

Online Algorithm for Opportunistic Handling of Received Packets in Vehicular Networks

Ala Al-Fuqaha¹, Senior Member, IEEE, Ammar Gharaibeh, Member, IEEE,
Ihab Mohammed, Student Member, IEEE, Sayed Jahed Hussini, Student Member, IEEE,
Abdallah Khreishah, Senior Member, IEEE, and Issa Khalil, Senior Member, IEEE

Abstract—In vehicular ad-hoc networks, due to high mobility, vehicles usually communicate for short periods of time with several neighboring vehicles and are required to process data fast; sometimes in the order of few milliseconds. This urgency of data processing is further heightened in safety-critical scenarios that involve many vehicles. Such scenarios require data to be prioritized and processed with minimum delay. While packet scheduling has been extensively studied, these studies focus on channel scheduling, our work focuses on processing received packets by a vehicle in dense scenarios. In this paper, we formulate the prioritized data processing problem as an integer linear program given a prior knowledge of the request sequence and prove that it is NP-complete. Due to the difficulty of predicting the traffic patterns and obtaining the request sequence in advance, we propose an online algorithm that does not require the prior knowledge of the request sequence and achieves an $\mathcal{O}(1)$ competitive ratio. The proposed online algorithm strives to accept higher severity packets for processing in order to maximize the cumulative severity given vehicular communications/computation capacity constraints. Using real traffic traces, we evaluate the performance of the online algorithm against three online algorithms, in which two of them use an exponentially weighted moving average-based threshold while the other one accepts requests as capacity permits. Our evaluation shows that our algorithm achieves up to 492% more cumulative severity compared to the three other baseline algorithms.

Index Terms—VANET, online algorithm, packet scheduling.

I. INTRODUCTION

ROAD safety remains a major challenge internationally [1] and is tackled by the industry and governments. In the US, the Intelligent Vehicle Initiative (IVI) was launched by the

Department Of Transportation (DOT) in 1997 with the primary goal of enhancing road and highway safety by increasing the development of driver assistance technologies [2]. To improve safety, it is important that we make vehicles more intelligent. Hence, VANETs have emerged as an exciting research area. The network of vehicles equipped with embedded sensors, processing and wireless communications capabilities offers countless life changing applications that can revolutionize road safety, comfort and driving experiences [3]. VANETs allow direct communications among vehicles and make it possible to share information in the absence of infrastructure.

The promises of wireless communications to support vehicle safety have led to the adoption of the Dedicated Short Range Communications standard (DSRC; ASTM E 2213-03) by ATSM and IEEE in July 2003, with the aim of providing standard for wireless communications capabilities in Intelligent Transportation Systems (ITS). DSRC operates in the 5.9 GHz band that is assigned for use by Intelligent Transportation Systems (ITS) vehicle safety and mobility applications, with different channels designated for different applications [4]. DSRC supports communications among vehicles traveling with speeds of up to 200 km/h at a default rate of 6 Mbps (up to 27 Mbps) and a transmission range of 300 m [3].

To ensure Quality of Service (QoS), DSRC divides its frequency band into six Service Channels (SCH) and one Control Channel (CCH) such that each packet has one of following priorities: low, for dissemination of messages transmitted over the SCH or high, for safety or control messages transmitted over the CCH [3]. Such division of band resources ensures that vehicles transmitting over CCH get priority access for transmission and information dissemination, but beyond this labeling there are no mechanisms to ensure that messages with higher priority will be processed earlier than lower priority messages at the receiver side.

While improving QoS in DSRC, as well as utilizing online algorithms to enhance the performance of VANET applications have been studied extensively, none of the studies have tackled the problem of minimizing delay for safety-critical data in VANETs on the receiver side. To the best of our knowledge, this is the first work of its kind that adopts an online algorithm for packet scheduling and prioritization with a view to minimize delay for safety-critical data while the incoming packets are processed.

Due to the urgency of safety messages and their strict QoS requirements, there is a need for methodologies in support

Manuscript received May 3, 2017; revised December 4, 2017 and February 11, 2018; accepted February 13, 2018. Date of publication March 15, 2018; date of current version December 21, 2018. This work was supported in part by NPRP through the Qatar National Research Fund (a member of Qatar Foundation) under Grant 71113-1-199 and in part by NSF under Grant CNS-1647170. The Associate Editor for this paper was X. Cheng. (Corresponding author: Ala Al-Fuqaha.)

A. Al-Fuqaha, I. Mohammed, and S. J. Hussini are with the Department of Computer Science, Western Michigan University, Kalamazoo, MI 49008 USA (e-mail: ala.al-fuqaha@wmich.edu; ihabahmedmoha.mohammed@wmich.edu; sayedjahed.hussini@wmich.edu).

A. Gharaibeh is with the Department of Computer Engineering, German Jordanian University, Amman 11180, Jordan (e-mail: ammar.gharaibeh@gnu.edu.jo).

A. Khreishah is with the Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ 07102 USA (e-mail: abdallah@njit.edu).

I. Khalil is with the Qatar Computing Research Institute, Hamad Bin Khalifa University, Doha, Qatar (e-mail: ikhalil@qf.org.qa).

Digital Object Identifier 10.1109/TITS.2018.2809917

of service differentiation and admission control. To solve this problem, we propose an opportunistic online algorithm that ensures that messages with higher context severity incur lower processing delay.

We tackle the problem of prioritized message processing by formulating it as an ILP that requires the request sequence in advance and prove that it is NP-complete. However, knowing the sequence of requests in advance is infeasible since the vehicular traffic pattern cannot be easily predicted. Therefore, we propose an online algorithm that maximizes the cumulative severity gain while respecting the vehicle's communications and capacity limits. Besides, the algorithm does not require the knowledge of the sequence of requests and has a competitive ratio of $\mathcal{O}(1)$. To measure the performance of the proposed algorithm, two other online algorithms are implemented which utilize an exponentially weighted moving average based threshold technique. The three algorithms are tested using real traffic traces and results show that the proposed algorithm outperforms the other two algorithms by more than 80% of the cumulative severity.

The rest of this paper is structured as follows: related work and the our model is discussed in section II. Section III discusses the problem settings and section IV shows the NP-completeness of the problem. The online algorithm is presented in Section V while performance analysis and validation experiments are presented in Sections VI–VIII, Section IX concludes the paper.

II. RELATED WORK

Quality of Service (QoS) in Mobile Ad-hoc Networks (MANETs) has been studied extensively [5]. In VANETs, as a subclass of MANETs, considerable amount of scientific work has been dedicated to packet scheduling as a means of improving QoS. Reducing and mitigating congestion via packet scheduling has been studied in [6]–[9]. In addition, scheduling to minimize packet delays in routing protocols has also been studied in [10]–[12]. But, all aforementioned studies focus on channel scheduling packets, while our work focuses on processing received packets by a vehicle in dense scenarios. Furthermore, Online algorithms have been utilized by [13] and [14] to enhance performance of VANET applications. In both studies an online algorithm, Oldest Job First (OJF) is used. In [13] an online algorithm is used to minimize delay across intersections, while in [14] an online algorithm is used for coordinating vehicles in platoons. In spite of the fact that both aforementioned studies use online algorithms, none of them have addressed the problem of minimizing delay for safety critical data from processing perspective of a vehicle. To the best of our knowledge, this is the first work of its kind that adopts an online algorithm for packet scheduling and prioritization with a view to minimize delay for safety critical data. We use an online algorithm for prioritizing received packets so that safety critical data experiences lower processing delay.

Puthal *et al.* [6] used the Congestion Level (CL) metric to detect congestion and ensure packets with higher priority, which are usually safety messages, get minimum delay.

