

Controlling a Network of Signalized Intersections From Temporal Logical Specifications

Samuel Coogan, Ebru Aydin Gol, Murat Arcaç, and Calin Belta

Abstract—We propose a framework for generating a control policy for a traffic network of signalized intersections to accomplish control objectives expressed in linear temporal logic. Traffic management indeed calls for a rich class of objectives and offers a novel domain for these formal methods tools. We show that traffic networks possess structural properties that allow significant reduction in the time required to compute a finite state abstraction. We further extend our approach to a probabilistic framework by modeling the traffic dynamics as a Markov Decision Process.

I. INTRODUCTION

Due to the growing need for intelligent use of existing transportation infrastructure [1], control of networks of signalized intersections has received considerable attention in recent decades, see [2] for a review. Many existing strategies focus on limited objectives such as maximizing throughput [3] or maintaining stability of network queues [4], [5]. However, efficient traffic management often calls for a range of objectives beyond those mentioned above. In this work, we consider control objectives expressible using *linear temporal logic (LTL)* [6], [7]. For example, LTL formula allows objectives such as “infinitely often, the queue length on road ℓ should reach 0” and “anytime link ℓ becomes congested, it eventually becomes uncongested” and “traffic flow throughput is always greater than a given threshold.”

In this paper, we leverage recent results on control synthesis from LTL specifications such as [8]–[13] to design signal control policies for traffic networks. We model the traffic network as a network of queues [3]–[5] with link capacities. The result is a piecewise-affine (PWA) dynamical model, and we describe a method for obtaining a finite state abstraction of the dynamics using polyhedral operations as proposed in [9]. A related approach to control of freeway traffic is proposed in [14], but [14] only considers control of freeway networks through ramp metering while in this paper, we study signaling control of road networks.

Next, due to the expense of performing the required polyhedral operations in high dimensions, we propose an approximate finite state abstraction which can be computed much more efficiently. This efficiency is due to properties of the traffic flow model which allow efficient computation

of bounds on the one-step reachable states of a given link. We then suggest a method for extending these results to a probabilistic framework, which leads to numerous interesting directions for future work. Our prior work [15], currently in submission to a journal, does not present this extension or a method based on polyhedral operations.

This paper is organized as follows: Section II describes the traffic network dynamical model. Section III presents the LTL-based approach to control synthesis based on a computationally efficient method of constructing a finite state representation of the network dynamics. We extend these results to a probabilistic formulation in Section IV. A case study is presented in Section V, and we provide concluding remarks in Section VI.

II. SIGNALIZED NETWORK TRAFFIC MODEL

A signalized traffic network consists of a set \mathcal{L} of *links* and a set \mathcal{V} of *signalized intersections* or *nodes*. For $\ell \in \mathcal{L}$, let $\sigma(\ell) \in \mathcal{V}$ denote the head node of link ℓ and let $\tau(\ell) \in \mathcal{V} \cup \emptyset$ denote the tail node of link ℓ . A link ℓ with $\tau(\ell) = \emptyset$ serves as an entry-point into the network, and we assume $\sigma(\ell) \neq \tau(\ell)$ for all $\ell \in \mathcal{L}$ (*i.e.*, no self-loops). Link $k \neq \ell$ is *upstream* of link ℓ if $\sigma(k) = \tau(\ell)$, *downstream* of link ℓ if $\tau(k) = \sigma(\ell)$, and *adjacent to* link ℓ if $\tau(k) = \tau(\ell)$. Roads exiting the traffic network are not modeled explicitly. For each $v \in \mathcal{V}$, define

$$\mathcal{L}_v^{\text{in}} = \{\ell : \sigma(\ell) = v\} \quad (1)$$

$$\mathcal{L}_v^{\text{out}} = \{\ell : \tau(\ell) = v\}. \quad (2)$$

Each link $\ell \in \mathcal{L}$ possesses a queue $x_\ell[t] \in [0, x_\ell^{\text{cap}}]$ representing the number of vehicles on link ℓ at time step $t \in \mathbb{N} \triangleq \{0, 1, 2, \dots\}$ where x_ℓ^{cap} is the capacity of link ℓ . We allow x_ℓ to be a continuous quantity, thus adopting a fluid-like model of traffic flow. Let

$$\mathcal{X} = \prod_{\ell \in \mathcal{L}} [0, x_\ell^{\text{cap}}]. \quad (3)$$

Movement of vehicles among link queues is governed by mass-conservation laws and the state of the signalized intersections. When a link is *actuated*, a maximum of c_ℓ vehicles are allowed to flow from link ℓ to links $\mathcal{L}_{\sigma(\ell)}^{\text{out}}$ per time step where c_ℓ is the known *saturation flow* for link ℓ . To simplify notation, we assume each intersection $v \in \mathcal{V}$ has two possible states actuating either “East-West” (EW) incoming links or “North-South” (NS) incoming links¹.

¹We can easily generalize to signal variables with more than two states and general network topologies at the cost of more complex notation.

This research was supported in part by the NSF under grants CNS-1446145 and CNS-1446151.

Samuel Coogan and Murat Arcaç are with the Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, {scoogan, arcaç}@eecs.berkeley.edu

Ebru Aydin Gol is formerly with the Division of Systems Engineering, Boston University, ebruaydin@gmail.com

Calin Belta is with the Department of Mechanical Engineering, Boston University, cbelta@bu.edu

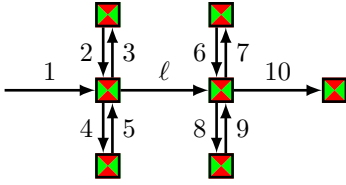


Fig. 1. A typical traffic network. In the figure, $\mathcal{L}_\ell^{\text{down}} = \{\ell, 7, 8, 10\}$, $\mathcal{L}_\ell^{\text{up}} = \{1, 2, 5\}$, and $\mathcal{L}_\ell^{\text{adj}} = \{3, 4\}$. At each time step, a signal is either actuating “East-West” (EW) links or “North-South” (NS) links where $\mathcal{L}^{\text{EW}} = \{1, \ell, 10\}$ and $\mathcal{L}^{\text{NS}} = \{2, 3, \dots, 9\}$. Note that if a link is actuated, it may send vehicles to any downstream link regardless of orientation.

To make this precise, we partition \mathcal{L} into EW links and NS links, denoted by \mathcal{L}^{EW} and \mathcal{L}^{NS} , respectively, so that $\mathcal{L} = \mathcal{L}^{\text{EW}} \cup \mathcal{L}^{\text{NS}}$. We define the signal variable $s_v \in \{0, 1\}$ as follows for each $v \in \mathcal{V}$:

$$s_v = \begin{cases} 1 & \text{if links } \mathcal{L}_v^{\text{in}} \cap \mathcal{L}^{\text{EW}} \text{ are actuated} \\ 0 & \text{if links } \mathcal{L}_v^{\text{in}} \cap \mathcal{L}^{\text{NS}} \text{ are actuated} \end{cases} \quad (4)$$

When a link ℓ is actuated, the *turn ratio* $\beta_{\ell k}$ denotes the fraction of vehicles exiting link ℓ that are routed to link k . It follows that $\beta_{\ell k} \neq 0$ only if $\sigma(\ell) = \tau(k)$ and

$$\sum_{k \in \mathcal{L}_{\sigma(\ell)}^{\text{out}}} \beta_{\ell k} \leq 1. \quad (5)$$

Strict inequality in (5) implies that a fraction of vehicles on link ℓ are routed off the network via unmodeled roads that exit the network. Even when a link is actuated, traffic flow can occur only if there is available capacity downstream. To this end, the supply ratio $\alpha_{\ell k}$ denotes the fraction of link k 's capacity available to link ℓ , that is, link ℓ may only send $\alpha_{\ell k}(x_k^{\text{cap}} - x_k[t])$ vehicles to link k in time period t . Since only incoming EW or NS links are actuated in each time step, it follows that, for all $k \in \mathcal{L}$,

$$\sum_{\ell \in \mathcal{L}_{\tau(k)}^{\text{in}} \cap \mathcal{L}^{\text{EW}}} \alpha_{\ell k} = \sum_{\ell \in \mathcal{L}_{\tau(k)}^{\text{in}} \cap \mathcal{L}^{\text{NS}}} \alpha_{\ell k} = 1. \quad (6)$$

