Severity-Based Prioritized Processing of Packets with Application in VANETs

Ala Al-Fuqaha, *Senior Member, IEEE*, Ihab Mohammed, *Student Member, IEEE*,
Sayed Jahed Hussini, *Student Member, IEEE*, and Sameh Sorour, *Senior Member, IEEE*

Abstract—To fully realize the potential of vehicular networks, several obstacles and challenges need to be addressed. Chief among the obstacles are strict QoS requirements of applications and differentiated service requirements in different situations. Although DSRC and WAVE have been adopted as the de facto standards, they do not address all the problems and there is room for improvements. In this study, we propose a generic prioritization and resource management algorithm that can be used to prioritize processing of received packets in vehicular networks. We formulate the generic severity-based prioritized packet processing problem as Penalized Multiple Knapsack Problem (PMKP) and prove that it is an NP-Hard problem. We thus develop a real-time heuristic that utilizes a relaxed version of the formulation. The relaxed formulation executes in polynomial time and guarantees a minimum delay per severity-level while respecting the processing rate constraint. To measure the performance of the proposed heuristic, real traffic data is used in a small-scale experiment. The proposed heuristic is tested against the PMKP solution and results show a small degradation of up to 4 percent in profit for the heuristic compared to the PMKP solution. Also, the proposed heuristic is tested against a non-prioritized processing algorithm that works using first come first served policy. Results show that the proposed heuristic gains 9 to 67 percent more profit than the non-prioritized processing algorithm in moderate and high congestion scenarios.

Index Terms—Context severity, prioritized processing, network calculus, VANETs

1 Introduction

1.1 Motivation

Dedicated short-range communications (DSRC) and Wireless Access in Vehicular Communications (WAVE) have become the de-facto vehicular communications technologies in VANETs. IEEE 802.11p and IEEE P1609 form the bases of the WAVE protocol as shown in Fig. 1 [1]. To address the unique challenges in VANETs (i.e., dynamic topology, short communication periods and application requirements), WAVE uses multi-channel operations to increase channel utilization, and differentiated service categories to provide Quality of Service (QoS). To provide multi-channel operations, WAVE utilizes the IEEE 1609.4 standard, which defines six Service Channels (SCH) and one Control Channel (CCH) for use in VANETs [2]. These channels have different frequencies, maximum transmission power and applications. The CCH for instance is used for transmitting safety and control data in order to insure fast and prioritized delivery of time critical data. Furthermore, to provide differentiated services, WAVE relies on IEEE 802.11p, which uses Enhanced Distributed Channel Access (EDCA) at the Medium Access Control (MAC) layer to provide differentiated QoS. The IEEE 802.11p EDCA specifies four different Access Categories (ACs): AC_VO (Voice or AC3), AC_VI (Video or AC2), AC_BE (Best effort or AC1) and AC_BK (Background or AC0), each of which has a different priority for accessing the communication medium [2], [3], [4]. This categorization offers low latency communications and differentiated services for applications. An overall schematic of multi-channel access and ACs assignment is shown in Fig. 2. We have to emphasize that the IEEE 802.11p scheme for prioritization only applies to applications during data transmission and does not affect the processing of data after its receipt. For more details on WAVE, please refer to [1], [5], [6].

The use of multi-channel access and data prioritization via ACs, as detailed in the EDCA standard, only guarantees prioritized access to the communications medium for broadcasting and sending data. However, this may not necessarily lead to improved system level performance, because each vehicle in the network has to process the received data based on its own context, not the transmitting vehicle's context. Assume the scenario shown in Fig. 3, where five different vehicles are involved. If vehicles B and C collide, then both vehicles would try to inform their neighbors by sending data using the highest priority AC. This flow of data is very important for vehicle "A", but not as important for vehicles "D" and "E". In such a scenario, vehicle "A" must prioritize the processing of packets that it receives from vehicle "B" (and/or vehicle "C"). On the

- *A. Al-Fuqaha is with the Information & Computing Technology Division, College of Science and Engineering (CSE), Hamad Bin Khalifa University (HBKU), Doha, Qatar, and the Department of Computer Science, Western Michigan University, Kalamazoo, MI 49008. E-mail: aalfuqaha@hbku.edu.qa.*
- *I. Mohammed and S.J. Hussini are with the Department of Computer Science, Western Michigan University, Kalamazoo, MI 49008. E-mail: {ihabahmedmoha.mohammed, sayedjahed.hussini}@wmich.edu.*
- *S. Sorour is with the Department of Electrical and Computer Engineering, University of Idaho, Moscow, ID 83844. E-mail: samehsorour@uidaho.edu.*

Manuscript received 5 Apr. 2018; revised 5 Sept. 2018; accepted 3 Jan. 2019. Date of publication 18 Jan. 2019; date of current version 6 Jan. 2020. (Corresponding author: Ala Al-Fuqaha.) Digital Object Identifier no. 10.1109/TMC.2019.2892980

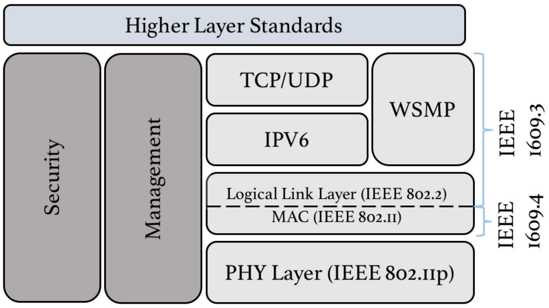


Fig. 1. WAVE stack.

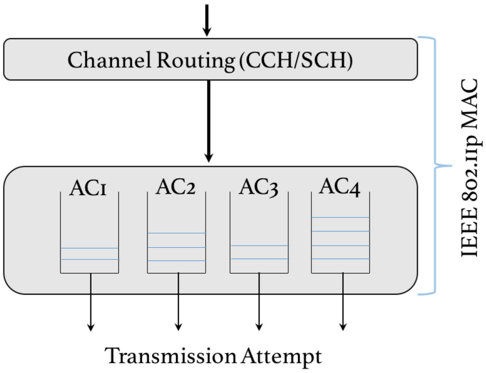


Fig. 2. User priority service in WAVE.

other hand, vehicles "D" and "E" do not necessarily benefit from processing the high priority safety application data flow(s) from vehicles "B" and/or "C" ahead of other flows with lower ACs, since the accident does not have impact on their mobility. In such scenarios, it is hence imperative that each vehicle processes the received data flows, not just based on their ACs, but also based on the flows impacts on the vehicles' mobility.

Prioritization of received data and management of computation resources are important tasks because autonomous and semi-autonomous vehicles have to process data from a wide range of sensors (e.g., LIDAR, GPS, compass, radar, infrared cameras) at any given moment, which in turn puts more computation overhead over the on-board computational resources. Beyond processing raw sensor data, the on-board computational resources must execute complex algorithms to perform proximity understanding and vehicle control.

Another example to illustrate the importance of the proper prioritization of received data flows is the security performance of basic safety messages (BSMs). To verify and sign BSMs, the standard recommends using the Elliptic Curve Digital Signature Algorithm (ECDSA). However, using ECDSA increases the processing overhead [7] and degrades the performance that could result in the loss of safety critical BSMs. This can become problematic especially in environments with higher density of vehicles. To solve this issue, the authors of [8] propose re-prioritizing signature verification of received BSMs based on location proximity of the sender's vehicle, such that nearby vehicles are assigned higher priority. This example demonstrates the importance of prioritized processing of data flows on the receiver's side in order to increase the overall performance of the system.

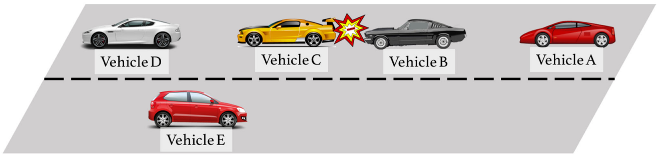


Fig. 3. Illustrative scenario.

From the discussion of the two previous scenarios, it is evident that achieving ideal performance in VANETs depends, not only on efficient algorithms and techniques for sending data, but also on a better prioritization of packets on the receiver side. This prioritization must be done according to a certain *severity metric* that corresponds to the nature of the application. The proposed algorithm prioritizes the processing of received data based on their impact on safety of a given vehicle.

1.2 Contributions

In this paper, we aim to design a real-time algorithm to opportunistically prioritize the processing of data flows based on a generic *severity* notion, such that the benefit to the overall system is maximized. Severity can thus be defined according to the application, as a metric (e.g., vehicle speed, direction, position) that underline the importance of the flow for the overall benefit of the system. The proposed algorithm models flow prioritization by classifying them according to their severity into multiple queues with different priority and capacity levels. For this proposed approach, we derive an upper bound on the delay of service for the flows classified in each queue using network calculus. We then formulate the problem of maximizing the severity of the admitted flows into the queue system as a penalized multiple knapsack problem with a service delay constraint on each of the queues. Being an NP-hard problem, we propose a real-time heuristic that divides the problem into sub-problems, each finding the optimal admission of flows into each of the queues such that the total severity of the admitted flows is maximized. Finally, both the optimal and heuristic solutions are tested using real traffic data and compared to un-prioritized first come first serve approaches.

1.3 Examples of Applications

The potential applications of our proposed algorithm can be numerous depending on the application's definition of *flow severity*. In the simplest form, flow severity can be defined as a QoS requirement that needs to be fulfilled. The dynamic topology of VANETs imposes strict QoS requirements on applications, especially the safety applications [9]. If severity is defined as application's QoS requirement, then the algorithm would strive to assign flows to queues such that flows with the most strict QoS requirements get higher priority. If severity is defined as the sending vehicle's proximity, then the algorithm would strive to provide prioritized service to flows that belong to the closest vehicles. In all these use cases, our algorithm acts as an admission control system, whereby it assesses the processing resources and incoming flows and strives to service the most severe flows first in order to achieve the best performance for the overall system.