CL is calculated from data that is gathered across the stack protocol. Such a mechanism is helpful in decreasing the congestion in VANET, but it is not designed, and cannot prevent congestion created by a malicious node in VANET that targets one or few other specific nodes. Furthermore, the study is focused on reducing congestion while transmitting data, but in our work we focus on prioritizing safety critical data when packets are received and processed by a vehicle. In scenarios with dense vehicle numbers (e.g., highways) due to high volume of received data, processing safety critical data in a timely fashion is important.

Kumar *et al.* [12] propose a prioritized based scheduling algorithm in order to improve QoS. The algorithm uses dynamic prioritization on a server to service vehicles. All vehicles send their requests to the RSU, which then forwards the requests to the server. Based on the request priorities, the server distributes the vehicle requests into prioritized queues. The main problem with such an approach is that it ignores the dynamic topology of VANETs. Parameters are only valid for few seconds in VANETs as vehicles move fast and communications times are only a few milliseconds or even microseconds [15]. But the major difference relative to other works is that this approach utilizes a central server that controls prioritization and decides which vehicles will be served, while in our model, our focus is on the scheduling of received flows of a vehicle.

In [16], a central priority based scheduler is used to control a platoon of vehicles in VANETs. Such scheduling is based on a central scheduler represented by a circular buffer. The scheduler is used to facilitate access among a platoon of vehicles in VANET, and is not suitable for general purpose scheduling of packets in VANETs.

Wu *et al.* [17] have tried to lower the service delay for users by using a packet scheduling architecture for classified services at Access Points (AP). This study focuses on enhancing the user experience in an environment that has infrastructure APs and users would connect to infrastructure for services. As vehicles are mobile in VANETs, users need to connect to several different APs during their journey. The work in [17] develops a scheduling scheme to predict the load and the service delay for different types of services and to choose an optimal AP for each service.

The study in [18] explores prioritized directional broadcasting for message dissemination in VANET and proposes an accident prevention Directional Broadcast and message priority assignment system. Initially, each message is assigned a priority statically. Then, based on its priority, a sending method is chosen. The research is concerned with how to disseminate messages with higher priority in minimal time rather than how to prioritize packets and minimize delay for all priority levels.

In [19], QoS parameter improvements such as minimizing delay among others are studied. The study focuses on minimizing delay for packets being routed using different routing algorithms, with focus being on improving message dissemination in VANETs. Zeng *et al.* [20] utilize channel prediction based scheduling in order to improve data dissemination. Boban *et al.* [10] have also investigated QoS

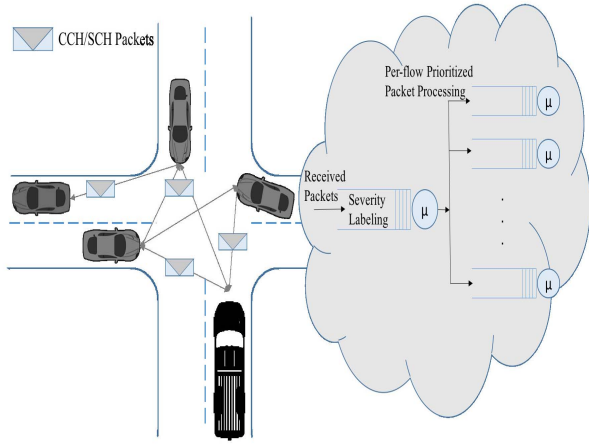


Fig. 1. Overall system model for severity labeling and admission control to minimize delay.

parameters (e.g., end-to-end delay) improvements albeit only for unicast routing algorithms. QoS and priority based routing protocols are also studied in [11] and Suthaputthakun and Sun propose a broadcast routing protocol that is capable of providing differentiated services based on message priority among other characteristics. Xu *et al.* [21] analyzed typical VANET routing algorithms to evaluate QoS performance under different conditions. Authors of [10], [18], [19], [21], study and/or evaluate different routing algorithms and their impact on QoS, but none of these studies focus on minimizing delay and packet scheduling on vehicular level. Most of routing algorithms target achieving the best performance at network level.

VANETs have very dynamic topologies and in order to ensure and promote safety, vehicles have to process safety-critical data with minimum delay. As shown in Figure 1, a given vehicle in VANETs at any given time might be surrounded by many other vehicles. In order to ensure safety, each vehicle must allocate a portion of its processing rate to each of its neighboring vehicles so that it can process messages from all its neighbors while ensuring that safety-critical data receives prioritized processing. In our model, as shown in Figure 1, a vehicle guarantees a minimum processing rate for all neighboring vehicles. In Figure 1, this is denoted by multiple queues. Beyond that processing rate, the residual processing capacity in the vehicle is auctioned by an online algorithm which gives priority to higher severity vehicles. To define severity, the algorithm assigns a *context severity* metric to its neighboring vehicles. Context Severity of each vehicle is calculated through deep packet inspection and processing of first and second order statistics of previous packets that include parameters such as the vehicle's speed, acceleration, GPS location, direction and segment ID. It should be noted here that the assignment of the context severity metric does not incur a significant overhead as it can be accomplished through rules that utilize the statistics of the received packets [22]. As shown in Figure 1, when a vehicle receives a packet, the packet's severity level is determined and labeled in the initial queue. Then based on the packet's severity level it will be assigned to one of the prioritized queues.

The vehicle will process the queues with higher priority first, and when no packets are left in that queue, it processes packets from lower priority queues. It should be emphasized here that the single vehicle in our model refers to a receiver (i.e., not a single source). In our model, there are multiple data sources and multiple receivers. We consider the case where every receiver takes decisions independently from the other receivers; thus, we focus on the case of a single receiver only.

III. SETTINGS

We assume a slotted time system. A VANET consists of V vehicles, where each vehicle $v \in V$ has a residual processing capacity of D_v messages per time slot. If the vehicle does not have residual processing capability, its residual processing capacity is set to 0. The input consists of a sequence of vehicle advertisement messages $\beta_1, \beta_2, \dots$, the j -th of which is represented by $\beta_j = (p_j, r_j, T_j, t_j(0))$, where p_j is the severity level of message β_j , r_j is the processing rate required by request β_j , T_j is the duration of processing β_j , and $t_j(0)$ is the time slot at which β_j appears. For simplicity, we assume that T_j is a multiple integer of time slots.

Our objective is to decide which requests a vehicle should accept in order to maximize the cumulative severity gained by the vehicle without violating the vehicle's capacity. We introduce the formulation of the optimization problem in the next section.

IV. PROBLEM FORMULATION

In this section, we consider the problem of maximizing the cumulative severity of each vehicle independently of the decisions made by other vehicles. Therefore, in the following, we study the problem from a single vehicle's perspective. We then show that the problem is NP-complete.

A. The Formulation

Before presenting our formulation, we introduce the following variable

$$X_j = \begin{cases} 1 & \text{if the } j\text{-th message is accepted} \\ 0 & \text{otherwise.} \end{cases}$$

Moreover, let

$$r_j(\tau) = \begin{cases} r_j & t_j(0) \leq \tau \leq t_j(0) + T_j \\ 0 & \text{otherwise.} \end{cases}$$

The problem of maximizing the cumulative severity of a single vehicle in a VANET is formulated as the following Integer Linear Program (ILP):

$$\max \sum_j X_j p_j \quad (1)$$

$$\text{s.t. } \sum_j X_j r_j(\tau) \leq D_v \quad \forall \tau \quad (2)$$

The objective is to maximize the cumulative severity gained by a vehicle while satisfying the capacity constraints of the vehicle at every time slot.

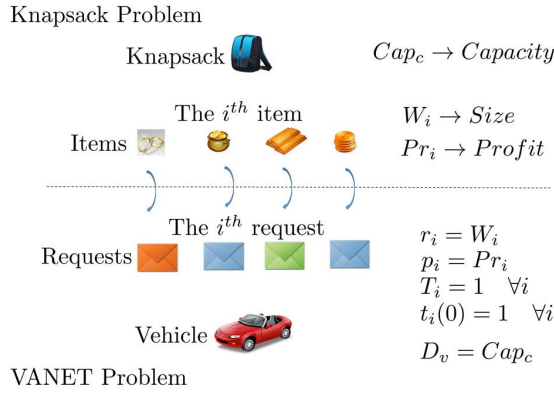


Fig. 2. Illustration of the reduction process used in the proof of Theorem 1.

