RLAB: A Reinforcement Learning-based Adaptive Broadcasting for Vehicular Ad-hoc Networks

Seyedali Hosseininezhad, Ghasem Naddafzadeh Shirazi, Victor C.M. Leung

Electrical and Computer Engineering, University of British Columbia, Vancouver, BC, Canada

Email: {seyedali,naddaf,vleung}@ece.ubc.ca

Abstract—Effective context-aware broadcasting of information to the areas of interest (AoI) is a challenging problem in vehicular ad-hoc networks. It is usually assumed that the information about these AoI are a priori known, either by a centralized source of information or by the entire set of vehicles. In this paper, we propose a self-adaptive broadcast scheme based on distributed reinforcement learning, in which the vehicles are able to collaboratively tune the rate of their broadcast based on the network dynamics and without the initial knowledge about geographical distribution of AoI. The proposed approach enables a more practical implementation of distributed context-aware broadcasting, where no global information and only a partial synchronization is required. The convergence and broadcast performance of the proposed learning system is evaluated using simulations for several setups. These results show a significant improvement, in terms of number of useful broadcasts and delay, over the existing approaches, such as gossip-based broadcasting.

Index Terms—Vehicular Ad-hoc Networks, Message Dissemination, Context Aware Broadcast, Delay Tolerant Networks, Reinforcement Learning.

I. INTRODUCTION

One of the main challenges in VANETs is to find an efficient and reliable approach to disseminate the information towards interested parties. There have been an extensive amount of literature on broadcasting and data propagation techniques and analysis for mobile ad-hoc networks, and in particular VANETs so far [1]–[9].

In inter-vehicle information sharing, there are two major approaches to provide users with information. Pull-based methods like [4] are designed to query information from the source and retrieve it to the destination. However, in vehicular networks with intermittent link losses, it is very challenging to handle two way connections in a timely manner and in many cases data could become obsolete.

Push-based methods like [2] are designed to treat information as broadcast messages. DV-Cast [8] proposes a protocol to handle both dense and sparse traffic regimes in highways. The main contribution is to handle seamless switch between broadcast suppression in dense traffics and store-carry-forward in sparse areas. By exploiting limited mobility patterns in highway roads, this method achieves better performance in broadcast success rate and scalability. However this method still needs more sophisticated control messages for handling urban traffic conditions.

The main focus in methods like [2] and [8] is on the efficiency and reliability of broadcasts by exploiting the mobility pattern of adjacent vehicles. Another concern in the context dissemination is to shape the data fusion towards favorable locations. This issue is addressed in [1] as Geographical-Temporal Multicasting, which is defined as delivering a message from a data source to all devices in a distant contoured area. Although this definition is close to what we address here, we should note that multicasting is applicable only if destinations are a priori known, while in VANETs there is no registered client. The vehicles may only know that receiver might be interested in a specific category of information. Moreover, the AoI might reside in more than one contoured area. In such cases, the complexity of finding the optimal broadcasting strategy is cumbersome.

In [6], broadcast strategies based on the Gossip protocol are presented. A Gossip(p, k) is defined as a deterministic strategy, in which all nodes which are more than k hops away from the source of data will broadcast with probability p. Although Gossip and similar epidemic-based broadcast strategies are simple to implement, the main disadvantage is the excessive number of useless broadcasts, even in the areas that are unrelated to the final AoI.

In [10], a scenario with multiple disjoint areas of interest is suggested. The decision to broadcast toward destinations is based on a custom defined propagation function. This function is aimed to classify locations and shows how useful it is to have a receiver or a transmitter residing in a location. However there is no discussion about the design of propagation function itself.

SPACE [5] is a spatial aware data dissemination approach designed for traffic dissemination systems (e.g. traffic and travel information (TTI) applications). Their main contribution is to confine broadcast of traffic flow information into predefined zones of interest.

Contributions: So far, to the best of our knowledge, the research has focused on finding the optimal or heuristic methods to relay broadcasts towards known AoI. In this paper, we consider a more realistic situation, i.e. when the AoI are unknown to the source traffic generators or their geographical location is dynamically changing. For instance, in the TTI application in [5] if the travel behavior of nodes (which also should contain the average road to road traffic) is not given, or this information is not updated to meet the latest changes, such as new constructions or roadblocks, then decisions to classify regions of interest for TTI data become considerably inefficient. Our goal is hence to provide a distributed self-
adaptive learning mechanism for vehicles to autonomously recognize AoI by means of dynamic feedback from other vehicles, and also to classify locations in which broadcasting is more helpful. This leads to an efficient delivery of information to the AoI by avoiding useless broadcasts and dynamically learning from the VANET environment.

