A Data Scheduling Approach for Software Defined Vehicular Ad-Hoc Networks

Sagarika Mohanty
I.G.I.T. Sarang
Dhenkanal, India

Kshira Sagar Sahoo
N.I.T. Rourkela
India

Srinivas Sethi
I.G.I.T. Sarang
Dhenkanal, India

Bibhudatta Sahoo
N.I.T. Rourkela
India

ABSTRACT

In recent years, vehicular ad-hoc networks (VANETs) is one of the rapid growing technology to form a network of vehicles and between vehicles and roadside units (RSUs) to provide a variety of services like safety, traffic efficiency and infotainment information to its drivers and passengers. In this paper, we have considered the communications are from V2I, I2V, I2I and on the basis of scheduling algorithm vehicles can download data from RSU. Usually vehicles may request for the same data, therefore we use multICASTing techniques to download data from RSU which reduces bandwidth and better scheduling performance. As vehicles are moving at a high speed, it may not be possible to download all the data from the service region of a single RSU and this will be forwarded to the nearby RSU. In our scheme we use SDN based VANET structure where SDN server push content information to the designated RSUs through switches and then to the respective vehicle.

Keywords
Software-Defined Network, RSU (Road Side Unit), scheduling, Vehicular Ad Hoc Network (VANET)

1. INTRODUCTION

Vehicular Ad-hoc Networks (VANETs) has gained a lot of attention in recent years both in academia and industry. The purpose is to enhance Intelligent Transportation System (ITS) by providing support for traffic efficiency, safety and infotainment information to its drivers and passengers. The two main components are the vehicles or OBUs (On Board Units) and the fixed infrastructure units (RSUs) placed along the road side. To disseminate timely data to vehicles, the communications are of V2V (Inter vehicular), V2I (vehicle to infrastructure) and within infrastructures (I2I). Though vehicles and RSUs both can store and forward data to other vehicles coming in the range but because of the limited storage capacity of vehicles they usually send the request to the nearby RSU to get updated data. Software Defined Networking (SDN) decouples the control plane from the data plane and provides programmability and flexibility to network components [1][2][3].

In this paper, we propose a data scheduling mechanism where vehicles in the V2I mode and within the transmission range of a RSU send requests to download data. This has strict time constraints as vehicles are the mobile nodes and they will be served before leaving the transmission range of RSU. We are considering two types of data: regular traffic or safety information and media data. We are considering three types of requests: emergency requests, first time requests and incomplete requests. In our proposed scheme, vehicle requests can be transmitted to the nearby RSU if it is not possible to serve the entire data within a single RSU range. Due to SDN approach our scheme achieves network performance, reduces delay in delivering data and proper bandwidth utilization due to multicasting.

The contributions of the paper are:

- To propose a scheme to serve maximum requests by using multicast techniques to utilize bandwidth properly.
- To reduce delay to improve network performance.
- To serve those requests that receive incomplete data from a RSU.

The rest of the paper is organized as follows. Section II summarizes the related work in this field by other authors. The preliminaries and performance metric are discussed in section III. Problem analysis and proposed algorithm are described in section IV. Section V presents the performance evaluation followed by conclusion in section VI.

