Stochastic Analysis of Delay Statistics for Intermittently Connected Vehicular Networks

Leonardo Badia and Maria Scalabrin
Dept. of Information Engineering, University of Padova, via Gradenigo 6B, 35131 Padova, Italy
email: {badia, scalabrin}@dei.unipd.it

(Invited Paper)

***Abstract*—We analyze information delivery over a network where direct data exchange via wireless is not possible and therefore intermittent connection through vehicular relaying is used. We refer to a multi-hop scenario where a stationary unit exchanges data packets with a destination outside its communication range, using passing-by vehicles as information carriers. In such a context, we propose an exact mathematical model based on Markov chains to quantify relevant performance metrics. In particular, while throughput-related parameters can be directly obtained via queueing analysis, delay statistics are more complex to derive, due to the in-order delivery requirement for the transmitted information content. Hence, data packets transported by fast vehicles still need to wait for the reception of all previous packets, possibly carried by slower vehicles. Nevertheless, by properly exploiting structural properties of the resulting Markov chains, the delay statistics can be fully characterized.**

***Index Terms*—Markov processes, vehicular networks, queueing analysis, delay analysis.**

I. Introduction

Vehicular ad-hoc networks (VANETs) are built from mobile nodes consisting of vehicles such as cars or buses, which are equipped with on board transmission units. The relatively high speeds of the mobile nodes imply a rapidly changing network topology, and for this reason, network connectivity is generally intermittent [1]. However, mobility patterns can be predicted and even exploited for networking purposes; in a sense, coverage is even extended by mobility, although without continuous connectivity [2].

Application-wise, VANETs are suitable for disaster recovery by realizing a self-organizing backup infrastructure in the aftermath of a catastrophe. Outside emergency contexts, they can be employed for public safety, e.g., to monitor vehicular traffic in real-time, notifying car accidents and/or road congestions [3]. They may also offer ambient surveillance for public safety, pollution, or other environmental parameters, and they can offer, to some extent, Internet connectivity to the on-board units of private vehicles [4]. For all these applications, timely delivery of data is an issue, and a characterization of the delays encountered by data transmission is key.

Wireless communications in a VANET may involve only vehicles or also stationary roadside units (SRU) [5]. The role of SRU is to empower the inherently unreliable communications among vehicles, for example, by storing relevant data that otherwise would be difficult to retrieve, since the vehicles are mobile. However, SRUs are assumed *not to be* in coverage range with each other, and therefore they need to resort to multihop routing through vehicles [6].

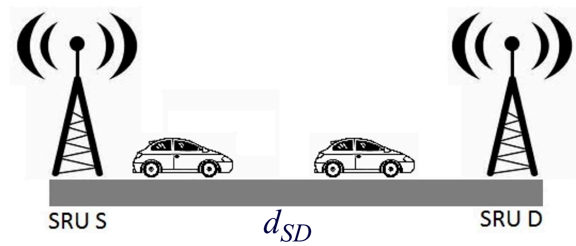

Fig. 1. The scenario: an intermittently connected VANET linking two SRUs.

The scenario investigated in this paper is presented in Fig. 1, similar to what proposed in [7] and [8], where it is referred to as an archetypal vehicular intermittently connected network. It consists of two SRUs, which are outside the communication range of each other. However, they can exploit vehicles, such as cars, traversing the connecting road, as data mules. More precisely, the source SRU (S) can transmit data to any vehicle moving towards the destination SRU (D) when it is within coverage range. In turn, the vehicle will move and traverse the distance d_{SD} between the borders of the radio coverages. Upon entering SRU D's coverage range, the vehicle will transmit the data to the destination. Data are therefore stored in a queue at the transmitter's side, and transmitted to passing-by vehicles according to a first-come first-serve policy, i.e., whenever a vehicle is present to piggyback them.

We will refer to any amount of data carried by one vehicle as a "data packet." For the purpose of the analysis, it does not really matter whether packet sizes are identical. Actually, it does not even matter how large a packet is; still, the data packets need to be received in the right order at the receiver's side. Any packet received out-of-order will be stored in a resequencing buffer, from which it will be released only when all the packets transmitted before it are received as well. Hence, if the system is seen as a queue of data packets, the service process is random due to the fact that a packet is served only when a vehicle passes by. Moreover, the service time is variable because of different vehicle speeds, and also due to the in-order delivery requirement.

For the sake of simplicity, we consider that all packet transmissions are fully reliable [7]. This assumption is reasonable if the time, during which a vehicle traverses the coverage range of each SRU, is large enough. The data channel between the two SRU can be evaluated in terms of stability and throughput, which in turn relate to the arrival rates of packets and vehicles. Conversely, it is much more challenging to quantify the delay encountered by data packets in such a system.

We can identify different delay terms that a packet may experience [9], [10]. The *queueing delay* is the time spent by a data packet in the queue of SRU S before being transmitted. The *transmission delay* in this case would be the variable time that a vehicle takes to traverse the distance d_{SD}. Before being released at the destination, a data packet may have to wait for a further *resequencing delay* in the destination's buffer. The *delivery delay* is then defined as the sum of these two last terms, i.e., the transmission delay and the resequencing delay.

