

Real-time Bus Information on Mobile Devices

Stuart D. Maclean, Daniel J. Dailey

Abstract—

We report on a real-time transit vehicle information system which delivers content to Internet-enabled mobile devices, such as cell phones. The content is in the form of predicted arrival/departure times for buses at user-selectable geographic locations within a transit region. We discuss how the physical restrictions of such devices, e.g., screen size and paucity of keyboard options, influences the user interface design. An overview of the system architecture is provided, together with some preliminary usage results.

Keywords— Transit, bus, prediction, mobile device, cell phone, browser, WAP, WML.

I. INTRODUCTION

Recent advances in computing technology have made real-time transit vehicle information systems a viable application area. Such applications utilise transit agency schedules and/or actual vehicle location reports. Their utility can be applied to both fleet management and bus rider information system environments. Applications which exist in this area include graphical positioning of buses on window displays for tracking purposes, alarm notification systems telling riders their bus is close to their bus stop, and airport terminal styled arrival/departure status reports relative to some published schedule.

The Intelligent Transportation Systems research group at the University of Washington has developed a real-time system for predicting bus departure times, based on access to transit agency data. Predictions are made available to the traveling public via a web site known as *MyBus*. More recently, the delivery of the prediction information has been targetted at mobile devices. This category of handheld devices, encompassing mobile phones, personal digital assistants, and pagers, is arguably the most suited to this application area — travel choices are often most crucial when standard (i.e., wired) Internet connectivity is not available. This *mobile MyBus* information system is the project discussed in this paper. It emphasises browsing the MyBus site from a mobile device, such as a cell phone, and not the prediction algorithms themselves.

II. WIRELESS COMPUTING

The number of wireless device subscribers continues to grow. Recent estimates suggest that the figure will reach the billion mark by 2004[1]. Of these, a substantial number will have multimedia capabilities. Already, email and text messaging features are widely available on a variety of personal electronic devices.

Within the wireless communications industry, the WAP Forum[2] has developed the Wireless Application Protocol (WAP) as a defacto standard for the delivery and pre-

sentation of wireless information to mobile devices. The WAP standards ensure that content developers can expect a certain minimum level of functionality on any conforming mobile device, and thus can format content accordingly. A typical “reference device” might have the following characteristics:

- A display area supporting 12 fixed-size characters per row of text, and up to four rows. One row may be reserved for function key labels.
- Support for the ASCII character set, with alphanumeric text input.
- Horizontal and vertical text scrolling and/or wrapping.
- Choice selection via arrow and/or numeric keys.
- Two programmable function keys.

A. Web Integration

Of advantage to web content developers is the fact that wireless communication technology is designed to utilise existing Internet protocols and standards. URL addressing is used for identification of resources, and HTTP is employed as the communications protocol between WAP gateways and content servers. Thus wireless content can be served using existing web server software, such as Apache[3]. Figure 1 shows a typical system architecture for WAP technology. A typical event sequence for downloading data to a mobile device is as follows:

1. The device transmits a message, whose contents includes a regular URL text string, to the WAP service provider’s gateway computer. This wireless transmission uses WAP.
2. The gateway then acts as a regular web client, forwarding the request to the web server identified in the URL. This transmission uses HTTP 1.1 just like any web browser request would.
3. The web server generates and sends an HTTP response back to the requester. The content of the resource identified in the URL is sent to the WAP gateway.
4. The gateway forwards the response to the mobile device using WAP. Metadata in the HTTP response from the web server, such as content cacheability, may be converted to WAP where applicable.

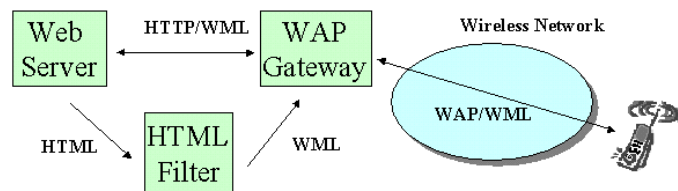


Fig. 1. WAP System Architecture

B. WAP Browsing, WML

Transmission of the URL from the mobile device and presentation of the final response download are both performed by a browser program hosted on the device. This browser software is commonly known as a *WAP browser*. A WAP browser is the mobile device analogy to a web browser on a standard computer. A device which hosts a WAP browser is said to be *WAP-enabled*. Such devices are the focus of the work described in this paper.