2. RELATED WORK

There are many research articles on data scheduling in vehicular ad-hoc networks. Zhang et al. [4][5] focused vehicle to roadside data access to download and upload requests. They proposed D^S scheme, D^S/N scheme and two-step scheduling to provide a balance between upload and download requests. Gui et al.[6] proposed a motion prediction based scheduling scheme for vehicle to RSU data access. He emphasized on balancing the data among a group of RSUs which give better performance. P.T. Eugster et. al.[7] discussed many variants of publish/subscribe scheme namely topic-based, content-based and type-based. Shahverdy et al. [8] presented a scheduling scheme where RSU can act as a buffer point, a router or server to provide information to vehicles. Sana et al. [9] proposed a scheduling algorithm using multicast technique to serve multiple requests. They have given priority to the data which are of emergency nature. Dubey et al. [10] presented a data scheduling scheme to serve requests whose deadline are about to expire. In their scheme, they have considered both priority and popularity of each request. A survey of various scheduling schemes in vehicular ad hoc networks is done by Negi et al. [11]. Liu et al. [12] presented a cooperative data scheduling approach where they have considered both I2V and V2V communication where centralized scheduling is done at the RSU which resembles VANETs as a software-defined network. Recently few research articles are available on how software defined networking provides new services and features to VANETs. Ku et al. [13] proposed some architecture and services towards software defined VANET. A type-based content distribution approach was proposed by Cao et al. [14]. They used push and pull pattern to transmit the content and used SDN technology to support content distribution in vehicular networks. Sahoo et. al. [15] presented a case study where SDN architecture can apply on fog devices to enhance the performance of traffic management.
3. BACKGROUND
3.1 Preliminaries
In software-defined VANET architecture RSUs are the data plane components controlled by the SDN controller (controls the network behavior). Vehicles are the mobile nodes (data plane or forwarding elements) forward their requests to the RSU to retrieve data. The main idea of SDN is to decouple the network control from the forwarding elements (data plane). In this scheme, we considered two types of data: regular traffic related information, safety and real-time information like nearby gas station, parking place, local map etc. and high bandwidth media data. The former one takes only tens or few hundred bytes whereas the later one takes few kilobytes to megabytes to gigabytes. When a vehicle enter the service region of a RSU, it forwards its request to upload or download data. For download type, if content information is available in the RSU it will be forwarded to the respective vehicle. Otherwise, the request will be forwarded to the content server. The server push the content towards the switches and the SDN logic agents on switches will determine the routing paths to specific RSUs depending on the location of vehicles and their type of requests. From the server the content information will pass through the core switches, aggregation switches and edge switches to the respective RSUs. For bandwidth intensive media data usually it is not possible to serve the request within the service region of a RSU. From the edge switches the data items will be forwarded to the respective RSU where the vehicle will probable to enter to reduce delay. Usually vehicles request for the same data i.e. regular traffic and safety information. We use multicast techniques to serve the requests which are of same nature to utilize bandwidth properly.

Fig.1. Software-Defined VANET structure (DSRC and cellular link for communication)

We assume that a vehicle can estimate the service deadline of the request from its geographic position and driving velocity (GPS, Global Positioning System). We also assume that each vehicle can estimate the service deadline of the request from its geographic position and driving velocity. We assume that a vehicle can estimate the service deadline of the request from its geographic position and driving velocity. We also assume that each vehicle can estimate the service deadline of the request from its geographic position and driving velocity. We assume that a vehicle can estimate the service deadline of the request from its geographic position and driving velocity. We also assume that each vehicle can estimate the service deadline of the request from its geographic position and driving velocity.

3.2 Scheduling Scheme
We propose a scheduling scheme where RSU scheduler manages three different queues:

- **Emergency data queue (E queue)**
  Stores requests for emergency services (e.g., accident notification etc.).

- **Incomplete data queue (I queue)**
  Stores requests for data item that are incomplete. Request items in the three queues sorted with FCFS algorithm i.e., first come request will be served first.

- **First-time data queue (F queue)**
  Stores requests for data item from scratch or first time.

We consider three parameters for scheduling V2I data access.

**Deadline:** The time period in which vehicle remain within the transmission range of a RSU.

**Data Size:** Smaller data size items will be considered first.

**Number of Pending Requests:** The number of unserved requests. We compare three schemes along with our proposed scheme.

**First Come First Serve (FCFS):** Earlier arrival request will be served first.

**Smallest Data Size First (SDF):** Smaller data size items will be served first.

**D*S Scheme:** Smaller data size items will be served first.

**D*S/N Scheme:** Here, each request is given a value called the DSN_Value:

\[
\text{DSN}_\text{Value} = (\text{deadline} - \text{currentTime}) \times \text{datasize} / \text{number_of_pending_requests},
\]

Requests with the minimum DSN_value will be served first.

3.3 Performance Metric

**Service Ratio:** It is the ratio of the total number of served or satisfied requests to the total number of arriving requests.

**Service Delay:** It is the duration from request submission time to the retrieval of the data item i.e., the waiting time.

