LITERATURE SURVEY

There are many vehicle tracking systems in use in both developed and developing countries today. Real-time tracking and management of vehicles has been a field of interest for many researchers and a lot of research work has been done for tracking system.

The GPS/GSM based tracking system is a system that makes use of a global positioning system (GPS) to determine the precise location of a vehicle, person or an asset to which it is attached. The gathered information can either be sorted within the tracking unit or transmitted and sorted on a database management system at a remote location. Transmission in real-time ensures that the propagation delay is minimal. Transmission of information can be done either by SMS, TCP/IP connection over the internet using a GPRS modem or satellite transmission using a satellite modem. The location is visually displayed on a map using a GPS tracking software.

Wajirakumara proposed a vehicle tracking system using GPS and SMS. This is a vehicle tracking system which does real time tracking of vehicles. His developed system touched mostly on the software interface development and the configuration of hardware parts which were not built by him but were bought as a kit. He assembled the kits. This project is different from ours but it has a similarity by incorporating the GPS technology.

Based on the reviewed literature and in consonance with the present systems, we proposed a GPS/GSM vehicle tracking systems with added functionalities not before incorporated in the reviewed literature most system in operation focus on tracking vehicle and not the safety of the students and drivers.

DESIGNED SOFTWARE

The proposed GPS vehicle tracking system is mainly implemented using JAVA 2 Enterprise Edition platform (J2EE 5)

Maps:

Google maps are provided to show the last updated position of all the user`s vehicles, and a static Google map is sent to the mobile application as an image that represents the current location of a selected vehicle. The reports are also provided with a map to show the location history of each vehicle. In addition to that, maps are used to draw geo-fences as circles that cover the area chosen by the administrator user and the coordination of these circles are saved in the database.

Alerts:

SMS notifications are sent to the user`s cell phone when one of these three events occur.

1. Over-Speeding.
2. Deviation from the route shown to the user and taking a detour.
3. Additional updates from the server side.

On server side, an alert will be sent to the vehicle admin in case an accident is detected.

Server Side:

The server side in our system consists of five components that interact with each other to provide the required services. These components are SMS server, GPRS server, Web server, Database and the GSM modem.

Web server:

The web server represents the interface for the client's applications. It receives HTTP requests from the mobile application. The requests will either need a database interaction like initializing user`s account, adding a new vehicle, changing password or it might need sending commands to the GPS device. In the later case, both the SMS server and the GPRS server are used in order to communicate with the GPS device.

<u>Database:</u>

MySQL database that stores all user`s information and her/his vehicles' details such as reports, alerts, etc.

<u>Client side:</u>

The client in our system is a mobile application. It interacts with the server side by sending and receiving HTTP requests as mentioned earlier.

<u>GPRS Server:</u>

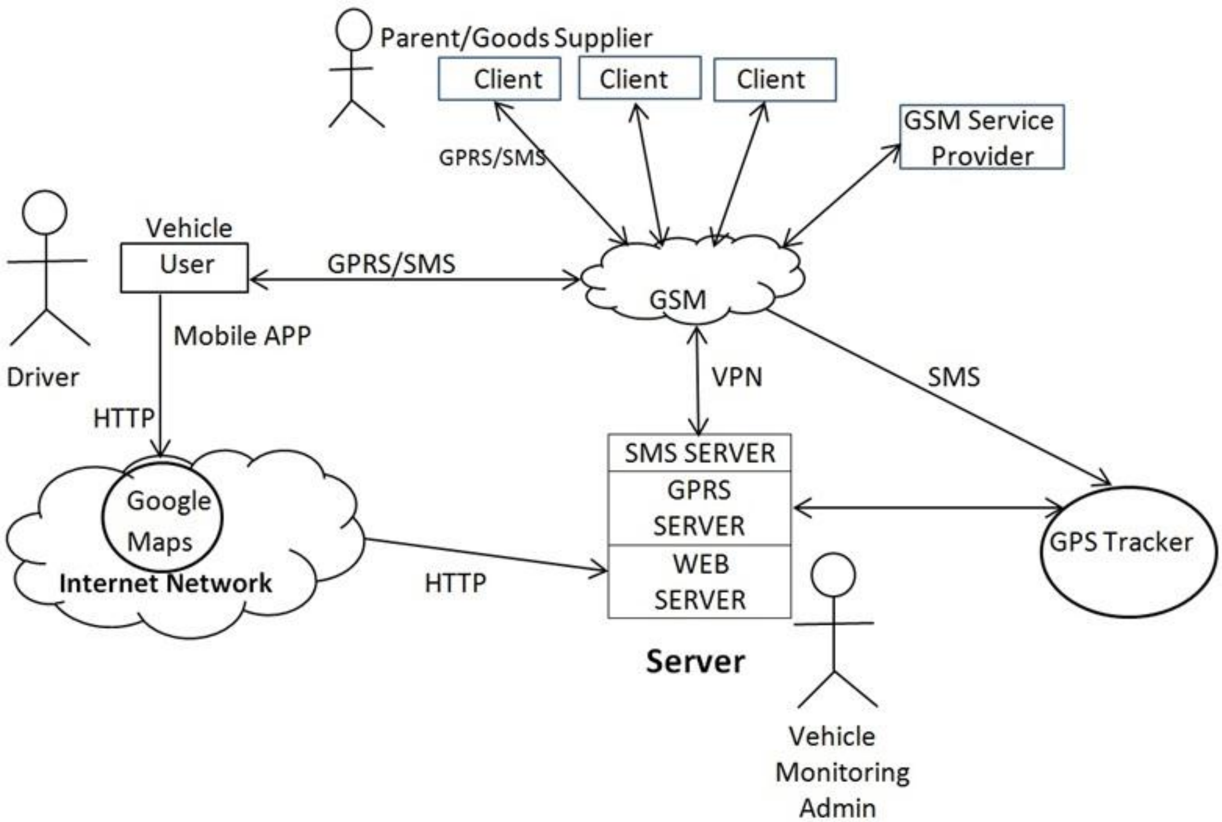
The GPRS server listens to any incoming data from the GPS device using a TCP socket. This data could be one of the following:

