# A Coordinated Spillback-Aware Traffic Optimization and Recovery at Multiple Intersections

Pratham Oza*, Thidapat Chantem*, Pamela Murray-Tuite†
*Department of Electrical and Computer Engineering, Virginia Tech
†Department of Civil Engineering, Clemson University
{prathamo, tchantem}@vt.edu, pmmurra@clemson.edu

***Abstract*—Efficient traffic control remains a challenging task, especially during and after special events such as emergency vehicle traversals or blocked links due to disabled vehicles. While existing approaches aim to reduce travel delays, they do not consider recovery from spillbacks caused by such interruptions in the traffic network. This paper (1) presents an optimal algorithm that maximizes the traffic flow through the road network while ensuring that spillbacks do not occur during normal operations, (2) proposes an effective, predictable mitigation strategy to recover from spillbacks caused by special events and which may have propagated through multiple links and/or intersections in the network, and (3) provides worst-case wait time bounds as well as recovery time bounds associated with the proposed techniques. Compared to existing approaches, our optimal strategy shows a 53.2% improvement in worst-case travel times. Additionally, our mitigation strategy can recover from spillbacks that have propagated through multiple links in the network by up to 50.9% quicker than the existing approaches.**

## I. Introduction

Rapid urbanization due to economic and social growth has led to a sharp increase in vehicular traffic. Congestion levels have increased in 239 countries across the globe since 2018 with most urban cities showing a 46–71% increase in traffic congestion [1]. The U.S. alone lost $88 billion in 2019 due to the time spent during commutes [2], which is just one of the consequences of congestion alongside fuel expenses and increased pollution [3]. With limited land resources, efficient urban traffic management is paramount.

Congested traffic networks can cause a multitude of repercussions such as (1) an increase in collisions due to stop-and-go traffic [4], and (2) spillbacks that could propagate through the road network leading to unpredictable travel delays [5]. Special events that could block the intersection(s) temporarily such as roadblocks, collisions, and emergency response vehicles (ERVs) traveling through the traffic network could further increase spillbacks and delays, bringing traffic to a halt, especially in congested urban areas [6].

Traffic controllers are often enabled with ERV preemption systems that block the intersection access to non-emergency vehicles facilitating rapid and safe traversal of ERVs [7]. Once the ERV has safely crossed, however, the traffic in the network likely has long queues due to the prolonged stoppage of the vehicles, especially on busy arterials. In closely-spaced urban intersections, such extended blockage could lead to spillbacks that could spread throughout the network [8].

To manage the congested roadways and intersections, adaptive traffic control systems [9]–[11] have been deployed that optimize the traffic flow by adjusting the pre-timed signal policies. Due to their strong reliance on the estimated traffic flow and road network design, such systems cannot easily be tuned online. Urban roads with tightly-spaced intersections do not necessarily follow the specified traffic flow models and are often prone to spillbacks [12]. Recently, many learning-based approaches [13]–[15] have been proposed where the models are trained with various traffic scenarios and deployed online. However, such models require training data with varying traffic patterns along with heavy computational resources that are unavailable in current traffic controller infrastructure [16].

Decentralized traffic management techniques organize traffic locally at each intersection by fine-tuning the traffic light timings and cycle lengths [17], [18]. Specifically, a real-time server-based approach was presented for an isolated intersection [19] that provides a heuristic strategy to avoid spillbacks and reduce wait times by leveraging traffic information from neighboring intersections. However, such an approach does not consider coordination among the intersections, leading to congestion and eventual spillback at intersections downstream (Section VI). In addition, most existing techniques [17], [19] only focus on managing the traffic under normal scenarios and do not have a strategy in place to address temporary overloads in the network caused by events such as ERV preemption.

To fill the gap in research, we present an adaptive traffic control strategy that: (1) optimally maximizes traffic flow (i.e., reduces travel times) through a road network with multiple intersections while avoiding spillbacks in *normal traffic states*, (2) mitigates spillbacks in *extreme traffic states*, which are caused by disruptions in the road network during a normal traffic state, and (3) provides the worst-case wait time guarantees for traversing the network under normal as well as extreme traffic states, which would help in estimating travel time delays and recovery time required to move from an extreme to a normal traffic state. Here, a *normal traffic state* represents a state with no existing spillbacks in the network and where

This work was supported in part by the National Science Foundation under grant numbers CPS-1618979 and CPS-1812524.

978-1-7281-4403-0/20/$31.00 ©2020 IEEE

traffic can be effectively managed. In contrast, an *extreme traffic state* indicates a state in which there exist spillback(s) that could quickly cause further disruptions and which need to be mitigated as quickly as possible. Our contributions are:

1) We leverage the real-time server-based approach for an isolated intersection [19] to design an adaptive coordinated traffic control for a road network with multiple intersections.
2) During normal traffic states, we provide an optimal server-based approach to minimize the wait times for vehicles traveling through the network while avoiding spillbacks and ensuring safe intersection crossings.
3) During extreme traffic states, we provide a heuristic server-based mitigation strategy to facilitate timely recovery from spillbacks caused by unexpected events.
4) We analytically derive the worst-case wait times for the vehicles in the network, as well as the worst-case recovery time to a normal traffic state.
5) We analyze the adaptability of our approaches through simulations with different traffic patterns and further validate them using a hardware-in-loop (HIL) testbed with robots representing vehicles and emulating human driving response in a typical urban traffic environment.

## II. System Model

### A. Road Network and Traffic Infrastructure

We consider a road network of $m \times n$ intersections with $m$ arterials traveling along the east-west direction and $n$ arterials traveling along the north-south direction forming $mn$ intersections. This network can be a standalone road network or a part of a larger urban area. Each arterial within the network may consist of multiple links for each direction of traffic flow. Each intersection is controlled by traffic lights corresponding to the four traffic flow directions. Figure 1 shows a $3 \times 3$ road network, therefore consisting of nine intersections. Each intersection is formed by an incoming link per the direction of flow. The notations used in this paper for identifying the links, arterials, and intersections are as follows:

- An arbitrary arterial in the $m \times n$ network traveling towards the direction $D$ is denoted by $A^D_\alpha$, which are formed by links $L^{D*}_\alpha$ connecting each intersection with the arterial. Here, $*$ denotes all lanes within the link traveling towards direction $D \in \{N, E, W, S\}$ and is dropped when the context is clear. $N, E, W$ and $S$ stand for north, east, west and south, respectively and $\alpha$ is a dummy variable such that $\alpha = ij,\ i \in [1, m],\ j \in [1, n]$.
- The $r^{th}$ arterial along the direction $D$ is therefore denoted by $A^D_{\alpha_r}$ where,

$$\alpha_r = \begin{cases} ir, \text{where } i \in [1, m], 1 \le r \le n, \text{ if } D = \{S, N\} \\ ri, \text{where } 1 \le r \le m, i \in [1, n], \text{ if } D = \{E, W\} \end{cases}$$

- Similarly, multiple arterials within the network, i.e., $A^D_{\alpha_1}, A^D_{\alpha_2} \dots A^D_{\alpha_r}$ are denoted as $A^D_{\alpha_i}, \forall i \in [1, r]$.
- An arterial is formed by links connecting multiple intersections within the arterial and can be represented as a set of multiple links within the arterial. Thus, an arterial $A^D_{\alpha_r}$ consists of $\{L^D_i\}, \forall i \in [1, \lambda]$ where, $\lambda = m$, if $D = \{S, N\}$ and $\lambda = n$ otherwise. Therefore, the $r'^{th}$ link within the arterial $A^D_{\alpha_r}$ is represented by $L^D_{r'}$.
- The intersection formed by $A^E_{\alpha_r}$, $A^W_{\alpha_r}$, $A^S_{\alpha_{r'}}$ and $A^N_{\alpha_{r'}}$ is denoted by $I_{rr'}$.

