Junction-assisted Routing Protocol for Vehicular Ad Hoc Networks in City Environments

PangWei*, Dong-Won Kum* *Regular Members*, You-Ze Cho* *Lifelong Member*

요 약

ABSTRACT

Vehicular Ad-Hoc Networks (VANETs) using inter-vehicle communication can potentially enhance traffic safety and facilitate many vehicular applications. Therefore, this paper proposes an inter-vehicle routing protocol called Junction-Assisted Routing (JAR) that uses fixed junction nodes to create the routing paths for VANETs in city environments. JAR is a proactive routing protocol that uses the Expected Transmission Count (ETC) for the road segment between two neighbor junctions as the routing paths between junction nodes. Simulation results showed that the proposed JAR protocol could outperform existing routing protocols in terms of the packet delivery ratio and average packet delay.

Key Words : Vehicular Ad-hoc Networks, Junction-assisted Routing, Ad hoc Routing, Greedy Forwarding Approach

Ⅰ. Introduction

Vehicular Ad-Hoc Networks (VANETs) are an application of Mobile Ad Hoc Networks (MANETs). However, while the existing routing protocols for MANETs work well in scenarios where the nodes are distributed evenly and moving freely in an open area, the specific constraints of VANETs, such as radio obstacles in city environments, mean these routing protocols can not be directly applied to vehicular environments. The route instability in a vehicular environments is the main problem with the existing MANET protocols, such as Ad hoc On-demand Distance Vector routing (AODV) [1], and Dynamic Source Routing (DSR)[2]. Therefore, variants of stateless geographic routing protocols, such as Greedy-Face-Greedy (GFG)[3] and Greedy Perimeter Stateless Routing (GPSR)[4], would seem to offer a possible solutions.

As a result, a number of junction-based routing protocols have already been proposed for VANETs [5-8], where the source node supplies each packet with a route vector that includes a list of junctions and their geographic locations through which the packets must pass. However, these protocols also suffer from various problems. For example, Geographic Source Routing (GSR)[5] can not ensure that there are enough vehicles on the street to provide connectivity between two involved junctions. While Anchor-based Street and Traffic Aware Routing (A-STAR)[6] tried to solve this issue using bus route information to indicate the number of vehicles on the street, historical data on bus route information is not an accurate indicator of the current road traffic conditions. Furthermore, the junction nodes in [5-8] are not fixed, and this

※ This research was supported by the National Research Foundation of Korea (NRF) funded by the Korean Government (2009-0076947).
* 경북대학교 전자전기컴퓨터학부 (80kumsy, yzcho@ee.knu.ac.kr)
논문번호 : KICS2010-03-136, 접수일자 : 2010년 3월 31일, 최종논문접수일자 : 2010년 8월 31일

absence of junction nodes affects all the protocols, since packets cannot be forwarded to another road segment.

Accordingly, this paper proposes an inter-vehicle ad hoc routing protocol called Junction-Assisted Routing (JAR) that uses fixed junction nodes to create the routing paths for VANETs in city environments. JAR selects a junction path based on the Expected Transmission Count (ETC) for each road segment. The ETC is estimated based on the periodic exchange of probe packets between neighboring junction nodes. The intermediate vehicles between junctions are used to forward the probe packets based on the greedy forwarding approach. As every junction node maintains ETCs for the whole network, the routing path with the lowest ETCs to the destination junction is selected to forward the data packet using the Dijkstra algorithm.

The rest of this paper is organized as follows. Section II describes related works, and Section III then introduces the detailed of the proposed JAR protocol. Section IV evaluates the performance of the proposed scheme using simulation, and some final conclusions are presented in Section V.

II. Related works

Traditional topology-based routing protocols for MANETs due to the frequent disconnections experienced by VANETs that cause high data loss rates. Another type of routing protocols for MANETs is position based routing protocols that share geographical position information, rather than relying on unstable topology information. In [9], the authors compare a position-based protocol with the DSR topology-based routing protocol in VANETs, and found that the position-based routing protocol performed better than DSR. Meanwhile, in the literature, GPSR is the most typical position-based protocols. Yet, GPSR suffers from network disconnection and paths that are too long, as stated in [5].

In [5], a junction-based routing protocol called GSR is proposed that combines position-based routing with topological knowledge. Based on the knowledge of the node positions and a layout map, the source node is able to calculate the shortest path using Dijkstra's algorithm. To reach the destination the source node calculates the sequence of junctions on the map through which the packet must traverse. Simulation results also show that GSR outperforms topology-based approaches (DSR and AODV) in a city environment. However, GSR can not ensure that there are enough vehicles on the street to provide connectivity between the two involved junctions.

