A MAC Protocol for Delay-sensitive VANET Applications With Self-learning Contention Scheme

Celimuge Wu and Satoshi Ohzahata
University of Electro-Communications
Chofu-shi, Tokyo, Japan
Email: {clmg,ohzahata}@is.uec.ac.jp

Yusheng Ji
National Institute of Informatics
Tokyo, Japan
Email: kei@nii.ac.jp

Toshihiko Kato
University of Electro-Communications
Chofu-shi, Tokyo, Japan
Email: kato@is.uec.ac.jp

***Abstract*—Packet delivery ratio and end-to-end delay are the two most important metrics for vehicular ad hoc network applications. In this paper, we propose a MAC layer protocol which can provide a high packet delivery ratio, low end-to-end delay, and high fairness for various scenarios. The proposed protocol uses a Q-Learning algorithm to adjust the contention window size in order to provide an efficient channel access scheme for various network situations. The simulation results demonstrate the advantage of the proposed protocol over other alternatives.**

I. Introduction

Vehicular ad hoc networks (VANETs) have been attracting interest in recent years. In VANETs, a multi-hop communication is required for many applications including driving assistance systems, Internet access and the collection of road traffic information. Many VANET applications require a low end-to-end delay and high packet delivery ratio. In VANETs, since the network environment varies for different road types or road segments, designing an efficient MAC protocol is particularly important.

IEEE 802.11p [1] is an approved amendment to the IEEE 802.11 standard to provide wireless access to support Intelligent Transportation Systems (ITS) applications in vehicular networks. IEEE 802.11p uses DCF (distributed coordination function) or an enhanced variant of the DCF, called an enhanced distributed channel access function (EDCAF) to contend for transmissions. Each node senses the wireless medium before transmission. If the medium is busy, each node defers until the medium is determined to be idle for a period of time equal to DIFS/EIFS (distributed interframe space / extended interframe space). After this DIFS or EIFS medium idle time, the node generates a random backoff period for an additional deferral time before transmission. The backoff time is a random number which is drawn from the uniform distribution over the interval [0,CW] where CW is the current contention window. The CW is a value determined by aCWmin and aCWmax depending on the access category. According to the IEEE 802.11p, aCWmin is 15, and aCWmax is 1023. If multiple nodes choose the same backoff time, collisions occur. This occurs very frequently when the number of sender nodes is large. After detecting a failed transmission, each node doubles its contention window size to avoid further collisions. However, after a successful transmission, the contention window size is reverted to the minimum value. In VANETs, since the node density and data traffic pattern vary for different road types or road segments, the MAC protocol should be intelligent enough to address this issue. When the number of sender nodes is large, a relatively large CW size is needed to avoid unnecessary collisions. In contrast, when the network traffic load is small, a small CW size is required to access the wireless medium with a short delay.

There have been some protocols applying contention window adjustment [2]–[6]. Shi et al. [2] have proposed a mechanism that adjusts the contention window size by estimating the number of sender nodes in the network. The protocol predicts the future node number based on the current network status. Shi et al. [3] also have proposed a backoff algorithm which is based on the estimation about the number of active nodes. However, in VANETs, the data traffic patterns are difficult to estimate. Those in Refs. [4]–[6] are proposed for wireless LANs and therefore do not take account of multi-hop data communications. Since the traffic flows are distributed and difficult to predict in VANETs, an intelligent contention window adjustment mechanism is required.

When two sender nodes cannot hear from each other, the hidden terminal problem could happen. Due to a neighboring transmitter, a node could be wrongly prevented from sending packets, which is known as the exposed terminal problem. The hidden terminal and exposed terminal problem degrade the system performance especially for a multi-hop communication. Some studies [7], [8] use RTS/CTS (request to send/clear to send) to solve the hidden terminal problem and exposed terminal problem. The RTS/CTS mechanism incurs a high control overhead and long delay. Wang et al. [9] have proposed an approach which uses an extra channel to eliminate the problems. Power aware or directional antenna based solutions [10], [11] are lack of generality. The physical layer approaches [12], [13] are difficult to modify after deployment.

