Credit-based Routing using Game-theoretical Approaches in Vehicular Wireless Networks

Bo-Hao Huang
Department of Computer Science, National Taichung University of Education
Formre0721@gmail.com

Tsan-Pin Wang
Department of Computer Science, National Taichung University of Education
tpwang@mail.ntcu.edu.tw

ABSTRACT

In vehicular wireless networks, network coding enhances information transmission efficiency. However, the major methods of packet transmission in vehicular networks do not consider the impact of network coding on routing protocols. Moreover, selfish behavior can result in serious performance degradation. Therefore, we propose Credit-based Routing using Game-theoretical Approaches (CBRGA) against the selfish behavior in Vehicular Wireless Networks. CBRGA builds a forwarding game. In this game, every node has its trust value. Based on the trust value, nodes request other nodes to forward packets. Moreover, we utilize the coding rate, loading, and trust value as a cost function to select the routing paths. Finally, simulation results show that CBRGA can enforce cooperation among nodes.

Keywords： Vehicular wireless networks, Credit based routing, Game theory

摘要

在車載無線網路環境下，採用網路編碼的機制能夠有效增加信息傳送的效率，然而，目前主要的路由選擇機制並未考慮網路編碼對路由造成的影響，除此之外，自私行為也會嚴重影響整體表現，因此本文提出車載無線網路中應用賽局理論之信用基礎路由，運用賽局理論建立一個傳輸賽局，此賽局中，所有節點擁有一個信任值，由此信任值決定如何傳輸。並且我們結合編碼機會、負載及信任值作為成本函數來選擇路由。最後，模擬結果顯示，本方法能鼓勵節點之間趨於合作。

關鍵字：車載無線網路,信用基礎路由,賽局理論

1. INTRODUCTION

Network coding is an efficient packet transmission mechanism to increase the overall efficiency for wireless networks. However, network coding [1] might be unsuitable for low-power wireless sensors due to the high power consumption. In Vehicular Ad Hoc Networks (VANETs), every vehicle is equipped with a power generator. Therefore, network coding is able to enhance information transmission efficiency without considering power consumption in VANETs. On the other hand, the major routing protocols (ex. AODV) [2] in VANETs do not take advantage of network coding. When selfish nodes are requested to forward a packet, they may not send packets for other nodes to maximize their utility. Selfish behaviors can result in serious performance degradation. To solve these problems, we propose Credit-based Routing using Game-theoretical Approaches (CBRGA). Game Theory is a study of mathematical models of conflict and cooperation between intelligent rational decision-makers. We design a forwarding game to analyze the behaviors and strategies of the network nodes. In this game, every node has a trust value according to its behavior. A node will request other node to forward a packet based on the trust value. On the other hand, when a node is requested to forward a packet, the probability of packet transmission is based on the trust value of source. First, we

determine a threshold of the trust value. Furthermore, nodes will reject to help a node which has a trust value under the threshold to forward packets. Therefore, selfish nodes may pay more cost for their selfish behavior. We utilize the coding rate, loading, and trust value as a cost function. We use the cost function to calculate the cost for each routing path. A path with a lower cost has a higher probability to be selected. We designed and performed a simulation using Network Simulator 2 (NS2). The simulation results showed that CBRGA can not only encourage cooperation between nodes but also increase the chance of network coding.

2. RELATED WORK

2.1 AODV route discovery

When a node wishes to send a packet to some destination, it will check its routing table to determine whether it has a current route to the destination or not. If a route exists, source will forward the packet to that route; if no routes exist, it will initiate a route discovery process. AODV route discovery is based on forwarding route request (RREQ) and responding route reply (RREP). Source node forwards RREQ packets to its neighbor via flooding, each node receiving RREQ will set up a reverse route and forwards RREQ to its neighbors before arrival to the destination. After receiving RREQ, the destination node will send route reply (RREP) from the reverse route. An intermediate node may also send RREP if it knows a more recent path than the one previously known to source node. A node may receive multiple RREP from more than one neighbor. The node will forward the first RREP it had received. If it gets another RREP, it will only forward the RREP that has largest destination sequence number or a smaller hop counts. Source can begin data transmission upon receiving the first RREP.

