Smart Parking Lot with Just in Time Shuttle (SPLITS)

Harish Venkatappa
Department of Computer Science
Oregon State University
Corvallis, OR 97331

A research project submitted in partial fulfillment of the degree of Master of Science

Major Professor: Dr. Timothy A. Budd
Minor Professor: Dr. Bose Bella
Committee: Dr. James A. Van Vechten

Contents

Abstract

The purpose of this project is to study the benefit at large parking lots having shuttle busses by providing a system that dynamically routes a shuttle. The shuttle route will be set by the placement of the travelers returning from the terminal. A parking space is assigned to the incoming traveler based on the shuttle's current route and where the shuttle's current passengers are going to be dropped off. The SPLITS simulator is a program designed to simulate the traffic of cars and shuttles for PDX's Economy Parking Lot. In this project we have shown how the SPLITS system would better the existing parking system by simulating and comparing both the processes.

Chapter 1

<u>Introduction:</u>

1.1 Problem:

In today's fast pace world, where time is money, people are constantly on the move. People are looking for technologies which save their time and money and businesses all over the world are trying to find the most efficient way to use the resources they have. Today, long term parking at airports with huge parking lots can be time consuming and sometimes frustrating for travelers. This is because the travelers have to search for their spot and then haul their luggage to the bus station and catch a shuttle. There are difficulties in searching for the space, especially when the lot is nearly full, and it may also result in congestion inside the lot when many cars are searching for free spaces to park. All these may cause anxiety and stress on travelers about being in time for their plane. A need for an organized and efficient parking system is necessary. This project aims at solving the problems of parking existing now and thus providing an efficient parking system.

1.2 Overview of the Existing Parking System:

The parking lot under consideration for this project is the Portland International Airport (PDX) Economy Parking Lot. At present, travelers who are interested in parking collect a ticket at the entrance of the lot. The ticket records the time and day travelers entered the lot. After entering the lot, travelers have to search for a free space to park. After successfully finding the space and parking they go to the nearest bus station in the lot and wait for the shuttle to pick them up. The shuttle picks them up and drops them at the terminal. Those travelers who want to get back to the lot from the terminal board a shuttle-bus which takes them to the lot. The travelers get off the shuttle at the station nearest to where they have parked. While exiting the lot, the travelers present to one of the parking lot attendants the ticket that they received at the time of entering the lot. The attendant calculates how much the travelers are required to pay, and the travelers make the payment and drive out.

1.3 Disadvantages:

We made a few trips to the PDX airport and observed the movement of traffic in the parking lot. We also talked with the manager of the lot. From our observation and talk we summarized some of the problems experienced by travelers:

- When the lot is almost full, travelers waste a lot of time searching for a free space.
- Travelers are frustrated when they are unable to find a free space.
- Congestion may result in the lot when many vehicles are driving around searching for a space.
- Travelers have to drag their luggage all the way to one of the stations.
- Travelers have to remember where they parked.

Chapter 2

2.1 Smart System

Considering the difficulties the travelers are experiencing now while parking, a need for an organized parking is necessary. The smart parking system proposed is known as **Smart Parking Lot with Just in Time Shuttle (SPLITS).** This is a system that directs travelers to free spaces. The system also directs the shuttle to pick the new in-coming travelers. This system is designed to decrease the amount of time that a traveler must take to find a spot as well as how long they have to wait for a shuttle to pick them up. The system chooses a space in the parking lot for the new traveler based on where the closest available shuttle is located and what spaces are open.

2.2 Physical Concept of SPLITS:

The physical concept is that each in-coming driver will either swipe a credit-card to start the transaction for a parking fee or will take a magnetically encoded smart-card from the Entrance computer (EC). The Control Computer (CC) will then inform, via

EC, each in-coming driver that it has assigned her or him to a specific space and that the bus will meet her or him there. The assigned space is determined depending on where the shuttles are in the lot and the location of empty spaces. The assignment of space will be explained in more detail in the next chapter. EC will also give each in-coming driver a map of the lot, which will display clearly the entrance point ("You are now Here.") and the location of the assigned spot ("Park here; this is where the shuttle will meet you.") and perhaps other information about the smart lot. The driver now has to follow the instructions and reach the spot. Once he has reached the spot the shuttle will arrive and pick him up.

Before they board the bus at the terminal out-going travelers will swipe at Ticket Counter (TC) either the same credit card or the smart card that they received from the EC at the entrance. The CC will access where each traveler's vehicle is located and prepare the best possible route for the bus.

It is assumed that each shuttle-bus has a global positioning system, GPS, which is wirelessly connected to the CC so that the CC always knows where each bus is. This helps the CC to route the shuttle to pick up the new traveler.

2.3 Goals

Let ITT-In be the time between a traveler entering the lot and arriving on the shuttle-bus at the terminal. X-ITT-In is the worst-case in-coming time of a traveler and ITT-Out is the time between traveler boarding a shuttle-bus at terminal and exiting the lot.

The goals of the Smart System are as follows:

i. The Primary goal is to Reduce ITT-In. This includes the time spent by the traveler searching for a space, waiting for the bus, and the time spent on the bus to get to the terminal.

ii. The secondary goal is to establish and to reduce the unluckiest traveler's incoming time X-ITT-In, so that the lot's management can assure pending travelers, perhaps via a variable message board outside the entrance, of the worst case time for them to get to the terminal if they choose to park in this smart lot

iii. The tertiary goal is to reduce outgoing time, ITT-Out. ITT-In is more important than ITT-Out because travelers are often very concerned that they arrive at the terminal in time to board their flight, but they are more relaxed about the time that they return to their vehicle.

2.4 Benefits of the Smart System

The smart system proposed provides a benefit in convenience and in safety because the bus picks-up and leaves-off travelers at their parking spaces so that they do not carry their luggage between their space and a bus-stop and, when in-coming, do not wait for a bus at a station. Some of the benefits provided by the Smart System are:

- Reduction in time spent searching for available parking space. At the same time, lower fuel consumption.
- Reduction in congestion due to fewer cars driving around searching.
- Elimination of queues entering the parking lot, because drivers will not go to a facility where there is no available space.
- Reduction in illegally parked vehicles.
- Better distribution of flow and parking demand through the area.