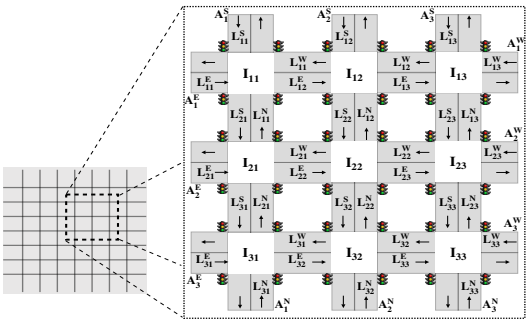


Fig. 1: An example $3 \times 3$ road network

Within this $m \times n$ network, the traffic flow in each arterial (and hence each link) has a corresponding non-conflicting flow that neither disrupts nor hampers the former arterial's (or links') ongoing traffic flow, i.e., the vehicles traveling through $A^E_\alpha$(and therefore links $L^E_\alpha$) do not interrupt the traffic flow travelling west-bound through $A^W_\alpha$(and therefore links $L^W_\alpha$). Considering this, the traffic lights at each intersection in the network follow the *phase sequences* such that the non-conflicting flows can enter and access the intersections simultaneously, i.e., the vehicles in links $L^N_\alpha$ and $L^S_\alpha$, or links $L^E_\alpha$ and $L^W_\alpha$, can utilize the intersections $I_\alpha$ without conflicting with each other's traffic flow.

The traffic flow within the $m \times n$ network is managed by a *traffic manager* (TM), which aggregates traffic information on the incoming traffic patterns and flows from the traffic sensors, traffic forecast data and/or data from the neighboring TMs. The TMs then execute some traffic control strategy to calculate the traffic timings at each intersection within the network under its purview. Our approach only requires that the necessary traffic information is readily conveyed to our TM without depending on how such data is acquired. This, however, necessitates that the TMs have minimal cloud connectivity to exchange the traffic information with other neighboring TMs and/or the traffic infrastructure. A TM managing an $m \times n$ road network is denoted by $TM_{m,n}$. In practice, the size of the $m \times n$ network is determined as per the feasible range of connectivity among the traffic infrastructure considering the communication latency overhead and the availability of the computational resources required to establish real-time control.

### B. Traffic Flow

Each intersection within the network is controlled by traffic signals that change between the green-yellow-red lights as per the assigned timings. A *cycle* is said to have completed when all the traffic signals located at any given intersection have completed one rotation of lights. The time taken to complete an entire cycle is called the *cycle time* $(T_c)$, after which the

signals repeat the pattern. The maximum number of vehicles that the lanes within the link can accommodate, known as link capacity, is an indicator to measure spillbacks [19] and can be estimated as,

$$z_\alpha^D = \frac{l_\alpha^D}{v+s}, \tag{1}$$

where $z_\alpha^D$ and $l_\alpha^D$ are the link capacity and the length of the link $L_\alpha^D$, respectively, while $v$ and $s$ denote the average passenger vehicle length and the safe spacing between two consecutive vehicles, respectively. A spillback is said to have occurred when one (or more) links within the network reach its capacity.

An incoming link $L_\alpha^D$ is characterized by the tuple $\{a_{\alpha,k}^D, q_{\alpha,k}^D, z_\alpha^D\}$, where $a_{\alpha,k}^D$ is the incoming vehicle flow rate during the $k^{th}$ traffic cycle, $q_{\alpha,k}^D$ is the number of vehicles queued in the link at the beginning of the $k^{th}$ cycle, $z_\alpha^D$ is the link capacity, which does not change over time, and $D$ is the direction of travel ($D \in \{N, E, W, S\}$).

While the incoming vehicle flow rate $a_{\alpha,k}^D(t)$ varies with time, we assume that the flow remains fixed within a given traffic cycle of $T_c$. Under dynamic traffic conditions, the worst-case flow rate within $T_c$ can be bounded and used in the analysis instead. We thus refer to $a_{\alpha,k}^D(t)$ as $a_{\alpha,k}^D$ for the rest of the paper. Note that the TM, in our case, only needs to acquire the flow rate information for the traffic flow entering the network it is managing, i.e., the TM needs to know the vehicle flow rates entering the links on the edge of the $m \times n$ network denoted by $L_1^D$ in the arterials $A_{\alpha_1}^D, D \in \{E, S\}$ and $L_\lambda^D$ in $A_{\alpha_\lambda}^D, D \in \{N, W\}$. To synchronize the traffic control over a network of intersections, a fixed $T_c$ value is assumed for each TM. The selection of $T_c$ can be made as per the traffic demands and network capacity [20].

To determine, $n_{\alpha,k}^D$, the number of vehicles that can be dispatched from each lane within a link in a given green time interval, direction, and cycle, we make use of the *saturation headway* model [21]. This model accounts for the lost time ($t_l$) and the time headway between the consecutive vehicles ($h$) while ensuring safe distances between said vehicles. $t_l$ comprises of the human reaction time for the vehicles to react to the light turning green, the yellow light time and the clearance time required to let the vehicles inside the intersection cross safely before the flow from the next phase is allowed to enter the intersection. Accordingly,

$$T_k = h \cdot n_{\alpha,k}^D + t_l. \tag{2}$$

In other words, $T_k$ denotes the time required to discharge $n_{\alpha,k}^D$ vehicles from link $L_\alpha^D$ during the $k^{th}$ traffic cycle when the saturation headway $h$ and the lost time $t_l$ are considered. $h$ and $t_l$ are defined as 2 s and 4 s, respectively [22].

## III. A Real-Time Task Scheduling Problem for a Network of Intersections

We now model an $m \times n$ road network (Figure 1) with multiple intersections controlled by the traffic manager $TM_{m,n}$ as a set of real-time tasks. To do so, we leverage an existing work [19] which models a single intersection as a real-time task scheduling problem. As described in Section II, each intersection within our network has incoming links from north, south, east, and west directions with four traffic lights located at the end of each link. Each intersection has two phase sequences, one for the north-south non-conflicting flow and the other allowing the east-west non-conflicting traffic.

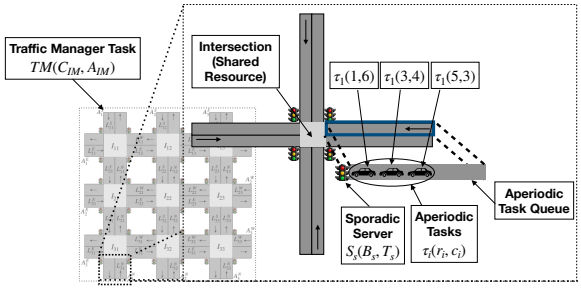



Fig. 2: Real-time task model for a single intersection [19].

### A. Review of Real-Time Task Model for a Single Intersection

We first briefly describe the existing real-time task model from existing work [19], which we leverage for a network of intersections. As shown in Figure 2, at a given intersection, the vehicles entering the network through one of the links as per the traffic flow rate and saturation headway model resemble aperiodic tasks ($\tau_i$) with varying arrival times ($r_i$) and execution times ($c_i$), respectively. The vehicles then form a queue and wait at the traffic signal until they are permitted to cross the intersection. This is equivalent to aperiodic tasks (vehicles) entering the aperiodic task queue (lanes) waiting to be executed to access the intersections. The traffic lights at an intersection permit (green light) or stop (red light) the vehicle flows from their corresponding links as per the signal timings and the phase sequence in the traffic cycle. Each traffic light is represented by sporadic servers [23] serving each task queue in the system and executing aperiodic tasks on the shared resource as per the available budget ($B_S$) and the inter-arrival times ($T_S$) of the sporadic servers ($S_S$), as shown in Figure 2.

