodia et al. 1998] with the standard AODV implementation. The simulated network contained 50 nodes distributed uniformly in a 1600 meters by 1600 meters area. Each simulation experiment lasts for 10 minutes. The mobility mode was set to the Random Waypoint Model; in which nodes moved to a random destination at a speed uniformly distributed between 0 and 5 m/sec and stayed at this destination for 20 sec. Misbehaved nodes are selected randomly, excluding source and destination nodes. Both UDP (supporting CBR) and TCP (supporting FTP) are evaluated, each with 10 pairs of source and destination nodes chosen randomly. Finally, *Th_alarm* = 0.49 and *Th_route* = 0.51 for the reputation system.

4.2. Misbehaviors and Performance Metrics

While there are many possible MANET misbehaviors, in the simulation we implement three major MANET routing behaviors [Ning and Sun 2003] that may be effectively dealt with by a reputation system:

- Data selfish (or selfish in data forwarding) – A misbehaved node does not forward any data packets.
- Forge reply (or warm-hole) attacks – A misbehaved node replies a *RREP* message with hop-count equal to 1 for any incoming *RREQ* message, thus the node always claims that it has the shortest path to the destination.
- Combination of forge reply and data selfish.
- Forge data– A misbehaved node modifies data packets that pass through.

Note that a forge reply attack directly affects (or fakes) a routing path since it replies wrong routing information, whereas a data selfish or forge data misbehavior affects a single node. Simulation results will show that the proposed path-based reputation scheme is most effective when dealing with routing path attacks.

Performance metrics include throughput, average end-to-end packet delay, packet lost ratio, and reputation message ratio, which is the percentage of reputation system messages in the overall

control messages (including those for routing) and represents the reputation system overhead.

4.2.1 Implementation of Misbehaviors Detection

- Data selfish (or selfish in data forwarding) – In simulation, the hello message is used to notify neighbors about the misbehavior. An extra field is added to the hello message to indicate that the source node is selfish in forwarding data packets.

- Forge reply (or warm-hole) attacks – In simulation, misbehave node replies a RREP message with hop-count equal to 1 for any incoming RREQ message, an extra field is added to the RREP packet to indicate that the packet contains incorrect information, so that the next hop who receives packet can detect the misbehavior.

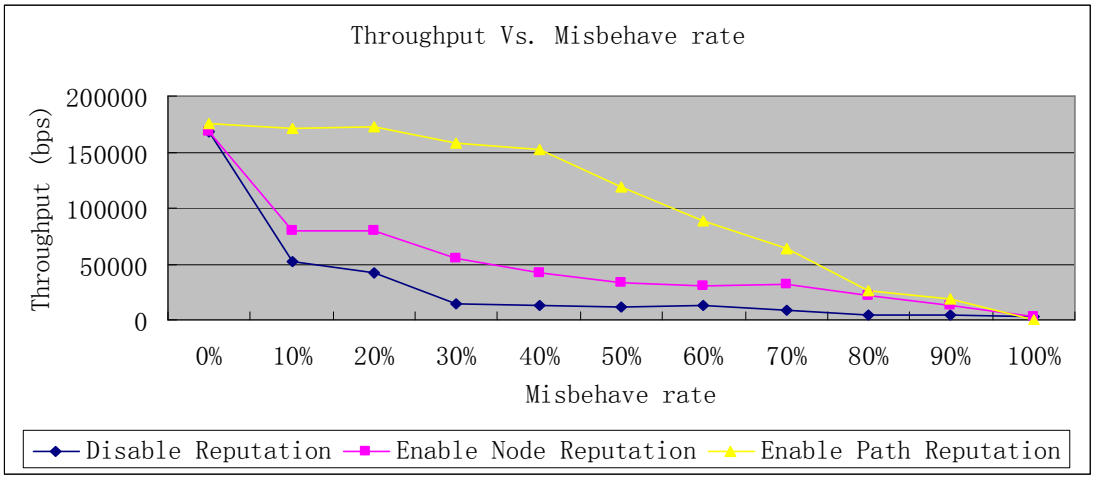
- Forge data – In simulation, an identity string is added to the data payload, thus next hop will know the packet is modified by detecting the identity string. Though the next hop detects the modification, it still forwards the packet, and the destination node drops the packet before it is delivered to the upper layer network. The modified packets are transmitted though the network but they are not counted into the throughput. The limitation of this implementation is that the data payload has to be longer than the identity string size.

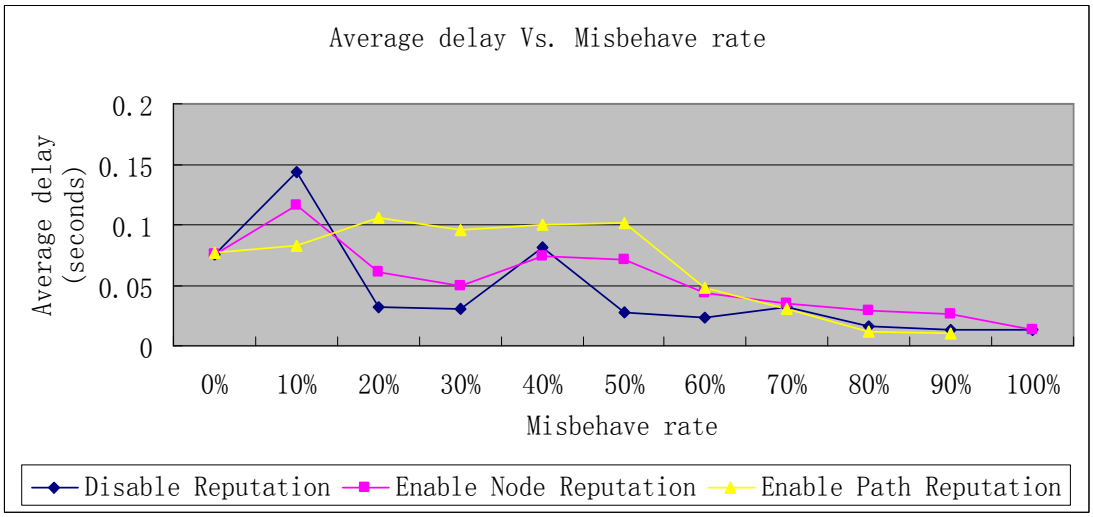
4.3. Simulation Results: CBR over UDP

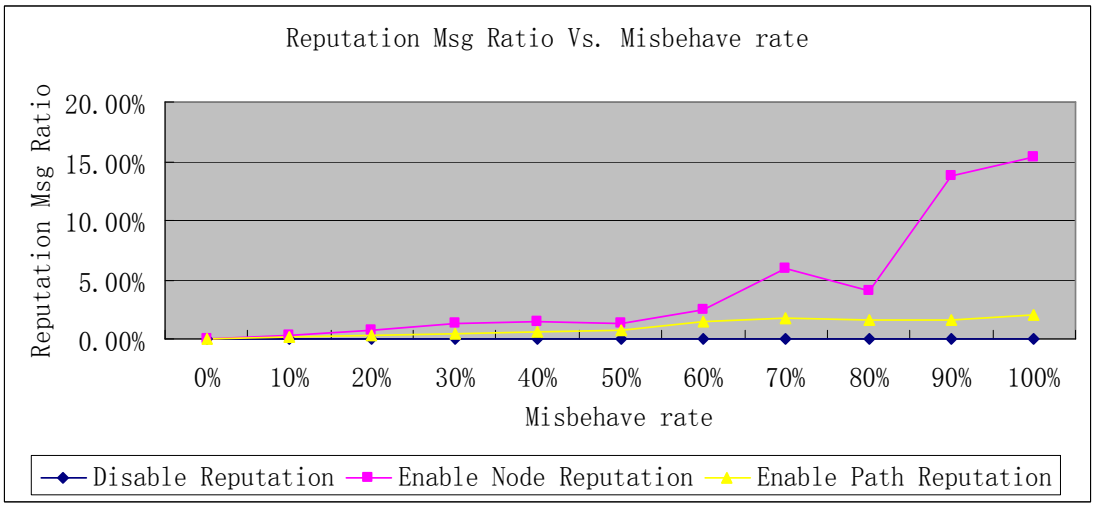
We first set the network application to CBR with 512 bytes packets continuously sent every 200 msec, resulting in approximately 20.5 kbps for each of the10 flows, or a total of 205 kbps in the network.

(A) Misbehavior: Forge Reply

Simulation results are shown in Figure 7, including six sub-figures: throughput, average delay, reputation message ratio, packet lost rate, routing overhead, and reputation overhead; each shows three schemes: reputation-disabled, node-based reputation, and path-based reputation. These results indicate that the path-based reputation system is very effective in dealing with forge reply attack. It has more than doubled throughput comparing with node-based reputation, and more than tripled comparing with the original AODV; while keep the packet lost rate much lower than the original AODV. On the other hand, delay has only been slightly increased as many more packets transmitted. Its reputation message overhead has stayed very low, largely because path reputation is discarded as soon as it falls below *Th_route*.


Throughput Vs. Misbehave rate
Throughput (bps)
200000
150000
100000
50000
0
0%
10%
20%
30%
40%
50%
60%
70%
80%
90%
100%
Misbehave rate
Disable Reputation
Enable Node Reputation
Enable Path Reputation


Average delay Vs. Misbehave rate
Average delay (seconds)
0.2
0.15
0.1
0.05
0
0%
10%
20%
30%
40%
50%
60%
70%
80%
90%
100%
Misbehave rate
Disable Reputation
Enable Node Reputation
Enable Path Reputation


Reputation Msg Ratio Vs. Misbehave rate
Reputation Msg Ratio
20.00%
15.00%
10.00%
5.00%
0.00%
0%
10%
20%
30%
40%
50%
60%
70%
80%
90%
100%
Misbehave rate
Disable Reputation
Enable Node Reputation
Enable Path Reputation