We are now in a position to define the dynamics of the link queues. We first define the following:

$$\mathcal{L}_\ell^{\text{down}} = \mathcal{L}_{\sigma(\ell)}^{\text{out}} \cup \{\ell\} \quad \mathcal{L}_\ell^{\text{up}} = \mathcal{L}_{\tau(\ell)}^{\text{in}} \quad (7)$$

$$\mathcal{L}_\ell^{\text{adj}} = \mathcal{L}_{\tau(\ell)}^{\text{out}} \setminus \{\ell\} \quad \mathcal{L}_\ell^{\text{loc}} = \mathcal{L}_\ell^{\text{down}} \cup \mathcal{L}_\ell^{\text{up}} \cup \mathcal{L}_\ell^{\text{adj}} \quad (8)$$

so that $\mathcal{L}_\ell^{\text{down}}$ includes link ℓ and the links downstream of link ℓ , $\mathcal{L}_\ell^{\text{up}}$ and $\mathcal{L}_\ell^{\text{adj}}$ are links upstream and adjacent to ℓ , respectively, see Fig. 1. As we will see subsequently, the flow of vehicles out of link ℓ is only a function of the state of links in $\mathcal{L}_\ell^{\text{down}}$, and the update of link ℓ 's state is only a function of links in $\mathcal{L}_\ell^{\text{loc}}$, that is, links “local” to link ℓ .

Let $\mathbf{x}[t] = \{x_\ell[t]\}_{\ell \in \mathcal{L}}$, $\mathbf{s}[t] = \{s_v[t]\}_{v \in \mathcal{V}}$, $\mathbf{x}_\ell^{\text{down}} = \{x_k\}_{k \in \mathcal{L}_\ell^{\text{down}}}$, and $\mathbf{x}_\ell^{\text{loc}} = \{x_k\}_{k \in \mathcal{L}_\ell^{\text{loc}}}$. We define the outflow of link ℓ for all $\ell \in \mathcal{L}$ as follows:

$$f_\ell^{\text{out}}(\mathbf{x}_\ell^{\text{down}}, s_{\sigma(\ell)}) = \begin{cases} s_{\sigma(\ell)} \cdot \phi_\ell(\mathbf{x}_\ell^{\text{down}}[t]) & \text{if } \ell \in \mathcal{L}^{\text{EW}} \\ (1 - s_{\sigma(\ell)}) \cdot \phi_\ell(\mathbf{x}_\ell^{\text{down}}[t]) & \text{if } \ell \in \mathcal{L}^{\text{NS}} \end{cases} \quad (9)$$

where

$$\phi_\ell(\mathbf{x}_\ell^{\text{down}}[t]) = \min \left\{ x_\ell[t], c_\ell, \min_{\substack{k \text{ s.t.} \\ \beta_{\ell k} \neq 0}} \left\{ \frac{\alpha_{\ell k}}{\beta_{\ell k}} (x_k^{\text{cap}} - x_k[t]) \right\} \right\}. \quad (10)$$

The number of vehicles in each link's queue then evolves according to the mass conservation equation

$$\begin{aligned} x_\ell[t+1] &= F_\ell(\mathbf{x}_\ell^{\text{loc}}[t], \mathbf{s}^{\text{loc}}[t], d_\ell[t]) \\ &\triangleq x_\ell[t] - f_\ell^{\text{out}}(\mathbf{x}_\ell^{\text{down}}[t], s_{\sigma(\ell)}) \\ &\quad + \sum_{j \in \mathcal{L}_\ell^{\text{up}}} \beta_{j\ell} f_j^{\text{out}}(\mathbf{x}_j^{\text{down}}[t], s_{\sigma(j)}) + d_\ell[t] \end{aligned} \quad (11)$$

where $d_\ell[t]$ is the number of vehicles that exogenously enters the queue on link ℓ in time step t , $\mathbf{d} = \{d_\ell[t]\}_{\ell \in \mathcal{L}}$, and $\mathbf{s}_\ell^{\text{loc}} = \{s_{\sigma(\ell)}, s_{\tau(\ell)}\}$, that is, $\mathbf{s}_\ell^{\text{loc}}$ is the state of the signals that are “local” to link ℓ (if $\tau(\ell) = \emptyset$, then we take $\mathbf{s}_\ell^{\text{loc}} = \{s_{\sigma(\ell)}\}$). We assume there exists $\mathcal{D} \subset \mathbb{R}^{|\mathcal{L}|}$ such that

$$\mathbf{d}[t] \in \mathcal{D} \quad \forall t. \quad (13)$$

We let $F(\mathbf{x}, \mathbf{s}, \mathbf{d}) = \{F_\ell(\mathbf{x}^{\text{loc}}, \mathbf{s}^{\text{loc}}, d_\ell)\}_{\ell \in \mathcal{L}}$ so that

$$\mathbf{x}[t+1] = F(\mathbf{x}[t], \mathbf{s}[t], \mathbf{d}[t]). \quad (14)$$

We then have $F(\mathbf{x}, \mathbf{s}, \mathbf{d}) : \mathbb{R}^{|\mathcal{L}|} \times \{0, 1\}^{|\mathcal{V}|} \times \mathcal{D} \rightarrow \mathbb{R}^{|\mathcal{L}|}$. Finally, we define the set $\mathcal{L}_{v, \mathbf{s}}^{\text{in}}$ to be the set of links actuated by signal $\mathbf{s} \in \{0, 1\}^{|\mathcal{V}|}$ at intersection v , that is,

$$\mathcal{L}_{v, \mathbf{s}}^{\text{in}} = \begin{cases} \mathcal{L}_v^{\text{in}} \cap \mathcal{L}^{\text{EW}} & \text{if } s_v = 0 \\ \mathcal{L}_v^{\text{in}} \cap \mathcal{L}^{\text{NS}} & \text{if } s_v = 1. \end{cases} \quad (15)$$

III. CONTROLLER SYNTHESIS FROM LINEAR TEMPORAL LOGIC SPECIFICATIONS

We now turn to the main objective this paper, namely, synthesizing a signal control strategy such that the resulting traffic dynamics and signal sequence satisfies a control objective expressed using *linear temporal logic (LTL)*. We first define and motivate the need for the rich class of control objectives expressible in LTL in the context of traffic networks. We then propose a control synthesis approach which relies on a finite state representation of the traffic dynamics.

A. LTL Specifications for Traffic Networks

LTL *formulae* describe properties of trajectories of the traffic network and are generated inductively using the Boolean operators \vee (disjunction), \wedge (conjunction), \neg (negation), and the temporal operators \bigcirc (next) and \bigcup (until). Formally, such formulae are expressed over the states of the finite state representation constructed below. We use LTL formulae to describe desired behaviors of the traffic network, which allows very rich control specifications that include derived logical and temporal operators such as \rightarrow (implication), \square (always), and \diamond (eventually), see [6], [7].

Examples of LTL formulae representing desired control objectives relevant to traffic networks include those from the Introduction as well as:

- $\varphi = \diamond \square (x_\ell \leq C)$ for some C
“Eventually, link ℓ has less than C vehicles for all future time”
- $\varphi = \square \diamond (s_v = 0)$
“Always, it is eventually the case (i.e., infinitely often) that signal v actuates NS traffic”
- $\varphi = \square ((s_{v_1} = 1) \rightarrow \circ (s_{v_2} = 1))$
“Whenever signal v_1 actuates EW traffic, signal v_2 must actuate EW traffic in the next time step”
- $\varphi = \square ((x_{\ell_1} \geq C_1) \rightarrow (x_{\ell_2} \geq C_2))$
“The number of vehicles on link ℓ_1 is allowed to exceed C_1 only if the number of vehicles has exceeded C_2 on link ℓ_2 .”