Another important use of our proposed severity-based prioritization algorithm is to provide adaptive security in VANETs. To provide security in VANETs, WAVE relies on cryptographic solutions [6]. In order to have better security, it is important that more robust cryptographic solutions are used. However, using more robust cryptographic solutions leads to increased computations, increased cryptographic loss [7], and can have severe impact on the achieved QoS of applications [7], [10], [11]. To overcome this problem, adaptive security measures can be used, where a vehicle can determine the robustness of the cryptographic solution it uses for each flow based on the load and performance of the overall system. In such scenarios, our algorithm can classify four different cryptographic solutions ranked from the most (i.e., requires more computations) to the least robust, and determine which cryptographic solution would result in the best performance based on the flow, application QoS requirements, and performance of the overall system.

In conclusion, we are proposing an algorithm that can be adopted to support any application benefiting from intelligent prioritized processing of requests. In the rest of this paper, we focus on prioritized packet processing in VANETs.

1.4 Organization

The remainder of the paper is organized as follows: The related literature is reviewed in Section 2. The proposed prioritization model and system parameters are described in Section 3. In Section 4, we derive the delay upper bounds for the service of each of the queues. Section 5 shows the formulation of the problem in terms of the penalized knapsack problem, while Section 6 provides a heuristic solution. In Section 6.4, we provide the upper bound delay proof. Section 7 describes our experiments and results while conclusions are drawn in Section 8.

2 Related Work

WAVE QoS, service prioritization, and performance improvements are studied extensively in the literature. Generally, the studies and solutions offered can be divided into two broad groups: (1) studies that offer solutions from the *sender's* point of view and (2) studies that tackle the issue from the *receiver's* point of view. In the first group, the offered solutions range from spectrum sharing in order to prioritize safety message delivery [12], [13], [14], [15], [16] to MAC layer enhancements and admission control techniques in order to improve QoS and channel access [17], [18], [19], [20], [21], [22], [23], [24], [25]. Furthermore, there have been studies [26], [27], [28] that use vehicle's contextual information (i.e., vehicle's position) in order to ensure effective utilization of resources and provide prioritized services. The major difference between our work and these studies is that we tackle the issue from the receiver's point of view. We consider a receiver's perspective (as opposed to the sender's perspective) to achieve prioritized processing at the receiver (as opposed to prioritized transmission from the sender). Our algorithm deals with data that is received by a vehicle with the objective of opportunistically prioritize the processing of the received data for the overall benefit of the VANET.

While there have been numerous studies to improve QoS, service prioritization, and performance of WAVE from the sender's perspective, in the second group, there have not been many studies that address these issues on the receiver's side. The few studies that have been conducted in this area focus on security performance. Verification of time-sensitive BSMs in order to decrease cryptographic loss, by prioritizing the verification of BSMs based on the physical position of the sender, has been studied in [8], [29]. The authors of [30] have proposed to reduce the verification time of messages at the receiver side in dense areas by assigning different priority levels to nearby vehicles based on their physical parameters after verifying those vehicles. Our study in this paper extends these special cases in the aforementioned sources to a more general prioritized processing of the received packets in VANETs, as a function of a generic severity metric, in order to maximize the overall profit to the system while respecting its QoS constraints. The generic severity metric can be defined based on the application, such as physical proximity of vehicles as in [30] or more complex settings as in [28]. Therefore, our proposed solutions can apply to a variety of prioritized packet processing applications by properly defining its corresponding severity metric.

3 System Model

3.1 Prioritization Model

In this paper, we propose a real-time algorithm that allows vehicles in VANETs to intelligently prioritize the processing of the received packets based on their severity. Upon receipt of data flows, each flow is assigned a severity-level and passed on to the next phase for flow policing. This policing prevents flooding attacks as it regulates each flow via a token bucket filter to a predefined processing rate. The last phase is assigning the flows to one of N queues based on both the flow's severity and the capacity of the queue. These queues have different priority levels and capacities, such that "Queue N" has the highest priority and smallest capacity, while "Queue 1" has the lowest priority and the largest capacity. The size of each queue is determined by the computational load and delay requirements of it's assigned flows, and it dictates the maximum delay that will be incurred by the flows assigned to that queue. While assigning flows to queues, the algorithm strives to assign the highest severity flows to the highest priority queue to ensure their fast processing. The other flows are passed to their corresponding queues according to their severity levels. Once the capacity of a queue is reached, the algorithm proceeds to place the next batch of unassigned flows of the same severity to the subsequent lower priority queue. This process continues until all flows are assigned to queues or queues are all filled. If some flows are left out after all the queues were filled, the algorithm discards these remaining flows as these cannot be serviced in a timely manner. Although it's possible that some flows are discarded, our algorithm makes sure to serve the highest severity flows first while guaranteeing performance bounds for the

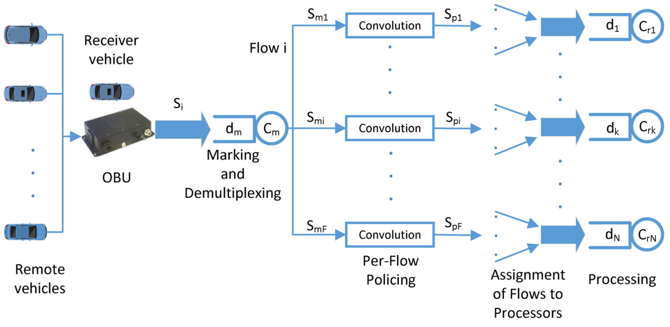


Fig. 4. System model.

accepted flows. The process of assigning flows to queues happens in snapshots. During every snapshot, which is done every Δt, the algorithm takes the available flows and assigns them to queues. The frequency of generating flow snapshots and the addition and removal of the flows to the snapshots are not handled in our algorithm. Rather, it assumes the availability of the snapshots and focuses on prioritizing flows available during each snapshot.

3.2 System Parameters

The system model and parameters is shown in Fig. 4. Assuming a VANET with V vehicles, where one vehicle $V_r \in V$ receives packets of length L (measured in bytes), sent from a set of sending vehicles $V_s = V \setminus V_r$ within its communications range. V_r receives all these packets via its On Board Unit (OBU), which has a channel reception rate R_b (measured in Mbps) and maximum burst length B (measured in packets). Let R be the rate in packets per second of the OBU (i.e., $R = \frac{R_b}{L}$). The received packets are then passed to the marking queue of the marking processor, which is assumed to be the On-Board Unit (OBU) processor. The marking processor assigns to each incoming packet a priority (severity) level based on the receiving vehicles perspective. The marking service requires simple computations (e.g., marking the packets based on the type of application, relative position, speed and direction of the sending vehicle); thus, induces a very small and almost negligible processing delay. The marking processor uses the packets' source MAC address in order to demultiplex the incoming packets into a set of flows $\mathcal{F}$ (i.e., a flow is defined between a source-destination pair of vehicles). The marking process applied to the scenario shown in Fig. 3, would be as follows: When vehicles B and C collide, they transmit data about collision to their surrounding vehicles. Since vehicle A is affected by this collision, when it receives data about the collision, based on its context severity, its marking processor will mark the received data about the collision with highest severity category and processes it accordingly. But since the collision doesn't affect vehicles D and E, when they receive this data, their marking processors would mark it with a lower severity level (severity level 2, 3, or 4 based on the defined context severity). Let S_i be the serverity level of the packets in flow i, $i \in \{1,\ldots,|\mathcal{F}|\}$. We assume M severity levels. Packets are assigned to flows in order to enforce per-flow policing, which is an important measure in order to protect the system from flow-based unfair exploitation (e.g., flooding attacks). The policing phase limits the per-flow rate and burstiness based on the severity level. Flow i, $i \in \{1,\ldots,|\mathcal{F}|\}$, with severity level S_i has a maximum rate and burstiness of $(\rho^i_{max}, \sigma^i_{max})$. Regardless of the traffic from neighboring vehicles, all flows are thus forced to respect the per-severity limits such that flows violating the limits are dropped. Finally, the last phase is the processing phase where flows are assigned to N queues. "Queue N" has the highest priority and the least capacity, which is appropriate for serving high severity flows. On the other end, "Queue 1" has the lowest priority and highest capacity. Although, "Queue N" may best serve flows of high severity, but this queue may admit flows of low severity depending on the available flows in order to maximize the utilization of system resources. Processor k, $k \in \{1,\ldots,N\}$, has a fixed processing rate of C_k and queue k, $k \in \{1,\ldots,N\}$, has a guaranteed maximum delay of d_{qk}. Additionally, the total delay induced by processing all accepted flows admitted to queue k must be less than or at most equal to the guaranteed maximum delay d_{qk} as explained later in Equation (9). The objective is to serve the maximum number of requests while providing the highest quality of service and maintaining guaranteed delay bounds in all queues.