In this paper, we propose QL-MAC, a Q-Learning based MAC protocol for delay sensitive VANET applications. QL-MAC uses a Q-Learning algorithm to adjust the contention window size in order to avoid packet (frame) collisions. QL-MAC also can use a small contention window size to provide a fast channel access when the network load is light. QL-MAC can use position information to determine transmission decisions without relying on RTS/CTS.

There have been some protocols focusing on the efficient

978-1-4799-2355-7/14/$31.00 ©2014IEEE

utilization of multiple channels [14], [15]. In this paper, we consider how to efficiently use the same channel under various traffic conditions. The proposed approach is applicable to the multi-channel case.

The remainder of the paper is organized as follows. In section II, we give a detailed description of the proposed protocol. Next, we present simulation results in section III. Finally, we present our conclusions and future works in Section IV.

II. Proposed Protocol: QL-MAC

A. Protocol overview

The proposed approach is a MAC layer solution to provide low delay transmissions. We assume each node knows the position information about its one-hop neighbors. This can be easily achieved by exchanging beacon (hello) messages among neighbors.

QL-MAC improves the contention efficiency by learning the best contention window size using a Q-Learning algorithm. Each node gets a positive reward when a data frame is successfully delivered, and gets a negative reward when the transmission was failed. By dynamically adjusting the contention window size, the protocol can provide a high packet collision ratio and low channel access delay.

QL-MAC takes account of the hidden terminal and exposed terminal problem in the protocol design. When the MAC data (payload) size is larger than the predefined threshold $\mathrm{TH}_{DataSize}$ (512 bytes by default), the protocol uses RTS/CTS to avoid the hidden terminal and exposed terminal problems. Otherwise, the protocol utilizes position information (without using the RTS/CTS) to decide whether to refrain from transmitting or not.

Since the MAC data (payload) size for the delay-sensitive applications (such as VoIP) is always smaller than 512 bytes, we mainly discuss QL-MAC without RTS/CTS. We define the MAC layer retry limit to 4.

B. Q-Learning based contention window adjustment

In the IEEE802.11p MAC specification, the contention window size is initialized to 15 (aCWmin). However, when the data traffic load is high (this could possibly occur when the network density is high), a low contention window size incurs a high collision probability. When the sender node fails to receive the corresponding ACK for a data frame in the predefined time, the sender node increases the contention window size and retransmits the frame, resulting in a long delay due to the retransmission. This is avoidable by setting better contention window size at the first attempt. The contention window size is reverted to 15 after a successful transmission. This is also unacceptable when the network load is high.

As a solution for the issue, QL-MAC dynamically adjusts the contention window (CW) size. By using the best CW size, each node can access the channel with a small CW size when the data traffic load is low, and can avoid the collisions by using a larger CW when the data traffic load is heavy.

There are multiple access categories in IEEE 802.11p. The proposed approach does not violate the existing access priority because the protocol only changes the aCWmin (see [1]) which is shared by all access categories.

1) Q-Learning Model: Q-Learning [16] is a recent form of reinforcement learning algorithm that works by estimating the values of state-action pairs without requiring a model of its environment. Q-learning adjusts behavior through trial-and-error interactions with a dynamic environment. The Q-value $Q(s, a)$ $(s \in S, a \in A)$ in Q-learning is an estimate of the value of future rewards if the agent takes a particular action a when in a particular state s. By exploring the environment, the agents build a table of Q-values (Q-Table) for each environment state and each possible action. Except when making an exploratory move, the agents select the action with the highest Q-value.

The Q-Learning algorithm that used in QL-MAC is defined as follows. The entire network is the environment. Each node in the network is an agent. Each possible contention window size is considered a state of the agent. The set of all possible contention window sizes in the network is the state space. The contention window size is selected from the set $\{15,31,63,127,255,511,1023\}$. The contention window size is initialized to 15. The actions at the agent can be 1) Increase (**I**), 2) Keep (**K**) or 3) Reduce (**R**). In here, "Increase" means the increase of the contention window size. "Keep" and "Reduce" mean to keep and reduce the contention window size respectively. A state transition occurs when a node selects an action. Every node maintains a Q-Table which consists of Q-value $Q(s, a)$ whose value ranges from -1 to 1, where s is the current contention window size and a is a possible action. After transmission of a MAC frame, each node gets a positive or negative reward depending on the transmission status. If the transmission was successful, the node gets a positive reward. Otherwise, the agent gets a negative reward. By observing the reward, each node adaptively adjusts its contention window size.

