On Deriving and Incorporating Multi-hop Path Duration Estimates in VANET Protocols

JOSIANE NZOUONTA, MARVIN K. NAKAYAMA and CRISTIAN BORCEA
New Jersey Institute of Technology

The expected duration of multi-hop paths can be incorporated at different layers in the protocol stack to improve the performance of mobile ad hoc networks. This article presents two discrete-time and discrete-space Markov chain based methods, DTMC-CA and DTMC-MFT, to estimate the duration of multi-hop road-based paths in vehicular ad hoc networks (VANET). The duration of such paths does not depend on individual nodes because packets can be forwarded by any vehicle located along the roads forming the path. DTMC-CA derives probabilistic measures based only on vehicle density for a traffic mobility model, which in this article is the microscopic Cellular Automaton (CA) freeway traffic model. DTMC-MFT generalizes the approach used by DTMC-CA to any vehicular mobility model by focusing on the macroscopic information of vehicles rather than their microscopic characteristics. The proposed analytical models produce performance-measure values comparable to simulation estimates from the validated CA traffic model. Furthermore, this article demonstrates the benefits of incorporating expected path durations into a VANET routing protocol. Simulation results show that the network overhead associated with route maintenance can be reduced to less than half by using the expected path durations.

Categories and Subject Descriptors: C.4 [**Performance of Systems**]: Performance attributes; G.3 [**Probability And Statistics**]: Markov processes; C.2.2 [**Computer-Communication Networks**]: Network Protocols

General Terms: Performance, Measurement, Theory

Additional Key Words and Phrases: multi-hop path duration,vehicular ad hoc networks,road-based routing

1. INTRODUCTION

Vehicular ad hoc networks (VANETs) can provide scalable and cost-effective solutions for applications such as traffic safety, dynamic route planning, and context-aware advertisement using short-range wireless communication. The highly dynamic nature of these networks leads to frequent broken paths, which subsequently decrease significantly the overall network performance (e.g., low throughput, high delay, high overhead, etc.) This problem could be lessened, however, if VANET protocols could be enhanced with the ability to determine dynamically the duration of continued multi-hop connectivity. Such

Corresponding author: Cristian Borcea, Department of Computer Science, New Jersey Institute of Technology, University Heights, Newark, NJ 07102. Emails: jn62@njit.edu, marvin@cs.njit.edu, borcea@cs.njit.edu.

This material is based upon work supported in part by the National Science Foundation under Grant Numbers CMMI-0926949, CNS-0831753, and CNS-0834585. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

knowledge can be used at different layers in the protocol stack to answer questions such as:

—What should be the size of the geographic region to consider in searching for a file in a VANET peer-to-peer system?

—What are the chances to establish a TCP connection to transfer a 100 MB file from a certain node?

—When should a new route-discovery process be started in case of a broken route?

Existing work on estimating the duration of connectivity in multi-hop VANETs is mostly based on simulations [Artemy et al. 2004; 2005]. The *RoadSim* traffic simulator was used to generate vehicle movement on an input map, and connectivity properties were then extracted from the data collected. While this method provides valuable insight into the connectivity patterns of that specific map, new simulations are likely needed for other areas of interest.

Analytical approaches, on the contrary, allow for estimation based on the characteristics of the roads (e.g., traffic density and maximum allowed speeds) that can more easily be applied to different areas. A number of models have been proposed to estimate path durations in mobile ad hoc networks (MANETs) [Tseng et al. 2003; Han et al. 2006; Trivino-Cabrera et al. 2008; Yu et al. 2003; Bai et al. 2003]. One may argue that these models can be applied to VANETs by constraining the movements of the nodes to road structures and increasing the node speeds. However, such an approach would create two issues. First, the mobility models used in these approaches do not accurately approximate VANET mobility. For instance, the commonly used Random Way-point model [Johnson and Maltz 1996] does not account for vehicular traffic interactions such as acceleration and slow down due to the presence of other vehicles.

Second, the MANET models consider node-centric paths, which are defined as fixed successions of nodes between the source and destination (i.e., the nodes forming the path do not change once the path is established). In VANETs, on the other hand, many protocols use road-based paths which have been shown to lead to better performance [Naumov and Gross 2007; Nzouonta et al. 2009]. These paths consist of successions of road intersections that have, with high probability, network connectivity among them. Geographical forwarding, used along these paths, allows any node present on the road segments forming the paths to transfer packets between two consecutive intersections. Therefore, the connectivity duration for road-based paths is generally greater than that for node-centric paths (i.e., fewer broken paths). This difference is illustrated in Figure 1. As such, MANET models cannot estimate accurately the road-based path durations.

This article presents two analytical (non-simulation) methods to compute the duration of connectivity for multi-hop road-based paths in VANETs. Specifically, we present two Discrete-Time and Discrete-Space Markov Chain (DTMC) based method, DTMC-CA (Cellular Automata based DTMC) and DTMC-MFT (Mean-Field Theory based DTMC), to analytically compute connectivity measures between two endpoints along a road-based VANET path. DTMC-CA is based on the Cellular Automata (CA) freeway traffic model [Nagel and Schreckenberg 1992], which provides a set of rules governing microscopic movements of vehicles. We chose the CA model because it has been validated in the literature [Nagel and Schreckenberg 1992; Simon and Gutowitz 1998; Nagel et al. 1998] and its simplicity makes it attractive to analytical studies of vehicular traffic properties. DTMC-MFT

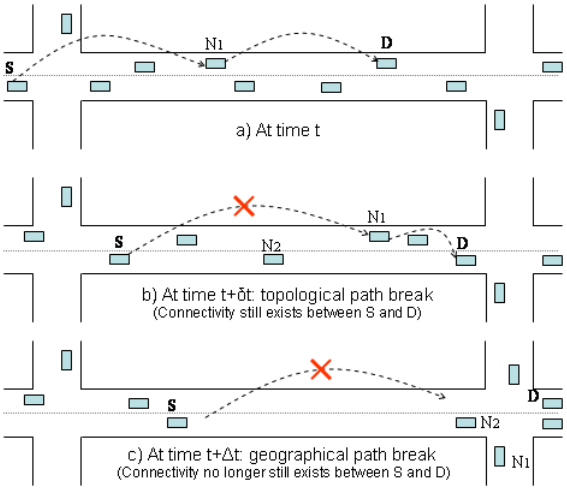


Fig. 1. This article considers the expected duration of connectivity for road-based paths, where any node can be used to forward data along road segments forming the path. For example, a road-based path will maintain connectivity when the topology changes from case (a) to case (b) because any node can be used to forward data between S and D. A node-centric (topological) path, on the other hand, will be broken under the same scenario; the break occurs because the path is fixed and must go through N_1 which is too far from S. The road-based path is broken only when there is no node to forward the packets, such as in case (c).

generalizes DTMC-CA to any vehicular mobility model by abstracting individual vehicle information and focusing on the macroscopic view of the road.

Unlike analytical models for node-centric paths, the proposed approaches do not assume independence between successive links in the network. This, however, comes at the cost of a potentially large state space. To address this issue in DTMC-CA, where we are interested in steady-state performance measures, we apply two methods to eliminate transient states [Schadschneider and Schreckenberg 1998] and to lump (i.e., to merge logically equivalent states of) the Markov chains [Kemeny and Snell 1976]. These methods lead to a decrease of more than 90% in the size of the state space in DTMC-CA without any loss of information for the computed measures. DTMC-MFT, on the other hand, generates smaller state space sizes because it uses approximations to determine vehicles velocities.

To evaluate the models, we compared their analytical results against simulation results from a (larger) CA traffic model. This evaluation shows that DTMC-CA and DTMC-MFT provide results consistent with validated models for expected steady-state duration of path connectivity and disconnectivity while taking into account the complex and dynamic interactions between the endpoints and all the intermediate vehicles. Between the models, DTMC-CA is more precise because it does not rely on Mean-Field theory vehicle speed approximations while DTMC-MFT is more scalable because it focuses on the macroscopic properties of the vehicular traffic.

Finally, this article demonstrates the benefits of incorporating path estimates into a VANET routing protocol. Simulations are performed to assess the impact of including the expected duration predictions in route maintenance component of the Reactive Road-Based with Vehicular Traffic (RBVT-R) routing protocol [Nzouonta et al. 2009], and the results show that the network overhead is reduced by up to half with the inclusion of the

DTMC-CA predictions.

The rest of this paper is organized as follows. In Section 2, we describe the cellular automaton traffic model used in our analysis and briefly discuss related work. Section 3 presents the DTMC-CA model, the state-reduction techniques, and examples to illustrate the concepts. Section 4 presents the DTMC-MFT model as well as examples to illustrate the concept. Section 5 compares numerical results of both models against simulations. The use of DTMC-CA predictions in a routing protocol and their benefits are presented in Section 6. The article concludes in Section 7.

2. BACKGROUND

In this section, we briefly describe the microscopic one-lane freeway cellular automaton (CA) traffic model [Nagel and Schreckenberg 1992] used in the analysis and then review related work in the literature.

2.1 Cellular Automaton (CA) traffic model

The CA traffic model is a discrete-time and discrete-space stochastic traffic model, which we selected because of its computational simplicity. The CA traffic model divides a road into cells, each of fixed length L_c in the direction of traffic (from left to right in Figure 2). At any point in time, a cell is either empty or occupied by at most one vehicle. Each vehicle has a speed $v \in \{0, 1, \ldots, v_{max}\}$ which changes over time, where the speed represents the number of cells advanced by a vehicle in one step (one step corresponds to one unit of time). The state of the system is updated at discrete time steps by applying the four rules listed below to all vehicles in parallel. The space interval between vehicle i and the vehicle just ahead of it is $gap(i)$.

(1) *Acceleration:* If $v(i) < v_{max}$, then $v(i) = v(i) + 1$, where $v(i)$ represents the speed of vehicle i.

(2) *Slow down due to other cars:* If $v(i) > gap(i)$, then $v(i) = gap(i)$.

(3) *Stochastic behavior:* If $v(i) > 0$, then $v(i) = v(i) - 1$ with probability p and $v(i)$ remains the same with probability $1 - p$.