Chapter 3

3.1 Description of the Parking Lot under Consideration:

The Lot considered for this project is the Portland Airport's Economy parking lot. The lot has 6999 spaces. It is located away from the airport terminal and it takes the shuttle bus about 6 minutes to get to the terminal. There are four shuttle busses, which run between the lot and the terminal. The lot has 18 bus-stations where travelers' can board the shuttle bus to get from the lot to the terminal. The bus also drops travelers' coming from terminal to lot at these stations. The lot has 161 pods (pod is a set of spaces separated by an aisle). There are 18 bus-stops within the lot, which are connected by the "main road" which is the only route that the busses now use.

3.2 Implementation

We have simulated both the existing parking system as well as the smart parking system. We have classified the existing system into 3 different categories after observing the flow of traffic in the PDX lot and talking with the manager of the parking lot.

Existing #1 – In this method the traveler searches for a free space from one pod of spaces to the next starting at the entrance and continuing until the traveler finds the first empty space. According to our observation at PDX lot, a large majority of travelers use this method.

Existing #2 – In this system, entering travelers find a shuttle bus and follow it until a traveler gets off and then follows the traveler to her/his space. This can be a relatively efficient tactic when the out-going person is a business traveler with little luggage and the weather is good. However, one of the travelers that we interviewed on the bus during a snow storm in the Christmas rush said that she had spent 30 minutes waiting for a family with a considerable amount of luggage to get from the bus-stop to their car, load up, remove the snow and ice, and, finally, to leave.

Existing #3 – In this system the travelers drive immediately from the entrance toward the far end of the lot where there are almost always many spaces available. Because the bus goes to all the stops in a fixed route, a passenger who boards at the far end of the lot will reach the terminal at the same time as one who boards the same bus at the first station. One might think that travelers would recognize the advantage of going to the far end and parking. However, as the bus-drivers will attest, only a small fraction of repeat travelers learn this lesson.

We have classified the smart system into 2 categories.
Splits #1 – This system assigns spaces independently from one to the next without consideration of opportunities to bunch in-coming travelers to allow more than one to board the bus at one stop. To bunch the assignments we either will need to have vehicle counting sensors at the entrance to inform the CC how many vehicles will enter in the next minute or use other means (time of day functions or direct observation) to determine entrance rates.

Splits #2 - This system would access the travelers' itineraries so that the CC can optimize for the return route, as well as for the out-going route.

For the purpose of simulation we have considered Existing #1 and Smart #1 systems. The Economy lot at PDX has 6999 spaces in 161 "pods", i.e., 13+/- 6 pairs of perpendicular spaces separated by an "aisle" and some string of spaces, mostly around the perimeter of the Lot. For the simulation of Existing #1 system we inputted observed values for time that a bus takes to go from one station to the next. There are also "feeders", roads that connect the pods to main road and thus to the bus stations. For the data in this report we set the following parameters in the program: acceleration and deceleration for travelers' vehicles and busses is 20% of the earth's gravitational acceleration ($0.2 \times g = 6.4$ ft/sec^2); all vehicles' maximum speed on aisles is 15 mph; all vehicles' maximum speed on feeders is 20 mph; all vehicles' maximum speed on main roads is 30 mph; all vehicles' time to turn 90 degrees is considered as the vehicle coming to a stop and then accelerating after the turn; and the bus travels for 6 minutes each way to go between the Lot and the Terminal. Of course, it would be a simple matter to change these parameters and to run the program again. We would be able to adapt the program to changing conditions (e.g., due to changes in weather) in any given lot and to adapt the program to new lots.

Mr. Cushman, the manager of the Lot, gave us a detailed map of the parking spaces in it. With this map and the above parameters, we generated two look-up tables. The larger table is 161x161 and gives the time it would take the bus to go from the entrance of each pod to the entrance of every other pod in the Lot. Since the lot is not a perfect rectangle and the spaces are haphazardly distributed, we could not use any algorithm to determine the time required in going from one space to any other space in the Lot. So we split the lot into 9 areas, each area is a collection of pods (perpendicular spaces separated by an aisle). The time required to get from each pod to any other pod in the same area was determined using a simple algorithm that takes as input the pod width, length and the speed of the vehicle. Once the time required between each pod in an area was computed, the time required to get from one area to another was established thus giving the time required to get from one area to any other area in the Lot. To run the simulation we add to these values a simple estimate of the time to drive from the entrance of the pod to a particular space in the pod. This is the same for the bus and for the travelers and for both the smart and the existing systems. The user can change the values in the appropriate look-up table to correct any misunderstanding regarding particular spaces.

The smaller look-up table is 1x161 and gives the time a traveler would use to drive from the entrance to each of the pods. We use this information to ensure that the bus will not need to wait long for the travelers to arrive at their assigned spaces. (By timing ourselves doing it we determined that a well-informed traveler can drive from the entrance of the Lot to any space within 2.5 minutes.)

Splits #1 directs shuttle-busses and assigns spaces to in-coming travelers with the following rules:

a) Travelers boarding the bus at the terminal must be left at their spaces in the lot. (We cannot take them back to the terminal.)

b) We cannot make any traveler wait on the bus at the terminal more than Y minutes before leaving for the lot. (The lot's management will set the value of Y.) We have assumed Y = 2 minutes.

c) Whenever a bus at the terminal is too full to take-on more travelers, or if the maximum allowed time at terminal, which is 2 minutes is over, it starts-out for the lot.

d) Whenever a bus in the lot has left-off all the travelers from the terminal and is too full to take-on more travelers, or when the maximum allowed time in lot is over, it starts-out for the terminal. (The lot's management will set the value of the maximum time in lot, we have assumed it to be 10 minutes.)

e) We cannot make any in-coming traveler wait on the bus in the lot more than X minutes before starting for the terminal. (The lot's management will set the value of X.) We have assumed X = 10 minutes.

Rule a) implies that the bus must go to a list of known spaces in the lot. While the bus is at the terminal, this list will be up-dated whenever new travelers board the bus. Once the bus leaves the terminal, this list is frozen. This information is used to determine one of the fastest routes, Route 1, to visit these spaces and the expected time, Time 1, to complete it. Because it takes 6 minutes for the bus to travel from the terminal to the lot and because the lot authority pledges to pickup travelers in 6 minutes or less, there is no time for a bus just leaving the Terminal to reach a customer in the Lot at that time. Thus, Route 1 does not include stops for new travelers. For the results reported here, we have used a simple heuristic algorithm that the bus goes from the entrance of the lot to the closest space that it needs to visit, from there it goes to the next space that is closest to its present location, and so on.