HTML is the language of the web. It is a rich and therefore large language and is continually evolving as the web hosts ever more presentation formats such as audio and video. Due to user interface restrictions, a mobile device cannot (yet) be used to surf the web in the same way as a desktop machine. Thus, the WAP forum has introduced a new language for WAP content formatting called the Wireless Markup Language (WML)[4]. WML is a tiny subset of HTML. The design of WML is influenced by what can be physically displayed on a WAP-enabled device, as described above. WML supports text in paragraphs, simple tables and images, and simple user input forms. WML is thus to WAP what HTML is to HTTP, the formatting language in relation to the protocol used to transport it.

The WAP system architecture and its use of existing web services means that the effort required to re-target the MyBus system to WAP-enabled devices is solely in the area of converting HTML responses into equivalent WML responses. This effort involves trimming the content to the absolute minimum required to preserve the original semantics. This task has been achieved manually for MyBus. An alternative approach is to automate this conversion using HTML to WML filter software. However, complete automation is impractical in all but the simplest cases. The disparities in capability between HTML and WML are such that most HTML web pages cannot be transformed to an equivalent WML page.

III. MYBUS

The stimulus for the mobile device based prediction system is an existing bus departure prediction application called MyBus[5]. The front end of MyBus is a publically accessible web site (<http://mybus.org>), which simply formats incoming prediction reports for display on standard computer web browsers. The actual prediction work is done upstream of the web site. Real-time transit vehicle location data flows through a set of processes (the MyBus backend). The final output of these processes is a continuous stream of bus departure predictions, available for query by the web site software. The MyBus system architecture is described more fully in Section V.

The first transit agency to be fully incorporated into MyBus is Metro King County, operating in the Seattle, Washington, area. Metro operates a large fleet, with more than 1000 buses in service at busy times. Furthermore, the number of geographic locations at which bus arrival/departure times are scheduled is also about 1000. These locations are distributed throughout the transit region, which extends over 1700 square miles. The scheduling locations are

termed *timepoints* in the US Transportation Communications for ITS Protocol (TCIP) standard[6]. Timepoints are generally associated with important road intersections on bus routes. Timepoints form the basis of location selection in the MyBus web site. Departure time predictions for individual bus stop locations are not yet available, since bus stop data rarely acts as direct input to schedule planning.

To view departure time predictions on the MyBus web site, the user selects the appropriate timepoint nearest to a desired bus stop. For the current Seattle-based site, a clickable map of the entire transit area is divided into several geographic regions, as shown in Figure 2.

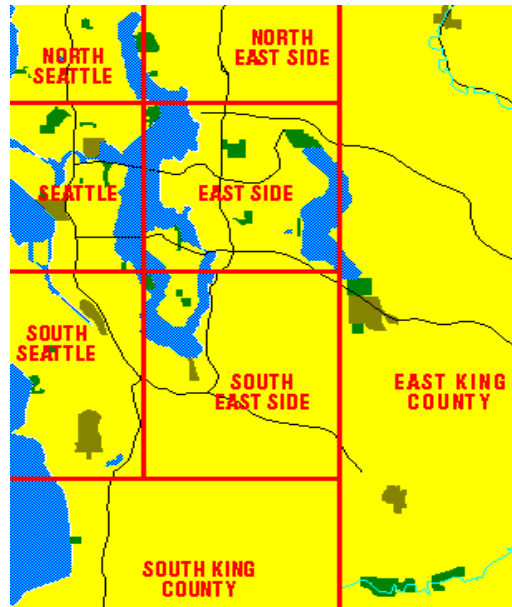


Fig. 2. MyBus Web Site Map of King County, Washington

Clicking on a region, e.g., *North Seattle*, displays a two-column list of hyperlinked timepoint names, a fragment of which is depicted in Figure 3. Link selection then delivers prediction data for that timepoint. Thus, to select a timepoint of interest, the user must interact with a combination of hyperlinked maps and lists. Using maps alone was deemed impractical, since 1000 distinct timepoint locations, some only a street block apart, cannot easily be graphically represented on a regular computer screen without resorting to some form of zooming facility.

