Publish Information:

Analysis of Communication Channel Establishment for Transit Signal Priority Service

Chin-Woo Tan, Meng Li, Sungsu Park, Peter Lau, and Hongchao Liu

Corresponding Author / Submitting Organisation

Chin-Woo Tan
California PATH, Headquarters
University of California, Berkeley
Richmond Field Stations, Bldg. 452
1357 South 46th Street
Richmond, CA 94804
U. S. A.

E-mails of Authors

Chin-Woo Tan: tan@eecs.berkeley.edu
Meng Li: meng_lee@berkeley.edu
Sungsu Park: sungsu@sejong.ac.kr
Peter Lau: p_lau@uclink.berkeley.edu
Hongchao Liu: hongchao.liu@ttu.edu

This paper is original and has not been published elsewhere.
Figure Captions

Figure 1 Timeline of a single priority request process

Figure 2 Number of buses in active service for 30 TSP bus routes

Figure 3 Distribution of number of calls (initial requests) made in each hour

Figure 4 Histogram of number of requests processed per sec, $T = 30$ sec

Figure 5 Histogram of queue size per sec, $T = 30$ sec

Figure 6 Histogram of queue waiting time, $T = 30$ sec

Figure 7 Histogram of queue waiting time, process rate = 1/sec, $T = 30$ sec

Figure 8 Timelines for multiple priority request processes

Figure 9 Histogram of number of requests processed per sec; multiple requests @ {40, 20, 10}

Figure 10 Distribution of no. of dropped (first) calls by hour; multiple requests @ {40, 20, 10}

Figure 11 Hourly distribution of non-empty queue; multiple requests @ {40, 20, 10}

Figure 12 Histogram of first call queue size; multiple requests @ {40, 20, 10}

Figure 13 Histogram of first call waiting time; multiple requests @ {40, 20, 10}

Figure 14 Hourly distribution of dropped calls, 20% no need for priority

Figure 15 Histogram of first call queue size, 20% no need for priority
Analysis of Communication Channel Establishment for Transit Signal Priority Service

Chin-Woo Tan, Meng Li, Sungsu Park, Peter Lau, and Hongchao Liu

Abstract: A critical issue encountered in implementing a transit signal priority system is to ensure that communication between transit vehicles and the traffic centre can be established for transmitting signal priority requests and vehicle arrival time information needed for executing signal priority control at signal-controlled intersections. The available communication channel bandwidth, however, is not guaranteed to meet the demand from a potentially large number of transit vehicles requesting signal priority service. In this paper, we define and analyse this supply-demand problem with simulations of a simple time model. The emphasis is to investigate the logistic issues and identify the critical parameters, so that this simple model might serve as a framework for constructing a more complex realistic model upon which an efficient scheduling protocol could be designed to meet signal priority service demand.

Keywords: vehicle arrival time, transit signal priority request, channel process rate, process priority, process queue, supply-demand analysis
1. Introduction

Transit signal priority (TSP) is an operational strategy that facilitates the movement of a transit vehicle through a signal-controlled intersection by modifying the normal signal operation process (Balke, 2000; Baker, 2002; Smith, Hemily and Ivanovic, 2005). It can be implemented in a variety of ways, such as passive priority, early green (red truncation), green extension, actuated transit phase, phase insertion, and phase rotation (Skabardonis, 2000). The intelligence of a TSP system includes a travel time prediction algorithm (Tan et al, 2006) that anticipates the arrival of a transit vehicle at a traffic signal and gives priority to the vehicle in order to minimise its delay at intersection and the impacts on the rest of traffic and pedestrian safety (Dion, Rakha and Zhang, 2004; Ngan, Sayed and Abdelfatah, 2004). The prediction algorithm predicts the time it takes for a transit vehicle to arrive at the next signalised intersection if its current distance from the intersection is known. This distance can be calculated using the current real-time GPS location of the vehicle and the GPS coordinate of the intersection. The predicted arrival time computed at the current time \( t \), denoted \( T_{a}(t) \), is called the time-till-arrival (TTA).

