SURTRAC: Scalable Urban Traffic Control

Stephen F. Smith
Research Professor
The Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213
sfs@cs.cmu.edu

Gregory J. Barlow*
Postdoctoral Fellow
The Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213
gjb@cmu.edu

Xiao-Feng Xie
Research Associate II
The Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213
xfxie@cs.cmu.edu

Zachary B. Rubinstein
Senior Systems Scientist
The Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213
zbr@cs.cmu.edu

* Corresponding author

6241 words + 3 figures + 2 tables = 7491 words

ABSTRACT

This paper defines and evaluates a pilot implementation of a recently developed approach to real-time, adaptive traffic signal control. The pilot system, which is called SURTRAC (Scalable Urban Traffic Control), integrates concepts from traffic control theory with recent work in the field of multi-agent planning and has several important distinguishing characteristics. First, to promote scalability and reliability, SURTRAC operates in a totally decentralized manner; each intersection independently and asynchronously allocates its green time, based on current incoming vehicle flows. Second, SURTRAC aims at managing urban (grid-like) road networks with multiple (competing) traffic flows; network-level coordination is accomplished by communicating projected outflows to downstream neighbors, which gives these intersections a more informed basis for locally balancing competing inflows while simultaneously promoting establishment of larger "green corridors". Third, SURTRAC truly operates in real-time; each intersection recomputes its allocation plan and re-communicates projected outflows as frequently as once per second in rolling horizon fashion, enabling both effective operation in tightly spaced signal networks and responsiveness to sudden changes in traffic conditions. After describing our basic approach to adaptive traffic signal control and the pilot implementation of SURTRAC, we present the results of a field test conducted on a nine-intersection road network in the East Liberty section of Pittsburgh, Pennsylvania. In this pilot test, SURTRAC is seen to achieve major reductions in travel times and vehicle emissions over pre-existing signal control.

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 (*1*). One of the largest recurring sources of traffic congestion are poorly timed traffic signals (*2*). 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 (*3, 4*).

This paper investigates the potential of a recently developed approach to real-time adaptive traffic signal control (*5, 6*) in an actual urban traffic control setting. The approach, which is realized in a system called SURTRAC (Scalable Urban Traffic Control), follows the perspective of recent work in the field of multi-agent planning (*7, 8*) 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 then 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.

The SURTRAC system design has three distinguishing characteristics, each of which offers important advantages. First, decision making in SURTRAC proceeds in a totally decentralized manner. Although more centralized approaches to adaptive traffic signal control have been effectively applied in many settings (e.g., SCOOT (*9*), BALANCE (*10*), ACS-lite (*11*), and SCATS (*12*)), they nonetheless require tradeoffs that can be limiting. Decentralized control of individual intersections enables maximum responsiveness to real-time traffic conditions. It promotes scalability by allowing incremental addition of intersections over time with minimal change to the existing adaptive network. There is also no centralized computational bottleneck and no single point of failure.

A second distinguishing characteristic of the SURTRAC design is its emphasis on real-time responsiveness to changing traffic conditions. Many adaptive traffic control systems (e.g., BALANCE, ACS-Lite and ACDSS(*13*)) 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. SURTRAC alternatively adopts the real-time perspective of prior model-based intersection control methods (e.g., ALLONS-D (*14*), PRODYN (*15*), OPAC (*16*), RHODES (*17*), CRONOS (*18*), and others (*19, 20*)) which attempt to compute intersection control plans that optimize actual traffic inflows. By using a novel reformulation of the optimization problem as a single machine scheduling problem, SURTRAC is able to compute near-optimal intersection control plans over an extended horizon on a second-by-second basis.

Finally, SURTRAC is designed to aim generally at managing urban (grid-like) road networks, where there are multiple (typically competing) dominant flows that shift dynamically through the day, and where specific dominant flows cannot be pre-specified (as in arterial or major crossroad applications). 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. SURTRAC determines dominant flows dynamically by continually communicating projected outflows to downstream neighbors (in similar fashion to the earlier PRODYN system (*15*)). This information gives each intersection a more informed basis for locally balancing competing inflows while simultaneously promoting establishment of larger "green corridors" when traffic flow circumstances warrant.

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-actuated 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 is seen to produce significant performance improvement, ranging from 20%-40% overall.