2.2 Game theory

Game theory is a study of mathematical models of conflict and cooperation between intelligent rational decision-makers. Game theory applies to a wide range of behavioral relations, for example, economics, political science, and psychology, computer science, and biology. There have been many propose for wireless networks [3]-[7]. A regular game contains player, strategy, and utility. Players always select a strategy not only selfish but also rationality. All participants will consider other participants' possible decision, and try to maximize their utility as far as possible. Game theory provides a mathematical model to analyze the behaviors and strategies between Players.

2.2.1 Game Theory base Load-Balancing Routing with Cooperation stimulation (GBLBR) [3]

GBLBR is presented for delay sensitive service in wireless Ad hoc networks. The service has strict requirements for delay that must not exceed the value of the limit. The traditional routing protocols establish routes according to the shortest hop-counts. Therefore, large numbers of packets were concentrated in certain nodes. It may lead to serious delay problems.

GBLBR still select a route on the basis of the shortest hop-counts. The difference is that GBLBR will select multiple paths and calculate the delay utility function for every possible path. The flow for each path will be allocated according to the utility function and load capability of different paths. Therefore, GBLBR can minimize the average delay for each packet.

GBLBR's another key point is Cooperation Stimulation. Cooperation Stimulation strategy is introduced to enforce cooperation among nodes. It presents a new parameter γ, where γ is equal to Ndelivery / Nrequest. Nrequest stands for the total number of packets requested for sending by the neighbor nodes, while Ndelivery represents the actual number of packets forwarded by the node. Higher γ represents that the node is more cooperating. When the source node chooses the next hop node, it will consider both the next hop node's service rate and γ. On the other hand, when a node has a packet to send, its neighbor nodes will checking its γ and forward its packet with probability γ. That will actually induce a lager packet delay for a selfish node. On the basis of Game Theory, the selfish node will change its scheme to obtain better service after a certain time.

2.2.2 Attack and Flee: Game-Theory-Based Analysis on Interactions among Nodes in MANETs [5]

In this paper, the author present that there are malicious nodes exists in MANETs.

Malicious nodes attack other regular nodes, make regular nodes to waste resources, and interfere with operation of the whole network to get its own benefit. Moreover, malicious nodes are able to flee to a new location after an attack. Furthermore, this paper analyzes interactions among regular nodes and malicious nodes in MANETs. The authors observe the node's utility changes with different strategies. The authors propose a dynamic Bayesian game. In this game, regular nodes will change their beliefs according to opponents' behavior, and malicious nodes will assess their risk of being caught to decide whether to flee. This paper analyzes the costs and benefits between the nodes of different action to find out the best strategy for every node. Furthermore, this paper not only analyzes the utility of regular nodes, but also analyzes the utility of malicious nodes. It is found that malicious nodes always maintain advantage strategies because of the flee strategy. Therefore, the author proposed several methods to prevent malicious nodes to escape.

3. Credit-based Routing using game theoretical approaches

3.1 Route selection

Network coding is a kind of technique that relay node encodes more than one packet at a time to reduce the overall transmission times. Therefore, network coding is able to enhance information transmission efficiency. In a forwarding game, every participant will maximize their utility as far as possible. For that reason, if a relay node is willing to help other node forwarding packets, it will be willing to apply network coding. Consequently, our purpose is to encourage cooperation between nodes which select the path with the highest coding opportunity.

If a node wants to decode a packet that is encoded by n packets, it needs to have other n-1 packets of earlier received packets. As a result, when a node which has more packets that were earlier received, it may have higher opportunity to take advantage of network coding. However, these nodes may have higher loading at the same time. On the other hand, we need to avoid selecting a node which is a selfish node because the selfish behavior can result in serious performance. We utilize the number of earlier received packets, loading and the probability of being selfish node as a cost function. A node will establish routing path according to the cost function. A path has higher cost with lower probability to be selected.

