EVALUATION OF DIJKSTRA'S ALGORITHM IN DEMAND RESPONSIVE TRANSIT BASED ON SMART DEVICES

Adel Khelifi[1], Razi Iqbal[2], Husnain Sherazi[3], Waqas Ahmad[4]
[1,2]American University in the Emirates, Dubai, United Arab Emirates
[3]Telematics Lab, DEI, Politecnico di Bari, Bari, Italy
[4]Lahore Leads University

ABSTRACT: *Intelligent transport system (ITS) is a set of applications responsible for providing services to different modes of transportation (such as traffic management, coordination, and information dissemination) to ensure the journey as smooth as possible. One of the primary modes of ITS can be referred to as Demand Responsive Transit (DRT) offering services keeping in view the demands of end users. DRT services offer flexibility to the users for choosing their routes based on the runtime information acquired by smart devices. Routes are predefined but users can modify the predefined routes as per their requirements. DRT is specifically aimed at improving transport services in low demand regions by different government and semi-government agencies. Private companies have also started deploying these services for commercial purposes. There has been numerous algorithms and techniques conventionally used for the implementation of DRTs. In this paper, the deployment of Dijkstra's algorithm has been presented as a part of employee pick-n-drop system. First, we propose the system model for such DRTs customized for Employee pick-n-drop service based on smart devices. Then, several simulations have been conducted to exhibit the superiority of Dijkstra's algorithm over the other conventional techniques in such kind of DRTs that encourages the widespread future deployment based on this model.*

KEYWORDS: Demand Responsive Transit, Intelligent Transportation Systems, Dijkstra's Algorithm, Wireless Technology

INTRODUCTION

ITS (Intelligent Transportation System) are smart transport systems. ITS is used to collect data about transport and process this data to improve the performance of transport systems for users. ITS provides users safety, increased throughput, cost effectiveness and quality of service. It enhances the usefulness of transportation services. It helps in improving the traffic congestion on the roads. Furthermore, it reduces the overall cost of the system, passenger travel time and wait time. An efficient deployment of ITS can significantly improves the overall transportation system of the city. One of the major advantages of the ITS is that they provide real time information for moving objects, e.g., vehicles. Different tools are used to collect process and transfer this information. Several ITS are working throughout the world to improve the road safety, overall transportation efficiency, toll collection, accident avoidance and traffic surveillance.

ITS applications work with information and control technologies. Information is core to ITS. Information follows in a chain which includes data gathering from transportation system, communication, data processing, information sharing, and information utilization for decision making for the users. ITS enabling technologies include location referencing (automatic vehicle location, Mobile phone location, and global navigation system), Data acquisition (traffic detection, incident detection, and vehicle detection), Data processing (map matching), communication, and Information sharing and utilization

DRT service is not operated in fixed routes it provides flexibility to the users. ITS provides DRT service in scheduling time for journey. In DRT although the routes are pre-scheduled but still vehicles can change the routes based on the demand of the passenger. If a new passenger requests for a pickup, the control center of DRT calculates the shortest path for all the available vehicles on the road and the vehicle closest to the passenger picks him up. This information of picking up a new passenger is transferred to the vehicle (driver of the vehicle) through special devices installed in the vehicles. On the other hand, this service also alerts the users about the vehicle location, which is on its way to pick him up. This real time information sharing and flexibility of vehicle routes improves the overall efficiency and reliability of the system. DRT service operates on the user demand, which provide the facility to pick up and drop off the user on their specific locations. This service operates mostly in the low population areas where regular transport is not common. Target users for this type of transportation systems can be old age people, airport pickup and drop facility and school buses etc. Some private organizations manage DRT services for commercial purpose. Organizations charge from passengers according to the time to reach the destination and distance travelled.

DRT emphasizes more on reducing passengers waiting time at the origin, riding time in the bus, operating cost and maximizing the service quality. This service works in two ways static or dynamic fashion. In the static mode passengers demand for the service first, while in the dynamic mode user demand for the service when service is operating. In the entire scenario with DRT we want to highlight the position of employ, in congested regions where DRT is operated but communication between the employee and the vehicle is nil.

GPS enabled mobiles are common now a day and are capable enough to obtain the position of the employees. Most algorithms are developed in the different cities for calculating the location, their location are chosen by vehicles drivers or company engineers. In real time employees do not know about the vehicle and vehicle does not know about the employees' location. And both employ and vehicle are not coordinated to their head office. Many algorithms are used to find the paths, but dijkstra's algorithm is the best to find the shortest path between the two nodes. In this research work the dijkstra's algorithm is used to find the shortest path. In the context of DRT system, we want to build a scenario that employee stays at the pre-defined location and vehicle will pick that passenger up from that location. Similarly, during the drop off time, the vehicle will drop the passenger at the pre-defined location. However, this scenario can be further elaborated by making vehicle pickup and drop employees from any location.

