GRID FABRICATION OF TRAFFIC MAINTENANCE SYSTEM CLUSTERING AT ROAD JUNCTIONS

Avula Anitha[1]

[1]*School of Informatics, Hawassa University, Ethiopia*

Abstract

Controlling congestion has become an important task in current traffic conditions. Different techniques are introduced to control the same. This paper designs and develops a novel traffic flow strategy based on simultaneous and dynamic clustering named Grid Fabrication of Traffic Maintenance System (GFTMS) Clustering algorithm which stands efficient while deciding the route to take to reach the intended destination within the expected time.

Keywords: *Congestion, simultaneous and dynamic Clustering, GFTMS Clustering.*

--***--

1. INTRODUCTION

The basic purpose of Traffic Management System is to control the congestion in the road network. To achieve this several congestion control techniques are given. But, when it should be controlled, it is necessary to exchange the information so that routing the vehicles through the less congested route may decrease the time to reach the destination. It can be through VANETs [4] and can also be through the Grid Computing [2, 3, 7]. When the vehicles get accumulated at junctions, the junctions have to take the responsibility of scheduling vehicles through less congested routes which can minimize both the travel time and the loss occurring due to extended waiting in the traffic jams.

For exchanging the information and for publishing aspects among the nodes of road traffic network by developing the routing strategies or alternative routing strategies which are computationally intensive High Performance Computing Facilities, distributed computing and synchronization of nodes are required. Grid Computing using Grid Services will stand useful. Thus a Grid Fabrication for Traffic Maintenance System (GFTMS) [1] is thought as a better solution which provides computational support as well as stateful web service.

If decision making is done at junctions, treated as Grid nodes [1], vehicles scheduled through the less congested routes leads to inferior solution because the addition of this load leads to inferior performance. Thus, load distribution among alternate paths for effective throughput needs to be worked out. Hence distributing the traffic among alternate routes is the subject of our work. To achieve this, some technique should be there to group the vehicles according to different criteria such as destination, priority, type of vehicle, so that routing of emergency, important and higher priority vehicles can be done through less congested route. Therefore, for grouping the vehicles at junctions or nodes, clustering stands an efficient technique. In this paper, for clustering of vehicles at nodes, the GFTMS-Clustering algorithm is proposed. This algorithm addresses step 1 of the proposed Grid Fabrication in [1] and it is a combination of Hierarchical clustering as the clustering is done through different levels and Leaders algorithm [11] as the number of clusters to be created cannot be decided prior. The number of clusters created depends purely on type, destination and priority of the vehicles considered. In the present study we are confining to only these features and one can consider all the features which influence QOS characteristics of traffic network instead.

The work of [10] has lead to the idea of constructing more stable clusters of vehicles to route the important, higher priority vehicles through the less congested routes.

In this paper, the following Section 2 gives the traffic maintenance analogy in Grid System. In Section 3 an overview of Grid System and clustering is covered. Section 4 covers the related work in controlling congestion using VAN based, Grid based approaches and some clustering strategies. Motivation for proceeding with this work and the proposed algorithm for clustering the vehicles at junction are given in Section 5. Section 6 demonstrates GFTMS-Clustering strategy and provides embedding strategy for real time road traffic monitoring and controlling system along with the comparison of the works which has motivated to proceed further with this paper. Section 7 provides simulation results for the contributions made in this paper and represents the comparison with and without using the proposed clustering approach. Conclusion is dealt in Section 8.

2. TRAFFIC MAINTENANCE ANALOGY IN GRID SYSTEM

The work proposed in [1], describes junctions as Grid Nodes. Road is a connection between two nodes. These are the services or resources available at nodes. The resources are utilized by the vehicles which will occupy the capacity and forms the load. The vehicles are the users and mobiles in the road network. Each vehicle will publish the information to the Grid Node or junction. The Grid node will compute the traffic parameters for the resources connected to it and will in turn exchange the information with other Grid nodes linked to it directly. This exchanging of information among Grid nodes will help the nodes to decide about the next best hop and the alternate hop to be taken by each vehicle accurately to avoid the formation of congestion at other nodes and also implicit occupation of the resources. The same analogy is considered even in this paper also

3. GRID SYSTEM AND CLUSTERING

3.1 Grid System

The supercomputing held in San Diego in 1995 has lead to the emergence of Grid Computing platform. The Global Grid Forum (GGF) also known as Open Grid Forum (OGF) developed standard interfaces, behaviour, core semantics etc. for grid applications based upon web services [8].

We can classify service state management in web services into two forms. 1. Interaction aware state, in which a client may interact with a server for a long period of time. These interactions are correlated with some information passed from the client along with the message such as session-id, cookies etc. In this the server will not create a specific instance for client and it won't manage any client state information. 2. Application aware state, in which the services are aware of its clients and create a specific instance for the specific client and pass the instance information back to the client for interaction. Therefore the client is holding a reference to the specific instance of the service/application and hence can interact with the service instance without passing any correlation information. These services are referred to as stateful services. Grid services are the stateful web services with a well-defined interfaces and behaviour for interactions.

In our problem, at any moment any junction can behave as an information requestor or information responder and to achieve this Service Oriented Approach (SOA) is required which can be done using either Web Services or Grid Services. As Web Services are stateless and the information gathered should be valid and existing for a long time, the statefulness should be provided. Therefore in our approach Grid Technology is used.

