2011 50th IEEE Conference on Decision and Control and
European Control Conference (CDC-ECC)
Orlando, FL, USA, December 12-15, 2011

Pull Protocols for Communication Constrained Advanced Traveler Information Systems

Rohit Kumar and David Castañón

***Abstract*— The advanced traveler information provides real-time information about estimated time of arrival for the bus to the commuters. The bus transmits its location information to the base station frequently and then base station uses this available location information and advanced prediction algorithms to estimate the time of arrival for rest of the stops on the itinerary. This frequent communication between the bus and base station adds a recurring cost to the implementation and management of such a system. In this paper we wish to study performance of the system wherein the bus transmits its information only when it is pulled by base station. Algorithms that determine when to request information are developed for the two scenarios: one where motion on each link on the road network is modeled with a single common statistical model, and another where each link on the road has an individual model for vehicle motion. The second scenario can generate more accurate predictions of vehicle motion, but the algorithms for determining when to request information from the vehicles are more complex. The resulting algorithms are evaluated using Monte Carlo simulations to illustrate the performance of the approaches.**

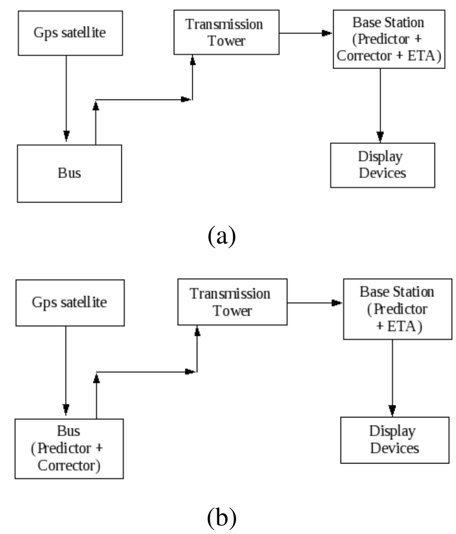


Fig. 1. Figure showing the block diagram of the ATIS system. The predictor and corrector are the predict and update steps of the conventional filter. ETA are the algorithms that are used to estimate the expected time of arrival for the future stops on the itinerary. (a) Existing System (b) Proposed System.

I. INTRODUCTION

One of the main reasons people avoid public transport is due to perceived wait time, caused by uncertainty in the time of arrival of the bus. Recently, there has been an increased interest in encouraging the use of public transport [1], to reduce oil consumption and traffic. Currently, transit authorities make the time of arrival information available to the commuters via printed schedules that are posted on the bus stops and other information dissemination mechanisms such as website, mobile phones, schedule pamphlets, etc. The problem with these methods is that the schedules are static and are not updated as the bus traverses its trip and faces new traffic conditions. The main function of an advanced traveler information system is to update dynamically the arrival time of the bus [2], [3], [4].

For an ATIS, each transit vehicle has a GPS receiver and a transmitter that transmits the location information of the vehicle to the base station [5], [6]. The base station then uses this location information and advanced algorithms to estimate the time of arrival (ETA) for rest of the stops on the itinerary [7], [8], [9], [10], [11]. A block diagram of the existing ATIS is shown in the Fig. 1(a).

As shown in the block diagram the predictor uses the location information of the bus to predict the estimated time of arrival for the subsequent stops on the route. Existing methods employ protocols of transmissions that are fixed a priori. Most of the existing system transmits the location information either every k^{th} time instant or after every m meters [12], [13]. The problem with these protocols is that are not adaptive, and they do not account for the traffic condition and the time of journey. For example if the bus is running when there is less traffic on the road, the number of communications to the base station can be reduced significantly. This frequent communication is one of the main hurdles for implementation of such systems, especially in developing countries. Another point is that the communication cost is not just one-time cost but a recursive one. In fact one transit authority (TriMet in Portland, Oregon) that did not account for the communication cost in its planning had to scale back the deployment due to the cost [1]. Alternative approaches that propose wire line communications have limitations because there is little flexibility in changing routes or bus stops, and can run into significant maintenance costs because of the distributed deployment of hardware.

In our previous work [14], we have developed stochastic models of bus motion in urban traffic, along with nonlinear estimation algorithms that allow the bus to use route maps along with traffic models and real-time position measurements such as GPS to estimate the bus state. Subsequently, in [15], we developed algorithms that select times at which the bus can decide to communicate its estimate to the

D. Castañón and R. Kumar are with the Department of Electrical Engineering, Boston University

This work was supported by AFOSR grants FA9550-07-1-0361 and by ODDR&E MURI Grants FA9550-06-1-0324 and FA9550-07-1-0528

base station, in order to keep the divergence between the information contained at the base station and the information at the bus small while limiting communications. These algorithms correspond to push protocols, as the recipient of new information (the bus) determines the times at which to share this information with the base station.

