Proceedings of the Twenty-Third International Conference on Automated Planning and Scheduling

Smart Urban Signal Networks: Initial Application of the SURTRAC Adaptive Traffic Signal Control System

Stephen F. Smith, Gregory J. Barlow, Xiao-Feng Xie, and **Zachary B. Rubinstein**
The Robotics Institute, Carnegie Mellon University
5000 Forbes Avenue, Pittsburgh, PA 15213
{sfs,gjb,xfxie,zbr}@cs.cmu.edu

Abstract

In this paper, we describe a pilot implementation and field test of a recently developed approach to real-time adaptive traffic signal control. The pilot system, called SURTRAC (Scalable Urban Traffic Control), follows the perspective of recent work in multi-agent planning and implements a decentralized, schedule-driven approach to traffic signal control. Under this approach, each intersection independently (and asynchronously) computes a schedule that optimizes the flow of currently approaching traffic through that intersection, and uses this schedule to decide when to switch green phases. The traffic outflows projected by this schedule are then communicated to the intersection's downstream neighbors, to increase visibility of vehicles entering their respective planning horizons. This process is repeated as frequently as once per second in rolling horizon fashion, to provide real-time responsiveness to changing traffic conditions and coordinated signal network behavior. After summarizing this basic approach to adaptive traffic signal control and the domain challenges it is intended to address, we describe the pilot implementation of SURTRAC and its application to a nine-intersection road network in Pittsburgh, Pennsylvania. Both the SURTRAC architecture for interfacing with the detection equipment, hardware controller and communication network at a given intersection and the extensions required to account for unreliable sensor data are discussed. Finally, we present the results of a pilot test of the system, where SURTRAC is seen to achieve major reductions in travel times and vehicle emissions over pre-existing signal timings.

Introduction

Traffic congestion in urban road networks is a substantial problem, resulting in significant costs for drivers through wasted time and fuel, detrimental impact to the environment due to increased vehicle emissions, and increased needs for infrastructure upgrades (Schrank, Lomax, and Turner 2011). Poorly timed traffic signals are one of the largest recurring sources of traffic congestion (Chin et al. 2004). Even when signals have been recently retimed, the inability to respond to current traffic patterns can cause pockets of congestion that lead to larger traffic jams. Inefficiencies in traffic signal timing stem from poor allocation of green time, inability to respond to real-time conditions, and poor coordination between adjacent intersections. It is generally recognized that traffic signal improvements offer the biggest payoff for reducing congestion and increasing the effective capacity of existing road networks, and that adaptive traffic signal control systems hold the most promise for improvement (Papageorgiou et al. 2003; Stevanovic 2010).

This paper investigates the potential of a recently developed approach to real-time adaptive traffic signal control (Xie et al. 2012b; Xie, Smith, and Barlow 2012) in an actual urban traffic control setting. The approach, which is realized in a system called SURTRAC (Scalable Urban Traffic Control), combines perspectives and concepts from recent work in the field of multi-agent planning (Lesser, Decker, and Wagner 2004; Smith et al. 2007) with prior work in traffic theory (Barriere, Farges, and Henry 1986; Mirchandani and Head 2001; Shelby 2004), and formulates traffic signal control as a decentralized, *schedule-driven* process. In brief, each intersection independently computes a schedule for servicing all currently approaching vehicles. This schedule is used locally to determine when to switch green phases and is recomputed in rolling horizon fashion every few seconds. Network level coordination is achieved through the exchange of schedule information between neighboring intersections. At each decision point, the scheduled output flows from an intersection's immediate upstream neighbors are combined with directly sensed traffic inflows to provide an expanded look-ahead planning horizon. Additional coordination mechanisms are layered over this basic protocol to cope with specific mis-coordinated situations.

To demonstrate the potential of the SURTRAC approach, a pilot implementation was installed at a nine-intersection road network in the East Liberty neighborhood of Pittsburgh, Pennsylvania, and a performance comparison was carried out with the existing traffic signal control scheme at this pilot site. Before the pilot test, control of these nine intersections was accomplished using coordinated timing plans that were optimized offline for AM and PM rush periods and simple actuated control (free mode) during non-rush periods. A series of "before" and "after" drive through runs were performed for each of 4 different periods of the day (AM rush, Mid Day, PM rush and Evening) and relevant performance metrics (travel time, speed, number of stops, wait time, fuel consumption, and emissions) were computed for each test condition. Across all metrics studied, SURTRAC was seen to produce significant performance improvement, ranging from 20%-40% overall.

The remainder of the paper is organized as follows. We first define the adaptive traffic signal control problem and discuss previous approaches. Next, the decentralized, schedule-driven approach to real-time, adaptive signal control that underlies the SURTRAC system is summarized, followed by a description of the pilot SURTRAC implementation. Mechanisms necessary to compensate for sensing uncertainty are then discussed. Finally, the pilot test and evaluation of SURTRAC is presented and conclusions are drawn.

The Adaptive Traffic Signal Control Problem

Operation of the traffic signals at a given intersection is typically governed by a *signal timing plan*. A timing plan assumes that compatible vehicle movement paths through the intersection (e.g., north and south lanes) have been grouped into movement *phases*. It specifies the sequence in which phases should be activated (turned green) and the duration of each green phase. The duration of each phase is subject to minimum and maximum constraints to ensure fairness and the transition from one phase to the next must obey safety constraints (fixed-length yellow and all red periods). A timing plan is graphically depicted in Figure 1.