The number of busses serving the Lot at PDX varies with the traffic level but it is typically 4, which is what we have assumed. From the time a bus loads travelers at the terminal until it leaves the lot to go to the terminal every bus has a route and can change. When a traveler enters the lot, the program will first determine if one of the busses already in the lot could pick up the traveler at an available space near its existing route and within the rules and allowing for the time that the traveler would spend driving to that space. If so, then program assigns the traveler that space and the route of the appropriate bus is altered.

If no bus already in the lot can pick up the just entering traveler within the rules, then the program considers the present routes, R1s, of busses at the terminal or coming from the terminal. It identifies and assigns the space that will increase the time for that route, T1, by the least amount. Any R1's that changes becomes R2a's with associated T2a's. If R2a changes again before the corresponding bus enters the Lot, then the new route becomes R2b with associated T2b. There are corresponding designations for further modifications before the bus enters the Lot.

When a given bus enters the Lot, its current route R2x is re-designated to be R3 with associated time T3. Modifications of the route to take on travelers who enter the lot after the bus has entered the lot, as described in the next to last paragraph above, are designated to be R3a,b,c... with associated T3a,b,c....

3.3 Simulations

The simulation was coded in C on windows-98. The main data structures were the Customer and Shuttle. Travelers are represented as customers in the program.

<u>Shuttle structure</u>

```
typedef struct {
   int id;               // Shuttle number
   int position;         // 1 - Shuttle going to Terminal from the Lot
                         // 2 - Shuttle is at the terminal
                         // 3 - Shuttle is coming to Lot from Terminal
                         // 4 - Shuttle is in the Lot
   int areaNo;           // holds which pod number the shuttle is at
   int noOfPass;         // holds the number of passengers on board
   int passToLot[maxFill]; // array which holds the passenger ids who are traveling to Lot from
                         terminal
```

```
    int passToTerm[maxFill];          // array which hold the passenger ids who are traveling to terminal
                                      from Lot
    int custToPickup[maxFill];        // array which holds the passengers the shuttle has to pickup in the
                                      lot
    int timeElapsed;                  // holds the time elapsed at the present position
    int route[maxFill];        // array which holds the route of the bus, the route is a series of pod
                               numbers
} Shuttle;
```

Customer Structure

```
typedef struct {
        int custId;          // Holds Customer id
        int AreaNo;          // Holds which pod # the customer has parked his car
        int timeStamp;       // records the time passenger entered the lot
        int picked;          // if 1 then the customer has been picked by a shuttle.
        int slotNo;          // holds which slot number
        int timeToSlot;      // records the time to get the slot from the entrance of the lot
        int timeToTerm;      // records the total time taken to reach the terminal after entering the lot
        int isAtTerm;        // if it is 1 then customer has reached terminal otherwise no.
        int timeStamp1;      // records the time when customer boards the shuttle at terminal

} Customer;
```

Simulation of Smart System:
The modules in the simulator are:

1. splits
2. init
3. sim
4. gate
5. report

1. splits routine

The splits routine is the main program for the simulator. All of the command line arguments are read from splits.c. It calls the init routine which performs initialization. The init routine is described in more detail in the next section. There is a loop which decrements for each minute in the number of days requested to run. On each iteration the sim routine is called which simulates one minute of the simulation. Once the number of days has expired, a report is produced by report routine.

2. init routine

The init routine is called once to initialize the parking areas of the lot, shuttles, and customers. First, the parking area structure, i.e., the pods in the lot is initialized. Then each pod is filled with customers up to percentage of lot to be filled, which the user has entered. The customers are assigned customer identification numbers and are defined to be at the terminal. The routine then initializes the shuttles by having them start at the terminal. The CreateTable function creates the lookup table that gives the time required for a vehicle to get from one pod to any other pod in the lot. Once each shuttle has been deployed at an equal interval from the previous shuttle the init routine is considered finish and the simulation is then started. This is done by calling the sim routine.

3. sim routine

The sim routine is a loop, which represents a single minute to the simulator. During this minute new customers enter at the gate thus calling the gate routine. The gate routine is described in detail later. Then the shuttles are moved, the location of each shuttle will determine which function is called. These locations are determined by shuttle[i].position and include: traveling to the terminal (to_term), at the terminal (at_term), traveling to the lot (to_lot) or in the lot (in_lot). Depending on these locations either the lot function or term function may be called. When traveling to and from the terminal shuttle[i].timeElapsed is used to increment for each minute and then tested to see if the value is such that it has reached it's destination (either the terminal or the lot). For example if shuttle[i].position = to_lot then shuttle[i].timeElapsed is checked to see if it has now reached the lot otherwise timeElapsed is incremented and the next shuttle is checked.

If on incrementing the position, a shuttle reaches the terminal then all the customers are dropped and their status is set to 'at_term' and the time taken by the customers to reach the terminal is computed. Now customers who are at the terminal and need to be taken to the lot are loaded. The cust[cust_num] is found using a rand(), which randomly selects customers who are at terminal. Once customers are loaded, the route of the shuttle-bus is set as described before.

If on incrementing the position, a shuttle reaches one of the drop-off or pickup pod in its route then the shuttle drops off passengers, if any, and picks the passengers waiting to board the shuttle. Metrics such as the time the customer was waiting for the shuttle bus or the time the customer spent to get to the lot from the terminal are calculated.

4. gate routine

The gate routine is used to enter new customers into the lot. Each time the gate routine is called one new customer enters the lot and is assigned a parking space. When a customer enters the lot, the program will first determine if one of the busses already in the lot could pick up the customer at an available space near its existing route and allowing for the time that the customer would spend driving to that space. If so, then program assigns the customer that space and the route of the appropriate bus is altered and the customer is added to the pickup list of the shuttle.

5. report routine

Once the number of days has been simulated a report is generated when splits calls report.c. This report gives how many customers were transported to the terminal. For these customers, it is then reported how long it took them to be picked up and how long to get to the terminal once they were picked up. The number of customers transported from the terminal to the lot is also reported. The average time taken by the customers to get to their parking spaces is calculated and reported.