Our goal in this paper is to present a rigorous analysis of all these terms. Especially, the delivery delay is interesting as it shows the non-trivial effect of packet transmission and re-ordering at the receiver's buffer. While the queueing delay is directly understandable as depending on the congestion in the sender's queue, the delivery delay is determined by the speed differences between vehicles. In this sense, our analysis justifies contributions such as [8] whose focus is on selecting the vehicles to carry the data packets according to their speed, rather than exploiting them all on a first-come first-serve basis.

The rest of this paper is organized as follows. Section II reviews related works from the literature. In Section III we describe the mathematical model and we derive the delay statistics. Section IV presents a numerical evaluation of the proposed analysis. Finally, we conclude in Section V.

II. Related Works

The scenario studied in the present paper is directly related to that analyzed by Khabbaz et al. in [8], where the transmission scheme also follows Fig. 1. However, the focus of that paper is on selecting the best vehicles to piggyback the packet to destination, and there is no derivation of the full delay statistics, but instead the goal is to achieve low delays on average. Thus, our analysis is kind of complementary to that study and can be seen as an enabler for a more detailed optimization framework. A similar related investigation is presented in [11]; the focus is also different from our contribution here, since it involves determining an optimal schedule of transmission, instead of the delay statistics. In both [8] and [11] there is no investigation either on the arrival rates of packets in the transmitter's queue, which is assumed to be always full.

According to the discussion of these papers, also following the system description made in [7], an analysis via Markov chains present similarities with the studies of retransmission-based techniques, such as automatic repeat request (ARQ), presented in [10], [12], [13]. Investigations of ARQ involving non-instantaneous feedback, such as [12], take into account that the round-trip time of a data packet and its acknowledgment is significantly larger than the packet transmission time. Thus, similar instruments can be employed to analyze our scenario, where no error correction is required (as the packets are always delivered) but the propagation delays are significant as they relate to the speeds of the vehicles.

In particular, a closely related model is explored in [13], which exploits the same definition of queueing and delivery delay. However, the focus there is not on vehicular networks, but on an ARQ system with variable round-trip time. Since in our scenario vehicles have different speed, that part of the model is also similar. Still, there are many relevant differences involving that variability of the transmission time in our scenario is just a consequence of different vehicle speeds. Moreover, in our analysis no retransmission is involved.

Another related paper is [5], which considers a low-density vehicular network, which makes it impossible to guarantee full connectivity among the vehicles and therefore impose to resort to SRUs. Depending on the traffic parameters of both vehicles and data, the authors investigate how to stochastically guarantee a packet delivery within a given time constraint. However, the main investigation of that paper does not involve the delay statistics, but just the worst-case delay; moreover, the analysis explores in detail on the placement of SRUs from a planning perspective, and in particular the inter-SRU distance as a parameter; in our analysis, such a value is fixed.

Because of intermittent connectivity of VANETs, Banerjee et al. in [6] discuss the improvement brought by auxiliary infrastructure such as base stations and stationary relay nodes. Therefore, also this paper is a source of inspiration for the present contribution, since we consider a vehicular connectivity between two SRUs. However, in that paper the focus is on the trade-off between infrastructure cost and the technical improvement brought. The main conclusion is that even a small amount of stationary nodes in a VANET can be extremely beneficial for the technical performance, especially for what concerns the delay. Thus, this result supports and motivates our analysis here for the scenario discussed above.

III. Analysis

For the following analysis we consider a discrete (slotted) time axis, where the time slot duration is τ seconds. For the purpose of system evolution, assume that relevant events, such as arrivals of data and/or transits of vehicles only occur at the end of a time slot. We also assume that vehicle arrivals are independent and following a Bernoulli process of intensity μ vehicles/second. Thus, denote with K_v the number of slots between the arrivals of two vehicles in the coverage range of SRU S; K_v is a geometrically distributed r.v. with distribution

$$f_{K_v}(k) \triangleq \mathrm{P}[K_v = k] = q(1-q)^k, \qquad k \geq 0 \qquad (1)$$

where $q = \mu\tau$ is the probability of vehicle arrival in a slot.

For the arrival of packets in the buffer of SRU S, we analogously assume an independent and identically distributed (i.i.d) Bernoulli process with intensity λ packets/second. Thus, denoting with K_b the number of slots between two packet arrivals at the transmitter's side, we have

$$f_{K_b}(k) \triangleq \mathrm{P}[K_b = k] = p(1-p)^k, \qquad k \geq 0 \qquad (2)$$

where $p = \lambda\tau$ is the probability of packet arrival in a slot.

If a packet arrives at the SRU S's buffer and finds it empty, and a vehicle is passing by, it is immediately available to be transmitted. Otherwise, because either there is no vehicle to carry it towards the destination, or there are other packets in queue, the newly arrived packet is enqueued at the transmitter buffer. For the sole purpose of evaluating the system throughput, one may consider a $Geo(p)$/$Geo(q)$/1 queue [7]. For instance, the stability condition of the queue simply corresponds to imposing $p < q$. However, to quantify the delay experienced by a data packet, we need to resort to a more detailed analysis. As we will show, we cannot limit the delay characterization to the queueing delay, but we also need the additional term represented by the delivery delay.

