Development and Evaluation of Algorithm for Resolution of Conflicting Transit Signal Priority Requests

Milan Zlatkovic, Aleksandar Stevanovic, and Peter T. Martin

The goal of this study was the development and evaluation of an algorithm for resolving conflicting requests for transit signal priority (TSP). This algorithm was designed to work with actual traffic controllers without the need for new hardware or software installations. The algorithm was tested in VISSIM microsimulation and ASC/3 software-in-the-loop controllers on an intersection that will be upgraded to serve two conflicting bus rapid transit (BRT) lines. The ASC/3 logic processor was used to control built-in TSPs in the case of conflicting requests and to develop custom-TSP strategies that would not rely on built-in TSP. Custom TSP provides a much higher level of TSP for transit vehicles than built-in TSP, and it creates opportunities for more adaptable TSP control. The results showed that the widely used first-come, first-served policy for resolution of conflicting TSP requests was not the best solution. Such a policy could perform worse than a policy that provided no priority. For the analyzed intersection, the first-come, first-served option even increased BRT delays by 13% more than did the no-TSP option. The presented algorithm can help resolve the problem of the conflicting TSP requests. The algorithm worked best when combined with several TSP strategies. For the custom-TSP strategies, the application of the algorithm reduced BRT delays by more than 30%, with minimal impact on vehicular traffic. The algorithm shows promising results, and with small upgrades, it can be applied to any type of TSP.

Transit signal priority (TSP) is a traffic control strategy for facilitating transit vehicles that is becoming more and more popular among transit agencies. Although it has been in use for more than 40 years (*1*), recent achievements in detection, communication, and traffic control technologies enable its implementation on a much wider level and in many forms. Many worldwide implementations of TSP have shown the benefits it brings to transit without affecting other users of traffic networks (*1, 2*). For that reason, it is extremely popular among researchers and practitioners.

One of the recognized problems of the expanding implementation of TSP is the conflict between two or more TSP requests. With an increasing number of prioritized transit lines within the same network, the probability of having two or more conflicting requests simultaneously is also increasing. Current standards for TSP implementation do not offer a good solution to this problem (*3*). Several research studies have identified some of the possible ways to overcome the problem (*4–7*), but the actual implementation of these methods in the field has not yet been resolved.

The goal of this paper is the development and evaluation of an algorithm for resolving conflicting TSP requests that can be implemented within existing traffic controllers. The algorithm was tested in VISSIM microsimulation and ASC/3 software-in-the-loop (SIL) controllers on an intersection that will be upgraded to serve two conflicting bus rapid transit (BRT) lines. The ASC/3 logic processor was used to control built-in TSP in the case of conflicting requests and to develop custom-TSP strategies that do not rely on built-in TSP.

The paper is organized as follows. The next section provides a review of the literature that presents methods for resolving conflicting TSP requests. It is followed by a description of the proposed multiple-TSP algorithm. The methods of creating and calibrating simulation models, and the implementation of the algorithm in some of them, are given in the section on modeling methodology. These topics are followed by results obtained through microsimulation and a discussion of results. The last section presents the major conclusions of the study.

LITERATURE REVIEW

TSP is an operational strategy that facilitates the movement of in-service transit vehicles through signalized intersections (*1*). The most important benefits of TSP are improved schedule adherence and reliability and reduced travel time for transit, all of which increase the quality of transit service. Potential negative effects consist primarily of delays to vehicular traffic, or, in some cases, impacts to pedestrian crossing opportunities. TSP can be implemented as passive, active, or adaptive priority (*1*). Passive TSP does not require transit detection or priority requests. It offers a simple progression for transit vehicles along corridors where transit operations are predictable. Active TSP follows transit vehicle detection and subsequent priority request activation. It is usually implemented as the green-extension strategy, early-green strategy, or both to provide a wider green-time bandwidth for transit vehicles. Active TSP can be implemented as unconditional or conditional. Unconditional TSP provides priority for each transit vehicle that sends a request. Conditional TSP provides priority only for transit vehicles that satisfy certain conditions, such as running behind schedule or having more passengers on board. Adaptive TSP considers the trade-offs between transit and

M. Zlatkovic, Room 1650, and P. T. Martin, Room 2137, Department of Civil and Environmental Engineering, University of Utah, 110 Central Campus Drive, Salt Lake City, UT 84112. A. Stevanovic, Civil, Environmental, and Geomatics Engineering, Florida Atlantic University, Building 36, Room 225, 777 Glades Road, Boca Raton, FL 33431. Corresponding author: M. Zlatkovic, milan@trafficlab.utah.edu.

Transportation Research Record: Journal of the Transportation Research Board, No. 2311, Transportation Research Board of the National Academies, Washington, D.C., 2012, pp. 167–175.
DOI: 10.3141/2311-16

traffic delay and allows adequate adjustments of signal timing by adapting the movement of the transit vehicle and the prevailing traffic condition.

Transit agencies are expanding the use of TSP, and such expansion leads to an increasing number of prioritized transit lines within the same network. This greater number in turn increases the probability of two or more transit vehicles approaching an intersection concurrently and sending TSP requests that are in conflict (*2*). The traffic controller's priority request server must decide which vehicle will be given preference. Most TSP implementations are not able to determine the optimal order in which individual requests should be served. The solution for the conflicting requests was found in the first-come, first-served policy, by which the first vehicle requesting priority is served first. Within this policy, the next TSP request may or may not be served in the following cycle.