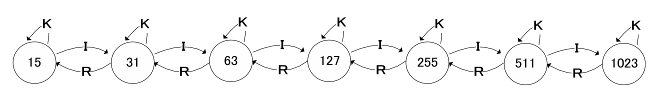

Fig. 1. States and actions.

2) Update of Q-values: As shown in Fig. 1, the possible actions are "Increase" (**I**), "Keep" (**K**) or "Reduce" (**R**). However, when the CW size is the minimum (maximum), it cannot be reduced (increased).

Q-table is updated after transmission of a MAC data frame. If the corresponding ACK for the data frame was received in the predefined time period, the node gets a positive reward. This means the current contention window size is large enough. If the node failed to receive the corresponding ACK in the predefined time interval, the node gets a negative result. This is because the packet losses are mainly because of the collisions with other packets. Q-table is updated as

$$Q(s_t, x) \leftarrow \alpha \times \left\{ R + \gamma \times \max_y Q(s_{t+1}, y) \right\} + (1-\alpha) \times Q(s_t, x). \quad (1)$$

The learning rate α is set to 0.6. Since the MAC frame transmissions happen frequently, 0.6 is enough to reflect the network topology changes. The discount factor γ (0.9) is to discount the reward when a change of CW size occurs (to avoid unnecessary CW modifications). The reward is calculated as

$$R = \begin{cases} R_{CW}, & \text{if the transmission (TX) was successful} \\ -1, & \text{if the TX was failed} \\ 0, & \text{if the TX was performed at the current state} \end{cases} \quad (2)$$

where R_{CW} denotes the positive reward for the current CW size. For a failed transmission, the reward is set to -1. The reward for a successful transmission (for various CW sizes) is defined in Table I. When the CW size is small and the transmission was successful, the reward is high. This is to encourage each agent to fully utilize the network bandwidth when the network load is light. In contrast, the negative reward makes each node to increase its CW size when the network load is heavy.

TABLE I
REWARD OF A SUCCESSFUL TRANSMISSION FOR VARIOUS SIZES OF CONTENTION WINDOW

CW size	15	31	63	127	255	511	1023
R_{CW}	1	$\frac{6}{7}$	$\frac{5}{7}$	$\frac{4}{7}$	$\frac{3}{7}$	$\frac{2}{7}$	$\frac{1}{7}$

The agent (node) updates Q-values for all states when a reward is received from the environment (see Algorithm 1). As shown in Eq. 1, $\max_y Q(s_{t+1}, y)$ is used to disseminate the reward to all possible states. s_{t+1} denotes the state after selecting the action x. For example, when the CW size is 15 and the node selects the "Increase" action, the state will be transited to $\{31\}$.

Algorithm 1 Update of Q-Table

1: Execute action x at state s, and transmit a data frame.
2: **if** (The TX was successful) **then**
3: Update Q-value for the $Q(s, x)$ with a positive reward.
4: **else**
5: Update Q-value for the $Q(s, x)$ with a negative reward.
6: **end if**
7: Update all other possible Q-values ($Q(\hat{s}, \hat{x})$) at other states.

3) Exploration, exploitation and convergence: Finding a balance between exploitation and exploration is important for a reinforcement learning algorithm. Typically, the exploitation is to choose the best action according to the agent's knowledge. The exploration is to discover new actions (in order to make the exploitation leads to the global optimum) by selecting a sub-optimal action.

In QL-MAC, the first action each agent would do is to set the CW size to 15. When the agent knows nothing about the network environment, taking the minimum CW size is the best choice. After that, the node makes an exploratory move with probability p (exploration), and chooses the action with the highest Q-value with probability $1 - p$ (exploitation). The probability p is calculated by $1 - Q_{max}$ where Q_{max} is the Q-value of the best action. Since an agent can get a higher reward when the CW size is smaller (see Table I), each agent never tries to increase its CW size as long as the current CW size can result in a successful transmission. When the CW size is larger than 15, QL-MAC also can change the CW size back to 15 by exploration when the network load becomes light. By using the exploration and exploitation, QL-MAC can adjust the CW size to the optimal value.