Grid Resource Access Manager is the module in Grid that provides the remote execution and status management of the execution. When a job is submitted by a client, the request is sent to the remote host and handled by the gatekeeper daemon located in the remote host. The gatekeeper creates a job manager to start and monitor the job. When the job is finished, the job manager sends the status information back to the client and terminates. Job manager is created by the gatekeeper daemon as part of the job requesting process using Resource Specification Language (RSL) by the clients. All job submission requests are described in RSL including the executable file and condition on which it must be executed. Job manager provides the interface that controls the allocation of each local resource manager such as a job scheduler like PBS or load-leveller. The functions are parse RSL, allocate job requests to the local resource manager, send call-backs to clients, receive status, cancel requests from clients and send output results to clients using Global Access to Secondary Storage if requested (GASS) [9].

3.2 Clustering

Clustering typically groups data into sets in such a way that the intra-cluster similarity is maximized while the inter-cluster similarity is minimized. The clustering technique has been extensively studied in many fields such as pattern recognition, customer segmentation similarity search and trend analysis. Data clustering is a method in which we make cluster of objects that are somehow similar in characteristics. The criterion for checking the similarity is implementation dependent. Clustering is often confused with classification, but there is some difference between the two. In classification the objects are assigned to predefined classes, whereas in clustering the classes are also to be defined. Precisely, Data Clustering is a technique in which, the information that is logically similar is physically stored together. In order to increase the efficiency in the database systems the number of disk accesses is to be minimized. In clustering the objects of similar properties are placed in one class of objects and a single access to the disk makes the entire class available. The distance between two clusters involves some or all elements of the two clusters. The clustering method determines how the distance should be computed.

A similarity measure SIMILAR (Di, D_j) can be used to represent the similarity between the documents or records. Typical similarity generates values of 0 for exhibiting no agreement among the assigned indexed terms and 1 when perfect agreement is detected. Intermediate values are obtained for cases of partial agreement. Threshold is used as the lowest possible input value of similarity required to join two objects in one cluster. Similarity between objects calculated by the function SIMILAR ($D_{i,}$ Dj), represented in the form of a matrix is called a similarity matrix. The dissimilarity coefficient of two clusters is defined to be the distance between them. The smaller the value of dissimilarity coefficient, the more similar two clusters are. First document or object considered of a cluster is defined as the initiator of that cluster i.e. every incoming object's similarity is compared with the initiator. The initiator is called the cluster seed.

There are many clustering methods available, and each of them may give a different grouping of a dataset. The choice of a particular method will depend on the type of output desired, the known performance of method with particular types of data, the hardware and software facilities available and the size of the dataset.

In general, clustering methods may be divided into two categories based on the cluster structure which they produce. 1. The non-hierarchical methods divide a dataset of N objects into M clusters, with or without overlap. These methods are sometimes divided into partitioning methods, in which the classes are mutually exclusive, and the less common clumping method, in which overlap is allowed. Each object is a member of the cluster with which it is more similar. However, the threshold of similarity has to be defined. The hierarchical methods produce a set of nested clusters in which each pair of objects or clusters is progressively nested in a larger cluster until only one cluster remains. 2. The hierarchical methods can be further divided into agglomerative or divisive methods. In agglomerative method, the hierarchy is build up in a series of N-1 agglomerations, or fusion, of pairs of objects, beginning with the un-clustered dataset. The less common divisive methods begin with all objects in a single cluster and at each of N-1 steps divide some clusters into two smaller clusters, until each object resides in its own cluster.

4. RELATED WORK

4.1 Vanet Based

Road traffic management mainly concentrates on alleviating the congestion and reducing the time taken by the vehicles to reach the intended destination by providing alternative routes based on the road conditions. The congestion controlling techniques introduced in wireless networks mainly VANETs are evolving much. The vehicular communication facilitates the exchange of information between vehicles in the prerequisite not only for extending the access to internet while on the road, but also to cater for special applications such as of road traffic and travel management. Novel vehicular communication system for road traffic congestion detection and dissipation by disseminating and exploiting road information is proposed in [4]. In this a method is provided for tuning mechanism to select the importance given to the least congestions over the travel distance. A flooding based geo-cast protocol using the Zone of Relevance(ZOR) concept for disseminating and exchanging road traffic information in a real time way and a modified version of Dijkstra algorithm which is used to dynamically recalculate the vehicles route to a given destination by finding the least congested route is introduced.

In [5] a method for scheduling visits of a tour is discussed and the routes used for each of several thousands of users which satisfy their needs as much as possible, while avoiding congestion. Given the tour plans of users in advance, the proposed method predicts congestion on each road and at each destination for every second, it generates a feasible schedule for each user by modifying each plan, so that the user can visit as many places as possible within the overall constraints. In this a heuristic algorithm to determine schedules of users from a given road network with service spots and tour plans of the users is introduced. The algorithm iteratively removes the least important spot from each user's plans of all users so that the set of modified plans of all users satisfies time constraints, taking into account the capacity of the roads and the destinations.

In [6] a new routing protocol called CCSR to improve the performance of multipath routing protocol for adhoc wireless networks is proposed. The CCSR uses the cumulative congestion status of the path rather than congestion status of the neighbourhood. According to the values of the congestion status for the path stored in a separate table and maintained by source node for processing the source node will distribute the packets such that more packets move to paths with less congestion. In this CCSR outperforms both Adhoc on-demand Distance Vector routing protocol (AODV) and Dynamic Source Routing protocol (DSR) because it balances the load according to the situation of the network and adaptively changes the decision by source node.