1. Periodic tracking reports sent in a user-defined time interval. When it receives these reports, the GPRS server parses these reports and saves them to the database.

2. A response to a command sent to the device through the SMS server, and in this case the GPRS server only forwards these responses to the SMS server.

3. An event notification and this will also be forwarded to the SMS server in order to send the user an SMS alert.

SYSTEM ARCHITECTURE



ALGORITHM

With the development of geographic information systems (GIS) technology, network and transportation analyses within a GIS environment have become a common practice in many application areas. A key problem in network and transportation analyses is the computation of shortest paths between different locations on a network. Although there have been a number of reported evaluations of shortest path algorithms in the literature (e.g., Glover et al. 1985; Gallo and Pallottino 1988; Hung and Divoky 1988), a recent study by Cherkassky et al. (1993) is one of the most comprehensive evaluations of shortest path algorithms to date.

The vehicular routing problem is a variant of the classical shortest path problem, with links representing road segments and vertices referring to road junctions. The links are weighted according to several parameters such as travel distance and travel time.

Several shortest path algorithms have been suggested by the researchers and Dijkstra's shortest path algorithm is the most appropriate one when there is a single source-single destination problem. Though Dijkstra's algorithm is targeted towards single source-single destination problem but it considers only the weights or distance between the nodes as a criterion for selecting the shortest path. Taking the real road networks into consideration, we can suggested a modification of the Dijkstra's algorithm, the multi-parameter Dijkstra’s algorithm (MDSP) that considers multiple parameters into consideration. Apart from the distance between any two nodes, it considers factors such as time taken to travel from the source to the destination, congestion of the route etc. so that the user can select the desired route based on his/her preferences.

For ITS systems, the available time to provide a re-routing solution must be limited due to road network constraints. In what follows, we classify the existing solutions into two categories; MDSP and Heuristic.

Modified Dijkstra’s Shortest Path Algorithm (MDSP)

A new shortest path algorithm called Modified Dijkstra’s Shortest Path algorithm (MDSP) is proposed. In this algorithm multiple parameters were used to find the valid shortest path instead of using single parameter. The efficiency of the MDSP algorithm is analyzed in terms of shortest path by measuring its nodes and Time complexity.

Bauer et al. identified that there was a need to find an efficient shortest path route for the road network. Hence, they developed a new shortest path algorithm by modifying the Dijkstra’s shortest path algorithm using Combining hierarchical and goal-directed technique. This algorithm shows the better results than the existing Dijkstra’s shortest path algorithm but it take high computational time than the existing algorithm.