Thus, to resolve the problems of GSR, another junction-based routing scheme was proposed called Anchor-based Street and Traffic Aware Routing (A-STAR)[6], which was designed specifically for city environments. This protocol adopts a junction-based routing approach similar to GSR. However, unique to A-STAR is the usage of information on the city bus routes to identify a junction path with a high connectivity for packet delivery. However, as mentioned in the introduction, the use of historical data does not present an accurate view of the current road conditions.

Similar to other junction-based routing protocols with spatial awareness, Greedy Traffic Aware Routing (GyTAR)[7] routes packets between road junctions towards the destination. Yet, the difference is that the junctions are chosen on-the-fly, taking into account the density of vehicles in between the junctions and the range to the next junction. This is performed based on the assumption that an on-road traffic estimation of all vehicles can be realized using a simple distributed mechanism. However, obtaining the real-time traffic density through a simple distributed mechanism is a challenging task.

Landmark Overlays for Urban Vehicular Routing Environments (LOUVRE)[8] classifies protocols like GSR, A-STAR and GyTAR as geo-reactive overlay routing. LOUVRE is a distributed traffic density estimation scheme that prunes disconnected road segments. LOUVRE also emphasizes the importance of junction nodes, as the absence of junction nodes affects all junction-based routing protocols, since no packets can be forwarded to another road segment.

Thus, from the existing junction-based routing protocols, selecting the correct sequence of junction to provide a high connectivity from the source to the destination is clearly the main issue. In other words, the key issue is how to make sure there are enough vehicles on the street to provide connectivity between two successive junctions. The next section provides a detailed description of the proposed JAR scheme.

Ⅲ. JAR: Junction-Assisted Routing

An inter-vehicle ad hoc routing protocol called JAR is proposed for VANETs in city environments. JAR selects a junction path based on the ETC for every road segment as the routing metric. The JAR protocol consists of two parts: 1) Each junction node sends probe packets to its neighbor junctions periodically to estimate the ETC for the road segment. Since not all junctions are able to communicate directly with their neighbor junctions due to limited power, intermediate vehicles on the road segments between the junctions are used to forward the probe packets. 2) Based on the ETCs measured for all the road segments, each junction node computes the sequence of junction nodes with the lowest ETCs on the paths to other junction nodes using the Dijkstra algorithm.

3.1 Assumptions

It is assumed that every vehicle can determine its current location using GPS navigation system that includes digital street-level maps. The information on the current geographical position of the destination vehicle is also assumed to be provided by a location service, like RLS (Reactive Location Service)[10]. In addition, a fixed junction node is set at each junction, otherwise no packets cannot be forwarded onto another road segment. Plus, each vehicle and junction node can determine the position of other junctions using a pre-loaded digital map. In JAR, all the control and data packets are transmitted using the greedy forwarding approach through intermediate vehicles on the road segments.

3.2 Estimation of the expected transmission count

The first part of JAR involves estimating the ETC for a road segment, which is based on the proactive exchange of probe packets between neighboring junction nodes. Fig. 1 shows an example of how the ETC is measured for a road segment between two successive junctions. Probe packets are exchanged with the neighbor junction nodes using the greedy forwarding approach. In JAR, since the junction nodes are static, the ETC for the probe packets during a period of time can be used to reflect the current vehicle density between junctions. Plus, since the junction nodes are not usually able to communicate with each other directly, the intermediate vehicles between junctions are used to forward the probe packets using the greedy forwarding approach.

The notations used to estimate the expected transmission count are defined in Table I.

During period τ_1, J_n sends a probe packet to J_c periodically at the period τ_0. The total number of packets sent from J_n to J_c is S, which is be calculated based on $\frac{\tau_1}{\tau_0}$. The number of packets successfully received by J_c from J_n is Q_{nc}. The P_{nc} is then caculated as $P_{nc} = \frac{Q_{nc}}{S}$. Therefore, the ETC routing metric can be represented as $ETC = \frac{1}{P_{nc}}$. The computed ETC is then included in a link state packets that is sent by each junction node to all the other junction nodes in the network periodically at the period τ_1.