The remainder of the paper is organized as follows. First, the decentralized, schedule-driven approach to real-time, adaptive signal control that underlies the SURTRAC system is summarized. Then the architecture and configuration of the pilot SURTRAC implementation are described. The pilot study design is presented next, followed by a discussion of the results obtained. Finally, some conclusions are drawn.

SCHEDULE-DRIVEN TRAFFIC CONTROL

As indicated above, the traffic signal control problem is formulated in SURTRAC as a decentralized, schedule-driven process (*5, 6*). 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.

Effective consideration of the significance of 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 (*5*). Key to this formulation is an aggregate representation of traffic flows as sequences of clusters (corresponding specifically to anticipated queues (*21*) and platoons) in a limited prediction horizon. These cluster sequences preserve the non-uniform nature of real-time flows while providing a more efficient *scheduling search space*. Interpreting each cluster as an input *job*, 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 (*5*). 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 (*3, 14, 15, 16, 17, 18, 19, 20*). 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), as in ALLONS-D, and time resolution is sufficiently fine to avoid any significant rounding errors for temporal values in 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 methods are approximated through space reduction and state elimination. There are some simple space reduction settings used in other existing methods such as a coarser time resolution (*14*), a short optimization horizon (e.g., using H_P as H_O (*15, 17*)), or a smaller number of phase switches (*16*). The use of variable time steps has also been attempted (*19*). In our approach, the scheduling search space provides the approximation - it is a subspace that is tailored to the intersection control problem. For further state elimination, existing methods, e.g., RHODES and PRODYN, group "equivalent" states when they are in the same time step; our approach introduces a new state variable, called schedule status, to analogously identify states with the same remaining jobs (and hence vehicles). If an intersection has a sufficiently long look-ahead horizon, our intersection scheduling approach can efficiently find near optimal solutions. In (*5*), it has been shown to reduce delay in comparison to other state-of-the-art intersection control strategies (e.g., COP (*22*)) with 2-4 orders of magnitude speedup. In the pilot test described later in the paper, H_P had a value of 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*, each intersection sends its projected *outflows* to its direct neighbors (*6*). Given an intersection schedule, projected outflows to all exit roads are derived from models of current inflows and recent turning proportions at the intersection (*6*). 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 enough jobs in the local observation and large jobs (platoons) in the local and non-local observation. It is also the case that minor changes in the schedules of neighbors can often be absorbed, if there is sufficient slack time between successive jobs. As mentioned earlier, this basic protocol is essentially the same coordination mechanism previously utilized in PRODYN (*15, 19*). One difference is that we assume asynchronous coordination, so that temporary communication failures can be mostly ignored.

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, which is handled by a second mechanism. Futher description of these coordination mechanisms can be found in (*6*).

THE SURTRAC SYSTEM

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.

The agent for each intersection is modeled as a multi-threaded service-oriented architecture, shown in Figure 1. 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.

SURTRAC is designed to be integrated with any type of traffic signal controller or vehicle sensor. All information sharing is routed through the Communicator service, so different Executor and Detector service modules may be loaded depending on the hardware configuration at the intersection. Since information is passed using standard message types, service modules that integrate hardware from different vendors can provide the same information to the rest of the system. This design allows SURTRAC to work with many types of hardware as well as microscopic road traffic simulators for testing.