In road traffic management, the concept of clustering is playing an important role. In Vehicular Networks [10], constructing clusters can improve the performance of wireless communications such as data forwarding, handoff, etc. In this N-hop cluster is a cluster in which the cluster head node is the N-hop neighbour of all cluster member nodes. As a result, the maximum distance in hops between the cluster head node and the cluster member nodes is N. The beacon messages are broadcasted by the vehicles. Upon receiving them vehicle nodes can calculate relative mobility with other vehicle nodes in its N-hop neighbourhood.

4.2 Grid Based

In [7] the vehicles are navigated and assigned to the optimal routes according to the information of real time and predicted traffic flow so that the optimization of traffic flow system and traveller's satisfaction degree can be both achieved. In this paper Genetic Algorithm is employed to solve the optimization problem in which the genes of each chromosome(decision variable) are to record the path number from the intersection before running the vehicle to the destination with constant signalling scheme at intersection. Through genetic operators the best routes can be evolved out. The web services are utilized to realize the grid computing of parallel Genetic Algorithm, where several services such as initializing parameters, selection, crossover and mutation are computed parallel.

4.3 Clustering Strategies

The paper [12] studies and compares different data clustering algorithm such as K-means, Hierarchical, Self-Organizing Maps and Expectation Maximization algorithms. All these algorithms are compared according to the following factors: size of dataset, number of clusters and type of dataset used.

5. MOTIVATION AND PROPOSED APPROACH

5.1 Motivation

K-Means[12] is a partitioning clustering method where partitions of a database of N objects constructs a set of K clusters with respect to an objective function. In this the input K i.e. number of clusters for partitioning is mandatory. In road network monitoring and management system, the dynamic nature of accumulation of vehicles at junctions makes it difficult to give exact K value. At different instances, different number of clusters is created depending on type, destination and priority of the vehicles accumulated at junction, hence K cannot be predicted. As, K-means is restricted to the notion of centre (centroid), assuming the centre object (here vehicle) is difficult.

DBScan [12] is a density based clustering method. These methods permit the growing of a given cluster until the number of objects or data points (i.e. density) in the neighborhood exceeds some limits. In case of clustering of vehicles at junctions, the limit cannot be predetermined, because at a given instance, there may be a situation where all vehicles may be of same type or some times, the percentage of different types of vehicles with different priorities and destinations will be varying.

Model based clustering Methods [13] builds a cluster on the basis of a model that receives the best data among others. A density function is built by model-based algorithm to locate clusters. The density function defines the spatial distribution of the data points. The COBWEB is a conceptual learning algorithm that performs probability analysis and takes 'concept' for modelling clusters. Expectation Maximization is based on statistical modelling and performs expectation maximization analysis. Self-Organizing Maps is a neural network based algorithm. It maps high dimensional data into a 2D or 3D feature map. When the vehicles are getting accumulated at junctions, there will be no choice of receiving best data among others. Every object (vehicle) will be important and every object has to be scheduled.

Hierarchical Agglomerative clustering [12] starts with every object forming a separate group and then successively combines the objects or groups that are near to one another until all groups are combined into one termination condition is satisfied. This is also referred to as bottom-up approach. In Hierarchical Divisive clustering [12], it starts with all objects in a single cluster and then on successive iterations a cluster is broken down into smaller clusters until each cluster has a single object or until termination condition is satisfied. This method is also referred to as top-down approach. Hierarchical Divisive clustering can be applied to cluster the vehicles at junctions with buffer zones. This requires that all the data objects (here vehicles) should be existing to apply the respective algorithms. The algorithm cannot be triggered until the completion of accumulation of vehicles at the buffer zone. If it is triggered prior, then the new vehicle joining the junction cannot be added to the existing, already created clusters. This is because the hierarchical clustering will not allow addition or reallocation once the step of dividing or combining is completed. As, we are clustering the vehicles using the vehicle properties, i.e. type, destination and priority, the hierarchical clustering strategy can be adopted, by using each vehicle property at each level in the hierarchy.

Hence, with the support of the existing general clustering algorithm an approach should be introduced where the number of clusters should not be predicted and can able to handle the dynamic situation at junctions where the vehicles should be clustered depending on the type, destination and priority of vehicles with the similarity as that of hierarchical clustering

The work of [10] has given that constructing clusters in VANET's can improve the performance for wireless communications, such as data forwarding etc. It is difficult to create stable clusters as VANET's are dynamic. The solution for constructing stable clusters by giving importance to reduce cluster head changes and increasing cluster member duration, a new mobility metric to represent N-hop mobility for vehicular networks is proposed. Based on this multi-hop clustering scheme is presented. Clustering the vehicles in VANET's is to grab the advantage of data forwarding, communication of congestion etc. As, we are performing clustering, for scheduling the vehicle through the best hop or alternate best hop, the clusters created at Grid Nodes, should be stable. Till they are not reaching the destination node, the clustering should be done at each node. By following this work the idea for creating stable clusters instead of dynamic clusters has emerged. These stable clusters will be valid only while the red light duration is active.