The Los Angeles Department of Transportation TSP software assigns a higher priority level to the transit line on which TSP was first implemented (*2*). Although not optimal, this method is consistent in resolving conflicting calls. This is one solution for classifying unconditional priority requests.

It is easier to determine the service preferences in the case of conditional TSP. This type of TSP provides priority only to transit vehicles that are behind schedule or carry passenger loads higher than the defined threshold. The National Transportation Communications for Intelligent Transportation Systems Protocol (NTCIP) Standard 1211 for signal control and prioritization classifies TSP requests into the request class types and request class levels (*3*). A priority request message that is sent from a priority request generator to a priority request server contains the following vehicle information: identification, class type, and class level. Once the priority request server receives this message, it determines the order in which to allow priority for conflicting requests on the basis of class type and class level. However, it cannot provide the best solution in the case of two or more requests of the same type and same level.

Resolving conflicting TSP requests has become an emerging topic in TSP research studies. Different optimization methods have been proposed by different researchers. A study that used colored Petri net models for transit priority and preemption looked into how to improve conditional conflicting TSP requests (*4*). The proposed method determines the best order to serve conflicting requests in three steps: (*a*) determine the priority level of transit vehicles on conflicting approaches, (*b*) consider the status of the operation of transit vehicles, and (*c*) decide the priority type, priority degree, and service sequence. The requests with higher priority preempt those with lower priority, creating an order in which to serve the requests.

A different study proposed a decision model for multiple-priority control based on precedence graphs (*5*). The precedence graph model is formed by representing each phase by an activity on arc, following the defined phase sequence and phase barrier constraints. The priority control problem is presented as a mathematical programming formulation with an objective function that minimizes total priority delay. The model is subjected to the constraints of precedence, phase duration, and service phase selection. The problem is defined as a mixed-integer mathematical programming model that can be solved by using readily available tools. It was tested and compared with the first-come, first-served policy. The results showed the potential benefit in developing strategies that are not simply first-come, first-served, in which priority requests can be received with sufficient lead time to allow intelligent service planning.

A dynamic programming model was also used to optimize TSP strategies in the case of conflicting requests (*6*). The objective function of the model is to minimize the total weighted transit delay. The model outputs are the optimal-serve sequence of multiple bus priority requests and corresponding signal timings. The model was tested for a case of conditional priority in which schedule deviation, the number of passengers on board, and the overall traffic were considered. It was compared with a no-TSP model and a first-come, first-served TSP model. The results showed an advantage for the proposed model over the other two in reduced transit delay and impacts on overall traffic.

A recent study proposed a heuristic algorithm for optimizing multiple priority requests at isolated intersections in the context of vehicle-to-infrastructure communications (*7*). The basic concept of this algorithm is to separate the assignment of priority requests to a cycle and phase from the optimization of signal durations. The algorithm was tested in microscopic traffic simulation and compared with the exact mixed-integer linear programming solution as well as with traditional priority algorithms. The results showed that the proposed algorithm was able to provide near-optimal solutions in transit delays and impacts on traffic.

These studies show that the problem of conflicting TSP requests can be solved by using some of the available optimization methods. However, the question of actual implementation of these methods in the field still remains. Each of them requires separate calculation–optimization software that would cause some difficulties and costs for actual implementation. This paper describes an algorithm for resolving conflicting TSP calls and its implementation within an existing logic processor that is using controller software. The algorithm focuses on finding the best way to serve conflicting TSP requests within the existing signal timing plan. It can be used for unconditional or conditional TSP calls of the same type and same level. Although the research used ASC/3 controller software, the principles and commands can easily be transferred to any other software that supports logic commands.

MULTIPLE-TSP ALGORITHM

The proposed algorithm for resolution of conflicting TSP requests is shown in Figure 1. The main postulate of this algorithm is that none of the phases (vehicular or pedestrian) can be omitted during a cycle. The algorithm is defined for unconditional priority, which means that any transit vehicle can place a TSP call and will be served accordingly. Because the network that was used to test the algorithm consists of two conflicting BRT lines, the calls for these lines in the algorithm are referenced as BRT 1 and BRT 2. In this test-case network, both lines are served with the corresponding through movements, but in general, the algorithm can be applied to any other movement. The algorithm works the same way, no matter which BRT line requests TSP first.

The two most important parameters in the algorithm that set the course of action are (*a*) the moment when a TSP call is placed by one or more transit vehicles and (*b*) the current signal phase at that moment. When a TSP call is received from BRT 1, the algorithm checks the current signal phase. If the phase that corresponds to BRT 1 is timing green at that moment, the algorithm will give priority to that BRT line. Then it checks for a TSP call from the conflicting line. If there are no conflicting calls, the TSP works as in the case of a single line with the green-extension strategy: if BRT 1 does not check out during the normal green phase, the BRT phase will be