B. NP-Completeness Proof

In this section, we show that the ILP optimization problem presented in the previous section is NP-complete as stated in the following theorem:

Theorem 1: *The ILP optimization problem is NP-complete.*

Proof: We prove this theorem by a reduction from the well-known knapsack problem. In the knapsack problem, we have a knapsack with capacity Cap . We have I items each with weight W_i and profit Pr_i . The objective is to select the items that fit in the knapsack such that the capacity of the knapsack is not violated and the total profit is maximized.

The reduction from the knapsack problem is done as follows: (1) The capacity of the knapsack is mapped to the capacity of the vehicle. (2) The number of requests in our problem is set to I . (3) The i -th item in the knapsack problem is mapped into the i -th request of our problem, where the rate r_i of the i -th request is set to W_i and the severity level p_i is set to Pr_i . (4) The arrival time of all requests $t_i(0)$ is set to 1 for all i . (5) The duration of all requests T_i is set to 1 for all i . The reduction is illustrated in Figure 2.

It is clear that a solution to the knapsack problem with total profit B is feasible iff it is a feasible solution to the instance of our problem with cumulative severity B , where the selected items in the knapsack problem correspond to the selected requests in our problem. The other direction of the iff statement is done in a similar way. $\square$

V. ONLINE ALGORITHM

We showed in Section IV-B that the ILP optimization problem is NP-complete. Combined with the fact that the ILP optimization problem requires the knowledge of all requests a priori, where obtaining such knowledge can be hard in real scenarios. In this section, we present an online algorithm that works on a per-request basis (i.e., decides whether to accept or reject a request when the request arrives) in order to maximize the cumulative severity gained from the accepted requests while satisfying the processing capacity of the processing vehicle at all times. We also show in Section VI that our algorithm achieves an $\mathcal{O}(1)$ -competitive ratio.

In the online version of the problem, the requests are revealed one by one. The online algorithm has to make

a decision on whether to accept or reject a request. The algorithm's decisions cannot be changed in the future, and the decision must be made before the next request is revealed, so the online algorithm works without the prior knowledge of requests as opposed to the offline ILP optimization problem. In the case where multiple requests arrive at the same time, the online algorithm can consider the requests in any order. The order does not affect the analysis of our proposed algorithm.

To measure the performance of the online algorithm against the offline one, we use the concept of *Competitive Ratio*. Other works have used the concept of competitive ratio, but for different problems such as energy efficiency [23] or online routing [24]. Competitive ratio is defined as the performance achieved by the offline algorithm to that achieved by the online algorithm, i.e., if we denote the offline performance as P_{off} and the online performance as P_{on} , the competitive ratio is:

$$\sup_{\text{all input sequences}} \frac{P_{off}}{P_{on}}.$$

As the ratio gets closer to 1, the online performance gets closer to the offline performance. In other words, the smaller the competitive ratio, the better the online algorithm's performance.

A. The Algorithm

Before presenting the online algorithm, we define the following:

- The relative load on a processing vehicle v at time τ when β_j arrives is

$$\lambda_v(\tau, j) = \sum_{\substack{k:k < j \\ k \in Process_v(\tau)}} \frac{r_k}{D_v},$$

where $k < j$ refers to the indices of all β_k that are being processed at vehicle v at the time when considering β_j to be accepted or not at vehicle v . We use $k \in Process_v(\tau)$ to represent the processing of β_k at vehicle v at time τ .

- The cost of processing a message β_j at vehicle v at time τ is:

$$C_v(\tau, j) = D_v(\gamma^{\lambda_v(\tau, j)} - 1),$$

where γ is a constant that will be defined in Section VI.

The core idea of the online algorithm is to assign an exponential cost function for the vehicle in terms of the vehicle's relative load. If the cost of processing a new message is less than the severity level of that message, the algorithm decides to process the new message. The choice of an exponential cost function guarantees that the vehicle's capacity constraints are not violated. We show that in the next Section. The online algorithm is presented in Algorithm 1.

In the algorithm, when a new request arrives at time $t_j(0)$, vehicle v computes the relative load ($\lambda_v(\tau, j)$) and the cost ($C_v(\tau, j)$) for every $\tau \in \{t_j(0), \dots, t_j(0) + T_j\}$. This is because a message currently being processed may be done before $t_j(0) + T_j$, thus the relative load and the cost should be adjusted for each time slot thereafter.

Algorithm 1 Online Algorithm

New request for β_j arriving at vehicle v at time $t_j(0)$
 $\forall \tau \in \{t_j(0), \dots, t_j(0) + T_j\}$, Compute $C_v(\tau, j)$

if $p_j \geq \sum_{\tau=t_j(0)}^{t_j(0)+T_j} \frac{r_j}{D_v} C_v(\tau, j)$ **then**

 process β_j on vehicle v

$\forall \tau \in \{t_j(0), \dots, t_j(0) + T_j\}$, $\lambda_v(\tau, j+1) = \lambda_v(\tau, j) + \frac{r_j}{D_v}$

else

 Do not process

end if

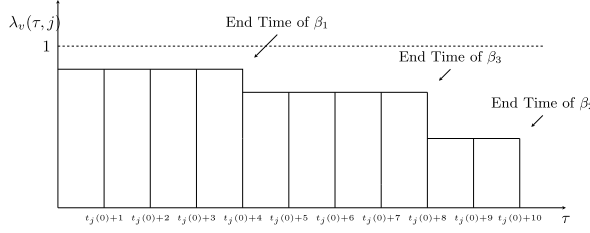


Fig. 3. Relative load and cost calculation example.

For example, Figure 3 shows the relative load at a vehicle for the next 10 time slots starting from $t_j(0)$, which is the arrival time of a new request β_j . The vehicle is processing three accepted requests β_1, β_2 , and β_3 that are going to finish at times $t_j(0) + 4$, $t_j(0) + 10$, and $t_j(0) + 8$, respectively. When a request β_j arrives at this vehicle at $\tau = t_j(0)$ and the duration of processing β_j is 10 time slots (i.e. $T_j = 10$), the cost calculation should include three accepted requests for 4 time slots, two accepted requests for 4 time slots, and one accepted request for 2 time slots. *Note that the relative load at every time slot required to calculate the cost is available to the vehicle at the arrival time of the new request, since these relative loads are calculated based on the already-accepted requests.* Referring to Figure 3, if we assume that the relative load for this vehicle is 0.75 for the first 4 time slots, 0.5 for the next 4 time slots, and 0.25 for the last two time slots, then the total cost is $\sum_{\tau=t_j(0)}^{t_j(0)+4} (D_v(\gamma^{0.75} - 1)) + \sum_{\tau=t_j(0)+4}^{t_j(0)+8} (D_v(\gamma^{0.5} - 1)) + \sum_{\tau=t_j(0)+8}^{t_j(0)+10} (D_v(\gamma^{0.25} - 1))$. If the severity level p_j of processing request β_j is higher than the total cost multiplied by $\frac{r_j}{D_v}$, then the request is accepted, and the relative load is increased by $\frac{r_j}{D_v}$ for all the 10 time slots. Otherwise, the request is rejected.