Simulation of Existing System:

The modules are:

splits

init

sim

gate

report

splits, init and report modules remain the same as in the smart system. The modules sim and gate are briefly explained below:

sim

The sim module almost remains the same as in the smart system except that the shuttle has a fixed route through the lot visiting all the stations. The shuttles at PDX now go to all 18 stations in the lot. We noted the time required for the shuttle to go from one station to other and these values were used to increment the position of the shuttle. If on incrementing the position of the shuttle, it reaches any of the station, then the shuttle drops off customers, if any and also picks up customers' waiting to board the bus at that station.

gate

The gate routine is used to enter new customers into the lot. Each time the gate routine is called one new customer enters the lot. When a customer enters the lot, instead of the program assigning him a free space, the customer now searches for a free space. We have simulated this by making the customer search for a free space beginning at the entrance until he finds the first free space. The time required is calculated and it depends on how much distance the customer drove his vehicle. After parking he goes to a station near to him. Each station has a list, which has the customer ids' of travelers waiting to board the shuttle. Once the travelers board the shuttle the list is cleared.

Physical Demonstration

To demonstrate the physical concept of SPLITS we connected 5 computers in which one is the central computer (CC) which acts as the server and the other four are connected as clients which serve as the Entrance Computer (EC), Terminal Computer, the Shuttle Computer, and the Exit Computer. A key pressed in the EC is regarded as a traveler entering the lot, the CC computes the best possible space as described before and informs the customer through EC regarding his space. The CC also updates the shuttle computer with the new route for the shuttle. If any of the shuttles reaches the terminal, customers' board the shuttle, the list of customers boarding the shuttle is displayed in the terminal computer. List of customers exiting the lot is displayed on the exit computer.

Chapter 4

4.1 Results

For the Existing #1 and Smart #1 systems we started with a distribution of spaces evenly divided among all the pods and let the simulation run with a turn-over rate of 2160 vehicles per day, i.e. 1.5 vehicles per minute on average but fluctuating randomly, for both in-coming and out-going vehicles. This ensures that the fill-factor remains essentially constant, where fill-factor is the percent of filled spaces in the lot. After 4 days of simulated time we found that the distributions were near a steady state that was different for the smart system than for the existing system. The smart system tends to a distribution that appears to be random. The existing system packs the filled space near the entrance and packs the empty space at the far end. We then ran the simulation for 5 simulated days with constant fill-factors of 50 % to 99.9 % to take the data for our statistical analysis. In all the Graphs shown, the graph on the left corresponds to simulation results for the first 4 days and the graph on the right shows the results for the next 5 days.

The output of the simulation gives the average time the travelers took to park their vehicle, average time spent in waiting for the bus to arrive, average time spent in the bus before arriving at the terminal and the average time required to get back to the lot from terminal.

A sample output for Existing system with 50% and 99% fill factor:

Fill Factor: 50%
# of simulated days = 4
Average time taken to park is 7.760990 minutes
Average time waiting for bus is 2.747690 minutes
Average time on bus is 13.070779 minutes
Average time to terminal is 23.579460 minutes
Average time to drop is 10.636222 minutes

Fill Factor: 99%
# of simulated days = 4
Average time taken to park is 17.800209 minutes
Average time waiting for bus is 2.610568 minutes
Average time on bus is 10.421562 minutes
Average time to terminal is 30.832338 minutes
Average time to drop is 12.162425 minutes

A sample output for Smart system with 50% and 99% fill factor:

Fill Factor: 50%
# of simulated days = 4
Average time to Park is 1.859914 minutes
Average time waiting for bus is 1.926811 minutes
Average time on Bus is 9.809001 minutes
Average time to Drop customers at Terminal is 13.595727 minutes
Average time to Drop customers at Lot is 10.637640 minutes

Fill Factor: 99%
# of simulated days = 4
Average time to Park is 1.772701 minutes
Average time waiting for bus is 2.202501 minutes
Average time on Bus is 9.512107 minutes
Average time to Drop customers at Terminal is 13.487310 minutes
Average time to Drop customers at Lot is 10.624807 minutes

The output of the smart system also shows the route of the bus and the pod to which the traveler was allotted the space. A sample output while the simulation is running is shown below. clk represents each minute of simulation. The route numbers are the pod numbers that the shuttle has to visit. The 'route before alloting' is the route of the shuttle before the customer is assigned a space and the 'route after alloting' is updated route of the shuttle.

```
clk = 291
Route of shuttle #2 before alloting 64 --> 127 --> 150 --> 41 --> 105 -->
Customer is allocated pod #41 and Slot No. 1 and BUS is 2
route after alloting 64 --> 127 --> 150 --> 41 --> 105 -->
```

```
clk = 292
Route of shuttle #2 before alloting 127 --> 150 --> 41 --> 105 -->
Customer is allocated Area 41 and Slot No. 3 and BUS is 2
route after alloting 127 --> 150 --> 41 --> 105 -->

clk = 293
Route of shuttle #1 before alloting 0 --> 26 --> 49 --> 3 --> 132 --> 154 --> 109 -->
Customer is allocated Area 49 and Slot No. 1 and BUS is 1
route after alloting 0 --> 26 --> 49 --> 3 --> 132 --> 154 --> 109 -->

clk = 294
Route of shuttle #1 before alloting 26 --> 49 --> 3 --> 132 --> 154 --> 109 -->
Customer is allocated Area 18 and Slot No. 26 and BUS is 1
route after alloting 26 --> 49 --> 3 --> 132 --> 18 --> 154 --> 109 -->

clk = 295
Route of shuttle #1 before alloting 49 --> 3 --> 132 --> 18 --> 154 --> 109 -->
Customer is allocated Area 26 and Slot No. 24 and BUS is 1
route after alloting 49 --> 3 --> 132 --> 18 --> 26 --> 154 --> 109 -->

clk = 296
Route of shuttle #1 before alloting 3 --> 132 --> 18 --> 26 --> 154 --> 109 -->
Customer is allocated Area 28 and Slot No. 27 and BUS is 1
route after alloting 3 --> 132 --> 18 --> 26 --> 28 --> 154 --> 109 -->

clk = 297
Route of shuttle #1 before alloting 132 --> 18 --> 26 --> 28 --> 154 --> 109 -->
Customer is allocated Area 3 and Slot No. 35 and BUS is 1
route after alloting 132 --> 18 --> 26 --> 28 --> 3 --> 154 --> 109 -->

clk = 298
Route of shuttle #1 before alloting 18 --> 26 --> 28 --> 3 --> 154 --> 109 -->
Customer is allocated Area 28 and Slot No. 31 and BUS is 1
route after alloting 18 --> 26 --> 28 --> 3 --> 28 --> 154 --> 109 -->
```

The following table is the statistical data obtained by running the simulation.