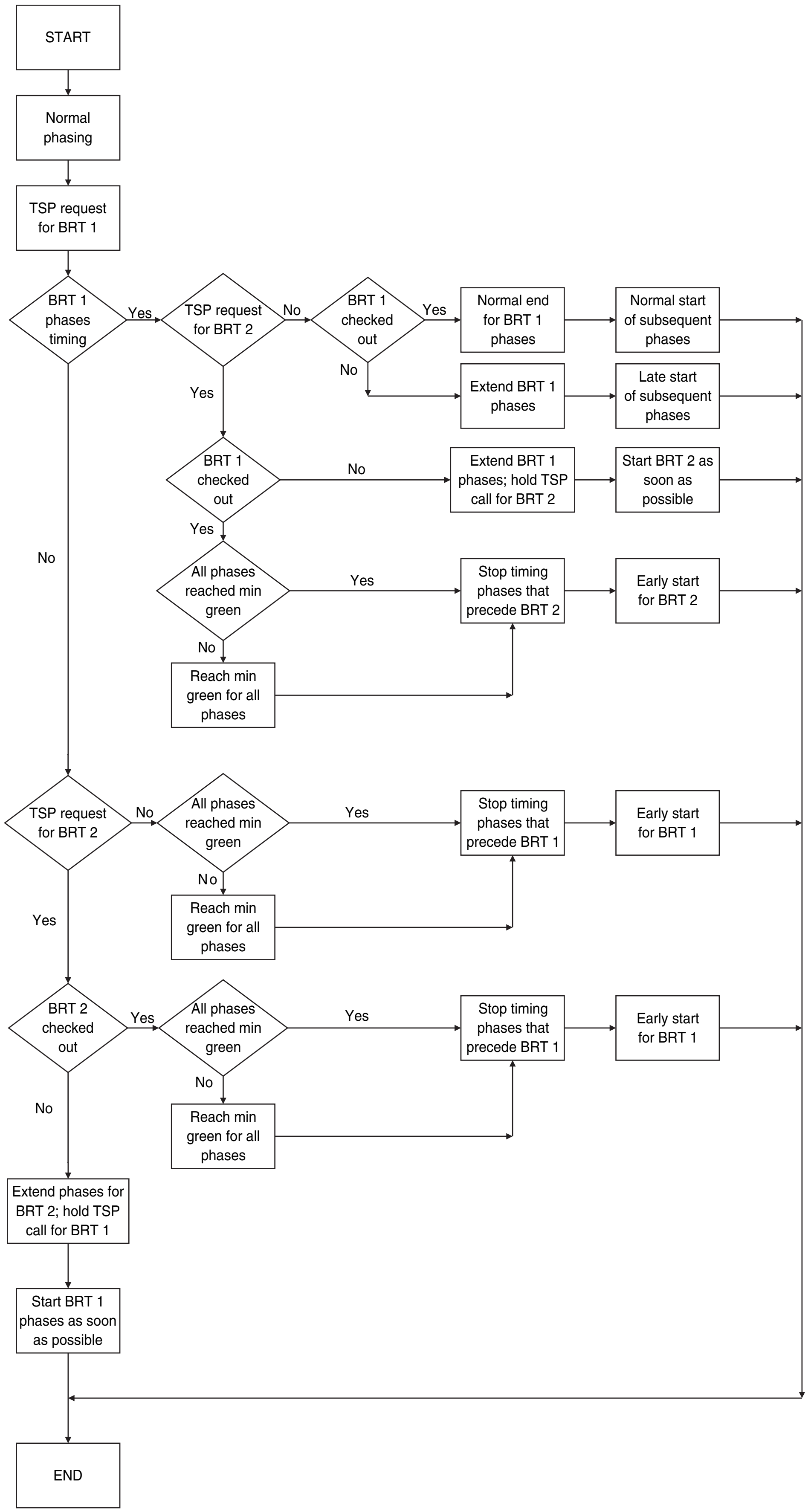

FIGURE 1 Multiple-TSP algorithm (min = minimum).

extended until it checks out or until that phase reaches its maximum green time. If there is a conflicting call from BRT 2, that call will be canceled until BRT 1 has cleared the intersection. Once BRT 1 checks out, the call for BRT 2 will be placed again. In this case, the early-green strategy becomes active to provide an early start for BRT 2. Early green can also be combined with phase rotation, which will rotate the regular sequence of phases to service BRT 2 first, if that is not the case within the normal phase sequence.

If the phase that corresponds to BRT 1 is timing red at the moment when BRT 1 checks in, then some of the conflicting phases are timing green. The algorithm then checks for a call from BRT 2. If there is no call from BRT 2, the early-green strategy (and a potential combination with phase rotation) will become active, shortening the green time for conflicting phases and giving an early start to BRT 1. If there is a call from BRT 2, the algorithm checks which phase is currently timing green. If that is the phase that corresponds to the BRT 2 movement, than the BRT 1 call will be canceled and BRT 2 served with green extension, as previously described. If phases that do not correspond to either BRT 1 or BRT 2 are timing green, generally it does not matter which TSP call is active: those phases will be shortened in accordance with the early-green settings. Then the phasing will go into the next sequence, serving either BRT 1 or BRT 2, which depends on the phases that are next in the phase ring. Potentially, phase rotation can come into play in this case too. When one BRT line is served, the call will again be placed for the other one, which will also be served in accordance with the early-green strategy.

There is a significant difference between the working of this algorithm and that of available software that uses priority request class type and request class level as defined in NTCIP Systems Protocol 1211. The algorithm classifies TSP calls on a case-to-case basis during each cycle in accordance with the current intersection operations. Priority request class type and request class level in existing controllers predefine which transit vehicles will be given priority, and this arrangement is fixed from cycle to cycle. In that way, the algorithm presented here provides a more adaptable way of controlling conflicting TSP calls.