B. Illustrative Example

In this example, suppose that a vehicle can handle requests with a rate up to 40 packets per time slot and this vehicle represents the receiver in this example. In addition, suppose that there are three other vehicles within the communications range of this vehicle and these vehicles represent senders. Furthermore, suppose that γ is 29. Initially, the receiver's relative load is 0, since the receiver is not serving any request yet and the cumulative severity is also zero. Consequently, the following steps will occur when senders start sending requests to the receiver:

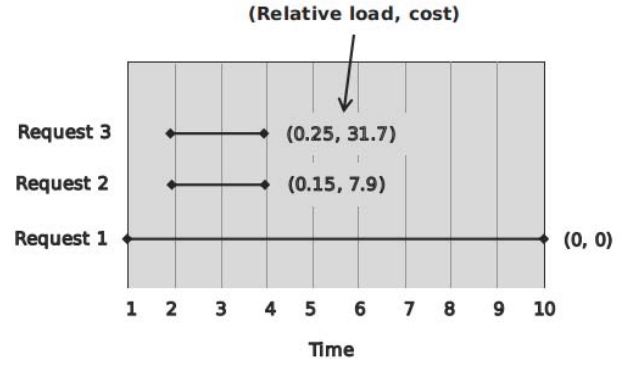


Fig. 4. Illustrative example of the proposed online algorithm.

Step 1: at time slot 1, the receiver receives the request (80, 6, 10, 1) which arrives at time slot 1 and finishes at time slot 10, asking for a processing rate of 6 packets per time slot and has a severity of 80. Now, since the relative load of the receiver is zero at the beginning then the minimum required severity to accept a bid is 0, so the request is accepted and the cumulative severity becomes 80. Also, the relative load is updated for the period from time slot 1 to time slot 10 to be 0.15

Step 2: at time slot 2, the request (12, 4, 3, 2) arrives requesting a processing rate of 4 packets per time slot with a severity of 12 and finishes at time slot 4. The relative load at time slots 2, 3, and 4 is 0.15 and the cost for this request is 7.9. Thus, since the severity is more than the computed cost, the request is accepted and the cumulative severity becomes 92. Also, the relative load is updated for the period from time slot 2 to time slot 4 to be 0.25 while it stays at 0.15 for time slot 1 and for the period from 5 to 10.

Step 3: at time slot 2, the request (30, 8, 3, 2) arrives with a severity of 30 and requesting a processing rate of 8 packets per time slot for 3 time slot. The relative load from time slot 2 to 4 is 0.25 and the cost of this request is 31.7 which is more than the severity of the request, so the request gets rejected. Since the relative load gets higher, the cost goes up in relation with the time period because the online algorithm tries to preserve the capacity for future requests with higher severity (c.f. Figure 4).

In the above example, the third request is rejected even when the severity is higher than the second request. However, compared to the first request, the severity of the third request is relatively low. The objective of the system is to maximize the cumulative severity which supports the safety of vehicles because high severity requests have a high potential of being accepted unless the residual capacity is low. Thus, emergency messages like safety messages have greater acceptance ratio than lower severity messages. However, when the system is heavily loaded with many requests being processed, the system may reject high severity requests in order to keep the system healthy and not to jeopardize the processing of already accepted requests. Note that this auctioning process is applied to the residual processing capability in the receiving vehicle. This is beyond the processing capacity guaranteed to all neighboring vehicles.

VI. PERFORMANCE ANALYSIS

In this Section, we show that our algorithm does not violate the capacity constraints, and achieves an $\mathcal{O}(1)$ -competitive ratio.

Before we start the proof of satisfying the capacity constraints and the competitive ratio, we need to state the following two assumptions:

$$1 \leq \frac{p_j}{r_j T_j} \leq F \quad \forall j, \quad (3)$$

and

$$r_j \leq \frac{D_v}{\log(\gamma)} \quad \forall j, \quad (4)$$

where F is any constant that is large enough to satisfy the assumption in (3), $\gamma = 2(TF + 1)$, and $T = \max(T_j), \forall j$. The assumption in (3) states that the severity level of a message scales with the required processing rate and duration. The assumption in (4) requires that the processing capacity of any vehicle should be greater than the required processing rate of any message, which is a practical condition to assume.

We start by proving that the online algorithm does not violate the capacity constraints. After that, we show that the online algorithm achieves a $\mathcal{O}(1)$ -competitive ratio. In all of the subsequent proofs, $\tau \in \{t_j(0), \dots, t_j(0) + T_j\}$.

Proposition 1: The Online algorithm does not violate the capacity constraints.

Proof: Let β_j be the first request that caused the relative load at vehicle v to exceed 1. By the definition of the relative load, we have

$$\lambda_v(\tau, j) > 1 - \frac{r_j}{D_v}$$

using the assumption in (4) and the definition of the cost function, we get

$$\begin{aligned} \frac{C_v(\tau, j)}{D_v} &= \gamma^{\lambda_v(\tau, j)} - 1 \\ &\geq \gamma^{1 - \frac{r_j}{D_v}} - 1 \\ &\geq \gamma^{1 - \frac{1}{\log \gamma}} - 1 \\ &\geq \frac{\gamma}{2} - 1 \geq TF \end{aligned}$$

Multiplying both sides by r_j and using the assumption in (3), we get

$$\frac{r_j}{D_v} C_v(\tau, j) \geq T F r_j \geq p_j$$

From the definition of our algorithm, β_j should not be processed at vehicle v . Therefore, the online algorithm does not violate the capacity constraints. $\square$

The next lemma shows that the cumulative severity gained by our algorithm is lower bounded by the sum of the processing costs.

Lemma 1: Let A be the set of indices of requests accepted by the online algorithm, and k be the last index, then

$$2 \log(\gamma) \sum_{j \in A} p_j \geq \sum_{\tau} C_v(\tau, k + 1)$$

Proof: We prove this lemma by induction on k . When $k = 0$, the vehicle is not processing any request and the right hand side of the inequality is 0. When β_j is not accepted by the online algorithm, neither side of the inequality is changed. Then it is enough to show, for an accepted request β_j , that:

$$2 \log(\gamma) p_j \geq \sum_{\tau} [C_v(\tau, j + 1) - C_v(\tau, j)]$$

since summing both sides over all $j \in A$ will yield (1).

The additional cost incurred by processing β_j is given by:

$$\begin{aligned} C_v(\tau, j + 1) - C_v(\tau, j) &= D_v [\gamma^{\lambda_v(\tau, j+1)} - \gamma^{\lambda_v(\tau, j)}] \\ &= D_v \gamma^{\lambda_v(\tau, j)} [\gamma^{\frac{r_j}{D_v}} - 1] \\ &= D_v \gamma^{\lambda_v(\tau, j)} [2^{\log \gamma \frac{r_j}{D_v}} - 1] \end{aligned}$$

Since $2^x - 1 \leq x$ for $0 \leq x \leq 1$ and using the assumption in (4)

$$\begin{aligned} C_v(\tau, j + 1) - C_v(\tau, j) &\leq D_v \gamma^{\lambda_v(\tau, j)} [\frac{r_j}{D_v} \log \gamma] \\ &\leq r_j \log \gamma [\frac{C_v(\tau, j)}{D_v} + 1] \\ &\leq \log \gamma [\frac{r_j}{D_v} C_v(\tau, j) + r_j] \end{aligned}$$

Summing over τ and the fact that β_j is processed, we get

$$\begin{aligned} \sum_{\tau} [C_v(\tau, j + 1) - C_v(\tau, j)] &\leq \log \gamma \sum_{\tau} [\frac{r_j}{D_v} C_v(\tau, j) + r_j] \\ &\leq \log \gamma [p_j + \sum_{\tau} r_j] \\ &\leq 2 \log(\gamma) p_j \end{aligned}$$

$\square$

Lemma 2: Let Q be the set of indice of requests accepted by the offline algorithm, but not the online algorithm. Let $l = \arg \max_{j \in Q} (C_v(\tau, j))$. Then

$$\sum_{j \in Q} p_j \leq \sum_{\tau} C_v(\tau, l)$$

Proof: Since β_j was not processed by the online algorithm, we have:

$$p_j \leq \sum_{\tau} \frac{r_j}{D_v} C_v(\tau, j) \leq \sum_{\tau} \frac{r_j}{D_v} C_v(\tau, l)$$

Summing over all $j \in Q$

$$\sum_{j \in Q} p_j \leq \sum_{\tau} C_v(\tau, l) \sum_{j \in Q} \frac{r_j}{D_v} \leq \sum_{\tau} C_v(\tau, l)$$

Since any offline algorithm cannot exceed a unit relative load, $\sum_{j \in Q} \frac{r_j}{D_v} \leq 1$. $\square$

Combining Lemma 1 and Lemma 2, we have the following theorem.