In this paper, we consider a different approach, where the base station decides when information needs to be communicated; we refer to such operation as pull protocols. We propose and analyze pull protocols for collecting information to predict the bus schedule based on communicated measurements collected by the bus. We use a hybrid dynamical model of vehicle motion presented in [14], together with nonlinear estimation algorithms to estimate the bus states. Based on the predicted gain in information quality, the base station determines whether an update is needed and selects the times to communicate.

The rest of this paper is organized as follows: Section II presents background results, including models of vehicle motion on links and state estimation algorithms. Section III describes the problem of selecting which times the base station requests reports from the bus. In sections IV and V, we present our proposed algorithms for determining request times for reports, and section V has an illustrative simulation example. Section VII has concluding remarks.

II. BACKGROUND

In this section we present the model that is used to define the dynamics of bus on the road and the estimation algorithm that will be used for tracking the motion of the bus, based on our previous modeling efforts [14]. We assume the route of the bus is known a priori and is divided into links $l = 1,\dots,N$. Each link will be modeled with different dynamic model that governs the evolution of the continuous state, described later. Specifically, the model at time k on link $l(k)$ in state space form is:

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{F}_{l(k)}\mathbf{x}(k) + \mathbf{W}_{l(k)}(k) && (1)\\ \mathbf{y}(k) &= \mathbf{h}_{l(k)}(\mathbf{x}(k)) + \mathbf{V}_{l(k)}(k) && (2)\\ l(k) &= g_1(\mathbf{x}(k)) && (3) \end{aligned}$$

where suffix $l(k) \in \{1,\dots,N\}$ indicates the currently active link, $\mathbf{F}_{l(k)}$ is a matrix indicating linear dynamics, $\mathbf{h}_{l(k)}$ is a nonlinear measurement equation, and g_1 is an integer valued map that identifies the current link at time k. The noise processes $\mathbf{W}_{l(k)}(k)$ and $\mathbf{V}_{l(k)}(k)$ are link dependent white Gaussian noise with zero mean and variance $\mathbf{Q}_{l(k)}(k)$ and $\mathbf{R}_{l(k)}(k)$, respectively.

A. Model

The proposed model assumes that each link has different average velocity. We model the velocity of the vehicle by *Ornstein-Uhlenbeck* process, which is specified as

$$dV_t = \lambda(V_t - V_0) + \sigma dW_t \tag{4}$$

dW_t is the differential of the Brownian motion. In (4), V_t is the velocity at time t, $-\lambda > 0$ is the rate of convergence to average speed, and V_0 is the average velocity of the link. The state of the vehicle on the link consists of its position along the trajectory and the vehicle speed. Discretizing the continuous dynamics with time step h yields the stochastic motion model for the link as:

$$\mathbf{x}(k+1) = \mathbf{F}^{(h)}_{l(k)}\mathbf{x}(k) + \mathbf{B}^{(h)}_{l(k)} u_{l(k)} + \mathbf{W}^{(h)}(k) \tag{5}$$

where $\mathbf{F}^{(h)}_{l(k)} = e^{\tilde{\mathbf{F}}_{l(k)}h}$, $\mathbf{B}^{(h)}_{l(k)} = \int_0^h e^{\tilde{\mathbf{F}}s}\tilde{\mathbf{B}}ds$, h is the sampling interval, and $\mathbf{W}^{(h)}_{l(k)}(k)$ is white Gaussian noise with zero mean and variance $\mathbf{Q}^{(h)}_{l(k)}$. The expressions for $\mathbf{F}^{(h)}_{l(k)}, \mathbf{B}^{(h)}_{l(k)}$ and $\mathbf{Q}^{(h)}_{l(k)}$ are given below.

$$\begin{aligned} \mathbf{F}^{(h)}_{l(k)} &= e^{\tilde{\mathbf{F}}_{l(k)}h} = I_{2\times 2} + \tilde{\mathbf{F}}_{l(k)}\frac{(e^{\lambda_{l(k)}h}-1)}{\lambda_{l(k)}} \\ \mathbf{B}^{(h)}_{l(k)} &= \int_0^h e^{\tilde{\mathbf{F}}_{m(t)}s}\tilde{\mathbf{B}}ds = \begin{bmatrix} \frac{1}{\lambda^2_{l(k)}}(e^{\lambda_{l(k)}h}-1) - \frac{h}{\lambda_{l(k)}} \\ \frac{1}{\lambda_{l(k)}}(e^{\lambda_{l(k)}h}-1) \end{bmatrix} \end{aligned}$$

$$\mathbf{Q}^{(h)}_{l(k)}(1,1) = \frac{h}{\lambda^2_{l(k)}} + \frac{1}{\lambda^3_{l(k)}}\left(\frac{e^{2\lambda_{l(k)}h}-1}{2} - 2(e^{\lambda_{l(k)}h}-1)\right)$$