MODELING METHODOLOGY

Test-Case Network

The proposed algorithm and its benefits and impacts on traffic and transit operations were evaluated through VISSIM microsimulation coupled with ASC/3 SIL controllers. The algorithm was tested for one intersection, 3500 S and 5600 W in West Valley City, Utah, which was modeled in accordance with the existing traffic conditions (traffic counts and signal timings). This intersection was selected because, in the future, it will serve two conflicting BRT lines, one along 3500 S (which is already operational) and the other along 5600 W (the construction for which will start in 2015). According to the design plans, both BRT lines are modeled as center-running lines (*8*). The layout of the (future) intersection is shown in Figure 2. The intersection operates on an actuated–coordinated pattern, with coordinated north–south movements and a 130-s cycle length. All approaches have two through lanes and separate left- and right-turn lanes (except the east approach, which has a shared through–right-turn lane). TSP check-in detectors are placed at each intersection approach about 600 ft from the intersection. The check-out detectors are placed after the intersection stop bars. All BRT stops are located on the far sides of the intersection. They are approximately 50 ft from the intersection, and their length is about 120 ft. In addition to this intersection, six surrounding intersections were also modeled with existing signal-timing plans to create more realistic traffic demand for the analyzed intersection.

The network was loaded with traffic in accordance with afternoon peak period (4:00 to 6:00 p.m.) traffic counts for the 3500 S and 5600 W intersection that were collected in the fall of 2008. The simulation model was then calibrated in accordance with the traffic counts for this intersection. The results of the model calibration are presented in Figure 3.

A high R^2 value shows a good correlation between the two data sets. The focus of the model calibration was on the second hour (5:00 to 6:00 p.m.) because this was the true peak hour. The calibration results were the same for the first hour of the simulation (with an R^2 value of .995).

Four model scenarios were used in this research: no TSP, TSP, multiple TSP, and custom multiple TSP. Each of the TSP scenarios was implemented through ASC/3 SIL controllers with signal timings from the field. The models were created by using built-in TSP strategies and the logic processor, as described below for each model. Each model was run for 2 h, with a 15-min build-up time. The results for each model were averaged from 10 randomly seeded simulation runs.

No-TSP Model

The no-TSP model introduced two center-running BRT lines along 5600 W (north–south) and 3500 S (east–west) that conflict with each other at the analyzed intersection. In this case, the BRT lines did not have any special control treatment. The headway for both BRT lines was set to 10 min in each direction, which is the planned service frequency for the future (currently the 3500 S BRT line operates on 15-min headways). Bus stops for both lines were located on the far sides of the intersection. Passenger activity at bus stops was also modeled in accordance with existing (or estimated) data. Traffic control for the surrounding intersections was modeled in accordance with actual signal-timing data. No special control treatment was introduced for BRT lines at those intersections (this also was true for other TSP models; TSP was modeled only at the analyzed intersection). This model served as a basis for creation of other TSP models, which differed from the base one only by their TSP logics.

TSP Model

This model introduced TSP at the analyzed intersection for both BRT lines. In this case, the built-in TSP strategies of the ASC/3 controller software were used. These strategies allowed for green extension–early green in accordance with the parameters that the user defined. The TSP check-in detectors were placed about 600 ft from the intersection. Because the speed of the BRT buses was between 40 and 45 mph, it would take them about 10 s to reach the intersection once they were detected. This fact was used to define the TSP parameters, which were set as follows:

- Maximum green extension for BRT phases: 10 s,
- Maximum red truncation for conflicting through movements: 10 s, and
- Maximum red truncation for (all) conflicting left turns: 5 s.