Average time a traveler takes to park his vehicle

	Existing System (time in minutes)		Smart System (time in minutes)	
lot full (%)	1-4 days	5-9 days	1-4 days	5-9 days
50	7.76	10.5	1.85	1.85
60	9.07	12.11	1.82	1.82
70	10.73	14.23	1.79	1.78
80	12.7	15.92	1.77	1.76
90	14.92	17.46	1.77	1.76
95	16.78	18.06	1.77	1.77
99	17.8	18.16	1.77	1.77
99.9	18.81	18.85	1.77	1.77

Average time a traveler spends waiting for the shuttle to pick him up

lot full (%)	1-4 days	5-9 days	1-4 days	5-9 days
50	2.74	2.58	1.72	1.92
60	2.66	2.71	1.75	2.06
70	2.73	2.68	1.8	2.15
80	2.68	2.65	1.9	2.14
90	2.65	2.68	1.94	2.17
95	2.66	2.6	2.02	2.19
99	2.61	2.48	2.05	2.2
99.9	2.26	2.21	2.05	2.2

Average time a traveler spends on shuttle-bus to get to terminal

lot full (%)	1-4 days	5-9 days	1-4 days	5-9 days
50	13.07	12.44	10.03	9.8
60	12.75	12.07	10	9.66
70	12.36	11.57	9.94	9.57
80	11.9	11.18	9.82	9.57
90	11.38	10.81	9.77	9.55
95	10.94	10.26	9.69	9.51
99	10.42	10.26	9.67	9.51
99.9	9.92	9.79	9.67	9.51

Average time a traveler takes to get to terminal after entering the lot

lot full (%)	1-4 days	5-9 days	1-4 days	5-9 days
50	23.57	25.53	13.61	13.59
60	24.5	26.9	13.58	13.55
70	25.83	28.49	13.54	13.51
80	27.29	29.75	13.49	13.49
90	28.96	30.96	13.5	13.49
95	30.4	30.93	13.5	13.48
99	30.83	30.91	13.5	13.48
99.9	30.84	30.9	13.5	13.48

Average time a traveler takes to get to the lot from terminal

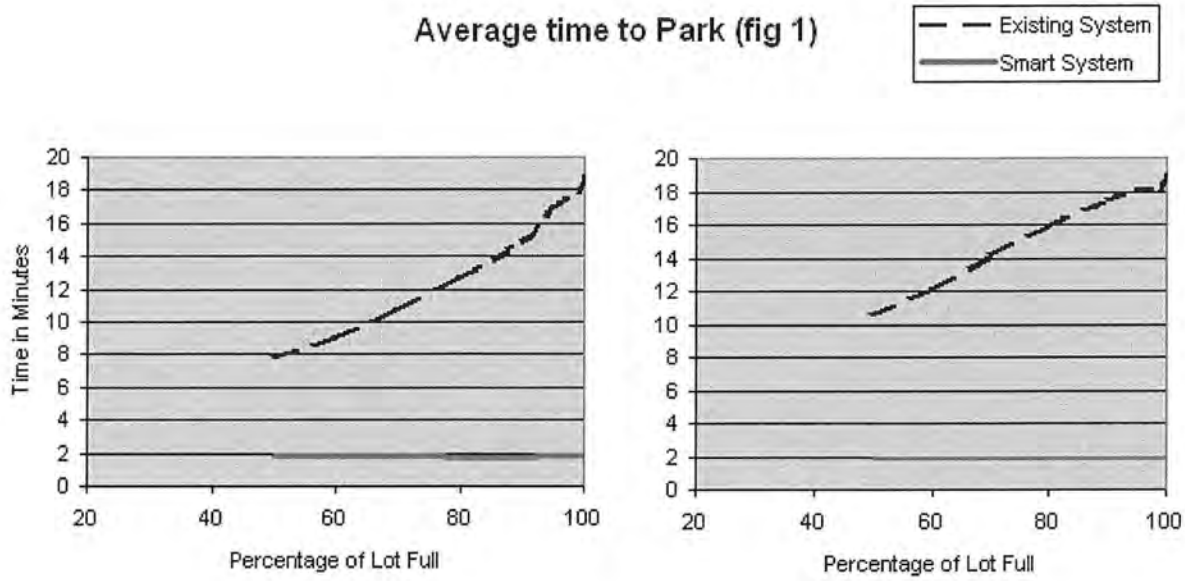
lot full (%)	1-4 days	5-9 days	1-4 days	5-9 days
50	10.63	10.54	10.61	10.63
60	10.9	10.91	10.62	10.63
70	11.17	11.34	10.61	10.62
80	11.44	11.7	10.62	10.63
90	11.74	12.04	10.61	10.63
95	12.01	12.17	10.62	10.63
99	12.16	12.16	10.62	10.62
99.9	12.16	12.17	10.62	10.62

Graphs

In Fig. 1 we show the average time for a traveler to drive to a parking space as a function of the fill-factor, i.e., the percent of filled spaces in the lot, for Smart #1 and Existing #1. The values for Existing #1 also contain the time for the traveler to walk, with any luggage, from the parking space to the bus-stop. We have assumed that this time is a constant 1-minute. There is no such walking time in the Smart #1 values because the bus comes to the space.

We see that the Smart #1 values are almost constant. We understand this because we have determined by doing trial runs that a driver who knows where he/she wants to go can reach any space in the lot from the entrance in a maximum of only 2.5 minutes and also that the Smart #1 system transforms any initial distribution of empty spaces into an almost random distribution of empty spaces as the simulation runs for several days of simulated time.