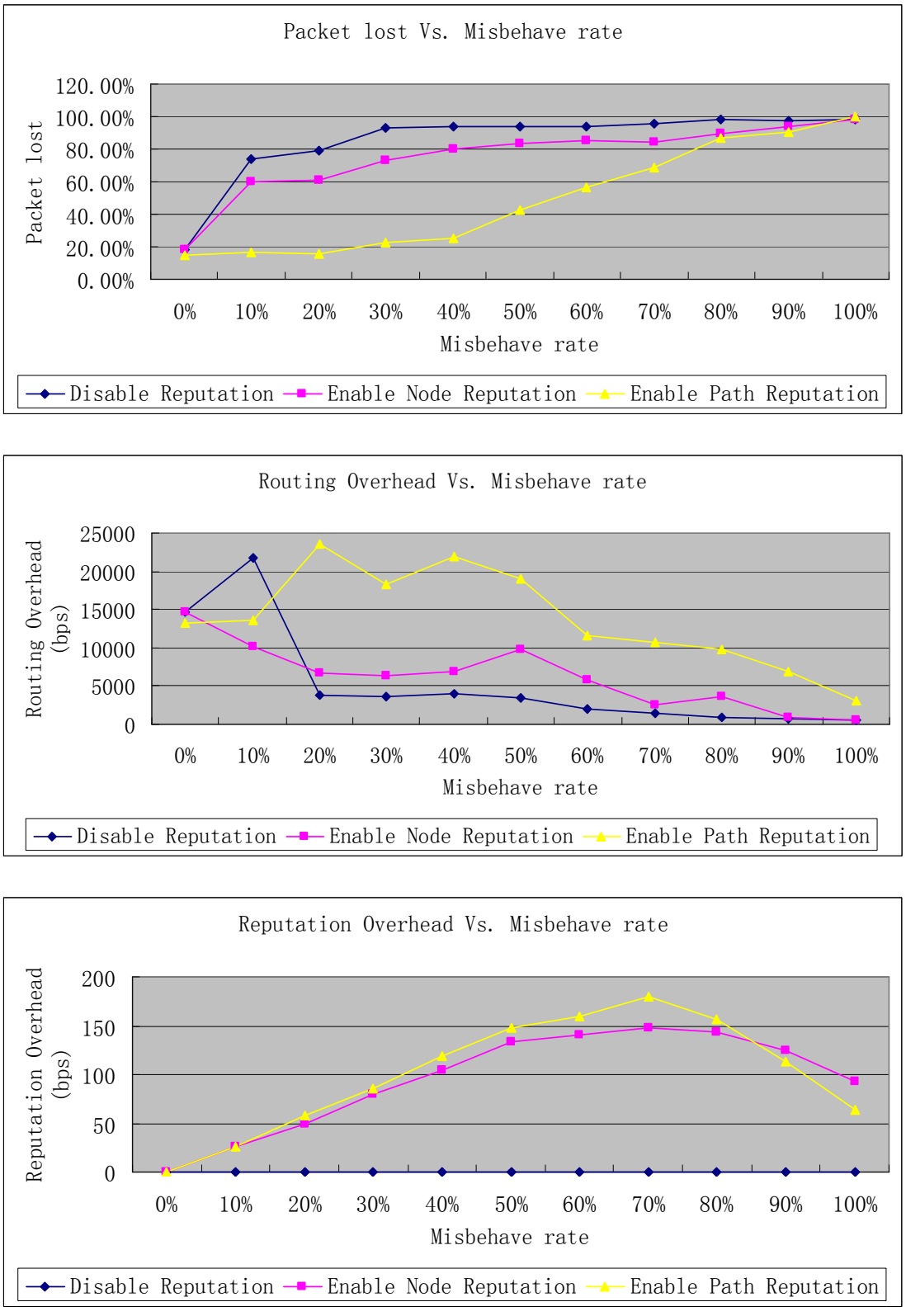
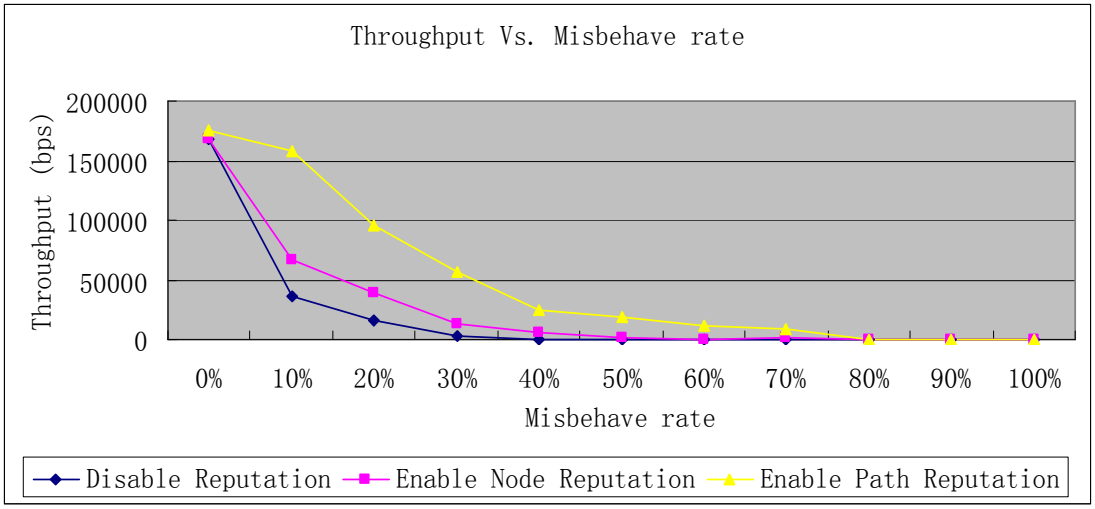
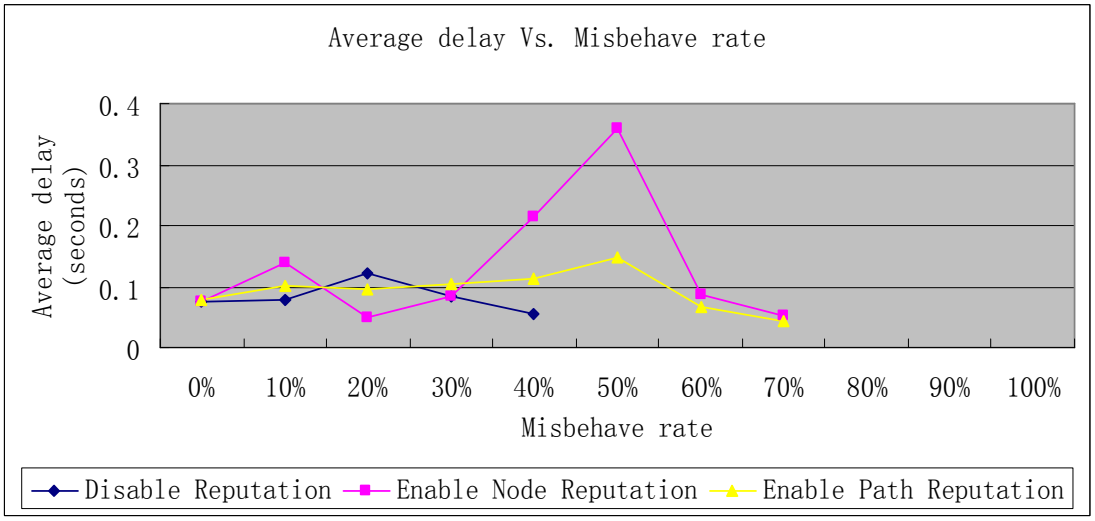


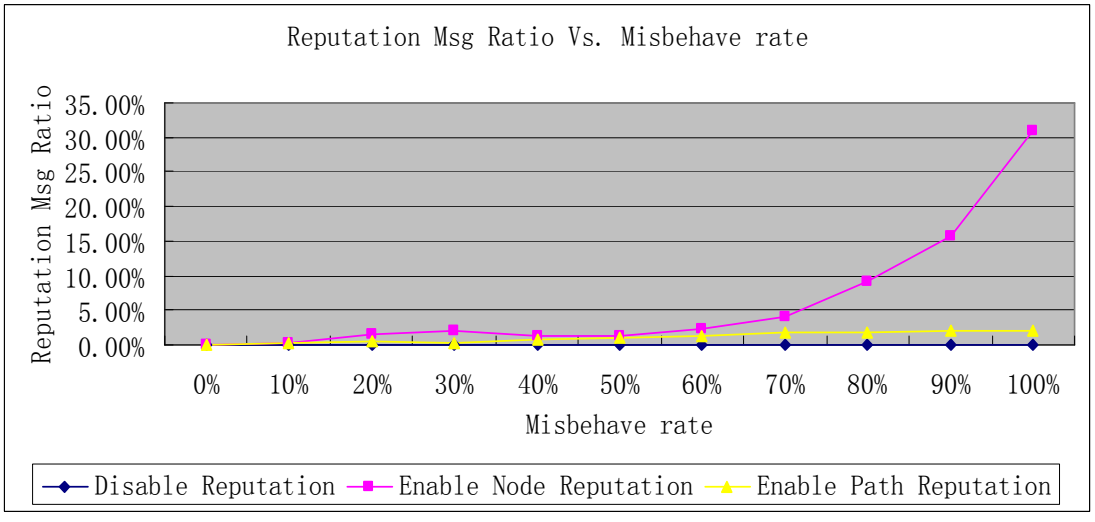
Figure 7. Performance in Forge Reply (UDP).

(B) Misbehavior: Forge Reply plus Data Selfish

As shown in Figure 8, throughput is improved significantly when path reputation is used (more than double comparing with node reputation), while the message overhead increase by less than 2% (delay is omitted since it is similar to the above). This again proves that path-based reputation scheme is exceptionally effectively in dealing with path misbehaviors.







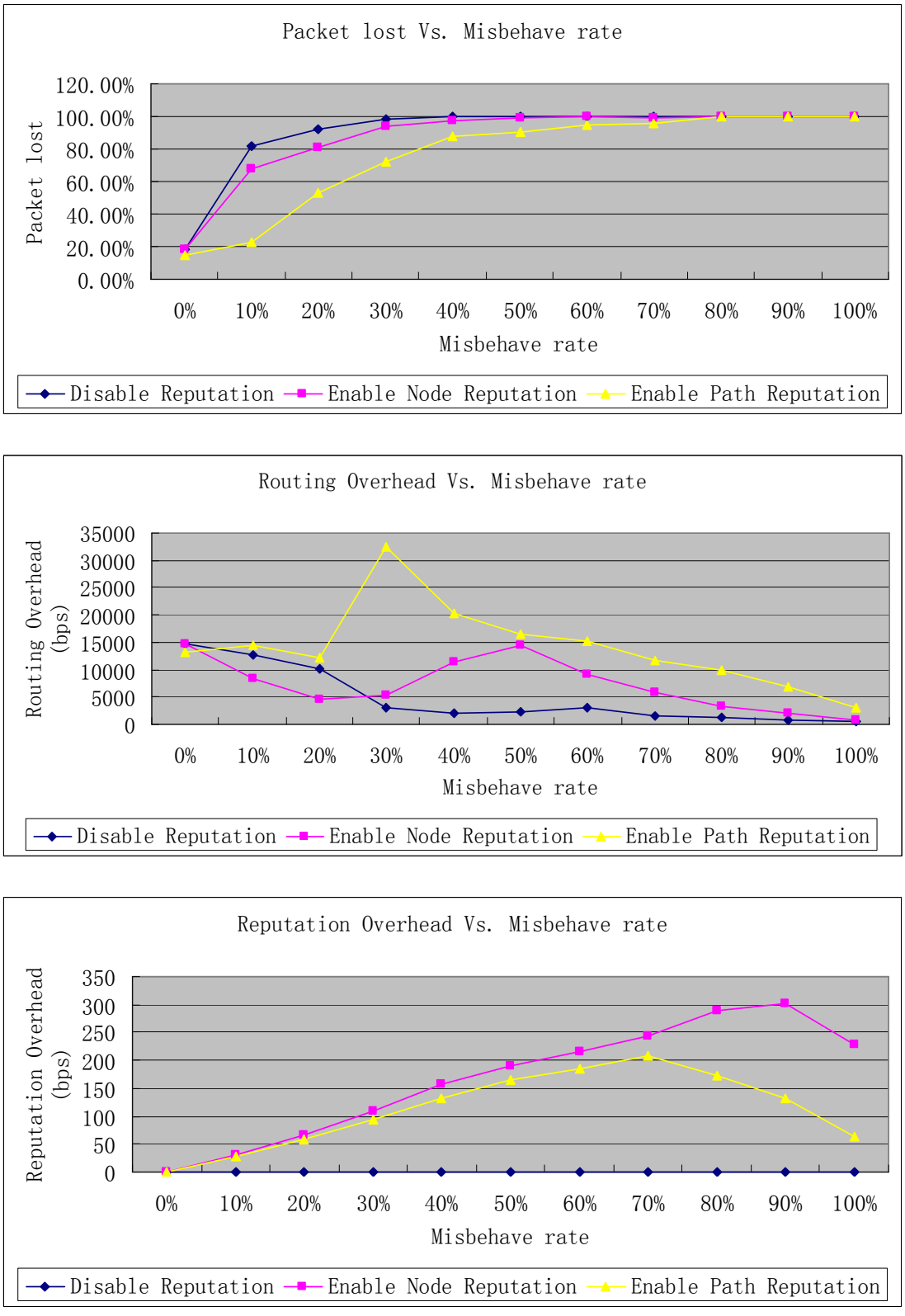
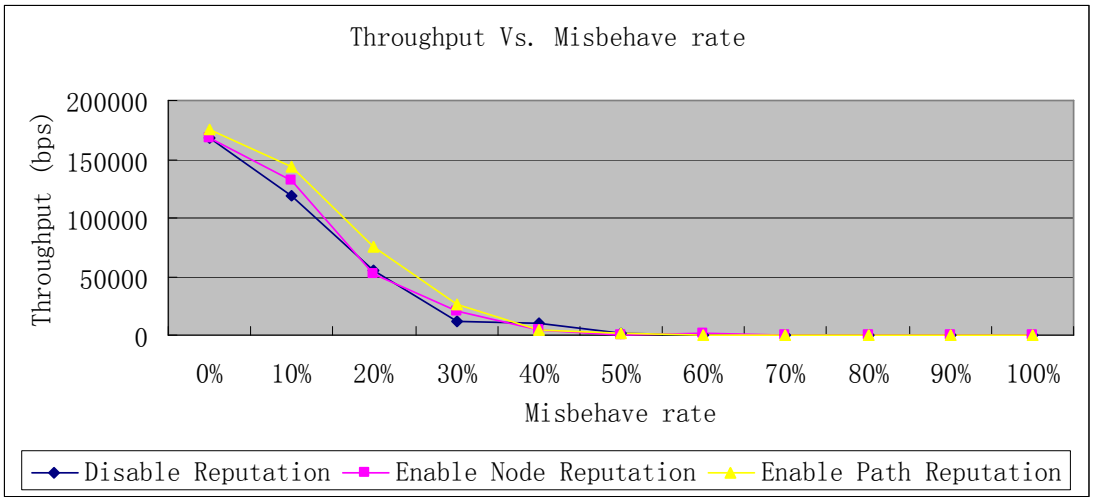