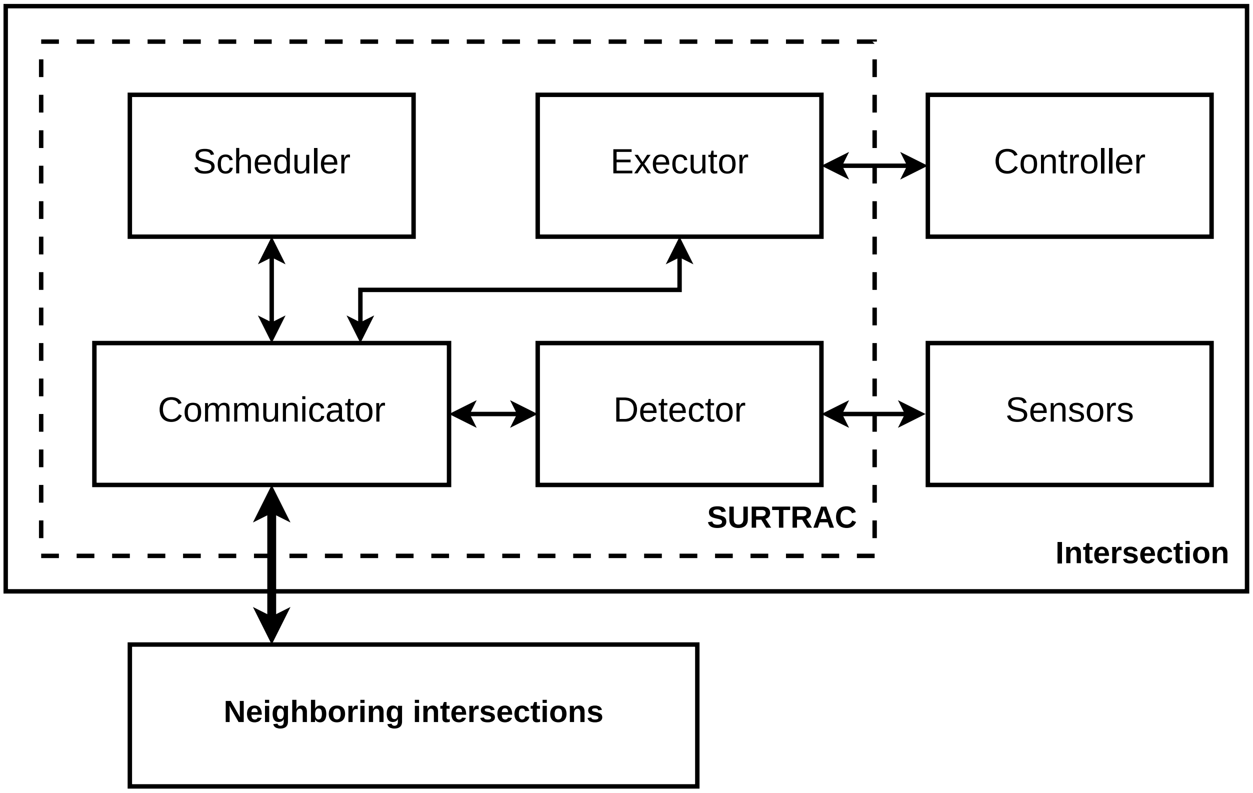


FIGURE 1 SURTRAC system diagram. The Detector interfaces with sensors located at the intersection, the Scheduler the Scheduler allocates green time based on incoming vehicle flows, the Executor interfaces with the controller to implement schedules generated by the Scheduler, and the Communicator routes messages locally and to/from neighboring intersections.

Communicator

The communication infrastructure of SURTRAC is designed to be flexible and general, allowing communication of many types of information. SURTRAC deployments must be networked, but 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 1, all communication is routed through the Communicator service for each intersection. Most messages are routed locally. All data are encoded as messages of predefined types, and can be addressed to any intersection. Formally, each message can be described as a tuple $< type, time, orig, dest, source, data >$, where $type$ is the message type, $time$ is the time that the message was generated, $orig$ is the intersection where the message originated, $dest$ contains a list of destination intersections for the message, $source$ gives the specific source of the message, and $data$ gives 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 Sched-