The rest of this paper is organized as follows. In Section II the system model is elaborated, and in Section III, our method, Reinforcement Learning-based Adaptive Broadcast (RLAB), is proposed. In Section IV, simulation results for investigating the performance of RLAB are presented. The conclusions are presented in Section V.

II. SYSTEM MODEL

We consider an adaptive broadcast framework for data dissemination in different VANET applications without prior knowledge about AoI. In Fig. 1, two different categories of such applications are shown. In Fig. 1a a traffic monitoring sensor equipped with standalone transmitter is located near junction C. Regardless of traffic condition, packets transmitted from this sensor can be relayed by nodes travelling in CD or it can be back-propagated in CB from there to A. If we don’t have the traffic flow information about this map (which is usually the case), existing methods are unable to choose the effective policy regarding broadcast in road segments CD, CB and BA. Another example is shown in Fig. 1b. Densely populated areas are potentially the best place to advertise. However, if distribution of interested users for different traffic profiles (e.g. rush hour vs. regular hours or weekdays vs. weekends) is unknown, then the geographical-temporal broadcasting used in the existing literature would have no advantage against context-oblivious broadcasting methods.

Based on these two examples we propose RLAB, in which the vehicles learn the AoI as well as the locations where broadcasts are useful. Specifically, RLAB tunes the rate of broadcasts based on distributed reinforcement learning so that vehicles are more likely to broadcast in the locations which boost the delivery of data towards AoI. In distributed reinforcement learning [11], every collaborator, i.e. a vehicle with transmission capability, learns their actions individually but they collaborate with each other to share their feedbacks and shape a distributed learning system.

In order to reduce the complexity, we divide the roads and intersections to fixed size segments. The size of each segment is less than the transmission range, and therefore, we can assume that all of vehicles inside a road segment can communicate with each other. After performing the segmentation, we calculate the communication overlap between adjacent road segments. The coverage is calculated as expectation of connection probability based on the log-normal shadowing model [12] and path loss exponents extracted from different vehicular environments in [13]. These values will be used later in reward assignments.

Inspired by distributed reinforcement learning approach in [11], we are considering every road segment as a unified packet distribution point. Vehicle(s) residing in a road segment act as agent(s). Presuming that road segments are not changing during a reasonably long time (unless there is a construction or road change), we can assume that every distributor has fixed neighbors and links to those neighbors have unchanging characteristics. Even if this condition doesn’t hold, our method will learn the environment but the learning would take more time to converge.

Let’s consider N vehicles commuting in the area within a set of road segments presented as $\mathcal{S}$. Since interest of users could be different based on the time of day or day of week or rush hour vs. non-rush hour, we differentiate among time sets and separate state space as it accommodates different measurements for each of them. $S_i(t)$ is defined as the road segment in which vehicle $i$ is residing at time $t$ ($S_i(t) \in \mathcal{S}, \forall i \in N$). We consider a MDP for each road segment $k \in \mathcal{S}$ as a tuple $M_k = (A_k, S_k, T_k, \rho_k)$ which is representing the behavior of every vehicle who resides in this road segment. $A_k$ is the set of actions possible to perform in $k$. $S_k$ is the set of states for road segment $k$. $T_k$ is representing the set of transition probabilities of MDP and $\rho_k$ is the set of rewards given for each action in this MDP.

A. State Space

For every road segment $k \in \mathcal{S}$ we define a FSMC including states ($T_k, r_k$). $\forall T_k \in \mathcal{T} \text{ and } \forall r_k \in \mathcal{R}$. $\mathcal{T}$ is the temporal traffic conditions set (rush hour v.s. non-rush hour) and $\mathcal{R}$ is the set of broadcast desirabilities. In other words, each state is representing a rate of message broadcast in a specific time category. For simplicity, we have defined two states for a road segment, namely Good and Bad, which indicate whether the nodes are encouraged to broadcast in this segment or not.

B. Actions

Logically all vehicles which are inside the same road segment in the same time, are experiencing the same state. But since our system should be completely distributed, each vehicle has to track it’s own state and save the transition probabilities and current state locally. Implementation details will be discussed later. We design our action based on state space of each road segment: Actions are selected from a set with two members: $\{\text{broadcast, carry}\}$. However there is a third action namely idle but in segments without a conveying vehicle to perform
broadcasting, there is no learning involved (This could be also explained as a standby state for learning agents).

C. State Transitions

The state of a road segment changes based on the feedback received from other road segments. If a packet reaches an interested user, reward is given to all of the road segments in which the broadcast was performed. Consequently, the state for the rewarded road segments are changed to Good. On the other hand, if no reward is given to a road segment during a predefined period, it causes a transition to state Bad. These transitions are performed by all nodes in their local version of states. It is necessary to synchronize the latest feedbacks for a road segments between those of nodes that are passing the road segment, because the decision on the state is based on how previous broadcasts in this road segment affected the environment.