Vehicles which are having some similarity i.e. association and which are obeying the interrelation of some characteristics of vehicles and which are expected to behave more coherently need to be routed through the same route. Thus association scheme based grouping is expected to give better, reliable, cohesive transport system with less data inconsistencies. This work stood as the initiation for proceeding with GFTMS-Clustering.

5.2 Proposed Approach

The following is a snapshot of road network with 9 nodes. The single channel roads are considered here. The vehicles start from junction 'W' and reach the junction 'A'. At Junction 'A' according to [1], clustering of vehicles is done as a first step.

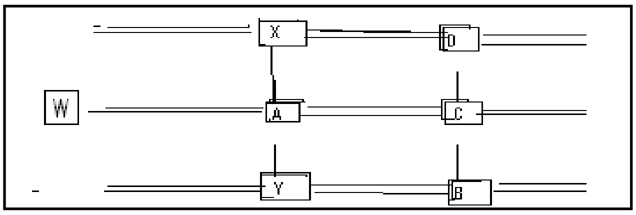

Fig 5.1 Snapshot of Road Network

Each vehicle maintains the information about Vehicle-Id, Source node, intermediate node, destination node, type of vehicle, priority of vehicle, dimension, colour of vehicle, speed range, best speed for that vehicle, etc. As soon as the vehicles reach a junction, they publish their information to the junction. The junction or node will maintain this captured information in a table known as Vehicle-Info table. Our proposed algorithm called 'GFTMS-Clustering' is invoked. The algorithm will cluster the vehicles according to the type of the vehicle, destination of the vehicle and the priority of the vehicle.

Each vehicle may be of different types such as emergency, public and normal vehicles and each may want to travel towards intended destination. Each vehicle can also set its priority randomly depending on the emergency. The priority levels considered here are very high i.e. level 5 for emergency vehicles, high i.e. level 4 for public vehicles and between medium to low i.e. 1 to 3 for normal vehicles respectively.

For any higher priority normal vehicle, the level considered is 4.

Our clustering algorithm is a hybridized type of algorithm where the new cluster is created depending upon the newly encountered criteria. As the number of clusters cannot be predefined in a traffic environment, the proposed hybrid method is designed by addressing this aspect, in contrast to k-means algorithm. It is also having the features of hierarchical because the clusters are created by taking first, type of vehicle into consideration and then further sub-clustering is done depending on destination of vehicle and then on priority.

For example, type of vehicles considered are 3, destinations of the vehicles can be any of the 2 destinations and each vehicle is having priorities ranging in between 1 to 5 (5 is for emergency, 4 is for public vehicles and 1 to 4 for normal vehicles). Then, according to our algorithm the maximum number of clusters created will be atmost 3*2*5=30 clusters. The vehicles will fall into one of these clusters. In a given snapshot instance, the clusters may be 1 or 2 or few. Hence, though upper bound is known, the actual number of clusters is not possible to predict. The number of clusters created is dynamic, it depends upon the type of vehicles, destination and priorities which we are considering in this example.

In this algorithm, the first level clusters are created on "type" attribute. This cluster set is referred as CS1. For each element of clusters of CS1 again clustering is done on "destination" attribute. Thereby, sub-clusters of clusters in CS1 are formed. This is referred as CS2. For each sub-cluster of cluster set CS2 at second level clustering, again clustering is done on "priority" attribute. Thereby, sub-clusters are again created for each element of cluster set CS2. This is referred as cluster set CS3. This algorithm can be generalized for other applications, where the attribute values can be changed accordingly and where the prior knowledge of number of clusters to be created is not known. The order for clustering i.e. type, then destination and then priority can be changed. It depends on the order of importance which we want to give to the vehicle attributes while clustering.

The following GFTMS-Clustering algorithm is used for clustering the vehicles at junctions.

```
Begin
        1.      Let RS is the Record Set of Vehicle-Info
        table at node.
        2.      First Level Clustering (let on 'type' attribute)
        2a. Initially Cluster Set CS1 is empty.
        2b. Take each record and find the group or cluster to
        which it belongs to depending on cluster attribute
        'type'.
        2c. Let Dtype is the domain of 'type' attribute.
        Dtype= {N, E, P}
        2d. call GFTMS-Clustering (CS1, RS, type)
        3.      Now, we get Cluster Set CS1={C1,C2,C3}
        4.      Second level clustering(let on 'destination'
        attribute)
        4a. initially cluster set CS2 is empty
        4b. For each element E of CS1
            4c. call GFTMS-Clustering (CS2, E, destination)
        5.      Third Level Clustering(let on 'priority'
        attribute)
        5a. Initially CS3 is empty.
        5b. For each element E of CS2
        5c. call GFTMS-Clustering (CS3, E, priority)
        End of 5b
        End of 4b
```

```
Algorithm GFTMS-Clustering (CS, RS, attr)
// CS Cluster Set initially is empty & RS is the record set of
Vehicle-Info table
   Begin
        While RS has unread tuple
          t=readCurrentTuple ()
        If CS is empty
                addNewCluster (id, t.attr)
        addTupletoCluster (id)
        else
                for each available cluster c in CS
                        similarity (t.attr, c.attr)
        end of for
        get the maximum value of similarity max-similarity
                getid of 'c' with maximum similarity
                if max-similarity >= 1
                        addTupletoCluster (c.id)
        else
                        addNewCluster (id,t. attr)
                        addTupletoCluster (id)
        end of if
        end of if
        end of while
        end GFTMS-Clustering

        similarity (p1, p2)
        Begin
        if equal (p1, p2)
        return 1
        return 0
        end similarity

        addTupletoCluster (id)
        // adds the tuple to the cluster with id
        End addTupletoCluster

        addNewCluster (id, label)
        //create new cluster with id and label
End addNewCluster
```

6. DEMONSTRATION

In this section we are going to apply the clustering strategy i.e. GFTMS-Clustering for data given in table Table 6.1.

