3-CE: A Cooperation Enforcement Technique for Data Forwarding in Vehicular Networks

Yao H. Ho[1], Ai H. Ho[1], Georgiana L. Hamza-Lup[2], and Kien A. Hua[1]

[1]School of Electrical Engineering and Computer Science, University of Central Florida, Orlando, FL 32816, USA
{yho, aho, and kienhua}@cs.ucf.edu

[2]Department of Computer Science and Engineering, Florida Atlantic University, Port St. Lucie, FL 34986, USA
ghamza@cse.fau.edu

ABSTRACT

Operations of vehicular ad hoc networks rely on the collaboration of participating nodes to route data for each other. This standard approach using a fixed set of nodes for each communication link cannot cope with high mobility due to a high frequency of link breaks. A recent approach based on virtual routers has been proposed to address this problem. In this new environment, virtual routers are used for forwarding data. The functionality of each virtual router is provided by the mobile devices currently within its spatial proximity. Since these routers do not move, the communication links are much more robust compared to those of the conventional techniques. In previous work [8], we investigate techniques to enforce collaboration among mobile devices by indentify and punish misbehaving users in supporting the virtual router functionality. The preliminary results showed the proposed 3CE approach is promising. In this paper, we provide a more detail and enhance version of the proposed technique. In addition, we provide more simulation results to indicate that the proposed technique is effective.

Keywords: Cooperation-enforcement; vehicular network; virtual routers; connectionless approach; selfishness.

1. INTRODUCTION

Vehicular Network (VNET) has attracted great research interest in recent years. Similar to Mobile Ad hoc NETworks (MANETs), a vehicular network is a self-organizing multi-hop wireless network where all vehicles (often called nodes) participate in the routing and data forwarding process. The deployment of ad hoc vehicular networks does not rely on fixed infrastructures such as router and base station, thereby posing a critical requirement on the nodes to cooperate with each other for successful data transmission. Many works (e.g., [2], [3], and [8]) have pointed out that the impact of malicious and selfish users must be carefully investigated. Existing cooperation enforcement techniques ([2], [3], [8], [10], [11], and [12]) cannot be adapted for recent advance in routing protocols – connectionless oriented approach ([4] and [7]). In particular, we are interested in the *Connectionless Approach for Street* (CLA-S) [6], in this paper. This technique does not maintain a hop-by-hop route for a communication session to minimize the occurrence of broken link. In CLA-S, the streets are divided into non-overlapping grid cells, each serving as a *virtual router*. Any physical router (i.e., mobile host), currently inside a virtual router, can help forward the data packet to the next virtual router along the virtual link. This process is repeated until the packet reaches its final destination. Since a virtual link is based on virtual routers which do not move, it is much more robust than physical link.

The goal of this research is to address the security and cooperation issues for *Connectionless Approach for Street* (CLA-S) in vehicular networks. There can be both selfish and malicious nodes in a vehicular ad hoc network. The selfish nodes are most concerned about their energy consumption and intentionally drop packets to save power. The purpose of malicious node is to attack network using various intrusive techniques. In general, nodes in an ad hoc network can exhibit Byzantine behaviors. That is, they can drop, modify, or misroute data packets. As a result, the availability and robustness of the network are severely compromised. Many works ([2], [3], [8], [10], [11], and [12]) have been published to combat such problem - misbehaving nodes are detected and a routing algorithm is employed to avoid and penalize misbehaving nodes. These techniques, however, cannot be applied to CLA-S since any node in the general direction towards the destination node can potentially help forward the data packets.

The primary contributions of this paper are as follows: 1) We introduce a cooperation enforcement technique, called 3CE (*3-Counter Enforcement*), for the *Connectionless Approach for Street* (CLA-S); 2) We apply the 3CE method to CLA-S; and 3) We present simulation results to show that with the 3CE features, CLA-S can prevent malicious nodes and enforce the cooperation among nodes to maintain the good performance of the network. The remainder of this paper is organized as follows. We review the *Connectionless Approach for Street* (CLA-S) in Section 2. In Section 3, we present our cooperation enforcement technique for CLA-S. We give simulation results in Section 4 to demonstrate the benefits of the proposed techniques. Finally, we draw conclusion on this work in Section 5.

2. CONNECTIONLESS APPROACH FOR STREET

To make the paper self contained, we first describe previous work, *Connectionless Approach for Street* (CLA-S), in more detail in this section. In CLA-S, the streets are divided into small "virtual cells." These cells are divided according to intersections and blocks (see Figure 1). Instead of maintain a hop-by-hop route between the source and destination node, the source only needs to maintain the location of the destination. Using this location information, the source dynamically computes and selects a list of grid cells that form a "connecting" path between the source and destination. The location of destination is discovered by the CLA's ***location discovery procedure*** where a simple broadcasting technique [5] is employed. The procedure is as follow. The source will broadcast a Location Discovery (LD) packet that contain source node ID, destination node ID, location (i.e., cell ID) of the source node, and a unique request ID to determine the location of destination. The LD packet will propagate unit it reaches the destination node. When the destination received the LD packet, the destination node will reply with Location Reply (LR) packet. The LR packet includes the location of the source node (i.e., source cell ID) and the location of the destination node (i.e., destination cell ID).

When a node (not the source node) receives a LR packet, it will determine if it is on the grid path using a Reference Line (see Figure 2). Reference Line is the straight line that connects the source (i.e., the center of the source cell) and the destination (i.e., the center of the destination cell). When the source node receives the LR packet, the source node can start communication sessions to the destination node by simply broadcast data packets which contain location information of Source and Destination (i.e., Source Cell ID and Destination Cell ID), Reference Line information, and current cell ID (i.e., cell ID of the node that is about to forward the data packet) in the packet header.

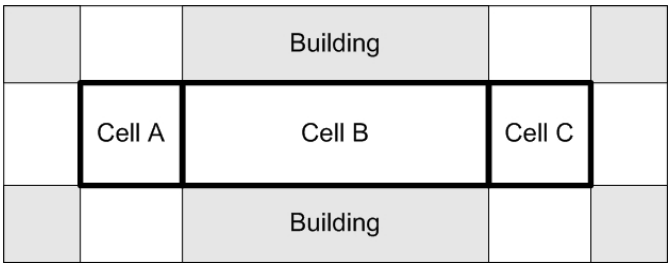

Figure 1. Grid path.

Once the Reference Line has been established, we need to determine the reference points. The *reference points* (RP's) on a reference line are the interceptions of the reference line and centerline of either a vertical street or a horizontal street (see Figure 2). Once all reference points of a reference line have been determined, we will use reference points to determine each *Forwarding Zone*. A *Forwarding Zone* is an area that is determined by a reference point or the center of a source cell. A reference point can be on a horizontal block, a vertical block, or an intersection (a block is considered as horizontal if the street it is on has a horizontal orientation; otherwise, it is vertical).

When a node *n* receives a data packet from *m*, the data forwarding procedure is as follows:

1) If *n* is the destination, *n* does not forward the data.
2) If *n* is not in the forwarding zones, *n* does not forward.
3) If *n* or any other node in the cell containing *n* has forwarded, *n* does not forward.
4) If Steps 1, 2, and 3 fail (i.e. *n* might need to forward the data), *n* delays the forwarding.
5) During this delay period, *n* will cancel the forwarding if *n* either hears the same packet from a neighboring node on the same cell or if *n* is in a block cell and *n* hears the same packet from both adjacent intersections.
6) At the end of the delay period, if the forwarding decision has not been cancelled, *n* forwards the data.