Conventional signal systems use pre-programmed timing plans to control traffic signal operation. Fixed timings allocate the same amount of time to a phase during each cycle, while actuated signals use vehicle detectors to allow simple, minor variations in phase durations within the constraints of the timing plan (e.g., the green may be indefinitely allocated to the dominant traffic flow, only shifting to a cross street phase when a waiting vehicle is detected). Different timing plans may be invoked at different periods of the day (e.g., during rush and off-peak periods), and the timing plans can impose additional constraints to coordinate the actions of signals at different intersections. The crucial distinction is that timing and coordination plans are computed off-line, based on expected traffic conditions. Adaptive signal systems, in contrast, sense the actual traffic flows approaching intersections and continually adjust intersection timing plans to match current conditions.

The design of adaptive signal systems has received considerable attention over the years. With respect to the control of traffic signal networks (the principal focus of this paper), most practical success has been achieved using more centralized approaches (e.g., SCOOT (Robertson and Bretherton 1991), ACS-Lite (Luyanda et al. 2003), and SCATS (Sims and Dobinson 1980)) and there are several examples of large-scale deployments. At the same time, the tradeoffs required to enable centralized coordination can be limiting. In some cases (e.g., ACS-Lite and ACDSS (Xin et al. 2010)), systems are designed to effect changes to traffic signal timings on the order of minutes based on average flow predictions, which limits how quickly and effectively a system can respond to locally changing traffic patterns.

To achieve greater real-time responsiveness, other work has focused on techniques for computing intersection timing plans that optimize actual traffic flows (e.g., ALLONS-D (Porche and Lafortune 1999), PRODYN (Barriere, Farges, and Henry 1986), OPAC (Gartner 1983), RHODES (Mirchandani and Head 2001), CRONOS (Boillot, Midenet, and Pierrelee 2006), and others (Shelby 2001; 2004)). This class of online planning approaches, sometimes referred to as model-based optimization, extends rather naturally to decentralized forms of network control (e.g., (Barriere, Farges, and Henry 1986; Mirchandani and Head 2001; Shelby 2001)), but often significant tradeoffs have to be made to achieve computational tractibility for real-time operation. One primary contribution of the SURTRAC approach is a novel reformulation of this optimization problem as a single machine scheduling problem, which allows SURTRAC to compute near-optimal intersection timing plans on a second by second basis. Other recent work (Richter, Aberdeen, and Yu 2006; Cai, Wong, and Heydecker 2009) has explored use of reinforcement learning to find policies that map current observations to signal control actions. However these methods are slow to converge and difficult to apply in real-time if traffic flows are changing frequently.

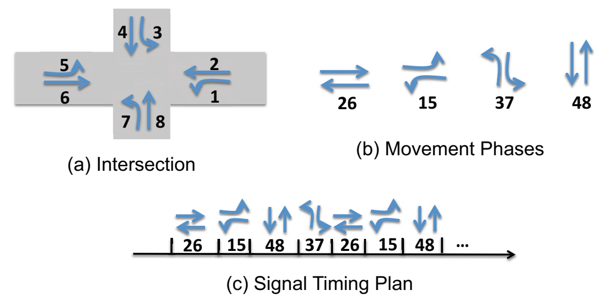


Figure 1: (a) An intersection, (b) possible movement phases and (c) a sample signal timing plan (adapted from (Sen and Head 1997))

If the dominant flow(s) in a road network are relatively static and can be identified in advance (e.g., an arterial roadway), then this global information can be leveraged to coordinate the independent, real-time actions of individual intersections. The InSync system (Chandra et al. 2011), for example, establishes synchronized periods during which a given intersection must service the dominant flow (in effect specifying a fixed global timing plan for corridor traffic), and then each individual intersection is free to independently manage side street traffic outside of those periods. The SURTRAC approach, alternatively, is aimed at urban (grid-like) road networks, where there are multiple (typically competing) dominant flows that shift dynamically and sometimes non-recurrently through the day. Urban networks also often have tightly spaced intersections requiring tight coordination. The combination of competing dominant flows and densely spaced intersections presents a challenge for all adaptive systems.

Schedule-Driven Traffic Control

As indicated earlier, the traffic signal control problem is formulated in SURTRAC as a decentralized, schedule-driven process (Xie et al. 2012b; Xie, Smith, and Barlow 2012). At the lowest level, each intersection is controlled independently by a local scheduler, which maintains a phase schedule that minimizes the total delay for vehicles traveling through the intersection and continually makes decisions to update the schedule according to a rolling horizon. The intersection scheduler communicates outflow information implied by its current schedule to its immediate neighbors, to extend visibility of incoming traffic and achieve network level coordination.

The ability to consider short-term (second-by-second) variability of traffic flows at the individual intersection level is made tractable by a novel formulation of online planning as a *single machine* scheduling problem (Xie et al. 2012b). Key to this formulation is an aggregate representation of traffic flows as sequences of clusters (corresponding specifically to queues and platoons) over a limited prediction horizon. These cluster sequences preserve the non-uniform nature of real-time flows while providing a more efficient *scheduling search space*.

More precisely, the input to SURTRAC is an ordered sequence of $\langle vehicle, arrival\,time, departure\,time\rangle$ triples for each approaching road segment, which constitutes the current projection of approaching and queued traffic that is sensed by the intersection's detectors. This input is then aggregated into sequences of vehicle clusters (queues, platoons) with associated arrival and departure times, based on relative vehicle proximity. Interpreting each cluster as an input *job* then, the scheduling problem is to construct an optimal sequence of all jobs that preserves the ordering of jobs along each inflow and treats all jobs as non-preemptable. A given sequence dictates the order in which jobs will pass through the intersection and can be associated with an expected phase schedule that fully clears the ordered jobs in the shortest possible time, subject to basic timing and safety constraints. The optimal sequence (schedule) is the one that incurs minimal delay for all vehicles.