Since GPS resolution is roughly 1 Hz, we can assume the TTA for each transit vehicle will be computed every second. This information is used to determine if and when the vehicle will need priority service as it approaches its next intersection (Shalaby, Abdulhai and Lee, 2003; Liu, Skabardonis and Li, 2006).

In a centralised TSP system, when a transit vehicle requests for signal priority to be executed at its next signalised intersection, the vehicle must first establish a communication with the traffic management centre (TMC). The availability of this communication channel, however, is not guaranteed because of the potentially large number of transit vehicles requesting priority service at the same time, especially in peak hours, and the limited number of processing time slots per second (bandwidth limitation) available to process these priority requests, one at a time. If the channel is busy, the request will not be processed. The request is repeated until it is processed. To distinguish the initial request from the subsequent requests, we will refer an initial request to as a call. A priority request process includes the initial request and the subsequent repeated requests. The process must be done before a scheduled process deadline. This is the time when the vehicle arrives at its next signalised intersection or a pre-specified time by which the request must be processed. After the process deadline, the transit vehicle cannot make any more requests. We say the request process is dropped, or simply the call is dropped, if none of the requests from the same request process is processed by the deadline. The time at which a request is granted, which is also the time when a communication with the TMC is established, is called the process time of the request process. The difference between the process time and the time of the initial request is the queue waiting time. We will explain the concept of a queue later. This timeline of a single request process is illustrated in Figure 1.

The problem is therefore to design a protocol for assigning priority requests to available process time slots so that objectives such as minimising the number of dropped request processes and the probability of dropping higher priority processes are met. Prior to this design task, the logistic and complexity of this channel supply-demand problem must be investigated. So the focus of this paper is to analyse the problem via simulations of a
simple time model. A few critical parameters are identified. They can be incorporated to construct a more complex realistic model that serves as a basis for the protocol design.

We first discuss some logistic issues. The demand for priority service is time-dependent. At any given time-of-day (TOD), this demand depends on the number of in-service transit vehicles that will need to have their priority requests processed in, say, \( T \) seconds. The demand is also stochastic since the TTA prediction for each link connecting two successive intersections is a random variable. On the supply side, the channel bandwidth is limited by the number of requests that it can process per second. This is the process rate. There are two available communication channels: a reserved polling channel and a contention channel. The polling channel is used by the TMC to initiate a communication with a transit vehicle. This operates according to a scheduling protocol which determines the times at which transit vehicles can communicate with the TMC. Typically, a vehicle is polled no more than once every two minutes. When it is the turn for a vehicle to talk on the polling channel, bus data such as GPS coordinates, predicted arrival time, on-time performance index, and signal priority request, if needed, will be transmitted. The polling channel cannot process all the requests because of its slow polling rate. To make their requests, vehicles have to “compete” for access on a contention channel. This is a shared channel that allows communication between the TMC and one transit vehicle at any time. Our simulations will be restricted to having only the contention channel. The analysis can be extended to include the polling channel once a polling schedule is defined.

In §2 we discuss how process priority, a condition that requires processes with earlier process deadlines are processed first, can be met via the probability of establishing communication on the channel. In §3 we consider a simple supply-demand time model in which the stochastic nature of arrival time prediction is replaced by a “mean arrival time” profile with deterministic scheduled arrival times. Some simulations of this model are presented. The analysis is further extended in §4 by varying the channel process rate. We consider implementation of multiple priority requests in §5-6. The on-time performance measure parameter is discussed in §7. Some remarks are noted in §8. Simulation results are presented in the Appendix.