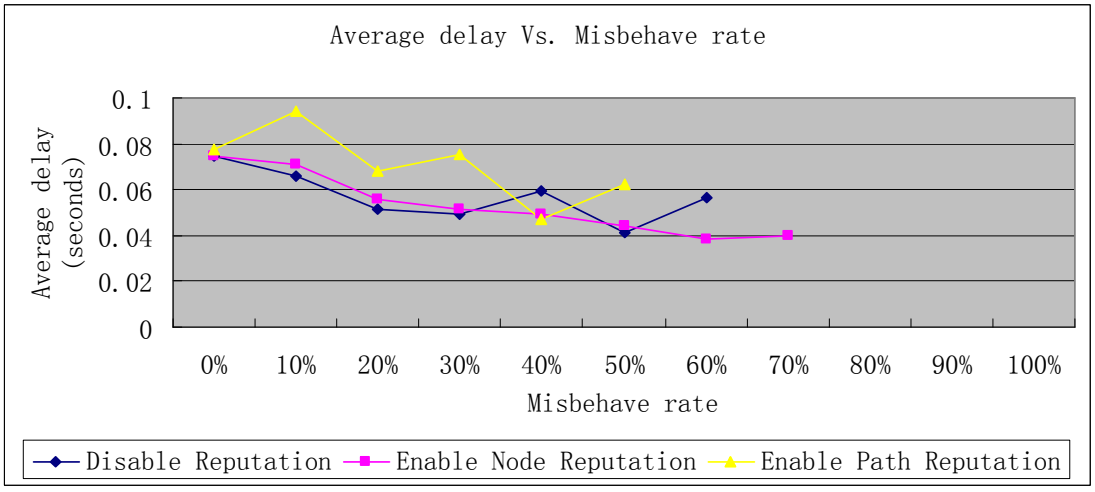
Figure 8. Performance in Forge Reply plus Data Selfish (UDP)

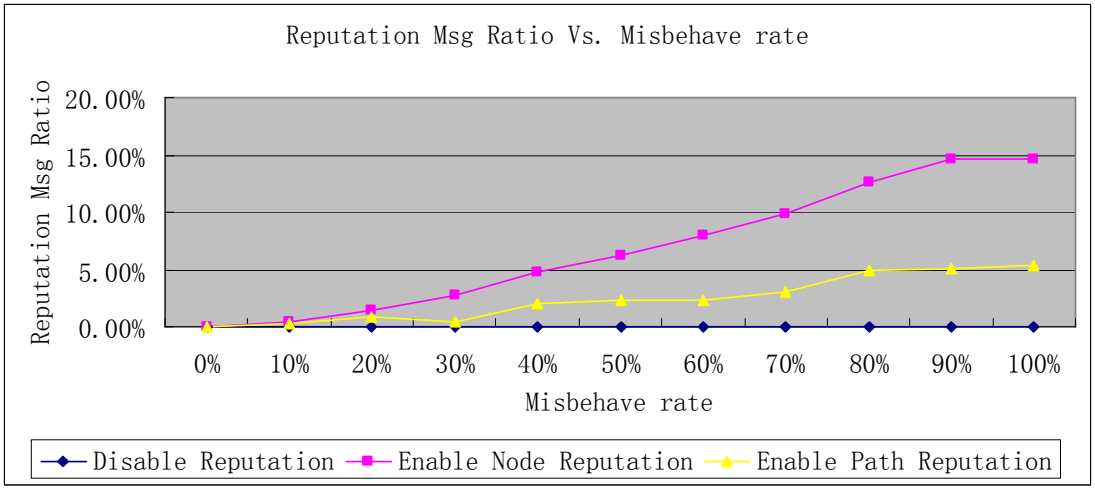
(C) Misbehavior: Data Selfish

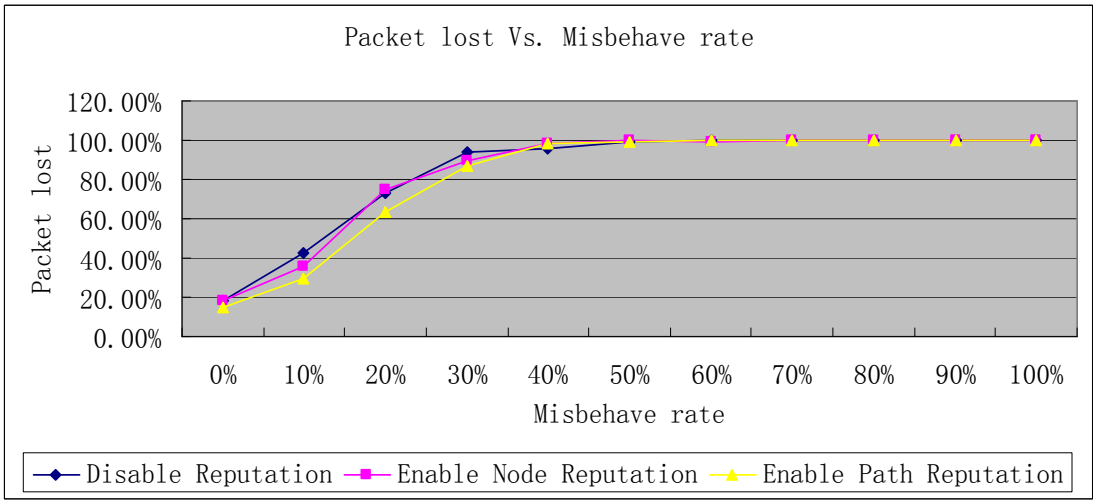
As shown in Figure 9, path-based reputation has improved throughput by 30% or more than the

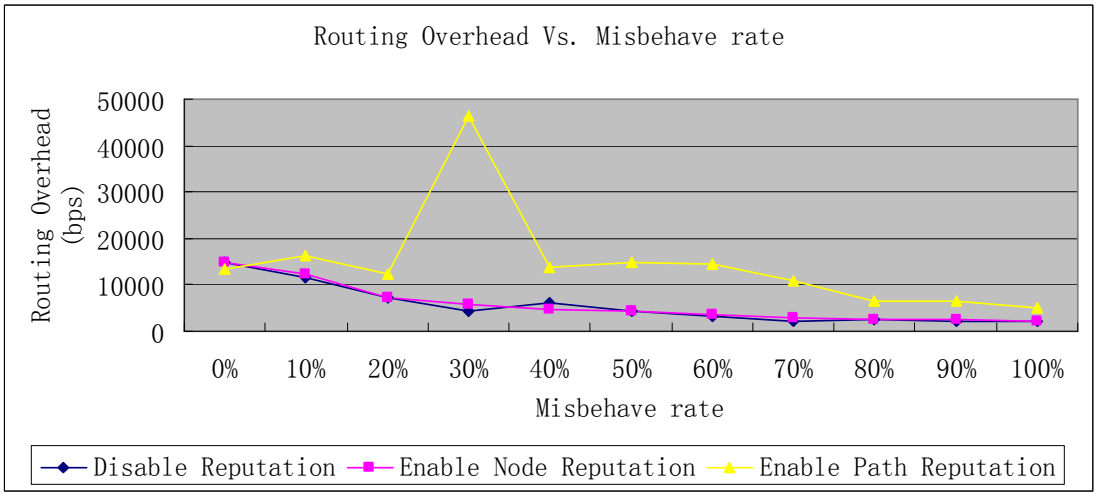
original AODV when misbehave nodes percentage is under 40%, but the overhead is only increased for under 3%. (Delay results are omitted from now on since they are similar to those in Figure 3.) Thus, it is also quite effective in handling data selfish. Comparing with node-based reputation, its advantage is not as evident. We believe it is because this misbehavior is a "local behavior" that can be successfully detected by a node-based reputation system, whereas the forge reply attack affects the entire route, in which case the proposed path-based system is exceedingly useful.






Reputation Msg Ratio Vs. Misbehave rate
Reputation Msg Ratio
20.00%
15.00%
10.00%
5.00%
0.00%
0%
10%
20%
30%
40%
50%
60%
70%
80%
90%
100%
Misbehave rate
Disable Reputation
Enable Node Reputation
Enable Path Reputation


Packet lost Vs. Misbehave rate
Packet lost
120.00%
100.00%
80.00%
60.00%
40.00%
20.00%
0.00%
0%
10%
20%
30%
40%
50%
60%
70%
80%
90%
100%
Misbehave rate
Disable Reputation
Enable Node Reputation
Enable Path Reputation


Routing Overhead Vs. Misbehave rate
Routing Overhead (bps)
50000
40000
30000
20000
10000
0
0%
10%
20%
30%
40%
50%
60%
70%
80%
90%
100%
Misbehave rate
Disable Reputation
Enable Node Reputation
Enable Path Reputation

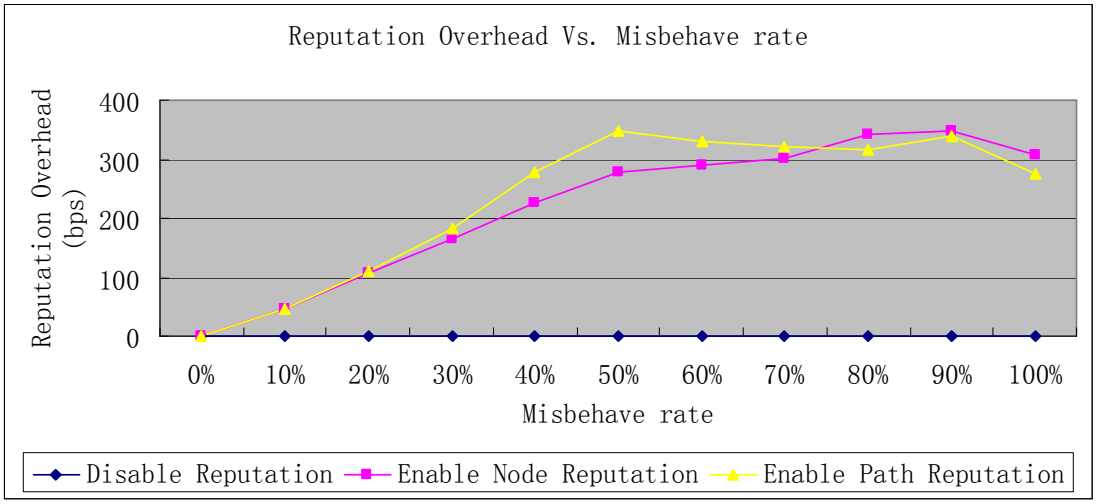
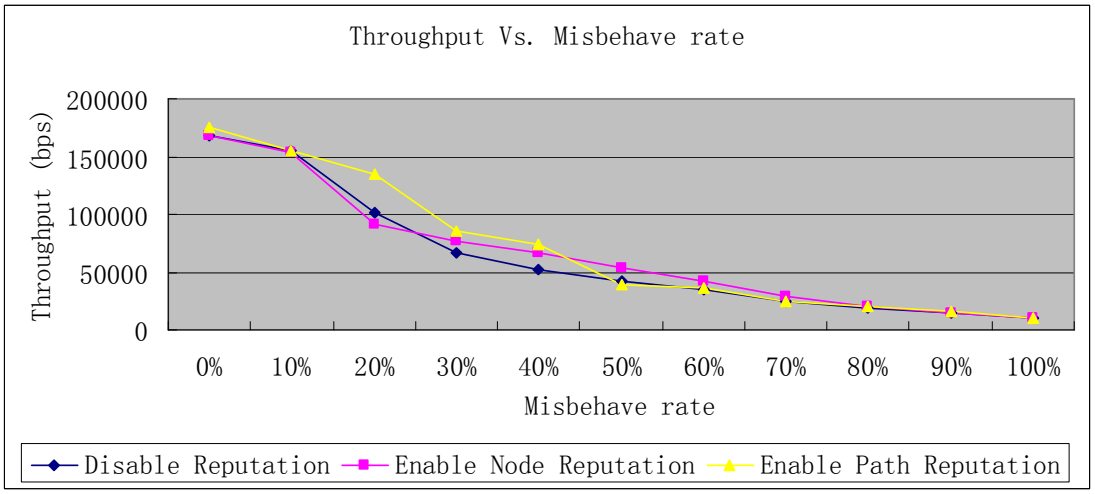
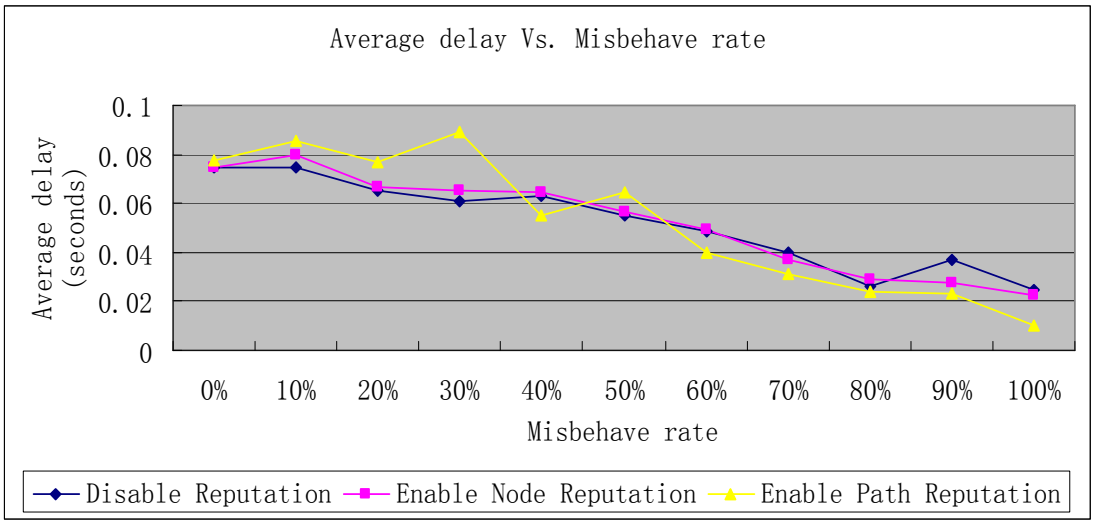