A forward recursion, dynamic programming process is used to solve this scheduling problem. From a constructive view, the state space can be organized as a decision tree: each schedule is built from the root node, and a new job is added to the end of the (partial) sequence at each stage. At the same depth in the tree, states are grouped if they designate the same jobs (with different orders) and the same last job (referring to the same last phase). A greedy state elimination strategy is then applied to each group, where only the state reached with the minimum delay is kept while all other states are eliminated. Thus, most branches are pruned during early stages. The total process has at most $|I|^2 \cdot \prod_{i=1}^{|I|}(J_i+1)$ state updates (where $J_i \geq 0$ is the number of jobs in the ith inflow and $|I|$ is the number of phases), and each state update can be executed in constant time. The time complexity is polynomial in the prediction horizon H_P, since $|I|$ is limited for each intersection in the real world. A nice property is that J_i is insensitive to the granularity of time resolution in H_P (Xie et al. 2012b). In practice, $J_i \ll H_P$. For minor inflows (e.g., protected left turns) that are only subject to queue clearance, $J_i \in \{0, 1\}$.

This approach to intersection control can be contrasted with previous research in model-based optimization methods (Gartner 1983; Barriere, Farges, and Henry 1986; Porche and Lafortune 1999; Mirchandani and Head 2001; Shelby 2001; Papageorgiou et al. 2003; Shelby 2004; Boillot, Midenet, and Pierrelee 2006). Under the standard model-based optimization formulation, the primary state space is defined differently - it contains all possible signal sequences over a discretized *optimization horizon* (H_O), where H_O is sufficiently long for clearing all vehicles in the *prediction horizon* (H_P), and time resolution is sufficiently fine to avoid any significant rounding errors in the temporal values of timing constraints and model parameters (e.g., start-up lost time). However, the size of this search space is exponential in the number of time steps in H_O. To be real-time tractable, all model-based optimization methods are approximated through space reduction and state elimination. Several simple space reduction constraints have been imposed such as a coarser time resolution (Porche and Lafortune 1999), a short optimization horizon (e.g., using H_P as H_O (Barriere, Farges, and Henry 1986; Mirchandani and Head 2001)), or a smaller number of phase switches (Gartner 1983). The use of variable time steps has also been attempted (Shelby 2001). For further state elimination, model-based optimization methods group "equivalent" states when they are in the same time step.

In SURTRAC, alternatively, the scheduling search space provides the approximation - it is a subspace that is tailored to the intersection control problem and can be efficiently searched. It is not possible to say definitively how far from optimal the solutions generated by the SURTRAC intersection scheduling procedure are. Although the abstract (single machine problem) formulation is solved optimally, this abstraction can omit optimal solutions to the original intersection control problem. In simulation experiments, however, use of this abstract formulation has proved quite effective. In (Xie et al. 2012b), this intersection scheduling procedure was shown to reduce total delay in comparison to COP (Sen and Head 1997)) which according to (Shelby 2004) still represents the state of the art in intersection flow optimization, with 2-4 orders of magnitude speedup. In the pilot test described later in the paper, H_P was set at 120 seconds with 0.1-second time precision (note that H_O would be much longer).

When operating within an urban road network, any local intersection control strategy might be susceptible to myopic decisions that look good locally but not globally. To reduce this possibility, network level coordination mechanisms are layered over SURTRAC's basic schedule-driven intersection control strategy.

As a basic protocol, referred to as *optimistic, non-local observation* (Xie, Smith, and Barlow 2012), each intersection sends its projected *outflows* to its direct neighbors. Given an intersection schedule, projected outflows to all exit roads are derived from models of current inflows and recent turning proportions at the intersection. Intuitively, the outflows of an intersection's upstream neighbors become its predicted non-local inflows. The joint local and non-local inflows essentially increase the look-ahead horizon of an intersection, and due to a chaining effect, a sufficiently long horizon extension can incorporate non-local impacts from indirect upstream neighbors. The optimistic assumption that is made is that direct and indirect neighbors are trying to follow their schedules. Normally, the optimization capability of the base intersection control approach results in schedules that are quite stable, given a sufficient number of jobs in the local observation and sufficiently large jobs (platoons) in the local and non-local observation. Minor changes in the schedules of neighbors can also often be absorbed, if there is sufficient slack time between successive jobs. This basic coordination protocol is similar to that previously utilized in (Barriere, Farges, and Henry 1986; Shelby 2001). One difference is that we assume asynchronous coordination, where each intersection operates independently and simply incorporates new projected non-local inflow information whenever it is received. Given this assumption, temporary communication failures can be mostly ignored. Compared to the immediate observations exchanged in distributed W-learning (Dusparic and Cahill 2012), projected outflows contain more predictive information.

However, circumstances can and do cause schedules to change, in which case mis-coordination can occur, especially for intersections that are very close together. To this end, additional coordination mechanisms are incorporated into SURTRAC for handling specific nontrivial mis-coordination situations. One common inefficiency is caused by spillback which, due to insufficient capacity on a road segment, can block the progress of traffic flow from an upstream intersection if the segment is short and/or the traffic demand is high. The basic coordination protocol is augmented with a *spillback prevention* mechanism that acts to detect and prevent unnecessary spillback in advance of its occurrence by accelerating phase changes. If spillback occurs, the basic protocol enables estimation of queue length across intersections and facilitates efficient clearance of highly congested links if downstream intersections allow. Another source of mis-coordination is *nervousness*, the tendency for the schedules of coordinating neighbors to oscillate due to small inconsistencies. This type of miscoordination is handled by a second mechanism. Further description of these coordination mechanisms can be found in (Xie, Smith, and Barlow 2012).