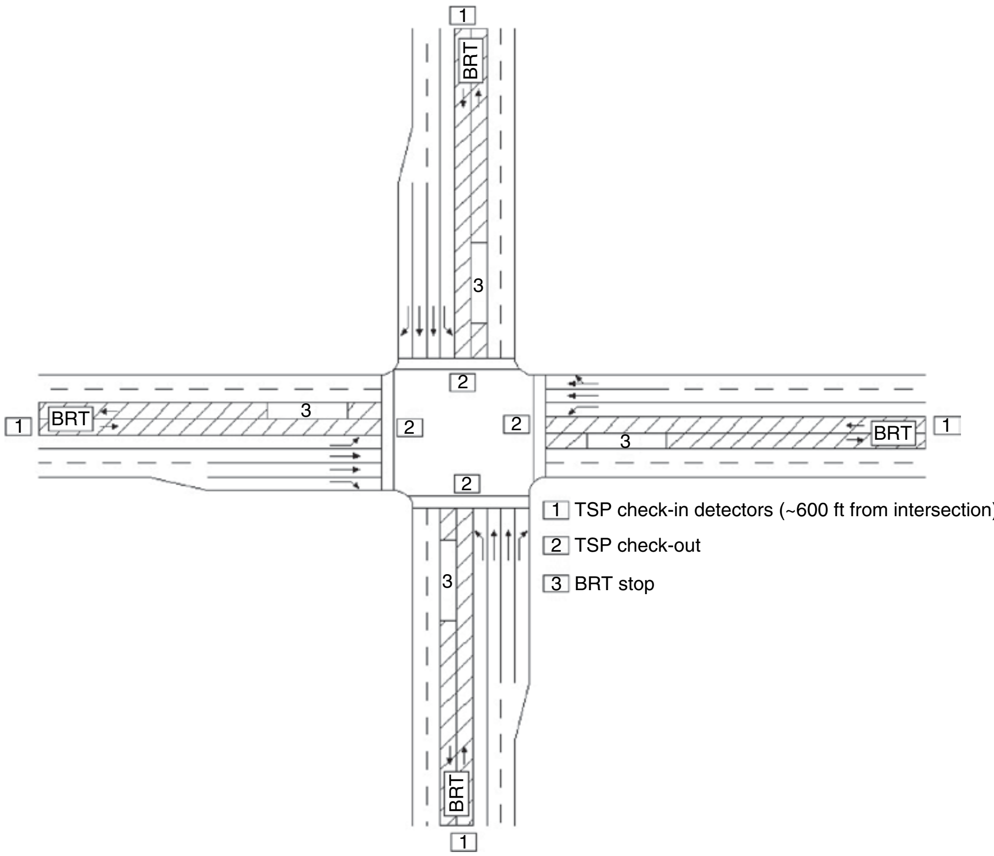

FIGURE 2 Intersection layout.

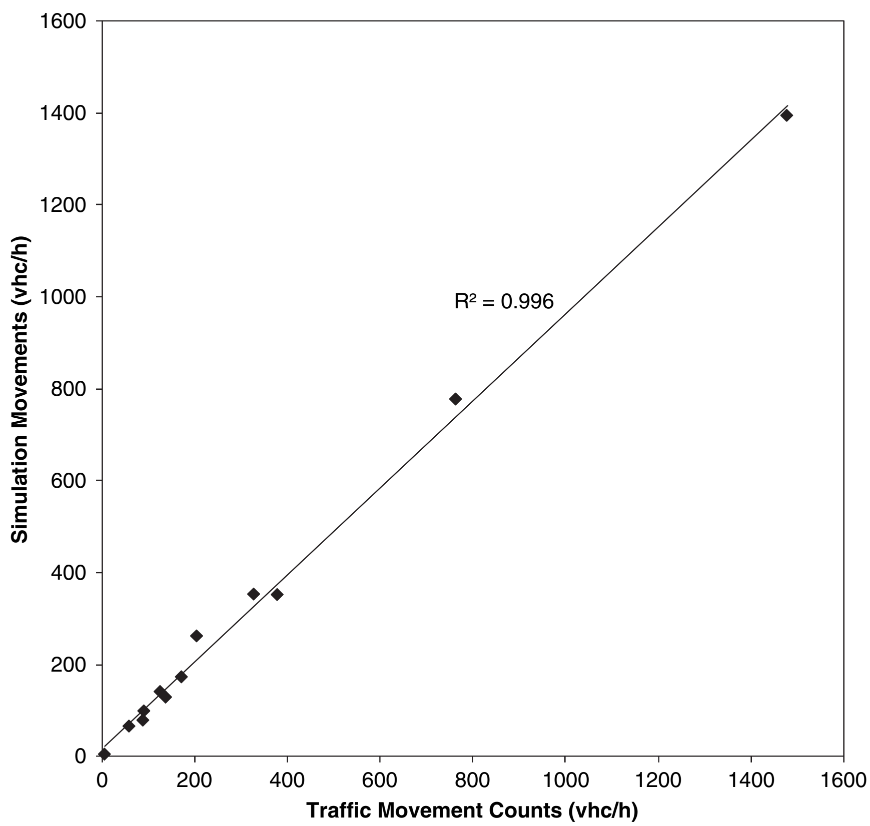

FIGURE 3 Calibration of VISSIM model (vhc = vehicles).

The built-in TSP strategies worked on a first-come, first-served policy. This means that the BRT bus that was detected first would be served first.

Multiple-TSP Model

The multiple-TSP model is an extension of the TSP model. This case used the same built-in TSP strategies and settings as in the previous case. In multiple TSP, the TSP calls were controlled by the logic processor on the basis of the proposed algorithm. This logic was defined in the controller as follows:

IF
  BRT 1 detected AND
  BRT 1 phases timing green AND
  BRT 2 detected
THEN
  Cancel TSP call for BRT 2
  Apply green extension for BRT 1 (if needed)

At the same time, the controller checked whether BRT 1 had cleared the intersection. This checking was achieved through the following set of logic commands:

IF
  BRT 1 checked out AND
  BRT 2 detected AND
  TSP call for BRT 2 not active
THEN
  Call TSP for BRT 2

These sets of logic commands were also controlling TSP calls when none of the BRT 1 or BRT 2 phases was active at the time it was detected approaching the intersection. In this case, the early-green strategy became active by shortening the green times for conflicting phases in accordance with the TSP settings. Any active TSP call could activate this strategy. Once the conflicting phases were forced off, the controller would start those BRT phases that were next in the phase sequence. It can be assumed that the BRT 1 phases were next in the sequence being served and that BRT 2 was waiting at the intersection so that both TSP calls were active. Once the BRT 1 phases started, the same defined logic would deactivate the TSP call for BRT 2, but as soon as BRT 1 checked out, the call for BRT 2 would be placed again so that the early-green strategy would reactivate to prioritize BRT 2.