Theorem 2: The online algorithm achieves an $\mathcal{O}(1)$ -competitive ratio.

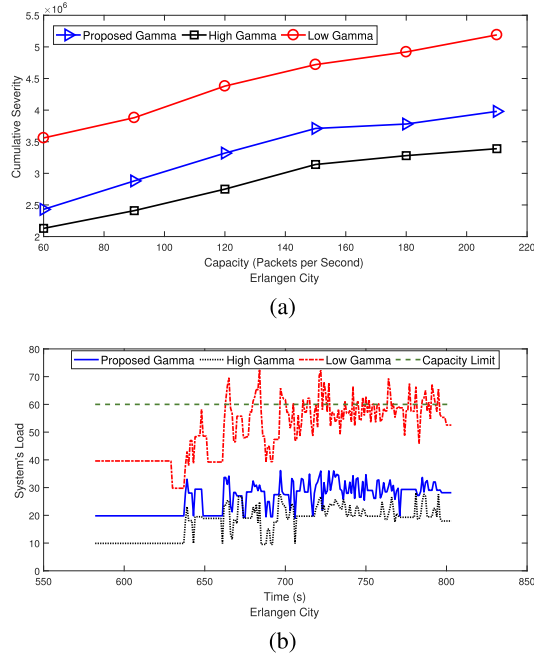


Fig. 5. Performance of the online algorithm for different γ values. (a) Cumulative severity for different γ values. (b) System load for different γ values.

Proof: Let A^* be the set of indices of the requests accepted by the offline algorithm, and let k be the last index. The cumulative severity of the offline algorithm is given by:

$$\begin{aligned}
 \sum_{j \in A^*} p_j &= \sum_{j \in Q} p_j + \sum_{j \in A^*/Q} p_j \\
 &\leq \sum_{j \in Q} p_j + \sum_{j \in A} p_j \\
 &\leq \sum_{\tau} C_v(\tau, l) + \sum_{j \in A} p_j \\
 &\leq \sum_{\tau} C_v(\tau, k+1) + \sum_{j \in A} p_j \\
 &\leq 2 \log(2\gamma) \sum_{j \in A} p_j
 \end{aligned}$$

Since γ is a constant defined at the beginning of this section, our algorithm achieves an $\mathcal{O}(1)$ -competitive ratio. $\square$

The value of the variable γ , the definition of the cost function, the steps of the algorithm, and the assumptions mentioned in Section V are all tied together in order to achieve a constant competitive ratio and satisfy the capacity constraints as shown in the proofs in Section VI. Any other value of γ may result in a worse competitive ratio or violating the capacity constraints.

VII. VALIDATION EXPERIMENTS

In Section VI, we defined the value of γ needed to achieve an $\mathcal{O}(1)$ -competitive ratio. To validate our analytical results, we conduct simulation experiments using different values of γ ; namely, values that coincide with our analytical results as well as lower and higher values. The results of our experiments are illustrated in Figure 5.

From Figure 5(a), it can be clearly seen that the algorithm gains less profit in terms of cumulative severity when a higher value of γ is used compared to our analytical expression. Furthermore, Figure 5(b) illustrates that lower values of γ compared to our analytical expression allow the algorithm to accept more requests than what can be handled by the system; thus, violating the capacity constraint. Figure 5 also illustrates how the value of γ generated based on our analytical expression achieves better cumulative severity while not violating the capacity constraint of the system.

VIII. EXPERIMENTAL PERFORMANCE ANALYSIS

To evaluate the performance of the online algorithm, we compare it against three heuristic algorithms. The first algorithm, named FirstFit, accepts requests as long as the capacity permits without taking the severity of requests into consideration. The other two algorithms are threshold-based heuristic algorithms. The threshold is updated based on the exponentially weighted moving average (EWMA) filter using the following equation:

$$\theta_j = (1 - \alpha) \cdot \theta_{j-1} + \alpha \cdot p_j$$

where θ_j is the newly computed threshold value for request j , θ_{j-1} is the previous estimation computed from request $j-1$, p_j is the current severity of request j , and α is a parameter that determines the reactivity of the algorithm. The two heuristic algorithms are:

- 1) Agile: where $\alpha = 0.7$.
- 2) Stable: where $\alpha = 0.3$.

A higher value of α means that the new threshold mostly depends on the current severity and therefore the algorithm will respond quickly to changes. Both the agile and the stable algorithms accept a request if the severity of the received request is higher than the computed threshold value and the capacity constraint is not violated; otherwise, the request is rejected.

In order to compare the performance of the aforementioned algorithms, we use the traffic traces generated as explained in Section VIII-A for two different cities: Bologna in Italy and Erlangen in Germany. These traffic traces contain the velocity of each vehicle, the start time and end time of the communication process, and the number of packets generated during the communication. These fields are used to produce the input data for the algorithms as follows:

- 1) Arrival time of the j -th request ($t_j(0)$): This is the same as the start time of the communication process.
- 2) Duration of the j -th request (T_j): This is represented by the difference between the end time and the start time of the communication process.
- 3) Processing rate of the j -th request (r_j): This is the number of packets of the j -th request divided by the Duration of the j -th request.
- 4) Severity of the j -th request (p_j): This is computed based on the following equation:

$$p_j = \frac{|U_s - U_r|}{U_a} \cdot M$$

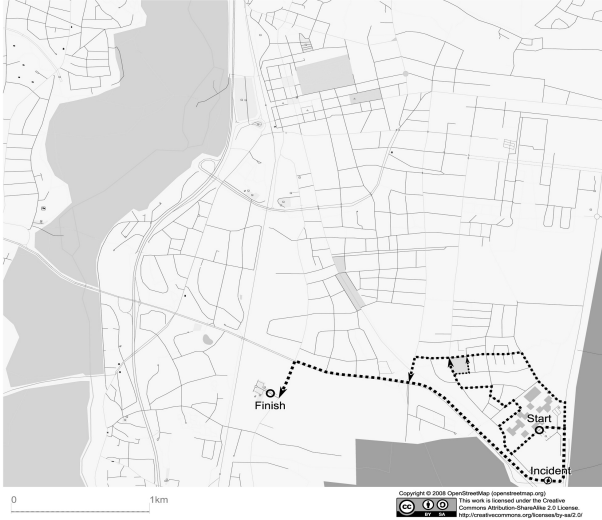


Fig. 6. Map of Erlangen, Germany, as available from the OpenStreetMap project. Overlaid are the scenarios start and finish positions, the fastest route between them, and popular alternative routes around the artificial incident. [27].

Where U_s is the stopping distance of the sending vehicle, U_r is the stopping distance of the receiving vehicle, U_a is the actual distance between sending and receiving vehicles, and M is a scaling factor. The stopping distances U_s and U_r are computed using the following equation [25]:

$$U = Vel \times R + \frac{Vel^2}{2\Delta g}$$

Where Vel is the vehicle's velocity, R is the driver reaction time in seconds which is set to 1.5, Δ is the friction coefficient which is set to 0.7, and g is the gravity of earth.

Our complete emulation traces can be found in [26]. Simulation settings and results obtained from running the three algorithms based on the simulation data are discussed next.

A. Simulation Settings

In order to generate realistic vehicular data for our proposed online algorithm, we simulate two different scenarios using Veins (Vehicles in Network Simulation) [27], because direct practical experiments are not feasible. The hybrid simulation framework, Veins, uses Omnet++ [28], a network simulator, and SUMO [29], a road traffic simulator, as its principle components to simulate network traffic and mimic traffic behavior.