2. Realisation of process priority

The first requirement is to guarantee that request processes with earlier process deadlines are processed first. This is similar to the Earliest-Deadline-First protocol commonly used in real-time operating systems, where a scheduling principle places processes in a priority queue\(^\text{1}\). A service deadline is assigned to each process and the protocol then always serves the process with the earliest deadline. Consider the timeline shown in Figure 1. Let \( t \) be the current time. Suppose an initial request is made when a transit vehicle is \( T \) sec from its next signalised intersection, so the process deadline is at time \( t + T \). If another initial request from a second vehicle is made at \( t' > t \), its deadline is at time \( t' + T \). The request from the first vehicle should be processed no later than that from the second vehicle. This can be achieved if the request processes are placed in a “service queue” so that the process with the earliest deadline is always processed first. In a TSP system, there is not a server that places processes in a queue, and requests made when the channel

\(^{1}\text{Note that process priority is different from signal priority.}\)
is busy are not processed. Nonetheless, the concept can be employed if the probability of processing the request with the earliest deadline is higher than the probabilities of processing other requests with later deadlines. This can be realised by requiring the number of requests made per second to be higher for the process with the earliest deadline, thus increasing its probability of establishing communication on the channel.

Table 1 Realisation of process priority as a service queue via the probability of channel establishment

<table>
<thead>
<tr>
<th></th>
<th>1st sec</th>
<th>2nd sec</th>
<th>3rd sec</th>
<th>4th sec</th>
<th>5th sec</th>
<th>6th sec</th>
</tr>
</thead>
<tbody>
<tr>
<td>veh. #1</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>8</td>
<td>dropped</td>
<td>-</td>
</tr>
<tr>
<td>veh. #2</td>
<td>-</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>8</td>
<td>dropped</td>
</tr>
</tbody>
</table>

The queue concept is illustrated in Table 1 with the number of requests made in the subsequent seconds increases arithmetically as \(2n\), if none of the previous requests is processed. We use \(t = 1\), \(t' = 2\), and \(T = 4\), so the priority request from vehicle #1 has an earlier process deadline at time \(t + T = 5\). There are more requests from vehicle #1 in each second prior to its process deadline, so the probability of vehicle #1 gaining access to the contention channel is always higher that of vehicle #2. This is consistent with the idea of a service queue that processes request from vehicle #1 no later than vehicle #2. One can increase this probability by changing the arithmetic rate or imposing another scheme, such as a geometric rate of \(2^n\). The queue size will depend on the channel process rate, the time an initial request is made, and the time distribution of the process deadlines. The TTA prediction accuracy could also be used to determine process priority, so that more accurate and reliable messages are processed with higher priorities.

3. Analysis of priority request demand

The demand for signal priority is complicated by the stochastic nature of arrival time prediction. Nonetheless, one could first analyse and investigate the priority request demand by considering the “mean arrival time” at each intersection. In this time model, transit vehicles arrive at their next intersections on time according to some deterministic schedules. This implies the process deadlines are pre-determined and the TTA calculated at the current time \(t\) is simply the scheduled arrival time minus the current time. This can easily be calculated without using a TTA prediction algorithm. On the supply side, we assume the contention channel can process two requests per second, one at a time. These are the supply-demand conditions for the baseline simulation analysis. The analysis can then be extended by to include statistical variants of the TTA prediction, on-time performance measure, and a polling schedule. Note that statistical distributions for TTA prediction are different for different links.

Our simulations use the schedules of 30 bus routes from a transit company in the San Francisco Bay Area. Each bus schedule lists the arrival times, with a time resolution in second, at some specific time points. The time points are a subset of the signalised intersections along the route. The arrival times at intersections that are not provided in the schedules are obtained by using linear interpolation of the scheduled arrival times at the
time points. This linear mean travel time model is the historical model developed in (Tan et al, 2006). The arrival time prediction algorithm developed there uses a historical model “continuously” tuned by a real-time adaptive model.