III. RLAB: DISTRIBUTED Q-LEARNING WITH LOCAL STATES

Here we explain our proposed learning mechanism RLAB (Reinforcement Learning for Adaptive Broadcasting). Since the global state of the system is not recognizable by vehicles, modeling of the problem as a distributed reinforcement learning is not possible. To solve this problem, we consider local states for each of distributors. The idea of using local states has been explained in [11]. Since local states are only perceivable by local agent(s), agents should negotiate their values, not their states. The value of every distributor is based on its current state and former rewards obtained by its actions.

In distributed Q-Learning with local states, every node has its own state set and there is no global knowledge about current value of system. However an action chosen by a node will affect the value of other nodes in system and therefore should be considered in calculating rewards [11]:

\[
Q_i(x_i, a_i) = (1 - \alpha)Q_i(x_i, a_i) + \alpha(R_i(x_i, a_i) + \gamma \sum_j f(i, j)V_j(x'_j)),
\]

\[
V_i(x_i) = \max_{a \in A} Q_i(x_i, a).
\]

\[\begin{array}{|c|c|}
\hline
\alpha, \gamma & \text{Learning and discount factors} \\
Q_i & \text{Local value of node } i \\
x_i & \text{Current local state of } x_i \\
a_i & \text{Local action chosen by } i \\
R_i & \text{Instant local reward to node } i \\
j & \text{Other nodes in system} \\
f(i, j) & \text{Importance factor of } j\text{'s reward in } i\text{'s value} \\
V_j & \text{ } j\text{'s current value} \\
A & \text{Set of actions for } i \\
\hline
\end{array}\]

We are going to define two feedbacks for every action. The first feedback is an immediate reward given to the local broadcaster after taking the decision to whether broadcast or not ($R_i$). This feedback will represent the local impact of the decision on other nodes in vicinity. Remember the goal is to disseminate the information as much as possible using as few broadcasts as possible. The second feedback is issued by vehicles who find the received message useful. We replace the value of adjacent distributors or $V_j$ in (1) with a dedicated reward if the broadcast action done in this distributor has lead to delivery of the packet to an interested user in segment $j$. $f(i, j)$ is calculated as $1 - \frac{\text{expire}}{T}$, or in other words, if the relay path is mostly based on broadcasting relay, the rewards would be higher for distributors compared to when the path is mostly comprised of data muling periods.

A. Calculating Immediate Rewards

For immediate rewards we consider local parameters effective in decisions. First factor is based on how vehicles are moving around the transmitter. If vehicles are moving with variant speeds it is more likely that information scatters around compared to when vehicle are moving with a nomadic pattern. Therefore the reward should be more for a broadcast in an area with variant speeds.

Secondly, we consider a broadcasting more effective if the area is populated and it is highly desirable to have more listeners as a packet is transmitted.

This, we consider the duplicate broadcasts less valuable. If vehicles in vicinity of a broadcaster have not received the message before, the reward for broadcasting is higher.

We consider the message lifetime as an effective factor in rewarding. However, to encourage the nodes to propagate messages in wider area, reward is decreased only when the message is expired.

Based on described decision factors, we formulate the reward function. If a node decides not to broadcast the packet, the immediate reward will be zero. However for node $s$ with a set of neighbors $N(s)$ reward for broadcasting is:

\[
R_s(b, t + 1) = \left( \left[ \frac{T(t + 1)}{T_{\text{expire}}} \right] \sqrt{\sum_{i \in N(s)} (v_s - v_i)^2} \right) + \kappa(t + 1),
\]

\[
\frac{K_s}{|N(s)|} I - 1,
\]

\[
\kappa(t + 1) = \begin{cases} 0 & \text{if } Q(t) = \text{Broadcast} \\ \kappa(t) + \epsilon & \text{otherwise.} \end{cases}
\]

$T_{\text{expire}}$ and $T(t)$ denote the message lifetime and remaining time of validity. $I$ represents the importance factor for the type of information that is being broadcast. $v_s$ and $v_i$ represent the moving speed of broadcaster and its neighbors. $dV$ is the maximum relative speed while connectivity can be preserved. $K_s$ is the number of neighbors who have not already received the broadcast information.

To avoid early back-offs in action selection, $\kappa$ is added to (2). $\kappa$ increases every time that a broadcast is neglected and the reward for broadcasting grows consequently, its value can grow to more than zero if message has not been broadcast for a while. The growth speed depends on the value of $\epsilon$.

B. Calculating Delayed Rewards

In Fig. 2 the vehicle in up-left corner carries the message and broadcasts it in road segment A1. Consequently blue car
receives the message and carries it to road segment A4. Since A4 and B3 are in range, the red car overhears the broadcast and carries the message to B5 and broadcasts it again. Eventually the black car which is interested in receiving messages of this specific type receives the packet and updates feedback rewards for road segments A1, A4, B3 and B5. These updates are local but they will be dispersed in to the network gradually until all nodes are updated.