The sporadic server (traffic light) serving a task queue executes the tasks on the shared resource as per its arrival time (traffic cycle) and its assigned budget (green time). Once the server exhausts its budget or all the tasks in the task queues are executed, the server stops executing (red light). As the sporadic servers within the system are invoked as per the arrival times and execute the tasks as per the assigned budget, the different vehicle flows from all four directions flow through the network. Sporadic servers serving non-conflicting flows are paired to execute on the shared resource at the same time, as they do not impede each other's flow. The resource allocation for the server groups is calculated as per the dominating flow within each traffic cycle, depending on the peak traffic movement [19].

### B. Extension to Multiple Intersections

In an $m \times n$ network, each link $L_\alpha^D$ in the network, where $D \in \{N, E, W, S\}$, is considered as an aperiodic task queue.

There are $4mn$ sporadic servers and $mn$ shared resources in an $m \times n$ network. The vehicles traveling on their desired route through the network, crossing the intersections in their route, and eventually exiting the network, are akin to aperiodic tasks joining the subsequent queues, waiting for execution at multiple shared resources.

The traffic manager for the $m \times n$ network, $TM_{m,n}$, is itself represented by a sporadic task. The TM task is responsible for calculating the budgets and the inter-arrival times for the sporadic servers such that the traffic flow through the network is maximized as per a budget assignment strategy. The TM task ensures that none of the traffic flows are starved of resources, the conflicting flows are not allowed to enter the intersection at once, and no resources are wasted within the network.

As discussed in Section I, our goal is to not only optimize traffic flow and avoid spillbacks in normal traffic states but also provide a mitigation mechanism to regulate and recover from the spillbacks that may have occurred leading to extreme traffic states within the road network due to special circumstances. The TM task is therefore responsible for choosing an appropriate strategy depending on the traffic state, as shown in Figure 3b. Figure 3a shows that, at the beginning of each traffic cycle, the TM task is activated and gathers the relevant traffic information for all the links and intersections within its network. Then, under a normal traffic state, the TM calculates the budgets for all the sporadic servers as per an optimal budget distribution strategy such that the traffic flow through the network is maximized while avoiding spillbacks. However, if a special event such as ERV preemption or a collision has caused spillbacks in any of the arterials, the TM task utilizes a PID controller assisted mitigation strategy to calculate the budgets for the server. All servers are released at the start of each traffic cycle and their priorities are assigned such that the conflicting servers at a given intersection are executed non-preemptively in a round-robin fashion. The vehicles are allowed to move through the network as the servers are executed as per their assigned budgets, exactly once in a given cycle. At the start of the next traffic cycle, the TM task is invoked again and the process is repeated.

For simplicity, the formulations in this paper focus on a link with a single lane without turning vehicles. However, as shown in Figure 4, additional servers can be deployed for modeling a link with multiple lanes for turning and going straight at each intersection within the network. The servers can be grouped as per the non-conflicting flows and the total budget ($T_c$) can then be divided among all servers at the intersection.

## IV. Optimal Budget Distribution Strategy

As discussed earlier, for calculating the budgets and the arrival times for the sporadic servers in a normal traffic state, the TM must rely on a budget assignment strategy. Our goal is to design an optimal strategy that maximizes the traffic flow through the entire network while avoiding spillbacks in any of the links within the network. By maximizing the traffic flow through the network, the vehicles will experience smaller wait times and hence reduced travel times. To achieve our goal, we formulate the budget distribution problem for the sporadic servers as a linear optimization problem, as discussed next.

### A. Optimization Constraints and Objective Formulation

Lemma 1 [19] defines the condition to avoid spillbacks in a given link. To enhance readability, we denote $L^D_\alpha$ as $L$ represented by the tuple $\{a, q, z\}$ with the sporadic server serving the link as $S$.

**Lemma 1** ( [19]). *For a link $L$ during the $k^{th}$ cycle, let us assume that $a_k$ is the traffic flow rate, $q_k$ is the existing queue length, i.e., number of vehicles already queued in link $L$, at the start of the $k^{th}$ cycle, and $z$ is the capacity. Let $n_{out_k}$ be the number of vehicles that are dispatched from lane $L$ in a cycle of length $T_c$. Then, a spillback is avoided within that cycle if*

$$n_{out_k} \geq a_k T_c + q_k - z + 1. \tag{3}$$

Let us now define the utilization of a sporadic server $S$ as $U = \frac{B_S}{T_c}$. Depending on the flow rate, queue, capacity, and arrival time, every server $S$ will have a minimum and maximum utilization demands, $U_{min_k}$ and $U_{max_k}$ respectively, within the $k^{th}$ cycle [19], given by

$$U_{min_k} = \frac{h \cdot n_{out_k} + t_l}{T_c} \tag{4}$$

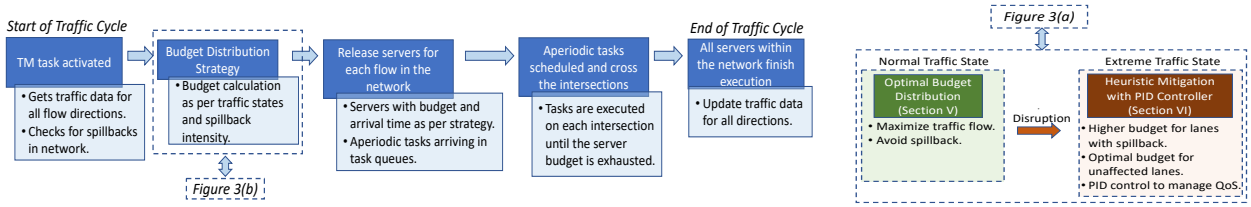
$$U_{max_k} = \frac{h \cdot (a_k \cdot T_c + q_k) + t_l}{T_c} \tag{5}$$

$U_{min_k}$ and $U_{max_k}$ correspond to the minimum budget requirement to avoid spillbacks in a given link and ensuring that no resources are wasted, respectively. As discussed earlier, the resource allocation to the server pairs is made as per the dominating flow within each traffic cycle. Thus, if there are no spillbacks in the dominating traffic direction, then this prevents spillbacks in the opposing direction too, since both the flows are given the same green time. Without loss of generality, we assume that the south-bound and east-bound traffic dominates the north-bound and west-bound traffic respectively, within each traffic cycle, in the rest of this work for simplicity. The proposed analysis can be extended for any dominating flows, depending on the traffic movement, by changing the notations. Using this assumption, we denote the servers $S^S$ and $S^N$ at each intersection as $S^{SN}$ and correspondingly $S^{EW}$, such that, $U^D_k = U^S_k = U^N_k$, if $D = \{SN\}$ and $U^D_k = U^E_k = U^W_k$, if $D = \{EW\}$.

Since our TM only gathers flow rates for the vehicles entering our network (Section II), (4) and (5) are used as constraints only for the servers serving the incoming links at the edges of our network, i.e., links $L^D_1$ for arterials $A^D_\alpha, \forall D = \{SN, EW\}$, e.g., links $L^S_{1j}$ and $L^E_{i1}, \forall i \in [1,3], \forall j \in [1,3]$ from Figure 1. Therefore, the utilization constraints for the traffic flows in links $L^D_1$ within all arterials $A^D_\alpha$, for $k^{th}$ traffic cycle are as follows:

$$U^D_{min_\alpha,k} \leq U^D_{\alpha,k} \leq U^D_{max_\alpha,k} \tag{6}$$

For the interior links, the vehicle flow rate depends on the assigned budgets of the servers located at the intersection




(a) (b)

Fig. 3: (a) TM activities during a traffic cycle with (b) budget distribution strategies to optimize traffic and mitigate spillbacks.