LTL formulae constitute a large class of control objectives that encompass reachability, safety, sequentiality, and fairness conditions, along with many other derived composite conditions.

To generate control strategies for the traffic network that guarantee satisfaction of a LTL formula, we first construct a finite state representation, or *abstraction*, of the model defined in Section II. In the next section, we present an abstraction approach based on polyhedral operations, and we then show that a modified abstraction can be efficiently constructed by exploiting the sparsity and the dynamical properties of the traffic network.

B. Finite State Representation

We begin by defining a *quotient transition system* obtained via a rectangular partitioning of the state space that over-approximates the traffic dynamics. In particular, we consider partitioning each interval $[0, x_\ell^{\text{cap}}]$ into the set of intervals

$$\{[0, x_\ell^1], (x_\ell^1, x_\ell^2], \dots, (x_\ell^{N_\ell-2}, x_\ell^{N_\ell-1}], (x_\ell^{N_\ell-1}, x_\ell^{N_\ell}]\} \quad (16)$$

where $x_\ell^i < x_\ell^{i+1}$ for all i and $x_\ell^{N_\ell} = x_\ell^{\text{cap}}$. By convention, we let $x_\ell^0 = 0$. For $X_\ell \in \{1, \dots, N_\ell\}$, let

$$\llbracket X_\ell \rrbracket = \begin{cases} [x_\ell^{X_\ell-1}, x_\ell^{X_\ell}] & \text{if } X_\ell = 1 \\ (x_\ell^{X_\ell-1}, x_\ell^{X_\ell}] & \text{if } X_\ell > 1, \end{cases} \quad (17)$$

i.e., $\llbracket \cdot \rrbracket$ gives the interval corresponding to the discrete state X_ℓ . For $\mathbf{X} = \{X_\ell\}_{\ell \in \mathcal{L}}$, we let

$$\llbracket \mathbf{X} \rrbracket = \prod_{\ell \in \mathcal{L}} \llbracket X_\ell \rrbracket \subset \mathcal{X}. \quad (18)$$

Remark 1. We assume each link is partitioned into a collection of intervals, which results in a grid-based partitioning of \mathcal{X} . We could instead partition \mathcal{X} into a coarser partitioning of hyperrectangles that is not based on a gridding of the state space, resulting in a smaller finite state abstraction, but this approach is not pursued here and is left for future work.

Definition 1 (Quotient Transition System). *Given an interval partitioning as in (16) for each link $\ell \in \mathcal{L}$, a finite, non-deterministic quotient transition system of the traffic model is defined as the tuple $T = (\mathbb{X}, \mathbb{S}, \rightarrow)$ where*

- $\mathbb{X} = \prod_{\ell \in \mathcal{L}} \{1, \dots, N_\ell\}$ is the set of states,
- $\mathbb{S} = \{0, 1\}^{|\mathcal{V}|}$ is the set of controls,

- $\rightarrow \subseteq \mathbb{X} \times \mathbb{S} \times \mathbb{X}$ is the set of transitions given by $(\mathbf{X}, \mathbf{s}, \mathbf{X}') \in \rightarrow$ if and only if

$$\exists \mathbf{d} \in \mathcal{D}, \exists \mathbf{x} \in \llbracket \mathbf{X} \rrbracket, \exists \mathbf{x}' \in \llbracket \mathbf{X}' \rrbracket \text{ such that} \quad (19)$$

$$\mathbf{x}' = F(\mathbf{x}, \mathbf{s}, \mathbf{d}). \quad (20)$$

In words, a discrete state \mathbf{X} transitions to \mathbf{X}' under signal input $\mathbf{s} \in \mathbb{S}$ if and only if for some disturbance $\mathbf{d} \in \mathcal{D}$, there exists continuous states $\mathbf{x} \in \llbracket \mathbf{X} \rrbracket$ and $\mathbf{x}' \in \llbracket \mathbf{X}' \rrbracket$ such that \mathbf{x} may transition to \mathbf{x}' under the signaling input \mathbf{s} .

Remark 2. It is important to note that the transition system T defined above *simulates* [7] the original traffic network model, that is, all possible trajectories of the traffic network dynamics are represented in T . Thus T is an *over-approximation* of the traffic network model; due to this approximation, there may exist *spurious* executions of the quotient system that do not correspond to any trajectory of the original traffic network. Furthermore, T is non-deterministic due to these spurious trajectories and due to the disturbance taking values within a set. While the over-approximation implies possible conservatism in our approach [7], it does not affect soundness of the synthesis algorithm. In particular, we synthesize a controller that guarantees satisfaction of the control objective for all executions of the quotient system, which encompasses all possible trajectories of the original system.

We now discuss a method for obtaining the quotient transition system $T = (\mathbb{X}, \mathbb{S}, \rightarrow)$ from the dynamics presented in Section III-B. We observe that (9)–(13) result in dynamics that are *piecewise affine*, that is, there exists a set of polytopes $\{\mathcal{X}_p\}_{p \in \mathcal{P}}$ for some index set \mathcal{P} such that $\cup_{p \in \mathcal{P}} \mathcal{X}_p = \mathcal{X}$ and $\text{int}(\mathcal{X}_p) \cap \text{int}(\mathcal{X}_q) = \emptyset$ for all $p \neq q$, and such that for each $p \in \mathcal{P}$, we have

$$F(\mathbf{x}, \mathbf{s}, \mathbf{d}) = A_{p,\mathbf{s}}\mathbf{x} + b_{p,\mathbf{s}} + \mathbf{d} \quad \forall \mathbf{x} \in \mathcal{X}_p \quad (21)$$

for some $A_{p,\mathbf{s}} \in \mathbb{R}^{|\mathcal{L}| \times |\mathcal{L}|}$, $b_{p,\mathbf{s}} \in \mathbb{R}^{|\mathcal{L}|}$. In other words, the traffic dynamics are affine in each polyhedral partition. The polytopes arise from the $\min\{\cdot\}$ functions in (10).

For $\mathbf{X} \in \mathbb{X}$, let $\{\mathcal{X}_p^{\mathbf{X}}\}_{p \in \mathcal{P}}$ be the partitioning of $\llbracket \mathbf{X} \rrbracket$ with respect to \mathcal{P} , that is, $\mathcal{X}_p^{\mathbf{X}} = \llbracket \mathbf{X} \rrbracket \cap \mathcal{X}_p$ (in general, many $\mathcal{X}_p^{\mathbf{X}}$ will be empty).

The set of states of system (14) that are reachable from a set $Y \subset \mathcal{X}$ under the control signal \mathbf{s} is denoted by the Post operator and given by

$$\text{Post}(Y, \mathbf{s}) = \{\mathbf{x}' = F(\mathbf{x}, \mathbf{s}, \mathbf{d}) \mid \mathbf{x} \in Y, \mathbf{d} \in \mathcal{D}\}. \quad (22)$$

It follows that for $\mathbf{X} \in \mathbb{X}$,

$$\text{Post}(\llbracket \mathbf{X} \rrbracket, \mathbf{s}) = \bigcup_{p \in \mathcal{P}} \text{Post}(\mathcal{X}_p^{\mathbf{X}}, \mathbf{s}). \quad (23)$$

If \mathcal{D} is assumed to be a polytope, then $\text{Post}(\mathcal{X}_p^{\mathbf{X}}, \mathbf{s})$ is computed through basic polyhedral operations since each $\mathcal{X}_p^{\mathbf{X}}$ is a polytope and the dynamics are affine in $\mathcal{X}_p^{\mathbf{X}}$ under the control signal \mathbf{s} as in (21), see [9] for details.