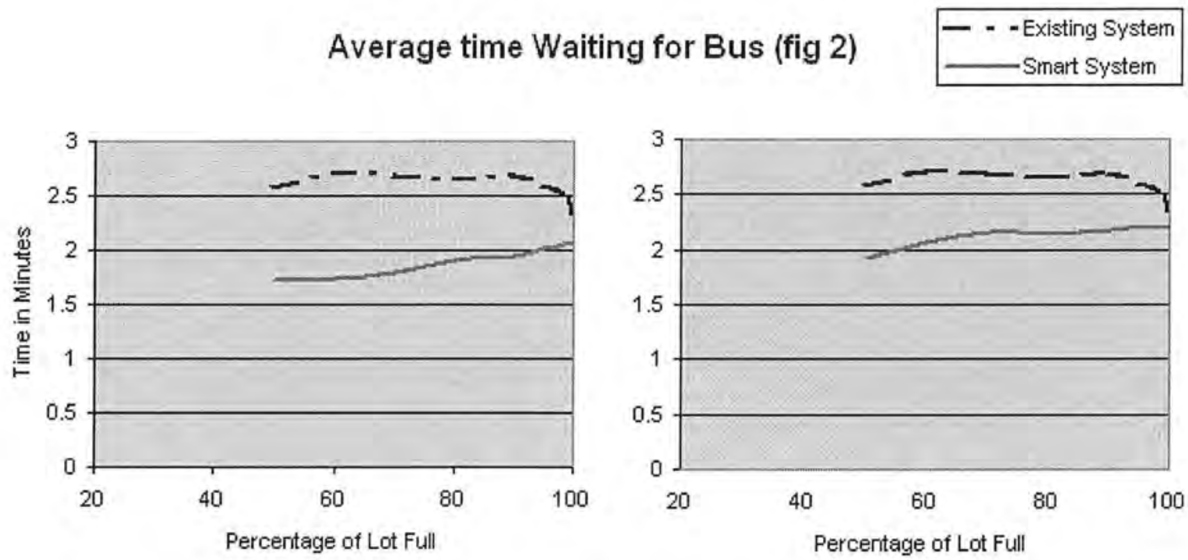
On the other hand, the Existing #1 times are as long as they are partly because, as we noted above, the Existing #1 system transforms any initial distribution of empty spaces into one that has most of the empty spaces far from the entrance.



Before Steady State

After Steady State

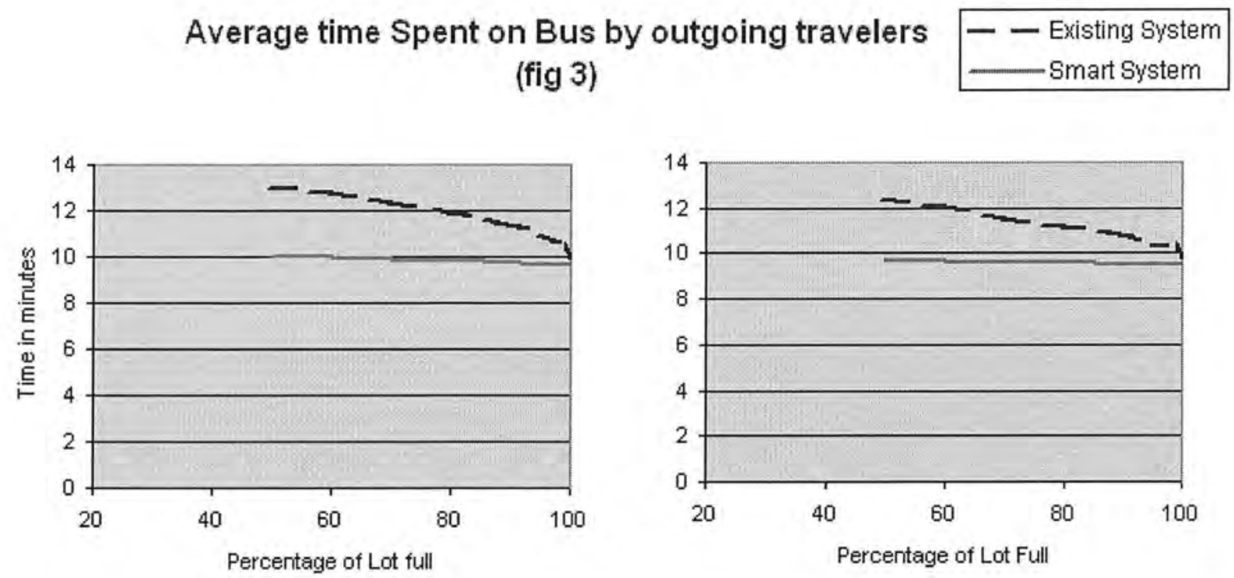
In Fig. 2 we show the average time that a traveler would wait for a bus at the bus-stop for Existing #1 or at her/his space for Smart #1. Here the values for Existing #1 are almost constant; they vary between 2.58 and 2.74 minutes in our simulation, which may just be statistical fluctuation. The interval between busses at PDX is now 6 minutes. The average values for Smart #1 varies between 1.72 and 2.19 minutes. The Smart values vary more than the Existing values partly because, when the lot gets full, the customers cannot be assigned spaces in the route of the shuttle, as there may be no spaces which are empty in its route, hence the bus travels a little more distance and the passengers wait a little more.