3.1.1 Modification of AODV RREP

We modify the packet format of the AODV routing protocol, and add three extra parameters (Distrust, Loading, Packets) in route reply packets. All three parameters' maximum value is normalized to 1, and the minimum value is 0. The value above 1 is set to 1, while the value under 0 is set to 0. Distrust value is referring to as Distrust. Loading value is the percentage in the waiting queue. Packets value is refer to the number of earlier received packets, if there is one packet, the value will set to 0.1. If there are ten or more packets, the value will set to 1.

When a relay node receives a RREP packet in the traditional AODV, it will compare its sequence number and RREP's sequence number first. If RREP's sequence number is higher than the earlier received one, it represents a new route discovery. The node will update the route and forward the RREP. If RREP's sequence number is equal to its sequence number, then it compares the hop-counts value. If RREP's hop-counts value is lower, it represents that this is a shorter route. Then, the node will update the route and forward the RREP. In other cases, the node will drop the packet. Relay node still compares the sequence numbers first in our method. However, if the sequence number is the same, it will compare the cost of path in its route table and RREP's route. The cost is **c**alculated by the cost function. If the cost of RREP's route is lower, relay node will update the route table and forward the packet. After source node receives the RREP, it will start a ten millisecond timer. Before the timer expired, source node records all routes from the received RREP. Source node sets up a route with the probability which is equal to the proportion of every route's cost. The remainder routes are saved as backup routes.

3.1.2 Cost calculation and route selection

When relay node receives RREP, and the sequence number is equal to itself, it needs to compare the cost of route in the route table and the cost of path in RREP, and then forward the lower one. We propose a cost function which

includes the number of earlier received packets (Packets), loading of node (Loading) and the probability of being selfish node (Distrust) as a cost function. Lower loading value and Distrust value is better than a higher one. Note that, Packets value is a positive integer. Therefore, we need to normalize Packets value. We set nodes preserve at most ten packets which were earlier received. Nodes will drop the oldest packet every 100 milliseconds or when received a new packet but the queue is full.

Cost = WA * Packets + WB * Loading + WC * Distrust,where WA + WB + WC = 1

In the cost function, WA, WB and WC represent the proportion of Trust, Loading, and Packets, respectively. We can modify the weight according to different applications. The default values of all weight were one third. If there have more than one route, node will set up a route with the probability which is equal to the proportion between the costs of every route. The remainder routes are saved as backup routes. If a node needs to resend a packet, it will use the route with a lower cost. If there is any selfish node or overloading node in the route, it may lead to packet lost. Therefore, we set thresholds for Distrust and Loading. If Distrust or Loading of route is higher than the thresholds, relay node will not forward the RREP packet. We can modify the threshold value according to different applications. The default thresholds were set to 0.5.

Figure 1 shows that AODV simply selects the shortest path (D->F1->S). But in Figure 2, CBRGA applies several parameters such like distrust (DT), packets (P) and loading (L) to calculate the cost of a path, In this case, we set WA equal to 0.6, WB and WC equal to 0.2 to get a trustable path. Then CBRGA selects the path with a lower cost (D->F3->F2->S).

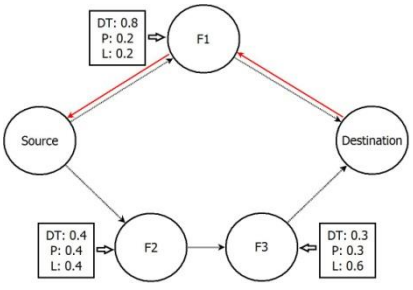

Figure 1 AODV route discovery

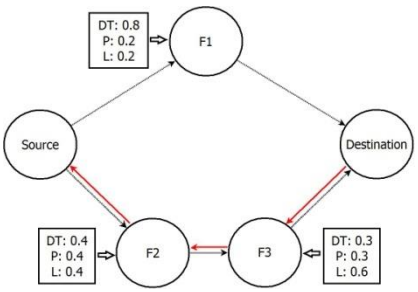

Figure 2 CBRGA route discovery

3.2 Game mechanism design

A regular game contains participants of game (Player), decisions of players (Strategy), and player's playoff from strategy (utility). We propose a forwarding game on the basis of game theory. The forwarding game consists of the following components.