Let $\text{Post}_T(\mathbf{X}, \mathbf{s}) \subset \mathbb{X}$ be the set of discrete states that the quotient transition system may transition to under signal \mathbf{s}

when in discrete state \mathbf{X} , that is

$$\text{Post}_T(\mathbf{X}, \mathbf{s}) = \{\mathbf{X}' \mid (\mathbf{X}, \mathbf{s}, \mathbf{X}') \in \rightarrow\}. \quad (24)$$

According to Definition 1, we have

$$\text{Post}_T(\mathbf{X}, \mathbf{s}) = \{\mathbf{X}' \mid \text{Post}(\llbracket \mathbf{X} \rrbracket, \mathbf{s}) \cap \llbracket \mathbf{X}' \rrbracket \neq \emptyset\}. \quad (25)$$

Therefore, the quotient transition system can be constructed by performing a set of polyhedral operations. However, these operations scale exponentially in the number of links. Next, we propose constructing an approximation of T that does not require polyhedral operations.

C. Approximate Quotient Transition System

We now introduce an approximate quotient system which can be constructed much more efficiently than the system proposed in Definition 1.

Definition 2 (Approximate Quotient Transition System). *Given an interval partitioning as in (16) for each $\ell \in \mathcal{L}$, an approximate finite quotient system of the traffic dynamics is defined as the tuple $T' = (\mathbb{X}, \mathbb{S}, \rightarrow')$ where*

- $\mathbb{X} = \prod_{\ell \in \mathcal{L}} \{1, \dots, N_\ell\}$ is the set of states,
- $\mathbb{S} = \{0, 1\}^{|\mathcal{V}|}$ is the set of controls,
- $\rightarrow' \subseteq \mathbb{X} \times \mathbb{S} \times \mathbb{X}$ is the set of transitions given by $(\mathbf{X}, \mathbf{s}, \mathbf{X}') \in \rightarrow'$ if and only if

$$\exists \mathbf{d} \in \mathcal{D} \quad \text{such that} \quad (26)$$

$$\forall \ell \in \mathcal{L}, \quad \exists \mathbf{x} \in \llbracket \mathbf{X} \rrbracket, \exists x'_\ell \in \llbracket X'_\ell \rrbracket \quad \text{such that} \quad (27)$$

$$x'_\ell = F_\ell(\mathbf{x}, \mathbf{s}, d_\ell). \quad (28)$$

In words, a discrete state \mathbf{X} transitions to $\mathbf{X}' = \{X'_\ell\}_{\ell \in \mathcal{L}}$ under signal input $\mathbf{s} \in \mathbb{S}$ if and only if there exists $\mathbf{d} \in \mathcal{D}$ and, for each link $\ell \in \mathcal{L}$, there exists $x'_\ell \in \llbracket X'_\ell \rrbracket$ and $\mathbf{x} \in \llbracket \mathbf{X} \rrbracket$ such that $x'_\ell = F_\ell(\mathbf{x}^{loc}, \mathbf{s}^{loc}, d_\ell)$.

The difference between Definitions 1 and 2 is that in verifying (28), a different $\mathbf{x} \in \llbracket \mathbf{X} \rrbracket$ may be chosen for each ℓ , whereas (20) must hold for a particular $\mathbf{x} \in \llbracket \mathbf{X} \rrbracket$. This subtle difference between Definitions 1 and 2 allows us to exploit the structure and sparsity of traffic network dynamics to efficiently compute T' . While T' may, in general, introduce additional conservatism, we remark that in the examples below, a satisfying controller is found which is valid for every initial condition, thus the approximate quotient transition system introduces no conservatism in the final synthesized controller.

Remark 3. The transition system T' is again an over-approximation of the transition system T and thus a controller synthesized for T' is guaranteed to also be correct for T and the original traffic network.

D. Efficient Computation of Approximate Quotient System

We now present an algorithm for efficiently computing T' when \mathcal{D} is a union of hyperrectangles where computation of $\{\mathbf{X}' \mid (\mathbf{X}, \mathbf{s}, \mathbf{X}') \in \rightarrow'\}$ only requires evaluating F_ℓ at two corners of the hyperrectangle $\llbracket \mathbf{X} \rrbracket$ for each ℓ .

We first assume $\mathcal{D} = \bigcup_{i=1}^{n_{\mathcal{D}}} \mathcal{D}_i$ for some $n_{\mathcal{D}} \in \mathbb{N}$ where

$$\mathcal{D}_i = \prod_{\ell \in \mathcal{L}} [d_i^\ell, \bar{d}_i^\ell] \quad \forall i \in \{1, \dots, n_{\mathcal{D}}\} \quad (29)$$

for $d_i^\ell \leq \bar{d}_i^\ell$ for all $\ell \in \mathcal{L}$. Let $\text{Post}_{T'}(\mathbf{X}, \mathbf{s}) = \{\mathbf{X}' \mid (\mathbf{X}, \mathbf{s}, \mathbf{X}') \in \rightarrow'\}$ and let

$$\text{Post}_{T', \mathcal{Y}}(\mathbf{X}, \mathbf{s}) = \{\mathbf{X}' \mid (\mathbf{X}, \mathbf{s}, \mathbf{X}') \in \rightarrow' \text{ when } \mathcal{D} \text{ is replaced with } \mathcal{Y} \text{ in (26)}\}. \quad (30)$$

Trivially,

$$\text{Post}_{T'}(\mathbf{X}, \mathbf{s}) = \text{Post}_{T', \mathcal{D}}(\mathbf{X}, \mathbf{s}) = \bigcup_{i=1}^{n_{\mathcal{D}}} \text{Post}_{T', \mathcal{D}_i}(\mathbf{X}, \mathbf{s}), \quad (31)$$

and we thus focus on computation of $\text{Post}_{T', \mathcal{D}_i}(\mathbf{X}, \mathbf{s})$. For given $\mathbf{X} \in \mathbb{X}$, $\mathbf{s} \in \mathbb{S}$, and \mathcal{D}_i , let

$$\underline{x}'_\ell = \min_{\mathbf{x} \in \llbracket \mathbf{X} \rrbracket, d_\ell \in [d_i^\ell, \bar{d}_i^\ell]} F_\ell(\mathbf{x}_\ell^{loc}, \mathbf{s}_\ell^{loc}, d_\ell) \quad (32)$$

$$\bar{x}'_\ell = \max_{\mathbf{x} \in \llbracket \mathbf{X} \rrbracket, d_\ell \in [d_i^\ell, \bar{d}_i^\ell]} F_\ell(\mathbf{x}_\ell^{loc}, \mathbf{s}_\ell^{loc}, d_\ell). \quad (33)$$

Proposition 1. *Given $\mathbf{X} \in \mathbb{X}$, $\mathbf{s} \in \mathbb{S}$, \mathcal{D}_i as in (29), and $\{\underline{x}'_\ell, \bar{x}'_\ell\}_{\ell \in \mathcal{L}}$ as defined in (32)–(33), we have*

$$\text{Post}_{T', \mathcal{D}_i}(\mathbf{X}, \mathbf{s}) = \{\mathbf{X}' = \{X'_\ell\}_{\ell \in \mathcal{L}} \mid \llbracket X'_\ell \rrbracket \cap [\underline{x}'_\ell, \bar{x}'_\ell] \neq \emptyset \quad \forall \ell \in \mathcal{L}\}. \quad (34)$$

Proposition 1 follows straightforwardly from the rectangular form of \mathcal{D}_i and the definition of \rightarrow' in Definition 1.

The computational advantage of Definition 2 over Definition 1 comes from the fact that the right hand sides of (32) and (33) can be computed efficiently. We first make the following technical assumption which is not particularly restrictive and can always be satisfied in traffic networks when a short enough time step is considered:

Assumption 1. *For all $\ell \in \mathcal{L}$,*

$$c_\ell \leq x_\ell^{cap} - \frac{\beta_{k\ell}}{\alpha_{k\ell}} c_j \quad \forall k \in \mathcal{L}_\ell^{up}. \quad (35)$$

Assumption 1 is a sufficient condition for ensuring that a link cannot completely clear its queue in one time step while simultaneously restricting flow from an upstream link, which is required for the following proposition.