When a node receives a packet with a new Forwarding Area (because of a new reference line), it will compute the *Forwarding Zones* and save the result as a list of streets and the ranges of the streets that are encompassed by the *Forwarding Zones*. This allows the node a quick and simple way to determine if it is in a *Forwarding Zone* for subsequent packets with the same Forwarding Area.

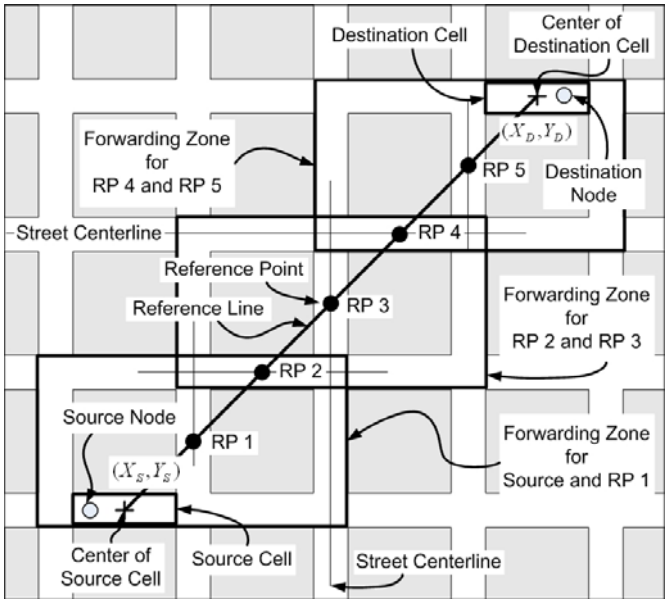

Figure 2. Reference Line, Reference Points, and Forwarding Zones

In the above procedure, the delay of a node *n* is computed as follows:

$$DELAY_n = \left| \frac{\alpha}{2 \bullet D\_Dist_n} - \frac{\alpha}{2 \bullet Dist_n} \right| \quad \textbf{(1)}$$

,where α is a maximum delay constant in μsec, D_Dist_n the distance between node *n* and the center of the cell denoted by the *Destination Cell ID* in the packet header, and $Dist_n$ the distance between node *n* and the center of the cell denoted by the *Current Cell ID* (cell of previous relaying node *m*) in the

packet header (See Figure 3). The significance of this equation is to select a node farther away from *m* and closer to the destination node to forward the data packet.

If the node *n* is at an intersection of two streets, we will set a shorter delay period. In the simulation, the delay for an intersection node is set to one third of the normal *DELAY*. The reason for this is that, when at an intersection, a node's effective radio range can cover the 2 intersecting streets compared to the single street coverage of another node on a block. The detail information for CLA and CLA-S such as Path Computation, Data Forwarding, Path Update, and Empty Cell/Obstacle Recovery can be found in [6] and [7].

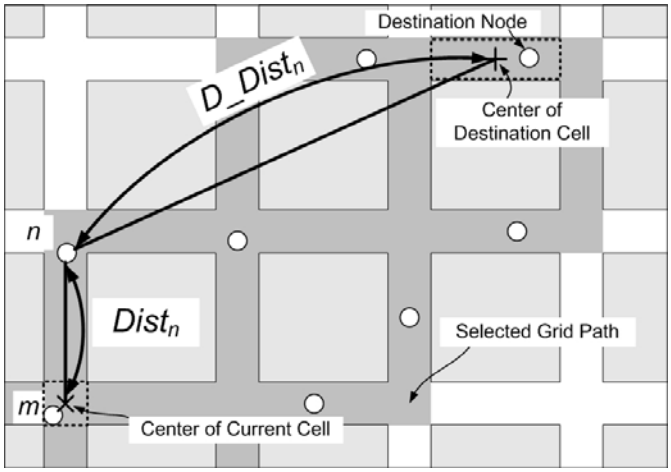

Figure 3. Delay for node *n*.

3. 3-COUNTER ENFORCEMENT (3CE) FOR COLLABORATION IN FOR CLA-S

In this section, we first briefly describe the configuration of mobile nodes and their Tamper Proof Module. We then present our cooperation enforcement techniques, called 3CE, for CLA-S.

3.1 Node Configuration and Tamper Proof Module

The proposed technique is based on nodes with the following configuration. First, nodes are equipped with wireless interface cards that can be switched to detection mode to "detect" data transmission on a "suspicious" node in their proximities. Second, connectionless-oriented routing protocol is employed in the network layer. Without loss of generality, we base our discussion on the more recent techniques developed for routing in VNETs (i.e., Connectionless Approach routing protocol (CLA-S) [6]). Nevertheless, the technique can be incorporated into any location-aid protocols to protect nodes against uncooperative behaviors. Third, reliable communication protocols such as TCP cannot be employed in this type of routing protocols. While other routing protocols need to maintain (proactively or reactively) neighbor nodes location information and establish a connection to the next hop before forwarding a data packet, CLA-S simply forward data packet without first establishing the link to the next node. Any node that happens to be in the general direction towards the destination node can compete for the "right" to forward data packets.

In addition, similar to the techniques presented in [3] and [8], we also equip each node with a tamper resistant module. All other hardware and software components are susceptible to illicit modifications. We notice that a tamper-proof security module remains controversial [13], but it proves to be inevitable in a large scale and high mobility network environment. Our approach guarantees that as long as the tamper resistant module is not compromised, nodes cannot benefit from uncooperative behaviors. Some mission critical data is stored in the tamper resistant module. This information include: 1) a unique *ID* of the node; 2) a pair of public/private keys; 3) a ***Forward Request Counter*** that counts number of packets that are received and need to be forwarded; 4) a ***Forward Counter*** that counts number of packets have been forwarded; 5) a ***Location Discovery Counter*** that counts number of Location Discovery packets initiated by a node; 6) a ***Session Table*** that keeps track ongoing communication sessions; 7) a **Counter Update Procedure** that updates the three counters; 8) a **Misbehavior Detection Procedure** that initiates the detection to identify a malicious node. Since the tamper proof module maintains information of three counters that are used to determine maliciousness of a node and initiate the detection, hereafter we also refer to this module as the 3C Module, and the proposed technique as the 3CE or 3C Enforcement technique.

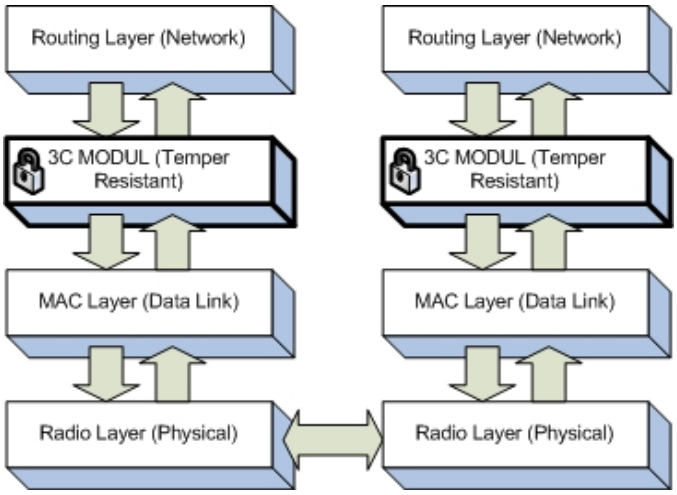

Figure 4. Layer Stucture.