Player：The players in this game are all nodes in the networks. There are two kinds of identity, Requester and Forwarder in the forwarding game. A node which requests neighbor node to forward its packet is called a Requester. On the other hand, a node which is requested to forward a packet is called a Forwarder. A node can be a Requester and a Forwarder at the same time.

Strategy：Nodes own different identity will have different strategies. Strategies of a Requester are selecting a route to send packets. Strategies of a Forwarder are whether to help forwarding packets or not. If a Forwarder helps forwarding packets, Requester will decrease the value of the Forwarder's Distrust as a reward. Otherwise, it increases the value of the Forwarder's Distrust. Forwarder will make decision by Requester's Distrust.

Utility: Utility in this game is Distrust exchanging between nodes. In this game, nodes will decide whether help others to forward packets according to the Distrust value of source node. The probability of dropping packets is the same as Distrust value. Nodes can help other nodes to forwarding packets to get lower Distrust value. Nodes pay some energy as cost to get higher packet delivery rates as playoff. If a node would like to get higher packet delivery

rates, it must help other nodes more often. On the other hand, a selfish node will be difficult to get helped from other nodes and get lower packet delivery rates. That is, we encourage nodes to cooperation.

3.2.1 Distrust design

Nodes use Distrust value to determine whether trust their neighbors or not. Every node will maintain a Distrust table to record values of Distrust. Distrust values depends on the neighbor node's behavior. If a neighbor node takes a cooperation operation, node will decrease Distrust value for that node. On the other hand, if a neighbor node takes a non-cooperation operation, node will increase Distrust value for that neighbor. We define the maximum value of Distrust value equal to 1, and maximum value is equal to 0.

We have two kind of Distrust design model. First is changing in certain number, and nodes have a probability to drop packets the same as the value of Distrust. We set the default Distrust value equal to 0. Default Distrust value represent the trust level for a new arrival. If a neighbor node takes a cooperation operation, node will decrease Distrust value 0.1 for that node. If a neighbor node takes a non cooperation operation, node will increase Distrust value 0.1 for that neighbor. We set a threshold of Distrust value equal to 0.5. Nodes will add its neighbor with Distrust higher than the threshold into Banlist. If a node received a request from a node in Banlist, it will drop the packet.

Second method is to set Distrust on different levels. We set Distrust with six levels. If a node which has Distrust level with level one or level two, the forwarder will never drop packets from that node. If a node which has Distrust level with level three, the probability of forwarder dropping packets is one hundred to twenty five, level four is one hundred to fifty, level five is one hundred to seventy five, and level six is one hundred percent. While a node loss packets two times continuously, its Distrust level will be increased with one level. On the other hand, if a node which would like to decrease its Distrust level, it needs to cooperate two times continuously in level two, four times in level three, eight times in level four and so on. Increasing Distrust level is more easily than decreasing Distrust level. If a node would like to get higher packet delivery rates, it must be more cooperation to increase its Distrust level. We expect the Distrust designing can encourage nodes to be more cooperation.

3.2.2 Distrust management

There are two kinds of Distrust management model. First is Distributed Distrust management. Every node has a Distrust table to save Distrust values of 1-hop neighbors. Nodes do not exchange Distrust values normally. When a node starts a route discovery, it will need to know the probability of successful transmission for every path to select a trustworthy path. Source will request nodes which received RREP add its Distrust to next hop in RREP. The advantage of distributed Distrust management is that every node maintains its Distrust table by itself. Therefore, there are unusually leaded a node overloading. However, the weak point is that there are several Distrust communications between nodes. It may be an inefficient way to find out mobile selfish nodes. The other way is the centralized Distrust management. There will be a trustable node to maintain every node's Distrust value in an area. Every node reports its neighbor's behaviors, and download the Distrust that need to be used from the node. In this way, the centralized node achieved high loading, but effective to discover mobile selfish nodes.