Proposition 2. *Given $\mathbf{X} = \{X_\ell\}_{\ell \in \mathcal{L}} \in \mathbb{X}$ where $X_\ell = [x_\ell^{X_\ell-1}, x_\ell^{X_\ell}]$ or $X_\ell = (x_\ell^{X_\ell-1}, x_\ell^{X_\ell}]$ as in (17). For given $\mathbf{s} \in \mathbb{S}$ and $\mathbf{d} \in \mathcal{D}_i$, let*

$$\underline{\mathbf{x}}_\ell^{loc} = \bigcup_{\substack{k \in \\ \mathcal{L}_\ell^{down} \cup \mathcal{L}_\ell^{up}}} \{x_k^{X_k-1}\} \cup \bigcup_{k \in \mathcal{L}_\ell^{adj}} \{x_k^{X_k}\} \quad (36)$$

$$\bar{\mathbf{x}}_\ell^{loc} = \bigcup_{\substack{k \in \\ \mathcal{L}_\ell^{down} \cup \mathcal{L}_\ell^{up}}} \{x_k^{X_k}\} \cup \bigcup_{k \in \mathcal{L}_\ell^{adj}} \{x_k^{X_k-1}\}. \quad (37)$$

Then

$$\min_{\mathbf{x} \in \llbracket \mathbf{X} \rrbracket, d_\ell \in [d_i^\ell, \bar{d}_i^\ell]} F_\ell(\mathbf{x}_\ell^{loc}, \mathbf{s}_\ell^{loc}, d_\ell) = F_\ell(\underline{\mathbf{x}}_\ell^{loc}, \mathbf{s}_\ell^{loc}, d_i^\ell) \quad (38)$$

$$\max_{\mathbf{x} \in \llbracket \mathbf{X} \rrbracket, d_\ell \in [d_i^\ell, \bar{d}_i^\ell]} F_\ell(\mathbf{x}_\ell^{loc}, \mathbf{s}_\ell^{loc}, d_\ell) = F_\ell(\bar{\mathbf{x}}_\ell^{loc}, \mathbf{s}_\ell^{loc}, \bar{d}_i^\ell). \quad (39)$$

We provide the following proof sketch: Observe that $\underline{\mathbf{x}}_\ell^{\text{loc}}$ is the collection of lower bounds for link ℓ and the links that are upstream and downstream of link ℓ , and the upper bounds for the links adjacent to link ℓ . Likewise, $\bar{\mathbf{x}}_\ell^{\text{loc}}$ is the collection of upper bounds for link ℓ and the links that are upstream and downstream of link ℓ , and the lower bounds for the links adjacent to link ℓ . Furthermore, the structure of traffic dynamics renders $F_\ell(\underline{\mathbf{x}}_\ell^{\text{loc}}, \mathbf{s}^{\text{loc}}, d_\ell)$ monotonically increasing in x_k for $k \in (\mathcal{L}_\ell^{\text{down}} \cup \mathcal{L}_\ell^{\text{up}}) \setminus \{\ell\}$ and d_ℓ , and monotonically decreasing in x_k for $k \in \mathcal{L}_\ell^{\text{adj}}$. Thus $\underline{\mathbf{x}}_\ell^{\text{loc}}, \underline{d}_\ell^{\ell}$ are the conditions ensuring that x'_ℓ achieves the minimum possible under the constraint $\mathbf{x} \in \llbracket \mathbf{X} \rrbracket$ and $\mathbf{d} \in \mathcal{D}_i$, and similarly $\bar{\mathbf{x}}_\ell^{\text{loc}}, \bar{d}_i^{\ell}$ are the conditions ensuring that x'_ℓ achieves the maximum possible under the same constraints.

Combining Propositions 1 and 2, we have the following:

Corollary 1. *Given $\mathbf{X} \in \mathbb{X}$, $\mathbf{s} \in \mathbb{S}$, \mathcal{D}_i of the form (29), and $\underline{\mathbf{x}}_\ell^{\text{loc}}, \bar{\mathbf{x}}_\ell^{\text{loc}}$ for each $\ell \in \mathcal{L}$ given by (38)–(39), we have*

$$\text{Post}_{T', \mathcal{D}_i}(\mathbf{X}, \mathbf{s}) = \{\mathbf{X}' = \{X'_\ell\}_{\ell \in \mathcal{L}} \mid \llbracket X'_\ell \rrbracket \cap [\underline{x}'_\ell, \bar{x}'_\ell] \neq \emptyset \forall \ell \in \mathcal{L}\} \quad (40)$$

where

$$\underline{x}'_\ell = F_\ell(\underline{\mathbf{x}}_\ell^{\text{loc}}, \mathbf{s}_\ell^{\text{loc}}, \underline{d}_i^{\ell}) \quad (41)$$

$$\bar{x}'_\ell = F_\ell(\bar{\mathbf{x}}_\ell^{\text{loc}}, \mathbf{s}_\ell^{\text{loc}}, \bar{d}_i^{\ell}). \quad (42)$$

Corollary 1 exploits the structure and sparsity of the traffic dynamics and provides the key for efficient computation of the approximate quotient system T' . Indeed, sparsity allows consideration of only links and signals that are “local” to a link, and the structure of the dynamics is the foundation for Proposition 2 and, in turn, efficient computation of \underline{x}'_ℓ and \bar{x}'_ℓ via (41)–(42). Note that (41)–(42) requires computing F_ℓ at two points for each $\ell \in \mathcal{L}$, and that the two points $\underline{\mathbf{x}}_\ell^{\text{loc}}$ and $\bar{\mathbf{x}}_\ell^{\text{loc}}$ are easily obtained from $\mathbf{X} \in \mathbb{X}$ using (36)–(37). For disturbance set \mathcal{D} of the form (29), we compute $\text{Post}_{T', \mathcal{D}_i}$ for each \mathcal{D}_i and combine the resulting transitions via (31).

It follows that computing $\{\mathbf{X}' \mid (\mathbf{X}, \mathbf{s}, \mathbf{X}') \in \rightarrow'\}$ for some \mathbf{X} and \mathbf{s} scales linearly with the dimension of the continuous state space, *i.e.*, the number of links in the network. If we instead compute this set with polyhedral operations, the calculation scales exponentially with the number of links.

E. Augmenting the State Space with Signal History

To capture control objectives that include the state of the signals themselves (which are modeled as inputs in the approximate quotient transition system T'), we augment the discrete state space. Examples of specifications that require this augmentation include conditions such as “the state of an intersection cannot change more than once per n^{min} time steps” or “an input signal cannot remain unchanged for n^{max} time steps.” We propose the following transition system which includes the current state of each signal as well as the number of steps the signal has remained unchanged:

Definition 3 (Signal History Transition System). *The signalized intersection history is modeled as a transition system $T_{\text{sig}} = (\mathbb{Q}_{\text{sig}}, \mathbb{S}, \rightarrow_{\text{sig}})$ where*

- $\mathbb{Q}_{\text{sig}} = \{0, 1\}^{|\mathcal{V}|} \times \{0, \dots, N\}^{|\mathcal{V}|}$ is a set of states,
- $\mathbb{S} = \{0, 1\}^{|\mathcal{V}|}$ is the set of controls,
- $\rightarrow_{\text{sig}} \subset \mathbb{Q}_{\text{sig}} \times \mathbb{S} \times \mathbb{Q}_{\text{sig}}$ is the set of transitions given by $((\xi, \mathbf{n}), \mathbf{s}, (\xi', \mathbf{n}')) \in \rightarrow_{\text{sig}}$ with $\xi, \xi' \in \{0, 1\}^{|\mathcal{V}|}$, $\mathbf{n}, \mathbf{n}' \in \{0, \dots, N\}^{|\mathcal{V}|}$, and $\mathbf{s} \in \mathbb{S}$ if and only if

$$\xi' = \mathbf{s} \text{ and } n'_v = \begin{cases} n_v + 1 & \text{if } \xi_v = s_v \text{ and } n_v < N \\ N & \text{if } \xi_v = s_v \text{ and } n_v = N \\ 0 & \text{if } \xi_v \neq s_v \end{cases} \quad \forall v \in \mathcal{V}. \quad (43)$$