System Architecture

SURTRAC (Scalable Urban Traffic Control) implements schedule-driven traffic control as part of a flexible signal control system that is designed to be easily integrated with controller and sensor hardware from any vendor. True to the schedule-driven traffic control model, SURTRAC is organized as a completely decentralized multi-agent system. Each intersection is controlled by an agent running on an embedded computer located in the traffic cabinet for the intersection. The agent for each intersection manages the control of the traffic signal and all of the vehicle detectors located at that intersection.

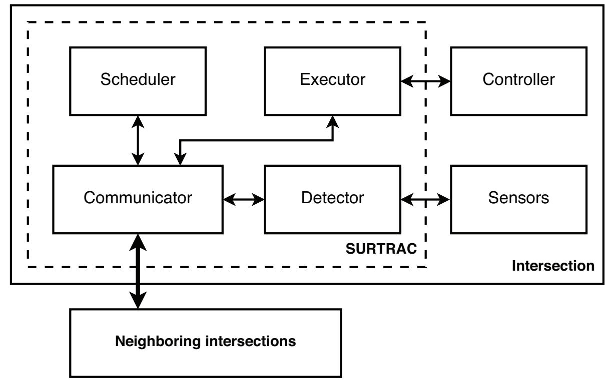


Figure 2: SURTRAC system diagram

The agent for each intersection is modeled as a multi-threaded service-oriented architecture, shown in Figure 2. The Communicator service handles the routing of all information between different services as well as information sharing between intersections. The Detector service interfaces with all vehicle sensors, processing real-time data into messages that can be used by local and remote services. The Executor service manages the interface with the traffic signal controller, reading status information about the state of the traffic signals and controlling the duration and sequence of phases. The Scheduler service uses data from the other services to create schedules that allocate green time at the intersection.

Communicator

SURTRAC deployments rely fundamentally on connectivity throughout the road network, but by design it is only necessary for an intersection to be able to communicate with direct neighbors. By keeping communication strictly between neighbors, the SURTRAC system can scale to very large signal networks. All communication is asynchronous and robust to temporary network failure.

As shown in Figure 2, all communication is routed through the Communicator at a given intersection. Most messages are routed locally. All data are encoded as messages of pre-defined types, and can be addressed to any intersection. By using standard types, Executor and Detector service modules that integrate hardware from different vendors can provide the same information to the rest of the system. Formally, each message can be described as a tuple $\langle type, time, orig, dest, source, data\rangle$ of the message type, the time that the message was generated, the intersection where the message originated, a list of destination intersections for the message, the service or detector that created the message, and the content of the message as a JSON (JavaScript Object Notation)-encoded string.

Detector

The Detector service manages the interfaces with all sensors located at an intersection. For each sensor, real-time data must be retrieved, encoded into a message, and then sent to the local Scheduler service. If the sensor functions as an

Figure 3: The placement of detectors in a typical installation. Exit detectors typically function as advance detectors for neighboring intersections downstream.

advance detector for a neighboring intersection, the message must also be sent to the remote Scheduler.

A wide variety of vehicle sensors are currently used in traffic systems, including induction loops, video detection, and radar systems. The pilot deployment of SURTRAC described below uses Traficon video detection, but other types of detectors are substitutable. Figure 3 shows the placement of detectors at a typical intersection. For each exit link, a group of exit detectors is placed near the intersection. For each entry link, a group of stop-bar detectors is placed near the intersection, and a group of advance detectors is placed far away from the intersection. To maximize the look-ahead horizon, the exit detectors of an upstream intersection are used as the advance detectors for the downstream intersection if possible. For intersections on the boundary of the system, advance detectors usually must be located closer to the intersection.

At each detection location, two types of data are reported: traffic counts and occupancy time of vehicles. For the video detection in the pilot system, these two measures are generated by separate detection zones: a data zone and a presence zone. Data zones are small enough to detect gaps between vehicles during congested conditions, whereas presence zones are large enough to prevent missing vehicle occupancy information. As a vehicle passes a data zone, a message is generated and routed through the Communicator. Occupancy for all presence zones is sensed every 0.1 seconds and aggregated every second, encoded into messages, and sent the same way.

Executor

To control the traffic signals at an intersection, SURTRAC interfaces with a traffic signal controller, which normally uses some combination of timing plans and simple actuation to allocate green time for the intersection. When the SURTRAC system is active, the controller continues to enforce maximum and minimum phase durations, transitions between phases, and other safety constraints, but SURTRAC adaptively allocates the green time for the intersection. SURTRAC places the controller into *free mode*, which normally uses vehicle calls (service requests) from detectors for simple actuated control. When the SURTRAC system is active, the controller is configured to only accept calls from SURTRAC, similar to some other real-time adaptive systems. Phase maximums are extended to allow longer phases, and the passage (gap) time that allows the controller to change phases is shortened to allow for quicker transitions. Such configuration changes are written at the time the SURTRAC system is activated to automate the startup process. The new configuration is placed in a separate memory page within the controller so that the intersection can easily revert to its original state.