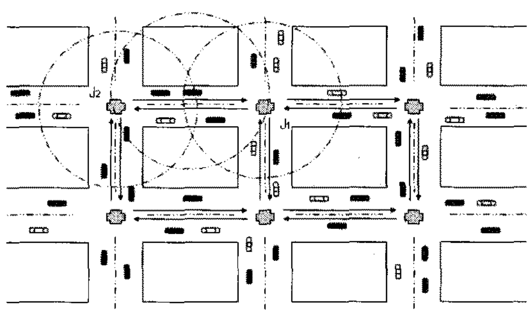


Fig. 1. Each junction node exchanges probe packets with its neighbor junctions

Table 1. Definition of parameter notations

J_c	current junction node
J_n	neighbor junction node of current junction
τ_0	period between sending of probe packets by junction nodes
τ_1	period used to calculate *ETC* between junctions and send a link state packets to other junction nodes
Q_{nc}	number of packets successfully received by J_c from J_n during period τ_1
P_{nc}	packet delivery ratio between J_n and J_c.
S	total number of packets sent from J_n to J_c during period τ_1
ETC	expected number of packets transmittable without error from J_n to J_c
ETC: Expected Transmission Count	

3.3 Path selection and data forwarding

The second part of JAR involves the selecting the sequence of junction nodes with the lowest ETCs among the paths to the destination junction node geographically closest to the destination vehicle.

Link state packets including the ETCs are periodically flooded to all the junction nodes in the network based on period τ_1. Each junction node then computes the junction paths to all the other junction nodes in the network using the Dijkstra algorithm based on the *ETC* routing metric.

When a source vehicle wants to send a data packet to a destination vehicle, it first needs to send the packet to the junction node. Therefore, the source vehicle sends a junction discovery packet to the nearby junction nodes using the greedy forwarding approach. When the nearby junction nodes receive the junction discovery packet, they send back a junction reply packet. Upon receiving the first junction reply packet from the nearest junction, the source vehicle then sends its data packet to this nearest junction node.

Once the junction node receives the data packet, it checks the routing cache and then finds the next junction node in the routing path to the destination junction node. Finally, the data packet is delivered to the destination vehicle through the destination junction node.

Ⅳ. Performance Evaluation

The proposed JAR protocol was implemented using ns-2 simulator. Meanwhile, the implementation of GSR and, A-STAR was based on [5, 6], respectively, and the GPSR simulation code was from [11].

4.1 Simulation environments

The experiment was based on a 1000 m ×1000 m area presented as a grid layout. Fig. 2 shows the map used for the simulation, and the setup parameters are listed in Table 2.

To model vehicle movement, the nodes were moved using VanetMobiSim[14], which focuses on

Table 2. Simulation Parameter Values

Parameters	Values
Simulation area	1000 m × 1000 m
Number of vehicles	150
Transmission range	250m
Simulation time	500 seconds
Vehicle velocity	30-50 km/h
CBR rate	0.1-1 packets/second
MAC protocol	IEEE 802.11 DCF
IEEE 802.11 data rate	1 Mbps
Data packet size	512 bytes
τ0	0.5 seconds
τ1	10 seconds

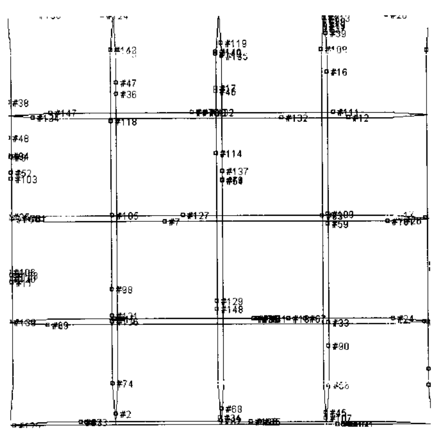


Fig. 2. Snapshot of the simulation setup area using the VanetMobisim.

vehicular mobility and features new realistic automotive motion models at both macroscopic and microscopic levels. Fig. 2 is a snapshot from VanetMobiSim.

GPSR is a basic position-based routing protocol, while GSR and A-STAR are improvement of GPSR for specific use in a city environment. Therefore, GPSR, GSR and A-STAR were used to compare with the proposed scheme.

4.2 Simulation results

To evaluate the simulation results, the following performance metrics were used.

Packet delivery ratio: ratio of data packets successfully delivered to their destinations.

Average packet delay: average packet delay for all data packets successfully delivered to their destinations.

Routing overhead: ratio of number of bytes for all control packets to number of bytes for all data packets successfully delivered to their destinations during the entire simulation.