The first scenario to simulate Inter-Vehicular Communications (IVC) was traffic in Erlangen city in Germany. Erlangen city simulation scenario is the default scenario that is coupled with Veins. The base road layout for the city is retrieved from the publicly available OpenStreetMap project [27]. The project produces detailed maps of cities by combining data from different sources. Figure 6, shows a detailed map of Erlangen with overlaid locations of traffic source and sink nodes.



Fig. 7. Bologna city map [30].

The second IVC simulation scenario was for traffic from Bologna city in Italy. To conduct this simulation, we used Bologna Ringway dataset [30]. Bologna is a city of 380,000 people located in central northern Italy. As shown in Figure 7, the urban area is surrounded by a ringway that features three lanes per direction. Morning traffic peak hours occur due to the flow of traffic from the city's outskirts to the city's commercial hub, downtown, where majority of commercial activities occurs. Thus, most of the vehicular traffic flows along the ringway [30]. Figure 7 also shows a distribution of trips and their starting and ending edges. As can be seen from the figure, a few of the edges are prominent in most of the trips. This is an important fact, as it demonstrates that certain routes are used more than others and as such, traffic on those routes would be much more severe than other routes. Heavy traffic means that a larger number of vehicles will be exchanging data; thus, increasing the processing load and the need for prioritization.

Bedogni *et al.* [30] generate four different mobility datasets, based on the number of vehicle trips. In our simulation, we use the complete travel demand dataset that maps to peak traffic hours between, 8 A.M. and 9 A.M.; with a total of 22,213 trips. Veins simulates the complete network stack for all communications among vehicles. Each vehicle has a wireless module that utilizes IEEE 802.11p network cards transmitting at 3 Mbps. Beacon generation rate was set to 10 Hz [31]–[33]. This rate also provides more saturated channel and its preferable instead of higher data rates due to the lower capture threshold [31]. Although the dynamic generation rate is recommended for VANETs in order to prevent network congestion [32], but since there is no explicit generation rate specified in the standard [34] we tried to adhere to the most common value used in literature. The complete parameter list is given in Table I. It should be noted that the selection of these datasets does not affect the results. We chose these two cities because their datasets are publicly available. The only parameter that affects the results is the number of cars present during simulation.

B. Results and Discussions

For validation purposes, we compare our online algorithm against the ILP optimization presented in Section IV-A.

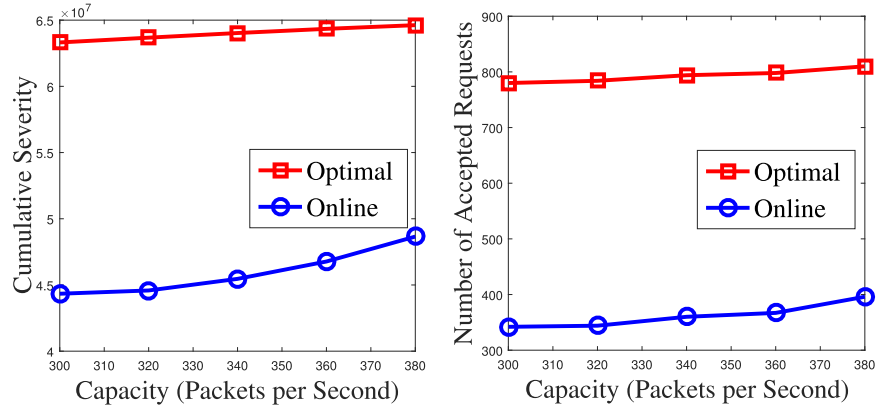


Fig. 8. Performance vs. capacity: online algorithm vs. ILP.

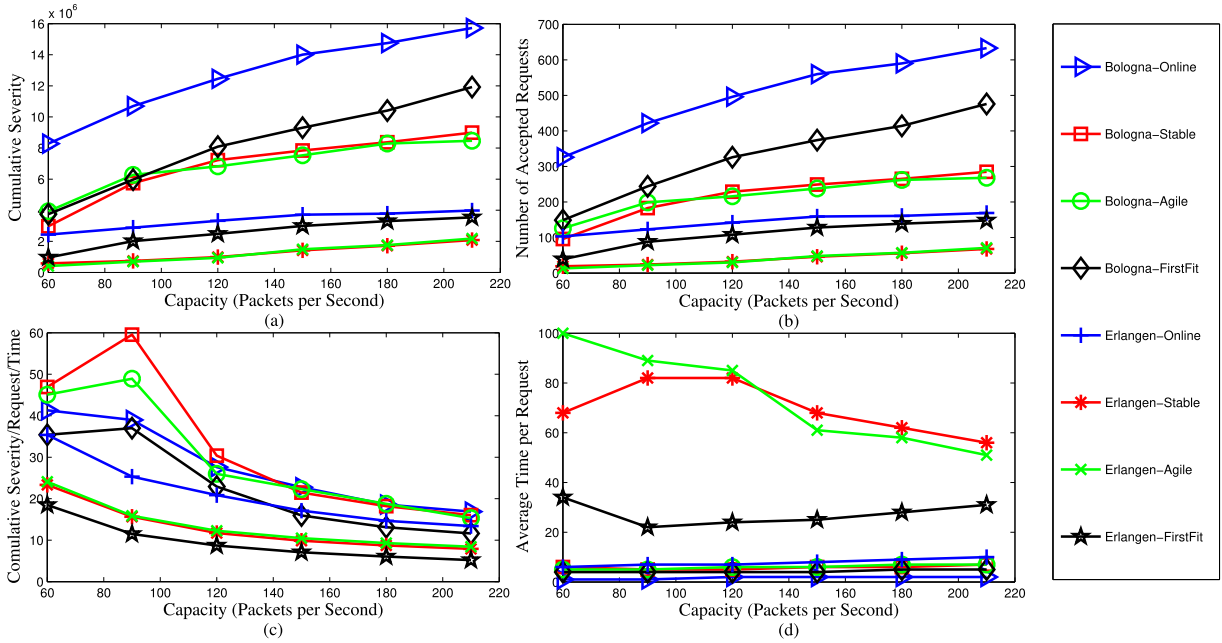


Fig. 9. Performance vs. capacity: online algorithm vs. agile, stable, and FirstFit online heuristics.

TABLE I
SIMULATION PARAMETERS

Parameters	Values
nic.phy80211p.txPower	20mW
nic.phy80211p.maxTXPower	10mW
nic.phy80211p.sensitivity	-89dBm
nic.phy80211p.thermalNoise	-110dBm
nic.mac1609_4.bitRate	3Mbps & BPSK
node.appl.headerLength	256 bit
node.appl.beaconLengthBits	544 bit
node.appl.beaconInterval	0.1 second
rsu.appl.sendBeacons	false

Due to the intractability of solving the ILP optimization, we run the simulations using a small dataset containing 800 requests from the Bologna case. The results are shown in Figure 8. As can be seen from the figure, a 30% to 40% increase in the cumulative severity gained by the optimal algorithm over the cumulative severity gained by the online algorithm is observed, which is inline with the worst-case

ratio proved in Section VI. Moreover, as the capacity of the processing vehicle increases, the gap between the online solution and the optimal solution decreases.

In our experiments, the four algorithms were tested using different capacities. Figure 9 and Figure 10 show the results for Bologna and Erlangen cities.

In the first experiment, we studied the cumulative severity for the four algorithms over different capacities (c.f. Figure 9(a)). This experiment demonstrates the superior performance of the proposed online algorithm over the other algorithms by up to 492% when the capacity is limited. As the capacity increases, the gap in cumulative severity between the proposed algorithm and the other three algorithms decreases. Actually, with large capacity, any algorithm can accept all the requests, which makes the problem simple and eliminates the need for an intelligent algorithm.