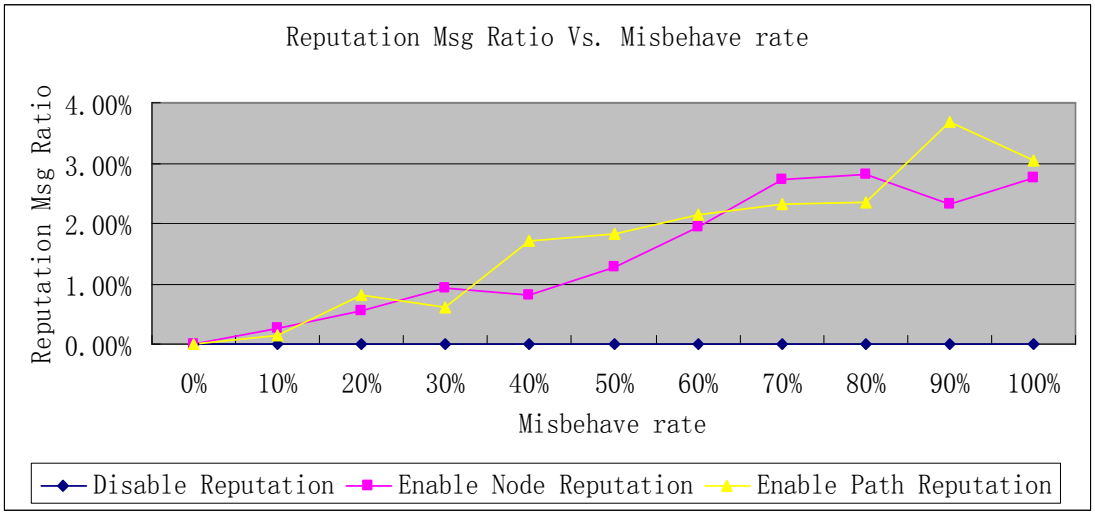
Figure 9. Performance in Data Selfish (UDP)

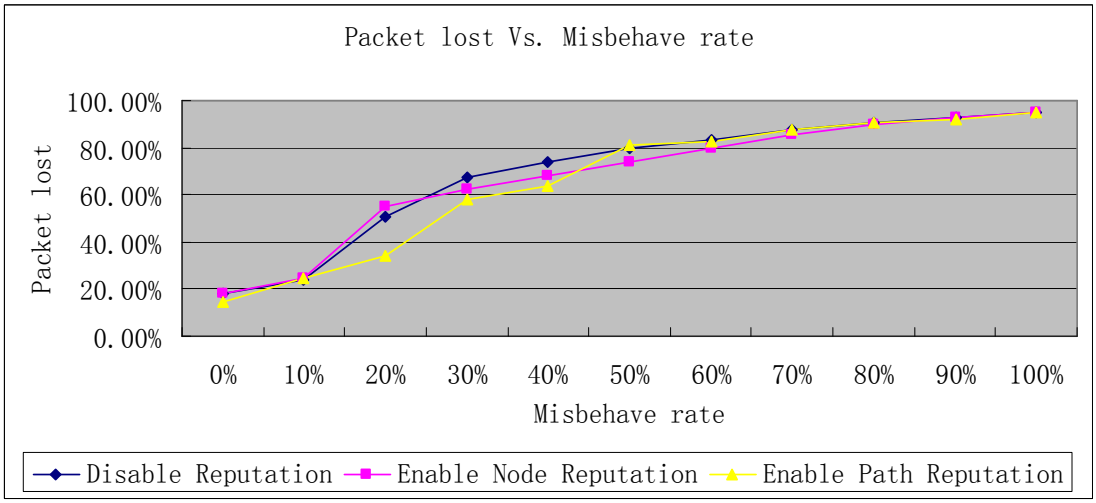
(D) Misbehavior: Forge Data

In this experiment, we did not see great improvement in throughput – about 20-30% when misbehave rate is 20-40% and message overhead remains 4%. These results also reflect that the proposed system is not effective in dealing with "local" misbehaviors.




Average delay Vs. Misbehave rate
Average delay (seconds)
0.1
0.08
0.06
0.04
0.02
0
0%
10%
20%
30%
40%
50%
60%
70%
80%
90%
100%
Misbehave rate
Disable Reputation
Enable Node Reputation
Enable Path Reputation


Reputation Msg Ratio Vs. Misbehave rate
Reputation Msg Ratio
4.00%
3.00%
2.00%
1.00%
0.00%
0%
10%
20%
30%
40%
50%
60%
70%
80%
90%
100%
Misbehave rate
Disable Reputation
Enable Node Reputation
Enable Path Reputation


Packet lost Vs. Misbehave rate
Packet lost
100.00%
80.00%
60.00%
40.00%
20.00%
0.00%
0%
10%
20%
30%
40%
50%
60%
70%
80%
90%
100%
Misbehave rate
Disable Reputation
Enable Node Reputation
Enable Path Reputation

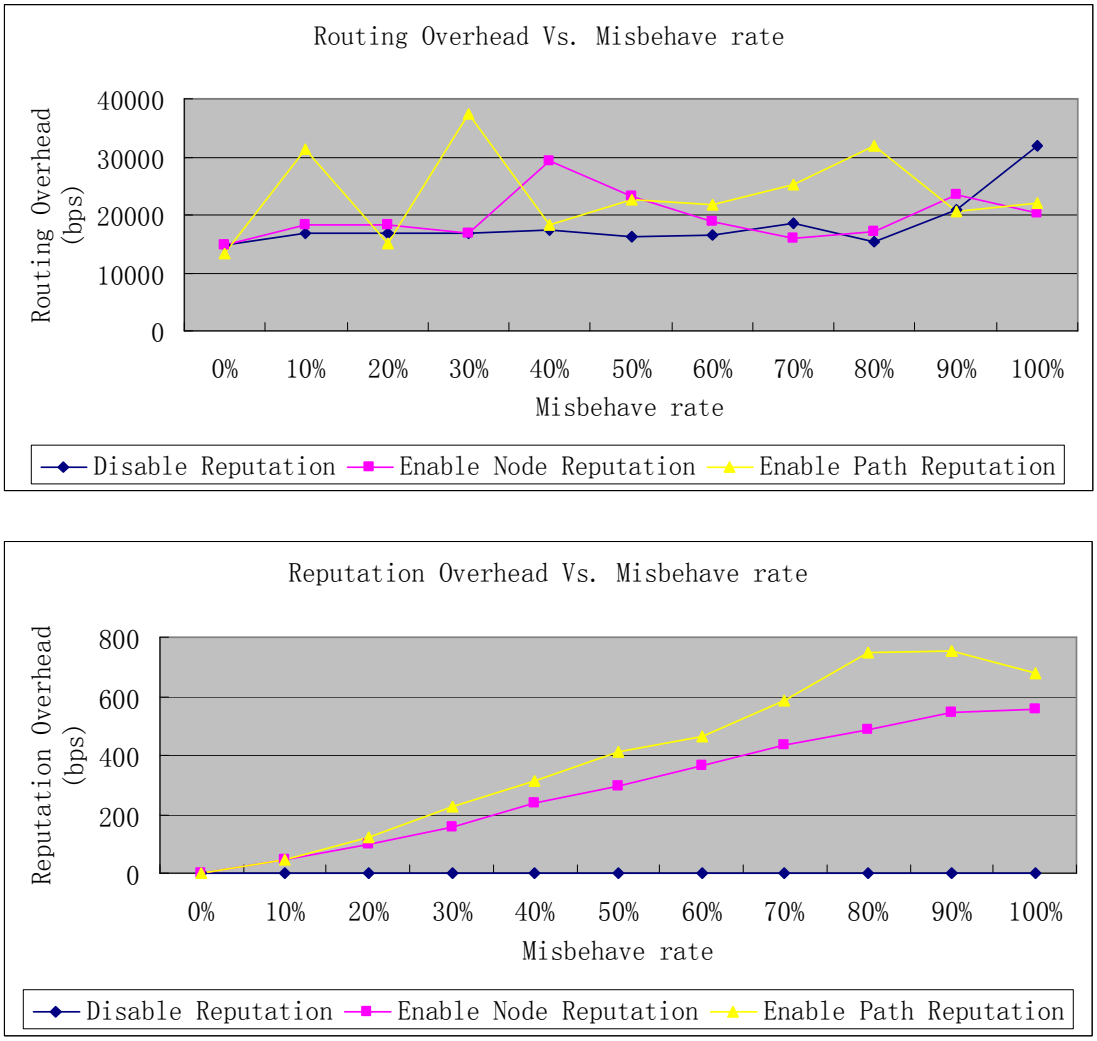


Figure 10. Performance in Forge Data (UDP)

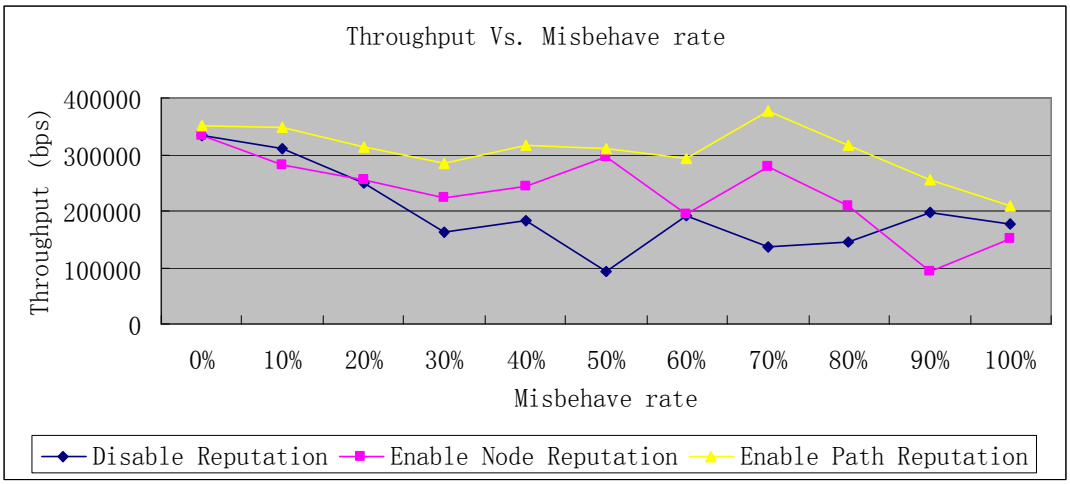
4.4. Simulation Results: FTP over TCP

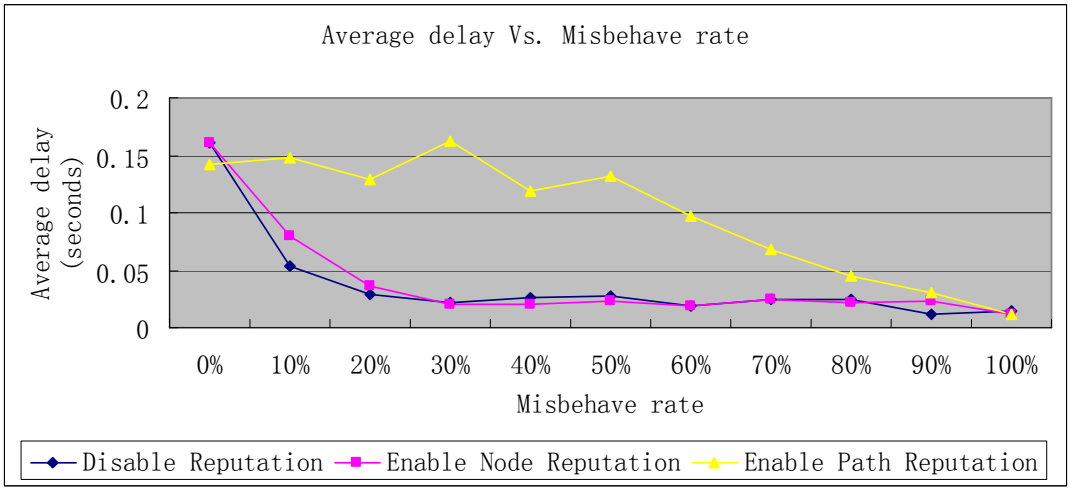
In this set of experiments, TCP traffic is applied with message size of 512 bytes continuously sent from the ten randomly chosen sources to their corresponding receivers. The results are similar to those in UDP.