The 3C Module inspects Location Discovery packets, Location Reply packets and data packets exchange between the network layer and the MAC layer (see Figure 4); and the module updates the counters as follows:

1. When a new packet arrives at a non-destination node, it updates (i.e., increment by one) its ***Forward Request Counter***;

2. When a node forward a packet, it updates (i.e., increment by one) its ***Forward Counter***; and
3. When a note initiates a Location Discovery packet, it updates (i.e., increment by one) it's ***Location Discovery Counter***.

In addition, the 3C Module constructs and adds 3C's header (i.e., the value of three counters) to the Location Discovery packet as in various layers of the OSI model.

3.2 3C Module

In CLA-S, the location of the destination node is needed before a node can start a data transmission session to another node. Thus, a Location Discovery packet is broadcasted to find the destination. Once its location is determined, intermediate nodes can forward data packet according to the general direction towards the destination; and all packets exchanged between nodes are examined by the nodes' 3C Module.

In a 3C Module, three counters (i.e., ***Forward Request Counter***, ***Forward Counter***, and ***Location Discovery Counter***) are updated according to the counter update procedure. These counters are maintained by the node's own 3C Module (see Figure 4). Similar to [3] and [8], we assume the 3C Module is a tamper resistant module that malicious users cannot contaminate it.

When a source node *S* initiates a Location Discovery packet, node *S*'s 3C Module adds the 3C's header to the Location Discovery packet as in various layers of the OSI model. **3C header** contains the value of three counters (i.e., ***Forward Request Counter***, ***Forward Counter***, and ***Location Discovery Counter***) of node *S*. Based on this header, neighboring nodes of *S* can decide to forward or discard the Location Discovery packet. If a node *n* "suspects" the source node *S* is misbehaved, *n* invokes its **Misbehavior Detection Procedure**. A node suspects another node is misbehaving if one of the following is true: a) the *Forward Ratio* (i.e., ratio of *Forward Counter* to *Forward Request Counter*) of *S* falls below the *Forward Ratio* of *n*; or b) the *Request Ratio* (i.e., ratio of the *Location Discovery Counter* to *Forward Counter*) of *S* rises above the *Request Ratio* of *n*. If so, *n* exchanges 3C information (i.e., the value of the three counters) with its neighboring nodes to determine the network condition in the local area (i.e., *n*'s neighboring nodes). If the source node *S* is identified (by **Misbehavior Detection Procedure**) as misbehaving, its neighboring nodes will penalize this node by not forwarding *S*'s Location Discovery packets.

In order for malicious nodes to rejoin the network, non-malicious nodes still allow malicious nodes to participate in forwarding data. Unlike many techniques that avoid the malicious nodes during the routing procedure, our approach allows malicious nodes to rejoin the network by contributing its share (i.e., forwarding data for others) of network workload. This way, nodes are given more incentive to act collaboratively. By forwarding data packets for other nodes, a malicious node can increase its ***Forward Counter***. When its ratio of ***Forward Request Counter*** to ***Forward Counter*** rises above threshold α and its ratio of ***Location Discovery Counter*** to ***Forward Counter*** fells below threshold β, the malicious node will again be allowed to join the network, i.e., its neighboring nodes again help forward its Location Discovery packets. We elaborate the above processes in the following sections.

3.3 Counters Update during the Location Discovery Phase

As mentioned earlier, a node needs to find the location of the destination before it can start to send data packets in connectionless-oriented protocols such as CLA-S. A node can initiate a Location Discovery procedure, receive a Location Discovery packet, or forward/reply a Location Discovery packet. To initiate a Location Discovery procedure, a source node broadcasts a Location Discovery packet.

Location Discovery packet: Location Discovery packet contains the following information: source node ID (*source_ID*), source node's location (*S_cell_ID*), destination node ID (*destination_ID*), destination node's location (*D_cell-_ID*), forward node ID (*forward_ID*), and forward node's location (*F_cell_ID*).

When a node receives a Location Discovery packet, it checks if it is the destination node. If so, it returns a Location Reply packet that contains its location (*D_cell_ID*); otherwise, if the node did not see this Location Discovery packet before, it adds its ID and its cell ID (i.e., forward node ID - *forward_ID* and the currently location - *F_cell_ID*) and broadcasts the Location Discovery packet to other nodes. In Figure 5's Routing Layer (i.e., Network Layer), we show the data forwarding procedure for CLA-S.

Session Table: Each node maintains a *Session Table* in its 3C Module to track all the ongoing communication session. An ongoing communication session is identified by a *session_ID* which is a pair of *source_ID* and *destination_ID* of the communication session. This table contains the following information for each entry (i.e., communication session): *session_ID* (i.e., a pair of *source_ID* and *destination_ID*) and a *time to live* (*TTL*) *timer*. An entry is deleted from the *Session Table* when one of the following information is true: (i) A communication session ended; (ii) Entry's *TTL* (time to live) *timer* expired; (iii) Entry belongs to an identified malicious node. An entry's *TTL timer* is reset when a packet received such that: *a*) the packet corresponds to this entry (i.e., *source_ID* and *destination_ID* = *session_ID*) and; *b*) it is not from a malicious node.