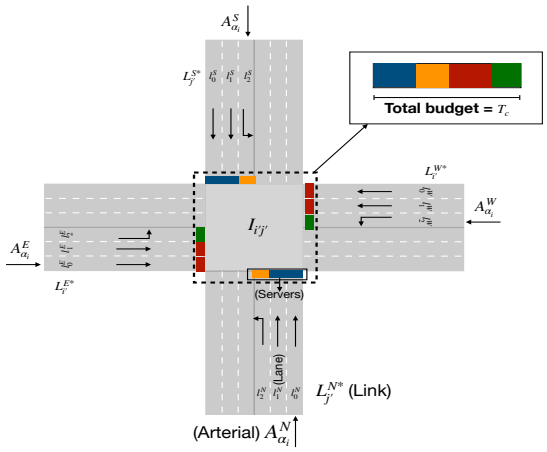



Fig. 4: Extension to a more realistic environment. The colored tabs resemble servers located at the intersection. Identically colored tabs indicate servers grouped together for execution.

upstream, i.e. the traffic flow rate for $r'^{th}$ link within any arterial $A^D_\alpha$, denoted by $L^D_{r'}$ depends on the budget at the intersection upstream, i.e., $U^D_{r'-1}$. If the budget of the server at the intersection upstream is $U^D_{r'-1}$, then the corresponding green time is $U^D_{r'-1} \cdot T_c$, where,

$$r'-1 = \begin{cases} (r'-1)i, \ r' \in [2,m], i \in [1,n] & \text{if } D = \{SN\} \\ i(r'-1), \ i \in [1,m], r' \in [2,n] & \text{if } D = \{EW\} \end{cases}$$

To avoid spillbacks in the interior links, Lemma 1 and (6) are utilized to impose constraints on the server's budget:

$$U^D_{\alpha-1} - U^D_\alpha \leq \frac{z^D_\alpha h}{T_c} \tag{7}$$

Equation (7) ensures that as the vehicle flow propagates through the network, none of the links exceed the capacity thereby avoiding spillbacks within $T_c$ at each intersection. Since the budget for each server at any intersection is reserved from the total cycle time, the constraint to avoid processor overloaded, is given by

$$U^{SN}_\alpha + U^{EW}_\alpha \leq 1 \tag{8}$$

The constraints (6) through (8) are calculated as per the traffic flow parameters of the dominating flow among north-south and east-west directions, at the start of each traffic cycle. Since we aim to maximize the traffic flow through the network, which is controlled through server budgets at intersections inside the network, the objective function of our linear optimization problem, which is solved for each cycle $T_c$, is

$$\begin{aligned} \text{maximize} \quad & \sum_{\substack{i=1\ldots m,\\ j=1\ldots n,\\ D_c \in \{SN,EW\}}} U^{D_c}_\alpha \\ \text{subject to} \quad & (6)-(8) \end{aligned} \tag{9}$$

### B. Cumulative Worst-Case Wait Time Analysis

We now provide the cumulative worst-case delay that a vehicle may be subjected to when it travels through an entire arterial $A^D_\alpha$ within an $m \times n$ network. This worst-case delay accounts for the total time that a vehicle has to completely stop while traversing through the arterial $A^D_\alpha$ in a normal traffic state with the proposed optimal budget distribution strategy.

**Theorem 1.** *Assuming that the traffic flow rate for vehicles entering the arterial $A^D_{\alpha_r}$, traveling through all links $L_i$ is a, the cumulative wait time $W^D_{\alpha_{r,p}}$ for a vehicle at the $p^{th}$ position in the queue upon entering the arterial in the $k^{th}$ cycle is bounded by*

$$W^D_{\alpha_{r,p}} \in \left[ 0, \ \sum_{k=1}^{\left\lfloor \frac{p}{n_{out_{1_k}}} \right\rfloor} \left( \lambda T_c - \lambda U_{min_{1,k}} T_c + \sum_{i=2}^{\lambda} (\lambda - i + 1) z_i h \right) \right] \tag{10}$$

*where $\lambda = n$, if $D = \{S, N\}$, and $\lambda = m$, otherwise. $n_{out_{1_k}}$ is the minimum number of vehicles dispatched from link $L_1$ of arterial $A^D_{\alpha_r}$ in the $k^{th}$ cycle, $U_{min_{1,k}}$ denotes the minimum utilization demand for link $L_1$ in the arterial, in the $k^{th}$ cycle, and all other variables are as defined previously.*

*Proof:*

Let us consider a vehicle entering our $m \times n$ network from the $r^{th}$ arterial with the desired route to travel on this arterial, $A^D_{\alpha_r}$. The $i^{th}$ link along the arterial is denoted by $L_i$ and the server $S_i$ serving the link $L_i$ has an assigned budget of $U_i$. The worst-case wait time will occur when the vehicles join the queues at each of the intersections exactly when the lights turn from green to red. Also, the worst-case wait time will only occur when the budget assigned to the servers serving all the links in the vehicle's route is its minimum utilization demand. As the cycle times at all intersections are fixed to $T_c$ and the servers are executed in the round-robin fashion, the worst-case

wait time for the first vehicle in the queue in link $L_i$ will be $T_c - U_{min_i} T_c$. Therefore, the total worst-case wait time for the vehicle to traverse through the entire arterial that entered in the $k^{th}$ cycle is

$$W^D_{\alpha_r} = \sum_{i=1}^{\lambda} \left(T_c - U_{min_{i,k}} T_c\right). \tag{11}$$

Here, $\lambda = m$ if $D = \{S, N\}$, and $\lambda = n$ otherwise. From (7), the minimum utilization for servers $S_i$, is given by,

$$U_{min_{i,k}} = U_{min_{(i-1),k}} - \frac{z_i h}{T_c}. \tag{12}$$

In addition, since

$$\begin{aligned} U_{min_{2,k}} &= U_{min_{1,k}} - \frac{z_2 h}{T_c}; U_{min_{3,k}} = U_{min_{2,k}} - \frac{z_3 h}{T_c}; \\ \Rightarrow U_{min_{3,k}} &= U_{min_{1,k}} - \frac{z_2 h}{T_c} - \frac{z_3 h}{T_c}; \\ U_{min_{\lambda,k}} &= U_{min_{1,k}} - \sum_{i=2}^{\lambda} \frac{z_i h}{T_c}, \end{aligned}$$

we have

$$\begin{aligned} U_{min_{1_k}} + U_{min_{2_k}} + U_{min_{3_k}} &= 3U_{min_{1_k}} - 2\frac{z_2 h}{T_c} - \frac{z_3 h}{T_c} \\ \Rightarrow \sum_{i=1}^{\lambda} U_{min_{i,k}} &= \lambda U_{min_{1,k}} - \sum_{i=2}^{\lambda} (\lambda - i + 1)\frac{z_i h}{T_c} \end{aligned}$$