(4) *Move vehicle:* $x(i) = x(i) + v(i)$, where $x(i)$ is the location of vehicle i.

Figure 2 illustrates a step of parallel update of the system. At time t (Figure 2(a)), vehicles 1, 2 and 3 have speeds of $v(1) = 2$, $v(2) = 2$ and $v(3) = 1$, respectively. At the end of the update period, vehicles 3 and 2 have their speeds equal to 1 while $v(1) = 2$. Vehicle 2 must decrease its speed because of the presence of vehicle 3 (application of rule 2, with $gap(2) = 1$, results in $v(2) = 1$). Note that the above rules are valid for single-lane roads. Rules for multi-lane traffic roads, including vehicle passing can be found in [Rickert et al. 1995].

2.2 Related Work

Continuous and discrete time/space models have been proposed for estimating path durations in mobile ad hoc networks. A discrete-time and discrete-space model for MANET path duration is proposed in [Tseng et al. 2003]. Mobile nodes move on an open 2-dimensional area according to the Random Way-point (RWP) model. The area is divided into hexagonal cells of radius r. A link between two nodes having coordinates (x, y) and (x', y') is represented as a vector of the differences of coordinates $(x' - x, y' - y)$. This

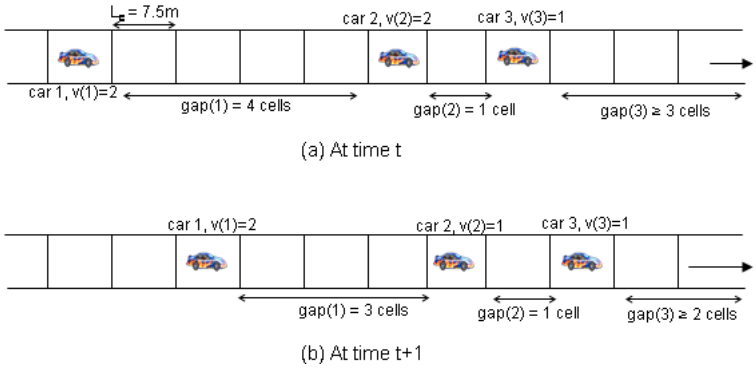


Fig. 2. A single step parallel update using the Cellular Automaton freeway traffic model.

vector also represents the state of the link. A Markov chain is created on this state space and, assuming independence between consecutive links on a path, the authors derive the expected duration of paths in the network.

A continuous model for the distribution of residual lifetime of link and path duration in MANETs using the RWP model is presented in [Han et al. 2006]. By application of Palm's theorem, the authors show that path durations converge to an exponential distribution. The authors also assessed the impact of the assumption of independence between consecutive links and found a weak correlation between them.

In vehicular networks, link durations and spatial node distributions have been studied through analytical derivations [Dousse et al. 2002; Khabazian and Mehmet 2008]. The authors derived the probability of two vehicles being connected at a time t as well as the distribution of the number of vehicles in communication range. The distribution of duration of one-hop links is also computed. Similar metrics are measured through simulations in [Fiore and Härri 2008]. In this paper, we focus on the duration of uninterrupted wireless connectivity for multihop communications in vehicular networks. A traffic simulator is used to generate vehicle movements and measure connectivity statistics on a closed loop road in [Artemy et al. 2004]. The difference of the present work is that we employ an analytical method to derive those measures.

3. DTMC-CA MODEL

This section presents DTMC-CA, a discrete-time and discrete-space Markov model of traffic movement, based on the Cellular Automata (CA) freeway model. We solve our model to compute the steady-state expected connectivity duration in vehicular networks. Recall that road-based path connectivity between two vehicle nodes is maintained as long as intermediate nodes can be found to form a multi-hop communication path, independently of the specific intermediate nodes used. A multi-hop path is a list of nodes such that the distance between any pair of successive nodes in the list is at most equal to the transmission range r. While real life wireless transmissions are subject to multipath propagation irregularities such as signal attenuation, interferences, our analytical models assume perfect transmission within the transmission range r.

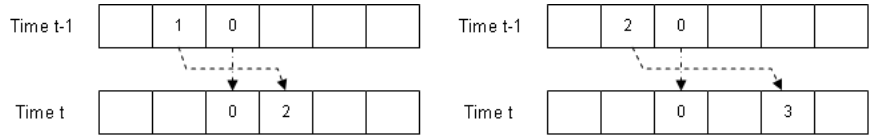


Fig. 3. Invalid states in the DTMC-CA model. On a single lane road, the configurations at time t are not possible because they would involve vehicles passing each other.

3.1 Description

We consider a road with vehicles moving with velocities from a set $\{0, 1, \ldots, v_{max}\}$. We focus on a section of the road located between two endpoints in communication through the vehicular network. This road section is divided into k cells of fixed length L_c as in Figure 2. A value from the set $V = \{0, 1, \ldots, v_{max}, \infty\}$ is associated with each cell. For each cell i, a finite positive value $v_i > 0$, $v_i \in V$, $i \in \{1, 2, \ldots, k\}$, corresponds to a vehicle in cell i moving with speed v_i. A cell value $v_i = 0$ represents a stationary vehicle, while a cell value $v_i = \infty$ corresponds to an empty cell. The source and destination of the communication are located near cells $i = 0$ and $i = k + 1$ respectively. To simplify the presentation, the following description assumes that the communication endpoints are stationary. Thus, the distance between the endpoints will not change during the communication. In the later part of this section, we discuss how this assumption can be relaxed.

We construct a Markov chain $(X_n : n \geq 0)$ defined by $M = (S, P)$, where S is the state space $S = \{s = (v_1, v_2, \ldots, v_k) : v_i \in V$, for all $1 \leq i \leq k\}$ and P is the transition probability matrix. No assumption is made on the distribution of the initial state X_0 of the Markov chain as only steady-state measures are of interest in this study. As described, the size of the state space would be $|S| = |V|^k$, which grows exponentially in the number k of cells. However, it is possible to reduce this size considerably (more than 90%) using the properties of the mobility model as well as techniques such as lumping of Markov chains.

3.2 State Space Reduction

We now discuss two techniques we apply to reduce the size of the state space. Each state $s \in S$ is of the form $s = (v_1, v_2, \ldots, v_k)$ with $v_i \in V$, for all $1 \leq i \leq k$. To limit the exponential growth in the state space, first we eliminate invalid (transient) states that violate the rules (Section 2) of the CA model (i.e., they are not accessible from other states). Then, we aggregate the remaining states in logically equivalent states using lumping.

3.2.1 *Invalid States.* The first reduction technique eliminate from S of states that violate the update rules of the mobility model used. These are transient states which cannot be reached from any other state in the system. For example, considering a road with $k = 6$ cells and $v_{max} = 2$ (Figure 3), one such invalid state is $s = (\infty, \infty, 0, 2, \infty, \infty)$. This state is invalid because of the pair of speeds $(0, 2)$ located in consecutive cells, which would require vehicles on a single-lane road to pass each other, which is not allowed in this model. From the configurations in Figure 3 at time $t - 1$, application of rule 2 would result in the vehicle at the left slowing down to speed 0 at time t. Using a similar reasoning, state $s = (\infty, \infty, 1, \infty, 3, \infty)$ is also found to violate the rules of the system (because of the configuration $(1, \infty, 3)$). Algorithm 1 outputs a state space set containing only non-transient states, for a given value of maximum speed v_{max} and number of cells k. This algorithm does not work by generating all possible states and then removing the invalid

ones because such approach would run out of memory very quickly (e.g., for $k = 10$ and $v_{max} = 5$, temporarily generating all possible states would require 7^{10} instances of vectors of 10 numbers). Instead, we prevent the formation of states known to contain invalid configurations. Lines $2-7$ show how this is achieved for $k = 1$, 2 and 3. For larger values of k (lines $9-19$), a recursive approach is used. The algorithm starts with the state set obtained for $k-1$ cells and adds one position to the left of each vector element. Note that the invalid states concept is an extension of *Garden of Eden* states described in [Schadschneider and Schreckenberg 1998].

Table I clearly confirms that removing the invalid states greatly decreases the state space size, with reduction of 85% or more when $k = 5$. The greater the maximum speed, the bigger the benefits of removing invalid states from the state space set.

Algorithm 1 Generation of Valid State Space S with length k states and Maximum Speed v_{max}

Notation:

S_m: State space set with element of the form $(v_1, v_2, ..., v_m)$
$S_{m,i,j}$: Subset of S_m with value i at position j
$V = \{0, 1, ..., v_{max}, \infty\}$: Set of possible values of v_i
$REP(i, m)$: Replicate value i, m times
$(c, S)_r$: Add c to beginning of each element of set S
$S[i : j]$: Extract positions v_i through v_j of each element of set S

1: **for** $l \leftarrow 1, k$ **do**
2: **if** $l == 1$ **then**
3: $S_1 = V$
4: **else if** $l == 2$ **then**
5: $S_2 = \{(v_1, v_2) : v_1, v_2 \in V, v_1 = \infty$ or $v_2 = \infty$ or $v_1 = 0, v_2 = 0\}$
6: **else if** $l == 3$ **then**
7: $S_3 = \{(v_1, v_2, v_3) : v_1, v_2, v_3 \in V, (v_1 = \infty, (v_2, v_3) \in S_2)$ or $(v_1 \in S_1, v_2 = \infty, v_3 = 1)$ or $(v_1 = 0, v_2 = 0, v_3 = 0)\}$
8: **else**
9: $tmpv0 = (0, S_{l-1,0,1})_r \cup (0, \infty, S_{l-1,1,1}[2 : end])_r \cup (0, \infty, S_{l-1,1,2}[2 : end])_r$
10: $tmpv1 = tmpv0$
11: **for** $e \leftarrow 2, min(l-2, v_{max})$ **do**
12: $tmpv1 = tmpv1 \cup (1, REP(\infty, e), S_{l-1,e,e+1}[e+1 : end])_r$
13: **end for**
14: $S_l = (\infty, S_{l-1})_r$;
15: $S_l = S_l \cup tmpv0 \cup tmpv1$
16: **for** $e \leftarrow 2, v_{max}$ **do**
17: $S_l = S_l \cup (V(e), tmpv1[2 : end])_r$
18: **end for**
19: **end if**
20: **end for**
21: Return S_k

Table I. Comparison of state space set sizes. "Potential" sizes represent state space set sizes with invalid states, while "actual" sizes represent state space sizes without the invalid states. Removing invalid states greatly decreases the potential computational cost.

k	1	2	5	7	9	10
$\|S\|$, $v_{max} = 5$, Potential	7	49	16807	823543	7^9	7^{10}
$\|S\|$, $v_{max} = 5$, Actual	7	19	321	2080	13460	34242
Percent decrease	0%	61%	98%	99.7%	99.97%	99.99%
$\|S\|$, $v_{max} = 2$, Potential	4	16	1024	16384	4^9	4^{10}
$\|S\|$, $v_{max} = 2$, Actual	4	10	156	979	6102	15235
Percent decrease	0%	37.5%	84.76%	94%	97.7%	98.5%

3.2.2 *Lumping the Markov Chain.* The next state-reduction technique applies the concept of lumping to create an aggregated Markov chain without losing any information, for the performance measures of interest. Other aggregation techniques can be found in the literature (e.g., nearly completely decomposability (NCD) [Courtois 1977; Dayar and Stewart 1997; P. Buchholz 1995]). In this study, the choice was made to lump the Markov chain because unlike other methods (e.g., NCD), (ordinary) lumping allows for the exact calculation of certain stationary and transient measures of the original Markov chain from the resulting lumped Markov chain [Buchholz 1994].