$$\mathbf{Q}^{(h)}_{l(k)}(1,2) = \mathbf{Q}^{(h)}_{l(k)}(2,1) = \frac{1}{\lambda^2_{l(k)}}\left(\frac{e^{2\lambda_{l(k)}h}-1}{2} - (e^{\lambda_{l(k)}h}-1)\right)$$

$$\mathbf{Q}^{(h)}_{l(k)}(2,2) = \frac{e^{2\lambda_{l(k)}h}-1}{2\lambda_{l(k)}}$$

$$\mathbf{Q}^{(h)}_{l(k)} = \begin{bmatrix} \mathbf{Q}^{(h)}_{l(k)}(1,1) & \mathbf{Q}^{(h)}_{l(k)}(1,2) \\ \mathbf{Q}^{(h)}_{l(k)}(2,1) & \mathbf{Q}^{(h)}_{l(k)}(2,2) \end{bmatrix} \times \sigma^2$$

where $\tilde{\mathbf{F}}_{m(t)} = \begin{bmatrix} 0 & 1 \\ 0 & \lambda_{m(t)} \end{bmatrix}, \tilde{\mathbf{B}} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$, $l(k)$ indicates the link number ($l(k) \in \{1,\dots,N\}$), $u_{l(k)} = -\lambda_{l(k)}V_{0l(k)}$ ($V_{0l(k)}$ is the average velocity of the $l(k)^{th}$ link) and $\lambda_{l(k)}$ is the rate constant for link $l(k)$).

The above model describes the dynamics on a link. As the state transitions from one link location to another, different discrete dynamics are applied, leading to a stochastic hybrid model with state-dependent switching. Additional details on the model above can be found in [14], [16].

B. Estimation Algorithm

In [14], we explored alternative estimation algorithms for determining the current state of the bus given periodic reports of GPS measurements from the bus. In this paper, we use a modification of the Extended Kalman filter (EKF) proposed in [14] for the system given in (1)) in order to generate estimates of the current bus state. The extension uses a two-step prediction approach to accommodate the switching between links, as described below:

Assume that at step $k-1$, the EKF updated estimate is $\mathbf{x}(k-1|k-1)$ and its updated covariance $\mathbf{P}(k-1|k-1)$. The first step is to compute the expected time remaining to switch links, τ_s, by predicting the updated estimate using the process model corresponding to the link exit time, then using the model for the next link to predict until the next sampling time. We approximate the predicted link exit time using the straightforward constant velocity approximation, accurate when the state is close to the exit of the link:

$$\tau_s = \frac{C - x_1(k-1|x-1)}{x_2(k-1|k-1)}$$

where C is the location of the switching corner, and $\mathbf{x}(k-1|k-1) = [\ x_1(k-1|k-1) \quad x_2(k-1|k-1)\ ]^T$. Then, if $\tau_s \geq h$, the prediction algorithm uses the current link model and generates a standard EKF prediction. In the case where $\tau_s < h$, the prediction uses a two-step process, where prediction from $t = kh$ to $kh + \tau_s$ uses the model from the first link, and prediction from $kh + \tau_s$ to $(k+1)h$ uses the model from the subsequent link, where the discrete model matrices are adjusted appropriately to the size of the prediction intervals. The update of the EKF remains the same. The relevant equations are summarized below.

Prediction Equations

$$function[\mathbf{x}(k|k-1), \mathbf{P}(k|k-1)] = \\ Predict\{\mathbf{x}(k-1|k-1), \mathbf{P}(k-1|k-1)\}$$

One step prediction equations (Away from corner):

$$\mathbf{x}(k|k-1) = \mathbf{F}^{(h)}_{l(k)}\mathbf{x}(k-1|k-1)$$
$$\mathbf{P}(k|k-1) = \mathbf{F}^{(h)}_{l(k)}\mathbf{P}(k|k-1)\mathbf{F}^{(h)T}_{l(k)} + \mathbf{Q}^{(h)}_{l(k)}$$

Two step prediction equations (Close to the corner)

$$\mathbf{x}(\tau_s|k-1) = \mathbf{F}^{(\tau_s)}_{l(k)}\mathbf{x}(k-1|k-1)$$
$$\mathbf{P}(\tau_s|k-1) = \mathbf{F}^{(\tau_s)}_{l(k)}\mathbf{P}(k|k-1)(\mathbf{F}^{(\tau_s)})^T_{l(k)} + \mathbf{Q}^{\tau_s}_{l(k)}$$
$$\mathbf{x}(k|k-1) = \mathbf{F}^{(h-\tau_s)}_{l(k)+1}\mathbf{x}(\tau_s|k-1)$$
$$\mathbf{P}(k|k-1) = \mathbf{F}^{(h-\tau_s)}_{l(k)+1}\mathbf{P}(\tau_s|k-1)(\mathbf{F}^{(h-\tau_s)})^T_{l(k)+1} + \mathbf{Q}^{(h-\tau_s)}_{l(k)+1}$$