In our second experiment, we studied the number of accepted requests for the four algorithms over different

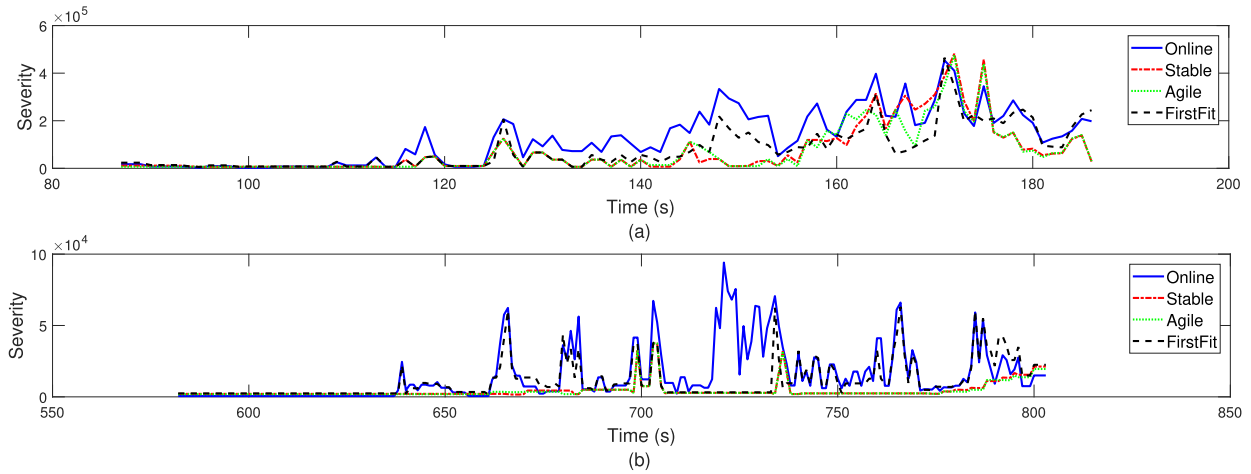


Fig. 10. Severity vs. time for capacity of 350 packets per second. (a) Bologna city. (b) Erlangen city.

capacities (c.f. Figure 9(b)). While this experiment shows that the proposed algorithm accepts more requests than the heuristic-based algorithms, the selection of the requests is not random. In other words, the online algorithm does not have to accept more requests to get higher cumulative severity. The advantage of the online algorithm over other algorithms is that it tends to accept requests with higher severity that span shorter periods of time in order to preserve the capacity for future potential requests as can be clearly seen in Figure 9(d).

Figure 9(d) compares the average time per request for the four algorithms using different capacities. The figure supports the fact that the proposed algorithm does not accept more requests randomly but rather makes intelligent decisions by accepting requests that tend to span shorter periods. In other words, the proposed algorithm has a better utilization of time compared to the other three algorithms because it does not allow a request to reserve the capacity for longer periods.

In Figure 10, the capacity is set to 120 packets per second. In each time slot, the sum of the severity levels of the accepted requests in that time slot for the four algorithms is plotted. The agile and stable algorithms accept requests that exceed their identified thresholds regardless of their time span; therefore, they end up accepting some requests that reserve the capacity for long periods preventing both algorithms from accepting further requests. Likewise, the FirstFit algorithm falls in the same trap and ends up accepting requests that reserve the capacity for long time. This implies that the three algorithms might end up regretting their previous decisions as illustrated in Figure 10 compared to our proposed online algorithm which does not regret previous decisions.

Figure 9(c) plots the cumulative severity per accepted request per unit of time for the four algorithms over different capacities. This figure provides the insight about how the online algorithm achieves its “no regret” behavior by accepting requests with higher severity that tend to span shorter intervals. Furthermore, Figure 9(c) shows that the proposed online algorithm is responsive to the system’s capacity as it accepts more requests when the system’s capacity is increased while the other three algorithms are insensitive to capacity increases.

IX. CONCLUSION

In this paper, we study the problem of prioritizing the processing of message requests in vehicular networks, where a vehicle needs to select which requests it needs to process based on the severity level of the request. This is motivated by the urgency of data processing in safety-critical scenarios such as collision avoidance applications. We formulate the problem as an Integer Linear Program that requires a prior knowledge of the request sequence to maximize the cumulative severity without violating the capacity constraints of the serving vehicle, and prove that the problem is NP-complete. Since obtaining the request sequence in advance is hard in real scenarios, we propose an online algorithm that works on a per-request basis and does not require the prior knowledge of the request sequence. Through detailed analysis, we show that the cumulative severity gained by the online algorithm is within a constant factor of the optimal cumulative severity. Our evaluation, based on real traffic traces, shows that our algorithm can outperform exponentially weighted moving average threshold-based heuristics and FirstFit by up to 492%.

ACKNOWLEDGMENT

The statements made herein are solely the responsibility of the authors.

REFERENCES

- [1] *Global Status Report on Road Safety: Time for Action*. World Health Org., Geneva, Switzerland, 2009.
- [2] Intelligent Transportation Systems Joint Program Office. (1997). *Intelligent Vehicle Initiative—Business Plan*. [Online]. Available: <https://www.fhwa.dot.gov/publications/research/operations/its/jpo98007/ivibus.pdf>
- [3] F. D. Da Cunha, A. Boukerche, L. Villas, A. C. Viana, and A. A. F. Loureiro, “Data communication in VANETs: A survey, challenges and applications,” Ph.D. dissertation, INRIA, Paris, France, 2014.
- [4] *Standard Specification for Telecommunications and Information Exchange Between Roadside and Vehicle Systems—5 GHz Band Dedicated Short Range Communications (DSRC) Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, Standard ASTM E2213, 2010.
- [5] L. R. Raju and C. Reddy, “A survey on routing protocols and QoS in mobile ad hoc networks (MANETs),” *Indian J. Sci. Technol.*, vol. 10, no. 9, pp. 1–8, 2017.