The remainder of this paper is organized as follows: The next section offers an overview of the current state of the art of DRTs and techniques used in Intelligent Transportation Systems. Section III presents the proposed methodology to be followed presenting the system model, work flows, proposed approach and deployment of Dijkstra's algorithm incorporated in proposed system. While the experimental setup describing the various simulations used for evaluation purpose is elaborated in section IV along with the results. And finally concluding remarks are stated in section V.

LITERATURE REVIEW

A lot of work has been done in Intelligent Transportation System (ITS) and its applications such as Demand Responsive Transport (DRT). DRT service is present in many cities and provides the user flexible route. DRT reduces the dependency of people on private taxis and cars. Private vehicles are very expensive and out of range from the individual users. A large number of papers are available on the work of DRT and their operation. Some literature review summaries are provided under below studied related to the DRT and its working.

Edwards et al, [1] described that in the composite area when passenger makes a request the system finds the best path for the passengers with minimum cost. This research paper defines both static and dynamic type of transport. Both types of requests for the users are considered. Most of consideration in this work is given on cost. Derek assigns the weight for cost, one weight assign to the user cost and the other to the vehicle operation cost. For cost point of view Derek considered the cost of maintenance, fuel and pay of deriver to be minimized. Derek illustrated in his work on how to find the optimal path and to reduce the cost with Matlab Demo. In this demo he notifies the vehicle stop and calculates the distance between these stops. To find the shortest path he used the Dijkstra's algorithm, and also specifies the route with minimum cost. He named the optimal route with minimum cost and shortest path. Some weakness in this research that is passengers are not directly involved in this means they cannot make request for the vehicle; there is no coordination between vehicle and passenger show. Mostly focus is given on minimizing the cost of operating the system, but how all system works is not discussed in detail.

Ghinita et al. [2] mentioned that when a user finds out his nearest location then user sends query about the location to the location server. In this process user sensitive detail can be exposed. To consider this danger Gabriel Ghinita et al. suggests the hybrid technique that gives protection to the user query and also to the database. Gabriel develops this technique with mix of two existing techniques (i) hiding the locations inside cloaking regions (CRs) (ii) encrypting location data using private information retrieval (PIR) protocol. They performed various experiments on the hybrid algorithm and compared their result with the existing algorithms. Their results protect the user request and the location server. They use the cryptographic techniques to find the user queries and process the database.

Gebeyehu [3] described the Traffic Analysis Zone (TAZ) technique in which large amount of area is covered without overlapping the route in minimum time duration. This work identifies the method to find link route between the cities and newly developed area of the city. According to this method the overlapping routes can be reduced along with the reach time. They elaborated this technique with the case study of the city Addis Ababa. In this city no rail service is available so all the focus is on buses. City is separated into 308 TAZ. City administrations provided each zone's population service of buses, Geographical information Systems (GIS) planned the TAZ area, and bus stop and path are also calculated using GIS.

Chamikara et al. [4] mentioned that in Sri Lanka police used the manual map for investigation purpose. Police drew the map with hand. So for these problems authors of this paper suggested algorithms that find the nearest police station on map. This algorithm uses the Geographic Positioning Systems (GPS) and Geographical Information Systems (GIS). Nearest Police Station Detection (NPSD) is a technique that is used when the user requests for the desired map Open Layers API can operate which is connected to the Google Maps and responds to the requests. The nearest location displayed on the web browser and user can select them by clicking on it. In the request longitudes and latitudes could also be specified by the users. Author tests this algorithm on specific area of Sri Lanka (Kandy). This algorithm accepts the coordinates of the user and calculates the nearest police station. They made different experiment using NPSD and found out that their results are 87.92% accurate.

Gomes et al. [5] mentioned that in DRT service context many passengers travelled in a vehicle at a one time and they have different origin and destination points, so in a real time this is very time consuming process because a vehicle may visit the same stop again and again. For these problems Rui Gomes et al. developed a Decision Support System (DSS) service with complete analysis of the European practices. DSS based on simulation model and heuristics. Simulation result shows that how the service can be affected. Heuristics algorithm tries to find the path against the request arriving from the users. This algorithm operates in two ways vehicle point of view and passengers' point of view. Vehicle considered the distance from specific point and number of passengers. While passengers wanted to reach on time in their desired position. Multiple weights are assigned in this algorithm for different factors like number of passengers, starting point, ending point, number of nodes, and for cost.