4. SYSTEM MODEL
4.1 Problem Analysis

1. In V2I communication, the set of data items requested by a vehicle is denoted by \( D = \{d_1, d_2, d_3, ..., d_n\} \). \( V = \{V_1, V_2, V_3, ..., V_n\} \) denotes the set of vehicles, where \( |V(t)| \) represents the number of vehicles. Each vehicle can be either in V2V or V2I mode at a time and is represented by \( V_i(t) \) and \( V_i(t) \) respectively i.e. \( V_{V_i}(t) = V_i(t) \) and \( V_{V_i}(t) = V_i(t) \). Each request is given a value called the DSN_Value:

\[
\text{DSN}_\text{Value} = (\text{deadline} - \text{currentTime}) \times \text{datasize} / \text{number_of_pending_requests},
\]

2. **Service Ratio:** It is the ratio of the total number of served or satisfied requests to the total number of arriving requests.

3. **Service Delay:** It is the duration from request submission time to the retrieval of the data item i.e., the waiting time.

4. **Performance Metric**

**Service Ratio:** It is the ratio of the total number of served or satisfied requests to the total number of arriving requests.

**Service Delay:** It is the duration from request submission time to the retrieval of the data item i.e., the waiting time.
(V, RQV(t), did, rtype, op, deadline), where did is the identifier of the requested data item or service, rtype denotes the request type (emergency requests, first-time requests and incomplete requests), op is the operation to be performed by the RSU(upload or download) and deadline is the time constraints after which vehicle moves out from the service region of a RSU. Incomplete requests are characterized by an extra tuple i.e., 7-tuples (V, RQV(t), did, pds, rtype, op, deadline), where pds represents the data size that is received until now from previous RSU.

2. In I2V communication, RSU multicast one data item in each scheduling period denoted by d(t) where d(t) ∈ D. To satisfy the condition

\[ V_i \in V \land V_i \in V(t) \land d_i(t) \in P Q(t) \]

This states that V_i must be in the RSU region, V_i must be in the I2V mode and d(t) has not yet been retrieved i.e., it is a pending request of V_i. Reply message consists of (V, SRV(t), did, pds), where V_i is the vehicle id, SRV is the satisfied request of V_i, did is the identifier of the data item and pds denotes the requested data.

In case due to mobility the requesting vehicle leaves the service region of the RSU then it will be forwarded to nearby RSU where the vehicle will probable to enter.

3. In I2I communication, one RSU send information to the nearby RSU for those vehicles whose requested data item is not complete or satisfied within its service region. To satisfy the condition

\[ R_i \neq R_j \land R_i \neq R \land \forall R_j \in R \]

This means that R_i and R_j must belongs to the set of RSUs and must be in the I2I mode. The message contain (R_i, V_i, did, pds), where R_i and V_i are the identifier of the RSU and vehicle and did is the identifier of the data item and pds is the data size that is received until now from pervious RSU.

Table 1. Notations used in the Problem Statement

<table>
<thead>
<tr>
<th>Notations</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>D={d_1,d_2,…d_n}</td>
<td>set of data items</td>
</tr>
<tr>
<td>V(t)={V_1,V_2,…V_n}</td>
<td>set of all vehicles at time t</td>
</tr>
<tr>
<td>R={R_1,R_2,…R_n}</td>
<td>Road side units(RSUs)</td>
</tr>
<tr>
<td>V_R(t)</td>
<td>set of vehicles within the RSU’s coverage</td>
</tr>
<tr>
<td>SW_E</td>
<td>set of edge switches</td>
</tr>
<tr>
<td>RQ_V(t)</td>
<td>requests submitted by V_i</td>
</tr>
<tr>
<td>SR_V(t)</td>
<td>satisfied requests of V_i</td>
</tr>
<tr>
<td>PR_V(t)</td>
<td>pending requests of V_i</td>
</tr>
<tr>
<td>IR_V(t)</td>
<td>incomplete requests of V_i</td>
</tr>
<tr>
<td>S(R_i)</td>
<td>DSRC or service region of R_i</td>
</tr>
<tr>
<td>D(R_i,V_i)</td>
<td>distance between RSU R_i and vehicle V_i</td>
</tr>
<tr>
<td>T_i serve</td>
<td>serving time of vehicle V_i</td>
</tr>
</tbody>
</table>

Our scheduling scheme includes the following steps:

5. PROPOSED ALGORITHM

When a vehicle enters the service region of a RSU, it forwards its request to the RSU. If request type is download then the scheduler appends it in any one of the three queues in FCFS order. The scheduler calculates the DSN_value for each request, where

\[ \text{slackTime} = \text{deadline} - \text{currentTime} \]

RSU scheduler waits for some time then multicast same type of requests whose slackTime is greater than T_serve (serving time of vehicle). Otherwise, packets will be dropped or forwarded to the nearby RSU. This V2I communication between a vehicle and a RSU is described in algorithm 1.