Substituting in (11), we can derive $W^D_{\alpha_r}$, the worst-case wait time for the first vehicle in queue at the $k^{th}$ cycle as

$$\begin{aligned} W^D_{\alpha_r} = \sum_{i=1}^{\lambda} \left(T_c - U_{min_{i,k}} T_c\right) &\Rightarrow T_c \left(\lambda - \sum_{i=1}^{\lambda} U_{min_{i,k}}\right) \\ &\Rightarrow T_c \left(\lambda - \lambda U_{min_{1,k}} + \sum_{i=2}^{\lambda} (\lambda - i + 1)\frac{z_i h}{T_c}\right) \\ &\Rightarrow \lambda T_c - \lambda U_{min_{1,k}} T_c + \sum_{i=2}^{\lambda} (\lambda - i + 1) z_i h \end{aligned}$$

For the $p^{th}$ vehicle, when $n_{out_{1_k}}$ is the minimum number of vehicles dispatched from link $L_1$ when $U_{min_{1,k}}$ budget is assigned, the worst-case wait time $W^D_{\alpha_r,p}$ is

$$W^D_{\alpha_r,p} = \sum_{k=1}^{\left\lfloor \frac{p}{n_{out_{1_k}}} \right\rfloor} \left(\lambda T_c - \lambda U_{min_{1,k}} T_c + \sum_{i=2}^{\lambda} (\lambda - i + 1) z_i h\right) \tag{13}$$

Clearly, the best-case scenario occurs when the vehicle never has to wait at any of the traffic signals on its route. Hence, the best-case wait time is zero and the theorem is proved. ■

## V. Recovery Approach

The optimal budget distribution strategy maximizes the traffic flow through the network while avoiding spillbacks in normal traffic states. However, special situations leading to blocked intersection(s), such as the triggering of an emergency preemption system or the existence of a temporary roadblock, can cause prolonged red lights and may even result in spillbacks [24]. Such extreme traffic states cannot usually be remediated by the optimal budget distribution strategy, as it does not prioritize eliminating spillbacks (see Section VI). Since spillbacks can have a domino effect that can eventually disrupt the entire network, a mitigation approach that focuses on eliminating spillbacks is needed during such extreme traffic states.

Our heuristic mitigation strategy prioritizes links already experiencing spillbacks after a disruption leading to an extreme traffic state. This heuristic approach ensures that the links experiencing spillbacks are cleared without creating bottlenecks in other links within the network so that the traffic can resume to a normal operating state as quickly as possible. The time required for the mitigation approach once it is triggered to return to a normal traffic state is defined as the *recovery time* and our approach aims at minimizing said recovery time without causing spillbacks in other links. We specifically target severe spillbacks which may have originated from one of the links in the arterial but which have now spread through all the links disrupting the entire arterial. For example, in Figure 1, let us consider that a disruption in $I_{23}$ causes the second link $L^E_2$ of the third arterial $A^E_{\alpha_3}$ to spillback. Due to a prolonged red light at $S^E_2$ serving the link $L^E_2$, the spillback propagates through the entire arterial $A^E_{\alpha_3}$. After the disruption is cleared, the arterial $A^E_{\alpha_3}$ needs to be returned to normal operations by eliminating the spillback while ensuring that the all links intersecting with $A^E_{\alpha_3}$, i.e., links $L^D_3$ in arterials $A^D_{\alpha_i}, \forall D \in \{N, S, W\}, i \in [1, \lambda]$ do not experience a spillback and propagate it through their corresponding arterials.

### A. Mitigative Budget Distribution

The mitigation strategy provides 100% of the total budget at each intersection along the arterial with spillback and 0% to the conflicting flow at a given intersection. However, giving 0% budget to any of the links will cause vehicles to accumulate as they are not allowed to dispatch, and thereby eventually experience spillback. To avoid this, we define the number of *cycles to spillback* to calculate the duration for which it is safe to give a 0% budget to any of the links without causing a spillback from the gathered traffic information. This number of *cycles to spillback*, $N$ can be found as follows.

**Property 1** (Number of cycles-to-spillback)**.** *Let a link $L^D_i$ (denoted by $L$) in arterial $A^D_\alpha$ passing through an intersection $I_\alpha$ with $T_c$ cycle time, where $a_k$ denotes the worst-case flow rate in $L$ at the $k^{th}$ traffic cycle and remains constant for the foreseeable future, $q_k$ is the existing queue length, i.e., number of vehicles already queued in $L$ at the beginning of the $k^{th}$ traffic cycle, and $z$ is the capacity calculated using (1), then, link $L$ will spillback after $N$ cycles where*

$$N = \left\lceil \frac{z - q_k}{a_k T_c} \right\rceil. \tag{14}$$

**Remark 1.** *This and the remaining proofs have been omitted due to space limit and are provided [25] for interested readers.*

Our recovery approach calculates the minimum of the cycles to spillback, $N_{min_{\alpha_r}}$ among all links with conflicting flows to the arterial $A^D_{\alpha_r}$ experiencing spillback, i.e., $N_{min_{\alpha_r}} = \min(N_r^{D'})$, where $N_r^{D'}$ denotes the cycles to spillback for links $L_r^{D'}$ in arterials $A^{D'}_{\alpha_i}$, $\forall i \in [1, \lambda]$, such that if $D \in \{S, N\}, D' \in \{E, W\}$ and vice-versa. $N_{min_{\alpha_r}}$ indicates that we can provide zero budget to all the conflicting links without causing a spillback in any of them for $N_{min_{\alpha_r}}$ cycles. At the $N^{th}_{min_{\alpha_r}}$ cycle, the recovery approach must terminate and switch to the optimal approach as any additional cycles with zero budget will cause a spillback in at least one of the links.

The heuristic budget distribution is only applied to the arterial experiencing spillback and the links entering the intersections affected by this spillback. Considering the example in Figure 1, if the arterial $A^S_{\alpha_2}$ is experiencing spillbacks, the servers serving the links $L^S_{12}, L^S_{22}, \ldots, L^S_{m2}$ and the corresponding non-conflicting links will be assigned 100% of the total budget and the servers serving links $L^E_{12}, L^E_{22}, \ldots, L^E_{m2}$ and their corresponding non-conflicting links will be assigned 0% budget for $N_{min_{\alpha_r}}$ traffic cycles, $\forall i \in [1, \lambda]$. However, the rest of the servers in the network will still follow the optimal budget distribution proposed in Section IV. This ensures that the traffic flows through the links unaffected by the spillback are maximized optimally while considering the prioritized budget distribution at the spillback-affected links.

### *B. Worst-Case Recovery Time Analysis*

Using Property 1, we now provide the worst-case recovery time when our heuristic mitigation approach is in place and spillback is propagated through multiple links in an arterial.

**Theorem 2.** *Assuming that the spillback has propagated through $p$ links within the arterial $A^D_{\alpha_r}$ and the heuristic mitigation approach is activated to recover from the spillback, then the recovery time is be bounded by*

$$R^D_{\alpha_r,p} \in \Big[\sum_{i=1}^{p} h \cdot z_i + p \cdot t_l, \big(T_c N_{min_{\alpha_r}} + W^D_{\alpha_r,p'}\big)\Big], \textit{where,}$$

$$W^D_{\alpha_r,p'} = \sum_{k=1}^{\left\lfloor \frac{v_{rem}}{n_{out}{}_{1_k}} \right\rfloor} \Big(p' T_c - p' U_{min_{1,k}} T_c + \sum_{i=2}^{p'} (p' - i + 1) z_i h\Big),$$

*$p' (\leq \lambda)$ indicates the number of links still experiencing spillbacks after dedicating 100% budget for $N_{min_{\alpha_r}}$ cycles, $v_{rem}$ indicate the number of vehicles still queued within the $p'$ links and all other notations are as defined earlier.*

### *C. Cumulative Worst-Case Wait Time Analysis*

During extreme traffic states, using the mitigation strategy in some of the links will change the worst-case wait times experienced by the vehicles as discussed next.