4 Delay Upper-Bounds

Since the OBUs are the sources of data and as long as the maximum transfer rate is known, the output of the ith transmitting OBU can be upper bounded, using network calculus tools, by an affine arrival curve (A_i) that is defined as [31]

$$A_i \sim (B_i, R_i). \tag{1}$$

Consequently, the ith sending OBU has burstiness B_i and rate R_i. The sum of burstiness of all OBU flows equals the received burstiness at the destination OBU as shown in the following equation:

$$B = \sum_{i=1}^{F} B_i.$$

Also, the sum of the rates of all OBU flows equals the received rate at the destination OBU as shown in the following equation:

$$R = \sum_{i=1}^{F} R_i.$$

Thus, the input to each receiving OBU D_i is upper bounded by an affine arrival curve

$$D_i \sim (B, R). \tag{2}$$

Since the input to the marking process is bounded by (2) and the marking service rate is C_m, then the delay for the marking phase d_m is

$$d_m = \frac{B}{C_m - R}.$$

The output of the marking phase D_{m_i} consists of $|\mathcal{F}|$ flows that is upper bounded by

$$D_{m_i} \sim \left(B_i + R_i \cdot \frac{B}{C_m - R}, R_i\right),$$

which can be reduced to the following form:

$$D_{m_i} \sim \left(\frac{B_i(C_m - R) + R_i B}{C_m - R}, R_i\right). \tag{3}$$

The policing process of the ith flow D_{p_i} is expressed by the following bound:

$$\begin{aligned} D_{p_i} \sim &\left(\left(\frac{B_i(C_m - R) + R_i B}{C_m - R}, R_i\right) * \right. \\ &\left.\left(\sum_{j=1}^{M} \sigma_{max}^j f_i^j, \sum_{j=1}^{M} \rho_{max}^j f_i^j\right)\right), \end{aligned} \tag{4}$$

where f_i^j is a flag that is set to 1 to indicate that the ith flow belongs to severity level j; zero otherwise. Since there exists M severity levels and each flow is assigned to only one of them, we have the following constraint on these flags:

$$\sum_{j=1}^{M} f_i^j = 1 \quad \forall i.$$

The $*$ operator in (4) represents the min-plus convolution operator, which is an infimum operation over the addition of the bursitness and the service. Consequently, Equation (4) can be rewritten as

$$\begin{aligned} D_{p_i} \sim \alpha(t) = \min &\left\{\frac{B_i(C_m - R) + R_i B}{C_m - R} + R_i t, \right. \\ &\left. \sum_{j=1}^{M} \sigma_{max}^j f_i^j + \sum_{j=1}^{M} \rho_{max}^j f_i^j t\right\}, \end{aligned}$$

which is equivalent to

$$\begin{aligned} D_{p_i} \sim &\left(\min\left(\frac{B_i(C_m - R) + R_i B}{C_m - R}, \sum_{j=1}^{M} \sigma_{max}^j f_i^j\right), \right. \\ &\left. \min\left(R_i, \sum_{j=1}^{M} \rho_{max}^j f_i^j\right)\right). \end{aligned} \tag{5}$$

Since we are looking for the upper bound, we can assume the maximum traffic-per-flow. In other words, only the maximum burstiness and rate per severity are selected from Equation (5). Thus, the output of the policing process is upper-bounded by

$$D_{p_i} \sim \left(\sum_{j=1}^{M} \sigma_{max}^j f_i^j, \sum_{j=1}^{M} \rho_{max}^j f_i^j\right). \tag{6}$$

Each flow i is assigned to one of the N prioritization queues of the system. The flag g_i^k is used to indicated these assignments, such that g_i^k is set to 1 to indicate that the ith flow is assigned to Queue k; otherwise it's set to zero. So, for any flow i we have

$$\sum_{k=1}^{N} g_i^k \leqslant 1 \qquad \forall i.$$

The delay during the processing phase d_{r_k} is per-queue, so the upper-bound on the delay for Queue k is computed using all flows that are processed by Queue k as

$$d_{r_k} = \frac{\sum_{i=1}^{|\mathcal{F}|} \sum_{j=1}^{M} \sigma_{max}^j f_i^j g_i^k}{C_k - \sum_{i=1}^{|\mathcal{F}|} \sum_{j=1}^{M} \rho_{max}^j f_i^j g_i^k}. \tag{7}$$

The total delay per-queue d_{t_k} is the sum of the marking delay and the processing delay as shown below:

$$d_{t_k} = \frac{B}{C_m - R} + \frac{\sum_{i=1}^{|\mathcal{F}|} \sum_{j=1}^{M} \sigma_{max}^j f_i^j g_i^k}{C_k - \sum_{i=1}^{|\mathcal{F}|} \sum_{j=1}^{M} \rho_{max}^j f_i^j g_i^k}. \tag{8}$$

Note that "Queue N" has the highest priority and can handle high severity flows faster than the other queues

5 Problem Formulation

As was mentioned earlier, our objective is to serve the maximum number of flows (i.e., vehicles) prioritized based on their severity while at the same time striving to provide the highest quality of service. However, these two goals might not always be in harmony. If we try to provide the highest quality of service (e.g., using the most robust cryptographic solution), it may result in some flows not being serviced within the required delay constraint. So, we have to find a balance between the quality of service that can be provided and the number of flows that can be serviced with priorities based on their severities. Thus, we formulate this problem as a Penalized Multiple Knapsack Problem (PMKP) with the goal of maximizing the profit (i.e., the total severity of the admitted flows). The next sections introduce the formulation of the problem of maximizing the profit of the system in terms of the total sum of severities of admitted flows, while fulfilling their corresponding delay constraints. In Section C, we show that the problem is NP-Hard.

5.1 Derivation of the Delay Constraints

Each queue has a delay d_{q_k}, and accepted flows must satisfy the condition that the total processing delay of accepted flows in Queue k must be less than the delay of that queue. This can be written mathematically as

$$d_{t_k} = \frac{B}{C_m - R} + \frac{\sum_{i=1}^{|\mathcal{F}|} \sum_{j=1}^{M} \sigma_{max}^j f_i^j g_i^k}{C_k - \sum_{i=1}^{|\mathcal{F}|} \sum_{j=1}^{M} \rho_{max}^j f_i^j g_i^k} \leqslant d_{q_k},$$

which states that the total delay-per-queue d_{t_k} must not exceed the kth queue delay d_{q_k}. The above inequality can be reduced to the following form:

$$\begin{aligned} &\sum_{i=1}^{|\mathcal{F}|} \left(\sum_{j=1}^{M} \sigma_{max}^j f_i^j + d_{q_k} \sum_{j=1}^{M} \rho_{max}^j f_i^j - \frac{B}{C_m - R} \sum_{j=1}^{M} \rho_{max}^j f_i^j\right) \\ &\cdot g_i^k \leqslant C_k \left(d_{q_k} - \frac{B}{C_m - R}\right). \end{aligned} \tag{9}$$

5.2 PMKP Formulation

Before presenting the formulation, we define ψ as the penalty term that represents the highest severity among the dropped flows and order the set $\mathcal{F}$ based on severity in a non-increasing order.

The problem can thus be expressed as a PMKP as follows:

$$\max \sum_{k=1}^{N} \sum_{i=1}^{|\mathcal{F}|} S_i \cdot g_i^k \cdot k - \psi, \tag{10}$$

s.t.

$$S_i\left(1 - g_i^k\right) \leqslant \psi \quad \forall\, i, k \tag{11}$$

$$\sum_{i=1}^{|\mathcal{F}|}\left(\sum_{j=1}^{M} \sigma_{max}^j f_i^j + d_{q_k} \sum_{j=1}^{M} \rho_{max}^j f_i^j - \frac{B}{C_m - R} \sum_{j=1}^{M} \rho_{max}^j f_i^j\right) \cdot g_i^k \leqslant C_k\left(d_{q_k} - \frac{B}{C_m - R}\right) \quad \forall\, k \tag{12}$$

$$\sum_{k=1}^{N} g_i^k \leqslant 1 \quad \forall\, i \tag{13}$$

$$g_i^k \in \{0, 1\} \quad \forall\, i, k. \tag{14}$$

Constraint (11) indicates that the penalty term is the highest severity among the dropped flows. The idea is to deduct a penalty from the overall profit for every dropped flow. The second constraint (12) ensures that the total delay of each queue does not exceed the guaranteed queue delay. The third constraint in (13) specifies that every flow i cannot be assigned to more than one queue. However, it is possible that flow i is not assigned to any of the queues; meaning it is dropped. The last constraint in (14) indicates that this optimization problem is an integer linear programming problem, where assignment of flows to queues can be either zero or one (i.e., no fractional assignment of flows to queues).

5.3 Proof of NP-Hardness

The problem discussed in this paper is a $PMKP$ and is $NP - Hard$ in the weak sense. To prove it, we only need to show that this problem is actually a Multiple Knapsack Problem (MKP). In fact, MKP is a special case of the $PMKP$ when the penalty term ψ is known.

6 Heuristic Solution

To devise a real-time heuristic, we divide the overall problem into N sub-problems, where each problem $k \in N$ finds the optimal flows that fit in Queue k capacity such that the total profit (i.e., admitted severity) of the system is maximized. To ensure the maximum system profit and the best quality of service for the received flows, the sub-problems are solved starting from Queue N down to Queue 1. The solution of the sub-problems is inspired by the work of Ceselli and Righini [32].

Before presenting the algorithm, we define $\mathcal{F}_k$ as the set of input flows to Queue k, i.e., flows available to be admitted in Queue k. This set $\mathcal{F}_k$ of input flows to Queue k includes all the unassigned flows after solving the flow assignment problems to the higher priority queues (i.e., $\mathcal{F}_k$ includes all flows from the original set $\mathcal{F}$ except those selected by the higher priority Queues $\{k+1, \ldots, N\}$). For example, the input set of flows $\mathcal{F}_{N-1}$ excludes flows selected by Queue N from input set $\mathcal{F}_N$. Furthermore, we define $\mathcal{X}_k$, (with $|\mathcal{X}_k| = |\mathcal{F}_k|$) as the vector of flow selection flags from $\mathcal{F}_k$, where a value of 1 in the ith flag $\mathcal{X}_k[i]$ indicates that the ith flow is selected for admission in Queue k, while 0 means that the ith flow is not selected by Queue k. Consequently, the set of input flows to Queue k can be mathematically expressed as

$$\mathcal{F}_k = \mathcal{F}_{k+1} \setminus \bigcup_{\forall\, i | X_{k+1}[i]=1} f_i \qquad \forall\, k = N-1, \ldots, 1,$$

where $\mathcal{F}_N$ is the set of input flows to Queue N, which is initialized to all $\mathcal{F}$.