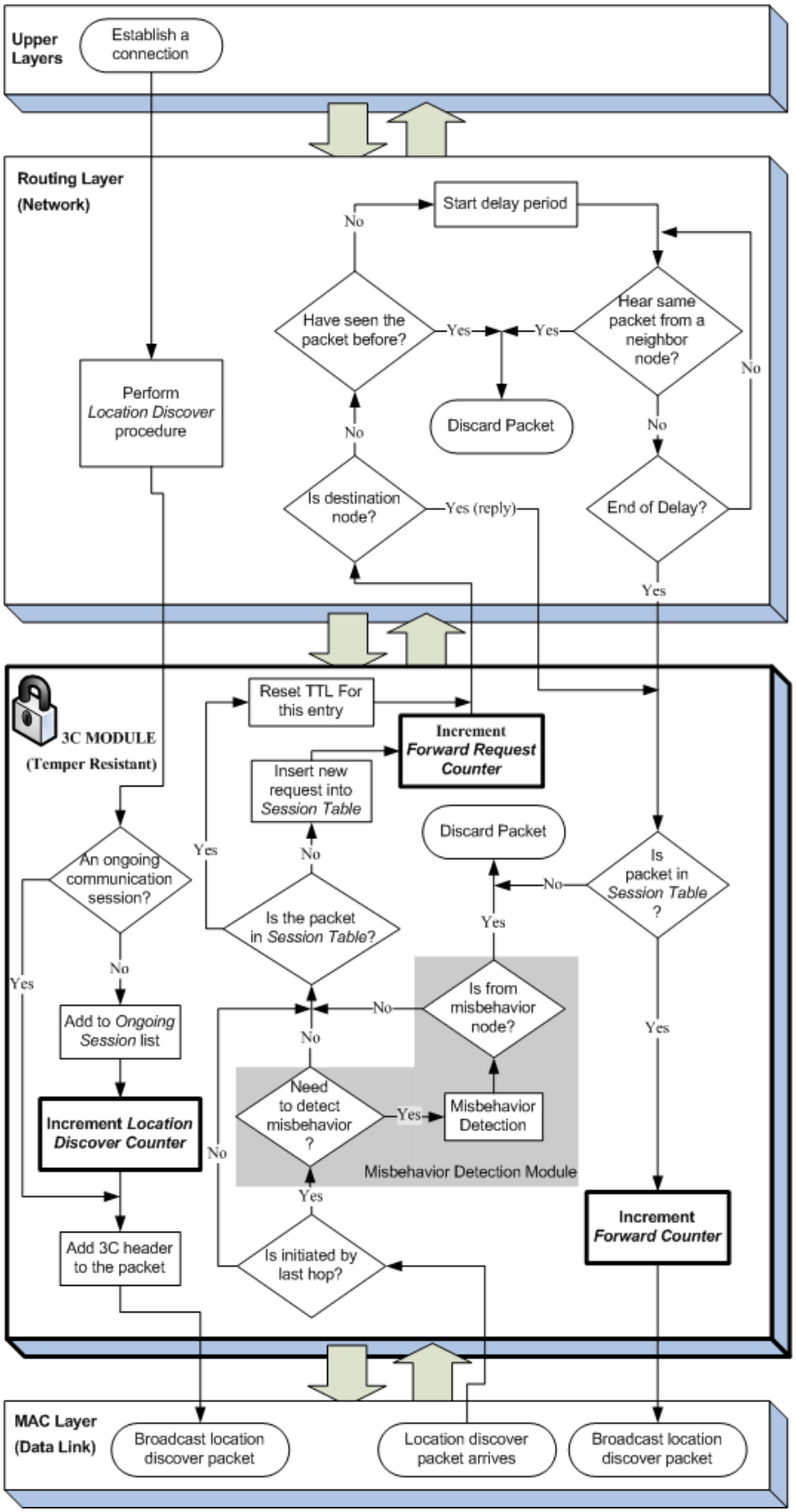

Figure 5. Update the counter during the Location Discovery phase.

3.3.1 Initiate Location Discovery

When a **Location Discovery procedure** in the routing layer passes an initiated Location Discovery packet to the 3C Module, it processes the packet and updates the ***Location Discover Counter*** as follows (see Figure 5):

1. The 3C Module determines if this Location Discovery packet belongs to one of the initiator's (i.e., the source node's) ongoing communication session in the *Session Table*. If it does not belong to an ongoing session, go to Step 2; otherwise, go to Step 3.
2. The 3C Module increments the ***Location Discovery Counter*** by one and adds it to the *Session Table* (and go to Step 3).
3. The 3C Module adds a 3C header containing the values of the three counters (i.e., ***Forward Request Counter***, ***Forward Counter***, and ***Location Discovery Counter***) to this Location Discovery packet before passing it to the MAC Layer for broadcast to other nodes.

In the connectionless-oriented approach, the destination of a communication session is periodically updated according to the mobility of the destination node. The location of the source node is updated by piggybacking the location information in the data packets. However, a source node sometime needs to re-discover the location of a destination node due to packet losses caused by congestion, mobility, or channel errors. Thus, we differentiate between the initial location discovery and the location discovery that is re-establishing an ongoing communication session.

3.3.2 Receive Location Discovery Packet

When a Location Discovery packet broadcast from a node *m* to any of its one-hop neighbor node *n*, *n*'s MAC Layer passes the packet to its 3C Module for processing the Location Discovery packet and updating the ***Forward Request Counter*** as follow (see Figure 5):

1. The 3C Module determines if *m* is the source node that initiated this Location Discovery packet (i.e., packet's *source_ID = forward_ID*). If so, go to Step 2; otherwise, go to Step 3.
2. If *m* is the source node of this Location Discovery packet, the 3C Module in *n* uses the information in the packet's 3C header to determine if there is a need to start the detection procedure to examine *m*'s behavior. We will discuss when to initiate the misbehavior detection and the procedure for misbehavior detection in Section 3.5 and 3.6, respectively. If node *m* is confirmed to be misbehaving, the 3C Module of node *n* discards the packet (as punishment); otherwise, go to Step 3.
3. Node *n* keeps records of ongoing communication session in its *Session Table*. If the arriving Location Discovery packet's *source_ID* and *destination_ID* match an entry in node *n*'s *Session Table* (e.g., packet's *source_ID* + *destination_ID* = *session_ID*), its 3C Module resets the *time to live* (TTL) timer of the corresponding entry. Next, the Location Discovery packet is then passed on to the routing layer (Step 5).
4. If the Location Discovery packet is not belonged to any ongoing session in the *Session Table* (e.g., packet's *source_ID* and *destination_ID* ≠ *session_ID*), the 3C Module updates the *Session Table* and increases the ***Forward Request Counter*** by one. The 3C Module then passes the Location Discovery packet to the routing layer for further processing (Step 5).
5. Depending on CLA-S routing protocols, node *n* can discard the packet, continue to forward (i.e., pass back down to lower layers), or initiate a reply procedure (i.e., reach the destination). In Figure 5, we show the routing protocol for CLA in the Routing Layer.

3.3.3 Forward or Reply Location Discovery Packet

Depending on the role of a node in a communication session (e.g., forwarding node or destination node), a node can forward the Location Discovery packet, reply to the Location Discovery packet with a Location Reply packet, or discard the Location Discovery packet according to its routing protocol. A Location Reply packet is generated by a node's Routing Layer when a Location Discovery packet arrived at a destination. This destination node needs to reply the source node of the Location Discovery packet. If a node is the destination, its Routing Layer generates a Location Reply packet and passes this reply packet to 3C Module.

When Routing Layer submits a Location Discovery packet or a Location Reply packet to 3C Module, 3C Module processes the packet and updates the ***Forward Counter*** as follows:

1. 3C Module determines if the Location Discovery packet or the Location Reply packet matches an entry in the *Session Table*. To determine if the Location Reply packet matches an entry in the *Session Table*, 3C Module simply reverses the order of *source_ID* and *destination_ID* of this packet. If the packet matches an entry in the *Session Table*, go to Step 2. Else, the packet is discarded because a malicious node can generate dummy packets to increase its ***Forward Counter*** to avoid detection.
2. 3C Module increases the ***Forward Counter*** by one. Then, the Location Discovery packet or the Location Reply packet is passed to MAC Layer.

3.4 Counters Update during the Data Forwarding Phase

Once the location of the destination node is determined, the source node can start a communication session. In CLA-S, nodes simply forward data packets without first establishing the link to the next node. Any node that happens to be within

the forwarding zone and in the general direction towards the destination node can compete for the "right" to forward data packets. When a source node *s* starts to send the data packet from routing layer to 3C Module, *s*'s 3C Module simply passes the data packet to the MAC layer without updating any counter.

3.4.1 Receive Data Packet

When a node *n* receives a data packet, its MAC Layer passes the data packet to its 3C Module. Then, *n*'s 3C Module updates the ***Forward Request Counter*** as follows:

1. 3C Module determines if the data packet corresponds to a communication session in *n*'s *Session Table*. If so, go to Step 2. Else, go to Step 3.
2. *n*'s 3C Module resets the *time to live* (*TTL*) *timer* of the corresponding entry in the *Session Table* and passes the data packet to the routing layer. Depend on different routing protocols, the data packet is either discarded or forwarded.
3. If the data packet is not belonged to any ongoing session in the *Session Table*, the 3C Module updates the *Session Table* and increases the ***Forward Request Counter*** by one. The 3C Module passes the Location Discovery packet to the routing layer for further processing (e.g., discard or forward data packet).

3.4.2 Forward Data Packet

Depend on the routing protocol, the data packet is either discarded or forward (see the "Routing Layer" in Figure 5). In connectionless-oriented approach, every node has equal probability of participate in the data forward procedure. If the routing layer decides to forward data packet, it returns a data packet to 3C Module. The 3C Module processes the data packet and updates the ***Forward Counter*** as follows:

1. 3C Module determines if the data packet matches any entry in the *Session Table*. If so, it increases the ***Forward Counter*** by one and passes the data packet to the MAC layer.
2. Else, the data packet is discarded. We discard any packets that are not in the *Session Table* due to the same reason as discussed in Section 3.3.3. A malicious node can generate dummy packets to avoid evoking the Misbehavior Detection procedure.