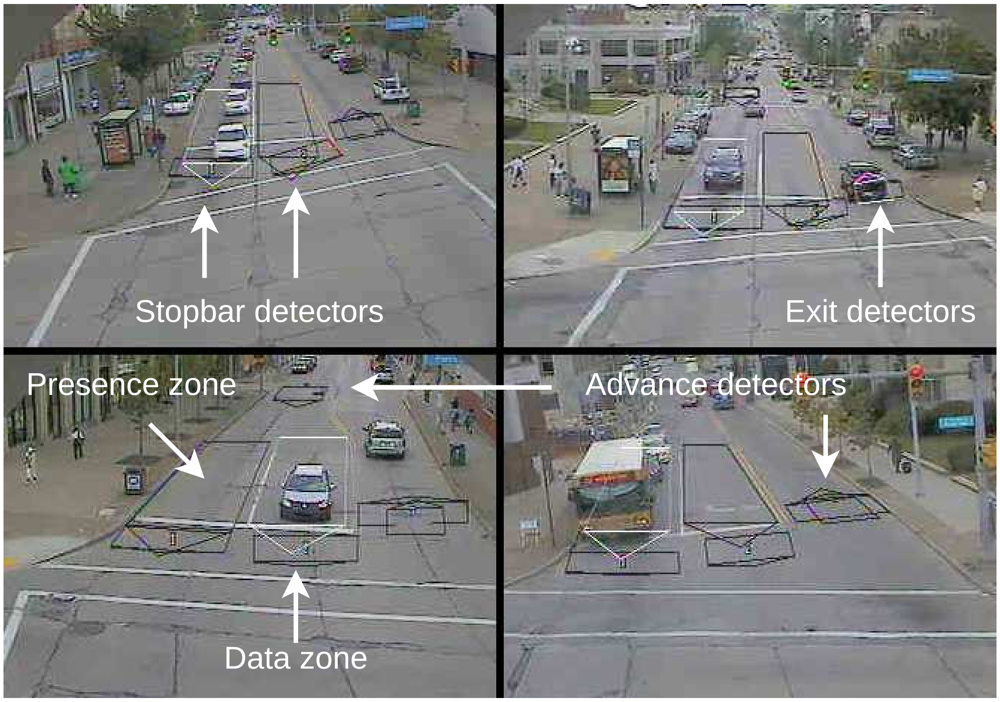


FIGURE 2 The placement of detectors in a typical installation.

uler service. If the sensor functions as an 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, and radar systems. The pilot deployment of SURTRAC described below uses Traficon video detection, but other types of detectors are substitutable. Figure 2 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. For intersections on the boundary of the system, advance detectors might 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 sent to the local Scheduler as well as any relevant neighboring intersections. Occupancy for all presence zones is sensed every 0.1 seconds and aggregated every second, encoded into messages, and sent through the Communicator in the same way.

Executor

In order to control the traffic signals at an intersection, SURTRAC interfaces with the traffic signal controller. The controller continues to enforce maximum and minimum phase durations, transitions between phases, and other safety constraints, while SURTRAC adaptively allocates the green time for the intersection. SURTRAC is designed to work with any controller. For the pilot test, an interface was developed for 170 controllers running the Wapiti firmware.

To operate, SURTRAC places the controller into *free mode*, which is normally used for actuated control. Instead of using vehicle calls (service requests) directly from detectors, the controller is configured to only accept calls from SURTRAC, similar to most other real-time adaptive systems (e.g., (*17, 23*)). 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. 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 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 vehicle calls to continue in the current phase until the scheduled phase end time, at which time the Executor sets vehicle calls for the next desired phase. When the Scheduler updates the schedule, it may extend the current phase by any amount $\geq$ 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, 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), such that schedules may arrive too late to extend the current phase. To protect against 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, and builds a model of the traffic approaching the intersection. It then constructs a schedule for allocating the green time at an intersection between phases. The leading portion of this schedule is then sent to the Executor for controlling the traffic signals, 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.

PILOT

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.