Assume that also the speeds of the vehicles are i.i.d. and distributed in the interval $[V_{\min}, V_{\max}]$, with average $\overline{V}$. For the numerical evaluations in Section IV, we will consider a truncated Gaussian distribution with standard deviation σ_V. For simplicity, we treat the speed of the vehicles as constant during the entire movement over distance d_{SD} between the borders of the coverage regions of SRUs. This assumption is not strictly needed, but just simplifies the description. Thus, the traversing time of a vehicle having speed $v \in [V_{\min}, V_{\max}]$ is given by d_{SD}/v. However, since the time axis is discrete, we need to take the traversing time as an integer number of slots, i.e., we consider it to be equal to $T = \lceil d_{SD}/v/\tau \rceil$. Such a value falls within the interval $[T_{\min}, T_{\max}]$, where $T_{\min} = \lceil d_{SD}/V_{\max}/\tau \rceil$ and $T_{\max} = \lceil d_{SD}/V_{\min}/\tau \rceil$. As a side remark, if the speeds of the vehicles are Gaussian distributed, the traversing time T is not Gaussian. This is a further difference from the numerical results shown in [13].

Now, let ρ_T be the probability that a vehicle traverses the distance d_{SD} in T time slots. Such a probability can be computed by partitioning $[V_{\min}, V_{\max}]$ in $T_{\max} - T_{\min} + 1$ intervals, each related to a different traversing time. The value of ρ_T is achieved by integrating the speed distribution of the vehicles over the corresponding interval. Thus, during each time slot, either no vehicle arrives with probability $(1-q)$, or a vehicle arrives, which will be reach the destination in T slots, and this happens with probability $q\rho_T$.

The system can be described by a discrete time Markov chain (DTMC), whose transitions occur every τ seconds [10]. A DTMC description fully characterizes the system, since it is immediate to prove that the system state at time t, denoted as $X(t)$, only depends on the state $X(t-1)$ in the previous time slot, regardless of the past history, i.e.

$$\begin{aligned}&\mathrm{P}[X(t)=x_t \,|\, X(t-1)=x_{t-1}, X(t-2)=x_{t-2}, \ldots, X(0)=x_0] \\ &= \mathrm{P}[X(t)=x_t \,|\, X(t-1)=x_{t-1}]\end{aligned}$$

The state $X(t)$ of the system must contain the sequence of the residual traversing time (in slots) of the vehicles that are moving from SRU S to SRU D at time t. This is a vector of integers $\mathbf{y}$, whose elements are within $[0, T_{\max}]$. Actually, to have a tractable model, one may think of keeping track only of a limited number of vehicles. Thus, we set a cap to $N_{\max}$ vehicles tracked. If such a value is properly chosen, the approximation induced is almost negligible. However, one can still have a fully exact model by setting $N_{\max} = T_{\max}$, since the highest number of vehicles is present in the system whenever $T_{\max}$ consecutive arrivals of vehicles happen for $T_{\max}$ consecutive time slots, and all of these vehicles have traversing times equal to $T_{\max}$.

At time t, if a vehicle arrives and its traversing time is T, we set the last element of $\mathbf{y}$ to T. At time $t+1$, the last element of $\mathbf{y}$ will contain the new traversing time of the next vehicle, and the previous vehicle will have traversing time equal to $T-1$, since one slot has elapsed. As an additional rule, we set the last element of $\mathbf{y}$ to 0 if no vehicle arrived in that time slot. Finally, the corresponding element of $\mathbf{y}$ is set to 0 also when a vehicle arrives, but no packets are available in the transmitter's queue to be piggybacked to SRU D.

To better understand the evolution of vector $\mathbf{y}$, assume that the queue is always full, and $N_{\max} = 4$, $T_{\max} = 5$. If no vehicle are present before time 0, and in the following four time slots the system sees four arrivals with traversing times T_0=5, T_1=2, T_2=3, T_3=4, then the evolution of $\mathbf{y}$ is

$$\mathbf{y}(0)=[0005]\,,\;\; \mathbf{y}(1)=[0042]\,,\;\; \mathbf{y}(2)=[0313]\,,\;\; \mathbf{y}(3)=[2024]$$

Thus, the number of possible values for the last entry of $\mathbf{y}$ is $T_{\max} - T_{\min}+2$, since all values are possible from $T_{\min}$ to $T_{\max}$, plus 0. A position $k < N_{\max}$ in $\mathbf{y}$ describes instead a vehicle that started moving towards the destination $N_{\max}-k$ slots before; thus, the highest value of the kth element of $\mathbf{y}$ is $T_{\max}+k-N_{\max}$. Just to simplify the computation, if $T_{\min} = 1$, the number of possible values of $\mathbf{y}$ is $(T_{\max} + 1)(T_{\max}) \ldots (T_{\max} - N_{\max} + 2)$. In the example above, $\mathbf{y}$ can thus take 360 possible values.