Algorithm 1

---

for each vehicle \( V_i \in V(t) \land V_i \in V(t) \) and \( R_i \in R \) do

if request_type=download

append request in E or F or I queue

endif

calculate DSN_Value

if \( D(R_i,V_i) < S(R_i) \) \&\& SlackTime>=T_i serve

multicast response message

else

drop packets

forward request to RSU R_i

endif

endfor

---

6. PERFORMANCE EVALUATION

We have simulated this model using Mininet simulator. The traffic characteristics are based on Green Shield’s Model.
which is widely used[16]. For performance comparison, we have used three data scheduling schemes: FCFS, SDF, D*S with our proposed scheme. The performance of the proposed scheme is evaluated in terms of service ratio and service delay.

**Fig. 3 Shows the Effect of Workload on Service Ratio**

Fig. 3 shows the effect of workload on service ratio. For increasing number of requests if downloading from scratch then it descends with lower slope in comparison to the request which is incomplete i.e., downloading from remaining. From the above figure it shows that if you are downloading from scratch then it gives very poor performance.

If the vehicle leaves the RSU region with incomplete data, then it will wait till it enter another RSU region. It sends the request containing the data size of the item (pds) that is received until now from previous RSU. The scheduler enter it in I queue. If data items are not available in the RSU it will be forwarded to the edge switches from where it will pass to the respective RSU. Otherwise, it will be pushed to the content server. This course of action is discussed in algorithm 2.

**Algorithm 2**

```plaintext
for each vehicle \( V_j \in V(t) \) and \( V_j \in V_{R_j} \) and \( R_j \in R \) do
    for each request \( Rq_{R_j}^{pds} \in HR_{V_j} \) do
        push request to RSU \( R_j \)
        if data is not available there
            \( SW_{E_j} \leftarrow Rq_{E_j}^{pds} \) (where \( SW_{E_j} \in SW_{E} \))
            \( R_j \leftarrow P(Rq_{E_j}^{pds}) \) (process request from edge switch will go to RSU)
            \( V_j \leftarrow R_j \)
        else
            push the content request to the server
        endif
    endfor
endfor
```

**Fig. 4 Service Delay for emergency requests**

Fig. 4 shows the service delay for emergency requests. From the figure it clearly shows that FCFS shows very poor performance when dealing with emergency requests. Due to multicasting and SDN architecture delay is much less in our proposed scheme compared to others. Our scheme gives better result as it is taking pending requests into consideration.

**Fig. 5 Service Delay for first-time requests**

As shown in fig. 5 the delay for first-time requests in our proposed scheme is much less compared to the other three schemes. FCFS has the maximum delay in comparison to others.
Fig 6 shows the service delay of proposed scheme using SDN compared to traditional scheme. In traditional approach delay comparatively increases along with the increase in data size compared to our proposed scheme using SDN approach.

7. CONCLUSION

In this paper, we have taken advantage of software-defined network to propose a data scheduling mechanism to distribute data from RSUs to vehicles. Here, RSU scheduler maintains three different queues: emergency data queue for emergency services, first time data queue for first-time requests and incomplete data queue for incomplete-requests to serve each type of request separately. Any one of three queues can be selected based on DSN_value that consider service deadline, data size and number of pending requests. We use multicast technique to serve similar requests at a time which offer desirable performance and higher service ratio. Due to SDN environment our proposed scheme reduces delay because from the edge switches data items will be transmitted to the RSU and from that to requesting vehicles. From the implementation we have seen that our proposed scheme provide better performance compared to the other schemes.

In our future work we will take advantage of V2V communication and consider disseminating data between neighboring vehicles within and outside the service region of a RSU and from nearby lane. This approach does not support multihop V2V communication as this will spread the message to a larger distance. So, in our future work we will also consider RSUs can coordinate multihop V2V communication in SDN environment.

8. REFERENCES