Let us consider the following data for 12 cars which are having the same destination D

Table 6.1 Snapshot of vehicle details at Node ‘A’

Vehicle-id	Type	Destination	Priority	Time-stamp(secs)
c1	N	D	1	1
c2	N	D	4	1
c3	P	D	4	2
c4	E	D	5	2
c5	E	D	5	2
c6	N	D	2	3
c7	P	D	4	3
c8	N	D	3	4
c9	E	D	5	4
c10	N	D	1	4
c11	N	D	4	5
c12	N	D	2	5

In this section we are going to apply the clustering strategy i.e. GFTMS-Clustering for data given in table Table 6.1.
The domain of type considered here is Dtype= {N, P, E}, where N stands for Normal vehicles, P stands for Public vehicles and E stands for Emergency vehicles. In this only one Destination is considered i.e. D. The valid priorities considered for normal, public and emergency vehicles are 1 or 2 or 3 or 4, 4 and 5 respectively.

For our data the number of clusters created by using the vehicle properties (roles) specified is shown in Table 6.2. This table is used for demonstrating the clustering strategy.

According to Table 6.2., we get 6 clusters, therefore we should have 6 routes to route the vehicles. But, the number of alternate less congested available routes will not be equal to number of clusters created.

Table 6.2 Vehicle Properties used for clustering

Vehicle Properties(VP)	Vehicles	Cluster-number
r1={Type=N,Dest=D,Priority=1}	{c1,c10}	TCL1
r2={Type=N,Dest=D,Priority=2}	{c6,c12}	TCL2
r3={Type=N,Dest=D,Priority=3}	{c8}	TCL3
r4={Type=N,Dest=D,Priority=4}	{c2,c11}	TCL4
r5={Type=P,Dest=D,Priority=4}	{c3,c7}	TCL5
r6={Type=E,Dest=D,Priority=5}	{c4,c5,c9}	TCL6

In our example according to Figure 5.1, graph of road network, we are having only 3 routes to D from A. Our clustering algorithm will be triggered at node A. Hence, there is a requirement to again agglomerate the clusters depending on the shared vehicle properties framed.

In our example we are using following policies.

Table 6.3 Shared Vehicle Properties

Shared Vehicle Properties(SVP)	Type	Destination	Priority	Vehicle properties shared by SVP (Set of VPs to agglomerate the clusters)
P1	N	D	1 or 2 or 3	{r1,r2,r3}
P2	N or P	D	4	(r4,r5}
P3	E	D	5	{r6}

Now, we will apply the following clustering strategy on the data present in Table 6.1., by considering the Vehicle properties (VPs) considered in Table 6.2., and by using the Shared Vehicle Properties given in Table 6.3.

6.1 Demonstration Using GFTMS Clustering

Step1

By applying GFTMS clustering, initially 1st level clustering is done on the attribute “type”. The following cluster set CS1 is created. In this each cluster grows and also the new clusters are created simultaneously.
The format used to represent the clusters is (Label, {objects})
After 1st level clustering
CS1={(N,{c1,c2,c6,c8,c10,c11,c12}),(P,{c3,c7}),
(E,{c4,c5,c9})}
Therefore CS1= {CL1, CL2, CL3}

Step2

On each element of resultant cluster set CS1 again apply 2nd level clustering on attribute destination is applied. As, destination considered here is single i.e. D, the sub-clusters for each element of CS1 will not be created. For every element of CS1 only one sub-cluster will be created.
Therefore CS2 is
CS2= {SCL1, SCL2, SCL3}

Step3

Third level clustering is performed using priority attribute on each element of CS2 cluster set created in 2nd level clustering. As, there are 3 elements in CS2, 3 calls are done
GFTMS_Clustering (CS3, SCL1, priority)
CS3={(1,{c1,10}),(2,{c6,c12}),(3,{c8}),(4,{c2,c11})}
Therefore CS3= {TCL1, TCL2, TCL3, TCL4}
GFTMS_Clustering (CS3, SCL2, priority)
CS3=CS3 + (4, {c3, c7})
CS3= {TCL1, TCL2, TCL3, TCL4, TCL5}
GFTMS_Clustering (CS3, SCL3, priority)
CS3=CS3 + {(5, {c4, c5, c9})
CS3= {TCL1, TCL2, TCL3, TCL4, TCL5, TCL6}

Depending upon the timestamp of the arrival of the vehicles given in Table 6.1 the Simultaneous and Dynamic cluster creation is as follows.