**Theorem 3.** *Let us assume that the arterial $A^{D'}_{\alpha_{r'}}$ is experiencing spillback and is using the heuristic approach to calculate the budgets and the conflicting traffic flow in the arterial $A^D_{\alpha_r}$ is a such that $D' \in \{E, W\}$ if $D \in \{S, N\}$ and vice-versa. The cumulative wait time $W^D_{\alpha_r,p}$ for a vehicle at $p^{th}$ position in the queue entering the arterial $A^D_{\alpha_r}$ in the $k^{th}$ cycle is bounded by*

$$W^D_{\alpha_r,p} \in \Big[0, \Big(W^D_{r(r'-1)} + W^D_{rr'} + W^D_{r\lambda}\Big)\Big] \tag{15}$$

*where,*

$$W^D_{r(r'-1)} = \sum_{k=1}^{\left\lfloor \frac{p}{n_{out}{}_{1_k}} \right\rfloor} \Big(r' T_c \big(1 - U_{min_{1,k}}\big) + \sum_{i=2}^{r'} (r' - i + 1) z_i h\Big),$$

$$W^D_{rr'} = {min_{\alpha_{r'}}} T_c,$$

$$W^D_{r\lambda} = \sum_{k=1}^{\left\lfloor \frac{p}{n_{out}{}_{r'_k}} \right\rfloor} \Big((\lambda - r' + 1) T_c \big(1 - U_{min_{r',k}}\big) + \sum_{i=r'+1}^{\lambda} (\lambda - i + 1) z_i h\Big).$$

While our heuristic approach can provide prioritized budget distribution to the links experiencing spillback for $N$ cycles, for cases with severe spillbacks spread through long arterials, $N$ cycles may not be enough to eliminate the spillback. In such cases, we still need to prioritize the links experiencing spillback. At the same time, the conflicting flows need some minimum traffic flow in each traffic cycle to avoid the domino effect of having spillbacks in the entire network. This minimum traffic flow may be decided as per the quality-of-service (QoS) required in the links with conflicting flows. We next design a PID controller to assist the heuristic approach to facilitate the QoS of the links with conflicting flow, while still eventually eliminating spillbacks from the network.

### *D. Controller-Assisted Heuristic Approach*

This PID controller-assisted approach is utilized by the TM only if the prior heuristic approach is not able to eliminate spillbacks from the network within $N$ (cycles to spillback). The PID controller requires a desired queue for each link to evaluate the amount of budget to be reserved for the servers serving the conflicting links. Let us consider the previously used notations, $A^D_{\alpha_r}, L^D_i$ and $\{a_i, q_i, z_i\}, i \in [1, \lambda]$ as the arterial, its links and the corresponding parameters, respectively. Ideally, to avoid spillback, this minimum desired vehicle queue would be one less than the link capacity $z^D_i$ in each link. However, depending on the QoS required in the conflicting links (see below), this desired queue value for each link can be re-set at the beginning of each traffic cycle. Once the desired queue $q_i^{'D}$ is set, the PID controller will adjust the budget allocation and hence the green light timings as follows:

- The PID controller tracks the difference between the set threshold for the desired queue and the actual vehicle queues at the beginning of each traffic cycle.
- The controller then outputs $\mu^D_i$ which is a ratio of the total budget to be dedicated to the corresponding server, $S^D_i$, i.e., the assigned budget $U^D_i = \mu^D_i T_c$.
- The remaining budget is assigned to the link with spillback, i.e., $U^{D'}_i = (1 - \mu^D_i) T_c$, where if $D \in \{S, N\}$, $D' = \{E, W\}$ and vice-versa.
- The updated vehicle queues after the execution of the servers acts as a feedback to the controller.

If $q_i^{'D}(t)$ is the reference queue value set for the link $L^D_i$ in a given arterial $A^S_{\alpha_r}$ and $q^D_i(t)$ is the actual vehicle queues at

time $t$, then the PID controller tracks the error signal $e_i^D(t) = {q_i^{'}}^D(t) - q_i^D(t)$ with the control law described as

$$\mu_i^D(t) = k_p e(t) + k_i \int e(t)dt + k_d \frac{de(t)}{dt}, \tag{16}$$

where $dt$ is the time step for the feedback control, $k_p, k_i$ and $k_d$ are proportional, integral and differential gains respectively. Since, we perform calculations at every traffic cycle, $dt = T_c$.

The PID controller gains ($k_p, k_i$ and $k_d$) must be tuned for varying traffic conditions. As the incoming flow rate changes, the controller gains need to be changed to correctly track the error in the vehicle queue. The controller gains can be tuned offline as per various QoS levels for a particular link and stored in an offline lookup table. Appropriate gain values can be fetched by the PID controller as per the actual traffic flow rates while executing the online control at the beginning of each traffic cycle. Once the spillback is eliminated, the TM switches back to using the optimal budget strategy for normal operations.

## VI. Simulations

We compare our approach against (i) a traffic flow-based adaptive traffic control technique, and (ii) a real-time spillback-based approach [19], i.e., the most related approach. The flow-based adaptive approach is used as a signal timing calculation method in well-known global traffic control techniques such as SCATS [20], where the signal timings within a traffic cycle are proportional to the incoming vehicle flow detected through sensors and/or other available data. Alternately, the real-time spillback-based approach [19] uses traffic information from neighboring intersections to adjust the signal timings at an isolated intersection to avoid spillbacks. More sophisticated network-wide optimization approaches [9], [10], [13] lack scalability while others require vehicle-to-vehicle connectivity and/or vehicle autonomy [26], [27] which may not be practical as most roadways will still be dominated by conventional vehicles for the near future [28].

### A. Simulation Setup

We use a tick-based traffic simulator developed in Python 3 to simulate desired vehicle flow with specified flow rates, safety distances, desired intersection/ network architecture and traffic control algorithms to be implemented. The traffic flow behaviour is as per the Highway Capacity Manual [22]. For a comprehensive evaluation in dynamic traffic, we simulate vehicle flow rates varying from 1–11 veh/min (60–600 veh/hr). This vehicle flow rate range encapsulates the critical volume-to-capacity ratios for signalized intersections, provided by the Federal Highway Administration [29] as shown in Table I.

Vehicle flows of more than 8 veh/min for a long duration entirely disrupt the network as multiple intersections within the network will be operating beyond capacity. We execute the simulator for different scenarios encompassing normal traffic states with optimal budget distribution strategy to control the traffic signals, as well as the heuristic mitigation approach by inducing spillbacks in a $3 \times 3$ network. We then collect the wait time data for each vehicle in the network.

TABLE I: Flow types simulated as per the traffic flow rates and its description as per the FHWA [29]

| Flow Type | Flow rate per lane (Net flow in $3 \times 3$ network) | Description |
|---|---|---|
| Low | 1–4 veh/min (540–2100 veh/hr) | Network running under capacity with reduced travel delays |
| Medium | 4–6 veh/min (2100–3200 veh/hr) | Network nearing capacity with longer queues |
| Heavy | up to 8 veh/min (up to 4300 veh/hr) | Unstable traffic flow with long wait times |

### B. Normal Traffic States

Results in Figure 5a indicate that our proposed optimal approach shows only slight improvement during low traffic (11%) since spillbacks do not occur with low vehicle movements. However, with heavy traffic flows, not only does our proposed approach outperform the flow-based and the spillback-based approaches by 37% and 33%, respectively, in worst-case wait times, but also ensures that no spillbacks occur, unlike the flow-based approach. Additionally, our proposed approach shows average wait time reductions of 9.1% and 16.3% over the flow-based and the heuristic spillback-based approach during heavy traffic. To validate the adaptability of our approach to fluctuating traffic flows, we simulate medium traffic flow (4–6 veh/min) with a surge in traffic of up to 11 veh/min for one traffic cycle, in every 2–20 cycles. Figure 5b shows that our proposed optimal strategy introduces 22.6% to 28.9% less wait time as compared to the spillback-based approach and 38.1% to 53.2% less wait time as compared to the flow-based approach, in the worst-case. Also, as observed in Figure 5b, both spillback-based and flow-based approaches are not able to adapt to highly fluctuating traffic (2–5 cycles) and induce spillbacks in the network, unlike the optimal approach.