A lumpable Markov chain is a Markov chain in which some states can be merged together (their transition probabilities are summed), and the resulting process is also a Markov chain. A more formal definition is provided below, followed by an application of the concept in the context of the DTMC-CA model.

Consider a Markov chain X defined with $\tilde{M} = (\tilde{S}, \tilde{P})$, where $\tilde{S}$ is a finite state space and $\tilde{P} = \{\tilde{P}(i,j) : i,j \in \tilde{S}\})$ is an irreducible transition matrix. Let $\tilde{\pi} = \{\tilde{\pi}(i) : i \in \tilde{S}\}$ be a row vector denoting the stationary distribution ($\tilde{\pi}\tilde{P} = \tilde{\pi}$ and $\tilde{\pi}e = 1$, where e denotes the column vector of all 1's). Also, let $L = \{L_1, L_2, \ldots, L_m\}$ represent a partition of the state space S; i.e., $L_i \cap L_j = \emptyset$ for $i \neq j$, and $\cup_{i=1}^{m} L_i = \tilde{S}$.

Definition: The Markov chain X defined by $\tilde{M}$ is *(ordinary) lumpable* [Kemeny and Snell 1976] with respect to partition L if, for all $L_i, L_j \in L$ and all $i', i'' \in L_i$,

$$\sum_{j' \in L_j} \tilde{P}(i', j') = \sum_{j' \in L_j} \tilde{P}(i'', j').$$

When the Markov chain defined by $\tilde{M}$ is lumpable with respect to a partition L, then L can be used to create an aggregated Markov chain defined by $\hat{M} = (L, \hat{P})$ in which each element of L represents a single state. The aggregated Markov chain does not introduce any error in the computation of stationary and some transient properties [Buchholz 1994]. For example,

$$\hat{\pi}(i) = \sum_{j \in L_i} \tilde{\pi}(j).$$

Lumpable states are most easily determined using the matrix of transition probabilities of the system. However, in the DTMC-CA method, the main reasons for lumping the states is to reduce spatial and computational costs. Thus, the approach we apply takes advantage of the characteristics of the system to determine the lumpable states.

Considering the system update rules, we note that if two states x, y are equal after the first two rules of update (Section 2), then row x and row y will be identical in the transition matrix. Hence, the two states can be lumped together because they are *equivalent* from a stochastic view. The connectivity status of each state is considered when creating the lumps (a state corresponding to disconnected status should not be lumped with a state corresponding to a connected status). A state $s = (v_1, v_2, \ldots, v_k)$ is connected if $\exists i_1 < i_2 < \ldots < i_m$ such that $m \geq 2$ and $v_{i_j} \in \{0, 1, 2, \ldots, v_{max}\}$ $\forall j = 1, 2, \ldots, m$ and $i_j - i_{j-1} \leq r$, $\forall j = 2, 3, \ldots, m$, where r is the transmission range of each vehicle, i.e. the maximum distance on which a direct wireless transmission is possible between a pair of vehicle nodes.

To illustrate the method, consider a system with $k = 4$ and $v_{max} = 2$. Then, the following two states result in identical rows in the transition matrix: $s_1 = (\infty, 1, \infty, \infty)$ and $s_2 = (\infty, 2, \infty, \infty)$. This is because after the first two rules are applied both will become $(\infty, 2, \infty, \infty)$.

Table II presents the reduction in state-space size after equivalent states are lumped, taking into account the connectivity status of each state. The transmission range $r = 4$ cells was used for the values in this table. Decreases of more than 80% of the state space size can be achieved using this technique.

Table II. Benefits of lumping together equivalent states shown by comparing state space set sizes with and without lumping. The aggregation is performed on the state space set without invalid states.

k	1	2	5	7	9	10
$\|S\|$, $v_{max} = 5$, without lumping	7	19	321	2080	13460	34242
$\|S\|$, $v_{max} = 5$, with lumping	7	12	79	363	1635	3484
Percent decrease	0%	36.8%	75.4%	82.5%	87.8%	89.8%
$\|S\|$, $v_{max} = 2$ without lumping	4	10	156	979	6102	15235
$\|S\|$, $v_{max} = 2$, with lumping	4	6	55	246	1103	2336
Percent decrease	0%	40%	64.7%	74.87%	81.9%	84.7%

3.3 Transition Matrix P

Once determined, the aggregated state space is used to compute the matrix of transition probabilities $P = \{P(x, y) : x, y \in S\}$, where $P(x, y)$ represents the probability of moving from state x to state y in one time step. Considering two states $x = (x_1, x_2, \ldots, x_k)$ and $y = (y_1, y_2, \ldots, y_k)$, only a few valid transitions from x to y are possible. Below, we show how to determine the valid successive states y and the corresponding transition probabilities, given a state x. Recall that the stochastic rule of the CA traffic model (third rule) specifies that a vehicle will choose to decrease its speed, if greater than 0, with probability p and it will remain at the same speed with probability $1 - p$. Let ν_i represent the position of the next occupied cell to the right of cell i, while γ_i represents the position of the previous occupied cell to the left of cell i. Figure 4 shows some examples. Let ν_0 be the position of the first occupied cell from the left border and γ_k be the position of the first occupied cell from the right border (last occupied cell from the left). The probability p_i, for $i = 1, 2, \ldots, k$, is the probability that cell i of state x will not change in the transition from x to y. Unless modified using the rules below, each cell i has $p_i = 1$.

For $\nu_0 \leq i \leq \gamma_k - 1$:

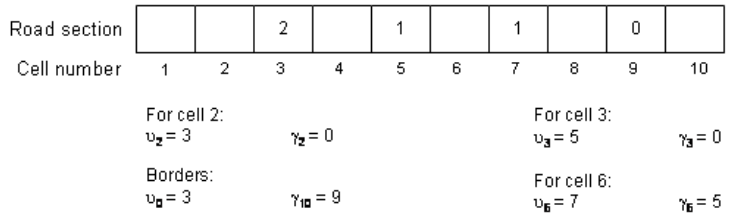


Fig. 4. Illustration of expressions used in the computation of the transition probabilities.

—If $x_i = j$, $j < v_{max}$ and $gap(i) \geq j + 1$, then $y_{i+j} = j$ with probability $p_i = p$ and $y_{i+j+1} = j + 1$ with probability $p_i = 1 - p$.

—If $x_i = j$, $j < v_{max}$ and $gap(i) < j+1$, then $y_{i+gap(i)-1} = gap(i)-1$ with probability $p_i = p$ and $y_{i+gap(i)} = gap(i) - 1$ with probability $p_i = 1 - p$.

—If $x_i = j$, $j = v_{max}$ and $gap(i) \geq j$, then $y_{i+j-1} = j - 1$ with probability $p_i = p$ and $y_{i+j} = j$ with probability $p_i = 1 - p$.

—If $x_i = j$, $j = v_{max}$ and $gap(i) < j$, then $y_{i+gap(i)-1} = gap(i) - 1$ with probability $p_i = p$ and $y_{i+gap(i)} = gap(i)$ with probability $p_i = 1 - p$.

For the border cells, we also need to take into account how new cars enter the area of observations. We do this by specifying additional probabilities for where a new car may first appear on the road segment. On the left border, a new car may enter the area of interest (through any of the cells 1 through $min(v_{max}, \nu_0 - 1)$). For example, in Figure 4, a new car may enter in cell 1 or 2. The probability that a new vehicle enters the area of observation at cell $e \in \{1, 2, \ldots, min(v_{max}, \nu_0 - 1)\}$ with speed h is $p_{e,h}$, which is a user-specified parameter of the model. For the left border

—If $x_i = \infty$ with $1 \leq i \leq min(v_{max}, \nu_0 - 1)$, then $y_i = h$, $1 \leq h \leq v_{max}$, and $y_j = \infty, j \neq i$, $1 \leq j \leq min(v_{max}, \nu_0 - 1)$, with probability $p_i = p_{i,h}$.

On the right border, the vehicle in position γ_k may leave the area of observation (if $\gamma_k > k - v_{max}$). We also take into account the possibility that the vehicle is blocked from leaving the area by other vehicles positioned in cells $k + 1, \ldots, k + v_{max} - 1$ which are beyond our observed segment. For this, we use p_β, which is the probability that a vehicle is blocked from exiting the area.

—If $x_{\gamma_k} = j$, $j < v_{max}$, $\gamma_k < k$ and $\gamma_k + j > k$, then
(1) $y_k = k - \gamma_k, y_i = \infty, \gamma_k \leq i < k$, with probability $p_i = p_\beta(1 - p)$;
(2) $y_{k-1} = k - \gamma_k - 1, y_i = \infty, \gamma_k \leq i \leq k, i \neq k$ with probability $p_i = p(p_\beta)$;
(3) $y_i = \infty, \gamma_k \leq i \leq k$, with probability $p_i = (1 - p_\beta)$.