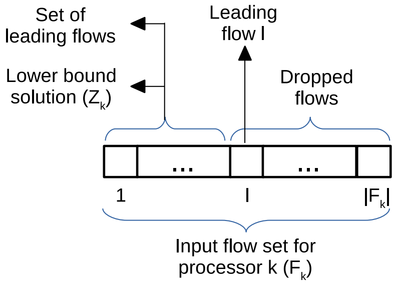

Set of leading flows
Leading flow l
Lower bound solution (Z_k)
Dropped flows
...
...
1
l
$|F_k|$
Input flow set for processor k (F_k)

Fig. 5. Input flow set divisions.

The proposed heuristic algorithm proceeds as follows: For $k = N$ down to 1, do the following steps:

- Order the set of input flows $\mathcal{F}_k$ for Queue k based on their severity.
- Find the initial flow selection vector $\mathcal{X}_k$ for admission to Queue k using the lower bound solution (See Section 6.1). Let Z_k be the sum of the severity of admitted flows in Queue k given the lower bound solution.
- Iteratively find better flow admission solutions for Queue k using the upper bound solution (See Section 6.2). If any flow admission solution $\widetilde{\mathcal{X}}_k$ with the sum-severity $\widetilde{Z}_k > Z_k$, let $Z_k = \widetilde{Z}_k$ and $\mathcal{X}_k = \widetilde{\mathcal{X}}_k$.
- $\mathcal{X}_k$ contains the optimal flow admission solution for Queue k having the maximum severity level Z_k, while respecting the flow constraints.

In the next sections, we will explain how the lower and upper bound solutions are obtained. The pseudo-code in Algorithm 1 details the steps of the entire process.

6.1 Lower Bound Solution

The input flows are admitted to Queue k in descending order of flow severity. Such flows are added until the flow with index l, for which its addition will exceeds the capacity C_k of Queue k (as illustrated in Fig. 5). Thus, the range of flows $\{1, \ldots, l-1\}$ can be admitted and processed by Queue k without violating the capacity constraint, and without penalty since no flow is dropped from this range of flows. This implies that the elements $\mathcal{X}_k[j]$ will have one values for $j \in \{1, \ldots, l-1\}$ and zero otherwise. We will refer to flow f_l^k as the initial leading flow, which represents the flow with the highest severity among the dropped flows in the range $\{l, \ldots, |\mathcal{F}_k|\}$. Consequently, the lower bound solution Z_k is initialized by adding the severity of those flows on the left side of flow l as shown in the following equation:

$$Z_k = \sum_{i=1}^{l-1} S_i.$$

This initial solution is a feasible solution. Next, we define $\mathcal{L}_k$ as the set of possible leading flows for Queue k. The set $\mathcal{L}_k$ is initialized with all the flows to the left side of the leading flow f_l^k (i.e., f_i^k, $\forall i \in \{1, \ldots l-1\}$) as illustrated in Fig. 5. Each leading flow is used to find an upper bound solution to the sub-problem.

6.2 Upper Bound Solution

In this part, we solve $|\mathcal{L}_k|$ Relaxed Penalized Knapsack Problems ($RPKP$) and compare their solutions with the lower bound solution to find the solution with the maximum profit. The following formulation represents the problem of finding the upper bound solution for Queue k's flow selection problem

$$\max \sum_{i=1}^{|\mathcal{F}_k|} S_i \cdot g_i^k \cdot k - \psi_k, \tag{15}$$

s.t.

$$S_i\left(1 - g_i^k\right) \leqslant \psi_k \quad \forall\ i \tag{16}$$

$$\begin{aligned}\sum_{i=1}^{|\mathcal{F}_k|}\left(\sum_{j=1}^{M} \sigma_{max}^j f_i^j + d_{q_k} \sum_{j=1}^{M} \rho_{max}^j f_i^j - \frac{B}{C_m - R} \sum_{j=1}^{M} \rho_{max}^j f_i^j\right)\\ \cdot g_i^k \leqslant C_k\left(d_{q_k} - \frac{B}{C_m - R}\right)\end{aligned} \tag{17}$$

$$0 \leqslant g_i^k \leqslant 1 \quad \forall i \tag{18}$$

$$\widetilde{\mathcal{X}}_k[i] = 1 \quad \forall\ i = 1 \ldots l-1 \tag{19}$$

$$\widetilde{\mathcal{X}}_k[l] = 0. \tag{20}$$

The upper bound solutions are computed using the following $|\mathcal{L}_k|$ iterative steps for every leading flow $i \in \mathcal{L}_k$ in descending order of severity (as detailed in the pseudocode of Algorithm 1):

While $l \geq 1$ do:

- Set the penalty term $\psi_k = S_l$; i.e., set it to the severity of the leading flow.
- Mark the leading flow l as dropped and all flows to its left as selected. This is basically equivalent to eleminating the current leading flow from $\mathcal{F}_k$ in the current iteration of the upper bound solution. Marking is represented by constraints (19) and (20).
- Solve the problem as an $RPKP$ with the penalty $\psi_k = S_l$.
- Fix the fractional results of $RPKP$ to 1 and 0 s using the proposed fixing heuristic (See Section 6.3). Compute the corresponding upper bound solution $\widetilde{\mathcal{X}}_k$ and its sum-severity $\widetilde{Z}_k$ as the severity of flows with $x_i = 1,\ \forall\ i$.
- If the current upper bound solution has a greater sum-severity than Z_k, update the solution $\mathcal{X}_k = \widetilde{\mathcal{X}}_k$ and its sum-severity $Z_k = \widetilde{Z}_k$.
- Select the new current leading flow l from the set $\mathcal{L}_k$ to be the one just to the left of the previous leading flow. In other words, set $l \leftarrow l - 1$.

The core philosophy of this solution is to eliminate one of the selected flows or the initial leading flow from the lower bound solution at a time, starting from the lowest severity among them and on. This opens room for selecting flows on the right of the current leading flow of each iteration, as long as this will maximize the profit of Queue k (i.e., the sum-severity of its final admitted flows). To do this, the problem is solved as an $RPKP$. Now since the penalty term ψ_k is known to be the severity of the leading flow, then the problem is reduced to an RKP. However, solving the problem using the relaxed version might result in a fractional solution. Thus, we use the proposed heuristic in Section 6 for fixing the relaxed solution to get a feasible integer solution.

Algorithm 1. Severity-Based Prioritized Processing Heuristic

```
Set $\mathcal{F}_k = \mathcal{F}$
for $k = N$ downto 1 do
  Sort $\mathcal{F}_k$ in descending order based on severity
  Find the minimum $l$ such that flows $1, \ldots, l-1$ fit in $C_k$
  Set $\mathcal{X}_k[i] = 1\ \forall\ i\{1, \ldots, l-1\}$
  Set $Z_k = \sum_{i=1}^{l-1} S_i$
  while $l \geqslant 0$ do
    if $l \geq 1$ then
      Set $\widetilde{\mathcal{X}}_k[l] = 0$
      if $l \geq 2$ then
        Set $\widetilde{\mathcal{X}}_k[i] = 1\ \ \forall i \in \{1, \ldots, l-1\}$
      end if
    end if
    Set $\widetilde{\mathcal{X}}_k[j]$ = $RPKP$ solution, $j \in \{l+1, \ldots, |\mathcal{X}_k|\}$.
    Set $a = l + 1$
    Set $b = |\mathcal{X}_k|$
    while a $<$ b do
      if $\widetilde{\mathcal{X}}_k[a] > 0$ AND $\widetilde{\mathcal{X}}_k[a] < 1$ then
        Round $\widetilde{\mathcal{X}}_k[a]$ to 1
        do
          if $\widetilde{\mathcal{X}}_k[b] > 0$ AND $\widetilde{\mathcal{X}}_k[b] < 1$ then
            Round $\widetilde{\mathcal{X}}_k[b]$ to 0
            Compute $D_t$
          end if
          $b = b - 1$
        while $D_t > D_k^{max}$ AND a $<$ b
        if $D_t > D_k^{max}$ then
          Undo any changes on $\widetilde{\mathcal{X}}_k$ for this iteration
          Round $\widetilde{\mathcal{X}}_k[a]$ to 0
        end if
      end if
      $a = a + 1$
    end while
    $\widetilde{Z}_k = \sum_{i|\widetilde{\mathcal{X}}_k[i]=1} S_i$
    if $\widetilde{Z}_k > Z_k$ then
      Set $Z_k = \widetilde{Z}_k$
      Set $\mathcal{X}_k = \widetilde{\mathcal{X}}_k$
    end if
    $l = l - 1$
  end while
  Set $\mathcal{F}_{k-1} = \mathcal{F}_k \setminus \bigcup_{\forall\ i|\mathcal{X}_k[i]=1} f_i$
end for
```

6.3 Fixing Upper Bound Solution

The solution produced by solving the relaxed problems (i.e., $RPKP$) in each of the $|\mathcal{L}_k|$ iterations of the upper bound solution results in values between 0 and 1. Such values do not constitute a feasible solution as they do not indicate which flows are assigned to Queue k. To resolve this problem, we propose an algorithm for rounding the fractional values to integer values.