Monica et al. [6] described that in South Africa (SA) people are poor and they cannot afford the private vehicle. People mostly travel on minibus taxi. Although the overall traffic situation in (South Africa) SA is not satisfied. This work helps the existing minibus taxi industry to integrate DRT+AVM system and help them improve performance. Some facilities that have been implemented in south Africa is on demand vehicle booking center, driver training centers and vehicle monitoring and controlling centers. With DRT the passenger calls the call center or visit on the specific website for the service. As the response to the passenger request vehicle is given the required route. As a sample or test to this technique the activity is performed in the city of South Africa Cape Town. This result shows how their technique is successful using the existing vehicle. This research opens new ways in the field of DRT in South Africa for future.

Quadrifoglio et al. [7] describe that in America DRT service for disable persons are working. To improve this service, authors use the zoning vs. no zoning practices. In zoning, reduction of the empty mile trip in between the vehicles pick up and drop off for customers is used. For more improvement service area is divided into centralized vs. decentralized regions. When the customer requests for the service the particular region is responsible for the response. This work provides the results by simulation. These results show that for time window affect 2 vehicles and 260 miles save. With the increase in time, the customers are not satisfied. Zoning strategy will increase effectiveness but with added more complexities. All the zoning vs. non zoning experiments are taken in city of Los Angeles.

Li et. al, [8] designed a prototype system that serves for DRT. This prototype uses the mix of existing route and the DRT. This system is for customers, derivers and administrator and also provides the deployment and configuration. This system provides a web based interface for user reservation on DRT. Customers request for the reservation using web based interface. These requests are forwarded to the dispatching systems that respond to the user request and respond with the available route and the bus. Customers can also choose their journey on the bus. The customer is given an identification ticket when he completes the process. The navigation system receives the information about reservation via the Internet, and then this navigation system provides derivers directions by help of voice and graphics. System provided in this work is hybrid technique. Users only request for service via website interface no other mean for the request [9]. When the user make request for the service all the requests (that is for DRT or simple route) are received in same place, no separate place for entitled the requests. Customers use the DRT service, when the user request for the service of DRT then short route is given to the DRT service [10].

Magzhan et al. [11] described a comparison of different shortest path algorithms with Dijkstra's algorithm. Main focus is given on finding shortest path, and how the graphs are used in computer networks to identify the shortest path. Dijkstra's algorithm and its working are enlightened. Each algorithm is explained with graphs and evaluates the results, and illustrated the working in the computer networks. Time complexities of all the algorithms are calculated. They compare the results of each algorithm and their time complexities. Each algorithm finds the shortest path and explains in graphical form to find out an optimal result [12]. On the basis of these entire algorithms they develop another algorithm called it Genetic Algorithm (GA). GA calculates the more optimal solutions then other algorithms [13].

Reviews of the literature shows that lot of work has been done in field of DRT. Some of them discussed its working keeping in view the cost point of view that how the cost of DRT system can be minimized. Some paper described algorithm about DRT according to different point of view, other work is on simulation technique but every paper uses different tools GPS (Global Positioning System), Matlab, and server computer, GIS (Geographic Information System) and short memory. Review of many paper show that no smart phone is used in the DRT system. Smart phones are very common and every tool is available in these like GPS (Global Positioning System), GPRS (General Packet Radio Service), and Wi-Fi (Wireless Fidelity) so smart phone can be used in DRT. This paper comes up with an idea of calculating the shortest path with the help of DRT and smart device. Smart phones are used to display the position of the users.

METHODOLOGY

A. System Model

The system model for DRT customized for employees pick-n-drops service has been presented in Fig.1. When employees want to reach their offices they are in hurry and they want to avoid traffic. In this diagram node represents the employee. Nodes are on different locations; they came out from their home to suitable location where vehicle can reach. Nodes are smart phones with GPS, Wi-Fi and Cellular Technology facilities.