4.2.1 Packet delivery ratio

As shown in Fig. 3, JAR achieved a higher packet delivery ratio than GSR, A-STAR, and GPSR. The two main reasons why JAR outperformed the other schemes were: First, the route path is selected using the ETC routing metric, which guarantees a higher packet delivery ratio to the destination. Second, the ETC routing metric is updated periodically, which ensures that the latest traffic condition is considered to calculate the route path.

The GSR protocol performed poorly, as the geographically shortest path still suffers from frequent network disconnections. When compared with GSR, A-STAR showed a higher packet delivery ratio as it estimates the connectivity using bus route information.

The GPSR protocol is stateless, which provides many advantages for routing data packets. However, if GPSR faces a network disconnection problem, it has to change to the perimeter forwarding, which routes the packet away from the destination. Moreover, due to the obstacles in city environments, network partitions occur frequently, which gave GPSR the lowest data delivery ratio.

4.2.2 Average packet delay

Fig. 4 shows the average packet delay according to the packet sending rate. As shown, the GPSR protocol with a perimeter recovery mode had a significantly higher end-to-end delay than GSR and JAR. This was because the greedy forwarding of GPSR causes the packets to encounter local maximum problems. Also, the main reason why GSR and A-STAR had a higher delay than JAR was the frequent occurrence of network disconnections, thereby increasing the average packet delay.

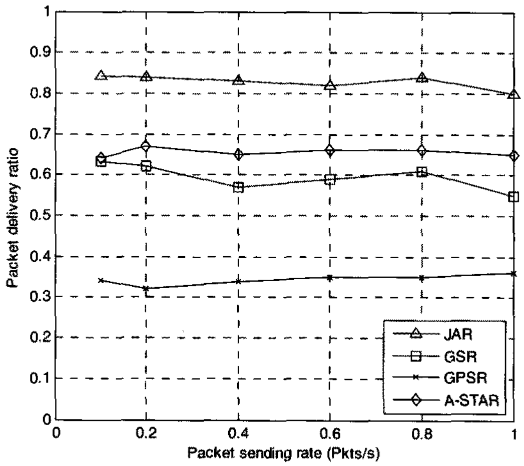


Fig. 3. Packet delivery ratio vs. packet sending rate

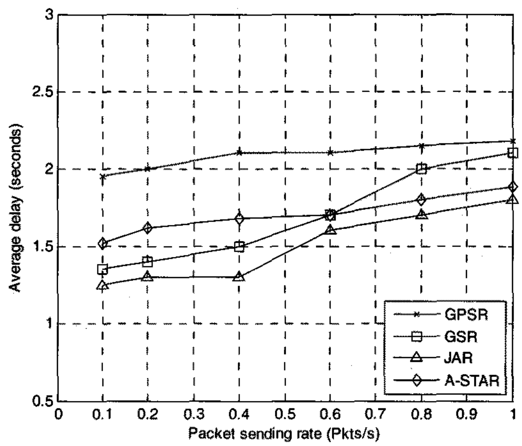


Fig. 4. Average packet delay vs. packet sending rate

4.2.3 Routing overhead

Fig. 5 shows that the routing overhead decreased for all the protocols when the packet sending rate increases. JAR has the highest routing overhead among all the schemes. The reason is that it incurs a certain overhead as transmitting the probe packets to estimate the packet delivery ratio between the junctions. However, JAR could significantly improve the packet delivery ratio and reduce the average packet delay. This is because it finds the higher reliable path than other schemes by using the ETC as routing metric. The results also showed that GPSR with perimeter recovery incurred a higher routing overhead than GSR, as the local maximum problems and longer recovery paths via perimeter routing require more data retransmissions to recover the data packets.

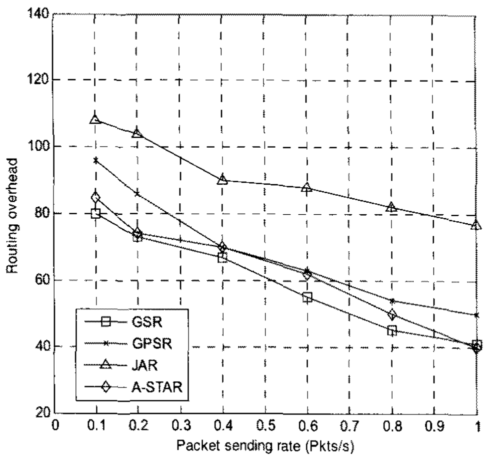


Fig. 5. Routing overhead vs. packet sending rate

V. Conclusion

This paper proposed a junction-based proactive routing protocol, called JAR, for vehicular ad hoc networks in city environments. JAR selects a junction path according to the ETC for each road segment as the routing metric, which is estimated based on the periodic exchange of probe packets between neighboring junction nodes. Thus, JAR can perform well in realistic vehicular environments including buildings and roads. The high packet delivery ratio in simulation results confirmed that JAR is ideally suited for VANETs in city environments.