Since every car has a local instance of rewards, synchronization becomes a very important and challenging part of our system. Although a global synchronization is not needed for convergence, but if nodes that are using these rewards in their action selection policy can access the latest updates, convergence becomes faster and actions are more accurate. An intuitive approach for synchronization is to spread new updates toward the rewarded segments as soon as possible and the latest version should be kept alive in vicinity as long as possible. Using LINGER method proposed by Borsetti et al. in [14], update message can be kept alive in the area as long as there exist an updated vehicle in the region. We do not focus on data synchronization problem here since it wouldn’t fit in the scope of this paper.

IV. PERFORMANCE EVALUATION

In this section, we show our simulation results and investigate the convergence and performance of RLAB, in terms of number of useful broadcasts as well as delay. We compare our results with a simple Gossip broadcast method from [6]. We should note that the scope of methods like [4] and [14] differs from ours since they are designed for scenarios with known AoI. We have implemented a simulator in Matlab which handles the broadcasting and learning sequence. We extract the mobility trace generated by SUMO traffic generator [15]. For the simulations, we have used a 4 km x 1 km square field with Manhattan (grid) topology and 100 commuting cars. Roads are divided into segments with length of 100 m, and intersections are considered as segments with the radius of 100 m. Variable $\alpha$ in (1) is decreased from 0.3 to 0.01 during the simulation time, and $\gamma$ is set to 0.9. Also, $\epsilon$ varies between 0.05 and 0.15 based on convergence rate of state values. We considered an AoI in the north-west of the map, and a packet generator at the south-east which is transmitting one packet per second. To avoid excessive overhead, each node will broadcast its buffered packet at most once per second.

A. Convergence

After running the simulation, the rewarded road segments remain in the state Good, and the rest of the road segments are considered as Bad segments for broadcast. In Fig. 3, the
standard deviation in value \((V)\) of state \textit{Good} is shown for a \textit{Good} road segment. As can be seen, the value of this road segment is eventually converged, and hence, the actions for this road segment is fixed. In other words, RLAB is able to converge to a broadcast strategy after some time. We should note that the convergence time depends on the performance of reward synchronization. Particularly, if a car does not come back to the road segments it has previously passed, the other vehicles cannot receive the reward and the convergence would be longer.

**B. Broadcast Comparison**

To show the effect of RLAB on the number of broadcasts, we compare our method with Gossip(0.65,1) borrowed from [6]. Fig. 4 shows the total number of broadcasts in each time. We also fit the data with curves of degree 1 and 4 for RLAB and Gossip(0.65,1), respectively. As can be seen, although the size of our network is relatively small (i.e. only 100 active cars), there is a significant difference between the total number of broadcasts in two methods. Specifically, the average number of broadcasts remains fixed in Gossip(0.65,1). On the other hand, as RLAB learns the best road segments for broadcasting, the total number of broadcasts decreases over time. This is due to the fact that RLAB avoids useless broadcasts in \textit{Bad} road segments.

In Fig. 5, we compare the ratio of useful packets, i.e. delivered inside AoI, to the total number broadcast packets. As can be seen, RLAB is able to deliver the same number of useful packets with almost 5 times less number of broadcasts compared to Gossip(0.65,1).

**C. Delay**

Fig. 6 depicts the average experienced delay by packets before reaching AoI in Gossip(0.65,1) and RLAB and their corresponding fitted curves. As can be observed, the delay is slightly increased over time for Gossip(0.65,1) because of the imposed \textit{one broadcast per second} restriction. On the other hand, the delay of RLAB is less than that in Gossip method. The reason is that in Gossip(0.65,1), the nodes broadcast with probability equal to 0.65 everywhere. But in RLAB, the probability of broadcast in \textit{Good} road segments, which deliver data to AoI faster, is higher. We can also observe that the RLAB delay is a decreasing function of time since the learning of \textit{Good} road segments is improved over time.

**V. Conclusions**

In this paper, we proposed a distributed reinforcement learning approach to tune the message broadcasting rate. We assume that Areas of Interest (the regions where interested users are residing in) for some applications are unknown and correlated for some vehicles. Our method is able to tune broadcasting in favor of packet relay towards the AoI. Using simulations we show that our system converges, and also has a better delivery ratio and less delay compared to the existing methods. While our proposed method can significantly increase broadcast efficiency, it is mostly applicable for location based services. Also, we should note that the convergence time is related to inter-vehicle synchronization as well as the persistancy of AoI regions. The study of synchronization overhead will be addressed in future works.

**References**