—If $x_{\gamma_k} = j$, $j < v_{max}$, $\gamma_k < k$ and $\gamma_k + j = k$, then
(1) $y_k = k - \gamma_k, y_i = \infty, \gamma_k \leq i < k$, with probability $p_i = p(1 - p_\beta) + p_\beta(1 - p)$;
(2) $y_{k-1} = k - \gamma_k - 1, y_i = \infty, \gamma_k \leq i \leq k, i \neq k$ with probability $p_i = p(p_\beta)$;
(3) $y_i = \infty, \gamma_k \leq i \leq k$, with probability $p_i = (1 - p)(1 - p_\beta)$.

—If $x_{\gamma_k} = j$, $j = v_{max}$, $\gamma_k < k$ and $\gamma_k + j - 1 = k$, then
(1) $y_k = k - \gamma_k, y_i = \infty, \gamma_k \leq i < k$, with probability $p_i = p(1 - p_\beta) + p_\beta(1 - p)$;
(2) $y_{k-1} = k - \gamma_k - 1, y_i = \infty, \gamma_k \leq i \leq k, i \neq k$ with probability $p_i = p(p_\beta)$;

(3) $y_i = \infty, \gamma_k \leq i \leq k$, with probability $p_i = (1-p)(1-p_\beta)$.

—If $x_{\gamma_k} = j$, $j > 0$, $\gamma_k = k$, then
(1) $y_k = 0$, with probability $p_i = p_\beta$;
(2) $y_i = \infty, \gamma_k \leq i \leq k$, with probability $p_i = (1-p_\beta)$.

—If $x_{\gamma_k} = j$, $j = 0$, $\gamma_k = k$, then
(1) $y_k = 0$, with probability $p_i = p(1-p_\beta) + p_\beta$;
(2) $y_i = \infty, \gamma_k \leq i \leq k$, with probability $p_i = (1-p)(1-p_\beta)$.

—If $x_{\gamma_k} = j$, $\gamma_k < k$ and $\gamma_k + j + 1 \leq v_{max} \leq k$, then probabilities are computed as in the case of $\nu_0 \leq i \leq \gamma_k - 1$.

From these individual cells probabilities, the transition matrix entry $P(x, y)$ is computed as $P(x, y) = \prod_{i=1}^{k} p_i$.

3.4 Probabilistic Measures

The transition matrix is used to compute various probabilistic measures of interest. The state space is divided in two subsets, S_1 and S_2, such that $S_1 \cup S_2 = S$ and $S_1 \cap S_2 = \emptyset$. The subset S_1 contains the states in which the status of the system is connected while S_2 contains the states corresponding to a disconnected status. The first measure is the steady-state expected time to disconnection, for which two variations are computed: one conditional on starting in S_1 and the other independent of the connection status of the starting state. The second measure is the steady-state expected time to connection, for which two variations are also computed: one conditional on starting in S_2 and the other independent of the connection status at the start of communication. The final measure is the steady-state probability of maintaining connectivity for a period longer than a value t. Below is a brief description of how each of these measures is derived from the transition matrix.

Steady-state Expected Time to Disconnection: Given the transition matrix P, we first compute another matrix $P_r = (P_r(x, y) : x, y \in S)$ from P as follows. For each $(x, y) \in S \times S$, set $P_r(x, y) = P(x, y)$ if $y \in S_1$, and $P_r(x, y) = 0$ if $y \in S_2$. Then define $h = (1, 1, \ldots, 1)$, a vector of size $|S|$ containing all 1's and compute the vector $g = (g(x) : x \in S)$ as $g = (I - P_r)^{-1}h$ [Norris 1996]. Thus, $g(x)$ is the expected time until the DTMC hits S_2 given that it started in a state $x \in S_1$. We then average the $g(x)$ values using the steady-state probabilities $\pi(x)$, conditional on starting in S_1. Therefore, the steady-state expected time to disconnection given that the system starts in a connected state is given by

$$E[T_{conn}|X_0 \in S_1] = \frac{\sum_{x \in S_1} g(x)\pi(x)}{\sum_{y \in S_1} \pi(y)}. \tag{1}$$

Also the steady-state expected time to disconnection independent of the connection status of the starting state is given by

$$E[T_{conn}] = \sum_{x \in S_1} g(x)\pi(x). \tag{2}$$

Steady-state Expected Time to Connection: Given the transition matrix P, we first define another matrix $P_{r2} = (P_{r2}(x, y) : x, y \in S)$ from P as follows. For the variables

$(x, y) \in S \times S$, set $P_{r_2}(x, y) = P(x, y)$ if $y \in S_2$, and $P_{r_2}(x, y) = 0$ if $y \in S_1$. h and $g_2 = (I - P_{r_2})^{-1}h$ are defined as in the case above. Then, the expected time to connection given that the system starts in a disconnected state is given by

$$E[T_{disconn}|X_0 \in S_2] = \frac{\sum_{x \in S_2} g_2(x)\pi(x)}{\sum_{y \in S_2} \pi(y)}. \tag{3}$$

Also, the steady-state expected time to connection independent of the connection status of the starting state is given by

$$E[T_{disconn}] = \sum_{x \in S_2} g_2(x)\pi(x). \tag{4}$$

Steady-state Probability of Connection Duration: The last measure outputs the probability of maintaining connectivity for more than t time units. In this case too, for $(x, y) \in S \times S$, set $P_r(x, y) = P(x, y)$ if $y \in S_1$ and $P_r(x, y) = 0$ if $y \in S_2$. Then,

$$P[T > t] = \sum_{x \in S_1} \pi(x) \sum_{y \in S} P_r^t(x, y). \tag{5}$$

where P_r^t represents matrix P_r raised to the power t.

3.5 DTMC-CA Example

We now illustrate the computation of the transition matrix P through a simple example. Assume $k = 4$, $v_{max} = 2$ and the threshold for connectivity is $r = 2$. The straightforward computation of the state space S would lead to a size of $|S| = 4^4 = 256$ states. By eliminating invalid states, the size becomes $|S| = 63$, which reduces to $|S| = 26$ after the states are lumped together. Table III lists the final set S of states involved in the computation of transition matrix P. The matrix obtained is a sparse matrix, with 116 non-zero entries out of the $26^2 = 676$ total entries.

Table III. Sample state space S after state reduction steps. Set S reduced from potentially size $|S| = 256$ states to $|S| = 26$ states.

(∞ ∞ 0 ∞)	(∞ 2 ∞ ∞)	(∞ ∞ 2 0)	(∞ 0 ∞ ∞)
(∞ 2 ∞ 0)	(∞ 2 ∞ 1)	(2 ∞ ∞ 0)	(2 ∞ ∞ 2)
(∞ ∞ 2 ∞)	(∞ 2 0 ∞)	(∞ 2 0 0)	(2 ∞ 0 ∞)
(0 ∞ ∞ 0)	(0 ∞ ∞ 1)	(2 ∞ 1 0)	(2 ∞ 1 ∞)
(2 0 ∞ ∞)	(2 0 ∞ 0)	(2 0 ∞ 1)	(2 0 0 ∞)
(2 0 0 0)	(∞ ∞ ∞ ∞)	(∞ ∞ ∞ 0)	(∞ ∞ ∞ 2)
(2 ∞ ∞ ∞)	(0 ∞ ∞ ∞)		

3.6 Bidirectional Traffic

This subsection considers a stretch of a road with two lanes, and traffic moving in different directions. The source and destination endpoints are still stationary. Each lane of the road section of interest is divided in k cells of fixed length L_c, as in the single-lane case. The cells are juxtaposed on the lanes (see Figure 5). A value from the set $\acute{V} = \{0, 1, \ldots, v_{max}, \infty\}$ is associated with each cell.

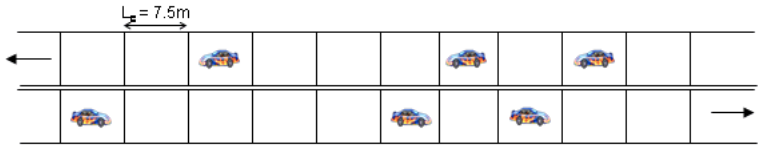


Fig. 5. Bidirectional traffic: the road interval is divided in juxtaposed cells of fixed length on each traffic lane of a bidirectional road.

A Markov chain $X = (X_n : n = 0, 1, 2, \ldots)$ defined by $\acute{M} = (\acute{S}, \acute{P})$, with state space $\acute{S} = \{\acute{s} = (v_{1,1}, v_{1,2}, \ldots, v_{1,k}, v_{2,1}, \ldots, v_{2,k}) : v_{i,j} \in \acute{V}$, for $i = 1, 2$ and $1 \leq j \leq k\}$, where $v_{i,j}$ is the speed of the vehicle in cell j of lane i and $\acute{P}$ denotes the transition probability matrix. The movement in each traffic lane is independent of the other lane. Hence, a state $\acute{s} \in \acute{S}$ can be considered as a vector of two states $\acute{s} = (s^{(1)}, s^{(2)})$ with $s^{(1)}, s^{(2)} \in S$, the state space of the single-lane traffic described in Section 3.1. The system state on lane 1 is $s^{(1)}$, while $s^{(2)}$ is the system state on lane 2.