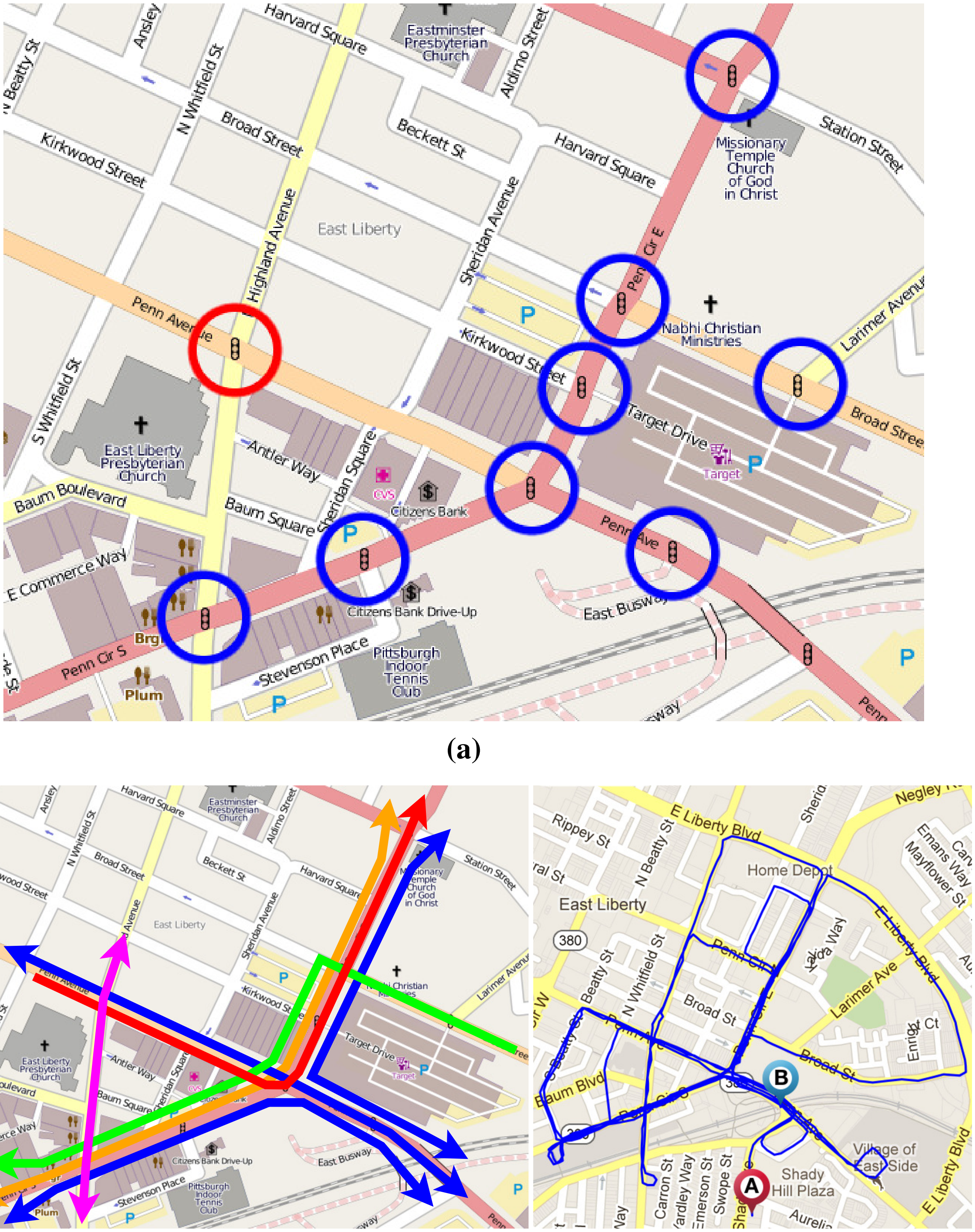
The pilot site, shown in Figure 3, consists of nine intersections. Distances between intersections range from 190 to 635 feet with an average of 382 feet, requiring tight coordination between intersections. Equipment at eight of these intersections—six on Penn Circle—was updated as part of recent redevelopment. 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. These intersections had been controlled with coordinated-actuated timing plans optimized by SYNCHRO during morning and afternoon rush periods and with actuated (free mode) control during the remainder of the day. These timing plans were installed in early 2011. 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. This intersection had been controlled by a single uncoordinated, pre-timed plan. As part of the pilot, this intersection was upgraded with cameras and joined to the existing network using Encom radios.

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 3(b). 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 3(c). 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



(a)

(b)

(c)

FIGURE 3 Map of the nine intersection pilot site in the East Liberty neighborhood of Pittsburgh, Pennsylvania (a). 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 (b). GPS traces were post-processed to evaluate only the fixed routes through the pilot site (c).

TABLE 1 Summary of pilot test results

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%

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. An analysis of traffic volume data for these two periods shows little difference, with roughly 5% higher volumes observed in June (see (*24*) 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) emissions 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 (*24*) 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.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%.

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

---

[1] The emissions numbers reported here are computed based on the fuel consumption model given in (*25*)—the model used by the metropolitan planning organization for the Pittsburgh region—and EPA and EIA data. See (*24*) for details.

TABLE 2 Projected emissions savings

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

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 (*24*) 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.