When the Executor is active, it communicates frequently with the controller, polling for state and setting vehicle calls multiple times per second. Transitions in the controller state—e.g. the beginning or end of a phase—are relayed to the Scheduler. The Executor follows the schedule provided by the Scheduler, sending calls to continue in the current phase until the scheduled phase end time, at which time the Executor sets calls for the next desired phase. When the Scheduler updates the schedule, it may extend the current phase by any amount greater than or equal to the minimum extension (a system parameter). The minimum extension time for the pilot was set to one second, so that the schedule could be adjusted as frequently as once per second. Although this setting was the same for all intersections in the pilot, it isn't necessary, since coordination is asynchronous. When the current phase is extended, the Executor notifies the Scheduler of the upcoming *decision point* in the schedule—the point by which a subsequent update to extend the phase must be received. For small minimum extension times, the time for the Scheduler to make a decision may be extremely short (less than half a second), and schedules may arrive to the Executor too late to extend the current phase. To protect against such "dropped" schedules, the Executor uses default phase durations calculated by the Scheduler. The Executor will only end a phase earlier than the default duration if the Scheduler chooses to terminate the phase. The Executor may also fall back to these phase durations in the case of prolonged sensor or network failure.

Scheduler

The Scheduler service implements the schedule-driven traffic control approach described earlier. It continuously receives real-time phase and detection data and scheduled upstream outflows, builds its abstract model of the traffic approaching the intersection, and constructs a new phase schedule. Once a new schedule has been constructed, the leading portion is sent to the Executor for controlling the traffic signal, and the scheduled outflows are sent out to downstream intersections. Some basic failure mitigation mechanisms are included to enhance reliability in the real world. These mechanisms only need to work locally due to the decentralized nature of the system.

If the network connection to a neighboring intersection fails, the local intersection may not be able to receive data from advance detectors or projected outflows. If the downtime is short (e.g., < 20 seconds), the local scheduler can still work properly using recent data. However, a longer failure might cause the link to be severely under-serviced since eventually no new vehicle information is received. Disconnections can be discovered quickly, since occupancy data are sent every second. For time periods with missing data, a moving average forecast is added using the current link flow rate at the stop-bar detectors. Thus, the scheduler operates using hybrid information when look-ahead information is only available for some links. The performance of the intersection might be degraded due to the loss of predicted non-local information on disconnected links, but its other neighbors will still receive good non-local information. Thus, short communication failures will not have major effects on the overall system performance.

Coping With Real World Uncertainty

One issue that was quickly recognized as the SURTRAC system was fielded for the pilot test was the need to cope with uncertainty in the operating environment. One primary source of uncertainty is sensing error (Rhodes et al. 2005). Vehicles turning too sharply at an intersection can be missed by detection zones, large vehicles (e.g., trucks, buses) sometimes trigger detection zones covering multiple lanes, reflections from the road surface in inclement weather can be misinterpreted by video processing software, and so on. Disruptions to normal assumptions about traffic flows constitute a second source of uncertainty. A stopped vehicle can give the false impression of a queue that needs to be serviced, or alternatively (e.g., in the case of a one lane roadway) can be blocking a queue from being serviced despite the fact that green time is being allocated. Both types of uncertainty work against SURTRAC's attempt to optimize the flow of traffic through the signal network.

With regard to the scheduling model, the main impact of uncertainty is to lessen the accuracy of queue length estimation, which in turn misrepresents the durations of the most pressing jobs to be scheduled. Over time, queue length is dynamically maintained by a cumulative input-output technique (Sharma, Bullock, and Bonneson 2007), using departure and arrival counts obtained from stopbar and advance detectors. However, the predicted queue length (q) is a hidden state, and detection errors can cause either over-estimation or under-estimation of q. Over-estimation of q can be seen as equivalent to insertion of buffer time (Herroelen and Leus 2005), which will naturally be taken advantage of by a continual, rolling horizon scheduling approach such as SURTRAC's. However, under-estimation should be avoided, since significant delay might occur from long residual queues, and these residual queues will not be visible in subsequent scheduling cycles before they are fully cleared. The situation can become significantly worse if the queue starts to spill back to upstream intersections.

To address the problem of queue under-estimation in the pilot implementation, we adopted a set of simple heuristic strategies:

- Use of *link arrival/departure ratios* ($ADRatio$) - The $ADRatio$ of a road segment is used to account for detection inaccuracy by hypothesizing that a road may have mid-block entrances or exits that contribute hidden flows that are not covered by any detectors. As in (Sharma, Bullock, and Bonneson 2007), we assume that the group of stop-bar detectors will yield an accurate estimation of departing vehicles. If $ADRatio < 1$, then some arriving vehicles have been missed, and the current counts of queued and arriving vehicles are under-estimated. Thus, when vehicles are detected at the advance detectors, the arriving vehicles count is divided by $ADRatio$ to reclaim those missing vehicles and avoid under-estimation.
- *Queue clearance management* - A second strategy utilizes "elasticity" and "tolerance" measures to more effectively manage queue clearance in the presence of uncertain disruptions. The "elasticity" measure assesses the queue clearing time t_{QC}—necessary for identifying the queue clearance state—using the unoccupied time at the stop-line. If t_{QC} is too small, a queue might be prematurely truncated. If t_{QC} is too large, green time is wasted. Thus, t_{QC} is defined as proportional to an elasticity ratio r_{QC}^{ela}, where r_{QC}^{ela} is a sigmoid function on the queue size q. A long queue size will have a large r_{QC}^{ela} and will be unlikely to be truncated, reducing the risk of leaving a long residual queue, while not wasting green time identifying a short queue. "Tolerance" is applied to avoid under-estimation if a long queue is unexpectedly truncated and becomes a residual queue (e.g., due to real-world uncertainty such as a mid-block bus stop or stop-bar miscounting). The current queue is stored as q', and is derived using the cumulative input-output technique in the same way as q. Then q' is retrieved as q for the following N_{TOL} scheduling cycles, where $N_{TOL} = 1$ is the default tolerance size.