We analyse the priority request demand in terms of the number of vehicles in service and the number of intersections that are crossed at the same time. Figure 2 (in Appendix: Simulation results) shows the number of active vehicles with respect to the TOD (in hour). In the morning commute hours, as many as \( N(t) = 57 \) transit vehicles could be in service in the network. In principle, the same number of buses could be making priority requests if they all arrive at their next intersections at the same time, that is, with the same process deadline. Given the scheduled arrival times at intersections (or process deadlines), we set a rule to determine when a priority request can be made. The rule permits a vehicle to make its initial request when it is at a fixed look-ahead time away from its next arrival time. If the initial request is not processed, additional requests are made before the vehicle arrives at its next intersection.

**Rule 1:** Request times for a Single Request Process: Let \( t \) be the current time and \( T_A(t) \) be the TTA computed at the current time, so the process deadline is \( t + T_A(t) \). Let \( T \) be a fixed “look-ahead” time. A priority request can be made if \( T_A(t) \leq T \).

The scheduled arrival times have the format of hh:mm:ss. If the time step is 1 sec or an integer fraction of 1 sec, an initial request (also referred to as a call) is made as soon as \( T_A(t) = T \). So the call time is the process deadline minus the look-ahead time. Each call has a one-to-one correspondence with a request process. So zero dropped call means zero dropped request process. We analyse the request demand in terms of the number of calls (initial requests) made. The simulations assume the type of process priority discussed in §2. We simulated the number of calls made in each second if calls are made at exactly TTA = 30 sec. This is also the number of intersections that will be crossed in exactly 30 sec. The hourly distribution is shown in Figure 3. In the 8th hour, 7-8am, the probability when there is no call in a second is 55.39%. This increases to 87.72% for 5-6am, which is an off-peak hour. Simulations also show that there can be as many as 14 simultaneous calls during the morning commute hour 7-8am. If we change the call time to TTA = 20 sec, the sec-by-sec time plot of the number of calls made simply shifts to the right by 10 sec. So it is time-invariant with respect to the call time.

On the average a call is made every 67 sec. This agrees with our belief that the polling channel alone cannot process all the priority requests, since the polling rate is about once every two minutes. The skew hourly distribution in Figure 3 suggests that it will be more meaningful to focus the analysis on the time window from 5am to 10pm. The probability of zero call in a second in this time window is 66.89%. It would be 74.23% if the entire 24-hour window is considered.

### 4. Supply analysis of channel process rate

Next we consider the supply side in the simulation analysis. The capability to process priority requests is limited by the process rate of the contention channel, and the polling frequency of the polling channel. Assume the contention channel can process two
requests per second, one at a time. The efficiency at which requests are processed will depend on (1) the time at which an initial request (or call) is made relative to the process deadline, (2) the time constraints of other request processes, such as that of a checkout process that will be imposed later, and (3) the channel process rate. The ultimate objective is to design a protocol to maximise the efficiency of this request-process (or demand-supply) system, with the inclusion of the statistical variants of prediction of arrival time at intersections, the polling schedule, and an on-time performance measure of real-time and/or historical data. In our model, we consider only the process rate. A more complex model needs to be developed in the future to include the statistical variants and polling frequency. The call time will be link-specific if we include the TTA prediction error distribution for each link. When a call is made, the request process is placed in an “imaginary” queue if the call is not processed immediately. A call with the earliest process deadline in the queue is processed first by requiring the probability of channel establishment increases with earlier process deadlines (see §2).

Figure 4 is a histogram of the number of requests processed per second when initial requests (or calls) are made at the look-ahead time, $TTA = T = 30$ sec. Figure 5 shows a histogram of the number of unprocessed calls in the queue. There are no dropped calls. The queue waiting time is the time difference between the process time and the time of initial request. A distribution of the waiting time is shown in Figure 6. If the look-ahead time $T$ is decreased, there is no change in the queue distribution until it drops below 7 seconds, the longest waiting time shown in Figure 6. Then we begin to see dropped calls. The sensitivity to the look-ahead time $T$ appears to be approximately concave, with no changes if $T \geq 7$ sec. The distribution is sensitive to the process rate. If we decrease it to one request per second, there are still no dropped processes, but the longest waiting time increases to 14 sec. This results in an increase in the average waiting time from 0.86 sec to 1.87 sec. Also the probability that a call is not processed immediately (i.e. with waiting time > 0) increases from 63% to 79%. A waiting time histogram is shown in Figure 7.