For higher values of $T_{\max}$ and $N_{\max}$ such a number rapidly increases. However, observe that, still referring to the example above, $\mathbf{y}$=[2 0 2 4] is equivalent to $\mathbf{y}$=[0 2 2 4], as they both represent a situation where 3 vehicles are traveling from SRU S to SRU D and are due to arrive in 2, 2, and 4 slots. We can therefore represent $\mathbf{y}$ with a notational convention that the entries are written in increasing order (in particular, zero entries are all moved to the left). If m is the number of non-zero entries in $\mathbf{y}$, we have the following situations:
– if $m = 0$, i.e., no vehicles are traversing d_{SD}, $\mathbf{y}$ can only be an all-zero vector.
– if $m = 1$, i.e., only one vehicle is traversing d_{SD}, $\mathbf{y}$ has only one non-zero entry that can take $T_{\max}$ different values.
– if $2 \le m \le N_{\max}$, there are m vehicles and $\mathbf{y}$ can take $T_{\max}(T_{\max}-1) \ldots (T_{\max}-m+1)$ different values.

Thus, the number of possible values of $\mathbf{y}$ is

$$\Upsilon = \sum_{m=0}^{N_{\max}} \prod_{i=1}^{m} (T_{\max} + 1 - i) + 1 \tag{3}$$

In the numerical example with $N_{\max}$=4, $T_{\max}$=5, we have $\Upsilon = 206$, with a considerable reduction from the 360 alternatives of the previous representation.

To complete the description, we consider, beyond $\mathbf{y}$, also the number q of data packets in queue at SRU S's buffer. If such a buffer can store up to $Q_{\max}$ packets, then at any time t, $q(t)$ is an integer in $[0, Q_{\max}]$. This value evolves to $q(t+1)$ in the next time slot through a transition matrix

$$\begin{pmatrix} 1-p & p & 0 & 0 & \ldots & 0 \\ q & 1-p-q & p & 0 & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \ldots & 0 & q & 1-p-q & p \\ 0 & \ldots & 0 & 0 & q & 1-q \end{pmatrix}.$$

Therefore, the buffer occupancy $q(t)$ evolves as a birth-and-death process, and the entire system $X(t) = (q(t), \mathbf{y}(t))$ can be framed as quasi-birth-and-death (QBD) [14].

Since any combination of q and $\mathbf{y}$ is possible, the total number of states is $\Sigma = (Q_{max} + 1)\Upsilon$. We can derive a corresponding transition matrix $\mathbf{T}$ of the full state $X(t)$. Each element t_{ij} of this $\Sigma \times \Sigma$-matrix will contain the transition probability from the ith state to the jth. Because of the QBD property, many transitions will have probability 0, and $\mathbf{T}$ will be sparse [10]. Also, $\sum_{j=1}^{\Sigma} t_{ij} = 1$. The non-zero entries in $\mathbf{T}$ will be all related to p, q, and the ρ_T values.

Consider a given $X(t) = (q, \mathbf{y})$, where $\mathbf{y}$ contains m traversing times $T_1, T_2, \ldots, T_m$ and $N_{\max} - m$ zero entries. For notational simplicity, we omit time index t where obvious. From this state, if $0<q<Q_{\max}$ and $m<N_{\max}$ with strict inequalities, $X(t)$ can only transition to $X(t+1)$ equal to

$$\begin{aligned}
&(q+1, \boldsymbol{y}^-) && \text{with probability } p(1-q) && \text{(4a)}\\
&(q, \boldsymbol{y}^-) && \text{with probability } (1-p)(1-q) && \text{(4b)}\\
&(q, \boldsymbol{y}^- \leftarrow T) && \text{with probability } pq\rho_T && \text{(4c)}\\
&(q-1, \boldsymbol{y}^- \leftarrow T) && \text{with probability } (1-p)q\rho_T && \text{(4d)}
\end{aligned}$$

where $\mathbf{y}^-$ denotes a vector obtained from $\mathbf{y}$ by decreasing all positive entries by one. In other words, $\mathbf{y}^- = [\mathbf{y} - \mathbf{1}]^+$ where $\mathbf{1}$ is an all-one vector and $[\cdot]^+$ denotes the maximum between the argument and zero, applied element-wise. Notation $\mathbf{a} \leftarrow T$ implies that T is added to vector $\mathbf{a}$ replacing a non-zero entry. Remember that in writing vector $\mathbf{y}$ we make no distinction on possible permutations of its entries and always write them in increasing order; thus, the transitions written above may require adjustments to keep this convention into account.