### C. Extreme Traffic States

Figure 5c shows the performance of the flow-based approach, our naïve mitigation approach (dedicating 100% budget to links with spillback), as well as our mitigation + PID-assisted heuristic in terms of the time it takes to recover to normal traffic state once the disruption has occurred. Note that the spillback-based approach does not have a mitigation technique in case spillbacks exist and relies on the flow-based approach until the network recovers from spillback and hence shows similar recovery times as the flow-based approach. The PID-controller is tuned to maintain vehicle queues of 20 veh in links with capacity of 24 veh. Our naïve mitigation approach recovers from the spillback within 52 s as compared to the mitigation + PID-assisted approach (108 s) and the flow-based approach (220 s). However, as explained in Section V, the mitigation + PID-assisted heuristic approach also allows the other affected flows in the network to maintain the desired QoS. In case of the naïve mitigation heuristic, the traffic flow is completely halted for multiple cycles posing a risk to those links in case there are traffic fluctuations. Therefore, even if the mitigation + PID-assisted heuristic takes slightly longer to recover from the spillbacks, the minimum QoS ensures that none of the other links in the network are at risk of spillback, while still recovering up to 50.9% quicker than the flow-based

approach. Additionally, Figure 5c shows that the vehicle queues remain close to the capacity for flow-based approach, making it susceptible to spillbacks even with minor traffic fluctuations.

## VII. Hardware-In-Loop Validation

We now validate our proposed budget distribution approach on a hardware-in-loop (HIL) testbed consisting of small-sized robots. This HIL testbed consists of the hardware setup (robots emulating vehicles in one of the intersections of the network) and a software framework that runs the traffic simulator (remaining intersections of the network) in tandem with the hardware setup. Such a HIL setup enables validating traffic approaches on a larger network in a confined space.

### A. HIL Setup

Our experimental setup consists of 30 small robots, each representing a vehicle. The HIL framework ensures communication between the software simulator and the robots to exchange the vehicle flow rates and signal timing information, ensuring emulation of a large traffic network in real-time. Each robot henceforth referred to as a vehicle, is affixed with multiple infrared (IR) markers. The hardware testbed consists of 24 IR cameras and the Optitrack motion capture system to track the vehicle positions. The position data is streamed to a command computer where the Robot Operating System-based (ROS) [30] HIL framework simulates the traffic network and controls the traffic signal timings and the vehicles.

The HIL framework consists of a controller application that processes the raw position data from the cameras using a Kalman filter to reduce sensor noise and accurately estimate the position and the velocity of the vehicles. A pre-planned map with path coordinates resembling an intersection is utilized by a *pure pursuit controller* and the intelligent driver model (IDM) [31] to traverse the vehicles on their desired paths. The IDM and a *low level PI controller* ensures that the vehicles reach their desired velocities while maintaining a safe distance between the consecutive vehicles. The IDM also ensures that human driving reaction delays in an urban environment are replicated through the robots. The desired velocities are then converted into left and right wheel speeds as per the differential drive kinematics model for each vehicle. The wheel speeds are commanded over Zigbee to each robot consisting of LPC1768 microcontroller on the Pololu m3pi platform interfaced with Digi Xbee receivers to acquire the speed commands.

For our HIL-based validation, the hardware setup as shown in Figure 6a, represents intersection $I_{11}$ of the $3 \times 3$ network (Figure 1) and the remaining are simulated through the software framework. Figure 6a shows vehicle flows from four different directions entering the intersection $I_{11}$ with two non-conflicting flows given green light to enter the intersection while the other two flows have a red light with vehicles waiting in the queue. Due to lack of space, we only compare the response of our proposed approach and the spillback-based approach [19] to heavy traffic flow, but the results are generally representative. We measure the distance from the upstream stop line over time for each vehicle accessing the intersection. Decreasing (increasing) distance over time indicates that the vehicle is moving towards (away from) the stop line.

### B. Results

Figures 6b and 6c show the trajectories for vehicles utilizing the intersection $I_{11}$ with the spillback-based approach and the proposed approach, respectively, for one of the traffic cycles during the simulations. The vehicles traveling along arterial $A_1^S$ enter through the link $L_{11}^S$, cross the intersection $I_{11}$ and proceed for the intersection $I_{21}$ using the link $L_{21}^S$. The link capacities of the links $L_{11}^S$ and $L_{21}^S$ are 8 and 7 respectively, to emulate a dynamic traffic environment. Figure 6b shows that with the spillback-based approach, all eight vehicles in $L_{11}^S$, waiting at $I_{11}$ are allowed to enter $L_{21}^S$ to avoid spillbacks in $L_{11}^S$. However, due to per-intersection decision making, this leads to a spillback in $L_{21}^S$ (eight vehicles in a link with the capacity of 7 veh). However, as shown in Figure 6c, using our proposed approach, nine vehicles flow through the network, but the activation time and the server budget is such that only seven vehicles cross the intersection to enter $L_{21}^S$, while two vehicles form a queue in $L_{11}^S$, thereby avoiding spillbacks in any of the links.

## VIII. Conclusions

In this paper, we provided an optimal strategy to control the traffic signals within an $m \times n$ road network during normal traffic conditions. When an extreme traffic condition, i.e., long queues and spillbacks propagated through multiple lanes, arises due to events such as emergency response vehicle preemption, roadblocks, and collisions, we designed a heuristic mitigation strategy assisted by a PID controller to allow the network to recover from the spillbacks as quickly as possible while also maintaining a minimum desired quality-of-service through the rest of the network. By leveraging the real-time properties of our model, we also derived the worst-case bounds on the wait times (delays in travel times) caused by our approach, thereby making our system more predictable. With the help of simulations, using our optimal strategy results in up to 53% in the worst case and 16% on an average improvement in delays experienced by the vehicles. Additionally, our spillback mitigation strategy is able to recover from the spillbacks by up to 50% faster than existing approaches.

## References


[1] TomTom, "Tomtom traffic index," April 2019. [Online]. Available: https://www.tomtom.com/en_gb/traffic-index/

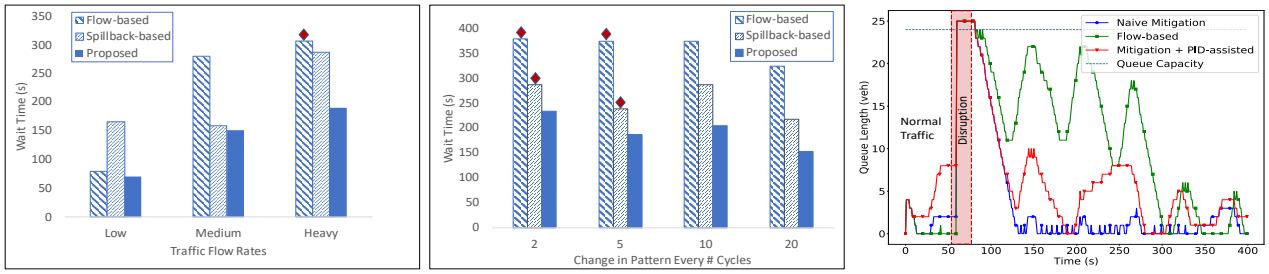
[2] INRIX, "Global traffic scorecard," INRIX, Tech. Rep., 2020.