3.5 Initiate Misbehavior Detection

By modifying its own routing protocol, a malicious node can intentionally drop (i.e., discard) packets to save its power. However, in the connectionless-oriented approach, every node has an equal chance to participate in a forwarding process. Thus, 3C Module needs to determine to whether to "**invoke"** the **Misbehavior Detection procedure**. In order to determine if there is a need to invoke the Misbehavior Detection procedure, 3C Module exams the 3C header in the Location Discovery packet and calculates two ratios, ***Forward Ratio*** (***FR***) and ***Request Ratio*** (***RR***) as follow:

- ***Forward Ratio***$_i$ $(\boldsymbol{FR}_i) = \dfrac{Forward\ Counter_i}{Forward\ Request\ Counter_i}$
- ***Request Ratio***$_i$ $(\boldsymbol{RR}_i) = \dfrac{Location\ Discovery\ Counter_i}{Forward\ Counter_i}$

  , where *i* is the node that initiated this Location Discovery packet (i.e., the source node).

When a node *n* receives a Location Discovery packet from a node *m*, *n*'s 3C Module checks if *m* is the initiator (i.e., source node) of this Location Discovery packet using the information included in the packet (see Section 3.3). If *m* is not the initiator, *n*'s 3C Module does not invoke the detection procedure. Then, this Location Discovery packet passes to the Counter Update procedure for further process (see Figure 5). If *m* is the initiator of this Location Discovery packet, *n*'s 3C Module checks the 3C header included in this Location Discovery packet for the following conditions:

1. $\boldsymbol{FR}_m < \boldsymbol{FR}_n$
2. $\boldsymbol{RR}_m > \boldsymbol{RR}_n$ * *Initiate Detection Threshold*

If one of the above condition is true, *n*'s 3C Module broadcasts a 3C packet (including *n*'s 3C information) to its one-hop neighbor nodes. When a node receives *n*'s 3C packet, it replies with its own 3C information. When *n* receives its neighboring nodes' replies, *n* calculates the ***Local Average Forward Ratio*** (***LAFR***). This ratio is calculated as follow:

$$\boldsymbol{LAFR}_n = \frac{\sum_{i=1}^{k}(FR_i) + FR_n}{k+1}$$

, where *k* is number of neighboring nodes for *n* (excluding *m*).

In Vehicular Network, network conditions, such as density and congestion, can change dynamically. Thus, the ***Local Average Forward Ratio***$_n$ ($\boldsymbol{LAFR}_n$) is merely the local network condition around *n*. If $\boldsymbol{FR}_m \geq \boldsymbol{LAFR}_n$, it means that network condition at area of *m* might be congested which causes *m* not forward packets. Thus, we do not need to invoke the Misbehavior Detection procedure. On the other hand, if $\boldsymbol{FR}_m < \boldsymbol{LAFR}_n$, then *m* might be misbehaving by not forwarding packets. In this case, *n* activates its **Detection Mode**. Notice that all the neighboring nodes of *m* and *n* can activate its **Detection Mode** (but not at same time) because their ***Forward Ratios*** are similar. When a node activates its **Detection Mode**, it continues to forward for other nodes except for the suspicious node.

To avoid evoking the Misbehavior Detection procedure, malicious nodes can initiate dummy packets to increase their own ***Forwarding Counter***. Although, by doing so, malicious nodes defeat the purpose of saving power. Nevertheless, 3C Module can prevent this misbehavior act by compare the outgoing packets against the *Session Table*. If the packet does not match any entry in the *Session Table*, 3C Module discards this dummy packet.

3.6 Detection Mode

The **Detection Mode** has two states: *Listening*-State and *Detecting*-State. Initially, a node in the Detection Mode is set to *Listening*-State. In the *Listening*-State, a node *n* waits for a random period of time. During this delay period of time, *n* does the following:

1. If *n* hears a Detection packet from another node to test node *m* (i.e., the suspect node), *n* resets the delay time. A Detection packet is generated by **Misbehavior Detection procedure** to test a suspicious node.
2. If *n* hears a Detection packet been forwarded by *m*, *n* exits the **Detection Mode**. By exiting the **Detection Mode**, *n* forwards *m*'s Location Discovery packet. Similarly, all other nodes that are in their Detection Mode (*Listen*-State) hear *m* forwarded the Detection packet will exist their Detection Mode.

At the end of delay period, node *n* enters the *Detecting*-State. In the *Detecting*-State, *n* invokes the **Misbehavior Detection procedure** to determine if *m* is a malicious node.

3.7 Misbehavior Detection Procedure

The detection mechanism can be implemented as a software application as proposed in [3] for lower cost. Alternatively, it can also be implemented as a build-in component of the temper resistant module for better security. Without loss the generality, we base our discussion on the latter option.

The purpose of the **Misbehavior Detection procedure** is to detect uncooperative behaviors that result in disruption or degradation of data transmission. We focus on network layer attacks and do not address lower level threats such as physical layer jamming and MAC layer disruptions. The attacks contained by the Misbehavior Detection Module are as follows. First, if there is a suspicion of dropping packets was detected during the location discovery phase, the Misbehavior Detection procedure is invoked. Second, the **Misbehavior Detection procedure** captures malicious users who deliberately discard packets that they are obligated to forward either for selfish purposes or to mount denial of service attacks.

When a node *n* invokes its **Misbehavior Detection procedure** to detect a suspect node *m*, the procedure is as follows:

1. *n* calculates a *virtual link* (see Figure 6) using the location information (i.e., cell ID) contained in *m*'s Location Discovery packet.
2. Based on this *virtual link*, *n* generates a Detection packet (i.e., similar to regular data packet). The source location and destination location of this Detection packet are as follow:
   - Source node's location (*S_cell_ID*) of this Detection packet is the cell behind of *n*, relative to *m*.
   - Destination node's location (*D_cell_ID*) of this Detection packet is the cell behind of *m*, relative to *n*.
3. Next, *n* broadcasts this Detection packet. All the neighboring nodes of *m* are in Detection Mode and will not forward this Detection packet.
4. *n* waits for a *t* period of time (*t* = maximum delay time in the routing layer).
5. At the end of the delay, if *n* does not receive the Detection packet forwarded by *m* (i.e., *forward_ID* = *m*), *n* repeats the process again for two times (total of 3 times).

If *n* receives the detection packet which is forwarded by *m*, *n* (and all the neighboring nodes of *m*) exits the Detection Mode. *n* forwards *m*'s Location Discovery packet because *m* has passed *n*'s Misbehavior Detection procedure. If *n* does not receive the detection packet from *m*, *n* punishes *m* by discard *m*'s Location Discovery packet for period of $t_{punish} = C \times (\boldsymbol{LAFR_n} - \boldsymbol{FR_m})$. Thus, the punishment period is proportion to individual (misbehaving) node's misbehaved level.

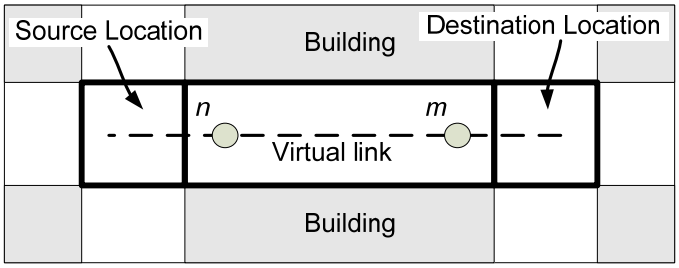

Figure 6. Virtual link for a Detection packet.