3.2.3 Distrust exchange

When node A requests its neighbor node B to help forwarding an RREQ or a data packet, it will start a ten millisecond timer. After node B received the request, it will determine whether help or not according to its Distrust to node A. If node B decides to help, it will broadcast the packet. If node A receives the packet before the timer expired, it will know that node B is cooperating. Then, node A decreases the Distrust value as reward to node B. However, if node A does not receive the broadcast from node B before timer expired, it will consider that is a selfish behavior. Then, node A increases the Distrust value to punish node B.

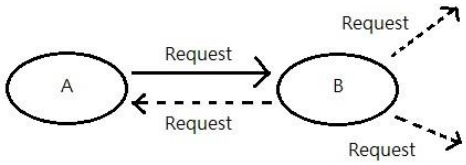

Figure 3 Node A requests node B. Node B broadcast the packet with cooperation strategy.

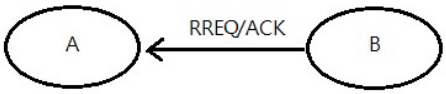

Figure 4 Node A receives the RREQ or ACK from node B

3.2.4 Banlist model

When a node's Distrust value to another node is lower than threshold, it will add that node into Banlist. A node will reject another node which is in Banlist to forward packets. On the other hand, in our route selection method, source will not select a route which includes any node in Banlist. Therefore, a node in Banlist has never chance to be forgiven by help others. To solve that, we give the node which is willing to cooperate with others a chance to be removed from Banlist. When a node is added in Banlist, it will be banned within 10 seconds. In the ban time, the node will never be requested, so have no chance to be forgiven. However, if source find that there are no any trustworthy route, it will select the path which have some nodes in Banlist but not in ban time. Therefore, if the node is willing to cooperate, it will get chance to decrease its own Distrust, even can be removed from Banlist. However, if the node is uncooperative again, it will have double ban time, first is ten seconds, second is twenty seconds, third is forty seconds and go on.

3.2.5 Analysis of default Distrust value

In this game, Distrust value is the same as the probability of forwarding packets. Therefore, how to determine the default Distrust value of nodes will be an important issue. In a game with high default value of Distrust, a new-arrival node has higher chance to be help forwarding packets, and higher chance to be asked to help forwarding packets. If this node is a regular node, high default value of Distrust can help it integrate into the new environment more easily. However, if this node is a selfish node, it can get high payoff by high default value of Distrust and reject other nodes packets. Moreover, after distrusted by other nodes, it will flee to another cluster to get a new distrust value. On the other hand, in a game of low default value of Distrust, a new-arrival regular node need more time to integrate into the new environment. It will be unsuitable for high-speed environment. However, a selfish node that is not cooperating will get low playoff too. In Section 4, we set the default value of Distrust to 0, and we will try other default value in the future.

3.3 Comparisons with the traditional strategies

In the forwarding game, every rational node will hope be helped by other nodes, but unwilling to help other nodes forwarding packets. This game is similar with the social dilemma of infinitely repeated games. Therefore, we select several strategies of the traditional game theory which can apply to the forwarding game. We analyze the efficiency between CBRGA and traditional strategies.

Tit for tat: If a player uses tit for tat strategy, he will select cooperation strategy in the first round. Selecting decisions depend on the choice of the opponent after the first round. If the opponent selects cooperation in this round, the player will select cooperation in the next round. If the opponent selects rejecting in this round, the player will select rejecting in the next round too. Applying to the forwarding game, if a node helps source to forward packets, source will help that node to forward packets next time. If a node rejects source to forward packets, source will reject that node to forward packets next time.

The grim trigger: If the opponent selects rejecting strategy once, the player will select rejecting that opponent afterward. In forwarding game, if a node which be rejected forwarding a packet once, it will reject to help that node forwarding packets afterward.

A tit for two tats/two tits for a tat: The

same as the grim trigger, but there is the opportunity to repent. A tit for two tats is rejecting once after be rejected twice. Two tits for two tats are rejecting twice after be rejected twice. In forwarding game, a node will reject forwarding specific times after be rejected specific times.