Custom Multiple-TSP Model

The custom multiple-TSP model is a multiple-TSP model without built-in TSP options. TSP was achieved through a series of logic commands by means of the ASC/3 logic processor. The result was a custom-made priority that allowed a higher level of TSP treatment for BRT. Three TSP strategies were defined: green extension, early green, and phase rotation.

The green extension strategy was achieved through the following logic commands:

IF
  BRT detected AND
  BRT phases timing green
THEN
  Turn off minimum recall for all phases
  Turn off detectors for conflicting phases
  Call MAX 2 maximum green time for BRT phases
  Set coordination free
  Set green for BRT phases

The IF condition for this strategy is that a BRT bus is detected approaching the intersection and the green time for BRT phases is currently on. The logic ensures that the bus clears the intersection before the green ends. The first step turns off detector actuations for all conflicting phases and any minimum-phase recalls. This step clears calls for conflicting phases and enables the BRT phases to continue timing green. However, the duration of this green time can be constrained by the maximum-phase green time or by the coordination offset. The ASC/3 controller has an option of activating three maximum-green times. MAX 1 is the standard maximum green, while MAX 2 and MAX 3 are optional. For the purpose of green extension, the logic refers to the MAX 2 time for the BRT phases, which in this case is large enough to allow the BRT bus to clear the intersection on green. To maintain the coordination offset, the controller can also end the green time of the coordinated phases at a certain point during the cycle. This decision can conflict with the green extension. The logic sets coordination to free running until the bus clears the intersection. Setting the control logic to dwell on green ensures that the BRT phases remain green while the IF conditions are satisfied. When the bus crosses the stop bar, this logic deactivates and the intersection returns to normal operation. The travel time of any BRT bus from the check-in detector to the intersection is about 10 s. So, in the worst case, the BRT phases will be extended by 10 s at most. This period corresponds to the previously defined TSP settings.

If a BRT bus is detected approaching the intersection while the conflicting phases are timing green, the early-green strategy terminates those phases and provides an earlier start for BRT. The logic that drives this strategy is as follows:

IF
  BRT detected AND
  Conflicting phase is timing green
THEN
  Turn off detectors for that conflicting phase

The call for the phase ends when the detectors for the conflicting phase are turned off, and the controller stops timing green once it reaches the minimum-phase green time. This strategy does not omit any phase, whether or not that phase is on a minimum recall. The logic activates once the phase green starts timing, which ensures the minimum green for that phase. If one of the conflicting pedestrian phases is active at the same time as this logic, the conflicting phases will end when the pedestrian phase turns red. This decision means that active pedestrian phases will not terminate earlier. Turning the detectors off for the conflicting phases is a better option than forcing their green time to end (which can also be achieved through the control logic) because, in this case, the conflicting phases gap out, which does not disturb intersection coordination.

Phase rotation changes the phase sequence so as to serve a transit phase faster. Only the phases on the same intersection approach (within the same control barrier) can be rotated. At the analyzed intersection, the phase sequence was defined as leading left turns and lagging through movements. All BRT phases time concurrently with vehicular through phases. If a BRT vehicle was detected at the intersection while some of the movements on the other approach timed green, phase rotation would change the sequence for left and through phases. This process would allow the through movements to be served first and left turns after that. This strategy reduces delays for transit vehicles, but it can also have safety benefits in the case of a transit lane that is positioned in the middle of the roadway: the strategy reduces conflicts between transit and left-turning vehicles. The logic works the same for both BRT lines and is defined as follows:

IF
  BRT detected AND
  Left turns on BRT approach timing red
THEN
  Select alternative sequence with leading through and lagging left phases

If a BRT bus is detected on one intersection approach, the second IF command checks the timing for the left-turn phases on that approach. If these left-turn phases are red, two options are possible: the through phases on that approach (and on BRT phases) are green or any phase on the other approach is green. In the first case, the bus will clear the intersection and deactivate phase rotation. In the second case, both left turns and through (and BRT) phases on the approach in question are timing red. The normal phase sequence in this case would start with leading left turns and lagging through and BRT phases. However, to facilitate BRT operations, the logic will select an alternative sequence, which is defined as leading through phases and lagging left turns, serving the BRT phases first. This alternative sequence has to be predefined in the ASC/3 controller configuration and referenced through a proper logic command. The early-green strategy is always active along with phase rotation.

In this case, the TSP parameters were set as follows:

- Maximum green extension for BRT phases: 27 to 54 s, depending on phase with an average of 39 s;
- Maximum red truncation for conflicting through movements: 6 to 28 s, depending on phase with an average of 19 s;
- Maximum red truncation for (all) conflicting left turns: 3 to 18 s, depending on phase with an average of 9 s; and
- Phase rotation active.

The same logic that controls multiple-TSP calls was also active here. The logic was set to control TSP calls in cases of green extension and early green in accordance with the algorithm.

RESULTS

Number of Conflicting Requests

The main input for the analysis was the number of conflicting TSP requests that appeared during the evaluation period. VISSIM was coded to record activation of BRT detectors for the 2-h period (simulation seconds 900 to 8,100). The results for the number of conflicting requests were obtained through filtering those TSP calls that appeared during the same cycle on conflicting approaches. Table 1 shows simulation times when two conflicting calls appeared and the directions of the conflicting vehicles. These results were extracted for one simulation run for each scenario, but the results were similar for all runs.