Further details of these strategies and a simulation analysis of their performance can be found in (Xie et al. 2012a). Uncertainty management in this context continues to be an active area of our current research.

Pilot Test

To demonstrate the potential of SURTRAC, a nine-intersection pilot system was deployed in the East Liberty neighborhood of Pittsburgh, Pennsylvania. East Liberty has experienced enormous redevelopment in the past 10 years, drastically changing traffic patterns in the neighborhood. A large portion of a one-way ring road called Penn Circle was recently converted to two-way traffic during the development of a new department store. The road network in this portion of East Liberty is now a triangular grid, with three major roads—Penn Avenue, Highland Avenue, and Penn Circle—crossing each other. Already high traffic volumes are increasing with ongoing development. Competing traffic flows shift throughout the day, making coordination difficult.

The pilot site, shown in Figure 4, consists of nine intersections. Road lengths between intersections range from 90 to 500 feet with an average of 272 feet, requiring tight coordination between intersections. Equipment at eight of these intersections—including six on Penn Circle—was updated

Figure 4: Map of the nine intersection pilot site in the East Liberty neighborhood of Pittsburgh, Pennsylvania.

as part of recent redevelopment. Each of these new intersections was equipped with Traficon cameras pointing in all inflow directions, and all eight were inter-connected with fiber-optic cable, providing the sensing equipment and networking infrastructure needed to deploy the SURTRAC system. The ninth intersection is located at the center of East Liberty, allowing SURTRAC to fully capture the grid network which has been returned to the area. As part of the pilot, this intersection was upgraded with cameras and joined to the existing network using Encom radios.

Prior to the introduction of SURTRAC at the pilot test site, the 8 networked intersections were controlled with coordinated-actuated timing plans during morning and afternoon rush periods and with simple actuated (free mode) control during the remainder of the day. These coordinated-actuated timing plans were generated using SYNCHO, a state-of-the-practice commercial package for offline timing plan optimization, and installed in early 2011. So arguably, this portion of the signal network was equipped with the most modern form of conventional signal control. The ninth intersection was previously controlled by a single uncoordinated, pre-timed plan.

Evaluation

To evaluate the performance potential of the SURTRAC system, a series of timed, drive-through runs of the pilot test site were conducted for each of two control scenarios. More specifically, the 12 highest volume routes through the pilot test site were identified and a drive through run involved a traversal of all 12 of these routes, shown in Figure 5(a). These routes included both directions following Penn Avenue, Highland Avenue, and Penn Circle, 3 left and 2 right turns at the intersection of Penn Avenue and Penn Circle, and the route from Broad Avenue turning left onto Penn Circle. A series of drive through runs were performed while the intersections were being controlled by the current combination of coordinated-actuated time-of-day plans and actuated free mode ("before" scenario). Then a second series of drive through runs were performed while the intersections were being controlled by the SURTRAC adaptive strategy ("after" scenario).

Travel data for a given run was collected through use of an iPhone app called GPS Kit Pro, which generates a GPS trace for an entire run of 12 routes. An example is shown in Figure 5(b). These data were then post-processed to extract only those subsequences corresponding to travel along the 12 evaluation routes, and evaluation metrics were computed from these subsequences.

For each control scenario, three evaluation runs were conducted for each of four periods of the day: AM rush (8-9 AM), Mid-day (12-1 PM), PM rush (4-6 PM), and Evening (6-7 PM). All 24 runs (12 for each scenario) were performed on weekdays other than Friday. Additionally, a fourth PM rush run was conducted for each scenario on a Friday to test this exceptionally high volume condition. All "before" runs were conducted in March 2012; all "after" runs were conducted in June 2012 (due to delays in obtaining a legal agreement to assume control of the signals). An analysis of traffic volume data for these two periods showed approximately 5% higher volumes in June, implying a slightly tougher challenge for SURTRAC (see (Smith et al. 2012) for details).

We computed the following set of performance metrics: travel time, speed, number of stops, wait time, and emissions. Travel time is normalized by canonical distances for each route to compensate for the differences in distance that arise due to GPS sampling variation in the locations of start and end points for a route. Emissions of carbon dioxide (CO_2), hydrocarbons, carbon monoxide (CO), nitrogen oxides (NO_x), and volatile organic compounds (VOC) are calculated as a function of fuel consumption[1]. When combining data from individual routes to produce aggregate performance results, the relative volumes along different routes were used to determine weights, which may be found in (Smith et al. 2012) along with further details on the evaluation.

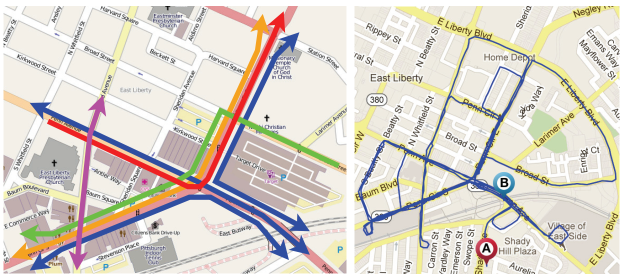
Results

Table 1 summarizes the performance improvement achieved by the SURTRAC adaptive traffic control system over the pre-existing traffic control scheme at the pilot test site. The levels of improvement are substantial across all performance metrics computed and for all periods of the day. Overall improvements are computed as a weighted average, using relative traffic volumes observed during each period (given in Table 1). With respect to efficiency of traffic flows, average travel times through the pilot site are reduced by over 25%, average vehicle speed is increased by 34%, the number of stops is reduced by over 31%, and the average wait time is reduced by over 40%. From the perspective of improving the quality of the air, which was the motivation behind the funding for this project, overall emissions are reduced by 21%.