The predictions page shows the predicted departure times of buses at the chosen timepoint location. Figure 4 shows such a predictions page for a busy intersection in downtown Seattle. Each row in the tabular display represents a bus which is scheduled to pass by the chosen timepoint within some specified time interval.

The page is designed so that rows are sorted by increasing route number and subsorted by increasing schedule time. This is the default, but the user can alter the display to be sorted by schedule time, with route number as the subsort. For each row, the leftmost three columns display data lifted from the schedule: bus route number, final destination, and scheduled time at which the timepoint

7 AV & 3 ST (KIRKLAND)	182 AV NE & NE 24 ST
70 AV SE & SE 24 ST	185 AV NE & REDMOND-FALL CITY RD
80 AV SE & N MERCER WY	BEAR CREEK P&R LAYOVER AREA
80 AV SE & SE 27 ST	BEAR CREEK PARK & RIDE
88 AV SE & SHOREWOOD DR	BELLEVUE CITY HALL
98 AV NE & NE 23 ST	BELLEVUE COMMUNITY COLLEGE
98 AV NE & NE 34 ST	BELLEVUE TRANSIT CENTER (ALL BAYS)
102 AV NE & NE 10 ST	BELLEVUE TRANSIT CENTER BAY #1
108 AV NE & NE 38 PL	BELLEVUE TRANSIT CENTER BAY #2

Fig. 3. Hyperlinked Timepoint Location List in MyBus



Route	Destination	Scheduled	Depart Status
2	Madrona	12:01pm	Departed At 12:05pm
2	Downtown Seattle	12:02pm	5 Min Delay
2E	Madrona	12:17pm	On Time
2	Downtown Seattle	12:23pm	No Info
36	Beacon Hill	12:19pm	2 Min Delay
36	Downtown Seattle	12:19pm	On Time
66E	Northgate P & R	12:19pm	1 Min Delay
70	University District	12:03pm	Departed at 12:04pm
358E	Aurora Village	12:19pm	On Time

Fig. 4. Standard Web Page Display of MyBus Prediction Data

should be reached. The real-time prediction appears in the rightmost column, as a deviation from the scheduled time. Examples of the possible prediction states are:

- *On Time* — within one minute of schedule time.
- *3 Min Delay* — the bus is 3 minutes late.
- *Departed at 12:57* — the bus has departed.
- *No Info* — prediction not available due to bad data.

The predictions display shows predictions for all buses (i.e., all routes in all directions) whose scheduled departure time from the chosen timepoint falls in a fixed time window. Currently, this window is configured to span the interval (current time - 20 minutes, current time + 30 minutes). The bi-directional property of the interval relative to the current time allows any late buses to be displayed. To minimise display clutter, departed buses are removed from the display ten minutes after actual departure.

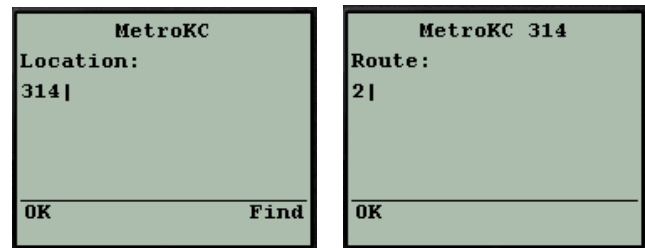
The MyBus web site has proved very popular since its introduction in July 2000. The site has received over two million requests for prediction data up to February 2001.