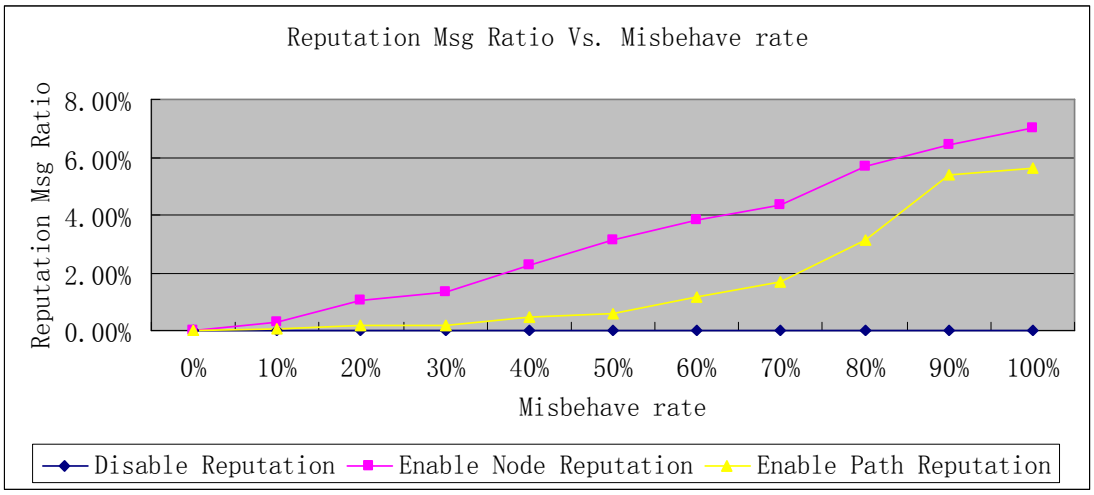
We got the similar results on all misbehaviors as the CBR: the path reputation system improved the throughput when there are nodes misbehaved in the network, and the system is more efficient for the routing misbehaviors such as forge reply than the data forwarding misbehaviors. The results also show that the proposed reputation system is less efficient for TCP traffic than UDP. We believe that this is mainly because the congestion control in TCP effectively limits its sending rate (and thus throughput) when misbehaviors take place. As the result, reputation systems are not as effective.

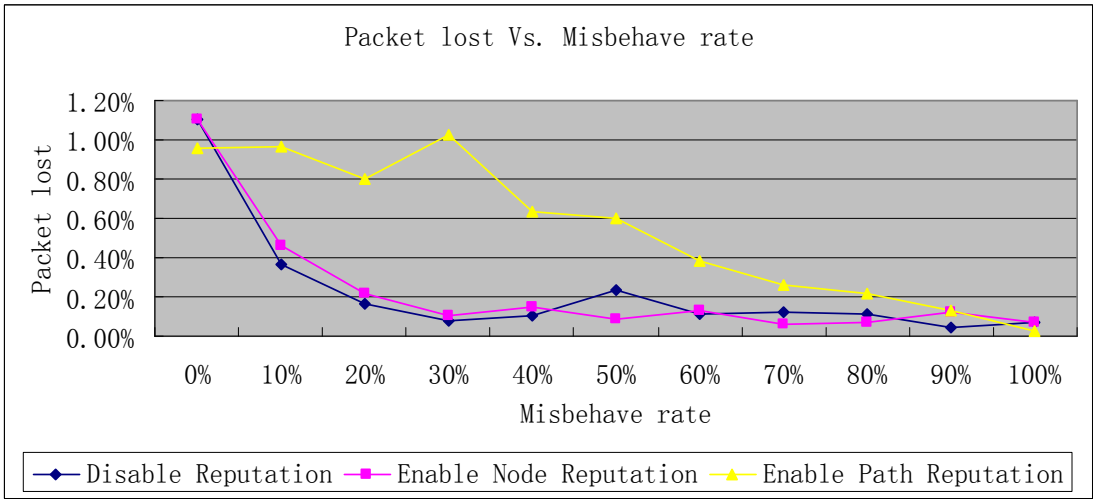
(A) Misbehavior: Forge Reply

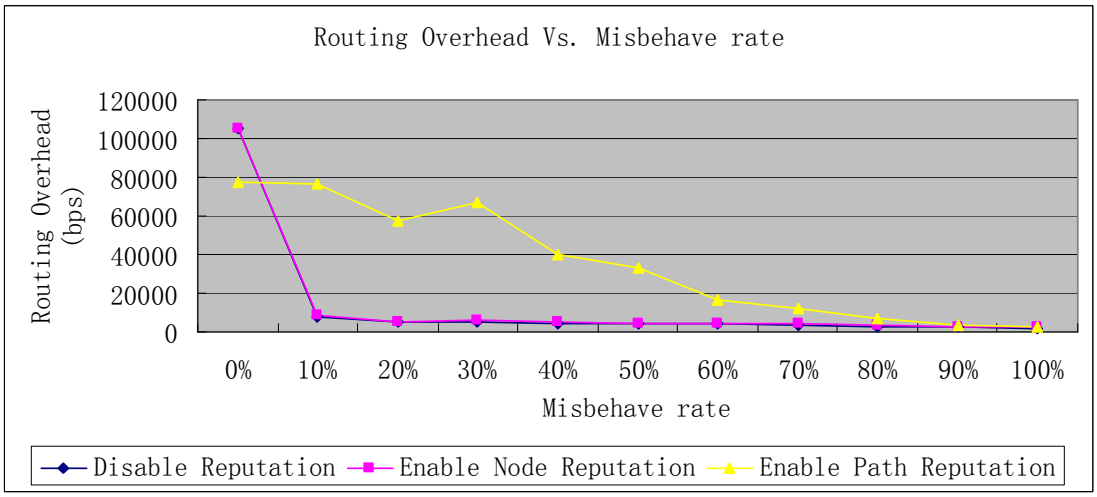
Simulation results shown in Figure 11 indicate that the path-based reputation system is also effective in dealing with forge reply attack in TCP, though not as effective as it does in UDP. It has about 50% increase in throughput comparing with node-based reputation, and more than doubled comparing with the original AODV; and it keeps a much lower packet lost rate comparing with the original AODV. On the other hand, different from the slightly increase in the UDP, delay has been increased more obviously in TCP, we believe that is because the TCP congestion control kicked in as there are much more packets transmitted through the network. In the mean time, its reputation message overhead has stayed at a low level.






Reputation Msg Ratio Vs. Misbehave rate
Reputation Msg Ratio
8.00%
6.00%
4.00%
2.00%
0.00%
0%
10%
20%
30%
40%
50%
60%
70%
80%
90%
100%
Misbehave rate
Disable Reputation
Enable Node Reputation
Enable Path Reputation


Packet lost Vs. Misbehave rate
Packet lost
1.20%
1.00%
0.80%
0.60%
0.40%
0.20%
0.00%
0%
10%
20%
30%
40%
50%
60%
70%
80%
90%
100%
Misbehave rate
Disable Reputation
Enable Node Reputation
Enable Path Reputation


Routing Overhead Vs. Misbehave rate
Routing Overhead (bps)
120000
100000
80000
60000
40000
20000
0
0%
10%
20%
30%
40%
50%
60%
70%
80%
90%
100%
Misbehave rate
Disable Reputation
Enable Node Reputation
Enable Path Reputation

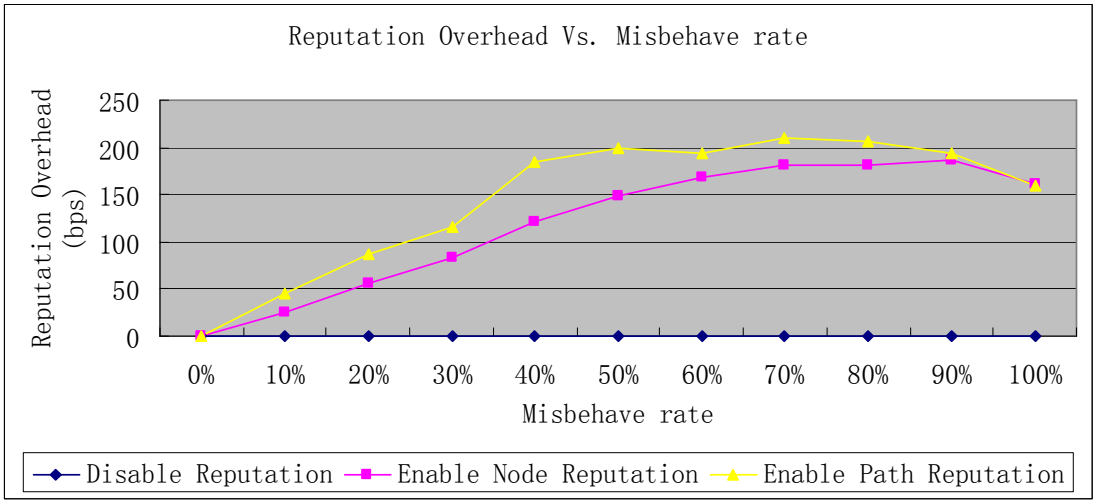
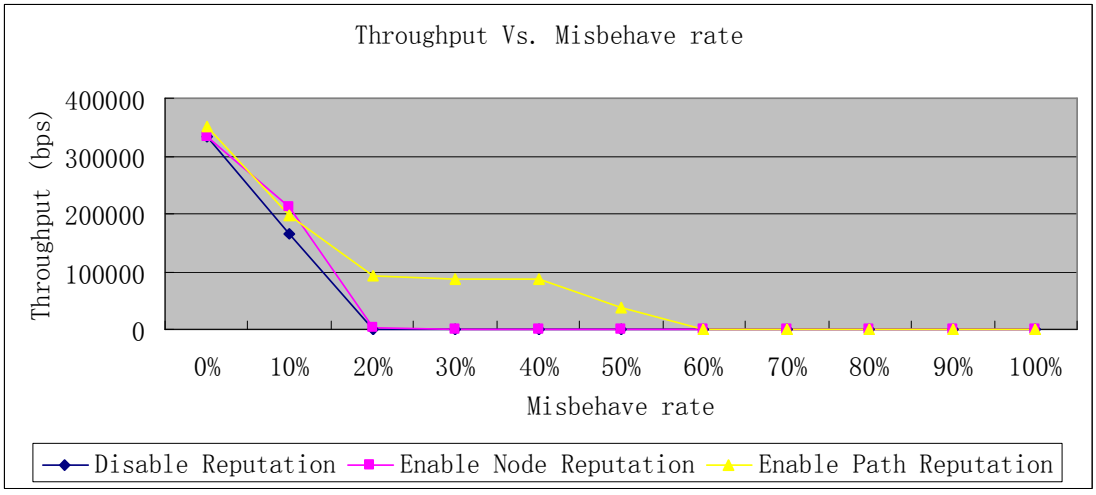
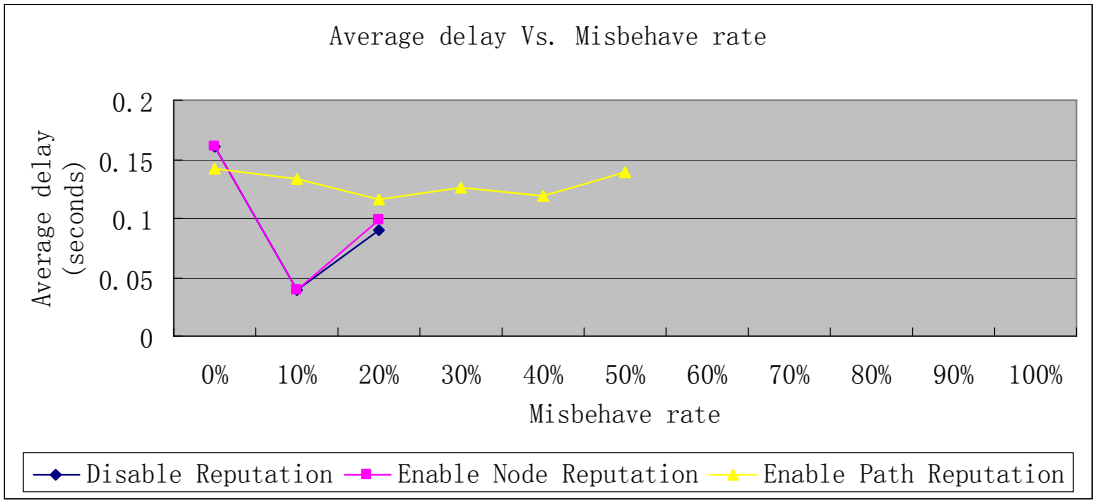