References

〔1〕 C. E. Perkins and E. M. Royer, "Ad hoc on-demand distance vector routing," *in Proc. of MCSA*, Feb. 1999.

〔2〕 D. B. Johnson and D. A. Maltz, "Dynamic source routing in ad hoc wireless networks," *Mobile Computing*, Vol.353, No.5, pp.153-161, 1996.

〔3〕 P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia, "Routing with guaranteed delivery in ad hoc wireless networks," *ACM Wireless Networks*, Vol.7, No.6, pp.609-616, 2001.

〔4〕 B. Karp and H. T. Kung, "GPSR: greedy perimeter stateless routing for wireless networks," *in Proc. of MobiCom*, Aug. 2000.

〔5〕 C. Lochert, H. Hartenstein, J. Tian, H. Fußler, D. Hermann, and M. Mauve, "A routing strategy for vehicular ad hoc networks in city environments," *in Proc. of IVS*, June. 2003.

〔6〕 B.-C. Seet, G. Liu, B.-S. Lee, C. H. Foh, K. J. Wong, and K.-K. Lee, "ASTAR: A Mobile Ad Hoc Routing Strategy for Metropolis Vehicular Communications," *in Proc. of IFIP*, Dec. 2004.

〔7〕 M. Jerbi, S.-M. Senouci, Y. G.-Doudane, and R. Meraihi, "GyTAR: Improved Greedy Traffic Aware Routing protocol for Vehicular Ad hoc Networks in City Environments," *in Proc. of VANET*, Sep. 2006.

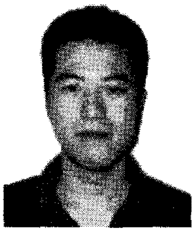
〔8〕 Kevin C. Lee, Michael Le, Jerome Harri, Mario Gerla, "LOUVRE: Landmark Overlays for Urban Vehicular Routing Environments" *in Proc. of WIVeC*, 2008.

〔9〕 H. FiiBler, M. Mauve, H. Hartenstein, M. Kasemann, and D. Vollmer, "Location-based routing for vehicular ad hoc networks," *in Proc. of Mobicom*, 2002.

〔10〕 M. Kasemann, H. Fiifler, H. Hartenstein and M. Mauve, "A reactive location service for mobile ad hoc Networks," Technical Report

TR-02-014, Nov. 2002.
〔11〕 Simulation code of "Greedy Perimeter Stateless Routing," Available from: http://www.icir.org/bkarp/gpsr/gpsr.html
〔12〕 J. H¨arri, F. Filali, C. Bonnet, and M. Fiore, "VanetMobiSim: generating realistic mobility patterns for VANETs," in Proc. of VANET, Sep. 2006.

팡 웨 이 (PangWei) 정회원



2008년 2월 중국 동북 대학교 전자정보공학과
2010년 8월 경북대학교 전자전기컴퓨터학부 석사.
2010년 9월~현재 미국 오클라호마주립대학교 박사과정
<관심분야> 차량 간 통신 네트워크

금 동 원 (Dong-Won Kum) 정회원

2003년 우송대학교 전자정보통신 공학과
2007년 경북대학교 전자공학과 석사
2007년~현재 경북대학교 전자전기컴퓨터학부 박사 과정
<관심분야> 차세대 이동네트워크, 무선 메쉬 네트워크, 차량 간 통신 네트워크

조 유 제 (You-Ze Cho) 종신회원

1982년 서울대학교 전자공학과
1983년 한국과학기술원 전자공학과 석사
1988년 한국과학기술원 전자공학과 박사
1989년~현재 경북대학교 전자전기컴퓨터학부 교수
1992년 8월~1994년 1월 Univ. of Toronto, 객원교수
2002년 2월~2003년 1월 미국 국립표준연구소 NIST), 객원 연구원
<관심분야> 차세대 이동네트워크, BcN, 무선 메쉬 네트워크, 센서 네트워크