4. EXPERIMENTAL RESULTS

We conducted various experiments to verify the effectiveness of the proposed 3CE (*3-Counter Enforcement*) scheme in enhancing performance of vehicular network. In this section, we first introduce the implemented schemes, simulation setup and parameters. We then study the proposed technique based on various performance metrics.

4.1 Schemes Implemented

We implemented three schemes, namely the **reference** scheme, the **defenseless** scheme and the proposed **3CE** scheme, for performance evaluation. In the **reference** scheme, all the nodes act collaboratively and relay data for each other. In the **defenseless** scheme, a certain fraction of nodes are misbehaving as they failed to participate in forwarding procedure. In other wards, these nodes discard any packets not destined at them. No detection or prevention mechanism is implemented so that the network is totally "defenseless". Finally, in the proposed **3CE** scheme, misbehaving nodes are detected and punished. A malicious node can recognize itself is been punished when Location Discovery packets of the node has been dropped four consecutively times. Once malicious nodes recognized themselves been punished, they participate in forwarding data to rejoin the network.

4.2 Simulation Setup

All the experiments were conducted using GlomoSim [14]. This simulator, developed at UCLA, is a packet-level simulator specifically designed for ad-hoc networks. It follows the OSI 7-layer network communication model. Although, popular simulators such as NS-2, OPNET Modeler, and GloMoSim provide advanced simulation environments to test and debug network protocols, we prefer GloMoSim due to its ability to handle high mobility of nodes and its scalability of handle large number of nodes and size of network area. Unlike other simulators, GloMoSim uses the parallel discrete-event simulation capability provide by Parsec [1].

Experiments were based on a mobile ad hoc network with 200 nodes. The field configuration is a 2000 by 2000 meters field with a street width of 10 meters and building block size of 100 by 100 meters. All nodes employ 802.11 at the MAC layer. Each node has a radio range of about 375 meters. The nodes move in the directions permitted in the streets. Upon arriving at an intersection, a node probabilistically changes its directions of movement (e.g., turn left, turn right, or continue in the same direction). Traffic applications are constant-bit-rate sessions involving 1/10 of all nodes. Each data packet is 512 bytes. Multiple simulation runs (100 runs per setup on average) with different seed numbers were conducted for each scenario and collected data were averaged over those runs. The total simulation duration for each run was 20 minutes (1200 seconds). We varied the number of misbehaving nodes (i.e., 5%, 10%, 20%, and 30% of total number of nodes) and node mobility (i.e., 10 *m/s* to 25 *m/s* or 22 *mile/hr* to 56 *mile/hr*). The *Initiate Detection Threshold* (*IDT*) is set to 1.2. This threshold determine percentage of a node require to participated in forward procedure in order not to initiate the 3C's detection procedure. For example, when the threshold set to 1.2, a node is allow of 20% of packet drop due to either network condition or mobility. Initially, misbehaving nodes drop all the received packets. Once misbehaving nodes been identified (i.e., all their Location Discovery packets are drop by other neighboring nodes), they behave normally until they are no longer identified as misbehaving nodes (i.e., their Location Discovery packets are forwarded by others).

4.3 Metric

In the experiments, we evaluated the proposed scheme based on the following metrics:

1. **Packet delivered ratio (*P*)**: The ratio of the data packets delivered to the destinations and the data packets generated by the CBR source. This measures the rate at which effective data transmission is performed. It is also a good indicator of the degree of collaboration among the nodes.
2. **Misbehaving node detection ratio (*D*)**: The ratio of the number of misbehaving nodes that were correctly identified to the total number of misbehaving node that have actually acted uncooperatively during the simulation.
3. **False accusation ratio (*F*)**: The ratio of the number of 3C Modules that incorrectly accused benign hosts to the overall number of misbehaving nodes that 3C Module identified.
4. **Control overhead ratio (*C*)**: The ratio of the number of routing packets transmitted per distinct data packet delivered to a destination.
5. **End-to-end delay (*D*)**: The number measured in *milliseconds*, includes detecting and processing malicious nodes delay, route discover latency, queuing delays, retransmission delay at the MAC, and propagation and transmission times. This measures the total delay time from a sender to a destination (without communication sessions that belong to misbehaving nodes).
6. **Active detection ratio (*A*)**: The ratio of the number of nodes activated their Detection Mode per misbehaving node's location discovery packet.

4.4 Experimental Results

We present the simulation results in this section.

4.4.1 Packet Delivered Ratio

By employing the proposed scheme, significantly more data can be successfully delivered to the destinations since nodes are now required to participating in data forwarding. Figure 7 depicts the practical scenarios where the number of malicious node is 10% and 20% of the total nodes. We observe in the case of fewer malicious nodes (less then 10%), the CLA-S with 3CE (*CLA-S-3C*) has very close throughput to the references CLA (*CLA-S-Reference*). The proposed technique improves the deliver ratio by more than 25% compare to the defenseless scheme.

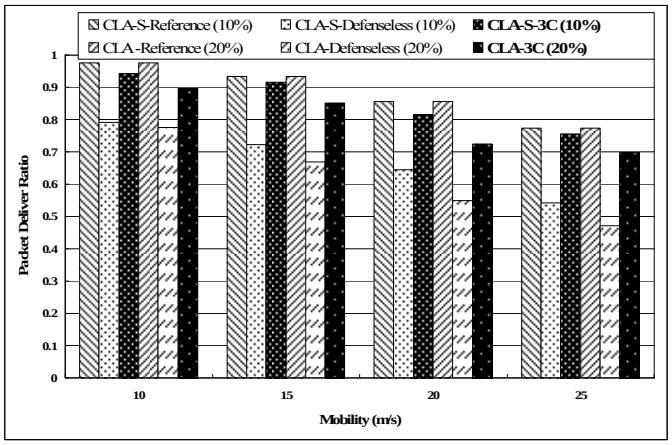

Figure 7. Packet Deliver Ratio (*P*) with 10% and 20% Malicious Nodes.

Another important factor to the performance of packet deliver ratio is the speed of mobility. Due to mobility of mobile hosts, addressing frequent and unpredictable topology changes is fundamental to MANET research. As the mobility of node (e.g., speed) increase, the performance of all three schemes (i.e., 3CE, reference, and defenseless) are decreased. Similarly when we increased mobility and number of malicious nodes

(see Figure 7), the packet deliver ratio is also decreased as the result. However, consider of mobility increased from 10 *m/s* (or 22 *miles/hour*) to 25 *m/s* (or 56 *miles/hour*), the deliver ratio decreased only 20% in average. Thus, the protocol is still suited for many applications (e.g., video and audio) with error correction code.

4.4.2 Misbehaving Node Detection Ratio

We list the results of misbehaving node detection ratio for various simulation scenarios in Table 1. They indicate that the proposed misbehaving node detection mechanism is very effective. In most cases, the 3CE's detection ratio is about 87%. The results demonstrate that on-demand misbehaving node detection is applicable. Since the proposed 3CE technique can adapt by the CLA-S, it is ideal for highly dynamic MANETs such as vehicle-to-vehicle networks.

Table 1. Detection ratio and False Accusation ration of CLA-S with 3CE.