In other words, for a state $(\xi, \mathbf{n}) \in \mathbb{Q}_{\text{sig}}$, ξ is the latest control signal applied to the intersections of the traffic network and n_v is the number of time steps for which the state intersection v has not changed. The counter n_v saturates at N time steps to ensure that T_{sig} remains finite. From Definitions 2 and 3, we construct the *synchronous product* of the approximate quotient system and the signal history transition system:

Definition 4 (Product Transition System). *The final product transition system that models the dynamical behavior of the traffic network is a transition system $T_{\text{prod}} = (\mathbb{Q}_{\text{prod}}, \mathbb{S}, \rightarrow_{\text{prod}})$ where*

- $\mathbb{Q}_{\text{prod}} = \mathbb{X} \times \mathbb{Q}_{\text{sig}}$ is a set of discrete states,
- $\mathbb{S} = \{0, 1\}^{|\mathcal{V}|}$ is the set of controls,
- $\rightarrow_{\text{prod}} \subset \mathbb{Q}_{\text{prod}} \times \mathbb{S} \times \mathbb{Q}_{\text{prod}}$ is the set of transitions given by $((\mathbf{X}, \xi, \mathbf{n}), \mathbf{s}, (\mathbf{X}', \xi', \mathbf{n}')) \in \rightarrow_{\text{prod}}$ if and only if $(\mathbf{X}, \mathbf{s}, \mathbf{X}') \in \rightarrow'$ and $((\xi, \mathbf{n}), \mathbf{s}, (\xi', \mathbf{n}')) \in \rightarrow_{\text{sig}}$.

F. Controller Synthesis

We omit the details of how a control strategy is synthesized from the nondeterministic transition system T_{prod} for a given LTL control objective as this is well-documented elsewhere in the literature, see *e.g.* [9], [16]. Instead, we summarize the main steps of this synthesis as follows: from the LTL control objective, we obtain a deterministic Rabin automaton that accepts all and only trajectories that satisfy the LTL specification using off-the-shelf software. We then construct the synchronous product of the Rabin automaton and T_{prod} , resulting in a nondeterministic Rabin automaton from which a control strategy is found by playing a Rabin game [16]. The result is a control strategy along with a set of initial conditions from which trajectories of the traffic network are guaranteed to satisfy the desired LTL specification.

As the discrete state space is finite, the control strategy takes the form of a collection of “lookup” tables over the discrete states of the system, \mathbb{Q}_{prod} , and there is one such table for each state in the Rabin automaton encoding the desired LTL property. Thus implementing the control strategy requires implementing the specification Rabin automaton, which is interpreted as “tracking” progress of the LTL specification and updates at each time step, and applying the control corresponding to the current discrete state $(\mathbf{X}, \xi, \mathbf{n}) \in \mathbb{Q}_{\text{prod}}$ of the system within the lookup table dictated by the current state of the Rabin automaton.

IV. EXTENSIONS TO A PROBABILISTIC MODEL

The above approach accommodates uncertainty in the disturbance input via nondeterminism in the finite state abstraction, and a controller is synthesized to satisfy the control objective for any disturbance input. However, in traffic networks, disturbance inputs are often characterized probabilistically. Furthermore, it is often sufficient for a controller to satisfy a specification with high probability rather than with certainty. For example, we may wish to find a controller that avoids congestion with 95% probability; the rare occurrence of a large exogenous input may prevent synthesis of a controller that avoids congestion with certainty.

To this end, we now extend the above methodology to a probabilistic framework. We model the traffic network as a Markov Decision Process (MDP), and seek to maximize the probability of satisfying a LTL specification.

A. Probabilistic Model of Freeway Traffic

We propose modifying the nondeterministic system developed in Section III to include transition probabilities obtained from a known probability distribution on the disturbance set \mathcal{D} . We create an MDP from the approximate quotient system T' ; applying this approach to a product transition system that includes signal history is a straightforward extension of Section III-E. From T' , we obtain an MDP M as follows:

Definition 5 (Traffic Model MDP). *From an approximate transition system and a probability distribution on \mathcal{D} , we obtain a Markov Decision Process (MDP) $M = (\mathbb{X}, \mathbb{S}, P)$ that models the traffic dynamics where:*

- $\mathbb{X} = \prod_{\ell \in \mathcal{L}} \{1, \dots, N_\ell\}$ is the set of states,
- $\mathbb{S} = \{0, 1\}^{|\mathcal{V}|}$ is the set of controls,
- $P : \mathbb{X} \times \mathbb{S} \times \mathbb{X} \rightarrow [0, 1]$ is the transition probability function satisfying for all $\mathbf{X} \in \mathbb{X}$ and $\mathbf{s} \in \mathbb{S}$, $\sum_{\mathbf{X}' \in \mathbb{X}} P(\mathbf{X}, \mathbf{s}, \mathbf{X}') = 1$ and $P(\mathbf{X}, \mathbf{s}, \mathbf{X}') > 0$ if and only if $(\mathbf{X}, \mathbf{s}, \mathbf{X}') \in \rightarrow'$. A formula for P is given subsequently.

We assume \mathcal{D} is a hyperrectangle, that is, $\mathcal{D} = \prod_{\ell \in \mathcal{L}} [\underline{d}^\ell, \bar{d}^\ell]$ for some $\underline{d}^\ell \leq \bar{d}^\ell$ for all $\ell \in \mathcal{L}$. Let $p^\mathcal{D} : \mathcal{D} \rightarrow [0, 1]$ be the probability distribution on \mathcal{D} so that $\int_{\mathcal{D}} p^\mathcal{D}(\delta) d\delta = 1$. We assume $p^\mathcal{D}$ is a product distribution, that is, $p^\mathcal{D}(d) = \prod_{\ell \in \mathcal{L}} p_\ell^\mathcal{D}(d_\ell)$ where $p_\ell^\mathcal{D} : [\underline{d}^\ell, \bar{d}^\ell] \rightarrow [0, 1]$ and $\int_{\underline{d}^\ell}^{\bar{d}^\ell} p_\ell^\mathcal{D}(\delta) d\delta = 1$ for all $\ell \in \mathcal{L}$ (note that $p_\ell^\mathcal{D}$ may be, e.g., a Dirac delta function). We suppose the exogenous disturbance is drawn independently from \mathcal{D} at each time step. Now define $p_\ell^{\mathbf{X}, \mathbf{s}} : [0, x_\ell^{\text{cap}}] \rightarrow [0, 1]$ as follows:

$$p_\ell^{\mathbf{X}, \mathbf{s}}(x'_\ell) = \frac{1}{\bar{y}_\ell - \underline{y}_\ell} \int_{\underline{y}_\ell}^{\bar{y}_\ell} p_\ell^\mathcal{D}(x'_\ell - z) dz \quad (44)$$

where

$$\underline{y}_\ell = F_\ell(\underline{\mathbf{x}}_\ell^{\text{loc}}, \mathbf{s}_\ell^{\text{loc}}, 0), \quad \bar{y}_\ell = F_\ell(\bar{\mathbf{x}}_\ell^{\text{loc}}, \mathbf{s}_\ell^{\text{loc}}, 0) \quad (45)$$

and $\underline{\mathbf{x}}_\ell^{\text{loc}}, \bar{\mathbf{x}}_\ell^{\text{loc}}$ are as given in (36)–(37). We substitute 0 for the disturbance in (45) as the disturbance is accommodated probabilistically in (44). Notice that $p^{\mathbf{X}, \mathbf{s}}(x_\ell)$ has support only on $[\underline{y}_\ell + \underline{d}_\ell, \bar{y}_\ell + \bar{d}_\ell]$ and integrates to one on this domain.