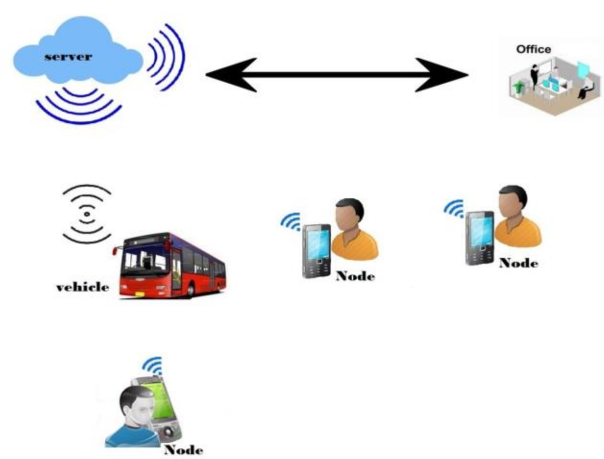

Figure 1: System Model

Vehicle is on the road and far from the nodes. Vehicle does not know about the location of nodes, and vice versa. Nodes and vehicle are communicated via the server. Server is connected with the nodes, vehicle and the head office. Server monitors all the activities. Nodes send their current location to the server, and server stores these locations. Vehicle asks the location of nodes from server, server sends these locations to the vehicle. When server receives the location from nodes, server calculates the nearest location between the received locations, and sends that nearest location to the vehicle. Vehicle receives that location and pick up the node from that location, vehicle sends the information about picked up node to server. Server sends another location to vehicle; server tells all the activities to head office. Head office has all the information about his employees. If any employ is missed and vehicle does not pick up that employ, then head office coordinated with server and tells the information of that employ. Server then sends the location of employ to the vehicle and vehicle picks up the employ.

B. System Flow

Users

Users are on different places. They have smart devices, they first request for login from the web server. Users send their request to the server. They wait until server confirms their requests. When server confirms the request, users receive the confirmation message. After the confirmation, user sends his location to the server. User asks the location of vehicle from the server. User receives the location.

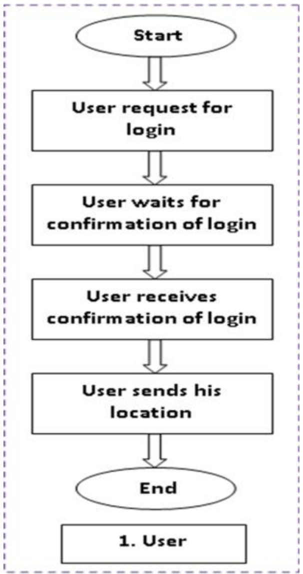

Figure 2: System Flow for User

Web Servers

Web server receives the request of different user. Server checks the requests of user, and authenticates the user. When server authenticates the user, then user can send their locations. Server receives the location of different user. Server stores these locations. Server checks the location of vehicle. Vehicle is also on different location. Fig. 3 below shows the flow of action at server side.

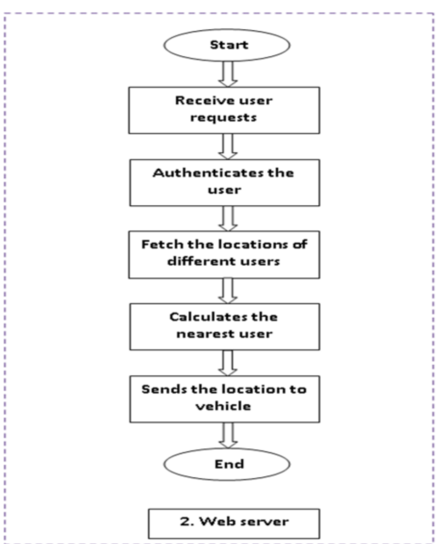

Figure 3: System Flow for Web Server

Vehicle

Vehicle does not know the location of users; vehicle requests the location from server. Server checks the received location from its database, and calculates the nearest user. Server calculates the nearest location using Dijkstra's algorithm. This nearest location is then sent to the vehicle. Fig. 4 below shows the flow of action at vehicle side.

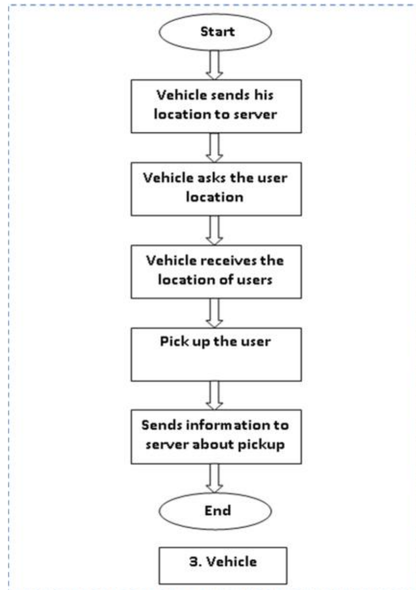

Figure 4: System Flow for Vehicle

When vehicle starts, it sends their location to the server. When server authenticates the vehicle, then vehicle is coordinated to the server, it asks about the location of node. When vehicle receives the location of node, it replies to the server about confirmation of receive locations. Afterwards vehicle also asks about the nearest location of node from the server, when server sends the nearest location to the vehicle, vehicle goes on and pick up the node. After pick up vehicle sends the information to the server about picked up node.