Algorithm 1: MDSP

```
begin
    for each vertex v in Graph do
        alternate_path[i]:=NULL;
        dist[v] := infinity;
        weight_update(choice);
        for each vertex v in Graph do
            if v= source or v = destination then
                for each neighbour u of v do
                    if alternate_path[i] > dist[u] + distance(u,v) then
                        alternate_path[i]:= dist[u] + distance(u,v);
                    end if
                end for
            end if
        end for
    end for
end
```

Algorithm 2: Including Multiple Parameter

```
begin
    if c = d then;
    else if c = t then
        d := d * s;
    else
        d:= d* z;
        return d;
end
```

where
c = choice
d= distance
t=time
s=time factor
z=congestion factor

Heuristic Algorithms (HA)

In 2004, Bogdan Tatomir et al. proposed a routing solution based on dynamic traffic data to tackle road traffic congestion problem using Ant Based Control (ABC). Ants can always find the shortest route for food source due to their particular method of information exchange. The authors use backward-forward ants mechanism to update the routing tables of agents distributed on each junction. Three years later, Hitoshi Kanoh applied the virus genetic algorithm (GA) to dynamic route planning. According to the principle of species evolution, the algorithm can find the optimal route, and then improve it iteratively.

Just one year later, an improved version of this approach has been proposed by the same author to balance the travel distance, travel time and easiness of driving with a better quality of initial infected virus generated by MDSP. Horst Wedde et al. introduced a new system called Bee Jam Avoidance (BJA) inspired by the form of bee foraging communication.

This distributed, multi-layered and adaptive congestion avoidance system can reduce the average travel time but cannot deal with other criteria. Several approaches have been proposed for smart routing of vehicles, but none of them is able to react efficiently and within short time threshold to the potential change in road conditions during the vehicle journey. This is because they use a fixed computational time threshold which is less-efficient and unrealistic. Additionally, no performance comparison among them is conducted and most of them have been evaluated against MDSP only.

Goldberg et al. analyzed the various shortest path algorithms. They found that there is lot of problems present in the existing algorithms. Hence they developed a shortest path algorithm called A* shortest path algorithm. This algorithm is the extension of Dijkstra’s. In this they include the new parameters like cost, modified weighs etc. This algorithm provides the better result compared to the existing Dijkstra’s shortest path algorithm.

The A* algorithm integrates a heuristic into a search procedure. Instead of choosing the next node with the least cost (as measured from the start node), the choice of node is based on the cost from the start node plus an estimate of proximity to the destination (a heuristic estimate). F. Engineer described this approach to solve the problem of optimal path finding. This project uses Euclidean distance as estimated distance to the destination. In the searching, the cost of a node V could be calculated as:

f(V) = distance from S to V + estimate of the distance to D.
= d(V) + h(V,D)
= d(V) + sqrt((x(V) – x(D))2 + (y(V) – y(D))2)

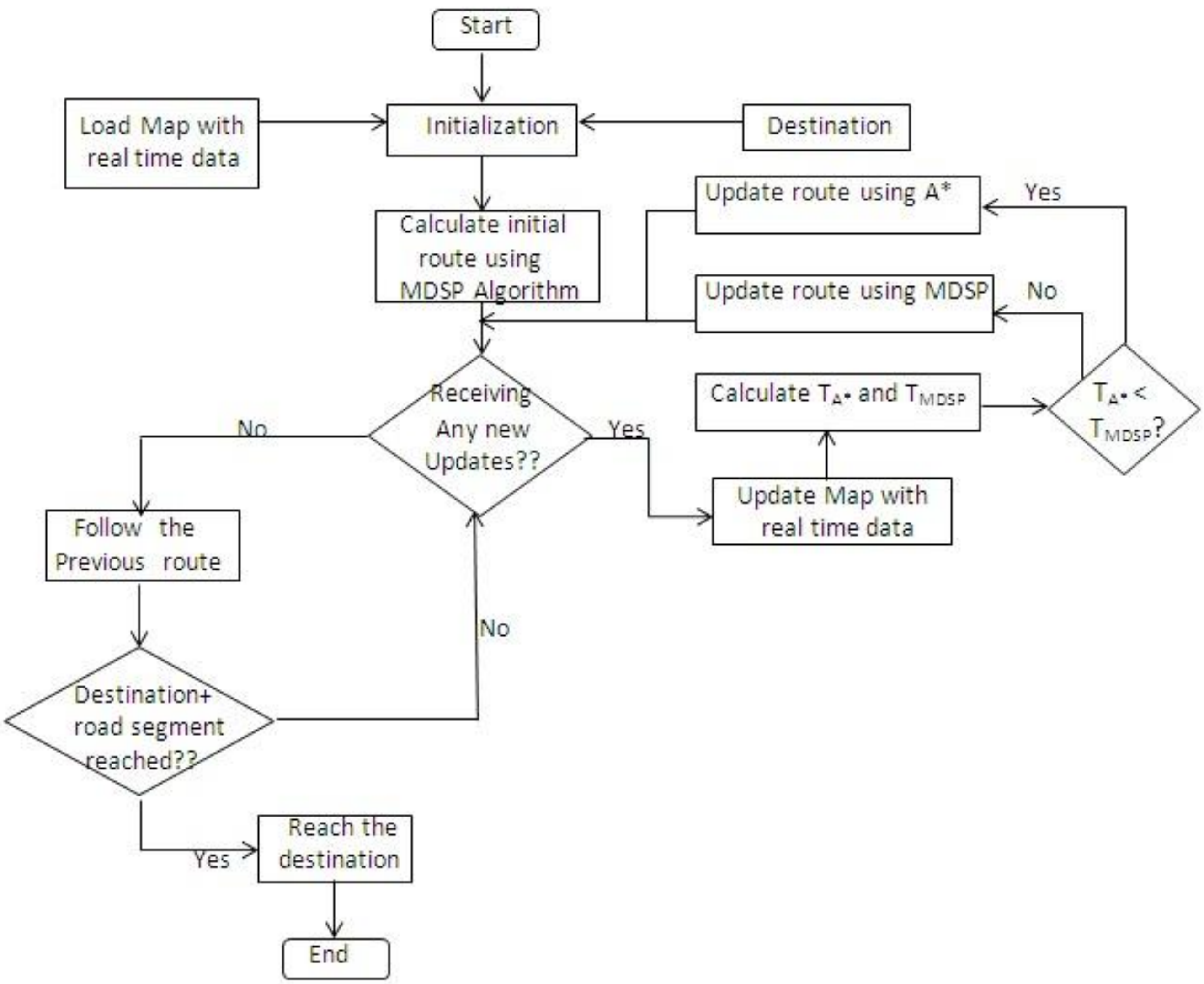
where x(V), y(D) and x(V), y(D) are the coordinates for node V and the destination node D.