The algorithm iteratively searches for a flow with strictly fractional value (i.e., between 0 and 1 neither including 0 nor 1) starting from the left side of the set $\widetilde{\mathcal{X}}_k$ and fixes it to 1 (i.e., select that flow). However, the $RPKP$ solution is optimal and rounding a value to 1 means accepting more portion of that flow. Consequently, this may result in more delays, which could violate the delay constraint. Thus, the next step is to search for a set of flows with strictly fractional values starting from the right side of the set $\widetilde{\mathcal{X}}_k$ and fix them to 0 (i.e., drop these flows) in order to regain the balance and satisfy the delay constraint. The pseudo-code in Algorithm 1 details these steps, where D_t is the total delay for selected flows ($\mathcal{X}_k[i] = 1, \forall i$) and D_k^{max} is the maximum delay for Queue k.

6.4 Upper Bound Solution Proof

First, we define the term $KP(l)$ to denote the optimal value of the binary knapsack with flows indices in the range $(1, \ldots, l-1)$ as selected for Queue k. The binary knapsack problem is defined in the following formulation:

$$KP(l) = \max \sum_{i=1}^{|\mathcal{F}_k|} S_i \cdot g_i^k \cdot k,$$

s.t.

$$\sum_{i=1}^{|\mathcal{F}_k|} \left(\sum_{j=1}^{M} \sigma_{max}^j f_i^j + d_{q_k} \sum_{j=1}^{M} \rho_{max}^j f_i^j - \frac{B}{C_m - R} \sum_{j=1}^{M} \rho_{max}^j f_i^j \right) \cdot g_i^k \leqslant C_k \left(d_{q_k} - \frac{B}{C_m - R} \right)$$

$$g_i^k \in \{0, 1\} \quad \forall\, i$$

$$\widetilde{\mathcal{X}}_k[i] = 1 \quad \forall\, i = 1 \ldots l-1.$$

Taking flow l as the leading flow, we define the term KP_l which is the optimal value of the binary KP with with flows indices in the range $(1, \ldots, l-1)$ as selected for Queue k and flow l as the flow with the highest severity among all dropped flows. The definition is expressed below:

$$KP_l = \max \sum_{i=1}^{|\mathcal{F}_k|} S_i \cdot g_i^k \cdot k,$$

s.t.

$$\sum_{i=1}^{|\mathcal{F}_k|} \left(\sum_{j=1}^{M} \sigma_{max}^j f_i^j + d_{q_k} \sum_{j=1}^{M} \rho_{max}^j f_i^j - \frac{B}{C_m - R} \sum_{j=1}^{M} \rho_{max}^j f_i^j \right) \cdot g_i^k \leqslant C_k \left(d_{q_k} - \frac{B}{C_m - R} \right)$$

$$g_i^k \in \{0, 1\} \quad \forall\, i$$

$$\widetilde{\mathcal{X}}_k[i] = 1 \quad \forall\, i = 1 \ldots l-1$$

$$\widetilde{\mathcal{X}}_k[l] = 0.$$

Note that KP_l is a special case of $KP(l)$ when the lth flow is eliminated from the optimization problem. Next, we define the term PKP_l as the optimal value of the binary PKP with penelty of subtracting the severity of the lth flow as the leading flow as shown below:

$$PKP_l = KP_l - S_l. \tag{21}$$

Finally, we use RKP_l to denote the relaxed knapsack probelm after eliminiating the lth flow (i.e., the leading flow), which is defined as follows:

$$RKP_l = \max \sum_{i=1}^{|\mathcal{F}_k|} S_i \cdot g_i^k \cdot k,$$

s.t.

$$\sum_{i=1}^{|\mathcal{F}_k|} \left(\sum_{j=1}^{M} \sigma_{max}^j f_i^j + d_{q_k} \sum_{j=1}^{M} \rho_{max}^j f_i^j - \frac{B}{C_m - R} \sum_{j=1}^{M} \rho_{max}^j f_i^j \right) \cdot g_i^k \leqslant C_k \left(d_{q_k} - \frac{B}{C_m - R} \right)$$

$$0 \leqslant g_i^k \leqslant 1 \quad \forall\, i$$

$$\widetilde{\mathcal{X}}_k[i] = 1 \quad \forall\, i = 1 \ldots l-1$$

$$\widetilde{\mathcal{X}}_k[l] = 0.$$

Ordering the set of input flows $\mathcal{F}_k$ based on severity in a non-increasing order results in the following inequalities:

$$S_l \geqslant S_{l+1} \tag{22}$$

$$KP(l) \geqslant KP(l+1) \tag{23}$$

$$KP(l) \geqslant KP_l \tag{24}$$

$$RKP_l \geqslant KP_l. \tag{25}$$

The following represents the computation of the upper bound solutions for all possible leading flows

$$u_i = RKP_i - S_i \quad \forall\, i = 1 \ldots l. \tag{26}$$

Where the flows in the range $\{1, \ldots, l\}$ represent the list of leading flows as explained previously.

Proposition 1: *Let $\mathcal{S}$ be the set of $RKP_i - S_i$ for leading flow i $\forall i = 1 \ldots l$. Also, let $\mathcal{P}$ be the set of solutions of PKP_i for leading flow i $\forall i = 1 \ldots l$. Then, u is an upper bound for the optimal value of the original PKP if*

TABLE 1
D2D Experimental Results

	Lecture Video		Rogue One Trailer	
	MPEG-2	MPEG-4	MPEG-2	MPEG-4
Average	38.9	23.1	78	84.1
Peak	92	29	97	97.6
Burst	2.3	1.2	1.2	1.1

Average/Peak values are in packets per second.

$$u \geqslant \max(\mathcal{S}) \geqslant \max(\mathcal{P}) \geqslant PKP. \tag{27}$$

Proof: From definitions (21), (25), and (26), and since

$u \geqslant u_i = RKP_i - S_i \geqslant KP_i - S_i = PKP_i \geqslant PKP \quad \forall i = 1 \ldots l$

then

$u \geqslant \max(\{u_1 \ldots u_l\}) \geqslant \max(\{PKP_1 \ldots PKP_l\}) \geqslant PKP \quad \square$

The core of the proposed heuristic (cf. Algorithm 1) is based on this proposition.

7 Experimental Results

In this section, we apply our proposed severity-based service prioritization scheme to VANETs. We first describe the VANET traffic data used in our experiments, experiment settings and environment. Furthermore, we will also discuss achieved results.

7.1 VANET Traffic Characterization

As explained earlier, our proposed heuristic works on any metric defined to serve as the *severity*. For our experiments, we define the severity to represent the type of application generating the VANET data. Data of safety applications (e.g., BSMs) are marked to have higher severity than data of infotainment applications (e.g., video streaming, congestion warning). Furthermore, we use the Bologna Ring-way dataset [33] to incorporate real vehicle mobility in our VANET simulations.

For all safety applications, the data rate is set to 10 packets per second as recommended by the standard for safety periodic messages (i.e., BSMs) [34]. Moreover, the burstiness and rate values for infotainment applications were acquired experimentally. In these experiments, a computer is setup as a VLC [35] server that streams videos over a Motorola Vehicle Mounted Modem or OBU acting as a device-to-device (D2D) communications medium. Another computer with VLC software acts as a client. While the VLC client is receiving the video stream, the network traffic is analyzed using WireShark [36] to determine the rate and burstiness of the traffic. It should be noted that two different types of videos, each with two different video encodings, were used for our experiments. The maximum rate of infotainment applications was set to the maximum rate of the WAVE protocol suite, while the burstiness value from our previous experiment was used as the average burstiness of infotainment applications. The results of the experiments are shown in Table 1.

To measure the performance of the proposed heuristic, we conducted two experiments. In the first experiment, we compared the results from the PMKP formulation in (10) against our proposed heuristic. In the second experiment, we compared our results against a baseline non-prioritized processor that operates using first come first serve (FCFS).

TABLE 2
Policing Queue Settings

Severity Level	Maximum Burst	Maximum Rate
1	2	50
2	2	50
3	1	10
4	1	10

TABLE 3
Queue Settings

# of Queue	Capacity	Guaranteed Delay
1	130	0.4
2	110	0.3
3	90	0.2
4	70	0.1

To conduct the experiments, we simulate the system model shown in Fig. 4 using a marking processor with a capacity (rate) of 9,000 packets per second. The capacity of the marking processor is set to a large value that induces minimum delay during the marking service. The policing processor configuration is shown in Table 2, which specifies for each severity level the maximum rate in packets per second and maximum burst size in packets.

The PMKP and our proposed heuristic use four queues with the configurations shown in Table 3, where the capacity is reported in packets per second and the guaranteed delay is reported in seconds. Severity level 1 is for safety messages while severity levels 2-4 are for infotainment messages. The baseline non-prioritized processor uses one queue without guaranteed delay (i.e., non-prioritized processing).

To evaluate the performance of the proposed heuristic against the baseline non-prioritized processor, we set the capacity of the baseline queue to equal the combined capacity of the four queues. Such configuration is more advantageous to the baseline no-prioritized processing algorithm as it will have one large queue with a rate that equals the rate of the four queues of the proposed prioritized processing algorithm. To validate the results, we run two other tests and set the baseline queue capacity to 50 percent of the combined capacity of the four queues in one experiment, and 150 percent of the combined capacity of the four queues in the other.

Obtaining the simulation data and result of the two experiments are discussed next.

7.2 Simulation Settings

In order to test the proposed heuristic, the Bologna Ring-way dataset was used to generate a basis of test data. The Bologna Ring-way dataset results were analyzed and a snapshot of the data was taken every 200 milliseconds. The snapshots provided us with an accurate number of flows for each vehicle during the simulation. It was revealed that a vehicle can have no more than 50 flows at a given time and most of the vehicles have 10 flows at some point during the simulation. Equipped with this data, we generated five data sets with 10, 20, 30, 40 and 50 flows. To simulate real-world traffic for each of these data sets, we further generated different percentages of safety-to-infotainment application data. Safety applications have the highest severity level, while infotainment and non-safety applications have three different severity levels. For each data set, five different combinations of safety-to-infotainment messages, which are 20-80 percent, 40-60 percent, 50-50 percent, 60-40 percent, and 80-20 percent were generated. The first two scenarios of 10 and 20 flows represent a light congestion case. A moderate congestion case is presented in the third scenario with 30 flows. To test high congestion traffic scenarios, the fourth and fifth scenarios with 40 and 50 flows are used. The number of flows in these scenarios represent the number of vehicles that a certain vehicle is receiving data from.