Before Steady State

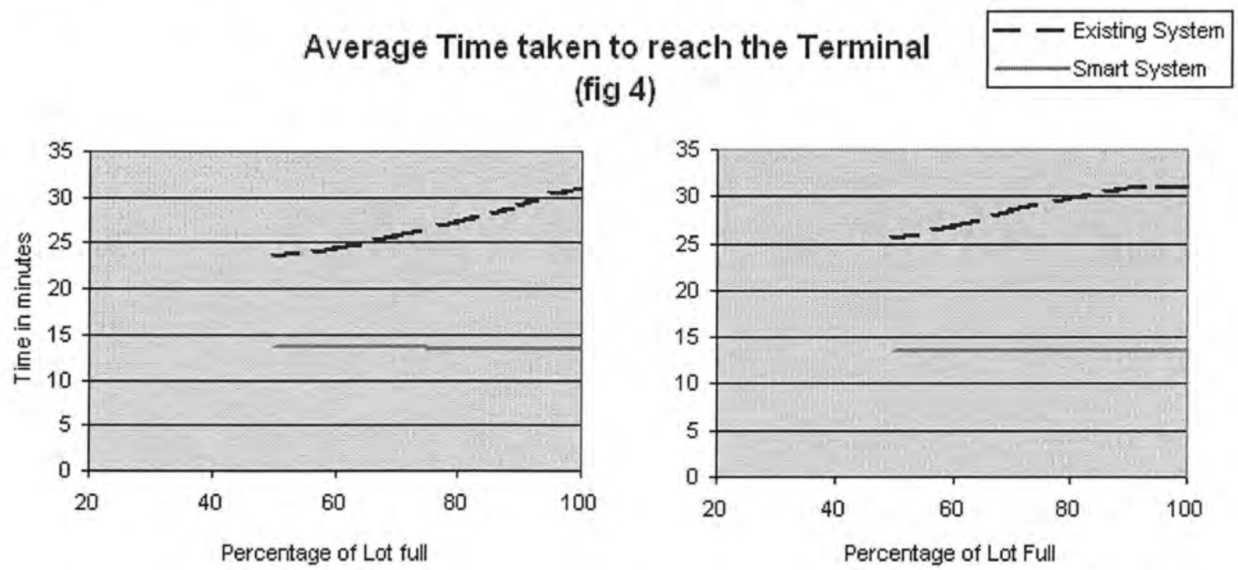
After Steady State

In Fig. 3 we show the average time that the traveler would spend on the bus going to terminal. This is the sum of the time on the bus within the Lot and the 6 minutes going from the Lot to the Terminal. For the Existing system these values range from 13.17 to 10.26 minutes. For the Smart system they range from 10.03 to 9.51 minutes. The advantage of the Smart system is partly because the Existing system travelers board the bus closer, on average, to the entrance and partly because the Smart system busses do not waste time going where they are not needed. The worst case times for our lowest level smart system are 10.7 minutes for all occupancy factors from 50 to 99.9% full.



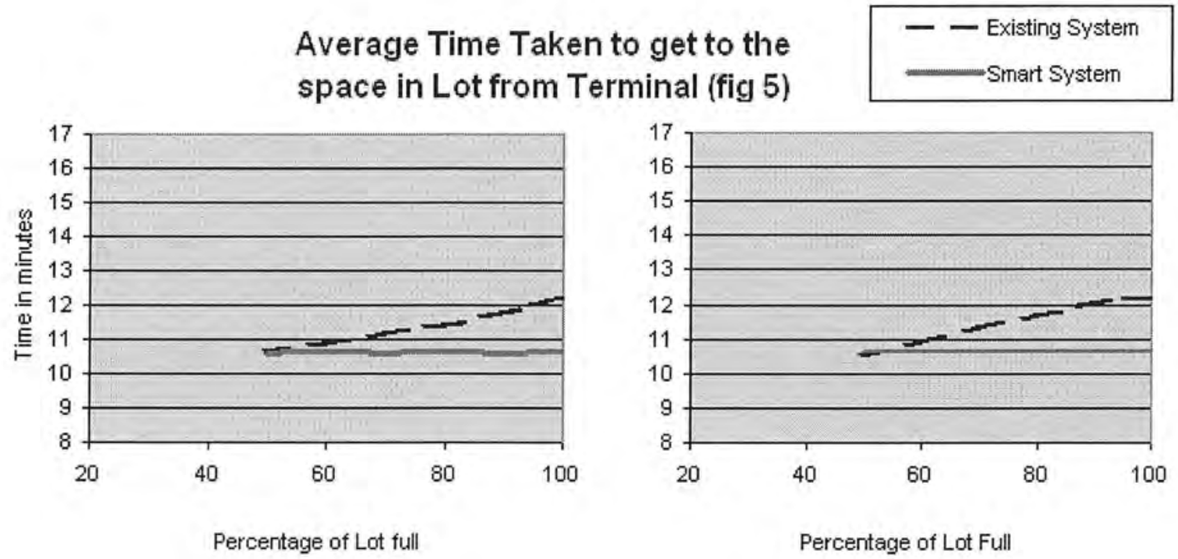
Before Steady State | After Steady State

In Fig. 4 we show ITT-In, the average time between entering the Lot and arriving at the Terminal. Again the Smart values are almost constant and 10 to 17 minutes less than the Existing values.



Before Steady State | After Steady State

In Fig. 5 we show ITT-Out, the average time for the travelers to get to their spaces from the Terminal, i.e., the 6 minutes from Terminal to Lot entrance plus the time on the bus in the Lot, and for the Existing only, 1.0 minutes to walk with luggage from the bus-stop to their vehicle. The Existing values increase, if not by much, 10.54 to 12.16 minutes as the lot fills from 50 to 99.9% because the traveler's space trend away from the entrance as the lot fills. The Smart values are almost constant at 10.6 minutes.



Before Steady State

After Steady State

Chapter 5
Future Work and Conclusions

5.1 Future Work

As shown in Figure 6, we have developed a block diagram for the physical implementation of SPLITS. A pilot system can be built to provide a bench-top demonstration of the SPLITS Smart #1 system.

We also want to consider any real-world effects that are not included in our present simulations. For example, Ken Turner of Trimet, a member of our Regional Experts Panel, told us at our meeting on 4/4/00 that people board busses much more rapidly when the bus is nearly empty than when it is nearly full. He suggested that the system should take account of this fact and therefore should drop off many of the out-going travelers before picking up in-coming travelers. The effect may be particularly strong when travelers have a lot of luggage, as occurs when there are major holidays. We plan to survey this effect and adapt our programs accordingly.

If we do leave off out-going travelers before we pick up any in-coming passengers, then it would often be advantageous to assign the parking spaces identified in Route 1 to in-coming travelers who arrive after the bus has entered the lot. These in-coming passengers would then park in the spaces vacated by the out-going travelers. Then we would reverse the route of the bus.

We could certainly improve ITT-In by assigning clusters of in-coming travelers to clusters of empty spaces so that the bus can load several travelers at one stop. This will require a modest amount of extra programming.