Table 6.4 Temporal Simultaneous and Dynamic Clustering

Time Stamp	Vehicle	1st Level(type)	2nd Level(destination)	3rd Level(priority)
T1	c1 c2	{N,{c1,c2}}		
T2	c3 c4 c5	{N,{c1,c2}} {P,{c3}} {E,{c4+c5}}	{D,{c1,c2}}	
T3	c6 c7	{N,{c1,c2+c6}} {P,{c3+c7}} {E,{c4,c5}}	{D,{c1,c2}} {D,{c3}} {D,{c4,c5}}	{1,{c1}} (TCL1) {4,{c2}} (TCL4)
T4	c8 c9 c10	{N,{c1,c2,c6+c8,c10}} {P,{c3,c7}} {E,{c4,c5+c9}}	{D,{c1,c2+c6}} {D,{c3+c7}} {D,{c4,c5}}	{1,{c1}} (TCL1) {4,{c2}} (TCL4) {4,{c3}} (TCL5) {5,{c4,c5}} (TCL6)
T5	c11 c12	{N,{c1,c2,c6,c8,c10+c11,c12}} {P,{c3,c7}} {E,{c4,c5,c9}}	{D,{c1,c2,c6+c8,c10}} {D,{c3,c7}} {D,{c4,c5+c9}}	{1,{c1}} (TCL1) {4,{c2}} (TCL4) {2,{c6}} (TCL2) {4,{c3+c7}} (TCL5) {5,{c4,c5}} (TCL6)
T6		{N,{c1,c2,c6,c8,c10,c11,c12}} {P,{c3,c7}} {E,{c4,c5,c9}}	{D,{c1,c2,c6,c8,c10+c11,c12}} {D,{c3,c7}} {D,{c4,c5,c9}}	{1,{c1,c10}} (TCL1) {4,{c2}} (TCL4) {2,{c6}} (TCL2) {3,{c8}} (TCL3) {4,{c3,c7}} (TCL5) {5,{c4,c5,c9}} (TCL6)
T7		{N,{c1,c2,c6,c8,c10,c11,c12}} {P,{c3,c7}} {E,{c4,c5,c9}}	{D,{c1,c2,c6,c8,c10,c11,c12}} {D,{c3,c7}} {D,{c4,c5,c9}}	{1,{c1,c10}} (TCL1) {4,{c2,c11}} (TCL4) {2,{c6,c12}} (TCL2) {3,{c8}} (TCL3) {4,{c3,c7}} (TCL5) {5,{c4,c5,c9}}(TCL6)

Depending upon the red light clearance duration, the partial clusters will be created and they can be scheduled immediately instead of waiting for complete clusters creation. The complete clusters created due to this are shown in Figure 6.4.

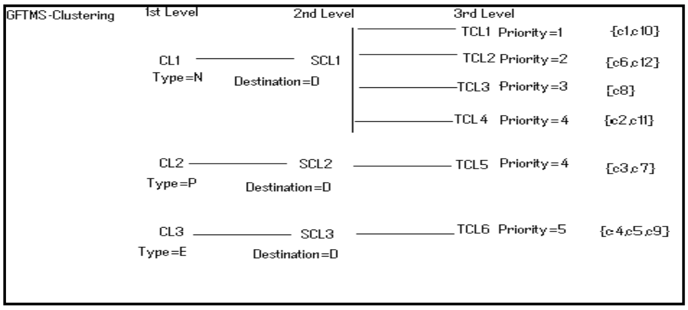

Fig 6.1 GFTMS-Clustering

6.2 Observation

The different types of vehicles with different priorities may arrive simultaneously. Therefore, the need arises to create the different clusters simultaneously as well as to grow the existing cluster. In snow ball technique clusters cannot be created simultaneously. Hence GFTMS-clustering stands useful.

In this system, decision should be on fly i.e. when clusters are ready, we need not wait till end. We can trigger the scheduling i.e. dispatching as a second step of [1] for the existing clusters.

If there is a situation where more number of vehicles is getting accumulated at node then the scheduling need not be postponed, it can be applied for the existing clusters which are ready to get dispatch. When the scheduling event is triggered, the organized clusters or partial clusters should be scheduled and parallelly cluster growing strategy also be operative. Thus it provides real time multi-tasking traffic scheduling strategy. Let the red light clearance duration is 4 secs, then the TCL1,TCL4,TCL5,TCL6 are the partial clusters created. The vehicles of these partial clusters can be scheduled. It the red light clearance is 7 secs then the complete clusters scheduling can be possible. Therefore, there should be consistency between the red light clearance time and the clusters completion time. If the red light time is short then partial clusters creation is more. Simultaneous dynamic cluster creation as shown in the timestamp demonstration is simulated using multithreading.

Hence, GFTMS-Clustering stand useful when there is a road network of dynamic nature where different types of vehicles arrive at different time stamps and when scheduling or dispatching decision should be done on fly.

As, the number of clusters created in each technique is not equal to the number of available routes in the road network, there is a necessity to agglomerate the clusters. Merge the clusters using the SVPs specified in table 6.3. Therefore, agglomerate TCL1, TCL2, TCL3 of GFTMS clustering to MC1, TCL4, and TCL5 to MC2 and TCL6 to MC3.
MC1= {TCL1, TCL2, TCL3}
MC2= {TCL4, TCL5}
MC3= {TCL6}

Each agglomerated cluster can be scheduled to each available route by taking the different traffic parameters into consideration such as density, capacity of the road, congestion level etc. The scheduling should be done by dispatching the highest priority cluster through less congested route.