	Detection Ration (D)				False Accusation Ratio (F)			
Speed (m/s)	10	15	20	25	10	15	20	25
5% misbehaving nodes	89%	88%	83%	81%	0%	2%	3%	2%
10% misbehaving nodes	93%	91%	86%	88%	1%	2%	2%	3%
20% misbehaving nodes	91%	85%	89%	87%	1%	1%	2%	2%
30% misbehaving nodes	91%	87%	84%	85%	2%	2%	4%	5%

4.4.3 False Accusation Ratio

We report the false accusation ratios of the proposed 3CE scheme under various scenarios in Table 1. We conclude that in all node mobility scenarios the false accusation ratio is very low. We observe that this ratio is higher when the speed of nodes is increased. This is due to the fact that some of the suspect nodes moved out of the detection node's radio range and were thus incorrectly classified by 3CE's Misbehaving Detection procedure as misbehaving nodes, thereby lifting the false accusation ratio. Nevertheless, further investigation of simulation log files shows that under all simulation configurations, on average less four nodes were incorrectly accused. Both results indicate that the proposed detection mechanism is able to detect most of the in-cooperative nodes with very low false accusation ratio.

4.4.4 Control Overhead Ratio

With 20% of malicious nodes, we observe that the Control Overhead Ratio is higher when the speed of nodes is increased (see Figure 8). Similar to False Accusation Ratio, this is due to the fact that some of the suspect nodes moved out of the detection node's radio range and were thus cause some nodes to invoke 3CE's Misbehaving Detection procedure, thereby lifting the Control Overhead Ratio. However, this is inevitable in most on-demand misbehaving node detection approaches.

4.4.5 End-to-End Delay

We report the increasing of end-to-end delay in Figure 9. With 20% of malicious nodes, we observe that the proposed scheme incurs minimum end-to-end delay. In most of cases, the length of delay increases approximately six *milliseconds* compared the reference schemes. This can due to the fact that other nodes can continue to forward data packet while one node is detecting a malicious node. Also, malicious nodes are unable to utilize the network resource once they are identified. Since we punish the misbehaving nodes by not forwarding their Location Discovery packet for a period of time, we did not include the communication sessions which the source nodes are misbehaving nodes.

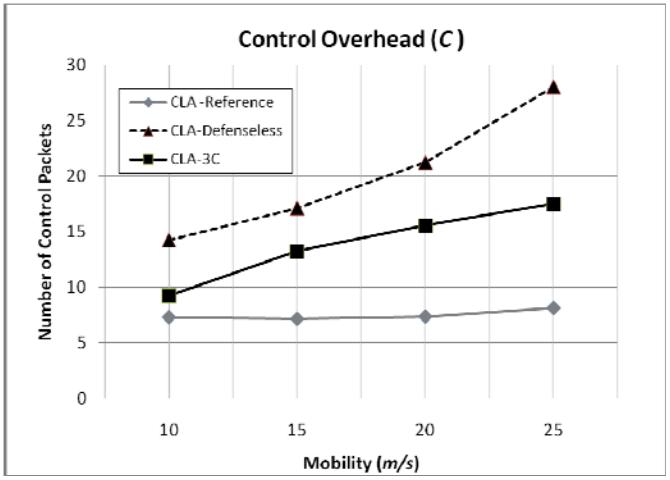

Figure 8. Control Overhead (*C*) with 20% Malicious Nodes.

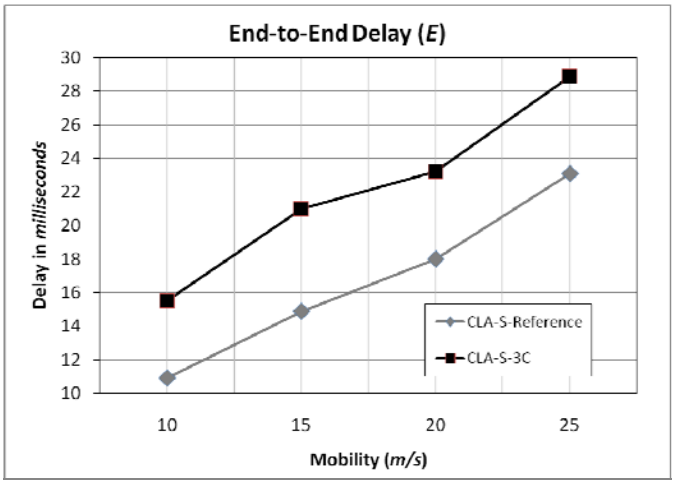

Figure 9. End-to-End Delay (*E*) with 20% Malicious Nodes.

4.4.6 Active Detection Ratio

With speed of 20 *m/s* and 20% of malicious nodes, we observe that the number of nodes activated Detection Mode per malicious node's location discover packet (that attempt to establish a connection) becomes fixed even the number of nodes in the network increased from 200 nodes to 1400 nodes (see Figure 10). In fact, if we assume a malicious node is stationary and no obstacle (e.g., buildings) within the network, the maximum number of neighboring nodes that are in the Detection Mode (i.e., *Detecting*-State) is six (see Figure 11). If a malicious node is moving at speed of 20 *m/s*, then the moving rang (i.e., a circle with radius of *r*) within the

maximum delay time (t = 2 *seconds*) of the Detection Mode is as follow:

$$r = speed * time = 20(m/s) * 2(s) = 40(m)$$

With radio range of a node is 375 meters; the radius of circular area of the maximum area of neighboring nodes that can activate Detection Mode is as follow:

$$r_{Detection} = r + \text{radio range} = 40(m) + 375(m) = 415(m)$$

Thus, the maximum number of neighboring nodes that are in the Detection Mode is seven nodes (see Figure 12). In order for a malicious node to move out of area where its neighboring nodes have activated the Detection Mode, the malicious node needs to travel of 790 *meters* (i.e., 415 *m* + 375 *m*). With maximum moving speed of 20 m/s, the time a malicious node to move out of this area is 39.5 seconds (i.e., 790(*m*) / 20 (*m/s*)). Thus, the upper bond of Active Detection Ratio (A) is 7 nodes per 39.5 seconds (or 0.18 nodes per second). This confirms with our simulation study. In fact, the result in Figure 10 shows that our approach is able to adapt under high mobility (i.e., variety of applications – vehicular networks) and high density networks (i.e., scalable).

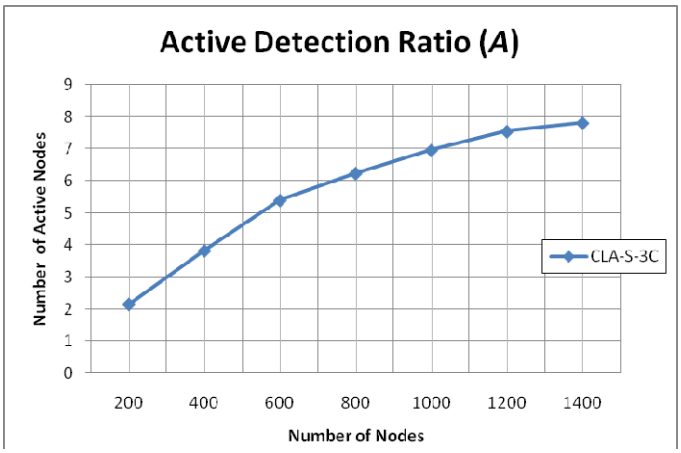

Figure 10. Active Detection Ratio (A) with 20% Malicious Nodes.