The matrix of transition probabilities $\acute{P} = \{\acute{P}(x, y) : x, y \in \acute{S}\}$ where $\acute{P}(x, y)$ represents the probability of moving from state $x = (x^{(1)}, x^{(2)})$ to state y in one time step, is computed as $\acute{P}(x, y) = \acute{P}((x^{(1)}, x^{(2)}), (y^{(1)}, y^{(2)}))$, with $x^{(1)}, x^{(2)}, y^{(1)}, y^{(2)} \in S$. And from the independence of the lanes movements, $\acute{P}(x, y) = P(x^{(1)}, y^{(1)})P(x^{(2)}, y^{(2)})$, where $P(x^{(1)}, y^{(1)})$ is the transition probability on lane 1 from state $x^{(1)} \in S$ to $y^{(1)} \in S$. The decomposition of the state space set $\acute{S}$ into two disjoint sets $\acute{S}_1$ and $\acute{S}_2$ representing, respectively, the states with connected status and those with disconnected status, is a little different in this case. Indeed, a state of disconnection on one lane does not translate to a state of disconnection when both lanes are considered. Thus, recalling that we partitioned the state space $S = S_1 \cup S_2$ for the original single-lane model, we similarly partition $\acute{S} = \acute{S}_1 \cup \acute{S}_2$. For $\acute{s} = (s^{(1)}, s^{(2)}) \in \acute{S}$, we define a function $f(\acute{s}) = (\bar{s}_1, \bar{s}_2, \ldots, \bar{s}_k)$, where $\bar{s}_i = 1$ if $s_i^{(1)} \in \{0, 1, 2, \ldots, v_{max}\}$ or $s_{k-i+1}^{(2)} \in \{0, 1, 2, \ldots, v_{max}\}$, and $\bar{s}_i = \infty$ otherwise. Then $\acute{s} \in \acute{S}_1$ if and only if $f(\acute{s}) = (\bar{s}_1, \bar{s}_2, \ldots, \bar{s}_k)$ satisfies the following: there exists $i_1 < i_2 < \cdots < i_m$ for some $m \geq 2$, $\bar{s}_{i_j} = 1$ for all $j = 1, 2, \ldots, m$, and $i_j - i_{j-1} \leq r$ for all $j = 2, 3, \ldots, m$, where r is the transmission range.

3.7 Moving Endpoints and Lane Changes

In this case, the endpoints are moving on the road and the length of the portion of road between the source and the destination may change even as the connectivity is maintained. Additional rules, described in [Rickert et al. 1995], are needed to regulate lane changes and the passing of vehicles. The remaining computations are performed in a fashion similar to the stationary endpoints previously discussed.

4. DTMC-MFT MODEL

In this section, we present the Mean-Field Theory based DTMC Model (DTMC-MFT), a discrete-time and discrete-space Markov model of traffic movement built by abstracting the DTMC-CA model (Section 3) and incorporating Mean-Field theory results in the model [Schreckenberg et al. 1995; Schadschneider and Schreckenberg 1998]. The DTMC-CA model is designed specifically for the CA traffic model and it incorporates the microscopic properties of this traffic model in its design (e.g., vehicles speeds at each instant). Contrary to DTMC-CA, the Markov states in the DTMC-MFT model only keep track of

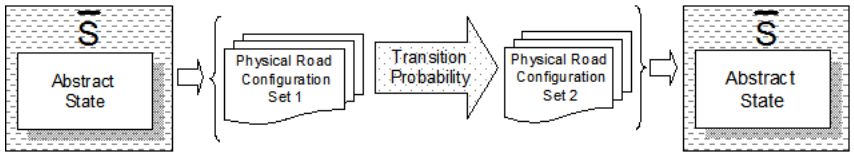


Fig. 6. Schematic view of the computation of transition probabilities in the Mean-Field theory based DTMC Model (DTMC-MFT). An abstract state is transformed into equivalent physical road configuration states with appropriate weights based on Mean-Field theory results. The transition probabilities are computed between physical states and their outputs are stored in the abstract states equivalents.

the presence of vehicles in cells, independently of their speeds. This allows for a more scalable and more adaptive approach to the computation of steady-state expected connectivity durations in vehicular networks, but at the cost of introducing some approximations. In the following, a detailed description of this model is presented along with examples to illustrate the concept.

4.1 Description

As in the DTMC-CA model, we consider a road with vehicles moving with velocities from a set $\{0, 1, \ldots, v_{max}\}$. We still focus on a section of the road located between two endpoints in communication through the vehicular network and, as before, we divide this road section into k cells of fixed length L_c. A value from the set $\bar{V} = \{1, \infty\}$ is associated with each cell. A cell value $v_i = 1$ identifies the presence of a vehicle in cell i with any speed, while a cell value $v_i = \infty$ corresponds to cell i being empty. The source and destination of the communication are located near cells $i = 0$ and $i = k + 1$ respectively. The following presentation assumes that the distance between the endpoints does not change during the communication. This assumption can be relaxed in a similar fashion as in the DTMC-CA model.

We construct a Markov chain $(X_n : n \geq 0)$ defined by $\bar{M} = (\bar{S}, \bar{P})$, where $\bar{S}$ is the state space with $\bar{S} = \{s = (v_1, v_2, \ldots, v_k) : v_i \in \bar{V}$, for all $1 \leq i \leq k\}$ and $\bar{P}$ is the transition probability matrix. No assumption is made on the distribution of the initial state X_0 of the Markov chain as only steady-state measures are of interest in this study.

DTMC-MFT is a generalization of DTMC-CA with smaller state space sizes than DTMC-CA. In DTMC-MFT, the size of the state space is independent of the maximum speed of vehicles v_{max} and can be obtained by $|\bar{S}| = |\bar{V}|^k$. Furthermore, while the DTMC-CA model is built around the CA traffic model, the DTMC-MFT model is a more generic model which can be easily adapted to other discrete-time vehicular traffic models.

4.2 Transition Matrix

The state space $\bar{S}$ is used to compute the matrix of transition probabilities $\bar{P} = \{\bar{P}(x, y) : x, y \in \bar{S}\}$, where $\bar{P}(x, y)$ represents the probability of moving from state x to state y in one time step. A schematic view of the process used to compute the elements of $\bar{P}$ is shown on Figure 6. Elements of $\bar{S}$ do not contain individual vehicle speed values, but rather only information on the presence or absence of vehicles. This abstract view of the road is translated into states of *physical road configuration*, i.e., states where known occupied cells are replaced with valid vehicle speeds. In the case of the CA freeway traffic model, the physical states obtained will all be elements of S, the state space of DTMC-CA. Thus,

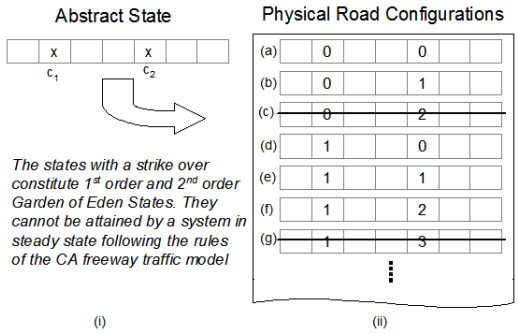


Fig. 7. Transformation of abstract state to physical road configurations for the CA traffic model in DTMC-MFT.

the same expressions for the transition probabilities between states of DTMC-CA can be applied between physical road configurations of DTMC-MFT. Additional information required is the probability of having a specific speed value in each occupied cell. We obtain this information from published Mean-Field theory results on the physics of vehicular traffic movements [Schreckenberg et al. 1995; Schadschneider and Schreckenberg 1998; Chowdhury et al. 2000].

Let ϑ_i, $i = 0, 1, \ldots, v_{max}$, represent the steady state probability of having a vehicle with speed i on the road. For $v_{max} > 1$, we can obtain the values of ϑ_i using the following expressions from [Schreckenberg et al. 1995]:

$$\begin{aligned}
\vartheta_0 &= \vartheta^2 \frac{1+pd}{1-pd^2} \\
\vartheta_1 &= q\vartheta^2 d \frac{1+d+pd^2}{(1-pd^3)(1-pd^2)} \\
\vartheta_\alpha &= \frac{1+(q-p)d^\alpha}{1-pd^{\alpha+2}} d\vartheta_{\alpha-1} - \frac{qd^\alpha}{1-pd^{\alpha+2}} \vartheta_{\alpha-2}, \text{ for } 2 \leq \alpha \leq v_{max}-2 \\
\vartheta_{v_{max}-1} &= \frac{1-qd^{v_{max}}}{1-d^{v_{max}}(q+pd)} qd^{v_{max}-1} \vartheta_{v_{max}-2} \\
\vartheta_{v_{max}} &= \frac{qd^{v_{max}}}{1-qd^{v_{max}}} \vartheta_{v_{max}-1}
\end{aligned}$$

where p is the probability that a vehicle decides to decrease its speed (third rule of CA model), $q = 1 - p$, the traffic density is ϑ with $\vartheta = \sum_{i=0}^{v_{max}} \vartheta_i$, and $d = 1 - \vartheta$. The traffic density is the average number of vehicles per unit of distance on the road. Alterations provided in [Schadschneider and Schreckenberg 1998] account for the effects of *Garden of Eden* states in these formulas.

We now describe how the values ϑ_i, $i = 0, 1, \ldots, v_{max}$, are used in the computation of the transition probabilities $\bar{P}(x, y)$ with $x, y \in \bar{S}$, $x = (x_1, x_2, \ldots, x_k)$ and $y = (y_1, y_2, \ldots, y_k)$. Let x_s be an occupied cell of x and let $\Psi_s, \Psi_s \leq v_{max}$ be the maximum speed allowed by the traffic model in the cell x_s. To determine Ψ_s, we need to identify all the positions where the vehicle in x_s could have been in the previous time step. For example, in Figure 7, to determine Ψ_{c_2}, we observe that there is a vehicle in c_1 and there are two empty spaces between c_1 and c_2. Thus, the maximum speed in c_2

is $\Psi_{c_2} = 2$. Once Ψ_s is determined, we compute the transition probabilities from x_s as $\sum_{i=0}^{\Psi_s} \frac{\vartheta_i}{\vartheta} p_{s,i}$ with $p_{s,i}$ corresponding to the transition probability p_s value obtained using the DTMC-CA expressions derived in Section 3.3, given that $x_s = i$. Once the matrix of transition probabilities is obtained, the probabilistic measures of interest are computed as in DTMC-CA model.

4.3 DTMC-MFT Example

This section details the transformation of an abstract state into equivalent physical road configurations and the computation of the transition probabilities for occupied cells. The abstract state in Figure 7 contains two vehicles on a 7-cells area. A subset of the equivalent physical speed states are shown on Figure 7(ii). Note that because of the CA rules, physical states (c) and (g) are not valid states (Section 2).