5. Multiple request processes

A TSP system modifies a normal signal operation cycle to better accommodate transit vehicles. After a vehicle has gone through an intersection, the traffic controller needs to be notified so that the normal signal cycle can be restored (Li et al, 2005; Liu, Lin and Tan, 2007). Another request process is needed to establish this communication (NTCIP, 2005). This is called the checkout request process. The request is made as soon as the transit vehicle has passed the intersection and needs to be processed within a specified amount of checkout time. This additional process will increase the overall demand, and consequently, the queue size and average queue waiting time. In fact, the number of calls made is doubled.

The supply-demand analysis in the previous sections is based on a set of deterministic arrival schedules at the intersections. In reality the arrival times are random and must be estimated using a TTA prediction algorithm (Tan et al, 2006). The TTA prediction has different statistical variations for different links connecting successive intersections. As a transit vehicle approaches its next intersection, the standard deviation of the prediction error becomes smaller and eventually converges to zero. So the predicted arrival time announced at the time of the initial request is likely to be inaccurate, with a fairly large
expected error. An unreliable TTA prediction is less useful for determining how a signal cycle should be modified to facilitate the passage of a transit vehicle. Therefore, a more accurate prediction needs to be announced in a second request process when the vehicle is closer to the intersection and the statistical errors in prediction are smaller. To continue with the analysis, we consider three types of request processes. A first request process begins when a transit vehicle is farther away from its next intersection, and a second request process begins when the vehicle is closer to the intersection. The initial request times for these two processes will affect the effectiveness of a TSP system (Liu et al, 2004). Finally, a checkout request process begins when the vehicle passes the intersection. Similar to that for a single request process, a request is repeated until it is processed before the deadline for each of the three types of request processes.

**Rule 2:** Request times for first and second request processes: Let \( t \) be the current time and \( T_A(t) \) be the TTA computed at the current time. Let \( T_1, T_2 \) be two fixed “look-ahead” times. For the first request process, a priority request can be made if \( T_A(t) - T_1 \leq T_2 \). For the second request process, a priority request can be made if \( T_A(t) \leq T_2 \).

**Rule 3:** Request times for a checkout request process: Let \( T_{out} \) be a fixed “checkout” time constant. The initial request of a checkout process is made when a transit vehicle arrives at an intersection at a time \( \hat{t} \) (that is, \( T_A(\hat{t}) = 0 \)). A checkout request can be made before the process deadline, which is \( \hat{t} + T_{out} \).

The initial request times (or call times) of the first and second request processes are \( \hat{t}_1 \) and \( \hat{t}_2 \), respectively. Note that \( T_A(\hat{t}_1) = T_1 + T_2 \) and \( T_A(\hat{t}_2) = T_2 \). The first process has a deadline at time \( \hat{t}_2 = \hat{t}_1 + T_1 \), while the second process has a deadline at time \( \hat{t} = \hat{t}_2 + T_2 \). The timelines for the three processes are illustrated in Figure 8.

It is reasonable to require that a second priority call be processed earlier than a first priority call if they are made at the same time, since the data message in the first priority call is less accurate. Also it is desirable to assign a high priority to a checkout call because of the need to restore a signal cycle to its normalcy quickly and smoothly, with minimal delays on the cross street. By using the queue concept discussed in §2, these requirements can be realised via the channel establishment probability. If two processes are of the same type (either first, second or checkout), the one with an earlier deadline is processed first (see §2). We summarise these requirements below.