Update Equations:

$$function[\mathbf{x}(k|k), \mathbf{P}(k|k)] = \\ Update\{\mathbf{x}(k|k-1), \mathbf{P}(k|k-1), \mathbf{y}(k)\}$$

$$\hat{l}(k) = g(\mathbf{x}(k|k-1)$$
$$\mathbf{H} = \frac{\partial}{\partial \mathbf{x}} h_{\hat{l}(k)}(\mathbf{x}(k|k-1))$$
$$\mathbf{innv}(k) = \mathbf{y}(k) - \mathbf{h}_{\hat{l}(k)}(\mathbf{x}(k|k-1))$$
$$\mathbf{S}(k) = \mathbf{H}\mathbf{P}(k|k-1)\mathbf{H}^T + \mathbf{R}(k)$$
$$\mathbf{K}(k) = \mathbf{P}(k|k-1)\mathbf{H}^T\mathbf{S}^{-1}(k)$$
$$\mathbf{x}(k|k) = \mathbf{x}(k|k-1) + \mathbf{K}(k)\mathbf{innv}(k)$$
$$\mathbf{P}(k|k) = (\mathbf{I} - \mathbf{K}(k)\mathbf{H})\mathbf{P}(k|k-1)$$

III. Problem Statement

As shown in Fig.1(a), the bus collects the GPS information and transmits it to the base station. The base station then uses the received observation along with an estimation algorithm such as the modified Extended Kalman Filter above to estimate the current location of the bus. This location information is then used by the ETA algorithms to predict the time of arrivals for rest of the stops on the itinerary. In this paper, we use the architecture proposed in [15] as in Fig.1(b). The architecture moves the estimation algorithm from the base station to the bus while retaining a copy of the predictor at the base station. The predictor at the base station is used to update commuters as to expected bus arrivals. We assume the bus is collecting periodic location measurements and using them to update its state estimate. We want to define the problem of when these state estimates should be communicated to the base station, trading off frequency of communication for accuracy of prediction.

Let $P_G(k)$ be the conditional distribution of the bus state given the information available at the base station at time k, and $P_L(k)$ be the conditional distribution of the bus state given all the measurements available at the bus. If j was the last communication time from the bus, then $P_G(k) = p(\mathbf{x}(k)|\mathbf{Y}^{[0,j]})$, i.e., conditional distribution of state $\mathbf{x}$ at time k given the measurements from time 0 to j $(j \leq k)$, and $P_L(k) = p(\mathbf{x}(k)|\mathbf{Y}^{[0,k]})$. Since we are using a modified EKF estimator as described above, we approximate the conditional distributions $P_G(k)$ and $P_L(k)$ by Gaussian distributions, resepresented as $\mathscr{N}(\mathbf{x_G}(k), \mathbf{P_G}(k))$, and $\mathscr{N}(\mathbf{x_L}(k), \mathbf{P_L}(k))$ respectively.

Let $u(k)$ be the time-varying control signal that indicates the times at which the bus is requested to communicate its estimate:

$$u(k) \quad = \quad \begin{cases} 0 & \text{No Transmission;} \\ 1 & \text{Transmission;} \end{cases}$$

If $u(k) = 1$, the base station pulls the most recent information from the bus and $P_G(k)$ is updated as $P_G(k) = P_L(k)$. The goal of the problem is to keep the difference between $P_G(k)$ and $P_L(k)$ small while using limited communications. Further, since base station does have not have access to the information available at bus (in between communications), the pull protocols schedules the transmission solely based on $P_G(k)$. In particular, we shall use the trace of the covariance of $P_G(k)$ as the cost function.

The problem for scheduling transmission for pull protocols is given as:

$$\min_{\gamma \in \Gamma_M} \sum_{k=1}^{T} trace(\mathbf{P_G}(k)) \quad s.t. \sum_{k=1}^{T} u(k) \leq M \tag{6}$$

with

$$\begin{aligned} P_G(k) &= Predict\{P_G(k-1)\} \quad if \quad u(k) = 0; \\ &= P_L(k) \qquad\qquad\qquad\ \ if \quad u(k) = 1. \end{aligned} .$$

where $\Gamma_M = \binom{T}{M}$ is the set of admissible policies for M resources. Note that the base station does not have access to $P_L(k)$ unless there is communication. We assume that the local estimator on the bus has achieved a steady-state updated estimate of the error covariance L_{s_l} of the link. For linear observation models, this steady state covariance can be computed a priori. We model the covariance dynamics at the base station when there is no update from the bus using the linear dynamics model of the link. The optimization problem in (6) becomes a deterministic optimization problem over the evolution of the predicted covariance matrix, with simple dynamics given by:

$$\mathbf{P}_G(k) = \begin{cases} L_{s_l} & \text{if } u(k) = 1 \\ \mathbf{F}^{(h)}_{l(k)}\mathbf{P}_G(k-1)(\mathbf{F}^{(h)}_{l(k)})^T + \mathbf{Q}^{(h)}_{l(k)} & \text{otherwise} \end{cases} \tag{7}$$

At communication times, the conditional mean and covariance of the local nonlinear estimator of the bus is communicated to the base station.