[1]The emissions numbers reported here are computed based on the fuel consumption model given in (Wallace et al. 1984)—the model used by the metropolitan planning organization for the region—and EPA and EIA data. See (Smith et al. 2012) for details.

Percent improvement	Average Vehicles	Travel time	Speed	Number of Stops	Wait Time	Emissions
AM rush	5,228	30.11%	33.78%	29.14%	47.78%	23.83%
Mid Day	8,007	32.83%	48.55%	52.58%	49.82%	29.00%
PM rush	9,548	22.65%	27.45%	8.89%	35.60%	18.41%
Evening	7,157	17.52%	27.81%	34.97%	27.56%	14.01%
Overall	29,940	25.79%	34.02%	31.34%	40.64%	21.48%

Table 1: Summary of pilot test results



(a) (b)

Figure 5: Evaluations were performed by recording GPS traces for a series of drive-through runs of the pilot test site. Each run contained 12 routes covering all major traffic movements (a). GPS traces were post-processed to evaluate only the fixed routes through the pilot site (b).

Examining the results by period of day, the largest improvement is observed during the Mid Day period. This is explainable by the relatively high volume of traffic and the relative inability of the free mode configuration to adequately cope. During this period, performance improvement was observed with respect to all measures for eleven of the twelve routes evaluated. During the AM Rush, PM Rush and Evening periods, performance improvement was observed for eight of the twelve routes. Three of the four routes whose performance deteriorated during the AM Rush period involved traffic moving along Penn Circle, suggesting an unbalanced bias in the pre-existing SYNCHRO generated timing plan. In the highest volume PM Rush period, SURTRAC exhibited quite robust performance; of the four routes whose performance deteriorated, two performed worse on only a single metric (number of stops) and a third had lesser values for just two metrics (average speed and number of stops). Please refer to (Smith et al. 2012) for further details and expanded discussion.

To quantify the absolute impact of SURTRAC on emissions, it is necessary to once again consider traffic volumes through the pilot test site. Given an average of 29,940 vehicles per day, Table 2 indicates projected savings in fuel and pollutant emissions. A daily savings in fuel of 247 gallons is estimated, which implies a daily reduction in emissions of 2.253 metric tonnes. Given this, an annual reduction in emissions of 588 metric tonnes is expected if SURTRAC continues to run the nine intersections at the pilot test site.

Emissions	Daily (kg)	Annual (tonnes)
Fuel Consumption	247 gal.	64,580 gal.
Carbon Dioxide (CO2)	2213.85	577.82
Carbon Monoxide (CO)	17.30	4.51
Nitrogen Oxides (NOx)	3.37	0.88
Volatile Organic Compounds (VOC)	4.01	1.05
Hydrocarbons	14.90	3.89
Total Emissions	2253.42	588.14

Table 2: Projected emissions savings

Conclusion

The pilot test results convincingly demonstrate the effectiveness and potential of decentralized, adaptive traffic signal control in urban road networks. In comparison to the current conventional approach to traffic control in use at the pilot test site, which involves a combination of coordinated timing plans during rush periods and actuated free mode during non-rush periods, the SURTRAC adaptive signal control system improved traffic flow efficiency through the pilot site by 25%–40% (depending on the metric considered) and reduced emissions by over 20%.

Many current approaches to adaptive traffic signal control tend to either aggregate sensed traffic flow data and coordinate network control centrally (which limits real-time responsiveness) or drive local intersection control with static, pre-computed global coordination plans. These approaches have proven most effective in arterial settings, where there is a single dominant traffic flow and traffic from side streets must be efficiently integrated. The SURTRAC system design, in contrast, aims specifically at urban road networks, where there are multiple, competing traffic flows that dynamically shift through the day. By controlling each intersection locally, responsiveness to real-time traffic conditions is maximized, and by communicating planned outflows to neighboring intersections larger corridor flows can be established on demand to match actual traffic flow volumes. Since the system operates in a totally decentralized manner, it is easily extended to incorporate additional intersections and inherently scalable to road networks of arbitrary size.

SURTRAC has been operating continuously at the pilot site since June 2012 and steps are underway to transfer operation of the system to the City of Pittsburgh. Improvements to SURTRAC are ongoing, including dynamic phase sequencing, which was not allowed as part of the pilot, improved service times for pedestrian calls, and bus detection and prioritization. An expansion of the pilot site to include nine more intersections is currently in process, and further expansions are in the planning stage.

Acknowledgements

This research was supported by the Traffic21 research initiative and the Robotics Institute at Carnegie Mellon, the Heinz Endowments, the Hillman Foundation, and the R.K. Mellon Foundation. The authors would like to thank City of Pittsburgh traffic engineer Amanda Purcell and the Department of Public Works, Farid Semmahi of Traficon USA, Traffic Control Products, Inc., Gabriel Somlo, and Torstein Stromme.

References

Barriere, J. F.; Farges, J. L.; and Henry, J. J. 1986. Decentralization vs hierarchy in optimal traffic control. In *IFAC Control in Transportation Systems*.

Boillot, F.; Midenet, S.; and Pierrelee, J. 2006. The real-time urban traffic control system CRONOS: Algorithm and experiments. *Transportation Research Part C: Emerging Technologies* 14(1):18–38.

Cai, C.; Wong, C.; and Heydecker, B. 2009. Adaptive traffic signal control using approximate dynamic programming. *Transportation Research Part C: Emerging Technologies* 17(5):456–474.