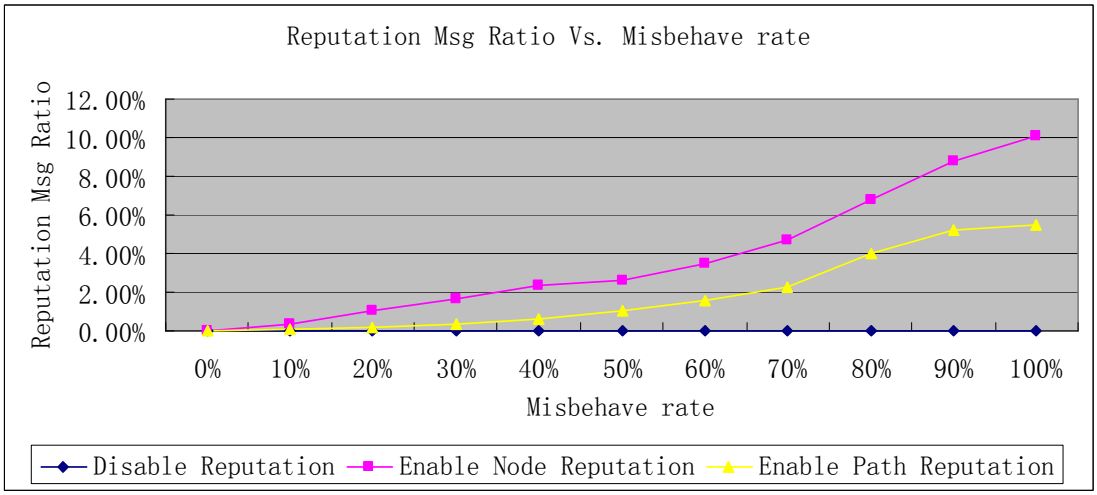
Figure 11. Performance in Forge Reply (TCP).

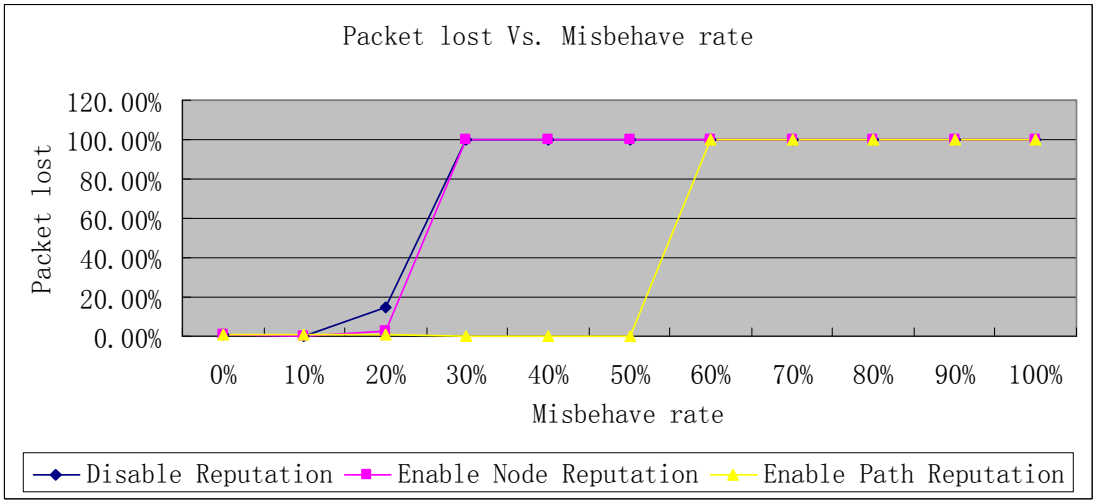
(B) Misbehavior: Forge Reply plus Data Selfish

As shown in Figure 12, throughput is dramatically decreased to 0 for the original AODV and node-based reputation when there are more than 20% node misbehaved in the network, we believe that is because the TCP is a connection oriented protocol and the pair stops transmitted data once the connection is lost and there is no mechanism build in out simulation to recover a connection after it fails. Comparing to the original AODV and node-based reputation, throughput is improved significantly when path reputation is used (keeps a reasonable throughput until the misbehave nodes rate increases to 60%), although the message overhead does not increase much.




Average delay Vs. Misbehave rate
Average delay (seconds)
0.2
0.15
0.1
0.05
0
0%
10%
20%
30%
40%
50%
60%
70%
80%
90%
100%
Misbehave rate
Disable Reputation
Enable Node Reputation
Enable Path Reputation


Reputation Msg Ratio Vs. Misbehave rate
Reputation Msg Ratio
12.00%
10.00%
8.00%
6.00%
4.00%
2.00%
0.00%
0%
10%
20%
30%
40%
50%
60%
70%
80%
90%
100%
Misbehave rate
Disable Reputation
Enable Node Reputation
Enable Path Reputation


Packet lost Vs. Misbehave rate
Packet lost
120.00%
100.00%
80.00%
60.00%
40.00%
20.00%
0.00%
0%
10%
20%
30%
40%
50%
60%
70%
80%
90%
100%
Misbehave rate
Disable Reputation
Enable Node Reputation
Enable Path Reputation

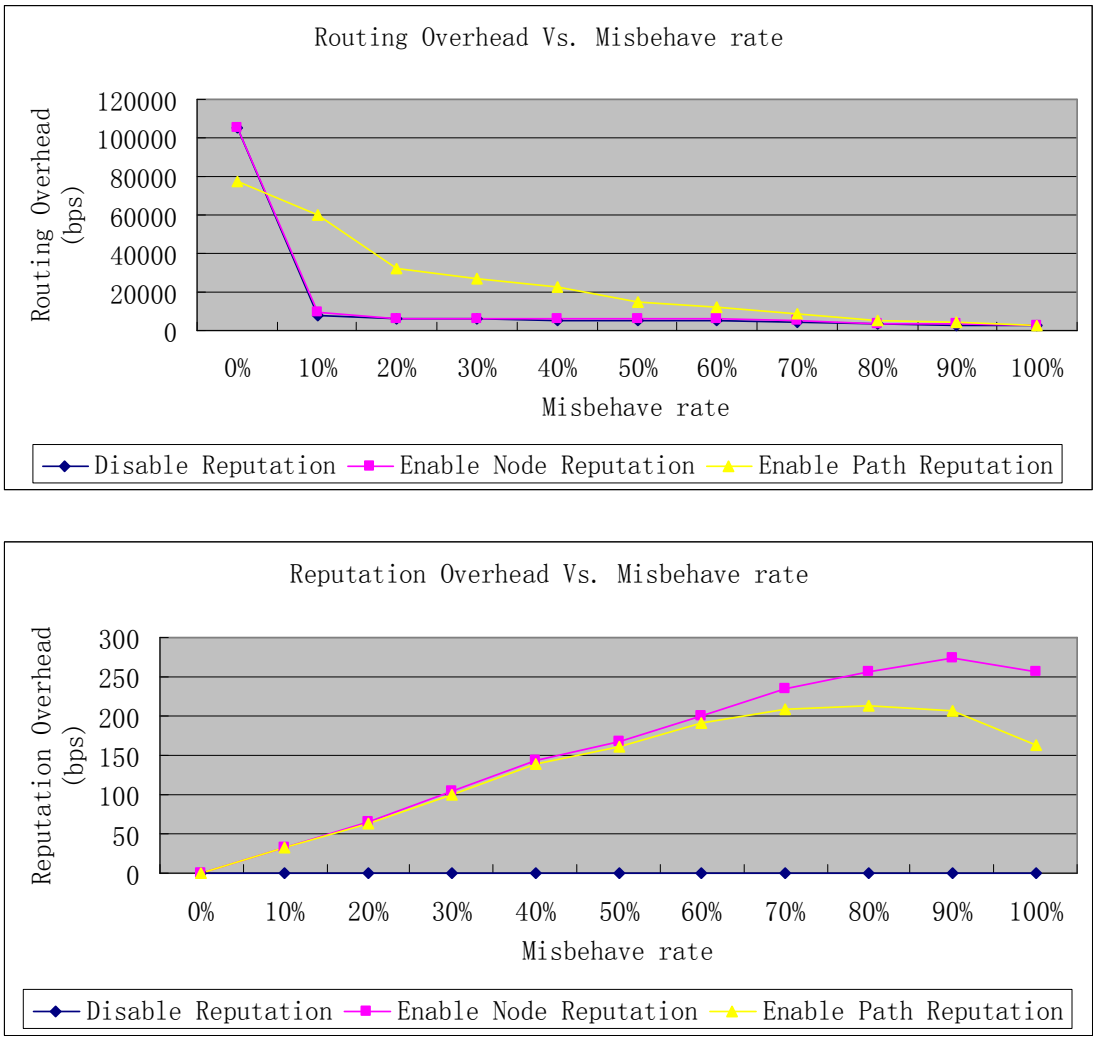
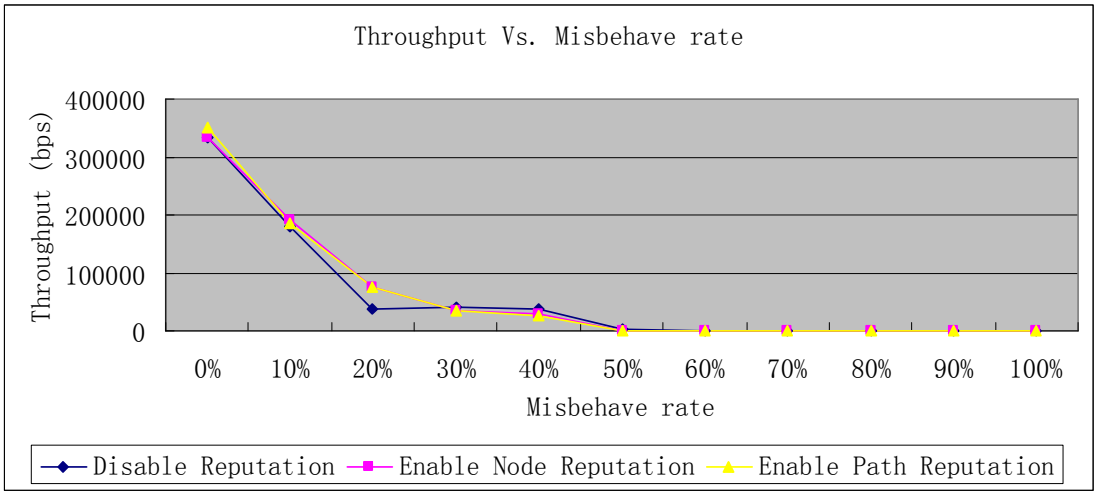
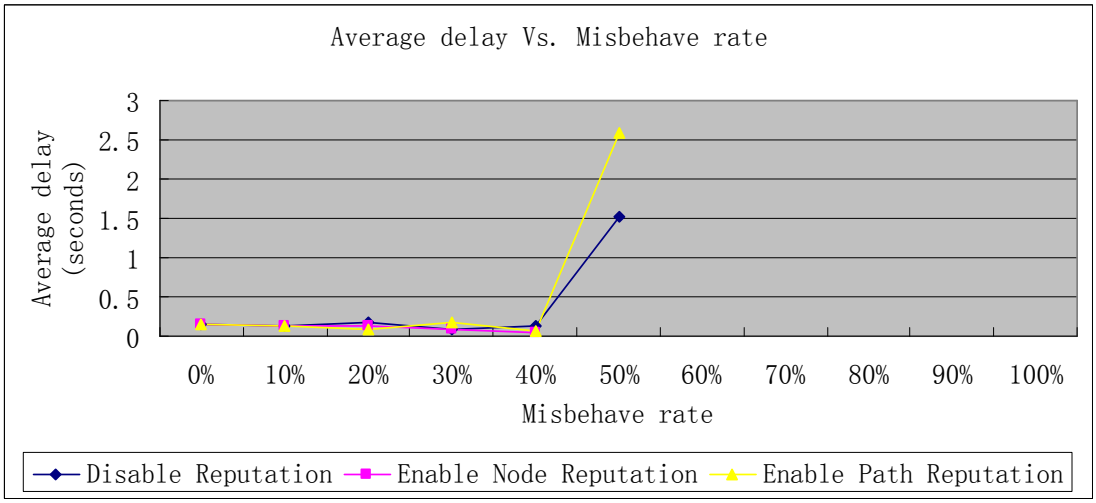