Unified System Flow

Employees are on different locations. Employees have to reach a designated location and send GPS coordinates (location) to the server. Server processes on user's information and stores these coordinates. Server sends the coordinates to vehicle, and tells the office about location of vehicle and employee. After getting coordinates of vehicle and employee, server calculates the shortest route by using Dijkstra's algorithm. When the shortest path is found, this information is sent to the vehicle which then picks up the employee from the said location.

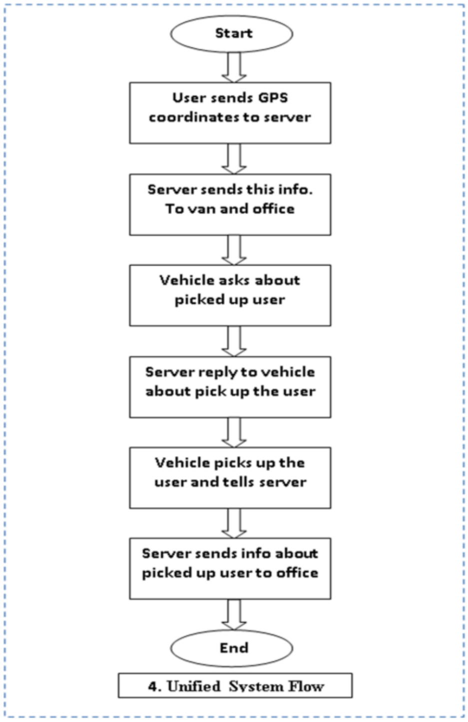

Figure 5: Unified System Flow

C. Approach

In this system GPS enabled smart phone are used. For this purpose, Cartesian coordinates are used to draw the location, each pixel is represented by x and y coordinates. Each node sends its GPS coordinates which are used by the Server to find the shortest path from user's current position to the vehicle using Dijkstra algorithm. When users send request to the servers, server process these requests. Similarly, vehicle's current position is also stored on server, server takes radius around the position of vehicle, when the server finds the shortest distance all the nodes within this radius is selected. Then Dijkstra algorithm calculates the shortest distance between this radii and server uses this information to further action.

D. Dijkstra's Alogorithm

Dijkstra's Algorithm [12] is a graph based algorithm that uses graph to solve the shortest path problem. In this work vertex represents the node, and edges represent distance between the nodes. Source node has edge from every other node; weight of the edge has positive real number. Edge with its positive weight called distance from that node. Every node sends its location to the server; server uses Dijkstra's Algorithm to find the shortest path between the nodes. When server receives all the locations of the nodes, Dijkstra's algorithm runs and finds the shortest path from the node to the vehicle. When shortest path is calculated server checks the presences of vehicle, if vehicle is not on that location then algorithm again runs and finds the new path. Server continuously checks the location of vehicle using GPS.

Equation of Dijkstra's Algorithm

Source node is represented by 's' and edge of node (i,j) length of node represented by Wij. Wij is a positive number. Length is measured by sum of all edge lengths along the path. Sum of all lengths is required for calculating shortest distance. Minimum length of path from s to every node is the shortest distance.

$$D_k = Min(D_i + W_{ik}, D_j + W_{jk}) \qquad (1)$$

'K' is any given node, Dk represent the distance to k. 'S' is a source node, all other nodes are reachable from source node. Predecessors of k are represented by i and j. Here, i is the predecessor of k so that there is an edge (i, k) in the graph. The shortest distance to k passes through either i, or j.
Following diagram explains the operation of Dijkstra's algorithm.