IV. MYBUS FOR WAP BROWSERS

The web site described in the previous section has been recently extended to deliver content suitable for WAP-enabled devices. WAP-enabled cell phones have been chosen as the preliminary target device. MyBus prediction data delivery to a class of larger devices, such as personal desktop assistants, is planned in the near future¹. This new system is known as the *MyBus WAP site*. The major problem to overcome is that of delivering the necessary bus departure information to a device with the physical restrictions discussed in Section II.

A. Predictions on a Phone

The user actions required to view MyBus predictions on a WAP-enabled cell phone are as follows. The browser must first navigate to the MyBus WAP home card², currently www.mybus.org/wml/, whence the user must enter a numeric value identifying the timepoint location for which departure predictions are of interest. This numeric identification of the timepoint, as opposed to named identification (Figure 3), is possible since the TCIP standards[6] state that timepoints have numeric codes as well as textual descriptions. Following location input, a numeric route number must be entered. Figure 5 shows the user interface, with 314 entered as the timepoint code and 2 for the bus route. The diagram sizes are equal to the actual screen size for a typical WAP-enabled cell phone.



MetroKC

Location:
314 |

OK **Find**

MetroKC 314

Route:
2 |

OK

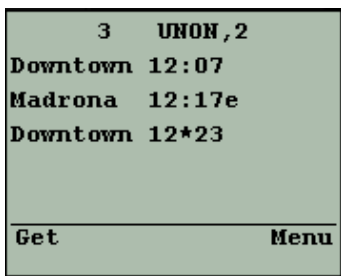
Fig. 5. Prediction Input Displays on a WAP Phone (actual size)

Accepting this numeric value pair for location and route, the browser sends out a URL request to the WAP gateway. The URL identifies the exact prediction data required. The URL is forwarded by the gateway to the MyBus web server, which responds with all current predictions relating to the location and route combination. The predictions display on a typical WAP-enabled phone is shown in Figure 6.

The display layout mimics that of the regular web page, with the location banner at the top, followed by a table of predictions, each row of which represents a departure prediction of a single bus. Each prediction shows the destination of the bus and the predicted departure time. Predictions for all buses on the chosen route whose scheduled departure time from the chosen location is up to 30 min-

¹WAP browser emulation software is available for such devices already, enabling them to connect to WAP sites such as MyBus.

²A WML card is analogous to a HTML page. A deck is a set of cards, and is the unit of WML transmission in WAP.



3 UNON,2	
Downtown	12:07
Madrona	12:17e
Downtown	12*23
Get	Menu

Fig. 6. WAP Phone Display of MyBus Prediction Data (actual size)

utes from the current time³ are retrieved, except those for buses predicted to have already departed.

The display in Figure 4 can be directly compared to that in Figure 6. The former shows predictions for all routes at the timepoint called **3 AV & UNION ST**. The latter shows predictions for route 2 buses at timepoint **3 UNON**. These are in fact the same timepoint; **3 UNON** is the eight character short name of the timepoint. The phone display relates to all route 2 buses, but not *all* buses, shown on the web page display. Since the phone does not display buses deemed to have already departed, the three rows on the phone display correspond to rows two through four on the web page. We shall refer to these rows as *A*, *B*, and *C* respectively.

The presentation of the four prediction fields can to be contrasted. These fields are route number, bus destination, scheduled departure time, and the real-time departure prediction. On the web page, predictions for buses on all routes are shown. The volume of information on the page, i.e., the table size, is not restrictive — the user can scroll the page to see predictions for any desired routes. On the phone, only one route is shown: in this case, the number 2. Thus the 2 appears in the title row on the phone only; in all subsequent prediction rows the route number is implied.

The destination string is presented differently on the two displays. On the web page, destinations such as **Madrona** (row *B*) and **Downtown Seattle** (rows *A* and *C*) are displayed. These strings are of arbitrary length, the longest being 32 characters in Metro’s case. Such strings are too long to display on a phone browser, so an eight character fixed length string is displayed instead. A web server function maps actual destination string to its shortened version. For example, **Downtown Seattle** becomes **Downtown**, and **Seattle Ferry Terminal** becomes **FerryTrm**.