The transitions described by (4a)–(4d) also hold if $m{=}N_{\max}$ but at least one of the element of $\mathbf{y}$ is equal to 1. In this case, there is a zero entry in $\boldsymbol{y}^-$ that can be overwritten by T. Instead, if $\mathbf{y}$ contains $N_{\max}$ entries larger than 1, vehicles arrivals in that slot are neglected and the transitions simply become towards $X(t+1)$ equal to

$$\begin{aligned}
&(q+1, \boldsymbol{y}^-) && \text{with probability } p && \text{(5a)}\\
&(q, \boldsymbol{y}^-) && \text{with probability } 1{-}p && \text{(5b)}
\end{aligned}$$

These equations may introduce a small approximation in the computations, which can actually made almost negligible by properly choosing $N_{\max}$. In particular, as argued before, there is no approximation if $N_{\max}$ is at least equal to $T_{\max}$.

If $q = 0$, there is no packet to draw from the queue, unless a packet arrives in the same time slot, since we assumed that a packet is immediately available for being transmitted in the slot it arrives. The transitions from $X(t) = (0, \mathbf{y})$ are towards $X(t+1)$ equal to

$$\begin{aligned}
&(1, \boldsymbol{y}^-) && \text{with probability } p(1-q) && \text{(6a)}\\
&(0, \boldsymbol{y}^-) && \text{with probability } 1-p && \text{(6b)}\\
&(0, \boldsymbol{y}^- \leftarrow T) && \text{with probability } pq\rho_T && \text{(6c)}
\end{aligned}$$

Instead, if $q{=}Q_{max}$ then the queue is full and cannot accept any more packets. Thus, the event of a packet arrival without a vehicle arrival in the same time slot forces the system to drop one packet. The transitions are towards state $X(t+1)$ equal to

$$\begin{aligned}
&(Q_{\max}, \boldsymbol{y}^-) && \text{with probability } 1-q && \text{(7a)}\\
&(Q_{\max}, \boldsymbol{y}^- \leftarrow T) && \text{with probability } pq\rho_T && \text{(7b)}\\
&(Q_{\max}-1, \boldsymbol{y}^- \leftarrow T) && \text{with probability } (1-p)q\rho_T && \text{(7c)}
\end{aligned}$$

By combining (5a)–(5b) with the subsequent systems (6a)–(6c) and (7a)–(7c) to keep into account all cases, such as, for instance, a full queue at the transmitter's side and m vehicles still traversing d_{SD}, all the transitions can be derived and collected into matrix $\mathbf{T}$. If the number of states Σ is manageable, such a matrix can be handled by mathematical software such as Matlab [15] exploiting its sparsity properties.

The transition matrix $\mathbf{T}$ heavily depends on the arrival probability $p = \lambda\tau$ of packets in the system. For example, if λ is increased so that $p{=}1$, the system becomes unstable as there is always a packet arriving in the queue, which can be sent only if a vehicle is available to carry it towards destination; such a condition is also referred to in the literature as *Heavy Traffic* [9]. In such a case, matrix $\mathbf{T}$ can be written as $\mathbf{T_1}$; under Heavy Traffic condition the queueing delay is infinite, but still the delivery delay can be quantified. Similarly, we denote with $\mathbf{T_0}$ the matrix describing the case, where there are no arrivals in the system, i.e., $\lambda{=}0$.

Matrix $\mathbf{T}$ can also be used to derive the statistics of the metrics of interest through a matrix-geometric approach [14]. In particular, one can obtain the overall delay experienced by a packet, denoted as τ_G, split into the previously defined terms, namely, the queueing delay τ_Q and the delivery delay τ_D [10]. The first step to derive such delay metrics is the evaluation of the steady-state probabilities of the Markov system. These correspond to the probability of finding the system in a given state $X(t)$ when $t \gg 0$ and can be found as a normalized fixed-point $\boldsymbol{\pi}$ of the matrix $\mathbf{T}$. In other words, if the Markov chain is positive recurrent (which is immediate to verify to hold in our case), we can collect the probabilities that the system is in each state $X(t)$ at time $t \gg 0$, denoted as π_X, into a Σ-sized vector $\boldsymbol{\pi} = \{\pi_X\}$, for which

$$\mathbf{T}\boldsymbol{\pi} = \boldsymbol{\pi} \tag{8}$$

$$\sum_{\text{all } X} \boldsymbol{\pi} = 1 \tag{9}$$

One can see (8) as a system of equations; however, it is not full-rank and normalization condition (9) must be added, as $\boldsymbol{\pi}$ is a probability vector. Thus, $\boldsymbol{\pi}$ is promptly found as

$$\boldsymbol{\pi} = \begin{bmatrix} \mathbf{T} - \mathbf{I}_\Sigma \\ \mathbf{1} \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \tag{10}$$

where $\mathbf{I}_\Sigma$ is a $\Sigma \times \Sigma$ identity matrix.

In the case of Poisson arrivals, any arriving packet would encounter the system in the steady-state probability $\boldsymbol{\pi}$, due to the so-called PASTA (Poisson arrivals see time averages) property [12]. However, in our DTMC, packet arrivals are Bernoulli distributed with probability p. Thus, the distribution $\mathbf{b}^{(0)}$ of the system state probabilities upon a packet arrival is

$$\boldsymbol{b}^{(0)} = \mathbf{T_1}\boldsymbol{\pi} \ . \tag{11}$$

(Recall that $\mathbf{T_1}$ is the transition matrix under Heavy Traffic.) The derivation of (11) was observed for the first time by [10] and is due to the one-step memory of the Markov chain. Thus, in the previous time slot the system state is distributed as $\boldsymbol{\pi}$ and after one arrival it transits according to $\mathbf{T_1}$.