7.3 Comparison Study: Proposed Heuristic versus PMKP

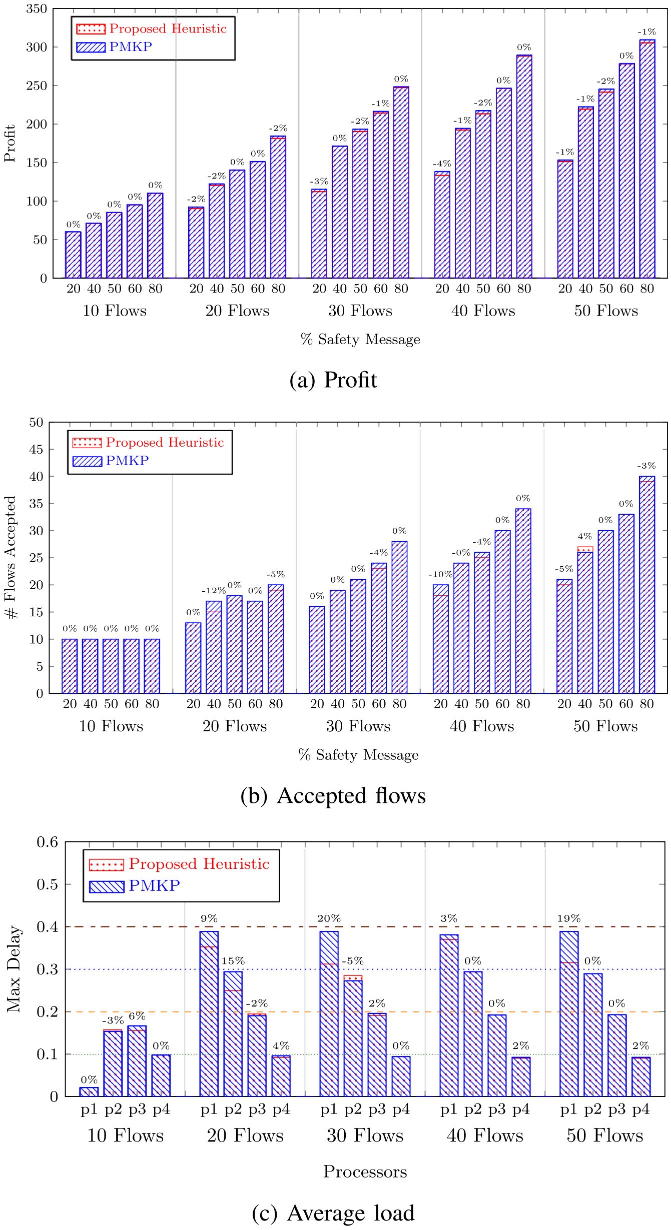
In this experiment, we executed the PMKP optimization and run the proposed heuristic on the 25 datasets. The testing results are shown in Fig. 6. As Fig. 6a indicates, the PMKP has a small gap of up to 4 percent in profit gain over the proposed heuristic. As for number of accepted flows, the PMKP and the proposed heuristic have similar results with the PMKP having more flows in some cases as illustrated in Fig. 6b. The PMKP achieves a better delay compared to the proposed heuristic, as clearly illustrated in Fig. 6c.

In this small-scale experiment that involves a high congestion scenario, the maximum number of flows is set to 50. This value can be higher in real-life traffic congestion scenarios. However, when raising the number of flows over 40, the optimal PMKP requires more processing time, unlike the proposed heuristic, which makes it useless in solving this real-time problem. This is due to the combinatorial nature of embedded integer linear programming in PMKP. Thus, this experiment demonstrates that the proposed heuristic performs almost at the same level as that of the PMKP in terms of profit and quality of service. This is the case when the number of sending vehicles is between 10 and 50. Our proposed heuristic outperforms the PMKP in terms of its execution time in all cases especially when the number of sending vehicles is over 40.

7.4 Comparison Study: Proposed Heuristic versus Non-Prioritized Processing

To compare the performance of the four queues when the proposed heuristic is employed with that of the single queue non-prioritized processing approach, we conduct three experiments. In these experiments, we set the baseline non-prioritized processor queue capacity to 50, 100, and 150 percent of the combined capacity of the four queues of the proposed heuristic, respectively. Experimenting with these various capacities aims to give more confidence to our claims on the superiority of our proposed prioritized processing heuristic.

One way to compare the proposed approach with the non-prioritized approach is to sum the severity of the accepted flows for each approach and compare the two results. However, in this paper we gave more advantage to the non-prioritized approach (i.e., baseline) over the proposed approach for the sake of fairness. Thus, in the proposed approach, the cumulative severity is the sum of severity of accepted flows. However, for the non-prioritized approach, the cumulative severity is the sum of the severity of all accepted flows multiplied by a factor based on the the total delay as shown below:



(a) Profit

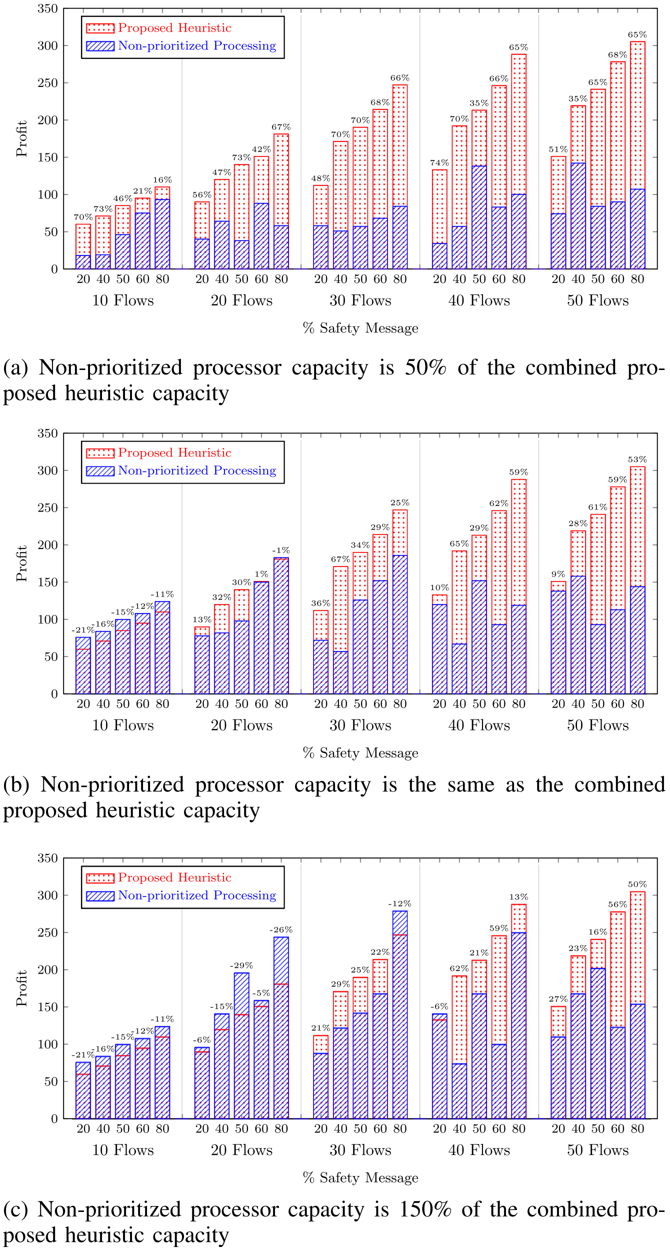
(b) Accepted flows

(c) Average load

Fig. 6. Performance of the proposed heuristic compared to the PMKP.

$$profit = \begin{cases} cumulative\ severity \cdot 1 & \text{if } d_t \geqslant 0.3, \\ cumulative\ severity \cdot 2 & \text{if } 0.2 \leqslant d_t < 0.3, \\ cumulative\ severity \cdot 3 & \text{if } 0.1 \leqslant d_t < 0.2, \\ cumulative\ severity \cdot 4 & \text{if } d_t < 0.1. \end{cases}$$

where d_t is the total delay. Basically, the profit of the baseline non-prioritized approach is multiplied by the equivalent queue number k of the proposed heuristic that offer an equivalent delay.