In Section 4, we simulate nodes using tit for tat as strategy, and we will simulate other strategies in the future.

4. Simulation results

Table 1 Experimental parameters

Mobility model	Manhattan
Area of model	500 m*500 m
MAC protocol	IEEE 802.11
Simulation time	120 s
Amount of vehicles (n)	40
Avg. speed (m/s)	20 m/s
Range of communication	100 m

In the simulation, we have three selfish scenarios that there are ten percent, twenty present, or thirty percent selfish nodes (SN) in the environments. Selfish nodes always reject to cooperate. Other regular nodes (RN) use different strategies to be a relay node. First is cooperation (co), the nodes will help other nodes sending packets in any case. Second is tit for tat (tft), if the nodes are rejected by other nodes, it will reject that nodes once as punishment. Third is dropping packets according to Distrust (Dt), where the nodes will drop packets with the probability equal to the sender's Distrust. We set all nodes' initial Distrust to zero. If a node helps other nodes to forward packets, its Distrust will decrease 0.1. On the other hand, if a node rejects other nodes, its Distrust will increase 0.1. We assume that the centralized Distrust model was used in this simulation. That is, nodes will know anyone rejecting others. We assume that a node will not increase Distrust when sender's Distrust is higher than 0.5. It represents that rejecting selfish node will neither increase Distrust nor be punished.

In Figure 5, we compare the impact of different strategies of relay nodes on packet delivery rates. There are four strategies, cooperation, tit for tat, and dropping packets by Distrust. The routing protocol is AODV. There are ten percent, twenty percent, or thirty percent selfish nodes in forty nodes. Simulation results show that tit for tat decreases selfish node's packet delivery rates but not decrease regular node's at the same time. Dropping packets by Distrust decreases selfish node's packet delivery rates. Node dropping packets by Distrust reduces selfish nodes' packet delivery rates. However, regular nodes' packet delivery rates will be reduced slightly because of misjudgments.

In Figure 6, we show the impact of sender's route selecting protocols, including AODV (A) and CBRGA (C) on packet delivery rates. Relay nodes will drop packets according to Distrust. There are ten percent, twenty percent, or thirty percent selfish nodes in forty nodes. Simulation results show that CBRGA improves the packet delivery rates, especially in higher proportion of selfish nodes.

Figure 7 shows the impact of sender's route selecting protocols, including AODV and CBRGA on the packet delay. Relay nodes will drop packets according to Distrust. There are ten percent, twenty percent, or thirty percent selfish nodes in forty nodes. Simulation results show that the selfish node's average delay is always lower than the regular node's average delay and average delay of AODV routing is always lower than CBRGA. However, we just calculate average delay of received packets in this case. Selfish node and AODV routing might lost more packets. Therefore, in Figure 8, we set the delay of packets which are lost equal to the simulation time, i.e., 120 seconds. Simulation results show that the regular node's average delay is always lower than the selfish node's average delay. CBRGA reduces the average delay of packets slightly from that in AODV.