Convergence is a concern for a reinforcement learning algorithm. Watkins and Dayan [17] have proved that Q-Learning converges to the optimum action-values with probability 1 if the following two conditions are satisfied: 1) actions are repeatedly sampled in all states and 2) action-values are represented discretely. In the proposed algorithm, each node is an agent, and each specific CW size is equivalent to a state. All possible actions are visible by the corresponding node (agent). After each MAC frame transmission, all Q-values at all states are updated. Obviously, the action-values (Q-values) are represented discretely in QL-MAC. Therefore, QL-MAC satisfies all conditions for the convergence.

C. Solution for the hidden terminal problem

The common approach to solve the hidden terminal problem is the use of RTS/CTS. However, the RTS/CTS mechanism incurs a high overhead and long delay. The collisions between RTS packets still occur. The RTS/CTS mechanism is effective only when the RTS/CTS frame size is significantly smaller than the data frame size. When the MAC payload size is larger than 512 bytes, QL-MAC uses RTS/CTS. When the payload size is smaller, the RTS/CTS mechanism incurs a significant overhead, resulting in a degradation of network performance. Therefore, in this case, the protocol does not use RTS/CTS. However, by finding the best contention window size, the protocol can provide an efficient medium access solution.

D. Solution for the exposed terminal problem

IEEE802.11 MAC, upon receiving a data frame (bound for any other node), a node updates its NAV (Network Allocation Vector) settings using a duration value equal to the time required to transmit the data frame, one ACK frame and one SIFS interval when the new NAV value is greater than the current NAV value. Due to the exposed terminal problem, a node can be prevented from sending packets. In IEEE802.11 MAC, as a solution for the problem, when a node hears an RTS from a neighboring node, but not the corresponding CTS, the node deduces itself as an exposed node and transmits to other neighboring nodes. However, this transmission can collide with the corresponding ACK of the on-going transmission. Therefore, in the proposed protocol, we check the wireless medium before the transmission of the ACK. The terminal which wants to send an ACK frame also waits for a certain time period (at maximum of T_{data}) to ensure no other transmission is going on. If some transmissions are still going on after the maximum deferring time period, the node transmits the ACK. The ACK can still collide with other traffics. However, this is acceptable because other nodes can adjust their behaviors by learning from the collision.

The proposed protocol uses position information to check whether or not the frame is allowed to transmit. Each sender node attaches its own ID, position and the receiver node ID and receiver node position in the MAC frame (both the data frame and ACK frame). Upon receiving a MAC frame, each node is aware of the transmission occurring in the neighborhood. As shown in 2, when node S2 wants to transmit a frame to node D2, the node first calculates the distance between S2 and D1 ($d(S2, D1)$), and the distance between S1 and D2 ($d(S1, D2)$). If both $d(S2, D1)$ and $d(S1, D2)$ are larger than the transmission range R, node S2 transmits the frame. By using the position information, the protocol can efficiently utilize the channel without RTS/CTS.

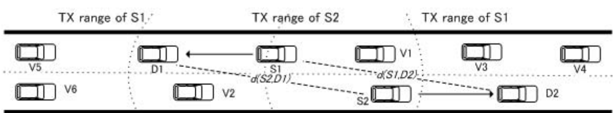

Fig. 2. An example for solving the exposed terminal problem.

E. Fairness

The fairness issue is also a concern for a MAC protocol. In original IEEE 802.11p specification, each node sets its contention window to the minimum value after a successful transmission. This incurs unfairness when the network load is high. This is because some nodes which sensed the collisions (nodes which have large contention window size) may difficult to get a transmission opportunity. In the proposed protocol, since each node adjusts its contention window according to the reward from the network, the fairness can be improved. The improvement of the packet delivery ratio also can contribute to the improvement of fairness.