7. SIMULATION RESULTS

Java, Jdk1.6. with the concepts of applets and animation using multithreading is utilized for the development of simulation program. This simulation program has the following phases.
1. Road network design phase, in which the road networks is designed with a nodes or junctions. Figure 5.1 is the Road network graph designed with this simulation program.
2. Generation of vehicles phase will generate the vehicles at W and they travel towards junction A. The vehicles are clustered at A and the scheduling is done at A as a second step of [1]. For the cluster having public vehicles the scheduling is not done and they have to reach the destination with the pre-specified route here it is ACD. And in our simulation the 3 types of clusters are concentrated. They are 1.Clusters with normal vehicles with priority 1, 2, and 3 with destination D as MC1. 2. Clusters with all public vehicles and normal vehicles with priority 4 with destination D as MC2. 3. Clusters with emergency vehicles with destination D and with priority 5 are merged as MC3. Here only clusters with only destination D are considered and simulation results are shown for the clusters with destination D.

For MC1 cluster the path with long distance is scheduled i.e. AYBCD. Assumption: long distance, high congestion and less speed.
For MC2 cluster, ACD is scheduled. Assumption: normal distance, medium congestion and average speed.
For MC3 cluster, AXD is scheduled. Assumption: short distance, less congestion and hence high speed.

The simulation program is run without using our approach where the vehicles are selecting the normal frequently used route. Hence, it is taken as ACD. All the vehicles immaterial of type and priority with destination 'D' will use this route. The maximum time taken to reach the destination is calculated for different speeds i.e. 20, 40, 60, 80,100 and 120km/hr. Here the max-time is the time taken by the last vehicle to reach the destination. The number of vehicles generated at W in this simulation program is 100.

The following graph shows the max-time taken vs. speed graph without using proposed approach.

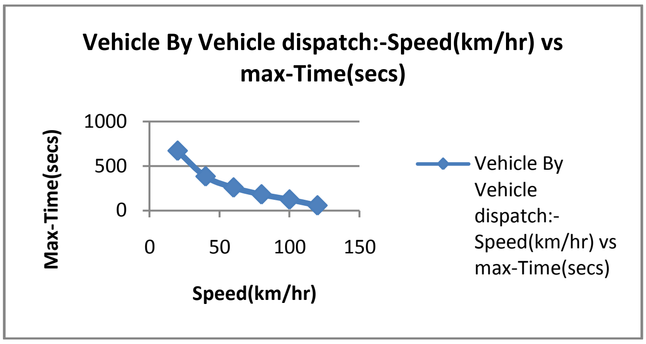

Fig 7.1 Vehicle by Vehicle dispatch without using GFTMS-Clustering

Now again the simulation program is run by invoking our proposed algorithm at junction A and to reduce the complexity it has further grouped the clusters into 3 main clusters i.e. MC1, MC2 and MC3 as mentioned earlier. The simulation program is run for 100 vehicles and scheduled the 3 clusters according to the path AYBCD, ACD and AXD for different speeds 20,40,60,80,100 and 120km/hr and the max-time taken is calculated. Here max-time taken is the time taken by the last vehicle of the last cluster which reached the destination. The following is the max-time taken vs. speed graph for the same.

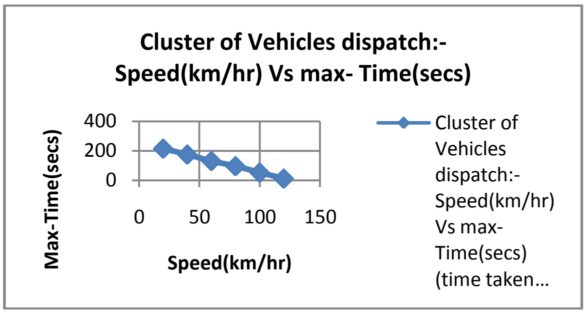

Fig 7.2 Cluster of vehicles dispatch using GFTMS-Clustering

It is observed that when clusters use different paths, the time taken for the maximum vehicles reaching the intended destination is reduced.

The comparison of results with and without using proposed approach is as follows.

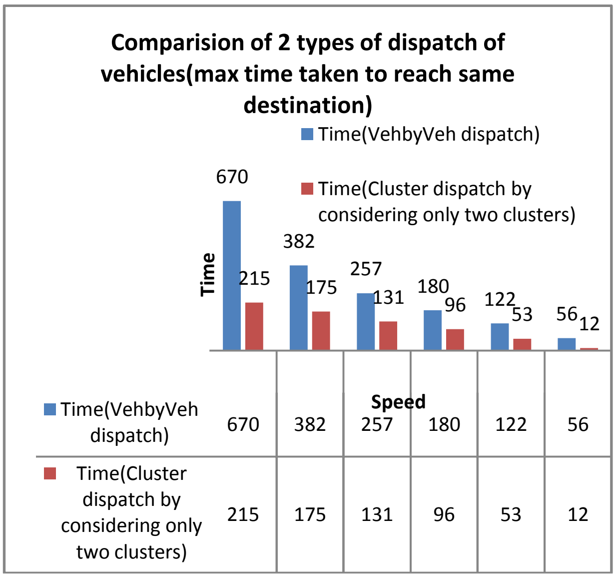

Fig 7.3 Comparison of results without and with using GFTMS-Clustering

It is observed that the max-time taken by the proposed approach has reduced considerably.

Our simulation program clusters the vehicles at junction A and schedules the clusters through different routes. Using this, the highest priority clusters are sent through the route with less congestion. The vehicles travelling through the less congested routes can accelerate their speed and can reach the destination soon compared to the vehicles travelling through the normal or highly congested routes. While demonstrating this the time taken by different clusters following different routes to reach a destination is calculated at different speeds i.e. 20,40,60,80,100 and 120km/hr.