**Rule 4:** Process priority of request processes: If the three types of request processes are initiated at the same time, the checkout call will be processed first, followed by the second call and then the first call, regardless of their process deadlines. So the probability of establishing communication is the highest for the checkout call, followed by those for the second and first calls. Two processes of the same type are processed in order of their deadlines.
We illustrate the requirements with an example in Table 2. The probability of channel establishment is the highest for the checkout process. If the arithmetic rates for the first, second, and checkout processes are $k_1n,k_2n,k_3n$, where $k_1 < k_2 < k_3$, then the probability for the checkout process is $k_3 / (k_1 + k_2 + k_3)$. This probability increases with $k_3$. So we can make $k_3$ large if it is highly desirable to first process a checkout call. Other schemes such as geometric rates of $2^{n-1},2^n,2^{n+1}$, can also be used.

### Table 2 Realisation of process priority for multiple request processes via the probability of channel establishment

<table>
<thead>
<tr>
<th></th>
<th>Number of requests per second</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1&lt;sup&gt;st&lt;/sup&gt; sec</td>
</tr>
<tr>
<td>veh. #1, checkout call</td>
<td>1</td>
</tr>
<tr>
<td>veh. #2, second call</td>
<td>1</td>
</tr>
<tr>
<td>veh. #3, first call</td>
<td>1</td>
</tr>
<tr>
<td>veh. #4, first call</td>
<td>-</td>
</tr>
</tbody>
</table>

We analyse a supply-demand scenario that involves multiple request processes. The process rate is two per second. A transit vehicle makes its first call when $T_1(\hat{t}_1) = T_1 + T_2 = 40$ sec, and makes its second call when $T_1(\hat{t}_2) = T_2 = 20$ sec. The checkout process begins as soon as the vehicle arrives at the intersection and needs to be completed in $T_{out} = 10$ sec. In this scenario, there are 827 dropped first request processes (or calls), but no dropped second or checkout calls. The high number of dropped first calls is a result of the process priority imposed in Rule 4. If we reverse the priority, then all the dropped calls are checkout calls. As shown in Figure 9, there are 36884 seconds during 5am-10pm (61200 sec) when the channel is processing at its full capacity (two requests per second), a 60.27%. The probability of a dropped (first) call is 0.955%, but they occur mainly during the peak hours, 7-8am and 3-4pm. The dropped calls in these hours account for 65% + 31% = 96% of the total dropped calls. In these hours, the probabilities of a dropped call are 21% and 11%, respectively. The hourly distribution of drops is shown in Figure 10. If first calls are made earlier at $T_1(\hat{t}_1) = 50$ sec, the number of dropped first calls decreases to 763, a reduction of about 7.74%.

There can be as many as 64 unprocessed calls in the queue. This queue is non-empty with a probability of 39.31%. If we zoom into the two peak hours, 7-8am and 3-4pm, the probabilities are very high at 76% and 68%, respectively. An hourly distribution of the frequency of non-empty queue is shown in Figure 11. There are no first calls in the queue for ~78% of the time in 5am-10pm, with the queue being empty ~60% of the time (see Fig. 11). During the two peak hours, roughly 65-70% of the calls in the queue are first calls. The average queue size is 1.477 and the average unprocessed first call is 0.837. This is consistent with the observation that the queue is empty for ~60% of the time (see Fig. 11) and most unprocessed calls are first calls. A histogram of the first calls in the queue is shown in Figure 12. This distribution is skew with a small variance.
The queue waiting time for the processed first call is shown in Figure 13. As expected, the probability that a type of call is not processed immediately (with queue waiting time > 0) decreases with the priority of the type of call. The same is true for the average waiting time. The average queue waiting time for processing a first call is 3.17 sec, but this average decreases to 1.25 sec for a checkout call which has the highest priority.