III. Simulation results

We used network simulator ns-2 (version 2.35) [18] to conduct simulations in freeway scenarios [19]. Simulation environment are shown in Table II. we used a freeway (2000 m in length) which had two lanes in each direction. The distance between any two adjacent lanes was 5m.

Nakagami propagation model was used to simulate the channel fading. The parameters of the Nakagami model are shown in Table III. We used these parameter values because they model a realistic wireless channel of vehicular ad hoc networks [20].

We used FQLAODV [21] as the routing protocol because FQLAODV has better performance than AODV and its other extensions. QL-MAC was compared with CW-15-RTS/CTS (IEEE802.11p MAC with RTS/CTS), CW-15 (IEEE802.11p MAC without RTS/CTS), CW-31-RTS/CTS (IEEE802.11p MAC with RTS/CTS, aCWmin equals 31), and CW-31 (IEEE802.11p MAC without RTS/CTS, aCWmin equals 31). Other simulation parameters were the default settings of ns-2.35. We evaluated the protocol by changing the number of CBR connections, CBR data rates, and number of nodes. In the following simulation results, the error bars indicate the 95% confidence intervals.

TABLE II
SIMULATION ENVIRONMENT

	Freeway scenario
Topology	2000 m, 4 lanes (two lanes in each direction)
Number of nodes	50–300
Mobility generation	Ref. [19], maximum velocity : 80 km/h
Traffic flows	5–10 random CBR flows (Access Category Voice)
Packet size	32–512 bytes
MAC	IEEE 802.11 MAC (11 Mbps)
Propagation model	Nakagami Model
Simulation time	500 s

TABLE III
PARAMETERS OF NAKAGAMI MODEL

gamma0_	gamma1_	gamma2_	d0_gamma_	d1_gamma_
1.9	3.8	3.8	200	500
m0_	m1_	m2_	d0_m_	d1_m_
1.5	0.75	0.75	80	200

A. Effect of CBR data rate

We first evaluate the protocol's performance for various CBR data rates. The number of nodes was 75. There were 5 CBR connections, and the CBR packet size was 64 bytes. Fig. 3 shows the packet delivery ratios. When the CBR data rate is low, RTS/CTS mechanism has a positive effect on the packet delivery ratio. However, when the CBR data is high, the mechanism does not work well due to a high overhead. The RTS/CTS packets can also collide with each other or with other packets. QL-MAC has a significant advantage over other protocols. This is because QL-MAC can adjust the CW size, resulting in a significant reduction of MAC frame collision ratio. As shown in Fig. 4, with the increase of CBR rates, the MAC frame collision ratio increases. The situation is worse when aCWmin is 15. This shows the importance of increasing aCWmin when the CBR data rate is high. QL-MAC can significantly reduce the collision ratio as compared with other protocols.

Fig. 5 shows the average end-to-end delays for various CBR rates. A higher aCWmin results in a higher end-to-end delay. QL-MAC shows the lowest delay. The following 2 properties of QL-MAC contributed to the result: 1) QL-MAC can increase the CW size when the current CW size incurs too many collisions. As a result, the collision probability remains at a very small level. 2) QL-MAC can change the CW to a smaller value when the collision probability decreases.

Fig. 6 shows the packet delivery ratio fairness (achieved by 5 CBR connections) measured using the Jain's Fairness Index [22]. QL-MAC can significantly improve the fairness by adjusting contention window size.

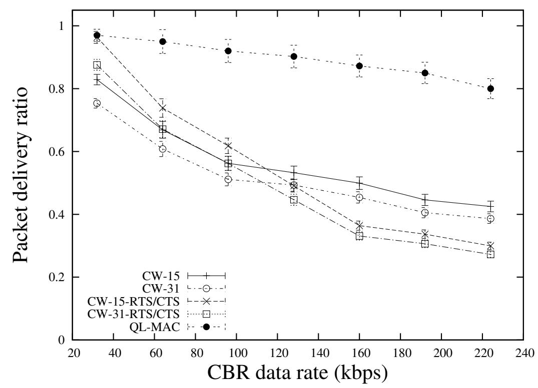

Fig. 3. Packet delivery ratio for various CBR data rates.