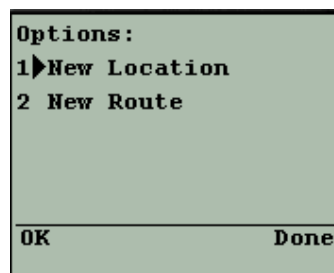
The scheduled and predicted departure times are given in separate columns on the web page. On the phone, with very few characters per row, the two values are summed and a single value presented. Thus, **12:02pm** and **5 Min Delay** (row *A*) on the web page become **12:07** on the phone, and similarly for rows *B* and *C*. When no real-time prediction is available, as in row *C*, an asterisk ***** replaces the colon: in the time string and the time shown is that from the schedule. The corresponding depart status on the web page

³Current time as deemed by the web server when the request arrives. No time value is extracted from the phone itself.

is **No Info**. Finally, an express bus (row *B*) is denoted on the phone using a trailing **e** character in the rightmost column. On the web page, this characteristic of the bus is part of the route number, e.g., **2E**.

Eight characters are required to identify the destination. One character of whitespace is included to delimit the destination string from the predicted departure time. The time string is five characters in length. A final column is required to identify an express bus. Thus the prediction display on the phone requires 15 characters for each row. Since the row length on some phones is smaller than this, some horizontal wrapping or line wrapping may ensue.

Once a prediction card is on display, various user actions are possible. A simple, one key gesture, that of pressing the **ACCEPT** key (normally labelled **OK** on phones), suffices to reload the ‘current’ predictions. This action is analogous to refreshing a web page with dynamic content. In Figure 6, the **ACCEPT** key is tied to the **GET** prompt. The second WAP/WML standard key, the **OPTIONS** key, leads to a menu offering options to select a new timepoint location, select a new route at the existing location, or to quit the site, as shown in Figure 7.



Options:	
1▶	New Location
2	New Route
OK	Done

Fig. 7. MyBus User Options after Prediction Download

B. Timepoint Identification

Using a numeric code to identify a geographic location suits cell phone user input extremely well, since digit keys are the general mode of input, i.e. for dialling telephone numbers. Alphabetic character input is more convoluted, requiring carefully timed repeat key sequences on the digit keys to generate character input. Location identification by name, e.g., **NE 45th St and 15th Ave NE**, would be problematic and error-prone on a WAP browser. Nevertheless, riders identify their bus stop by street and/or intersection name.

The problem with timepoint identification by number is that the user must know the numeric code for each timepoint of interest. The number of timepoints may be large — the Metro King County transit region has over 1000. Initially, the standard MyBus web site hosted a number of pages showing the mapping of all timepoint names to codes, e.g., **3 AV & UNION ST --- 314**. Consequently the user would have to note all timepoints of interest and have that information available when using the WAP based system. This made the MyBus WAP site somewhat awkward to use. To solve this problem, a self-contained help facility has been incorporated. Given a timepoint name, the

corresponding numeric code can be located on the phone itself⁴.

The timepoint help facility for the Metro transit operation is organised as a tree. Below the root are seven regions, corresponding to the regions selectable on the standard MyBus web site (Figure 2). Successively deeper tree nodes are organised by sorting of timepoint names. The name to numeric value mappings lie at the leaf nodes. A fragment of the data structure is shown in Figure 8. In the figure, only leaf nodes have been identified (timepoints 9013, 3015, 7308, 7239). In reality the tree has 1000 leaves. The tree root, labeled FIND, corresponds to the FIND action key in Figure 5.