5. Conclusion

In this paper, we proposed an efficient 3CE (*3-Counter Enforcement*) scheme to enforce collaboration for CLA-S in vehicular network. Our contributions are as follows. 1) We introduce an on-demand approach to misbehaving-node detection for the CLA-S approach. Since the CLA-S addresses highly dynamic networks (i.e., vehicle-to-vehicle networks), the existing misbehaving-node detection techniques are not suitable. Our approach supports this type of routing protocol under high mobility environments. 2) Each node maintains three counters to represent its own status (i.e., reputation). Since nodes only determine their neighboring nodes' counters information when a location discovery phase, no additional information is needed under a normal operation (i.e., nodes behave normally).

We conducted various experiments to study the effectiveness and efficiency of the proposed 3CE technique. The simulation results indicated that the proposed technique is very effective in enforcing collaboration. The degree of collaboration is significantly strengthened as the network throughput is greatly improved compare to a defenseless network. Such improvement is accomplished with almost no false accusation of cooperative nodes. As of efficiency, the proposed scheme incurs minimum overhead.

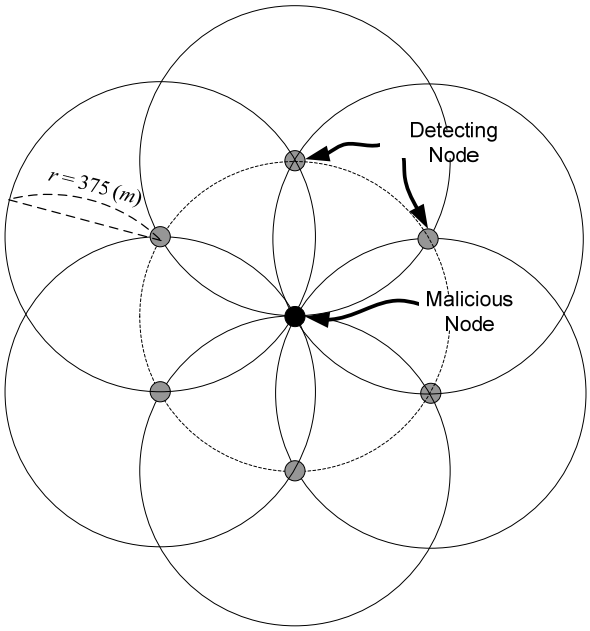

Figure 11. Number of detecting nodes needed per malicious node at 0 (*m/s*).

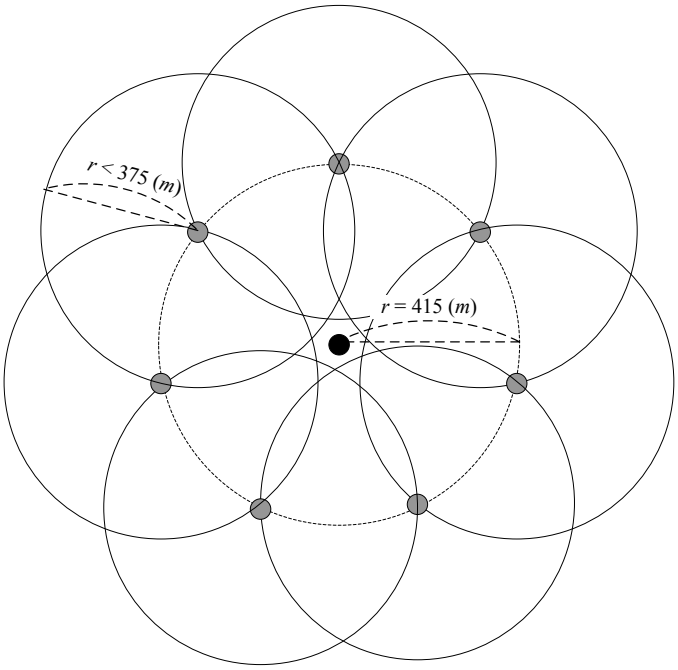

Figure 12. Number of detecting nodes needed per malicious node at 20 (*m/s*).

6. REFERENCES

[1] R. Bagrodia, R. Meyer, M. Takai, Y. A. Chen, X. Zeng, J. Martin H. Y. Song. Parsec: A Parallel Simulation Environment for Complex Systems. IEEE Computer. Vol 31, Issue 10, pp. 77 – 85. Oct 1998.

[2] S. Buchegger and J. L. Boudec. Performance Analysis of the CONFIDANT Protocol: Cooperation of Nodes – Fairness in Dynamic Ad Hoc Networks. In Proceedings of IEEE/ACM MobiHoc, Lausanne, CH, June 2002.

[3] L. Buttyan and J. Hubaux. Enforcing Service Availability in Mobile Ad Hoc WANs. In Proceedings of IEEE/ACM MobiHoc, Boston, MA, USA, August 2000.

[4] H. Fubler, J. Widmer, M. Kasemann, M. Mauve, and H. Hartenstein. Contention-Based Forwarding for Mobile Ad-Hoc Networks, Elsevier's Ad-Hoc Networks, Vol 1, no 4, pp. 351-369, 2003.

[5] A. H. Ho, A. Aved, and K. A. Hua. "A Novel Broadcast Technique for High-Density Ad Hoc Networks," Proceedings of International Wireless Communications and Mobile Computing Conference (IWCMC 2006), Vancouver, Canada. July 3-6, 2006. pp. 425 – 430.

[6] A. H. Ho, Yao H. Ho, and Kien A. Hua. "A Connectionless Approach to Mobile Ad Hoc Network in Street Environments." Proceedings of IEEE Intelligent Vehicles Symposium (IV 2005). Nevada, USA. June 2005.

[7] Y.H. Ho, A.H. Ho, K.A. Hua, and G.L. Hamza-Lup. A Connectionless Approach to Mobile Ad hoc Networks. Proc. of Ninth International Symposium on Computers and Communications (ISCC), Vol 1, pp. 188-195, Alexandria, Egypt, 2004.

[8] Y. H. Ho, Ai Hua Ho, Georgiana L. Hamza-Lup and Kien A. Hua, "Cooperation Enforcement in Vehicular Networks," the IEEE International Conference on Communication Theory, Reliability, and Quality of Service (CTRQ) 2008 (as one of the best papers).

[9] N. Jiang, S. Sheu, K. A. Hua, and O. Ozyer. A Finite-State Model Scheme for Efficient Cooperation Enforcement in Mobile Ad Hoc Netowkrs. In Proceedings 11th International Conference on Parallel and Distributed System, Fukuoka, Japan, 2005.

[10] B. Karp and H. T. Kung. GPSR: Greedy Perimeter Stateless Routing for Wireless Networks. In Proc. of MOBICOM '00, page 243-254, Boston, MA, U.S.A., August 2000.

[11] S. Marti, T.J. Giuli, K. Lai, and M. Baker. Mitigating Routing Misbehavior in Mobile Ad Hoc Networks. In Proceedings of MOBICOM 2000, page 255-265, 2000.

[12] P. Michiardi and R. Molva. CORE: A Collaborative Reputation Mechanism to Enforce Node Cooperation in Mobile Ad Hoc Network. 6th IFIP Conference on Security Communications and Multimedia (CMS 2002), Portoroz, Slovenia, 2002.

[13] A. Pfitzmann, B. Pfitzmann, M. Schunter, and M. Waidner. Trusting Mobile User Device and Security Modules. In Computer, pp. 61-68. IEEE, February 1997.

[14] X. Zeng, R. Bagrodia, and M. Gerla "GloMoSim: a library for parallel simulation of large-scale wireless network," Proc. of the 12th workshop on Parallel and distributed simulation, pp. 154-161, May 1998, Banff, Alberta, Canada.