TABLE 1 Conflicting TSP Requests: Simulation Time and Directions

Simulation Time (s)	Directions of TSP Calls	Simulation Time (s)	Directions of TSP Calls
No TSP		Multiple TSP	
1,058	↓ ←	1,058	↓ ←
2,871	↓ ←	2,340	→ ↓
4,055	↓ ←	2,870	↓ ←
5,336	→ ↓	4,055	↓ ←
5,872	↓ ←	5,872	↓ ←
7,140	→ ↓	7,139	→ ↓
7,693	↓ ←	7,695	→ ↓
TSP		Custom Multiple TSP	
1,058	↓ ←	1,058	↓ ←
2,340	→ ↓	2,870	↓ ←
2,870	↓ ←	4,055	↓ ←
4,055	↓ ←	5,872	↓ ←
5,336	→ ↓	7,139	→ ↓
5,872	↓ ←	7,691	↓ ←
7,140	→ ↓		
7,693	↓ ←		

Intersection and Network Performance

The analysis was focused on one intersection, so the best way to assess the impacts and benefits of the tested scenarios was through parameters of intersection performance. The main parameters used in this case were the number of vehicles, the delay per vehicle, and the number of stops per vehicle for through, left, and BRT movements (right turns were not analyzed because they are allowed on red). The results for the second hour (5:00 to 6:00 p.m.) for each scenario are shown in Table 2. Table 3 shows the aggregated values of these parameters for the entire 2-h period, calculated on the intersection level separately for vehicles and BRT.

Some of the most important performance parameters were observed on a networkwide level. This analysis can show the impacts that the tested strategies have on the surrounding network. These parameters, aggregated for the 2-h period, are shown in Table 4.

DISCUSSION OF RESULTS

Number of Conflicting Requests

Table 1 shows that conflicting requests appeared at approximately the same time in each scenario. However, the number of those requests was not the same for each scenario. Seven conflicting requests were recorded for the no-TSP and multiple-TSP scenarios, eight for the TSP scenario, and six for the custom multiple-TSP scenario. Changes in the treatment of BRT vehicles at the intersection introduced in each scenario caused changes in transit operations, such as the number of passengers on board, the number of alighting and boarding passengers at BRT stops, and waiting time at the intersection. For that reason, some of the BRT vehicles appeared during a different cycle in different scenarios. The majority of conflicts occurred between the

TABLE 2 Parameters of Intersection Performance by Scenario

	No TSP			TSP			Multiple TSP			Custom Multiple TSP		
Movement	Vehicles	Delay (s)	Stops	Vehicles	Delay (s)	Stops	Vehicles	Delay (s)	Stops	Vehicles	Delay (s)	Stops
NBT	773	32	0.8	772	34	0.8	769	34	0.8	777	27	0.7
NBL	130	47	0.9	128	50	0.9	128	50	0.9	128	61	1.0
SBT	1,393	11	0.3	1,396	11	0.3	1,393	11	0.3	1,388	19	0.6
SBL	262	69	1.1	262	69	1.1	263	67	1.0	259	64	1.2
EBT	353	49	0.9	353	49	0.9	353	49	0.9	358	47	0.8
EBL	5	65	0.9	5	58	0.9	5	65	0.9	5	74	1.0
WBT	353	43	0.9	353	42	0.9	354	43	0.9	363	44	1.0
WBL	141	74	1.3	141	82	1.3	141	79	1.3	143	77	1.3
BRT NB	6	29	0.5	6	34	0.6	6	37	0.6	6	8	0.1
BRT SB	6	31	0.4	6	33	0.5	6	29	0.4	6	43	0.8
BRT EB	6	43	0.7	6	48	0.7	6	49	0.7	6	21	0.4
BRT WB	6	52	0.8	6	60	0.7	6	50	0.8	6	41	0.7
Total vehicles	3,410	32	0.7	3,410	32	0.7	3,406	32	0.7	3,421	34	0.8
Total BRT	24	39	0.6	24	44	0.6	24	41	0.6	24	28	0.5

NOTE: NB = northbound; SB = southbound; EB = eastbound; WB = westbound; ( ) T = through traffic; ( ) L = left turn.

TABLE 3 Aggregated Parameters of Intersection Performance by Scenario

	No TSP			TSP			Multiple TSP			Custom Multiple TSP		
Mode	Vehicles	Delay (s)	Stops	Vehicles	Delay (s)	Stops	Vehicles	Delay (s)	Stops	Vehicles	Delay (s)	Stops
Vehicles	6,660	32	0.7	6,649	33	0.7	6,646	32	0.7	6,650	34	0.8
BRT	49	42	0.7	49	47	0.7	49	45	0.7	48	29	0.6

peak-directed BRT vehicles (southbound and westbound) in each scenario. Some of the conflicts occurred between the southbound and eastbound vehicles, while the northbound vehicles did not cause any conflicts.

Intersection and Network Performance

A comparison of the parameters of intersection performance in Table 2 shows that the defined TSP strategies have little impact on vehicular traffic. Each scenario had certain impacts on all intersection movements. In general, all scenarios increased delays for left turns and reduced delays for through movements. However, the scenarios had little or no impact on delays at the intersection level. Table 3 shows that TSP and multiple TSP had no impact on vehicular traffic and that the impact was minimal (about a 2-s increase in delays, or around 6%) for the custom multiple TSP.