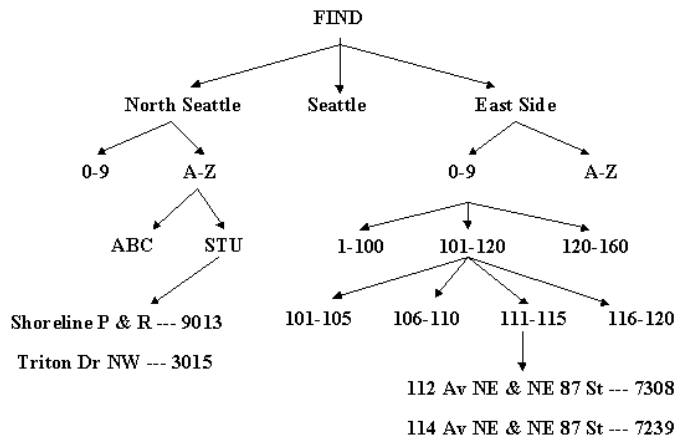


Fig. 8. Timepoint Name and Numeric Codes Organisation

The tree nodes are presented on the WAP browser as selectable menus. Figure 9 shows the sequence of menus visited to obtain the numeric identifier for the timepoint Shoreline Park & Ride.

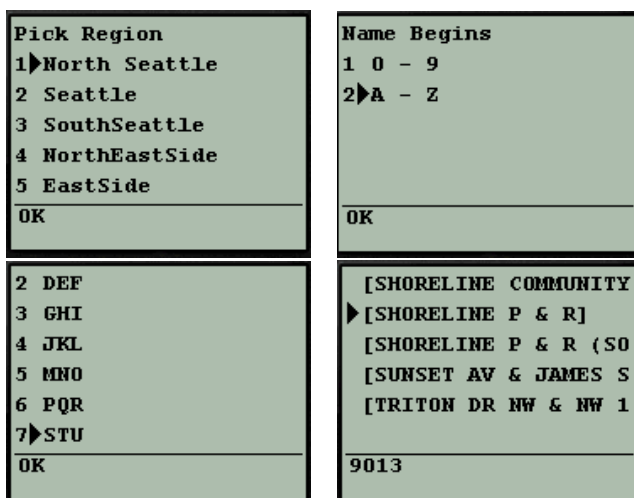


Fig. 9. Timepoint Number Lookup via Menu Navigation

The tree structured help system offers a compromise be-

⁴The user must still know the *name* of the timepoint of interest. The current geographic location of the phone, available via perhaps GPS, is not (yet) considered.

tween complexity of user interaction and download of pertinent information. For a large transit operation, a simple exhaustive menu of timepoint name to value pairs would be impractical. Furthermore, such a structure could be financially expensive. The current pricing structure of WAP services is based upon a per-minute or even per-byte cost. A design goal is thus to minimise the download of irrelevant information. As WAP technology matures, different user interface designs may apply.

V. SYSTEM ARCHITECTURE

MyBus is organised as a set of communicating processes. Real-time bus-oriented data flows between the processes. Figure 10 shows the system architecture. The overall functionality is to transform a raw transit agency AVL feed into departure-time predictions for display on the MyBus web and WAP sites. The protocols used at each stage of the computation are noted. Raw AVL data is received in *legacy* format, i.e., its format is total due to the transit agency's software configuration. This legacy data is then transformed into the Self-Describing Data (SDD) format[7]. SDD is the data transport protocol used in all ITS/UW projects⁵. Each bus report packet received from the legacy stream is converted to SDD. The tracker component makes use of Kalman filtering algorithms[8] to continuously estimate bus locations when physical location reports are infrequent, e.g., in the order of one to five minutes. Most existing automatic vehicle location (AVL) systems fall into this category. The tracker's data output rate is thus higher than its rate of input.

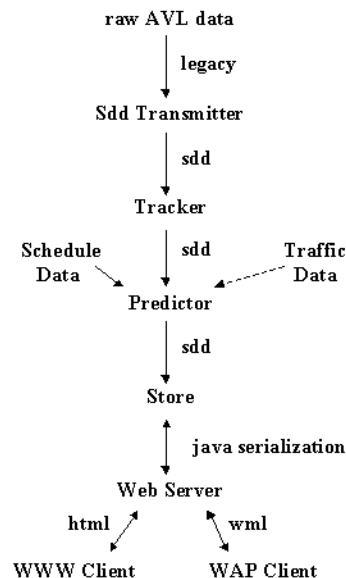


Fig. 10. MyBus System Architecture

The predictor reads the tracker output and transit agency schedule data to determine departure times from all timepoints that each bus is scheduled to visit in some future