7. On-time performance measure

The location of a transit vehicle is made available by means of an AVL system installed on-board of the vehicle. The GPS data includes the absolute position of the vehicle and its corresponding coordinated universal time. When a transit vehicle is in a vicinity\(^2\) of a signalised intersection, the AVL system compares the predicted arrival time with its scheduled arrival time and report whether the vehicle is “late”, “normal” or “early”. Time windows are defined for these on-time performance measures. These performance data are recorded, transmitted to the TMC using the polling channel when the transit vehicle is polled, and stored in a database. The data for each route can be grouped by the times-of-day (peak, off-peak) and by seasons. If a vehicle is predicted to be on-time (normal or early), no signal priority is needed. The statistics of these historical data can therefore be used to further analyse and understand how much priority request demand could be reduced when there are actually high probabilities that some transit vehicles will not need signal priority at some intersections and at certain times-of-day.

We simulate a scenario when transit vehicles are on time and do not request for signal priority for 20% of the time. In the simulation, a random number uniformly distributed on \([0, 1]\) is generated for scheduled arrival. If it is less than 0.2, there is no need for signal priority. Consider a sample run where signal priority is not needed at 5837 intersections that were randomly and uniformly chosen from the total 28872 intersections. This accounts for 20.22% of all the priority requests. Figure 14 shows that there are only 44 dropped calls, which is a significant reduction compared to that depicted in Figure 10 (827 dropped first calls) when on-time performance was not considered. Also the ratio of first calls dropped over those made is only 0.35% for 7-8am. It was 21.54% if on-time performance was not considered (see Fig. 10). A histogram of the first call queue size is shown in Figure 15. If we compare this to Fig. 10, the queue length is obviously much shorter, and the average number of first calls in the queue is also smaller. The average waiting time for processing a first call is shorter and a higher percentage of them are processed immediately (with no waiting time).

These statistics are for one sample run. We need to simulate more runs to obtain a better approximation of an “average” scenario and account for variations in the scheduled travel times between successive intersections. One can employ a variance reduction technique such as the Monte Carlo Method. We believe it is highly probable that priority is needed only during peak hours. If this hypothesis is true, the request demand would be even lower than the 20% on-time performance scenario we just simulated, making it possible to process all the requests with only a small number of dropped calls.

8. Concluding remarks and future work

\(^2\) The exact radius of this neighbourhood also depends on the statistical variance of the GPS error in the AVL data.
We have formulated and analysed the problem of establishing communication between transit vehicles and TMC for signal priority as a supply-demand problem. In this problem there is a potentially large number of priority requests made at the same time, and a limited number of time slots per second available to process these requests. The demand for signal priority depends on the time it is requested and the time it will actually arrive at the next intersection. The arrival time is stochastic and needs to be predicted. In our model, we consider deterministic scheduled arrival times. These schedules allow us to estimate an “average” demand profile for signal priority at each intersection. On the supply side, the capability to process priority requests is limited by the polling rate of the polling channel and the process rate of the contention channel. We have considered only the contention channel. The attributes in the model are: (1) time-dependent priority request, (2) an “average” demand profile with deterministic arrival times, (3) channel process rate, (4) process priority with respect to process deadlines (see §2), (5) process priority for multiple request processes (see §5), (6) “uniform” on-time performance measures of arrival times (see §7). The idea of process priority is realised via probability of channel establishment.

Our model serves as a framework for constructing a more complex model that captures the realistic data structure, such as statistical uncertainties in arrival times. The key additional attributes in the extended model are: (2E) statistical variations of arrival times, (6E) on-time performance measures for different routes, links, times-of-day, and seasons, (7) polling schedule. For a link with small variance, the corresponding set of request processes should have higher priority. This can be another guiding principle in revising the model. With this model, an efficient scheduling protocol could be designed to coordinate the request processes and meet signal priority demand.