CONCLUSION

The East Liberty 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 progress, and further expansions are also planned.

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

[1] Schrank, D., T. Lomax, and S. Turner. *Annual Urban Mobility Report*. Texas Transportation Institute, Texas A&M University System, TX, 2011.

[2] Chin, S. M., O. Franzese, D. L. Greene, H. Hwang, and R. C. Gibson. *Temporary Losses of Highway Capacity and Impacts on Performance: Phase 2*. ORNL/TM-2004/209, Oak Ridge National Laboratory, 2004.

[3] Papageorgiou, M., C. Diakaki, V. Dinopoulou, A. Kotsialos, and Y. Wang. Review of road traffic control strategies. *Proceedings of the IEEE*, Vol. 91, No. 12, 2003, pp. 2043–2067.

[4] Stevanovic, A. *Adaptive Traffic Control Systems: Domestic and Foreign State of Practice*. NCHRP Synthesis 403, Transportation Research Board, National Research Council, Washington, DC, 2010.

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

[6] Xie, X.-F., S. F. Smith, and G. J. Barlow. Schedule-driven coordination for real-time traffic network control. In *International Conference on Automated Planning and Scheduling (ICAPS)*, Sao Paulo, Brazil, 2012, pp. 323–331.

[7] Lesser, V., K. Decker, and T. Wagner. Evolution of the GPGP/TAEMS domain-independent coordination framework. *Autonomous Agents and Multi-Agent Sytems*, Vol. 9, No. 1-2, 2004, pp. 87–143.

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

[9] Robertson, D. I. and R. D. Bretherton. Optimizing networks of traffic signals in real time - the SCOOT method. *IEEE Transactions on Vehicular Technology*, Vol. 40, No. 1, 1991, pp. 11–15.

[10] Friedrich, B., T. Sachse, M. Hoops, W. Jendryschik, and G. Reichert. Balance and Varia Methods for Traffic Adaptive Control. In *World Congress on Intelligent Transport Systems*, Yokohama, Japan, 1995, pp. 2356–2361.

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

[12] Sims, A. and K. Dobinson. The Sydney coordinated adaptive traffic (SCAT) system: Philosophy and benefits. *IEEE Transactions on Vehicular Technology*, Vol. 29, No. 2, 1980, pp. 130–137.

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

[14] Porche, I. and S. Lafortune. Adaptive look-ahead optimization of traffic signals. *ITS Journal*, Vol. 4, No. 3-4, 1999, pp. 209–254.

[15] Barriere, J. F., J. L. Farges, and J. J. Henry. Decentralization vs hierarchy in optimal traffic control. In *IFAC Control in Transportation Systems*, Vienna, Austria, 1986.

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

[17] Mirchandani, P. and L. Head. A real-time traffic signal control system: Architecture, algorithms, and analysis. *Transportation Research Part C: Emerging Technologies*, Vol. 9, No. 6, 2001, pp. 415–432.

[18] Boillot, F., S. Midenet, and J. Pierrelee. The real-time urban traffic control system CRONOS: Algorithm and experiments. *Transportation Research Part C: Emerging Technologies*, Vol. 14, No. 1, 2006, pp. 18–38.

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

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

[21] Lämmer, S. and D. Helbing. Self-control of traffic lights and vehicle flows in urban road networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008, p. P04019.

[22] Sen, S. and K. Head. Controlled optimization of phases at an intersection. *Transportation Science*, Vol. 31, No. 1, 1997, pp. 5–17.

[23] Chandra, R. J., J. W. Bley, S. S. Penrod, and A. S. Parker. *Adaptive control systems and methods*. U.S. Patent 8,050,854, Rhythm Engineering, LLC, Filed Sep 24, 2008, Issued Nov 1, 2011.

[24] Smith, S. F., G. J. Barlow, X.-F. Xie, and Z. B. Rubinstein. *Real-time adaptive traffic signal control for urban road networks: The East Liberty pilot test*. CMU-RI-TR-12-20, The Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, 2012.

[25] Wallace, C. E., K. G. Courage, D. P. Reaves, G. W. Schoene, and G. W. Euler. *TRANSYT-7F User's Manual*. UF-TRC-U32 FP-06/07, Office of Traffic Operations, U.S. Department of Transportation, 1984.