We now illustrate how to compute the transition probabilities between two abstract states. Suppose, for example, that we are interested in $\bar{P}(x, x)$ where x is the abstract state in Figure 7(i). Since the vehicles do not move in this transition, the only physical speed configuration of interest is the physical configuration (a) from Figure 7(ii). Indeed, if either occupied cell had a velocity $v > 0$, then, by the CA rules, that vehicle would change cell in the next time step because it is not being blocked by any obstacle. Thus $\bar{P}(x, x) = \frac{\vartheta_0}{\vartheta} P(0, 0)$ x $\frac{\vartheta_0}{\vartheta} P(0, 0)$ where $P(0, 0)$ is the DTMC-CA transition probability for the corresponding cells and speeds.

Similar processing is made for bidirectional traffic roads as well as roads with external obstacles such as traffic stop-signs.

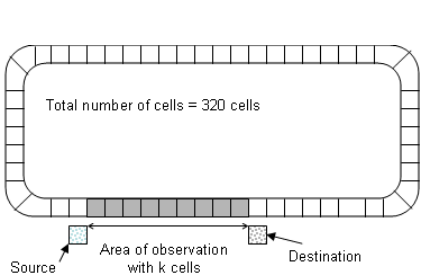
5. PERFORMANCE EVALUATION

This section presents the evaluation of the DTMC-CA and DTMC-MFT models. The goal is to verify whether the outputs from these analytical models match statistics from data collected through simulations of a larger CA traffic model. In the following, we present the evaluation methodology, the metrics used for assessing the performance, and an analysis of the results obtained.

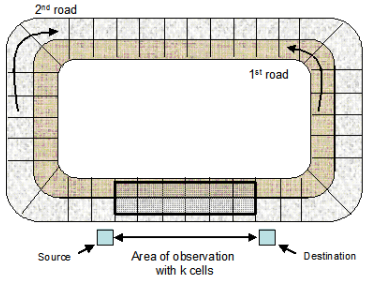
5.1 Evaluation Methodology

The analytical models are evaluated using 3 scenarios: a single-lane one-direction scenario, a two-lane bidirectional road and an intersection operated using a stop-sign. The analytical results are compared against CA traffic simulations on roads in the form of a closed rings (Figure 8). The closed ring layout ensures a constant density on the road during the simulations. Each ring layout contains a large number of cells from which a fixed portion is selected as the interval between source and destination. Data collection is made only on this portion of the ring (shaded area on figures). Even though the large system (ring) has an impact on the connectivity of the shaded area, the analytical models consider only the selected interval in the computations. However, this evaluation will show that they are able to account for the impact of the whole ring.

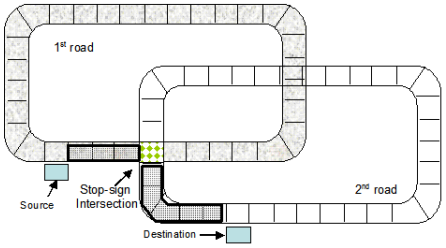
Each road in the layouts is a closed loop with $L = 320$ cells and there are k cells between the endpoints. The length of a cell is often considered equal to $7.5m$ in the literature. This value is used, in part, because the average vehicle length is about $5m$, thus the distance $7.5m$ accounts for about $2.5m$ safety distance for both front and rear of vehicle. To measure the effects of density $\vartheta = \frac{N}{L}$, the number of vehicles N on the loop is varied. The first



(a) Single-lane one-direction



(b) 2-lane Bidirectional



(c) Intersection with Stop-sign

Fig. 8. Road layouts used in the simulation scenarios

Table IV. Simulation Setup

Parameter	Value
Simulation area	320 cells. Cell length $\geq$ 7.5m
Number of vehicles	35-50-75
Transmission range	4-13 cells
Vehicle velocity	2 - 5 cells per second

3600 seconds of simulation time are discarded to eliminate initial transient effects of movement. Next, a total of about 3.2×10^6 CA traffic steps are generated and the statistics of connectivity are measured. More specifically, for each $i = 1, 2, \ldots, 1000$, we start an observation at a randomly chosen time point uniformly from the interval $[3200i, 3200i+100]$, which results in a sample T_i. To ameliorate any remaining autocorrelations present in the collected observations, we then use the method of batch means (with 10 batches) to construct a 95% confidence interval for the mean. These confidence intervals are shown in the figures. The simulation parameters are summarized in Table IV. In the intersection with stop-sign scenario, vehicles come to a full stop at the intersection before being allowed (if possible) to cross the intersecting road. Note that the two roads share the junction cell containing the stop-sign signal.

The analytical models are calibrated to the simulations. The CA randomization factor is set to $p = 0.5$; Other calibration parameters include p_i, the probability that a vehicle enters the interval of observation (shaded area in Figure 8) with speed i, and p_β the probability that a vehicle is blocked from exiting the shaded area. To obtain these parameters, we proceeded as follows. After eliminating the initial 3600 seconds of simulation time, we

kept track of where each new vehicle entered the observed segment. The fraction of those that entered in cell i was then used as our value for p_i in our analytical models. We applied a similar idea for determining a value for p_β. For all of the vehicles that tried to leave the observed segment, we noted the fraction that could not leave because there was another vehicle in cell $k+1$ that prevented the potentially leaving vehicle from actually leaving because of rule 2 of the CA traffic model. This fraction was then used for p_β. These resulting values obtained function of the density are: for 35 vehicles, $p_1 = 0.012$; $p_2 = 0.073$; $p_3 = 0.0067$; $p_4 = 0.0322$; $p_5 = 0.0285$; $p_\beta = 0.10$; for 50 vehicles, $p_1 = 0.026$; $p_2 = 0.0130$; $p_3 = 0.0096$; $p_4 = 0.0276$; $p_5 = 0.023$; $p_\beta = 0.3060$; for 75 vehicles, $p_1 = 0.048$; $p_2 = 0.0211$; $p_3 = 0.0131$; $p_4 = 0.0202$; $p_5 = 0.0148$; $p_\beta = 0.446$.

5.2 Metrics

The models are evaluated by varying the threshold connectivity range, the network density, and the time needed for connectivity. The metrics used to assess the performance are the following:

—**Steady-state expected duration of connectivity.** This metric defines the amount of time a source can expect to communicate with a destination multiple hops away. Two values of this metric are reported, depending on the existence (or not) of at least one communication path at initial time. These values correspond to equation (1) and equation (2) respectively.

—**Steady-state expected duration of disconnectivity.** This metric defines the amount of time a source can expect not to be able to communicate with a destination multiple hops away. Two values of this metric are reported, depending on the non-existence (or not) of at least one communication path at initial time. These values correspond to equation (3) and equation (4) respectively.

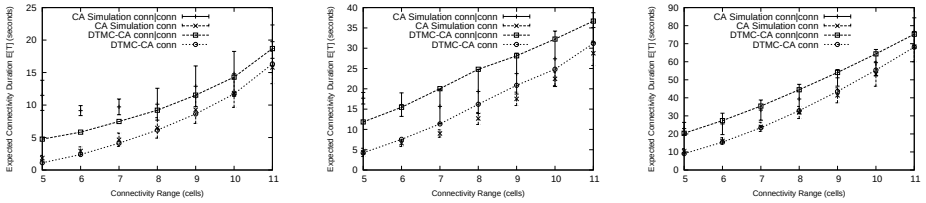
—**Steady-state probability of connectivity duration.** This metric defines, in steady-state, the probability of uninterrupted connectivity lasting at least for a time duration t. It corresponds to equation (5). Using this metric, one can evaluate the independence of connectivity estimations on adjacent portions of road.

5.3 DTMC-CA Numerical and Simulation Results

Expected Duration of Connectivity: Figure 9 shows that the expected duration of connectivity values obtained using the DTMC-CA model match well with the results from the CA simulations. Comparing analytical and simulation results for a given density, we observe that for transmission ranges of 7 cells and greater, the output from the analytical results are close to the simulation results. Another observation from this figure is that, as expected, an increase in the transmission (or connectivity) range leads to an increase in the expected duration of connectivity for a fixed value of k.

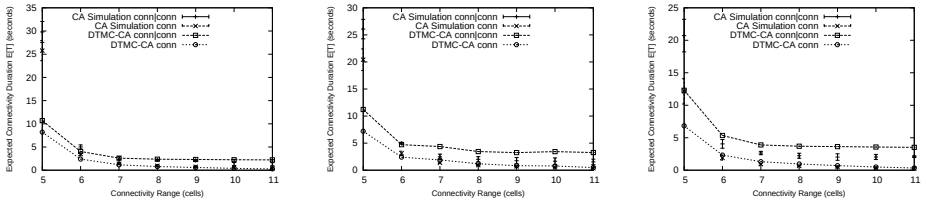
Expected Duration of Disconnectivity: Figure 10 shows a very good agreement between expected duration of disconnectivity values obtained using the DTMC-CA model and the CA simulation results. Over different densities, the outputs from DTMC-CA are within 2 seconds of the values obtained through simulations, for transmission ranges of 6 cells and higher. Combined with the expected connectivity duration results, these results show that the DTMC-CA model provides a good level of accuracy, and it can be used by protocols at different layers of the protocol stack.

Interestingly, this figure also shows that the expected duration of disconnectivity, given



(a) 35 vehicles (b) 50 vehicles (c) 75 vehicles

Fig. 9. Expected duration of connectivity when the system is initially in a connected state (or not) with different node densities and different connectivity (transmission) ranges. $v_{max} = 5$ cells per time step and $k = 12$ cells.



(a) 35 vehicles (b) 50 vehicles (c) 75 vehicles

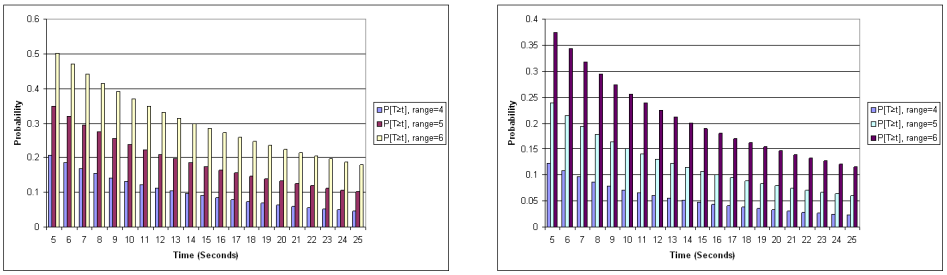
Fig. 10. Expected duration of disconnectivity when the system is initially in a disconnected state (or not) with different node densities and different connectivity (transmission) ranges. $v_{max} = 5$ cells per time step and $k = 12$ cells.

that the system starts in a disconnected state, does not decrease as the density increases. This contrasts with Figure 9 where a clear difference in expected connectivity duration values can be observed as the density increases. Thus, longer periods of network connectivity do not necessarily translate in shorter periods of network disconnection using the CA traffic model. Further tests are required with other vehicular traffic models to identify whether this is an artifact of the CA model.