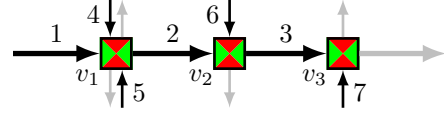


Fig. 2. Signaled network consisting of a major corridor road (links 1, 2, and 3) which intersects minor cross streets (links 4, 5, 6, and 7). The gray links are not explicitly modeled.

We interpret (44) as the probability distribution for the state of link ℓ in the next time step when initialized in discrete state \mathbf{X} . We define the joint probability distribution

$$p^{\mathbf{X}, \mathbf{s}}(x) = \prod_{\ell \in \mathcal{L}} p_\ell^{\mathbf{X}, \mathbf{s}}(x_\ell) \quad (46)$$

and are now in a position to define the probability transition function P :

$$P(\mathbf{X}, \mathbf{s}, \mathbf{X}') := \int_{\mathbf{X}'} p^{\mathbf{X}, \mathbf{s}}(x) dx. \quad (47)$$

We thus have a complete definition for an MDP model of the traffic network dynamics. For simple $p^\mathcal{D}(\cdot)$, P is straightforward to compute. To synthesize a control policy for the MDP that maximizes the probability of satisfying an LTL specification, we apply the learning based approach suggested in [17], see [7, Ch. 10], [13] for alternative approaches.

B. Discussion

The advantages of a probabilistic approach include more realistic assumptions on disturbance inputs, control synthesis that achieves objectives with high probability when they cannot be achieved with certainty, and a natural foundation for incorporating measured data and online control updates.

We emphasize that the results proposed in this section are preliminary. Unlike the nondeterministic case where a control strategy synthesized for the approximate transition system is guaranteed to apply to the original traffic network, the probability of satisfying a control objective computed from the MDP model may not completely reflect the actual satisfaction probability exhibited by the traffic network. This is due to the over-approximating nature of T' inherited by M and also due to the inherent Markov property of M . That is, the implicit assumption in Definition 5 is that transition probabilities from a given state are independent of the trajectory taken to this state. In the limit as the size of the interval partitions decreases to zero, we recover the discrete time stochastic process defining the traffic dynamics, thus the transition probabilities computed here are reasonable for a sufficiently refined partitioning. Furthermore, these preliminary results are promising and suggest an important future role for a probabilistic framework in traffic control synthesis. For example, we may update estimated transition probabilities based on observed traffic flow, allowing us to incorporate measured data.

V. EXAMPLE NETWORK

A. Nondeterministic Abstraction

We consider the example network in Fig. 2 which consists of a main corridor (links 1, 2, and 3) with intersecting cross streets (links 4, 5, 6, and 7) and three intersections. The gray links exit the network and are not explicitly modeled. Network parameters are given in Table I where the time step is 15 seconds. We divide the continuous state of each link into partitions of length 10. We assume that at each time step, the exogenous input \mathbf{d} belongs to one of the sets defined by:

- 0 to 20 vehicles joins link 1, or
- 0 to 10 vehicles joins links 4 and 5 each, or
- 0 to 10 vehicles joins link 6, or
- 0 to 10 vehicles joins link 7

where we assume that vehicles join a link only if the link is not at capacity. The above conditions establish the disturbance set \mathcal{D} . If a disturbance input would result in a link exceeding its capacity, we assume the link state is set to capacity. This can be interpreted as excess vehicles choosing not to enter the network, but we could alternatively explicitly prevent this condition with appropriate choice of control specification.

We wish to find a control policy for the three signalized intersections that satisfies the following linear temporal logic property:

$$\varphi_1 = \square \diamond (s_{v_1} = 0) \wedge \square \diamond (s_{v_1} = 1) \wedge \quad (48)$$

$$\square \diamond (s_{v_2} = 0) \wedge \square \diamond (s_{v_2} = 1) \wedge \quad (49)$$

$$\square \diamond (s_{v_3} = 0) \wedge \square \diamond (s_{v_3} = 1) \wedge \quad (50)$$

$$\diamond \square (x_2 \leq 30 \wedge x_3 \leq 30) \quad (51)$$

and the following additional restriction on the signaling:

$$s_v[t] \neq s_v[t+1] \text{ implies } s_v[t+1] = s_v[t+2] \\ \text{for } v \in \{v_1, v_2, v_3\}. \quad (52)$$

In words, (48)–(52) represent the following desired property: “(Always eventually each signal is red) and (always eventually each signal is green) and (eventually, links 2 and 3 have adequate supply for all future time) and (a signal’s state cannot change twice in two periods)”

Above, “adequate supply” for links 2 and 3 means the number of vehicles on links 2 and 3 does not exceed 30, that is, these links can always accept upstream demand. The control objective (48)–(51) is transformed into a Rabin automaton with 18 states. To accommodate (48)–(52), we augment the discrete state space \mathbb{X} with the signaling history up to two time steps as described in Section III-E, resulting in the final discrete state space \mathbb{Q}_{prod} .

Fig. 3(a) shows a sample trajectory of the network using a naive signaling strategy whereby each intersection actuates EW traffic for four time steps and then NS traffic for four times steps, and the signals are coordinated so that the actuated directions are synchronized. The disturbance input is chosen as the maximum from one of the four sets defined above to demonstrate the approach, and the particular set is chosen uniformly randomly. As the figure suggests, the

$$(x_1^{\text{cap}}, \dots, x_7^{\text{cap}}) = (30, 50, 50, 20, 20, 20, 20)$$

$$(c_1, \dots, c_7) = (10, 20, 20, 10, 10, 10, 10)$$

$$\beta_{12} = \beta_{23} = \beta_{42} = \beta_{52} = 0.5, \quad \beta_{63} = 1$$

$$\alpha_{42} = \alpha_{52} = 0.5, \quad \alpha_{12} = \alpha_{23} = \alpha_{63} = 1$$

TABLE I

NETWORK PARAMETERS FOR THE EXAMPLE NETWORK IN SECTION V.

trajectories are not guaranteed to satisfy the control objective, in particular, (51) is violated. Fig. 3(b) shows a sample trajectory of the system with a control strategy synthesized using the approximate quotient system. The control strategy is correct-by-construction and thus guaranteed to satisfy (48)–(52).

It is interesting to observe that the controller exhibits the so-called “green wave” phenomenon [2] whereby the green time for the three signals are offset to facilitate a vehicle traveling along the arterial without stopping. This control technique is well known to produce increased throughput, however the controller also reacts to undesirable disturbances or initial conditions, such as during the first five time periods or at time period 16 when the disturbance increases the number of vehicles on Link 2 to undesirable levels and the controller actuates EW traffic at junction 2 in response.

The final automaton contained 76,800 discrete states and required approximately 14 minutes to obtain a solution. A satisfying control policy was found from any initial condition, and thus, once a control strategy is synthesized, a controller can be implemented with negligible on-line computation costs.

B. Probabilistic Abstraction

We now consider the same network as above, but assume the disturbance is drawn from a uniform distribution over of the box

$$\mathcal{D} = \{\mathbf{d} \mid 0 \leq \mathbf{d} \leq [20 \ 0 \ 0 \ 10 \ 10 \ 10 \ 10]^T\}. \quad (53)$$

We obtain a probabilistic abstraction of the dynamics in the form of an MDP as described in Section 5. The MDP is not augmented with the signal history and contains 432 discrete states. We wish to maximize the probability of satisfying the following temporal logic specification:

$$\varphi_2 = \diamond \square (x_2 \leq 30 \wedge x_3 \leq 30) \wedge \quad (54)$$

$$\square \diamond (x_4 \leq 10 \wedge x_5 \leq 10 \wedge x_6 \leq 10 \wedge x_7 \leq 10) \wedge \quad (55)$$

$$\square ((x_1 > 30) \rightarrow \diamond x_1 \leq 10). \quad (56)$$

In words, (54)–(56) represent the following desired property: “(Eventually, links 2 and 3 have adequate supply for all future time) and (infinitely often, the queue on links 4, 5, 6, and 7 are short), and (whenever the queue on link 1 exceeds 30 vehicles, the queue eventually is short).”