<u>The A* Search algorithm:</u>

```
for each u belonging to G:
        d[u] = infinity;
        parent[u] = NIL;
end for
d[s] = 0;
f(V) = 0;
H = {s};
while NotEmpty(H) and targetNotFound:
        u = Extract_Min(H);
        label u as examined;
        for each v adjacent to u:
                if d[v] > d[u] + w[u, v] , then
                d[v] = d[u] + w[u, v];
                p[v] = u;
                f(v) = d[v] + h(v, D);
                DecreaseKey[v, H];
        end for
end while
```

The proposed strategy is a hybrid approach which takes full advantage of both MDSP and Heuristic algorithms (A* Search Method) and meets the requirements of dynamic time constraints of real road traffic scenarios.

- The proposed strategy is a hybrid approach which takes full advantage of both MDSP and Heuristic algorithms (A* Search Method) and meets the requirements of dynamic time constraints of real road traffic scenarios.
- The selection of primary algorithm will be from Modified Dijkstra's algorithm and Restricted Search Algorithm.
- While implementation of the system, the selection will be made based on the ease of implementation as well as considering other factors like time and space complexities.



To overcome the drawbacks of the aforementioned solutions, we propose a hybrid vehicle route planning strategy based on proactive real-time event report mechanism. As shown in the Figure, at the initialisation stage, the road network map is loaded along with real time data and the corresponding prediction information (e.g. estimated average speed of vehicles in a certain road segment, estimated traffic congestion level after X minutes, etc). Simultaneously, the driver should input the desired destination, the vehicle features (e.g. type, height, size, emission of engine etc.) and route planning metrics (travel time, easiness of driving or fuel consumption level) to the system.

Then, the driver gets the best route to destination using MDSP before the journey starts. While the vehicle is running on the road, if it receives real-time event report, the map with real-time data should be updated. Notice that for the whole hybrid re-routing system, the report mechanism works as an interruption in computer architecture; it can push the event information to the related vehicles with highest priority pro-actively, rather than letting the vehicles check the related status periodically.

We assume that the vehicle will use the initial route plan until its destination is reached, if there is no real-time event received during the entire trip. If an event is received, according to the comparison between the computation time of MDSP (TMDSP) and current due time (i.e. estimated travel time from current position to next intersection) Td, the algorithm uses MDSP or HA to provide the new optimal route. In this case, HA can be implemented by A* Algorithm. The idea behind this design is to make full use of the dynamic due time. If Td is long enough, we do not need HA which can only provide near-optimal solution instead of the optimal (best) one.

Finally, the algorithm ends up with a sign of destination road segment reached. It is worth noting that during the vehicle journey, if a re-routing is required then our algorithm will choose the scope of a map from a rectangle whose diagonal is from current position to the destination.