IV. Policy for a Single Link Model

Consider the problem when the bus travels according to a common dynamical model on all links, so there are no switching dynamics. The evolution of the covariance at the base station will be given as:

$$\mathbf{P}_G(k) = \begin{cases} L_s & \text{if } u(k)=1 \\ \mathbf{F}^{(h)}\mathbf{P}_G(k-1)(\mathbf{F}^{(h)})^T + \mathbf{Q}^{(h)} & \text{otherwise} \end{cases} \quad (8)$$

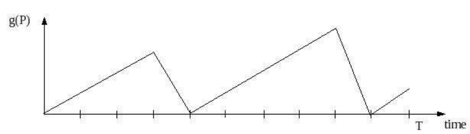
We solve this via dynamic programming. Let $V(L_u, M, T)$ is the optimal solution with M available resources, T time horizon, and L_u is the starting covariance matrix. Let γ_{opt} be the corresponding optimal policy for the following optimization problem.

$$V(L_u, M, T) \quad = \quad \min_{\gamma \in \Gamma_M} \sum_{k=1}^{T} tr(\mathbf{P_G}(k)) \ s.t. \sum_{k=1}^{T} u(k) \leq M \quad (9)$$

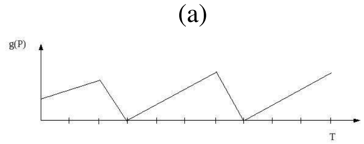
where $\mathbf{P}_G(1) = L_u$ (unknown covariance matrix) , $u(k) \in \{0,1\}$ and $\mathbf{P}_G(k)$ is updated with the steady state covariance L_s if the information is pulled as in (8).

A. Starting with steady state covariance

A basic building block is to consider the above problem when $L_u = L_s$. For this case, we have a simple characterization of the optimal policy:



(a)



(b)

Fig. 2. Changes in the cost function with respect to time. In between communications the cost function is generally increasing and reduces to a specific value on communications. (a) Cost function with communications at time 0 . (b) Cost function with no communications at time 0.

Theorem: For the problem given in (9) with $\mathbf{P_G}(1) = L_s$, the best communication policy is to divide the total time period T into $M+1$ windows such that there a_1 windows of size b_1 and a_2 windows of size b_2, where

$$a_1 \quad = \quad T - (M+1) \times \left\lfloor \frac{T}{M+1} \right\rfloor ; \quad a_2 = M+1-a_1$$
$$b_1 \quad = \quad \left\lfloor \frac{T}{M+1} \right\rfloor + 1; \quad b_2 = \left\lfloor \frac{T}{M+1} \right\rfloor$$

This result is a consequence of the prediction error dynamics starting from the filter steady state covariance. This guarantees the following inequality:

$$trace(\mathbf{P_G}(k+1)) \geq trace(\mathbf{P_G}(k))$$

Hence, it is more efficient to sample as uniformly as possible. The detailed proof of the theorem is omitted here. Interested readers are referred to [16] for more details.

We illustrate this result with an example. Let us assume that $T = 27, M = 4$. Consequently

$$a_1 = 27 - 5 \times \left\lfloor \frac{27}{5} \right\rfloor = 2; \ a_2 \qquad = M+1-a_1 = 3$$
$$b_1 = \left\lfloor \frac{27}{5} \right\rfloor + 1 = 6; \ b_2 \qquad = \left\lfloor \frac{27}{5} \right\rfloor = 5$$

So there are two windows of length six and three windows of length five. Let us denote $P^{[1-K]} = \sum_{i=1}^{K} trace(\mathbf{P_G}(i))$. The total cost for the numerical problem above with the proposed transmission policy will be given as:

$$V_{\gamma_{opt}} \quad = \quad 2 \times P^{[1-6]} + 3 \times P^{[1-5]}$$

Let $\gamma = [i_1, i_2, \ldots, i_M]$, where i_i denotes a communication time. Note that decision vector is not unique, as there are different combinations of $i_1, i_2, \ldots, i_M$ that produce the above given windows. For the example above, we choose $i_1 = 5, i_2 = 10, i_3 = 15, i_4 = 21$.