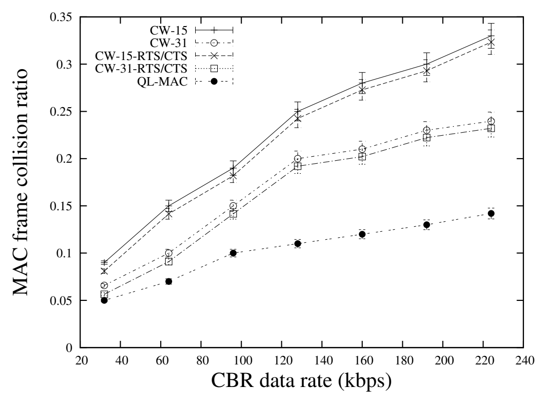

Fig. 4. MAC frame collision ratio for various CBR data rates.

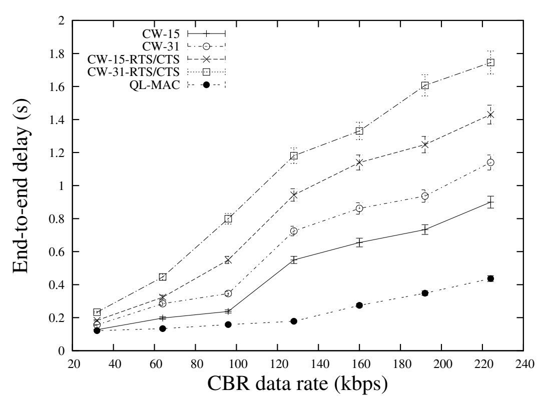

Fig. 5. End-to-end delay for various CBR data rates.

B. Effect of CBR packet size

Fig. 7 shows the packet delivery ratio for various packet sizes. The CBR rate was 128 kbps. There were 10 random

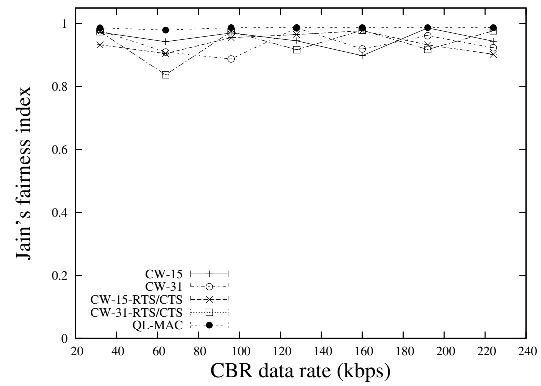

Fig. 6. Jain's fairness index of packet delivery ratio for various CBR data rates.

CBR connections. When the packet size is small, RTS/CTS mechanism results in a very poor performance. This is due to the high overhead of the RTS/CTS packets. The overhead is also the main reason for the high end-to-end delay of CW-15-RTS/CTS and CW-31-RTS/CTS (see Fig. 8). Due to the Q-Learning based CW adjustment, QL-MAC can work in various situations.

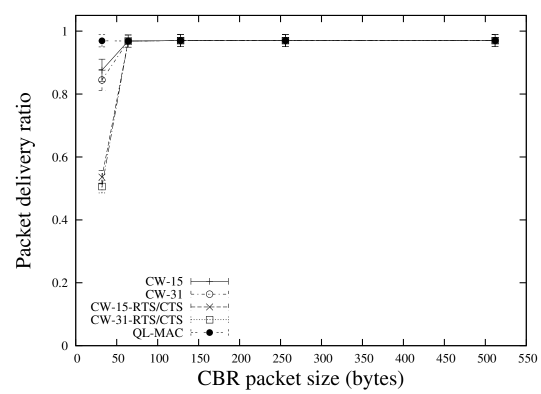

Fig. 7. Packet delivery ratio for various CBR packet sizes.