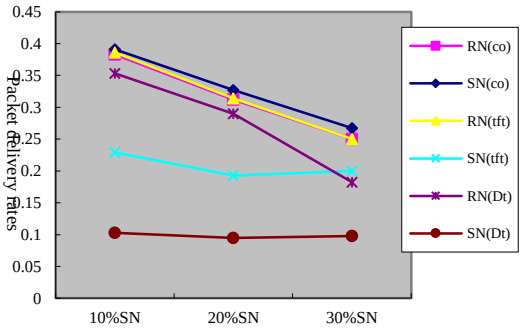

Figure 5 Impact of relay node's strategies on packet delivery rates

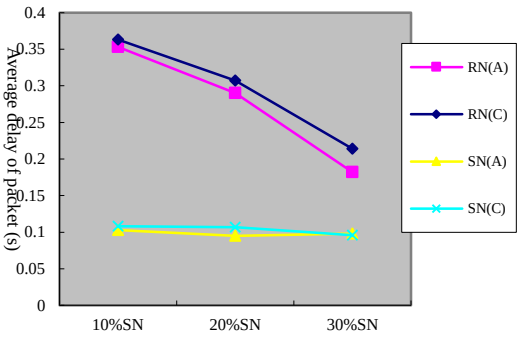

Figure 6 Impact of sender's route selecting protocol on packet delivery rates

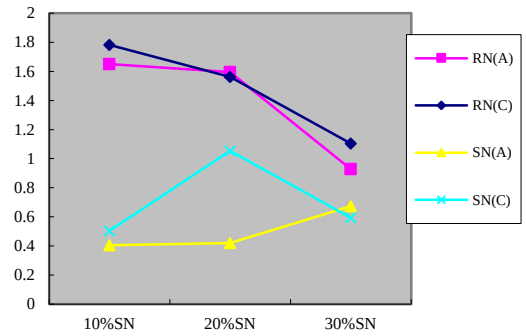

Figure 7 Impact of sender's route selecting protocol on average delays (received packets only)

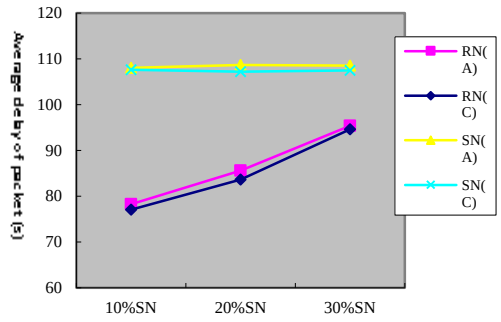

Figure 8 Impact of sender's route selecting protocol on average delay (delay of lost packets is set to 120 seconds)

5. Conclusion and future work

In this paper, we propose a strategy for Credit-based Routing using Game-theoretical Approaches against the selfish behavior in Vehicular Wireless Networks. We design a forwarding game to analyze the behaviors and strategies of the network nodes. In this game, every node has its trust value. Based on the trust value, nodes request other nodes to forwarding packets. Moreover, we utilize the coding rate, loading, and trust value as a cost function to select the routing paths. Finally, extensive simulation results show that CBGRA can enforce cooperation among nodes. In the future, we will extend our work to recover the dropped packet and show the performance impact with/without retransmissions.

Acknowledgement

This work was supported in part by the Ministry of Science and Technology under grant No. MOST 103-2221-E-142 -002.

Reference

[1] Li, Shuo-Yen Robert, Raymond W. Yeung, and Ning Cai. "Linear network coding." Information Theory, IEEE Transactions on 49.2 (2003): 371-381.

[2] Perkins, Charles, Elizabeth Belding-Royer, and Samir Das. Ad hoc on-demand distance vector (AODV) routing. No. RFC 3561. 2003.

[3] TIAN Hui, JIANG Fan, CHENG Weijun, A Game Theory based Load-Balancing Routing with Cooperation Stimulation for Wireless Ad hoc Networks, 2009 11th IEEE International Conference on High Performance Computing and Communications

[4] Miao Jiang and Paul A.S. Ward, A Cooperative Game-Theory Model for Bandwidth Allocation in Multi-hop Wireless Networks, 2011 IEEE 7th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob).

[5] Feng Li, Member, IEEE, Yinying Yang, Student Member, IEEE, and Jie Wu, Fellow, IEEE, "Attack and Flee: Game-Theory-Based Analysis on Interactions among Nodes in MANETs," IEEE Transactions on Systems, Man, And Cybernetics—Part B: Cybernetics, Vol. 40, No. 3, June 2010

[6] Hiteshkumar Nimbark, Paresh Kotak and Nikesh Shah, "Intelligent computer networks:

A Game Theoretic Approach to Compute the Traffic Equilibrium of Various Routing Schemes for multimedia applications in wireless networks,"2012 International Conference on Communication Systems and Network Technologies.

[7] Yu Sun, Guan Wang, Jianwei Liu, "The Design of Lightweight Secure AODV Protocol Based on Game Theory and Cryptology," 2010 IEEE International Conference on Information Theory and Information Security