To illustrate that the above policy is optimal, consider a new decision vector given as $\gamma = [i_1, i_2, i_3, i_4 + 1]$ so that we decrease the length of the last window from 6 to 5, while increasing the length of the next-to-last window from 6 to 7. Now we show that V_γ is greater than $V_{\gamma_{opt}}$.

$$\begin{aligned} V_\gamma &= P^{[1-7]} + 4 \times P^{[1-5]} \\ &= P^{[1-6]} + 4 \times P^{[1-5]} + trace(\mathbf{P}_G(7)) \\ &\geq P^{[1-6]} + 4 \times P^{[1-5]} + trace(\mathbf{P}_G(6)) \\ &\geq 2P^{[1-6]} + 3 \times P^{[1-5]} = V_{\gamma_{opt}} \end{aligned}$$

Similar inequalities are easily shown for any other policy that does not use the optimal window lengths.

The following properties of the optimal solution are easily established, where $L_b \geq L_a$ means $L_b - L_a$ is a positive semidefinite matrix:

$$\begin{aligned} &\text{If} \quad L_a \leq L_b \text{ then } \quad V(L_a, M, T) \leq V(L_b, M, T) \\ &\text{If} \quad T_a \leq T_b \text{ then } \quad V(L, M, T_a) \leq V(L, M, T_b) \\ &\text{If} \quad M_a \leq M_b \text{ then } \quad V(L, M_a, T) \geq V(L, M_b, T) \end{aligned}$$

B. Starting with any given covariance

The result from last subsection IV-A can be easily extended to a link with any unknown starting covariance matrix, using dynamic programming.

Theorem: For the problem given in (9) with $\mathbf{P_G}(1) = L_u$, the optimal policy is given by

$$\gamma_{opt} = \left[\begin{array}{cc} t_{opt} & \chi_{opt} \end{array} \right]$$

where $\mathbf{e}_{t_{opt}}$ is the optimal time of the first measurement, obtained as the solution of the following optimization problem:

$$\min_{t \in \{1,\ldots,T\}} \left\{ \sum_{i=1}^{t} tr(\mathbf{P}_G(i)) + V(L_s, M-1, T-t) \right\}$$

and χ_{opt} is the the optimum policy for the second term in the above given minimization, for the interval starting at t_{opt} with covariance L_s. The optimum value of the one link problem will be given by

$$V(L_u,m,T) = \left\{ \sum_{t=1}^{t_{opt}} trace(\mathbf{P}_G(t)) + V(L_s,m-1,T-t_{opt}) \right\}$$

The above theorem reduces the optimization problem to a line search plus the solution of the previous problem when the initial covariance was L_s.

V. Pull Protocol for Hybrid Multilink Systems

In this section we develop algorithms for requesting informatio in multilink systems with different models for each link. Define the following variables that will be used in development of the solution.

C_n: Indicates the cost-to-go function at the n^{th} link.

M_n: Indicates the resources available at the n^{th} link to plan for the remaining itinerary.

K_n: Indicates the number of time steps between the switching time to link n and the time that the last communication took place.

The pull protocol problem given in (6) can be rewritten for link-by-link as given below:

$$\min_{\gamma \in \Gamma_M} \sum_{l=1}^{N} \sum_{k=1}^{T_l} trace(\mathbf{P}_G(k))$$
$$s.t. \sum_{l=1}^{N} \sum_{k=1}^{T_l} u(k) \leq M$$

where T_l is the time spent by the bus on the l^{th} link. Note that, in general, T_l will be random, based on the stochastic model used for motion in link l. To define an approximate optimization problem, we compute estimates of T_l by propagating the current state estimate $\mathbf{x}_G(k)$ and observing the times at which the predicted trajectory changes links. Also note that $\Gamma_M = \left[\begin{array}{ccc} \Gamma^1_{m_1} & \ldots & \Gamma^N_{m_N} \end{array} \right]$ with $\sum_{l=1}^{N} m_l = M$, where $\Gamma^l_{m_l}$ is the set of policies for the l^{th} link that has m_l resources. Since we are using the formulation above, it not only reduces the computation cost but also allows us to use the results from earlier sections.

Consider the last link N. We compute the cost-to-go associated with measurements at the last link, $C_N(K_N,M_N)$, as the solution of the problem with initial covariance given by $L_{s_{N-1}}$, propagated using the model for link $N-1$, for K_N steps, with T_N periods to go on link N. This optimal cost is computed for feasible pairs of integers K_N, M_N using the results of the previous subsection. We continue the solution via dynamic programming for previous links as follows. Assume we have computed $C_k(K_k,M_k)$ for $k > n$. Then,

$$C_n(K_n,M_n) =$$
$$\min_{\gamma \in \Gamma^l_m, m} \left\{ \sum_{k=1}^{T_n} tr(\mathbf{P}_G(k)) + C_{n+1}(K_{n+1},M_n-m) \right\} \text{s.t.} \sum_{k=1}^{T_n} u(k) \leq m,$$
$$K_{n+1} = T_n + 1 - \max\{k : u(k) = 1, 1 \leq k \leq T\} \quad (10)$$

where m indicates the resources allocated to n^{th} link, $\mathbf{P}_G(1)$ is computed by predicting $L_{s_{n-1}}$ forward by K_n steps using the model of link $n-1$, and $\mathbf{P}_G(k)$ is updated as in (7).