Chandra, R. J.; Bley, J. W.; Penrod, S. S.; and Parker, A. S. 2011. Adaptive control systems and methods. *U.S. Patent* (8,050,854).

Chin, S. M.; Franzese, O.; Greene, D. L.; Hwang, H.; and Gibson, R. C. 2004. Temporary losses of highway capacity and impacts on performance: Phase 2. Technical Report ORNL/TM-2004/209, Oak Ridge National Laboratory.

Dusparic, I., and Cahill, V. 2012. Autonomic multi-policy optimization in pervasive systems: Overview and evaluation. *ACM Transactions on Autonomous and Adaptive Systems* 7(1):Article No. 11.

Gartner, N. 1983. OPAC: A demand-responsive strategy for traffic signal control. *Transportation Research Record* 906:75–81.

Herroelen, W., and Leus, R. 2005. Project scheduling under uncertainty: Survey and research potentials. *European Journal of Operational Research* 165(2):289–306.

Lesser, V.; Decker, K.; and Wagner, T. 2004. Evolution of the GPGP/TAEMS domain-independent coordination framework. *Autonomous Agents and Multi-Agent Sytems* 9(1-2):87–143.

Luyanda, F.; Gettman, D.; Head, L.; Shelby, S.; Bullock, D.; and Mirchandani, P. 2003. ACS-Lite algorithmic architecture: Applying adaptive control system technology to closed-loop traffic signal control systems. *Transportation Research Record* 1856:175–184.

Mirchandani, P., and Head, L. 2001. A real-time traffic signal control system: Architecture, algorithms, and analysis. *Transportation Research Part C: Emerging Technologies* 9(6):415–432.

Papageorgiou, M.; Diakaki, C.; Dinopoulou, V.; Kotsialos, A.; and Wang, Y. 2003. Review of road traffic control strategies. *Proceedings of the IEEE* 91(12):2043–2067.

Porche, I., and Lafortune, S. 1999. Adaptive look-ahead optimization of traffic signals. *ITS Journal* 4(3-4):209–254.

Rhodes, A.; Bullock, D. M.; Sturdevant, J.; Clark, Z.; and Candey, D. G. 2005. Evaluation of the accuracy of stop bar video vehicle detection at signalized intersections. *Transportation Research Record* 1925:134–145.

Richter, S.; Aberdeen, D.; and Yu, J. 2006. Natural actor-critic for road traffic optimisation. In *Advances in Neural Information Processing Systems*, 1169–1176.

Robertson, D. I., and Bretherton, R. D. 1991. Optimizing networks of traffic signals in real time - the SCOOT method. *IEEE Transactions on Vehicular Technology* 40(1):11–15.

Schrank, D.; Lomax, T.; and Turner, S. 2011. Annual urban mobility report. Technical report, Texas Transportation Institute, Texas A&M University System, TX.

Sen, S., and Head, K. 1997. Controlled optimization of phases at an intersection. *Transportation Science* 31(1):5–17.

Sharma, A.; Bullock, D.; and Bonneson, J. 2007. Input-output and hybrid techniques for real-time prediction of delay and maximum queue length at signalized intersections. *Transportation Research Record* 2035:69–80.

Shelby, S. G. 2001. *Design and Evaluation of Real-Time Adaptive Traffic Signal Control Algorithms*. Ph.D. thesis, University of Arizona, Tucson, AZ.

Shelby, S. G. 2004. Single-intersection evaluation of real-time adaptive traffic signal control algorithms. *Transportation Research Record* 1867:183–192.

Sims, A., and Dobinson, K. 1980. The Sydney coordinated adaptive traffic (SCAT) system: Philosophy and benefits. *IEEE Transactions on Vehicular Technology* 29(2):130–137.

Smith, S. F.; Gallagher, A.; Zimmerman, T.; Barbulescu, L.; and Rubinstein, Z. 2007. Distributed management of flexible times schedules. In *6th International Conference on Autonomous Agents and Multiagent Systems*, 484–491.

Smith, S. F.; Barlow, G. J.; Xie, X.-F.; and Rubinstein, Z. B. 2012. Real-time adaptive traffic signal control for urban road networks: The east liberty pilot test. Technical Report CMU-RI-TR-12-20, The Robotics Institute, Carnegie Mellon University.

Stevanovic, A. 2010. Adaptive traffic control systems: Domestic and foreign state of practice. Technical Report NCHRP Synthesis 403, Transportation Research Board, National Research Council, Washington, DC.

Wallace, C. E.; Courage, K. G.; Reaves, D. P.; Schoene, G. W.; and Euler, G. W. 1984. TRANSYT-7F user's manual. Technical Report UF-TRC-U32 FP-06/07, Office of Traffic Operations, U.S. Department of Transportation.

Xie, X.-F.; Barlow, G. J.; Smith, S. F.; and Rubinstein, Z. B. 2012a. Accounting for real-world uncertainty in real-time adaptive traffic control. Technical report, The Robotics Institute, Carnegie Mellon University.

Xie, X.-F.; Smith, S. F.; Lu, L.; and Barlow, G. J. 2012b. Schedule-driven intersection control. *Transportation Research Part C: Emerging Technologies* 24:168–189.

Xie, X.-F.; Smith, S. F.; and Barlow, G. J. 2012. Schedule-driven coordination for real-time traffic network control. In *22nd International Conference on Automated Planning and Scheduling (ICAPS)*, 323–331.

Xin, W.; Chang, J.; Bertoli, B.; and Talas, M. 2010. Integrated adaptive traffic signal control with real-time decision support. In *Transportation Research Board Annual Meeting*, 21p.