To evaluate the queueing delay τ_Q we remark that each time a new packet enters the transmitter's queue, future packet arrivals do not affect its delay [13]. So, we can "turn off" the arrival process and see how long it takes for the queue to become empty. A newly arrived packet can even be immediately transmitted in the same time slot it arrives, in which case $\tau_Q{=}0$. This happens when the queue is empty and a vehicle is available to carry it to SRU D; this means that

$$\mathrm{P}[\tau_Q = 0] = \mathrm{P}[\tau_Q \leq 0] = \sum_{j \in \mathcal{Q}_0} b_j^{(0)} \tag{12}$$

where $\mathcal{Q}_0$ is defined as the subset of the Σ possible states for which $q = 0$, so that the summation is made on all values of the state $X = j$ where q is zero and instead $\mathbf{y}$ is arbitrary.

Since we turned off the arrival process, i.e., we set λ to 0, the system evolves according to transition matrix $\mathbf{T_0}$. If no packet arrives in the system, the queue is bound to be empty after sufficiently many time slots. The vector of state probabilities after k slots is thus

$$\boldsymbol{b}^{(k)} = (\mathbf{T_0})^k \, \boldsymbol{b}^{(0)} \tag{13}$$

and the same reasoning of (12) can be applied to $\boldsymbol{b}^{(k)}$ to derive

$$\mathrm{P}[\tau_Q \leq k] = \sum_{j | q=0} b_j^{(k)} \tag{14a}$$

$$\mathrm{P}[\tau_Q = k] = \mathrm{P}[\tau_Q \leq k] - \mathrm{P}[\tau_Q \leq k{-}1] \tag{14b}$$

To justify the above equations, observe that the packet arrived when the state distribution is $\boldsymbol{b}^{(0)}$ can be seen as exiting the queue when the value of q becomes 0 for the first time. Thus, τ_Q can be seen as the first passage time through any of the states in $\mathcal{Q}_0$ [10]. Also, note that $\mathcal{Q}_0$ is an absorbing state.

A similar reasoning that involves the $\boldsymbol{b}^{(k)}$ vectors can be employed to evaluate the distribution of the overall delay τ_G. This time, instead of set $\mathcal{Q}_0$ we consider the all-zero state, also absorbing, denoted as $X = (0, \mathbf{0})$, where not only $q = 0$ but also all traveling vehicles have reached destination. Thus,

$$\mathrm{P}[\tau_G \leq k] = b_{(0,\mathbf{0})}^{(k)} \tag{15a}$$

$$\mathrm{P}[\tau_G = k] = \mathrm{P}[\tau_G \leq k] - \mathrm{P}[\tau_G \leq k{-}1] \,. \tag{15b}$$

Now, to derive the distribution of the delivery delay τ_D we simply observe that $\tau_G = \tau_Q + \tau_D$ and therefore

$$\mathrm{P}[\tau_G = n] = \sum_{k=0}^{n} \mathrm{P}[\tau_Q = k] \, \mathrm{P}[\tau_D = n{-}k] \,. \tag{16}$$

In other words, $\mathrm{P}[\tau_G{=}n]$ is the discrete convolution of $\mathrm{P}[\tau_Q{=}n]$ and $\mathrm{P}[\tau_D{=}n]$. Since we have found the distributions of τ_Q and τ_G, the distribution of τ_D can be promptly found by reversing the convolution operator, i.e., as the *discrete deconvolution* of $\mathrm{P}[\tau_G = n]$ and $\mathrm{P}[\tau_Q = n]$.

Extension: correlated vehicle speeds – In the equations above, we can consider correlation between the vehicle speeds, tuned by a real parameter r, with values between 0 and 1. Then, we replace the value of r.v. T with the same value of the preceding vehicle with probability r, and with probability $1{-}r$ we keep drawing an i.i.d. value from distribution ρ_T (for $r = 0$, the results coincide with the i.i.d. case). This model just involves correlation on the speed values, not on whether a vehicle arrivals between two subsequent slots.

IV. Numerical Results

We report a practical evaluation of the mathematical framework discussed above, considering the following numerical values. The time slot duration has been taken equal to $\tau{=}5$ seconds, packet arrival and vehicle arrival rates are $\lambda{=}0.02$ packets/second and $\mu{=}0.04$ vehicles/second, respectively. We consider a source-destination distance $d_{SD}{=}220$ m, and the vehicle speed follows a truncated Gaussian distribution with the following parameters: $V_{\min}{=}10$ m/s, $V_{\max}{=}50$ m/s, $\overline{V}{=}20$ m/s, $\sigma_V{=}10$ m/s. For this choice of speed parameters, we have $T_{\min}{=}1$ and $T_{\max}{=}5$. Fig. 2 represents the resulting distribution of the vehicle speeds. From that distribution, it is also straightforward to derive that of the traversing time of the vehicles, i.e., of the transmission delay.