The queue size and queue waiting time are sensitive to the channel process rate. The sensitivity to the call time (time of initial request) is approximately concave, with little or no sensitivity if the look-ahead time is larger than the longest queue waiting time. If there are multiple request processes, process priority (see §5) is critical. Dropped calls most likely come from calls with the lowest priority, especially in peak hours. If on-time performance measures are included, there are significantly less dropped calls. So the channel can process requests from a larger network of in-service vehicles, if necessary. We expect the need for signal priority is much lower in off-peak hours when transit vehicles are more likely to be on time, thus further reducing the drops.
References


**Figure 1** Timeline of a single priority request process

- **Channel access granted**
- **Queue waiting time**
- **Request repeated before deadline**

**Figure 8** Timelines for multiple priority request processes

- **1st call time**
- **2nd call time = 1st process deadline**
- **Checkout call time = 2nd process deadline**
- **Checkout process dropped**
- **Vehicle arrives at intersection**
Appendix: Simulation results

**Figure 2** Number of buses in active service for 30 TSP bus routes

Active fleet size for 30 TSP bus routes

max = 67, mean = 26.7775
max occurs at 07:20:00

**Figure 3** Distribution of number of calls (initial requests) made in each hour

Notes: (1) The maximum occurs in the 8th hour, 7-8am, during which the buses in the network go through 2479 intersections (so there are 2479 calls / initial requests). In this hour, probability of no call in a sec = 189403000 = 55.39%. (2) Total for 24 hours = 339552.
**Figure 4** Histogram of number of requests processed per sec, $T = 30$ sec

Distribution of no. of requests processed per sec for 5am-10pm, calls made @ TTA = 30 sec, process rate = 2/sec

Note: Total no. of requests processed = 12,252 + (6810 × 2) = 26,872, which equals the total no. of calls (initial requests) made (i.e. no dropped calls).

**Figure 5** Histogram of queue size per sec, $T = 30$ sec

Distribution of queue size for 5am-10pm, calls made @ TTA = 30 sec, process rate = 2/sec

Notes: (1) empty queue occurs for 59007 sec in 5am-10pm (96.42%)
(2) no dropped calls (i.e. calls in the queue are processed before the scheduled arrival times at their next intersections)
Figure 6 Histogram of queue waiting time, $T = 30$ sec

Figure 7 Histogram of queue waiting time, process rate = 1/sec, $T = 30$ sec
Figure 9  Histogram of number of requests processed per sec; multiple requests @ {40, 20, 10}

Figure 10 Distribution of no. of dropped (first) calls by hour; multiple requests @ {40, 20, 10}
Figure 11 Hourly distribution of non-empty queue; multiple requests @ \{40, 20, 10\}

**Notes:**
1. The queue is non-empty for 24059 sec in 5am-10pm. So the percentage of time when the queue is non-empty = \(24059/81200 = 0.298\%\).
2. Peak 5th hour, 7-8am: the queue is non-empty with a probability = \(2732/81200 = 0.337\%\).
3. Peak 18th hour, 3-4pm: the queue is non-empty with a probability = \(2444/81200 = 0.372\%\).

Figure 12 Histogram of first call queue size; multiple requests @ \{40, 20, 10\}

**Notes:**
- There are no first calls in the queue for 48255 sec in 5am-10pm (74.8\%).
- Ave. no. of first calls in the queue = 0.837.
- Maximum first call queue size = 81, with qsize[i] = (i = 20, i = 23, ... , 81).
Figure 13 Histogram of first call waiting time; multiple requests @ \{40, 20, 10\}

Figure 14 Hourly distribution of dropped calls, 20% no need for priority
Figure 15 Histogram of first call queue size, 20% no need for priority

Histogram of no. of first calls in the queue, 5am-10pm, 20% no need for priority (run #1),
calls made @ TTA = 40 & 20 sec, checkout = 10 sec, process rate = 2/sec

There are no first calls in the queue for 63408 sec in 5am-10pm (87.27%).

Ave. no. of first calls in the queue = 0.250.