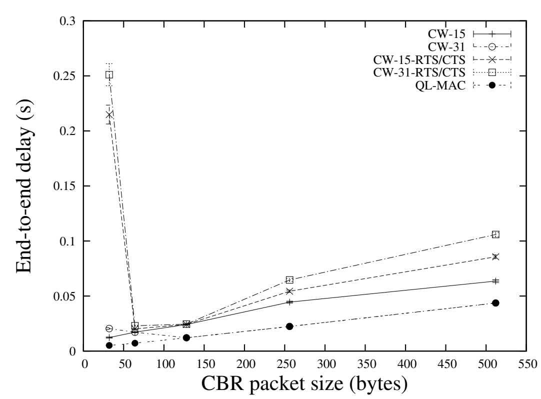

Fig. 8. End-to-end delay for various CBR packet sizes.

C. Effect of node density

Beacon (hello) messages are important for VANETs. In VANETs, the node density changes with road types and time. Therefore, we evaluate the proposed protocol for various numbers of nodes. CBR packet size was 128 bytes, and CBR data rate was 128 kbps. Beacon messages were transmitted using the AC_BE (best effort access category).

Fig. 9 shows the Packet delivery ratio for various numbers of nodes. The packet delivery ratio of CW-15 decreases with the increase of the node density due to the collisions between data packets and beacon messages. Since QL-MAC can efficiently manage the size of CW, the protocol shows the highest performance. Fig. 10 shows the end-to-end delay. We observe that QL-MAC can provide the lowest end-to-end delay by reducing the collision probability.

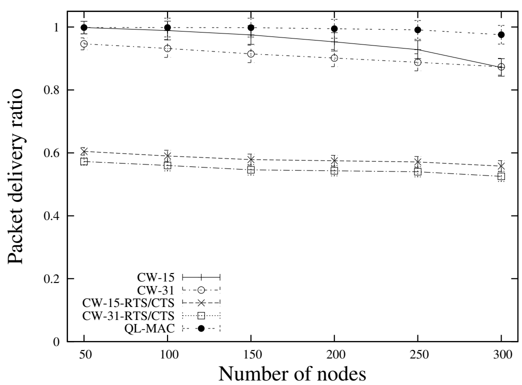

Fig. 9. Packet delivery ratio for various numbers of nodes.

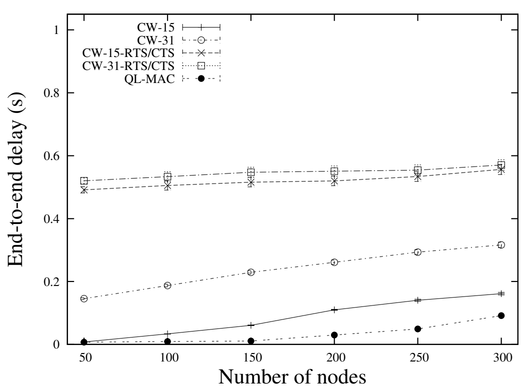

Fig. 10. End-to-end delay for various numbers of nodes.

IV. Conclusions and Future Works

We proposed QL-MAC, a Q-Learning based MAC layer protocol for VANETs. QL-MAC dynamically adjusts the contention window size based on a Q-Learning algorithm. QL-MAC can provide a significantly higher packet delivery ratio, lower end-to-end delay, and higher fairness than the original IEEE802.11p (with/without RTS/CTS, for different aCWmins) for various situations. In our future work, we will take account of the protocol performance for TCP flows.

Acknowledgment

This work was supported by JSPS KAKENHI Grant Number 25730053.

References

[1] IEEE Std 802.11 Amendment 7: Wireless Access in Vehicular Environments, 802.11p$^{\mathrm{TM}}$ -2010, 2010.

[2] C. Shi, X. Dai, P. Liang, and H. Zhang , "Adaptive Access Mechanism with Optimal Contention Window Based on Node Number Estimation Using Multiple Thresholds," *IEEE Trans. Wireless Commun.*, vol.11, no.6, pp.2046–2055, 2012.

[3] S. Kang, J. Cha. and J. Kim, "A Novel Estimation-Based Backoff Algorithm in the IEEE 802.11 Based Wireless Network," in *Proc. IEEE CCNC*, pp.1–5, 2012.

[4] S. H. Low, and J. C. Doyle, "Random Access Game and Medium Access Control Design," *IEEE/ACM Trans. Networking*, vol.45, no.2, pp.1303–1316, 2010.