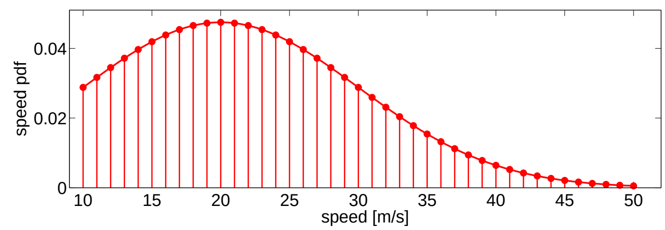

Fig. 2. Truncated Gaussian distribution of the vehicle speeds.

Finally, we set parameters $N_{\max}$ and $Q_{\max}$ to 4 and 10, respectively. These choices have negligible effect on the analysis, but enable a heavy reduction in the number of states.

We preliminary remark that all the following evaluations have been verified by means of simulation, and the simulation results have been found to be in perfect agreement with the analysis, since the analysis is exact and the simulation actually just derive the statistics through Monte Carlo iterations instead of the solution of the Markov chain.

Fig. 3 reports the cumulative distributions of the *queueing delay* τ_Q, the *global delay* τ_G, and the *delivery delay* τ_D. The queueing delay exhibits a smooth transition from low to high values, and essentially depends on how many packets are found in the transmitter SRU's buffer. Note in particular that there is a non-zero (about 10%) probability that a newly arrived packet finds this buffer empty, and therefore τ_Q is equal to 0. This does not apply to the delivery delay, which is at least

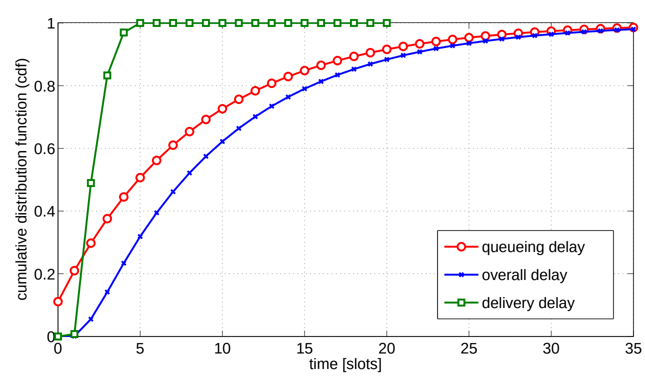

Fig. 3. Cumulative distributions of the delay terms: queueing delay τ_Q, delivery delay τ_D, and overall delay τ_G.

equal to the lowest traversing delay $T_{\min}$. The overall delay is therefore also at least $T_{\min}$. More in general, the overall delay exhibits a similar distribution to the queueing delay, however with a horizontal bias caused by the delivery delay. Such a value can be actually relevant; indeed, recall that the time slot τ is equal to 5 seconds, and τ_D can be then of the order of tens of seconds. Alternatively, we can observe that about 73% of the traffic has been sent within 50 seconds (i.e., 10 slots) since its arrival, but only 62% has also been delivered within the same time, with a fraction of about one tenth of the traffic that is still being transmitted or is waiting for some older packet to be received.

Thus, the delivery delay has a relevant effect; especially, it is due to not only the traversing time but also to resequencing at the receiver's buffer. To better highlight this fact, we focus in Fig. 4 on the distribution of the delivery delay τ_D alone. Clearly, the delivery delay cannot exceed $T_{\max}$, which happens when the packet is transported by a slow vehicle whose speed is around $V_{\min}$, and therefore it is received after a traversing time equal to $T_{\max}$, and all past packets have been surely received by then. However, the packet can even be carried by a faster vehicle, so that its traversing time can be down to $T_{\min}$, but in that case it may have to wait for the reception of some older packet, which is carried by a slower vehicle. As shown in the figure, the delivery delay distribution is for this reason skewed to the right, and the average τ_D is 2.698 slots, i.e., 13.49 seconds.

Finally, Fig. 5 investigates the effect of correlation in the speeds of subsequent vehicles. The general effect is found to be almost negligible. However, likely depends on the correlation model: preliminary results involving also correlation in the arrival process of vehicles show a more pronounced difference. Surely a thorough investigation of the impact of correlation [12] on the delays is worth of deeper investigations, which are left for future research.

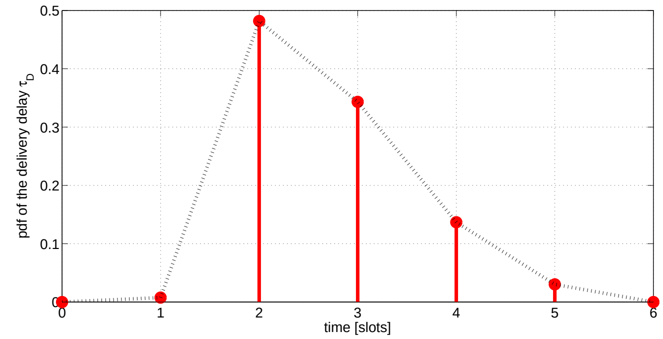

Fig. 4. Distribution of the delivery delay.