Probability of Connectivity Duration: The probability of connectivity duration represents the likelihood of uninterrupted connectivity between the endpoints lasting at least t steps. Figure 11 shows the probabilities corresponding to different transmission range values and different number of cells k. As expected, longer uninterrupted connectivity periods are less likely than shorter periods, independent of the number of cells k between the endpoints. Additionally, the probabilities decrease when the number of cells, k, increases.

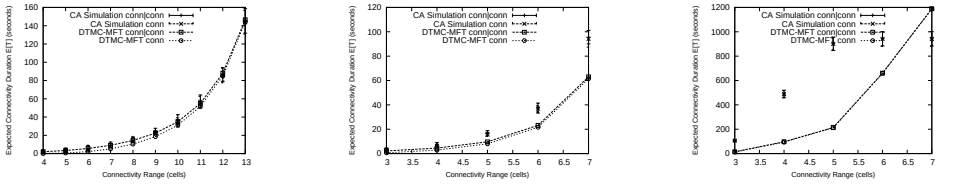
5.4 DTMC-MFT Numerical and Simulation Results

Expected Duration of Connectivity: Figure 12 shows that the expected duration of connectivity values obtained using the DTMC-MFT model have an overall good match with the results from the CA simulations for simple configurations. For more complex scenarios, the approximations in DTMC-MFT lead to estimates that are not as accurate, but the model still provides qualitatively similar results to the simulation outputs. Thus the DTMC-MFT model, while using only vehicle presence information in each state together



(a) k = 8 cells (b) k = 10 cells

Fig. 11. Probability of connectivity duration greater or equal to different time limits, for different connectivity (transmission) range thresholds using the DTMC-CA model.



(a) Single Lane One Direction (b) Two Lanes Two Directions (c) Intersection with Traffic Stop

Fig. 12. Expected duration of connectivity with DTMC-MFT model when the system is initially in a connected state (or not) with different connectivity (transmission) ranges and different layout scenarios. $v_{max} = 2$ cells per time step. In case (a), $k = 14$ cells and in cases (b and c), $k = 8$ cells.

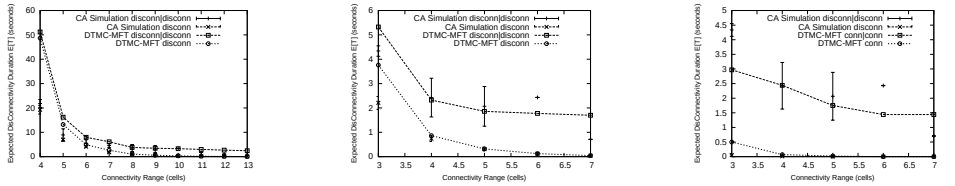
with Mean-Field theory results, is able to display behavior similar to simulation results. Comparing analytical and simulation results for the different scenarios, we observe that the closest matching is obtained in the case of the one-lane one-direction scenario.

Expected Duration of Disconnectivity: Figure 13 shows a close match between expected duration of disconnectivity values obtained using the DTMC-MFT model and the CA simulation results. The figures confirm, as one might expect, that the two lanes bidirectional road would lead to smaller average expected disconnection times than the one lane scenario.

5.5 Approximating Connectivity for Longer Paths

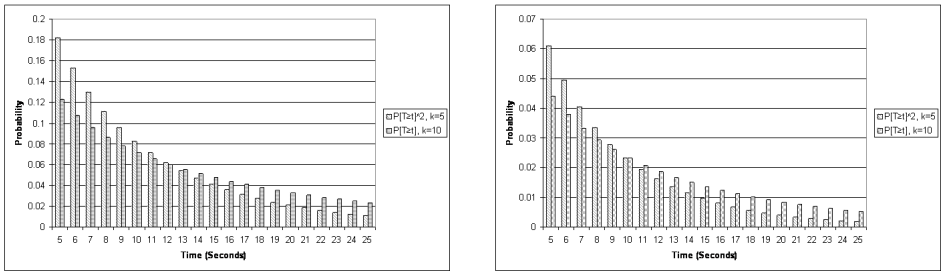
So far, we have shown that the analytical models output expected duration of connectivity as well as expected duration of disconnectivity with relatively good accuracy when compared with the measures estimated by simulation on the larger CA traffic model on a loop. This accuracy is obtained without making any assumption of probabilistic independence between any pairs of communication links between the source, destination, and any intermediate vehicles on the observed segment. The cost is high memory and computation requirements.

In this section, the question addressed is whether probability measures from adjacent sub-stretches of a road segment can be considered as independent. If sub-stretches are found independent or weakly correlated, then the probability of connectivity duration of a



(a) Single Lane One Direction (b) Two Lanes Two Directions (c) Intersection with Traffic Stop

Fig. 13. Expected duration of disconnectivity with DTMC-MFT model when the system is initially in a disconnected state (or not) with different connectivity (transmission) ranges and different layout scenarios. $v_{max} = 2$ cells per time step. In case (a) $k = 14$ cells and in cases (b and c), $k = 8$ cells.



(a) Range 4 cells (b) Range 5 cells

Fig. 14. Probability of connectivity duration greater or equal to different time limits, for different source-destination lengths. For a distance d and a transmission range r, the probability of connectivity duration $\geq t$ is comparable to the square of the same probability for a distance $\frac{d}{2}$ and a range r using the DTMC-CA model.

long road stretch can be approximated by simple multiplication of probabilities of smaller parts.

To verify independence between adjacent sub-stretches of a road segment, two ranges, $r = 4$ cells and $r = 5$ cells, and two number of cells, $k = 5$ cells and $k = 10$ cells are considered. The values of k were selected so that one is half of the other. For a given transmission range r, the values $P[T > t]_{k=10,r}$ and $P[T > t]^2_{k=5,r}$ are computed, and we compare the two values. If adjacent sub-stretches of a road segment are independent, the two values are equal, so their difference provides a measure of the dependence. If the difference is small, then it may be reasonable to approximate $P[T > t]_{k=m,r}$ for large m by $P[T > t]^2_{k=m/2,r}$.

Figure 14 shows the results of the comparison of $P[T > t]_{k=10,r}$ and $P[T > t]^2_{k=5,r}$ using the DTMC-CA model. Both sets of probability values are comparable, though not equal. Certain values of time steps t lead to closer match between values of $P[T > t]_{k=10,r}$ and $P[T > t]^2_{k=5,r}$. For example, for $t = 12$ seconds (assuming 1 step is 1 second) and $r = 5$ cells, the expected values for a larger stretch can be approximated from a smaller sub-stretch (about 0.001 or 2% difference in the probability values). This suggests that although not exactly independent, smaller sub-stretches can be used to approximate longer

road stretches with a small error. This can be used to approximate DTMC-CA estimates of duration of connectivity on longer road segments.

6. A CASE STUDY OF INCORPORATING PATH ESTIMATES IN VANET PROTOCOLS

The path estimates derived using the models presented here can be incorporated at different layers in the protocol stack to improve VANET performance. This section illustrates this idea by showing how DTMC-CA path estimates can be used to improve the route maintenance performance of the RBVT-R routing protocol [Nzouonta et al. 2009]. We start this section with a brief overview of the RBVT-R routing protocol followed by the implementation and simulation results of the RBVT-R route maintenance using path estimates.

6.1 RBVT-R Routing

RBVT-R is a reactive source routing protocol for VANET which creates road-based paths using connected road segments between source and destination. A source vehicle which wishes to communicate with a destination node first broadcasts a route request packet in the network. Intermediate nodes re-broadcast the route request packet, recording the traversed road intersections in the packet header. This sequence of intersections constitutes the RBVT-R road-based path of connected segments between the endpoints. This path is added to the data packets headers (i.e., road-based source routing) and geographical forwarding is performed along the individual road segments forming the path.

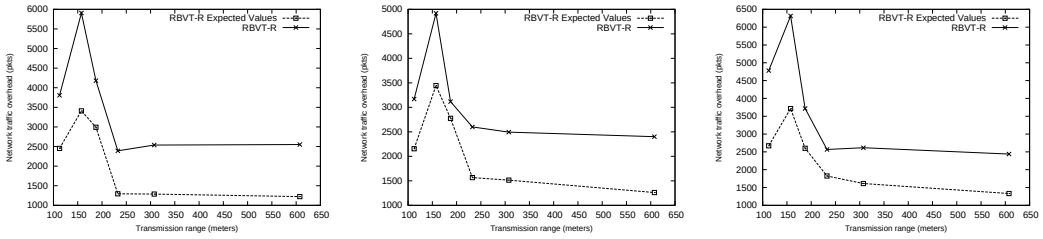
In case of a route error, (i.e., no forwarding node can be found to reach the next intersection on the path), the source of communication holds transmissions for a nominal period of time and then re-attempts, for a number of times, to use the same route. Currently, the length of the wait time is determined through heuristics. A better strategy is to compute this waiting time as function of the expected duration of disconnectivity of the path. Thus, the waiting time will be dynamically estimated for each communication.

DTMC-CA provides the average duration of the disconnectivity on the path. To apply it, we use a linear increment method. When the source receives a route error, it waits for one third of the expected duration of disconnectivity, after which it tries to send data packets through the same path. If the path is still broken, the source increases the waiting time to $\frac{2}{3}$ of the average value. A new route discovery is launched after the third failed attempt.