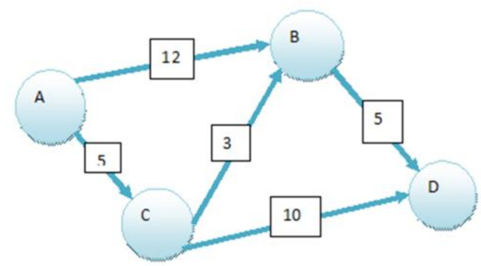

Figure 6: Connected graph exhibiting Dijkstra's algorithm

In this diagram A, B, C, D nodes and their distances are given. A is the source node and D is a destination node. Distance from 'A' to 'B' is 12 and 'A' to 'C' is 5. Short distance from 'A' is 5. So short route is chosen that is 5. Now from node 'C' to node 'B' distance is 3 and node 'C' to node 'D' is 10. Here short distance from 'C' to 'B' is 3 so short route followed. From node 'B' to node 'D' distance is 5. After reviewing and adding up all the distance from node 'A' to node 'D', Path (A to B to D) total cost is 12+5=17, Path (A to C to D) total cost is 5+10=15, Path (A to C to B to D) total cost is 5+3+5=13. Diagram shows that from node 'A' to node 'D' three paths are possible. Minimum path's cost is 13 and hence the shortest distance from node 'A' to node 'D' is 13.

E. Dijkstra's algorithm in DRT

In this work, Dijkstra's algorithm is used with some modification. Node represents the different stops and edges represent the distance between these stops. Employees are also represented as nodes; all employees that are on different locations have smart phones to send their current locations to the server, server checks the location of vehicle on the road. Server has all the nodes location, when the server checks out the location of vehicle, then Dijkstra's algorithm is used for calculating the nearest stops to the vehicle. Different employees walk from home to nearest stop for the vehicle so that vehicle can pick up the employees. Algorithm finds the shortest distance from vehicle to the stop. When the shortest distance is calculated then server sends this information to the vehicle. Server is updated every minute.

SIMULATIONS AND RESULTS

This section presents simulation study conducted for the evaluation of Dijkstra's algorithm in DRTs customized for employees' pick-n-drop system.

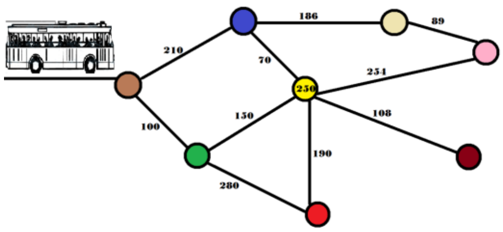

Figure 7: Simulation with Distance Tree (1)

In Fig. 7, vehicle starts from source point and receives the location of two employees the shortest route that the vehicle choose, is calculated with the Dijkstra's algorithm and explanation is given below using Table 1.

Table 1: Results of Firth Short Path

	E1	**E2**	**Short path**
V1	**210**	**100**	**100**

In this table "V1" is a vehicle one and "E1" is employee one and "E2" is employee two. Algorithm use values to calculate the short path. In the first pass dijkstra's algorithm calculates the short path and that is 100. When the vehicle starts from its position, vehicle receives only two employee's location, first employee far from vehicle so vehicle short route is to pick up employee E2. In the first step short route is determined between "E1" and "E2". All the tables calculate short route between two employees. Now again when the vehicle moves on, new locations are received by the vehicle, again algorithm runs and calculates the short route now the short route is 250. At this point three route are available the Dijkstra's algorithm again finds the shortest path the values calculated are given in Table 2.

Table 2: Results of Second Short Path

	E2	**E3**	**Short path**
V1	**100**	**70**	**250**

In this table vehicle receives the location of employees E2 and E3. Based on the information in table 2, E3 is shortest route from vehicle, but actual path is not shortest because vehicle picked up the employee E2. Vehicle is at employee two location. So at that point algorithm calculates the shortest route. First point where vehicle starts location receives is 210 and 100.

Now looking the diagram if calculates short path (210+70=280) another location that vehicle receives of employee four is given in table under below is 150. (100+150=250) so minimum path between these two 280 and 250 is 250. Vehicle short route is 250 that are given in the table. When vehicle moves on locations receives at this point three routes are available.

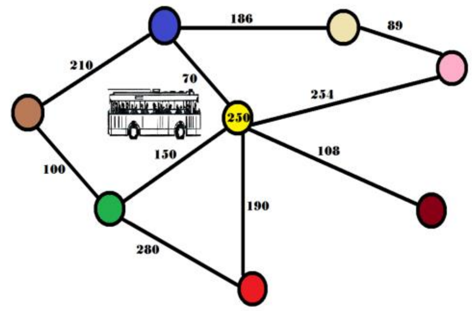

Figure 8: Simulation with Distance Tree (2)

The dijkstra's algorithm again finds the short path the values calculated are given in table 3.

Table 3: Results of Third Short Path

	E3	**E4**	**Short path**
V1	**70**	**150**	**358**

First location received by vehicle is 254 second 108 and the third is 190. Vehicle is at point 250. At that point calculated shortest route (250+254=504), (250+108=358), (250+190=440) is 358.