- [6] D. Puthal, Z. H. Mir, F. Filali, and H. Menouar, "Cross-layer architecture for congestion control in vehicular ad-hoc networks," in *Proc. IEEE Int. Conf. Connected Vehicles Expo (ICCVE)*, 2013, pp. 887–892.
- [7] M. Y. Darus and K. A. Bakar, "Congestion control algorithm in VANETs," *World Appl. Sci. J.*, vol. 21, no. 7, pp. 1057–1061, 2013.
- [8] S. S. Salihin, R. M. Noor, and S. Ghahremani, "Dynamic congestion control algorithm for vehicular ad-hoc networks," *Int. J. Softw. Eng. Appl.*, vol. 7, no. 3, pp. 95–108, 2013.
- [9] X. Cheng, L. Yang, and X. Shen, "D2D for intelligent transportation systems: A feasibility study," *IEEE Trans. Intell. Trans. Syst.*, vol. 16, no. 4, pp. 1784–1793, Jan. 2015.
- [10] M. Boban, G. Misek, and O. K. Tonguz, "What is the best achievable QoS for unicast routing in VANETs?" in *Proc. IEEE Globecom Workshops*, Nov./Dec. 2008, pp. 1–10.
- [11] C. Suthaputchakun and Z. Sun, "Priority based routing protocol in vehicular ad hoc network," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, Jun./Jul. 2011, pp. 723–728.
- [12] V. Kumar, K. S. Vaisla, and S. D. Sudarsan, "Priority based data scheduling in VANETs," in *Proc. Int. Conf. Adv. Comput. Commun. Eng. (ICACCE)*, Nov. 2016, pp. 19–22.
- [13] K. Pandit, D. Ghosal, H. M. Zhang, and C. N. Chuah, "Adaptive traffic signal control with vehicular ad hoc networks," *IEEE Trans. Veh. Technol.*, vol. 62, no. 4, pp. 1459–1471, May 2013.
- [14] B. Priyadarshini and J. Dhivyadharini, "Platooning for adaptive traffic signal monitoring using a two phase approach," *Int. J. Adv. Res. Comput. Sci. Manage. Stud.*, vol. 2, no. 3, pp. 27–33, 2014.
- [15] X. Cheng, C. Chen, W. Zhang, and Y. Yang, "5G-enabled cooperative intelligent vehicular (5GenCIV) framework: When benz meets marconi," *IEEE Intell. Syst.*, vol. 32, no. 3, pp. 53–59, May/Jun. 2017.
- [16] G. Guo and S. Wen, "Communication scheduling and control of a platoon of vehicles in VANETs," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 6, pp. 1551–1563, Jun. 2016.
- [17] T.-Y. Wu, W.-T. Lee, T.-H. Lin, W.-L. Hsu, and K.-L. Cheng, "Using service delay for facilitating access point selection in VANETs," in *Proc. IARIA*, 2013, pp. 13–19.
- [18] S. Lakshmi and D. R. W. Banu, "Prioritized directional broadcast technique for message dissemination in VANETs," *J. Theor. Appl. Inf. Technol.*, vol. 68, no. 1, pp. 181–190, 2014.
- [19] P. D. Dorge, S. S. Dorle, M. B. Chakole, and D. K. Thote, "Improvement of QOS in VANET with different mobility patterns," in *Proc. IEEE Int. Conf. Radar, Commun. Comput. (ICRCC)*, Dec. 2012, pp. 206–209.
- [20] F. Zeng, R. Zhang, X. Cheng, and L. Yang, "Channel prediction based scheduling for data dissemination in VANETs," *IEEE Commun. Lett.*, vol. 21, no. 6, pp. 1409–1412, Jun. 2017.
- [21] S. Xu, P. Guo, B. Xu, and H. Zhou, "QoS evaluation of VANET routing protocols," *J. Netw.*, vol. 8, no. 1, pp. 132–139, 2013.
- [22] M. A. Salahuddin, A. Al-Fuqaha, and M. Guizani, "Exploiting context severity to achieve opportunistic service differentiation in vehicular ad hoc networks," *IEEE Trans. Veh. Technol.*, vol. 63, no. 6, pp. 2901–2915, Jul. 2014.
- [23] S. Albers and H. Fujiwara, "Energy-efficient algorithms for flow time minimization," *ACM Trans. Algorithms*, vol. 3, no. 4, p. 49, 2007.
- [24] P. Jaillet and M. R. Wagner, "Generalized online routing: New competitive ratios, resource augmentation, and asymptotic analyses," *Oper. Res.*, vol. 56, no. 3, pp. 745–757, 2008.
- [25] D. Fambro, K. Fitzpatrick, and R. Koppa, "NCHRP report 400: Determination of stopping sight distances," Transp. Res. Board, Nat. Res. Council, Washington, DC, USA, Tech. Rep. 400, 1997.
- [26] *Online Algorithm for Opportunistic Handling of Received Packets in Vehicular Networks: A Public Dataset*. Accessed: Mar. 6, 2018. [Online]. Available: <https://www.dropbox.com/sh/xha0p7uyued56i5/AAD6aRxQpqiSgEA9BrSqMvzVa?dl=0>
- [27] C. Sommer, R. German, and F. Dressler, "Bidirectionally coupled network and road traffic simulation for improved IVC analysis," *IEEE Trans. Mobile Comput.*, vol. 10, no. 1, pp. 3–15, Jan. 2010.
- [28] (2014). *Objective Modular Network Testbed in C++ (4.6)*. [Online]. Available: <https://omnetpp.org/>
- [29] Institute of Transportation Systems. (2016). *Simulation of Urban Mobility (0.22.0)*. [Online]. Available: <http://www.dlr.de>
- [30] L. Bedogni, M. Gramaglia, A. Vesco, M. Fiore, J. Härri, and F. Ferrero, "The bologna ringway dataset: improving road network conversion in SUMO and validating urban mobility via navigation services," *IEEE Trans. Veh. Technol.*, vol. 64, no. 12, pp. 5464–5476, Dec. 2015.
- [31] H. Hartenstein and L. P. Laberteaux, "A tutorial survey on vehicular ad hoc networks," *IEEE Commun. Mag.*, vol. 46, no. 6, pp. 164–171, Jun. 2008.
- [32] M. van Eenennaam, W. K. Wolterink, G. Karagiannis, and G. Heijenk, "Exploring the solution space of beaconing in VANETs," in *Proc. IEEE Veh. Netw. Conf. (VNC)*, Oct. 2009, pp. 1–8.
- [33] R. Reinders, M. van Eenennaam, G. Karagiannis, and G. Heijenk, "Contention window analysis for beaconing in VANETs," in *Proc. IEEE 7th Int. Wireless Commun. Mobile Comput. Conf.*, Jul. 2011, pp. 1481–1487.
- [34] J. B. Kenney, "Dedicated short-range communications (DSRC) standards in the United States," *Proc. IEEE*, vol. 99, no. 7, pp. 1162–1182, Jul. 2011.



Ala Al-Fuqaha (S'00–M'04–SM'09) is currently a Professor and the Director of the NEST Research Laboratory at the Department of Computer Science, Western Michigan University. His research interests fall in the areas of vehicular networks, dynamic spectrum access etiquettes in cognitive radio networks, smart services in support of the Internet of Things, and planning of software-defined networks. He currently serves on the editorial board of multiple journals.



Ammar Gharaibeh received the B.S. degree (Hons.) from the Jordan University of Science and Technology in 2006, the M.S. degree in computer engineering from Texas A& M University in 2009, and the Ph.D. degree from the Department of Electrical and Computer Engineering, New Jersey Institute of Technology, in 2017. He is currently an Assistant Professor with the Department of Computer Engineering, German Jordanian University. His research interests spans the areas of wireless networks and network caching.



Ihab Mohammed received the B.S. and M.S. degrees in computer science from Al-Nahrain University, Baghdad, Iraq, in 2002 and 2005, respectively. He is currently pursuing the Ph.D. degree with the Department of Computer Science, Western Michigan University. In 2005, he joined the Department of Computer Science, Al-Nahrain University, as an Assistant Lecturer and promoted to Lecturer in 2011. He is currently a Research Assistant with the NEST Research Laboratory. His current research interests include Internet of Things, vehicular networks, and big data.



Sayed Jahed Hussini received the B.S. degree from the Chemical Engineering Department, Balkh University, and the M.S. degree in computer science from the Technical University of Berlin in 2010. He is currently pursuing the Ph.D. degree with the Department of Computer Science, Western Michigan University.



Abdallah Khreishah (S'07–M'11–SM'18) received the B.S. degree in computer engineering from the Jordan University of Science and Technology in 2004, and the M.S. and Ph.D. degrees in electrical and computer engineering from Purdue University in 2006 and 2010, respectively. While pursuing his Ph.D. degree, he was with NEESCOM. He is the Chair of North Jersey IEEE EMBS Chapter. He is an Assistant Professor with the Department of Electrical and Computer Engineering, New Jersey Institute of Technology. His research interests fall in

the areas of visible-light communication, green networking, network coding, wireless networks, and network security.



Issa Khalil (S'04–M'09–SM'17) received the Ph.D. degree in computer engineering from Purdue University, USA, in 2007. He joined the College of Information Technology, United Arab Emirates University, where he served as an Associate Professor and the Department Head of the Information Security Department. In 2013, he joined the Cyber Security Group, Qatar Computing Research Institute, a member of Qatar Foundation, as a Senior Scientist, where he recently promoted to Principal Scientist. His research interests include wireless and wireline

network security and privacy. He is especially interested in cloud security, botnet detection and takedown, and security data analytics. His novel technique to discover malicious domains following the guilt-by-association social principle attracts the attention of local media and stakeholders. He served as an organizer, a technical program committee member, and a reviewer for many international conferences and journals. He is a member of ACM and delivers invited talks and keynotes in many local and international forums. In 2011, he was granted the CIT Outstanding Professor Award for outstanding performance in research, teaching, and service.