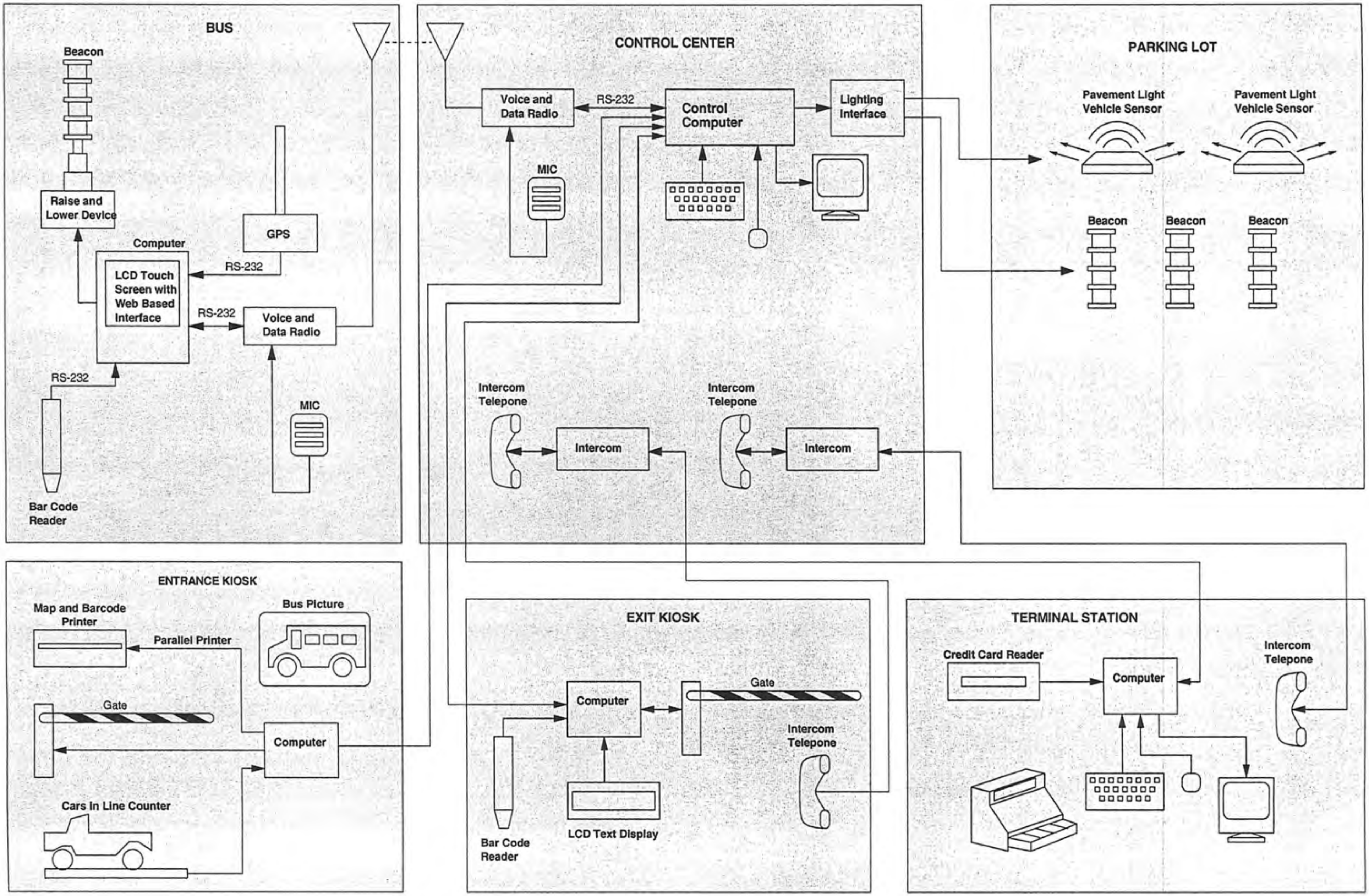
We think that we can improve ITT-OUT significantly if we obtain information about the return flights of the travelers as they enter the lot when there is a correlation between the travelers going out and the travelers coming back as a function of time. We would assign spaces so that the bus could also leave off clusters of travelers with one stop. Correlations occur, e.g., when many travelers are going to the same conference or sporting event or vacation tour. This would have a beneficial effect of creating more clusters of empty spaces so that we would improve ITT-In also. The information that we would need is, of course, in the Airline Flight Reservation Data Bank. With access to that data, we can estimate who will want to park in the lot and to leave it at which times. If the traveler uses a credit card at the entrance, then the computer can search its list of possibles in very little time and decide whether or not to direct this particular traveler to one of the pre-planned clusters.

The lot management could implement this policy by inviting travelers to reserve their parking spaces in advance. There should be a window of times for coming and going and some penalty for not adhering to the reservation if the traveler does not advise the lot's management in good time.

5.2 Conclusions

We have developed our computer program to the point that it could save out-going travelers as much as 17 minutes on average. The worst-case intermodal transition time for the Smart System for a traveler to get to the Terminal is 17 minutes when there is one space available. Perhaps more importantly, travelers could avoid most of the anxiety that they now experience while trying to get onto flights. Jeff Cushman, manager of ACE Airport Parking Management at PDX, has assured us that he will enthusiastically support the deployment of such a system.

Fig. 6


BUS
Beacon
Raise and Lower Device
Computer
LCD Touch Screen with Web Based Interface
GPS
RS-232
RS-232
Voice and Data Radio
RS-232
MIC
Bar Code Reader
CONTROL CENTER
Voice and Data Radio
RS-232
Control Computer
Lighting Interface
MIC
Intercom Telepone
Intercom
Intercom Telepone
Intercom
PARKING LOT
Pavement Light Vehicle Sensor
Pavement Light Vehicle Sensor
Beacon
Beacon
Beacon
ENTRANCE KIOSK
Map and Barcode Printer
Parallel Printer
Bus Picture
Gate
Computer
Cars In Line Counter
EXIT KIOSK
Computer
Gate
Intercom Telepone
Bar Code Reader
LCD Text Display
TERMINAL STATION
Credit Card Reader
Computer
Intercom Telepone

References:

- *Introduction to Algorithms* by Thomas H. Cormen, Charles E. Leiserson, and Ronald L. Rivest

- *Fundamentals of Software Engineering* by Carlo Ghezzi, Mehdi Jazayeri, and Dino Mandrioli.

- *The traveling salesman problem: A guided Tour of Combinatorial Optimization"* Edited by E.L Lawler, JK Lenstra, AHG Rinnooy Kan and DB Shmoys.