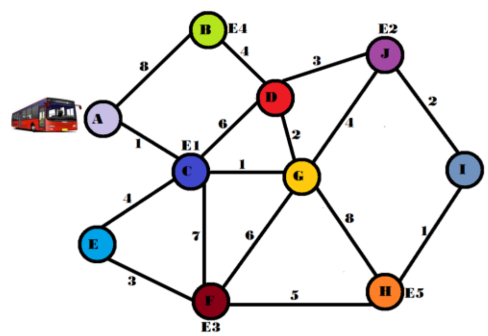

Figure 9: Simulation for Shortest Path

In Fig. 9, nodes A, B, C etc. represent the stop of the bus. Employees are on different stops. Distances between these stops is given in the table 4. E1 represents the employee one, E2 is employee second and so on. E1 is on stop C, E2 is on stop J, E3 is on stop F, E4 is on stop B and E5 is on stop H.

Table 4: Edges and their respective length

Edge	**Length**
(A,B)	**8**
(A,C)	**1**
(B,D)	**4**
(C,D)	**6**
(C,G)	**1**
(C,E)	**4**
(C,F)	**7**
(D,J)	**3**

(D,G)	**2**
(E,F)	**3**
(F,G)	**6**
(F,H)	**5**
(G,H)	**8**
(G,J)	**4**
(H,I)	**1**
(I,J)	**2**

Vehicle starts from node ‘A’, employees send their current location to the server; server sends these locations to the vehicle. When server sends locations to vehicle it calculates the shortest distance using dijkstra’s algorithm. In the first pass vehicle receives location are given in table below.

Table 5: Employees and their distance in 1st pass

	E1	**E2**	**E3**	**E4**	**E5**
V	**1**	**6**	**8**	**8**	**9**

Server calculates shortest distance using algorithm and sends the information to the vehicle. Now looking at the table, E1 is on shortest distance from vehicle. Vehicle picks up E1. When vehicle reaches on stop ‘C’, again shortest path is calculated and sent by the server to vehicle that is given in table below.

Table 6: Employees and their distance in 2nd pass

	E1	**E2**	**E3**	**E4**	**E5**
V	**-**	**5**	**7**	**9**	**7**

Vehicle is on stop ‘C’ and picks up E1. Now from stop C shortest distance calculated using algorithm and given in the table. E2 is on shortest distance. Vehicle picks up the E2. E2 waits on stop ‘J’. From stop ‘J’ shortest distance given below in the table.

Table 7: Employees and their distance in 3rd pass

	E1	**E2**	**E3**	**E4**	**E5**
V	**-**	**-**	**8**	**7**	**3**

Now looking at the table it finds that E5 is on short distance from vehicle, so vehicle picks up E5. E5 is on stop ‘H’ now shortest distance from stop ‘H’ is given in the table below.

Table 8: Employees and their distance in 4th pass

	E1	**E2**	**E3**	**E4**	**E5**
V	**-**	**-**	**5**	**10**	**-**

E3 is shortest distance from vehicle. Vehicle picks up the E3. Now shortest distance calculated using Dijkstra’s algorithm is given below in the table.

Table 9: Employees and their distance in 5th pass

	E1	**E2**	**E3**	**E4**	**E5**
V	**-**	**-**	**-**	**12**	**-**

Finally, vehicle picks up E4. In this figure different routes and their time slots are given. Chart shows the comparison between traditional routes and the Dijkstra’s routes. Employees are on different points or stops, different routes are available from vehicle to employees.

When vehicle starts to pick up the employee, vehicle has different routes available where vehicle can go to pick up but traditional routes are long or time to reach using traditional route is maximum, Dijkstra’s is a shortest path algorithm which calculates the shortest path for vehicle to pick up the employees. R1, R2, R3…Rn are the different routes followed by the vehicles.

Comparison between different available routes are given in Figure 11. Results show that Dijkstra’s calculates shortest path as compared to other conventional techniques that is very helpful for the vehicle to pick up the employees following short route.