⁵The real-time stream of Metro's AVL data is freely available in SDD format from address `sdd.its.washington.edu:8412`.

time window. Predictions are then fed to and held in a store process. The store acts as a database, queryable by the web server for prediction data relevant for a particular time-point (web client) and route (WAP client). Such queries are initiated by browsers, either web or WAP. Note that data transfer up until the store process is unidirectional. SDD supports a unidirectional communication model only — data are streamed from transmitter to receiver.

Both the store process and web server process are written in Java[9], with the consequence that Java object serialization is the natural model of communication between the two. Finally, the web server formats the predictions into HTML or WML according to the client characteristics.

VI. USAGE

The MyBus WAP site for Metro King County bus departure predictions has received over 9000 hits since its initial release in September 2000. Figure 11 shows the steady increase in usage each month. Unlike the MyBus web site, the WAP site server does not send deck refresh information to the cell phone, for financial reasons as much as any. All hits counted here are actual user-instigated actions. Of the hit total, 3500 are actual prediction deck downloads, and 2000 are help system requests (the remainder are welcome decks).

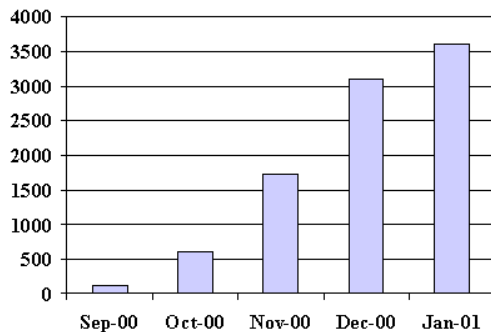


Fig. 11. WAP Browser Hits to the MyBus Web Server By Month

VII. CONCLUSIONS, FUTURE RESEARCH

We have shown that a WAP-enabled cell phone is a suitable device for receiving real-time transit information. Its physical restrictions can be overcome through considered user interface design. The nature of the information delivered by the MyBus prediction system is better suited to mobile users such as bus riders.

Future work on this project includes development of predictions formatted for personal digital assistant devices. The capabilities of such devices falls somewhere between a regular computer and a cell phone. To further improve prediction accuracy, data from other sources are to be incorporated into the predictor algorithms. Example sources are real-time traffic condition data (see Figure 10) and historical prediction performance.

REFERENCES

- [1] “Wireless Data Users to Reach 1.3 Billion by 2004,” http://www.instat.com/pr/2000/md0003md_pr.htm, 2000.
- [2] “Wireless Application Protocol Forum,” <http://www.wapforum.com>, 2001.
- [3] “Apache HTTP Server Project,” <http://httpd.apache.org>.
- [4] WAP Forum, “Wireless Application Protocol Wireless Markup Language Specification, Version 1.3,” <http://www.wapforum.com/what/technical.htm>, Feb. 2000, WAP-191.
- [5] D.J. Dailey, S.D. Maclean, F.W. Cathey, and Z. Wall, “Transit Vehicle Arrival Prediction: An Algorithm and a Large Scale Implementation,” in *Proceedings of the Transportation Research Board Annual Meeting*, Jan. 2001.
- [6] S.D. Maclean and D.J. Dailey, “MyBus: An Advanced Public Transportation System Based on the US TCIP Standard,” in *Proceedings of the Seventh Annual World Congress on Intelligent Transport Systems*, Turin, Italy, Nov. 2000.
- [7] D.J. Dailey, D. Meyers, and N. Friedland, “A Self Describing Data Transfer Methodology of ITS Applications,” *Transportation Research Record*, vol. 1660, pp. 140–147, 1999.
- [8] B.D.O. Anderson and J.B. Moore, *Optimal Filtering*, Prentice Hall, 1979.
- [9] Ken Arnold and James Gosling, *The Java Programming Language*, Addison-Wesley, 1996.