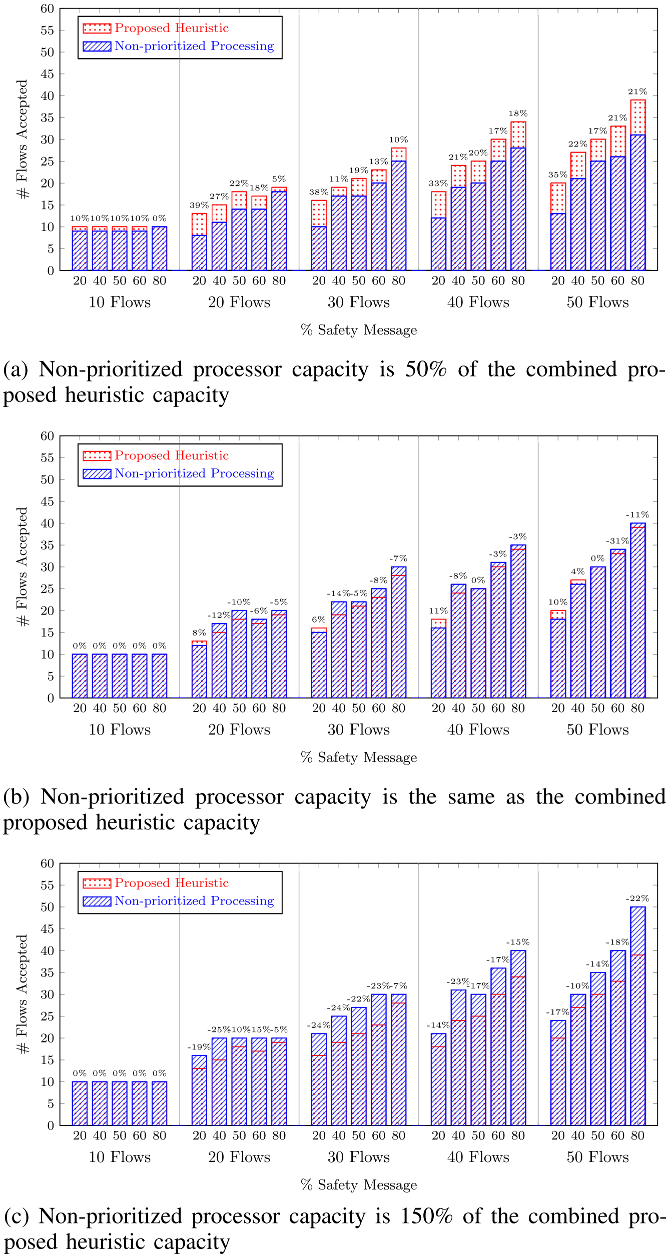


(a) Non-prioritized processor capacity is 50% of the combined proposed heuristic capacity

(b) Non-prioritized processor capacity is the same as the combined proposed heuristic capacity

(c) Non-prioritized processor capacity is 150% of the combined proposed heuristic capacity

Fig. 7. Profit of the proposed heuristic compared to the non-prioritized processing algorithm.



(a) Non-prioritized processor capacity is 50% of the combined proposed heuristic capacity

(b) Non-prioritized processor capacity is the same as the combined proposed heuristic capacity

(c) Non-prioritized processor capacity is 150% of the combined proposed heuristic capacity

Fig. 8. Number of accepted flows of the proposed heuristic compared to the non-prioritized processing algorithm.

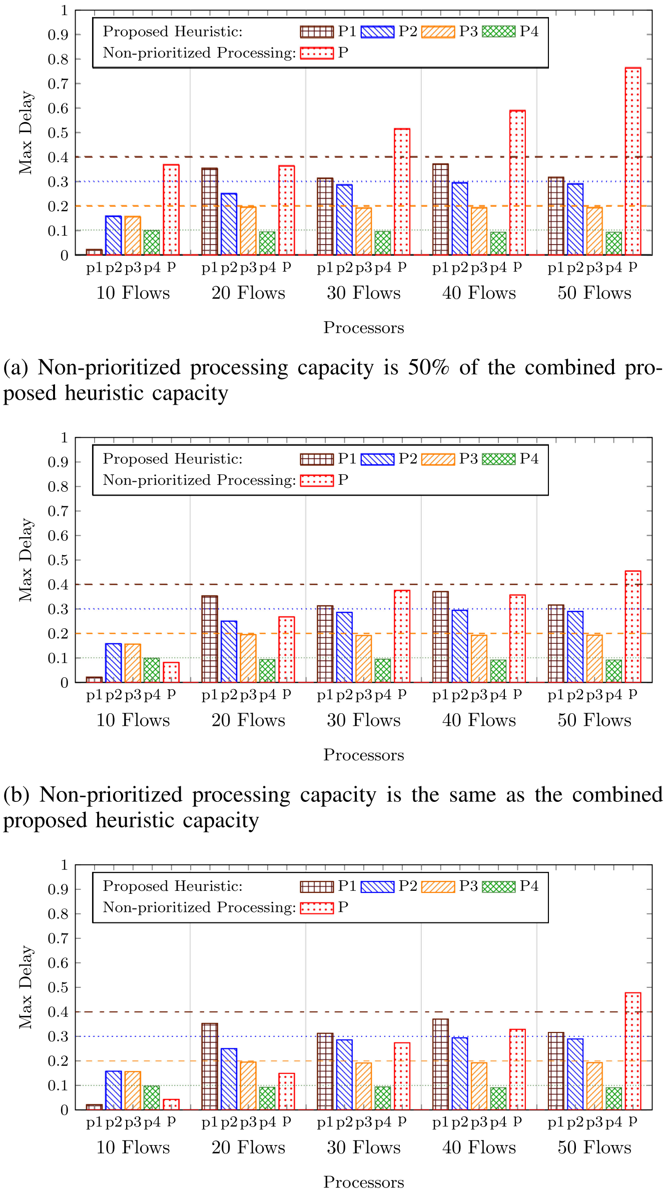
Profit-wise, the proposed heuristic outperforms the baseline non-prioritized processing approach in moderate and high congestion scenarios as shown in Fig. 7. When the capacity of the baseline non-prioritized processor queue is increased to 150 percent, the proposed heuristic still collects more profit than the baseline non-prioritized processing approach, as shown in Fig. 7c. Furthermore, Fig. 7a, illustrates the superior performance of the proposed heuristic when the baseline non-prioritized processing queue capacity is reduced by half.

The non-prioritized processor accepts almost the same or sometimes more flows compared to the proposed heuristic as indicated in Fig. 8. However, the extra flows processed by the baseline non-prioritized approach can be ignored, since the proposed heuristic still gets more profit.

To measure the quality of service, the worst case delay is observed, which is the maximum delay for every queue be it prioritized or non-prioritized. Fig. 9 shows that all four queues of the proposed heuristic guarantee a delay level for all flows based on their severity. This delay level never exceeds the required QoS limit. Contrary to our approach, the non-prioritized approach provides no delay guarantees, which results in high severity flows suffering long processing delays. In other words, all flows regardless of their severity levels encounter similar processing delays when the baseline non-prioritized approach is used. Conversely, our proposed prioritized processing heuristic fulfills its promise of processing flows based on their severity level and processing them within the QoS and time requirements.

8 Future Directions and Conclusion

In this paper, we presented a generic real-time heuristic that provides differentiated services based on a given set of flows



(a) Non-prioritized processing capacity is 50% of the combined proposed heuristic capacity

(b) Non-prioritized processing capacity is the same as the combined proposed heuristic capacity

(c) Non-prioritized capacity is 150% of the combined proposed heuristic capacity

Fig. 9. Maximum delay of the proposed heuristic compared to the non-prioritized processing algorithm.

and their corresponding *severity* metric. Furthermore, we demonstrated that the proposed heuristic can be used to offer differentiated services and improve QoS in VANETs. The proposed heuristic intelligently prioritizes the processing of flows in VANETs based on their corresponding severity metric. The problems is formulated as a PMKP, which is proved to be NP-Hard. Due to the complexity of the PMKP, a polynomial time algorithm based on a relaxed version of the PMKP formulation is proposed to perform the desired prioritization in real-time. The proposed heuristic is tested against the PMKP solution and a baseline non-prioritized processing approach. Results obtained through simulations with real traffic data demonstrated a minor difference in performance between the proposed heuristic and the PMKP. On the other hand, the proposed heuristic surpasses the baseline non-prioritized approach by 9 to 67 percent more profit in moderate and high congestion scenarios. As the results suggest, differentiated services are not required when the system has resources to satisfy all the requests, but rather when the system is under higher loads. In such scenarios, results show that our proposed prioritized processing heuristic is superior and provides better performance.

In our future research, we plan to pursue applications of our proposed approach beyond VANETs. Specifically, we plan to explore the potential use of our approach in support of Industrial IoT applications (IIoT) with real-time QoS constraints below 10 ms.

Acknowledgments

This research was made possible by NPRP grant # [71113-1-199] from the Qatar National Research Fund (a member of Qatar Foundation). The statements made herein are solely the responsibility of the authors. The publication of this article was funded by the Qatar National Library.

References

[1] Y. L. Morgan, “Notes on DSRC & WAVE standards suite: Its architecture, design, and characteristics,” *IEEE Commun. Surveys Tuts.*, vol. 12, no. 4, pp. 504–518, Oct.–Dec. 2010.

[2] IEEE Standard for Wireless Access in Vehicular Environments (Wave) – Multi-Channel Operation - Redline, *IEEE Standard 1609.4–2016 (Revision of IEEE Std 1609.4-2010) - Redline*, pp. 1–206, Mar. 2016.

[3] Y. Liu, “802.11 enhanced distributed channel access,” U.S. Patent 9 155 027, Oct. 6, 2015.

[4] N. Gupta, A. Prakash, and R. Tripathi, “Medium access control protocols for safety applications in vehicular ad-hoc network: A classification and comprehensive survey,” *Veh. Commun.*, vol. 2, no. 4, pp. 223–237, 2015. [Online]. Available: //www.sciencedirect.com/science/article/pii/S2214209615000546

[5] Y. J. Li, *An Overview of the DSRC/WAVE Technology*. Berlin, Germany: Springer, 2012, pp. 544–558. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-29222-4_38

[6] IEEE Guide for Wireless Access in Vehicular Environments (Wave) - Architecture, *IEEE Standard 1609.0-2013*, pp. 1–78, Mar. 2014.

[7] M. B. Brahim, E. B. Hamida, F. Filali, and N. Hamdi, “Performance impact of security on cooperative awareness in dense urban vehicular networks,” in *Proc. IEEE 11th Int. Conf. Wireless Mobile Comput. Netw. Commun.*, Oct. 2015, pp. 268–274.