The models performed differently from the standpoint of BRT performance. The built-in TSP performed even worse than when no TSP was implemented. In this case, the built-in TSP even increased BRT delays by 13% compared with no TSP. This result shows that the first-come, first-served policy was not a good choice for conflicting TSP calls. In addition, the TSP scenario had the highest number of conflicting requests of all the scenarios. Multiple TSP performed better but still did not show much improvement over no TSP. However, the peak-period transit directions (southbound

TABLE 4 Aggregated Parameters of Network Performance by Scenario

Parameter	No TSP	TSP	Multiple TSP	Custom Multiple TSP
Total number of vehicles	21,172	21,168	21,170	21,173
Average delay time per vehicle (s)	67	69	69	71
Total delay time (h)	394.8	407.7	408.2	416.0
Average stopped delay per vehicle (s)	44	45	45	46
Total stopped delay (h)	259.2	265.7	265.5	270.9
Average number of stops per vehicle	1.6	1.6	1.6	1.7
Total number of stops	33,266	34,398	34,564	35,664
Average speed (mph)	24.6	24.3	24.3	24.2

and westbound for the given network) showed some improvement with this strategy. When one considers that the majority of conflicts occurred between the southbound and westbound BRT vehicles, one sees that the algorithm brought some improvements over the conventional TSP. The custom multiple TSP yielded the best BRT performance. During the 2-h period, the average reduction in delays for each BRT vehicle was about 13 s, which was an improvement of more than 30% over no TSP. The improvements were substantial for all BRT movements. These results show that custom multiple TSP was the best of the four strategies for resolving conflicting TSP calls. It also offered the highest level of TSP among the strategies.

The impacts of the tested strategies were also minimal on the networkwide level. In this case, the custom multiple TSP had the biggest impacts on networkwide delays (the increase in delays was also around 6%).

CONCLUSIONS

The main contribution of the paper is the development of an algorithm for programming field traffic controllers to implement conflicting TSP requests. This procedure can use existing hardware and software and thereby reduce the costs associated with new installations. The paper also shows the benefits of using microsimulation to test various real-world controllers' strategies through the ASC/3 SIL platform.

The conclusions of the study are as follows:

1. The widely used first-come, first-served policy for resolution of conflicting TSP requests is not the best option. It can perform worse than a policy that provides no priority for any of the conflicting BRT lines.
2. The algorithm presented in the paper can help resolve the problem of the conflicting TSP requests. The algorithm works best when combined with several TSP strategies.
3. The logic processor (which exists in most traffic control software) can be successfully used in defining custom-TSP strategies, which can perform better than built-in TSP options. It can also help to control built-in TSP better.

Future work should focus on networks with several intersections and more conflicting TSP requests to find the best option for resolving those conflicts. The proposed algorithm shows advantages when applied to a single intersection with two conflicting TSP requests. It should be upgraded to allow for more than two requests and optimized for coordinated networks. The algorithm should also be upgraded to include conditional and adaptive TSP and combined with some available TSP software that classifies multiple TSP requests.

REFERENCES

1. Smith, H. R., B. Hemily, and M. Ivanovic. *Transit Signal Priority (TSP): A Planning and Implementation Handbook.* Intelligent Transportation Society of America, Washington, D.C., 2005.
2. Li, Y., P. Koonce, M. Li, K. Zhou, Y. Li, S. Beaird, W. B. Zhang, L. Hegen, K. Hu, A. Skabardonis, and Z. S. Sun. *Transit Signal Priority Research Tools.* FTA-CA-26-7049-2008.1. FTA, U.S. Department of Transportation, 2008.
3. *NTCIP 1211: Object Definitions for Signal Control and Prioritization.* National Electrical Manufacturers Association, Rosslyn, Va., 2004.
4. Li, L., W. H. Lin, and H. Liu. Traffic Signal Priority/Preemption Control with Colored Petri Nets. *Proc., 8th International IEEE Conference on Intelligent Transportation Systems,* Vienna, Austria, 2005, pp. 694–699.
5. Head, L., D. Gettman, and Z. Wei. Decision Model for Priority Control of Traffic Signals. In *Transportation Research Record: Journal of the Transportation Research Board, No. 1978,* Transportation Research Board of the National Academies, Washington, D.C. 2006, pp. 169–177.
6. Ma, W., Y. Liu, and X.-G. Yang. A Dynamic Programming Model for Bus Signal Priority with Multiple Requests. Presented at 90th Annual Meeting of the Transportation Research Board, Washington, D.C., 2011.
7. He, Q., K. L. Head, and J. Ding. Heuristic Algorithm for Priority Traffic Signal Control. In *Transportation Research Record: Journal of the Transportation Research Board, No. 2259,* Transportation Research Board of the National Academies, Washington, D.C., 2011, pp. 1–7.
8. Martin, P. T., and M. Zlatkovic. *Evaluation of Transit Signal Priority Strategies for Bus Rapid Transit on 5600 West Street in Salt Lake County, Utah.* MPC-09-213A. Mountain–Plains Consortium, Fargo, N.Dak., 2010.

The Traffic Signal Systems Committee peer-reviewed this paper.