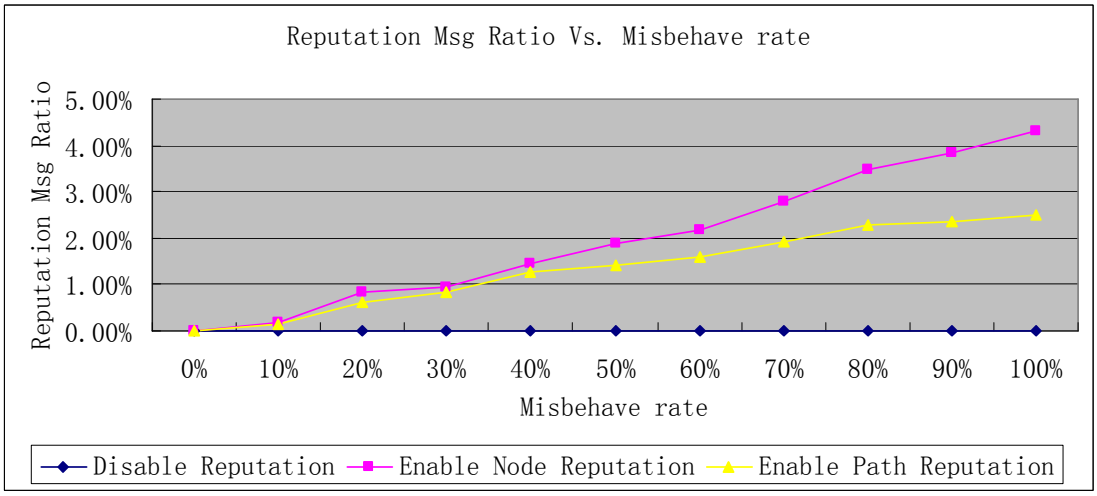
Figure 12. Performance in Forge Reply plus Data Selfish (TCP)

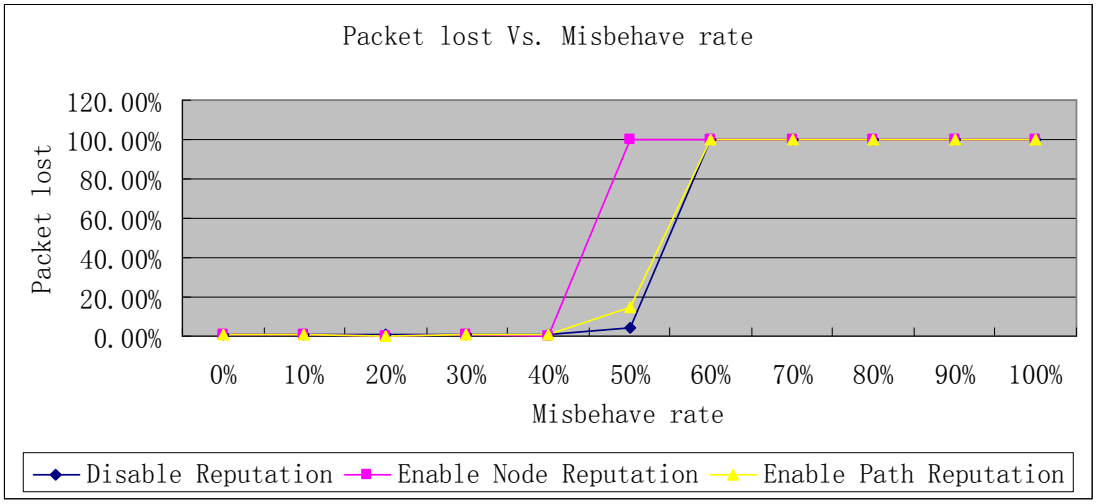
(C) Misbehavior: Data Selfish

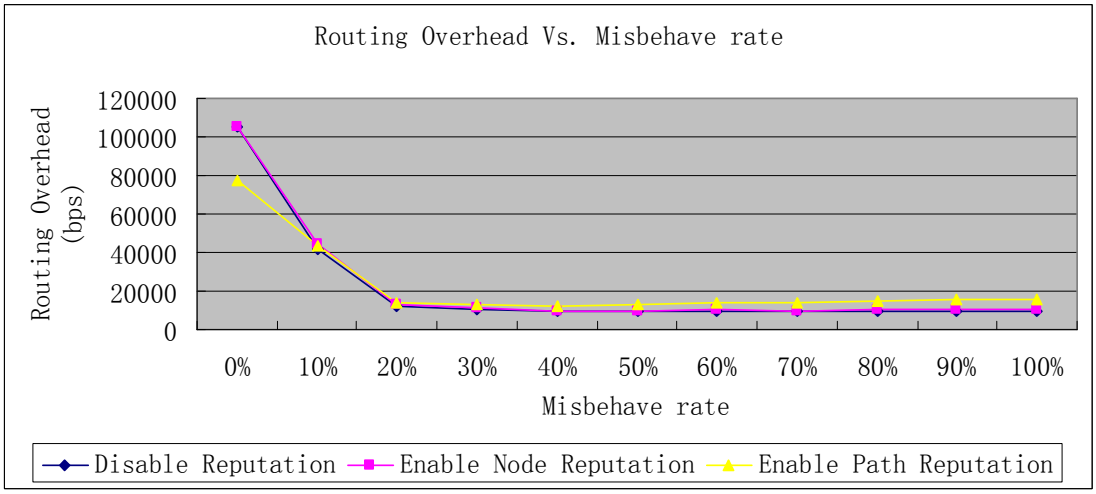
As shown in Figure 13, throughput is only limited improved for path-based reputation comparing with the original AODV, about 5%-40% when misbehave nodes percentage is under 30%. Its message overhead keeps the same or even lower. Thus, it is still effective in handling data selfish in TCP, but not as good as in UDP. Its advantage comparing with node-based reputation is not that evident. We believe it is because the same reason that has been discussed in UDP results - data selfish is a local misbehavior and the proposed system is not good at dealing with local misbehaviors.


Throughput Vs. Misbehave rate
Throughput (bps)
400000
300000
200000
100000
0
0%
10%
20%
30%
40%
50%
60%
70%
80%
90%
100%
Misbehave rate
Disable Reputation
Enable Node Reputation
Enable Path Reputation


Average delay Vs. Misbehave rate
Average delay (seconds)
3
2.5
2
1.5
1
0.5
0
0%
10%
20%
30%
40%
50%
60%
70%
80%
90%
100%
Misbehave rate
Disable Reputation
Enable Node Reputation
Enable Path Reputation


Reputation Msg Ratio Vs. Misbehave rate
Reputation Msg Ratio
5.00%
4.00%
3.00%
2.00%
1.00%
0.00%
0%
10%
20%
30%
40%
50%
60%
70%
80%
90%
100%
Misbehave rate
Disable Reputation
Enable Node Reputation
Enable Path Reputation





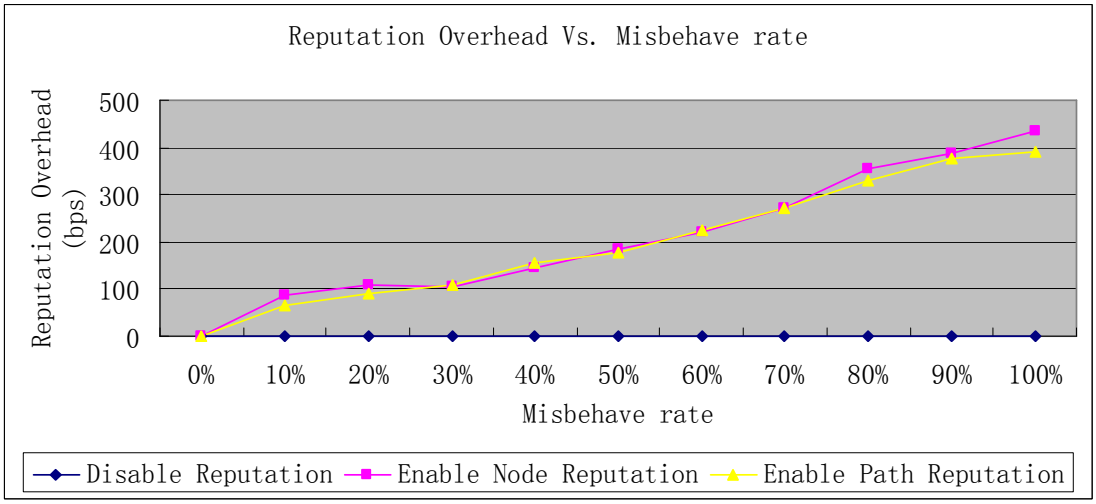
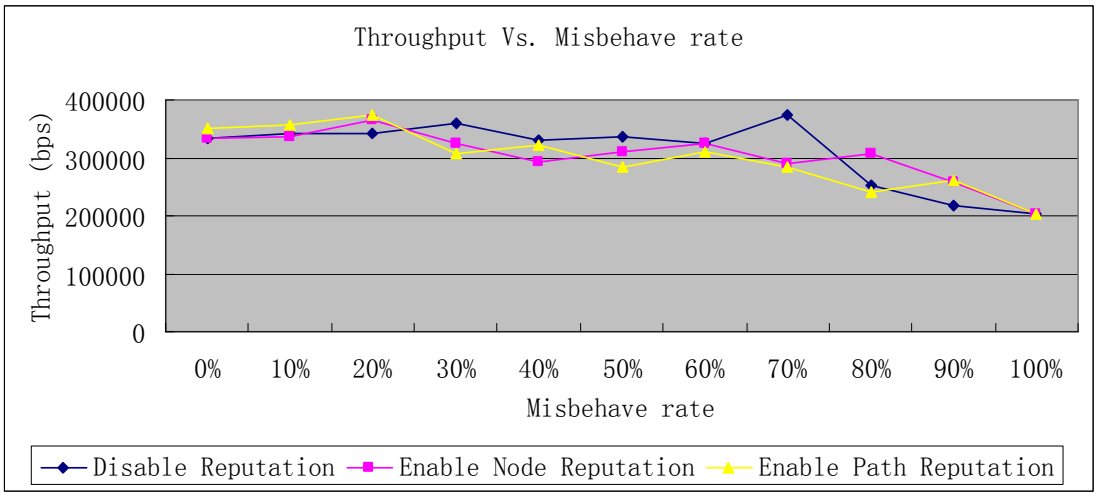


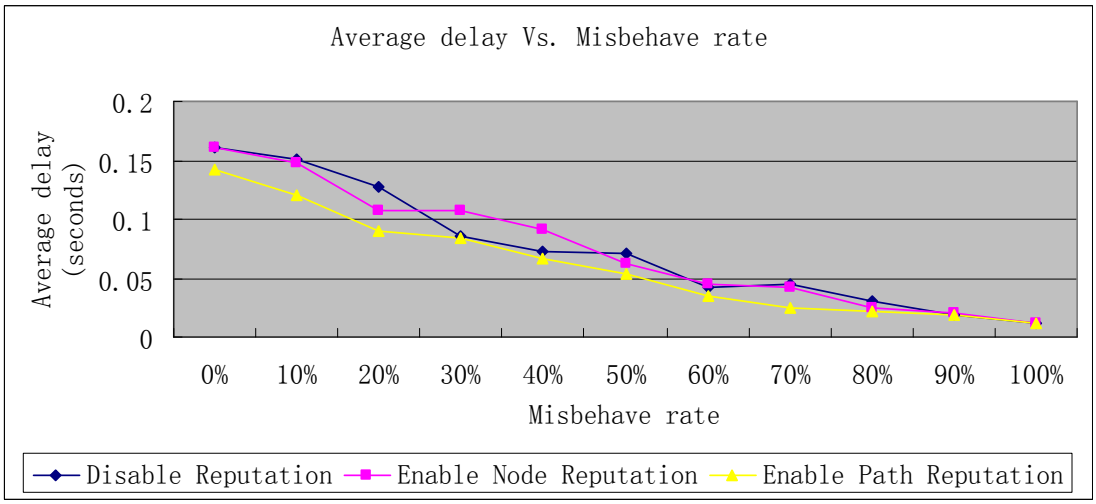
Figure 13. Performance in Data Selfish (TCP)