[8] E. B. Hamida and M. A. Javed, “Channel-aware ECDSA signature verification of basic safety messages with k-means clustering in VANETs,” in *Proc. IEEE 30th Int. Conf. Adv. Inf. Netw. Appl.*, Mar. 2016, pp. 603–610.

[9] S. Al-Sultan ,M. M. Al-Doori ,A. H. Al-Bayatti, and H. Zedan, “A comprehensive survey on vehicular ad hoc network,” *J. Netw. Comput. Appl.*, vol. 37, pp. 380–392, 2014. [Online]. Available: //www.sciencedirect.com/science/article/pii/S108480451300074X

[10] M. A. Javed, E. Ben Hamida, and W. Znaidi, “Security in intelligent transport systems for smart cities: From theory to practice,” *Sensors*, vol. 16, no. 6, 2016, Art. no. E879.

[11] M. A. Javed and E. B. Hamida, “Adaptive security mechanisms for safety applications in internet of vehicles,” in *Proc. IEEE 12th Int. Conf. Wireless Mobile Comput. Netw. Commun.*, Oct. 2016, pp. 1–6.

[12] M. Timmers, S. Pollin, A. Dejonghe, L. V. der Perre, and F. Catthoor, “A distributed multichannel MAC protocol for multihop cognitive radio networks,” *IEEE Trans. Veh. Technol.*, vol. 59, no. 1, pp. 446–459, Jan. 2010.

[13] H. El Ajaltouni, A. Boukerche, and A. Mammeri, “A multichannel QoS MAC with dynamic transmit opportunity for VANETs,” *Mobile Netw. Appl.*, vol. 18, no. 6, pp. 814–830, 2013. [Online]. Available: http://dx.doi.org/10.1007/s11036-013-0475-6

[14] J. M.-Y. Lim, Y. C. Chang, M. Y. Alias, and J. Loo, “Cognitive VANET with enhanced priority scheme,” in *Proc. Int. Conf. Telecommun. Multimedia*, Jul. 2014, pp. 116–121.

[15] D. Lee, S. H. Ahmed, D. Kim, J. Copeland, and Y. Chang, “Distributed SCH selection for concurrent transmissions in IEEE 1609.4 multichannel VANETs,” in *Proc. IEEE Int. Conf. Commun.*, 2017, pp. 1–6.

[16] D. Lee, S. H. Ahmed, D. Kim, J. Copeland, and Y. Chang, "An efficient SCH utilization scheme for IEEE 1609.4 multi-channel environments in VANETs," in *Proc. IEEE Int. Conf. Commun.*, 2016, pp. 1–6.

[17] D. B. Rawat, D. C. Popescu, G. Yan, and S. Olariu, "Enhancing VANET performance by joint adaptation of transmission power and contention window size," *IEEE Trans. Parallel Distrib. Syst.*, vol. 22, no. 9, pp. 1528–1535, Sep. 2011.

[18] R. Baldessari, D. Scanferla, L. Le, W. Zhang, and A. Festag, "Joining forces for VANETs: A combined transmit power and rate control algorithm," in *Proc. 6th Int. Workshop Intell. Transp.*, 2010, pp. 1–5.

[19] M. Amadeo, C. Campolo, and A. Molinaro, "Enhancing IEEE 802.11p/wave to provide infotainment applications in VANETs," *Ad Hoc Netw.*, vol. 10, no. 2, pp. 253–269, 2012. [Online]. Available: //www.sciencedirect.com/science/article/pii/S1570870510001538

[20] L. Urquiza-Aguiar, A. VÃ*a*zquez-Rodas, C. Tripp-Barba, M. A. Igartua, L. J. de la Cruz Llopis, and E. S. Gargallo, "Max-min based buffer allocation for VANETs," in *Proc. IEEE 6th Int. Symp. Wireless Veh. Commun.*, Sep. 2014, pp. 1–5.

[21] C. Y. Chang, H. C. Yen, and D. J. Deng, "V2V QoS guaranteed channel access in IEEE 802.11p VANETs," *IEEE Trans. Depend. Secure Comput.*, vol. 13, no. 1, pp. 5–17, Jan. 2016.

[22] Q. Wang, S. Leng, Y. Zhang, and H. Fu, "A QoS supported multichannel MAC for vehicular ad hoc networks," in *Proc. IEEE 73rd Veh. Technol. Conf.*, 2011, pp. 1–5.

[23] H. T. Cheng, H. Shan, and W. Zhuang, "Infotainment and road safety service support in vehicular networking: From a communication perspective," *Mech. Syst. Signal Process.*, vol. 25, no. 6, pp. 2020–2038, 2011. [Online]. Available: //www.sciencedirect.com/science/article/pii/S0888327010004127

[24] D. T. Tuan, S. Sakata, and N. Komuro, "Priority and admission control for assuring quality of I2V emergency services in VANETs integrated with wireless LAN mesh networks," in *Proc. 4th Int. Conf. Commun. Electron.*, Aug. 2012, pp. 91–96.

[25] C. Chrysostomou, C. Djouvas, and L. Lambrinos, "Dynamically adjusting the min-max contention window for providing quality of service in vehicular networks," in *Proc. 11th Annu. Mediterranean Ad Hoc Netw. Workshop*, Jun. 2012, pp. 16–23.

[26] W. Alasmary and W. Zhuang, "Mobility impact in IEEE 802.11p infrastructureless vehicular networks," *Ad Hoc Netw.*, vol. 10, no. 2, pp. 222–230, 2012. [Online]. Available: //www.sciencedirect.com/science/article/pii/S1570870510000703

[27] 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.

[28] M. A. Salahuddin, A. Al-Fuqaha, F. Jacquelin, and Y. Shim, "Context severity based opportunistic service reprioritization for IEEE 802.11 p VANETs," in *Proc. 9th Int. Wireless Commun. Mobile Comput. Conf.*, 2013, pp. 1623–1628.

[29] S. Banani and S. Gordon, "Selecting basic safety messages to verify in VANETs using zone priority," in *Proc. 20th Asia-Pacific Conf. Commun.*, Oct. 2014, pp. 423–428.

[30] S. Biswas and J. Mii, "Relevance-based verification of VANET safety messages," in *Proc. IEEE Int. Conf. Commun.*, Jun. 2012, pp. 5124–5128.

[31] J. Y. L. Boudec and P. Thiran, "A short tutorial on network calculus. I. Fundamental bounds in communication networks," in *Proc. IEEE Int. Symp. Circuits Syst.*, 2000, pp. 93–96.

[32] A. Ceselli and G. Righini, "An optimization algorithm for a penalized knapsack problem," *Operations Res. Lett.*, vol. 34, no. 4, pp. 394–404, Jul. 2006. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0167637705000751

[33] 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.

[34] H. Rakouth, P. Alexander, A. J. Brown, W. Kosiak, M. Fukushima, L. Ghosh, C. Hedges, H. Kong, S. Kopetzki, R. Siripurapu, and J. Shen, "V2X communication technology: Field experience and comparative analysis," in *Proc. FISITA World Automotive Congr.*, 2013, pp. 113–129. [Online]. Available: http://dx.doi.org/10.1007/978–3-642-33838-0_10

[35] VLC media player. [Online]. Available: http://www.videolan.org/

[36] Wireshark - network protocol analyzer. [Online]. Available: https://www.wireshark.org/

Ala Al-Fuqaha (S'00-M'04-SM'09) received the PhD degree in computer engineering and networking from the University of Missouri-Kansas City, Kansas City, Missouri, in 2004. His research interests include the use of machine learning in general and deep learning in particular in support of the data-driven and self-driven management of large-scale deployments of IoT and smart city infrastructure and services, wireless vehicular networks (VANETs), cooperation and spectrum access etiquette in cognitive radio networks, and management and planning of software defined networks (SDN). He serves on the editorial boards and technical program committees of multiple international journals and conferences. He is a senior member of the IEEE and an ABET Program Evaluator (PEV).

Ihab Mohammed received the BS and MS degrees in computer science from Al-Nahrain University, Baghdad, Iraq, in 2002 and 2005, respectively. He is working toward the PhD degree in the CS Department, Western Michigan University. He is currently a research assistant with the NEST Research Lab. In 2005, he joined the Department of Computer Science, Al-Nahrain University, as an assistant lecturer and was promoted to a lecturer in 2011. He is a student member of the IEEE.

Sayed Jahed Hussini received the BS degree from the Chemical Engineering Department, Balkh University, and the MS degree in computer science from the Technical University of Berlin, in 2010. He is working toward the PhD degree in the CS Department, Western Michigan University, since 2014. He is a student member of the IEEE.

Sameh Sorour received the bachelor's and master's degrees in electrical engineering from Alexandria University, in 2002 and 2006, respectively, and the doctorate degree in electrical and computer engineering from the University of Toronto, in 2011. He is an assistant professor in the area of computer, network, and communication engineering. After his doctorate, he worked as an industrial researcher at Siradel Canada based on a MITACS industrial fellowship. He then worked for a year as a research fellow with the King Abdullah University of Science and Technology (KAUST). In 2013, he joined the King Fahd University of Petroleum and Minerals (KFUPM) as an assistant professor before moving to the University of Idaho, in 2016. He is a senior member of the Institute of Electrical and Electronics Engineers (IEEE), with more than 50 publications in IEEE refereed journals and conferences. His research interests lie in the board disciplines of computer networking and communication engineering, with strong emphasis on the areas of network coding, device-to-device communications, internet of things (IoT) and its applications, cloud storage networks, femtocaching and fog networking, management of dense cellular networks, and indoor positioning. He is also interested in using diverse mathematical tools in modeling, optimizing, and smartening the operation of various engineering systems, such as transportation networks, power grids, charging of plug-in electric vehicles, and e-health services.