The recursion is valid for any link except the currently active link. In case of the currently active link, (10) is still valid with a minor caveat about the way $\mathbf{P}_G(1)$ is computed since this covariance is the initial covariance in the problem and is already known.

The above recursion is approximate in that the times T_n were computed by propagating the mean of the state. In order to provide an adaptive algorithm that uses the real-time information provided by the bus, whenever new information is pulled from the bus, the current information at the base station is updated, including a new mean state estimate. This new average state estimate is used to recompute the expected travel times on each link, and to resolve the problem in (10).

Note that the cost-to-go function in (10) not only depends on K_n and M_n but also depends on time spent on the link T_n. One has to compute cost-to-go matrix (C) for all the future links every time a transmission takes place as there will be new predicted T_n. Thus, much effort will be spent computing costs for transitions that are far in future and are hardly accurate. As an alternative, we can assume that vehicle will be traveling with the average velocity of each link for future links. Therefore, the times on each link will be approximated as $T_l = \frac{d(L_l)}{V_{0L_l}}$ where $d(L_l)$ is the length of the link and V_{0L_l} is the average velocity on the link. In this case, the new information will only be used to compute the time spent on current link.

VI. Simulation Example

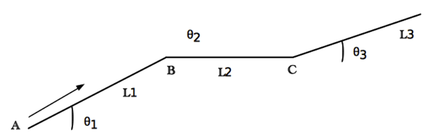
In this section we illustrate the performance of the above algorithms as compared with alternative approaches using simulated examples. In this example we take a linear system of three links as shown in Fig. 3. The dynamics of each link is modeled with process model from section II, with following parameters: L1: $V_0 = 1500, \lambda = -1, \sigma^2 = 300$; L2: $V_0 = 2000, \lambda = -1, \sigma^2 = 450$; L3: $V_0 = 1750, \lambda = -1.5, \sigma^2 = 350$. The distances are measured in feet and time in minutes. We use parameters that are closer to those expected in urban traffic. In Fig. 3, each street has a different elevation ($[\theta_{L_1}, \theta_{L_2}, \theta_{L_3}] = [\frac{\pi}{4}, 0, \frac{\pi}{3}] rads$) with respect to the reference coordinate system and the measurements are made via a GPS receiver. The GPS is assumed to have variance of 500^2 along x and y directions. In addition, we assume that the average velocity in each link is unknown, and must be estimated as part of the state variable $\mathbf{x}(k)$ based on the observations of actual position based on progress against traffic. The observation equation is given as:

$$\mathbf{y}(k) = \mathbf{H}(k)\mathbf{x}(k) + \mathbf{M}(k) + \mathbf{v}(k)$$

where $\mathbf{H}(k) = \left[\begin{array}{ccc} cos\theta_{l(k)} & 0 & 0 \\ sin\theta_{l(k)} & 0 & 0 \end{array} \right]$, and $\mathbf{M}(k) = \left[\begin{array}{c} x^c_{l(k)} - d_{l(k)} cos\theta_{l(k)} \\ y^c_{l(k)} - d_{l(k)} sin\theta_{l(k)} \end{array} \right]$, $d_{l(k)}$ is the sum of link lengths prior to the current link. These equations can be easily derived using geometry [16].

Fig. 4 shows the transmission schedule for the periodic transmission and for the case when transmissions are scheduled using the proposed algorithms in this paper, while the tracking results are given in Fig. 5. We ran 200 Monte Carlo simulations and compared the results. Note that transmission schedule for the pull protocols in this paper is developed at the base station which does not have direct access to the measurements. Consequently, the schedule used in these results is based on the most recent bus state estimate at the last transmission. For analytical simplicity, we assume that when a transmission takes place, the base station error covariance will be updated with the steady state error covariance of the underlying link. The results clearly show that the pull protocols can adapt better to the changing environment as compared to periodic transmission of information. Under normal conditions and single traffic link, a periodic schedule yields best results, as shown in our results, and our proposed protocols also yield the same results.

The gain in accuracy with the proposed communication protocols will be more pronounced in traffic conditions when the algorithms can use transmitted information for re-planning. We have not simulated replanning, as the resulting transmission schedules will be sample-path dependent.



(b)

Fig. 3. Examples Set up. L1,L2 and L3 are the links. A, B and C are the corners with coordinates (x_1^c, y_1^c), (x_2^c, y_2^c), and (x_3^c, y_3^c), respectively. The arrow indicates the direction of travel.