CONCLUSION

Demand Responsive Transit (DRT) provides operations in designated areas. A lot of work has been done in the field of DRT. DRT was customized for employees in this paper to make their on-time arrival at offices. Employees share their current location with the server who further lets the vehicle to know the exact pick up location for the employees. As the server is receiving simultaneous requests from multiple passengers, hence it is responsible to decide the sequence of passengers to schedule a timely pick up. Due to dispersed passenger locations within a specific area, vehicle pick up the passengers on the basis of short distance calculated from its own location. Dijkstra’s algorithm is used for calculating the shortest route in this situation. Vehicle can follow their predefined routes but traditional routes are longer enough in terms of distance and time hence leading to higher costs. Dijkstra’s algorithm solves this problem by calculating the shortest routes in DRTs empowering the users to control dynamic situations. A set of simulations have been performed and results were very encouraging. Bar charts show that the

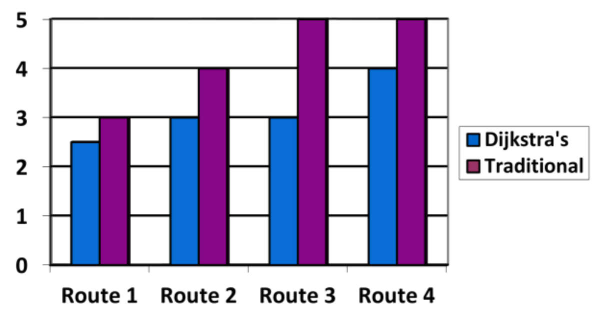

Figure 10: Comparison between Dijkstra’s and Traditional

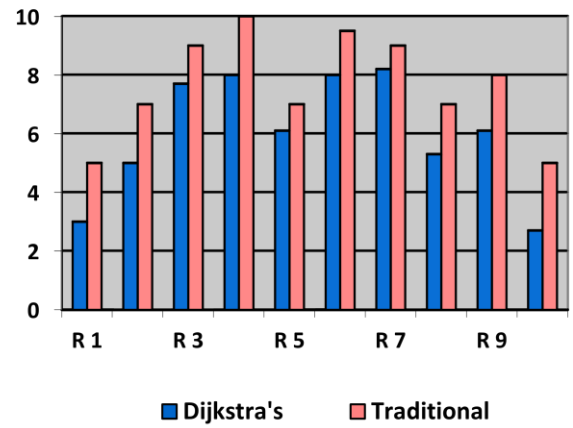

Figure 11: Comparison between Dijkstra’s and Traditional

routes calculated by employing Dijkstra’s algorithm are optimized enough to save the time and cost as compared to the predefined routes DRT.

REFERENCES

1. Derek L. Edwards 2011. A Network Inspired Transportation System Optimal Control of Integrated Demand-Responsive Transportation.
2. Ghinita, G., Kalnis, P., Kantarcioglu, M., & Bertino, E. (2011). Approximate and exact hybrid algorithms for private nearest-neighbor queries with database protection. GeoInformatica, 15(4), 699-726.
3. Gebeyehu, M., & Takano, S. E. (2008). Demand responsive route design: GIS application to link downtowns with expansion areas. Journal of Public Transportation, 11(1), 3.
4. Chamikara, M. A. P., Yapa, Y. P. R. D., Kodituwakku, S. R., & Gunathilake, J. An Efficient Algorithm to Detect The Nearest Location Of A Map For A Given Theme.
5. Gomes, R., de Sousa, J. P., & Dias, T. G. (2012, September). Design and operation of Demand Responsive Transportation systems. In EWGT 2012 International Scientific Conference.
6. Giannini, M. (2009). Demand Responsive Transport as integration to urban regular transport service: a case in Cape Town. In Proceedings of “8th ITS European Congress (Vol. 6).
7. Quadrifoglio, L., Dessouky, M. M., & Ordóñez, F. (2008). A simulation study of demand responsive transit system design. Transportation Research Part A: Policy and Practice, 42(4), 718-737.
8. Reservation, Scheduling, and Navigation System for a Checkpoint DRT Service. California PATH Program, Institute of Transportation Studies, University of California at Berkeley, 2007.
9. Thompson, R. G., Sakulchariyalert, K., Haasz, J., Winter, S., & Mendis, P. Determining the Viability of a Demand Responsive Transport System. methods, 12(2), 11.
10. Engineer, F. (2001, December). Fast shortest path algorithms for large road networks. In Proceedings of 36th annual ORSNZ conference.
11. Magzhan, K., & Jani, H. M. (2013). A Review and Evaluations of Shortest Path Algorithms. International Journal of Scientific & Technology Research, 2(6).
12. Skiena, S. (1990). Dijkstra's Algorithm. Implementing Discrete Mathematics: Combinatorics and Graph Theory with Mathematica, Reading, MA: Addison-Wesley, 225-227.
13. Iqbal, R., Arif, S., & Sherazi, H. (2016). Proficient Car Parking System Based on Cluster Head Routing Protocol Utilizing IEEE 802.15.4. International Journal Of Online Engineering, 12(06), 58.