[5] S. Teruhi, F. Nuno, and K. Watanabe, "Centralized Multiple Random Access Protocol with Contention Window Adjustment," in *Proc. IEEE VTC Fall*, pp.1628–1632, 2007.

[6] Q. Xia, and M. Hamdi, "Contention Window Adjustment for IEEE 802.11 WLANs: A Control-Theoretic Approach," in *Proc. IEEE ICC*, pp.3923–3928, 2006.

[7] P. Karn, "MACA - A New Channel Access Method for Packet Radio," in *Proc. ARRL/CRRL Amateur Radio 9th Computer Networking Conf.*, pp.134–140, 1990.

[8] L. B. Jiang, and S. C. Liew, "Improving Throughput and Fairness by Reducing Exposed and Hidden Nodes in 802.11 Networks," *IEEE Trans. Mobile Computing*, vol.7, no.1, pp.34–49, 2008.

[9] P. Wang, H. Jiang, and W. Zhuang, "A New MAC Scheme Supporting Voice/Data Traffic in Wireless Ad Hoc Networks," *IEEE Trans. Mobile Computing*, vol.7, no.12, pp.1491–1503, 2008.

[10] H. Cui, G. Wei, Z. Zhang, and J. Zhang, "Medium access control scheme supporting real-time traffic with power control in wireless ad hoc networks," *IET Commun.*, vol.4, no.4, pp.377–383, 2010.

[11] A.A. Abdullah, L. Cai, and F. Gebali, , "DSDMAC: Dual Sensing Directional MAC Protocol for Ad Hoc Networks with Directional Antennas," *IEEE Trans. Veh. Technol.*, vol.61, no.3, pp.1266–1275, 2012.

[12] P.-J. Wan, O. Frieder, X. Jia, F. Yao, X. Xu, S. Tang , "Wireless link scheduling under physical interference model," in *Proc. IEEE INFOCOM*, pp.838–845, 2011.

[13] L. Wang, K. Wu, M. Hamdi, "Combating Hidden and Exposed Terminal Problems in Wireless Networks," *IEEE Trans. Wireless Commun.*, vol.11, no.11, pp.4204–4213, 2012.

[14] C. Han, M. Dianati, R. Tafazolli, X. Liu, and X. Shen, "A Novel Distributed Asynchronous Multichannel MAC Scheme for Large-Scale Vehicular Ad Hoc Networks," *IEEE Trans. Veh. Technol.*, vol.61, no.7, pp.3125–3138, 2012.

[15] Q. Wang, S. Leng, H. Fu, and Y. Zhang, "An IEEE 802.11p-Based Multichannel MAC Scheme With Channel Coordination for Vehicular Ad Hoc Networks," in *IEEE Trans. Intell. Transp. Syst.*, vol.13, no.2, pp.449–458, 2012.

[16] C. Watkins, Learning from Delayed Rewards, PhD thesis, King's College, Cambridge, 1989.

[17] C. J.C.H. Watkins, and P. Dayan, "Q-Learning," *Machine Learning, vol.8*, no.3-4, pp.279–292, 1992.

[18] The Network Simulator - ns-2, http://www.isi.edu/nsnam/ns/, Accessed on Feb. 23, 2013.

[19] F. Bai, N. Sadagopan and A. Helmy, "Important: A Framework to Systematically Analyze The Impact of Mobility on Performance of Routing Protocols for Adhoc Networks," in *Proc. 22nd Annual Joint Conf. of the IEEE Comp. and Commun. Societies*, pp.825–835, 2003.

[20] A. Khan, S. Sadhu and M. Yeleswarapu, "A comparative analysis of DSRC and 802.11 over Vehicular Ad hoc Networks," Project Report, University of Californai, Santa Barbara, pp.1–8, 2009.

[21] C. Wu, S. Ohzahata and T. Kato, "Routing in VANETs: A Fuzzy Constraint Q-Learning Approach," in *Proc. IEEE GLOBECOM*, pp.213–218, 2012.

[22] R. Jain, D. Chiu, and W. Hawe, "A Quantitative Measure Of Fairness And Discrimination For Resource Allocation In Shared Computer Systems," DEC Research Report TR-301, September 1984.