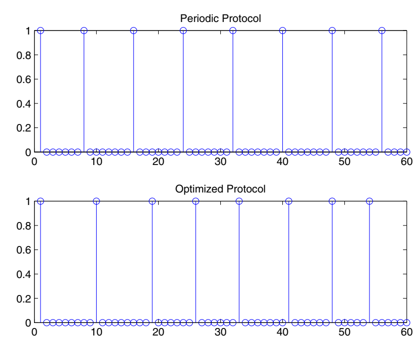


Fig. 4. Transmission Schedule for periodic reporting (top) and planned transmission schedule adapted to different link models (bottom)

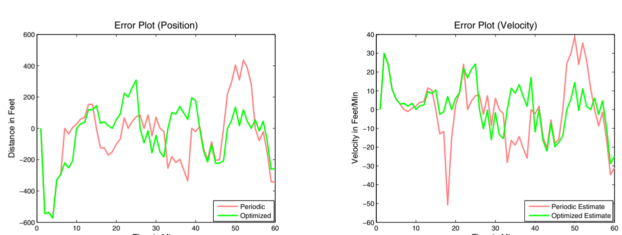


Fig. 5. Error in Position (left) and velocity (right) for periodic (green) and optimized (red)

VII. Conclusion

In this paper, we present an algorithm to schedule requests by a base station for transmissions of state estimates from buses to support an advanced advanced traveler information system. The algorithm is based on a stochastic hybrid system model of bus motion, and uses approximate dynamic programming techniques to formulate an optimization problem for determining a communication schedule. The schedule is recomputed when new information arrives from the bus, as the updated state estimates change the estimated time that is spent on each link of a route. Our simulation results establish that the schedules computed by our algorithm achieve superior performance to periodic reporting schedules.

We are currently conducting further experiments to evaluate the adaptive performance of the algorithm in comparison with other communication protocols. Other future research directions include integration of information from multiple buses in order to refine the models used in prediction of traffic dynamics on each link, and incorporation of models for bus stops as part of the routes.

Type	Mean x	mean v	tr(Variance)	Avg. KL Dist.
KF	207.1365	23.712	0.6847×10^5	-
Periodic	240.379	20.1899	1.8599×10^5	1.3275
Optimized	230.9324	21.6225	2.2830×10^5	1.1616

TABLE I

RMS error in experiment for the different algorithms

References

[1] L. Cham, G. Darido, D. Jackson, R. Laver, and D. Scheneck, "Real-time bus arrival information systems return-on-investment study," *Transportation Research Board*, 2006.

[2] C. Drane and C. Rizos, *Positioning Systems in Intelligent Transportation Systems*. Artech House, 1998.

[3] K. J. Schmier, "Public transit vehicle arrival information system," United States Patent 6006159, December 1999.

[4] C. L. Schweiger, "Real-time bus arrival information systems," *Transportation Research Board, TCRP Synthesis Report 48*, 2003.

[5] R. H. Jeong, "The prediction of bus arrival time using automatic vehicle location systems data," Ph.D. dissertation, TAMU, December 2004.

[6] F. Cathey and D. J. Dailey, "A prescription for transit arrival/departure prediction using automatic vehicle location data," *Transportation Research Part C*, vol. 11, pp. 241–264, 2003.

[7] ——, "An algorithm and implementation to predict the arrival of transit vehicles," *IEEE Intelligent Transportation Systems Conference Proceedings, Dearborn (MI), USA*, 2000.

[8] W.-H. Lin and J. Zeng, "An experimental study on real time bus arrival time prediction with gps data," *Transportation Research Record: Journal of Transportation Research Board*, vol. 1666/1999, pp. 101–109, January 2007.

[9] S. I.-J. Chien and C. M. Kuchipudi, "Travel time prediction with real-time and historic data," *Journal of Transportation Engineering*, vol. 129, no. 6, pp. 608–616, 2003.

[10] R. Jeong and L. R. Rilett, "Bus arrival time prediction using artificial neural networks," *IEEE Intelligent Transportation Systems Conference, Washington, D.C.*, October 2004.

[11] Z. wall and D. J. Dailey, "An algorithm for predicting the arrival time of mass transit vehicles using automatic vehicle location data," *Transportaion Research Board 78th Annual Meeting Washington, D.C.*, January 1999.

[12] "www.nextbus.com."

[13] "http://www.connexionz.co.nz/home."

[14] R. Kumar and D. A. Castanon, "Modeling and tracking of public transit in urban environments," *2010 American Control Conference, Baltimore, MD*, 2010.

[15] ——, "Advanced traveler information system with communication constraints," *American Control Conference*, June 29-July 1 2011.

[16] R. Kumar, "Efficient tracking of public transit in urban environment," Ph.D. dissertation, Boston University, May 2011.