The following graph shows the time taken vs. speed for different clusters. It can be observed that C3 cluster of vehicles takes less time to reach compared to C2 and C1.

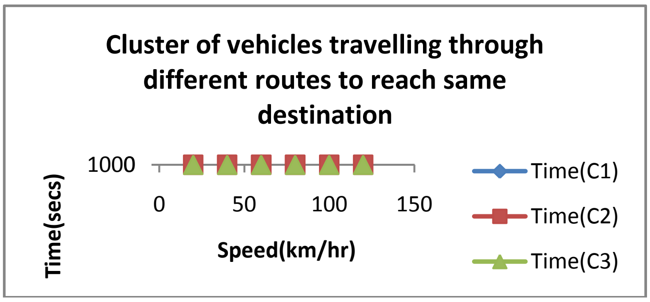

Fig 7.4 Results showing time taken versus speed for different clusters

8. CONCLUSIONS

In this paper we have proposed a novel solution for clustering the vehicles at junctions. In this the clustering is done at a junction. The proposed algorithm has reduced the maximum time taken to reach the destination. It is observed that the performance has been improved in terms of maximum time taken at different speed when compared without using our approach.

At junction, different criteria such as distance, congestion in that route can be taken into consideration and after clustering, different clusters except public vehicles can be scheduled to different routes depending on type, destination and priority. Clusters having public vehicles are not scheduled, as public vehicles have to follow the predefined route and we cannot alter the predefined routes.

In this paper as a first step of [1] clustering procedure is mentioned. The different issues should be considered for scheduling the vehicles which is reserved for the future work. In this congestion, distance through different routes are assumed while scheduling the vehicles. The solution in this paper is an example for alleviating the traffic jams and for avoiding the traffic jams by dissipating vehicles after clustering them

REFERENCES

[1] AvulaAnitha, Rajeev Wankar and C. RaghavendraRao, “A Grid Fabrication of Traffic Maintenance System”, International Conference on Advances in Computer Science, Engineering and Applications, Advances in Intelligent and Soft Computing, SpringerLink, Volume 167/2012, 999-1008, 978-3-642-30111-7_96, 2012.

[2] Feilong Tang, MinyiGuo, Minglu li and Cho-Li Wang, “Implementation of an Intelligent Urban Traffic Management System Based on a City Grid Infrastructure”, Journal of Information Science and Engineering 24, 1821-1836, 2008.

[3] Subhhamita Mukherjee, Indraji Pan and KashiNathDey, “Traffic Organization by utilization of resources through Grid Computing Concept,”, World Congress on Nature & Biologically Inspired Computing (NaBIC 2009) [IEEE], 2009.

[4] AbderrahmaneLakas, MoumenaCheqfah, “Detectionn and Dissipation of Road Traffic Congestion Using Vehicular Communication”, Microwave Symposium, Mediterranrean, IEEE Communications, 978-1-4244-4665-0/09, 2009.

[5] HisakKuriyama, Yoshihiro Murata, Naoki Shibata Keiichi Yaumoto etzal, “Congestion Alleviationn Scheduling Technique for Car Drivers Based on Prediction of Future Congestion on Roads and Spots”, Intelligent Transportation Systems, IEEE, ISBN-978-1-4244-1396-6, 2007.

[6] Rambabu Yerajana, A.K. Sarje, “An Adaptive Multipath Souurce Routing Protocol for Congestion Control and Load Balancing in MANET”, International Conference on Advances in Computing, Communication and Control (ICAC3), ACM, ISBN 978-1-160558-351-8, 2009, pp.456-459.

[7] Yonghua Zhou, “Web Services-based Grid Computing for Traffic Flow Predictive Control”, Global Congress on Intelligent Systems, IEEE, 978-7695-3571-5, 2009.

[8] Kunszt P.Z. and Guy L.P., “The open grid services architecture and data grids. In Grid Computing Making the Global Infrastructure a Reality”, Berman F, fox GC, Hey AJG (eds). Wiley series in Communications, Networking and Distributed Systems, Wiley:Chichester, 2003, pp 9-50.

[9] Cameron D.G., Carvajal-Schiaffino R, Millar A.P., Nicholsan C, Stockinger K., and Zini F., “Evaluating scheduling and replica optimization strategies”, Proceedings of the Fourth International Workshops on Grid Computing (GRID’03), 2003, pp-65-77.

[10] Zhenxia Zhang, AzzedineBoukerche and Richard W. Pazzi, “A Novel Multi-hop Clustering Scheme for Vehicular Ad-Hoc Networks”, MobiWac., ACM, 978-1-4503-0901-1/11/10,2011.

[11] P.A.Vijaya, M. Narasimha Murthy, D.K. Subramanian, “ Leaders-SubLeaders An Efficient Hierarchical Clustering Algorithm for Large Data Sets, Pattern Recognition Letters, Elsevier, vol 25,pp 505-513, 2004.

[12] Osama Abu Abbas, “Comparisons between Data Clustering Algorithms” International Arab Journal of Information Technology, Volume 5, No. 3, July 2008.

[13] Marina Meila, David Heckerman, “An Experimental Comparison of several clustering and initialization methods”, Technical Report, MSR-TR-98-06, February, 1998.