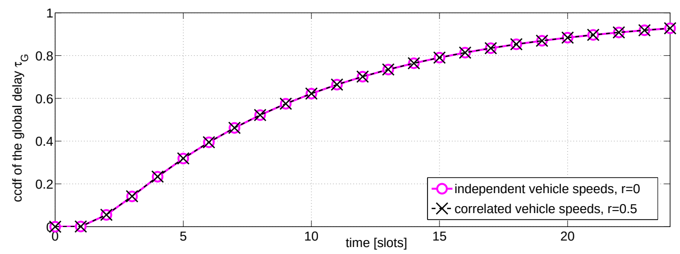

Fig. 5. Comparison of cumulative distributions of the global delay τ_G: correlated vs. independent vehicle speeds.

V. Conclusions and Future Work

We investigated an intermittently connected vehicular network scenario and we derived via Markov chains the full statistics of the delay terms experienced by a packet. Our analytical model can be used as the basis to formulate more general optimization problems where data delivery in such a scenario is subject to delay constraints.

Notice that, thanks to our proposed approach, delays are not only quantified on average, but the entire statistics is obtained for each delay term, therefore enabling also to quantify higher-order moments, such as delay jitters, and formulate stochastic constraints on the probability that the delay exceeds a given threshold.

Our analysis can also be extended to more complex investigations with a similar approach. For example, packet errors and retransmissions can also be included in the analysis; in this spirit, the investigations could borrow even more results from the studies of ARQ schemes. Moreover, more complex arrival statistics for packets and vehicles can be considered. It could be worth considering the impact of correlation models in the vehicle movements (e.g., both vehicle arrivals and speeds), which is left for future work as an interesting extension.

References

[1] M. Gerla and L. Kleinrock, "Vehicular networks and the future of the mobile internet," *Comp. Netw.*, vol. 55, no. 2, pp. 457–469, Feb. 2011.

[2] F. H. P. Fitzek, L. Badia, M. Zorzi, G. Schulte, P. Seeling, T. Henderson, "Mobility and stability evaluation in wireless multi-hop networks using multi-player games," *Proc. 2nd ACM workshop on Network and system support for games (NetGames)*, pp. 77–87, 2003.

[3] P. Pereira, A. Casaca, J. J. P. C. Rodrigues, V. N. G. J. Soares, J. Triay, and C. Cervelló-Pastor, "From delay-tolerant networks to vehicular delay-tolerant networks," *IEEE Commun. Surveys Tuts.*, vol. 14, no. 4, pp. 1166–1182, 2012.

[4] M. Emmelmann, B. Bochow, C. Kellum. *Vehicular Networking: Automotive Applications and Beyond*. Wiley, 2010.

[5] A. Abdrabou and W. Zhuang, "Probabilistic delay control and road side unit placement for vehicular ad hoc networks with disrupted connectivity," *IEEE J. Sel. Areas Commun.*, vol. 29, no. 1, pp. 129–139, Jan. 2011.

[6] N. Banerjee, M. D. Corner, D. Towsley, and B. N. Levine, "Relays, base stations, and meshes: enhancing mobile networks with infrastructure," *Proc. ACM Mobicom*, Sept. 2008.

[7] M. J. Khabbaz, W. F. Fawaz, and C. M. Assi, "A simple free-flow traffic model for vehicular intermittenly connected networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 3, pp. 1312–1326, Sept. 2012.

[8] M. J. Khabbaz, H. M. K. Alazemi, and C. M. Assi, "Delay-aware data delivery in vehicular intermittenly connected networks," *IEEE Trans. Commun.*, vol. 61, no. 3, pp. 1134–1143, Mar. 2013.

[9] J. G. Kim and M. M. Krunz, "Delay analysis of selective repeat ARQ for a Markovian source over a wireless channel," *IEEE Trans. Veh. Technol.*, vol. 49, no. 5, pp. 1968–1981, 2000.

[10] L. Badia, M. Rossi, and M. Zorzi, "SR ARQ packet delay statistics on Markov channels in the presence of variable arrival rate," *IEEE Trans. Wireless Commun.*, vol. 5, no. 7, pp. 1639–1644, Jul. 2006.

[11] V. Ramaiyan, E. Altman, and A. Kumar, "Delay optimal schedule in a two-hop vehicular relay network," *Springer Mobile Netw. Appl.*, vol. 15, pp. 97–111, 2010.

[12] L. Badia, "On the impact of correlated arrivals and errors on ARQ delay terms," *IEEE Trans. Commun.*, vol. 57, no. 2, pp. 334–338, Feb. 2009.

[13] ——, "A Markov analysis of selective repeat ARQ with variable round trip time," *IEEE Comm. Lett.*, vol. 17, no. 11, pp. 2184–2187, Nov. 2013.

[14] M. F. Neuts, *Matrix-Geometric Solutions in Stochastic Models*. New York: Dover Publications, inc., 1981.

[15] Matlab version R2011a, *www.mathworks.com*.