Above, a “short” queue has 10 or fewer vehicles, and we do not require a signal to remain unchanged for two time periods as in the previous section. Using the learning-based approach proposed in [17], we obtain a control strategy that achieves φ_2 with probability one as verified using the PRISM

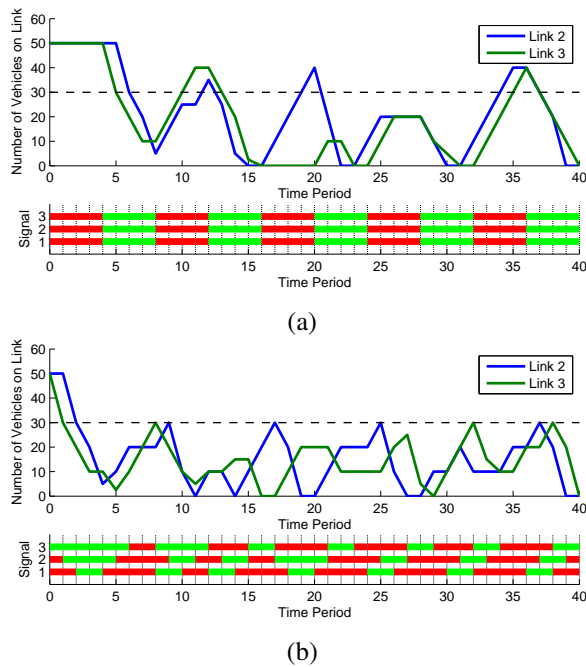


Fig. 3. (a) A sample trajectory of a naive strategy that actuates EW traffic and then NS traffic for four time steps each in a synchronized fashion. This policy does not satisfy the desired control objective φ_1 , in particular, (51) is not satisfied. (b) A sample trajectory resulting from the synthesized control policy that is guaranteed to satisfy the LTL policy (48)–(52). In the lower plots of (a) and (b), green (red) for the signal trace indicates EW (NS) traffic is actuated.

model-checker [18] (total computation time for synthesis was 23.0 seconds). On the other hand, if we model the system as a nondeterministic transition system as in Section III-C, we find that no controller exists satisfying φ_2 , and thus the probabilistic formulation is crucial.

We see in Fig. 4 that we indeed obtain a controller that satisfies φ_2 and behaves as expected. Signal 1 remains green for long periods of time so that the queue on link 1 can become short, as required by (56). The controller does exhibit frequent changes in control input since we did not require the signal to remain unchanged for a certain number of periods; if this is undesirable for the application, we could augment the state space with the signaling input and prevent frequent switching as done in Section V-A.

VI. CONCLUSIONS

We have proposed a framework for synthesizing a control strategy for a network of signalized intersections that ensures the resulting traffic dynamics satisfy a control objective expressed as a linear temporal logic formula. The large class of control objectives accommodated by LTL is well-suited for modern transportation infrastructure where there are many, sometimes competing, objectives. Furthermore, we exploit structural properties of the traffic network to drastically reduce the time required to compute a finite state abstraction of the dynamics. Our future research will address ways of further exploiting the structure inherent in such systems to reduce the number of discrete states in the abstraction.

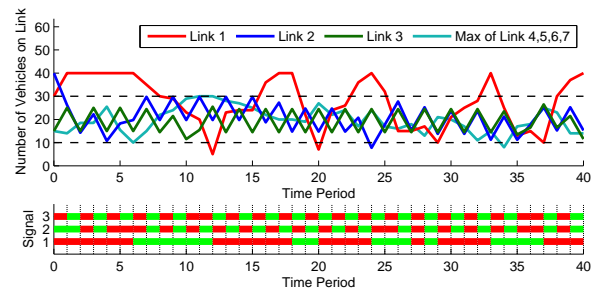


Fig. 4. A sample trajectory of the strategy obtained by modeling the traffic network as an MDP that satisfies φ_2 .

REFERENCES

- [1] A. A. Kurzhanskiy and P. Varaiya, “Active traffic management on road networks: A macroscopic approach,” *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 368, no. 1928, pp. 4607–4626, 2010.
- [2] M. Papageorgiou, C. Diakaki, V. Dinopoulou, A. Kotsialos, and Y. Wang, “Review of road traffic control strategies,” *Proceedings of the IEEE*, vol. 91, no. 12, pp. 2043–2067, 2003.
- [3] T. Wongpiromsarn, T. Uthacharoenpong, Y. Wang, E. Frazzoli, and D. Wang, “Distributed traffic signal control for maximum network throughput,” in *Intelligent Transportation Systems (ITSC), 2012 15th International IEEE Conference on*, pp. 588–595, Sept 2012.
- [4] P. Varaiya, “The max-pressure controller for arbitrary networks of signalized intersections,” in *Advances in Dynamic Network Modeling in Complex Transportation Systems*, pp. 27–66, Springer, 2013.
- [5] P. Varaiya, “Max pressure control of a network of signalized intersections,” *Transportation Research Part C: Emerging Technologies*, vol. 36, pp. 177–195, 2013.
- [6] E. M. Clarke, O. Grumberg, and D. A. Peled, *Model checking*. MIT press, 1999.
- [7] C. Baier and J. Katoen, *Principals of Model Checking*. The MIT Press, 2008.
- [8] M. Kloetzer and C. Belta, “A fully automated framework for control of linear systems from temporal logic specifications,” *IEEE Transactions on Automatic Control*, vol. 53, no. 1, pp. 287–297, 2008.
- [9] B. Yordanov, J. Tůmová, I. Černá, J. Barnat, and C. Belta, “Temporal logic control of discrete-time piecewise affine systems,” *IEEE Transactions on Automatic Control*, vol. 57, no. 6, pp. 1491–1504, 2012.
- [10] E. Aydin Gol, M. Lazar, and C. Belta, “Language-guided controller synthesis for discrete-time linear systems,” in *Proceedings of the 15th ACM international conference on Hybrid Systems: Computation and Control*, pp. 95–104, ACM, 2012.
- [11] P. Tabuada and G. Pappas, “Linear time logic control of discrete-time linear systems,” *IEEE Transactions on Automatic Control*, vol. 51, no. 12, pp. 1862–1877, 2006.
- [12] T. Wongpiromsarn, U. Topcu, and R. M. Murray, “Receding horizon control for temporal logic specifications,” in *Proceedings of the 13th ACM international conference on Hybrid systems: computation and control, HSCC ’10*, (New York, NY, USA), pp. 101–110, ACM, 2010.
- [13] X. C. Ding, S. L. Smith, C. Belta, and D. Rus, “Optimal control of Markov decision processes with linear temporal logic constraints,” *IEEE Transactions on Automatic Control*, 2014.
- [14] S. Coogan and M. Arcak, “Freeway traffic control from linear temporal logic specifications,” in *Proceedings of the 5th ACM/IEEE International Conference on Cyber-Physical Systems*, pp. 36–47, 2014.
- [15] S. Coogan, E. A. Gol, M. Arcak, and C. Belta, “Traffic network control from temporal logic specifications,” 2014. Submitted.
- [16] F. Horn, “Streett games on finite graphs,” *Proc. 2nd Workshop Games in Design Verification (GDV)*, 2005.
- [17] D. Sadigh, E. S. Kim, S. Coogan, S. S. Sastry, and S. A. Seshia, “A learning based approach to control synthesis of Markov decision processes for linear temporal logic specifications,” in *IEEE Conference on Decision and Control*, 2014. To appear.
- [18] M. Kwiatkowska, G. Norman, and D. Parker, “PRISM 4.0: Verification of probabilistic real-time systems,” in *Proc. 23rd International Conference on Computer Aided Verification (CAV’11)* (G. Gopalakrishnan and S. Qadeer, eds.), vol. 6806 of LNCS, pp. 585–591, Springer, 2011.