6.2 Simulation Setup and Metrics

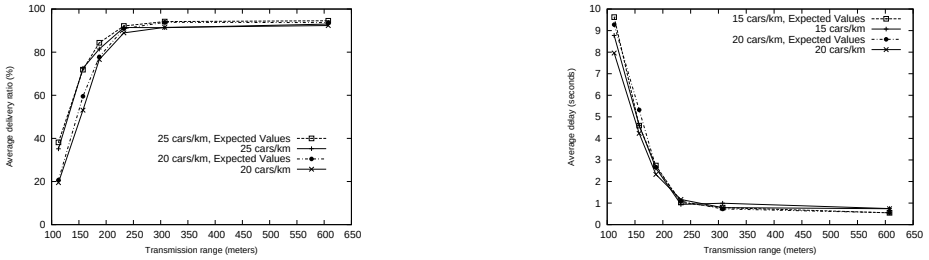
We use NS-2 [Network Simulator 2] to evaluate the impact of the modified route maintenance on RBVT-R routing performance. We simulate a road with a ring-layout and a circumference of 2400m. An obstacle in the middle prevents communication through the center of the ring. The endpoints are positioned 600m away and they exchange Constant Bit Rate (CBR) UDP packets at a rate of 4 packets/sec. The transfer lasts for 600sec. We vary the transmission range from 100m to 600m and the vehicle densities are varied from 15 vehicles/km to 25 vehicles/km. Each data point is averaged over 5 simulation runs.

The metrics used to assess the performance are the network overhead (i.e., the total number of control packets associated with the routing protocol), the end-to-end packet delivery ratio, and the average end-to-end packet delay incurred in the transmissions. The results compare these metrics for two cases: original RBVT-R, and RBVT-R with path estimates for route maintenance.



(a) 15 cars/km (b) 20 cars/km (c) 25 cars/km

Fig. 15. Network overhead vs. transmission range for the original RBVT-R and the RBVT-R with path estimates



(a) Delivery ratio (b) Average end-to-end delay

Fig. 16. Delivery ratio and average end-to-end delay vs. transmission range for the original RBVT-R and the RBVT-R with path estimates

6.3 Simulation Results

Figure 15 shows a network overhead decrease of up to half for RBVT-R with path estimates versus the original RBVT-R. This is because source nodes are able to use the path duration estimates to better determine an appropriate amount of waiting time when a route break occurs. Therefore, many unnecessary control packets associated with new route discoveries are avoided. We note that this decrease in the overhead does not negatively affect the delivery ratio of successful packets (Figure 16(a)) or the average end-to-end delay (Figure 16(b)) in the network. The average delay values are comparable because the paths traversed by the data packets are similar for both the original and modified RBVT-R. The delivery ratios are comparable as well, as both schemes transfer as much data as possible during the periods with network connectivity. These results show that integrating the expected duration of disconnectivity in RBVT-R provides a net gain for the overall protocol performance.

While these results clearly show the benefits of the proposed analytical methods on road-based routing protocols, we now discuss two ways by which these analytical methods can also be used to improve the performance of other classes of VANET routing protocols such as node-centric protocols. First, for node-centric source-routing protocols in VANETs, a source node could use the estimates from DTMC-CA or DTMC-MFT to restrict new route discoveries only to those road segments previously used for communication. This could

significantly improve the routing overhead of those protocols. Additionally, the knowledge of the expected road-based path connectivity duration can be used to limit the number of local recovery attempts on a node-centric path.

From a practical point of view, and independently of the classes of VANET protocols, the main requirement needed to ensure adequate usage of the analytical path estimates in VANET protocols is the determination of the traffic density and the probabilities of speed of entry on the roads. This could be done by using existing transportation sensors placed at entry points along the roads or by employing road-side wireless sensor networks [Bohli et al. 2008] to dynamically compute the required values and share them with the passing cars.

7. CONCLUSIONS

The duration of multi-path connectivity in VANETs is currently mostly determined through simulations. This article presented two analytical methods to estimate the duration of connectivity as well as the duration of disconnectivity for road-based paths in VANETs. DTMC-CA and DTMC-MFT are two discrete-time and discrete-space Markov chain models that track the evolution of connectivity between the endpoints in communication. The comparison between analytical results and simulation results showed that these models are able to estimate, with high accuracy, the duration of path connectivity and disconnectivity. DTMC-CA provides a greater precision for the CA traffic model while DTMC-MFT is more scalable and adaptive to other traffic models. Additionally, this article demonstrated that integrating the path duration estimates derived at application layer into VANET protocols at different layers can improve network performance. Specifically, simulation results showed that the expected disconnectivity duration can be used to reduce to less than half the network overhead for a VANET routing protocol.

REFERENCES

ARTEMY, M. M., ROBERTSON, W., AND PHILLIPS, W. J. 2004. Connectivity in inter-vehicle ad hoc networks. In *Proceedings IEEE ECE Canadian Conference (CCECE)*. Niagara Falls, Canada, 293–298.

ARTEMY, M. M., ROBERTSON, W., AND PHILLIPS, W. J. 2005. Assignment of dynamic transmission range based on estimation of vehicle density. In *Proceedings 2nd ACM International Workshop on Vehicular Ad Hoc Networks (VANET)*. ACM, Cologne, Germany, 40–48.

BAI, F., SADAGOPAN, N., AND HELMY, A. 2003. Important: a framework to systematically analyze the impact of mobility on performance of routing protocols for adhoc networks. In *Proceedings IEEE International Conference on Computer Communications (INFOCOM)*. San Francisco, CA, USA, 825–835.

BOHLI, J.-M., HESSLER, A., UGUS, O., AND WESTHOFF, D. 2008. A secure and resilient WSN roadside architecture for intelligent transport systems. In *Proceedings 1st ACM Conference on Wireless Network Security (WiSec)*. Alexandria, VA, USA, 161–171.

BUCHHOLZ, P. 1994. Exact and ordinary lumpability in finite markov chains. *Journal of Applied Probability 31,* 2, 59–75.

CHOWDHURY, D., SANTEN, L., AND SCHADSCHNEIDER, A. 2000. Simulation of vehicular traffic: A statistical physics perspective. *Computing in Science and Engineering 2,* 5 (September–October), 80–87.

COURTOIS, P.-J. 1977. Decomposability: Queueing and Computer System Applications. Academic Press.

DAYAR, T. AND STEWART, W. J. 1997. Quasi-lumpability, lower bounding coupling matrices, and nearly completely decomposable markov chains. *SIAM Journal on Matrix Analysis and Applications 18,* 2 (April), 482–498.

DOUSSE, O., THIRAN, P., AND HASLER, M. 2002. Connectivity in ad-hoc and hybrid networks. In *Proceedings IEEE International Conference on Computer Communications (INFOCOM)*. New York, NY, USA, 1079–1088.

FIORE, M. AND HäRRI, J. 2008. The networking shape of vehicular mobility. In *Proceedings 9th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC)*. Hong Kong, China, 261–272.

HAN, Y., LA, R. J., MAKOWSKI, A. M., AND LEE, S. 2006. Distribution of path durations in mobile ad-hoc networks: Palm's theorem to the rescue. *Elsevier Computer Networks 50,* 12 (August), 1887–1900.

JOHNSON, D. B. AND MALTZ, D. A. 1996. Dynamic source routing in ad hoc wireless networks. *Mobile Computing 353,* 5, 153–161.

KEMENY, J. G. AND SNELL, J. L. 1976. Finite Markov Chains. Springer-Verlag, New York.

KHABAZIAN, M. AND MEHMET, M. K. 2008. A performance modeling of connectivity in vehicular ad hoc networks. *IEEE Transactions on Vehicular Technology 57,* 4 (July), 2440–2450.

NAGEL, K. AND SCHRECKENBERG, M. 1992. A cellular automaton model for freeway traffic. *Journal de Physique I 2,* 12, 2221–2229.

NAGEL, K., WOLF, D. E., WAGNER, P., AND SIMON, P. 1998. Two-lane traffic rules for cellular automata: A systematic approach. *Physical Review E 58,* 2 (August), 1425–1437.

NAUMOV, V. AND GROSS, T. 2007. Connectivity-aware routing (CAR) in vehicular ad hoc networks. In *Proceedings IEEE International Conference on Computer Communications*. Anchorage, AK, USA, 1919–1927.

NETWORK SIMULATOR 2. http://www.isi.edu/nsnam/ns. Last accessed April 2010.

NORRIS, J. R. 1996. Markov Chains. Cambridge University Press.

NZOUONTA, J., RAJGURE, N., WANG, G., AND BORCEA, C. 2009. VANET routing on city roads using real-time vehicular traffic information. *IEEE Transactions on Vehicular Technology 58,* 7.

P. BUCHHOLZ. 1995. Lumpability and nearly-lumpability in hierarchical queueing networks. In *Proceedings IEEE International Computer Performance and Dependability Symposium*. Los Alamitos, CA, USA, 82–91.

RICKERT, M., NAGEL, K., SCHRECKENBERG, M., AND LATOUR, A. 1995. Two lane traffic simulations using cellular automata. http://arxiv.org/abs/cond-mat/9512119. Last accessed April 2010.

SCHADSCHNEIDER, A. AND SCHRECKENBERG, M. 1998. Garden of Eden states in traffic models. *Journal of Physics A: Mathematical and General 31,* 11 (March), 225–231.

SCHRECKENBERG, M., SCHADSCHNEIDER, A., NAGEL, K., AND ITO, N. 1995. Discrete stochastic models for traffic flow. *Physical Review E (Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics) 51,* 4 (April), 2939–2949.

SIMON, P. M. AND GUTOWITZ, H. A. 1998. Cellular automaton model for bidirectional traffic. *Physical Review E 57,* 2 (February), 2441–2444.

TRIVINO-CABRERA, A., DE-LA NAVA, J. G., CASILARI, E., AND GONZALEZ-CAETE, F. J. 2008. Application of path duration study in multihop ad hoc networks. *Telecommunication Systems 38,* 1-2 (June), 3–9.

TSENG, Y.-C., LI, Y.-F., AND CHANG, Y.-C. 2003. On route lifetime in multihop mobile ad hoc networks. *IEEE Transactions on Mobile Computing 2,* 4 (October–December), 366–376.

YU, D., LI, H., AND GRUBER, I. 2003. Path availability in ad hoc network. In *Proceedings 10th International Conference on Telecommunications*. Tahiti, Papeete, French Polynesia, 383–387.