[3] FHWA, "Intersection Safety, Safety Data and analysis," https://highways.dot.gov/research-programs/safety/intersection-safety, 2018.

[4] Y. Li, Z. Li, H. Wang, W. Wang, and L. Xing, "Evaluating the safety impact of adaptive cruise control in traffic oscillations on freeways," *Accident Analysis & Prevention*, vol. 104, pp. 137–145, 2017.

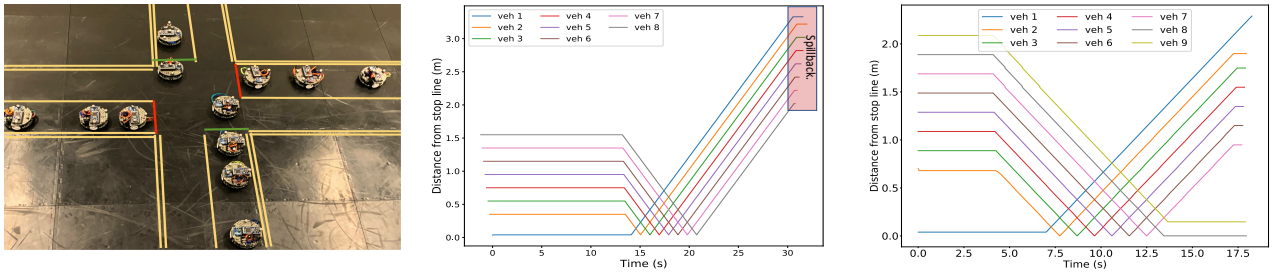
[5] D. Ma, X. Luo, S. Jin, W. Guo, and D. Wang, "Estimating maximum queue length for traffic lane groups using travel times from video-imaging data," *IEEE Intelligent Transportation Systems Magazine*, 2018.

[6] A. Downs, "Traffic: Why it is getting worse, what government can do," url=https://www.brookings.edu/research/traffic-why-its-getting-worse-what-government-can-do/, 2004.




(a) (b) (c)

Fig. 5: (a) Worst-case wait times for network-wide travel with varying traffic flow, (b) Worst-case wait times for unbalanced traffic surge at certain intervals for heavy traffic flow (red rhombus indicates that spillbacks occurred within the network), and (c) Recovery to normal traffic state post disruption.




(a) (b) (c)

Fig. 6: (a) HIL testbed resembling an intersection; Vehicle trajectories from the HIL testbed for (b) spillback-based approach, and (c) proposed optimal budget distribution approach.


[7] V. Paruchuri, "Adaptive preemption of traffic for emergency vehicles," in *2017 UKSim-AMSS 19th International Conference on Computer Modelling & Simulation (UKSim)*. IEEE, 2017, pp. 45–49.

[8] Y. Yuan, Y. Liu, and J. Yu, "Trade-off between signal and cross-elimination strategies during evacuation traffic management," *Transportation research part C: emerging technologies*, vol. 97, pp. 385–408, 2018.

[9] A. G. Sims and K. W. Dobinson, "The Sydney coordinated adaptive traffic (scat) system philosophy and benefits," *IEEE Transactions on vehicular technology*, vol. 29, no. 2, pp. 130–137, 1980.

[10] P. Hunt, D. Robertson, R. Bretherton, and R. Winton, "Scoot-a traffic responsive method of coordinating signals," Tech. Rep., 1981.

[11] J.-J. Henry, J. L. Farges, and J. Tuffal, "The prodyn real time traffic algorithm," in *Control in Transportation Systems*. Elsevier, 1984.

[12] P. D. Pant, Y. Cheng, A. Rajagopal, N. Kashayi *et al.*, "Field testing and implementation of dilemma zone protection and signal coordination at closely-spaced high-speed intersections," *Rep. No. FHWA/OH-2005/006, Ohio Dept. of Transportation, Columbus, OH*, 2005.

[13] Z. Shen *et al.*, "A novel learning method for multi-intersections aware traffic flow forecasting," *Neurocomputing*, 2019.

[14] D. Garg *et al.*, "Deep reinforcement learning for autonomous traffic light control," in *2018 3rd IEEE International Conference on Intelligent Transportation Engineering (ICITE)*. IEEE, 2018, pp. 214–218.

[15] X. Liang *et al.*, "A deep reinforcement learning network for traffic light cycle control," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 2, pp. 1243–1253, 2019.

[16] M. Inc., "2070e controller," https://www.mccain-inc.com/products/controllers/2070-controllers/2070e-controller, 2018.

[17] H.-C. Hu and S. F. Smith, "Using bi-directional information exchange to improve decentralized schedule-driven traffic control," in *Proceedings of the International Conference on Automated Planning and Scheduling*, vol. 29, no. 1, 2019, pp. 200–208.

[18] A. Ahmed, S. A. A. Naqvi, D. Watling, and D. Ngoduy, "Real-time dynamic traffic control based on traffic-state estimation," *Transportation research record*, vol. 2673, no. 5, pp. 584–595, 2019.

[19] P. Oza and T. Chantem, "A real-time server based approach for safe and timely intersection crossings," in *International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA)*. IEEE, 2019.

[20] A. Stevanovic, C. Kergaye, and P. T. Martin, "Scoot and scats: A closer look into their operations," in *88th Annual Meeting of the Transportation Research Board. Washington DC*, 2009.

[21] C. Q. Shao and X. M. Liu, "Estimation of saturation flow rates at signalized intersections," *Discrete Dynamics in Nature and Society*, 2012.

[22] HCM, "Highway capacity manual, 2010," *Transportation Research Board, National Research Council, Washington, DC*, 2010.

[23] D. Faggioli, M. Bertogna, and F. Checconi, "Sporadic server revisited," in *Proceedings of the 2010 ACM Symposium on Applied Computing*. ACM, 2010, pp. 340–345.

[24] E. J. Nelson and D. Bullock, "Impact of emergency vehicle preemption on signalized corridor operation: An evaluation," *Transportation research record*, vol. 1727, no. 1, pp. 1–11, 2000.

[25] P. Oza *et al.*, "A coordinated spillback-aware traffic optimization and recovery at multiple intersections," Virginia Tech, Tech. Rep., 2020. [Online]. Available: https://www.rtx.ece.vt.edu/resources/Files/oza20-tech.pdf

[26] S. Aoki and R. Rajkumar, "V2v-based synchronous intersection protocols for mixed traffic of human-driven and self-driving vehicles," in *2019 IEEE 25th International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA)*. IEEE, 2019, pp. 1–11.

[27] C. B. Rafter *et al.*, "Augmenting traffic signal control systems for urban road networks with connected vehicles," *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[28] Y. Feng, K. L. Head, S. Khoshmagham, and M. Zamanipour, "A real-time adaptive signal control in a connected vehicle environment," *Transportation Research Part C: Emerging Technologies*, 2015.

[29] FHWA-USDoT, "Signalized intersections: Informational guide," April 2019. [Online]. Available: https://www.fhwa.dot.gov/publications/research/safety/04091/07.cfm

[30] M. Quigley *et al.*, "ROS: an open-source robot operating system," in *ICRA workshop on open source software*. Kobe, Japan, 2009.

[31] M. Treiber, A. Hennecke, and D. Helbing, "Congested traffic states in empirical observations and microscopic simulations," *Physical review E*, vol. 62, no. 2, p. 1805, 2000.