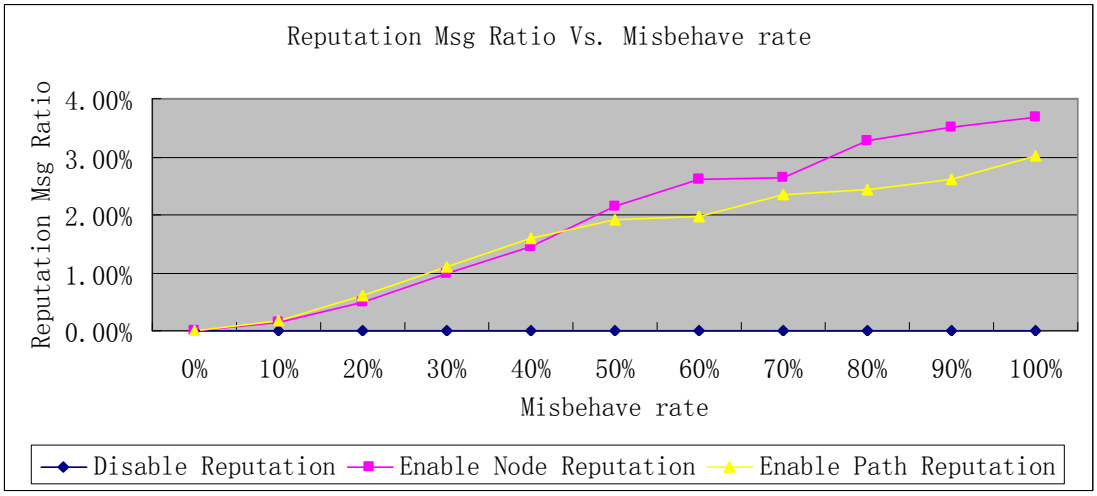
(D) Misbehavior: Forge Data

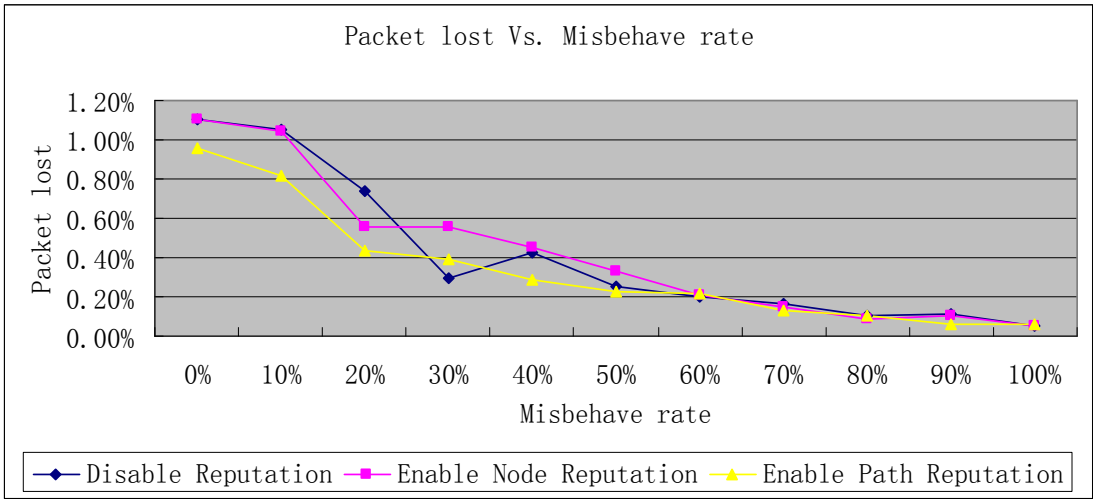
As shown in Figure 14, there is no obvious improvement in throughput – about 5-10% when

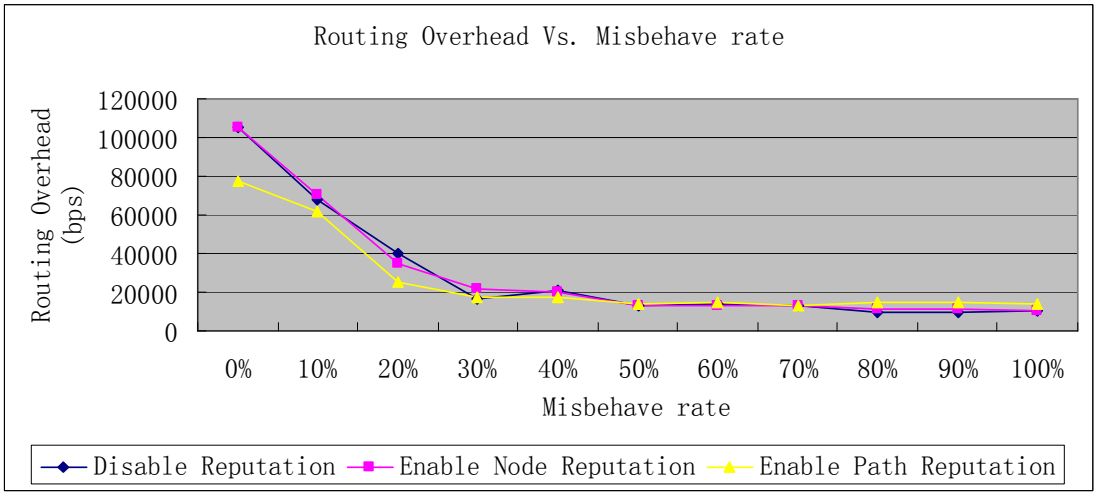
misbehave rate is under 20%; message overhead also stays very low. Again the results help one to see that the proposed system is not effective in dealing with "local" misbehaviors.











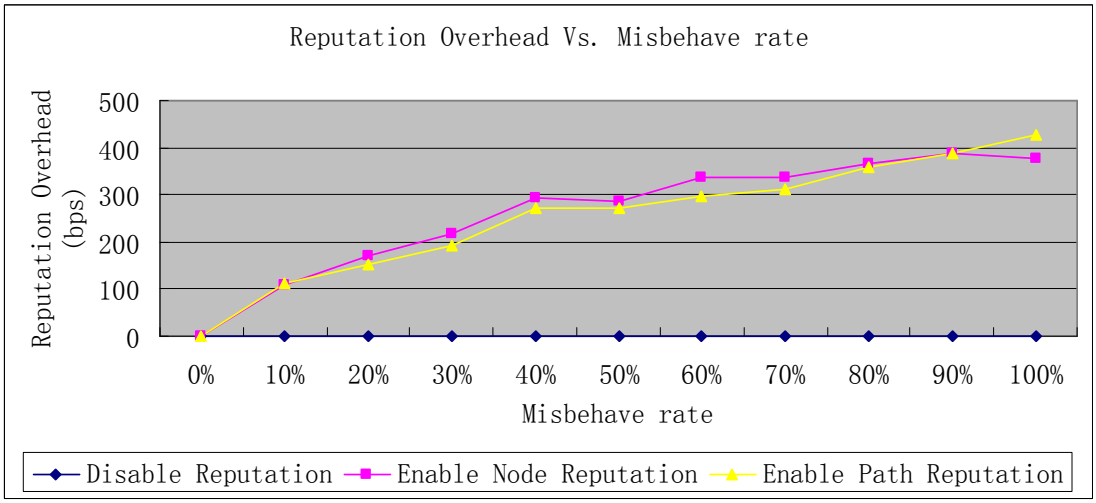


Figure 14. Performance in Forge Data (TCP)

5. Conclusion

Existing reputation systems in MANET routing only evaluate one hop distance neighbors' reputation when they select the route. Since no node in the route has a complete view of the entire route, it is hard to select the best route when the path contains multiple hops. A path-based reputation system that considers path reputation as a function of reputation and trust of every node in the path has been proposed to solve this issue. Since the source node has a complete view of the entire path, it is much easier to select the best route based on the knowledge. This system utilizes a category (or range) approach when evaluating trust of second-hand information. It has been illustrated on top of AODV. Simulation results show that the proposed system is most effective when handling routing misbehaviors such as forge reply (or worm-hole) attacks; in most cases it doubles or even triples throughput comparing with a node-based reputation scheme and with the original AODV. Future work would include optimizing and analyzing the various parameters for a best performance. Further, the application of fuzzy logic in the trust category approach that we proposed could be explored for the practical use to fuzz the border between categories.

6. References

1. Akyildiz, I. F. and X. Wang (2005). "A Survey on Wireless Mesh Networks." *IEEE Commnunications Magazine* 43(9): 23-30.
2. Balakrishnan, V., V. Varadharajan, et al. (2007). "Trust Enhanced Secure Mobile Ad-Hoc Network Routing." *Proceedings of 21st International Conference on Advanced Information Networking and Applications Workshops 2*: 21-33.
3. Bansal, S. and M. Baker (2003). "Observation-based Cooperation Enforcement in Ad Hoc Networks." *Technical Report*, Standford University.
4. Buchegger, S. and J. Le Boudec (2002). "Performance analysis of the CONFIDANT protocol." *Proceedings of the 3rd ACM International Symposium on Mobile Ad Hoc Networking and Computing*, Lausanne, Switzerland.
5. Buchegger, S. and J. Le Boudec (2003). "A Robust Reputation System for Mobile Ad-hoc Networks." *Technique Report.*
6. Griffiths, N., C. Kuo-Ming, et al. (2006). "Fuzzy Trust for Peer-to-Peer Systems." *Proceedings of Distributed Computing Systems Workshops*: 73.
7. Hu, J. and M. Burmester (2006). "LARS: a locally aware reputation system for mobile ad hoc networks." *Proceedings of the 44th Southeast regional conference*.
8. Hubaux, J. P., L. Batttan, et al. (2001). "The quest for security in mobile ad hoc networks." *Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking & computing*: 146-155.
9. Liu, J., Z. Li, et al. (2005). "A security enhanced AODV routing protocol based on the credence mechanism." *Wireless Communications, Networking and Mobile Computing, 2005. Proceedings. 2005 International Conference on*.

10. Marti, S., T. Giuli, et al. (2000). "Mitigating Routing Misbehaviors in mobile Ad Hoc networks." *Proceedings of the 6th annual international conference on Mobile computing and networking*.
11. Michiardi, P. and R. Molva (2002). "CORE: A COllaborative REptation mechanism to enforce node cooperation in Mobile Ad Hoc Networks." *The IFIP TC6/TC11 6th Joint Working Conference on Communications and Multimedia Security: Advanced Communications and Multimedia Security*.
12. Ning, P. and K. Sun (2003). "How to misuse AODV: a case study of insider attacks against mobile ad-hoc routing protocols." *Information Assurance Workshop, 2003. IEEE Systems, Man and Cybernetics Society.*
13. Papadimitratos, P. and Z. J. Haas (2002). "Secure Routing for Mobile Ad Hoc Networks." *Proc. SCS Communication Networks and Distributed Systems Modeling and Simulation Conf.*
14. Perkins, C. E. and P. Bhagwat (1994). "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers." *Proc. SIGCOMM '94 Conf. Communications Architectures, Protocols and Applications,*: 234-244.
15. Perkins, C. E. and E. M. Royer (1999). "Ad-hoc on-demand distance vector routing." *Mobile Computing Systems and Applications, 1999. Proceedings. WMCSA '99. Second IEEE Workshop on.*
16. Rebahi, Y., V. Mujica, et al. (2005). "SAFE: Securing pAcket Forwarding in ad hoc nEtworks." *5th Workshop on Applications and Services in Wireless Networks.*
17. Sanzgiri, K. (2002). "A Secure Routing Protocol for Ad Hoc Networks." *Proc. 10th IEEE International Conference Network Protocols*: 78-87.
18. Yih-Chun, H., D. B. Johnson, et al. (2002). "SEAD: secure efficient distance vector routing for mobile wireless ad hoc networks." *Mobile Computing Systems and Applications, 2002. Proceedings Fourth IEEE Workshop on.*
19. Yih-Chun, H. and A. Perrig (2004). "A survey of secure wireless ad hoc routing." *Security & Privacy Magazine, IEEE* 02(3): 28-39.
20. Zapata, M. G. and N. Asokan (1999). "Securing Ad Hoc Distance Vector Routing." *Proc. 2nd IEEE Workshop Mobile Computing Systems and Applications*: 90-100.
21. Zeng, X., R. Bagrodia, et al. (1998). "GloMoSim: a library for parallel simulation of large-scale wireless networks." *Parallel and Distributed Simulation, 1998. PADS 98. Proceedings. Twelfth Workshop on*.

7. Publications

Li, J., Moh, M., and Moh, T. S. “On Path-Based Reputation System for MANET Routing”, *IEEE ICC 2009, submitted*.