International Journal of Innovative Computing, Information and Control
Volume **8**, Number **10(A)**, October **2012** ISSN 1349-4198
pp. 6811–6836

A LOCATION BASED PREDICTION SERVICE PROTOCOL FOR VANET CITY ENVIRONMENT

Salim M. Zaki, M. A. Ngadi, Maznah Kamat, Shukor Abd Razak and Johan Mohamad Shariff

Faculty of Computer Science and Information Systems
Universiti Teknologi Malaysia
Johor 81310, Malaysia
salimzki@gmail.com; { dr.asri; kmaznah; shukorar; johan }@utm.my

Received July 2011; revised November 2011

Abstract. *Location-based service is used in Vehicular Ad hoc Networks (VANETs) to locate node's position before the start of any communication. The existing location services proposed for VANET do not distribute the load on multi servers, and some of them do not consider stable nodes for selecting location servers. Additionally, predicting node's locations using grey prediction model accuracy is affected by nodes acceleration in VANET. This paper proposes a Vehicular Quorum Prediction-based Location Service protocol (VQPLS), which was designed for urban area topology and which utilised node's information such as distance to intersection centre point and speed in selecting stable location servers. Formation of quorum of location servers was done by main location server by nominating some other nodes located at the intersection based on their moving directions. The quorum of location servers was used to distribute the load on multi servers and for fault tolerance. A hybrid of Grey Prediction Model and Alpha-Beta-Gamma Filter (GP-ABGF) was adopted in the VQPLS to accurately predict the next position of nodes that were moving away from the intersection. The GP-ABGF mitigated effects of VANET nodes acceleration and irregular update time intervals on grey prediction model accuracy. Prediction algorithm also filtered the noise data and produced accurate location of destination and overcame the problem of outdated locations stored by quorum systems. The VQPLS showed good performance compared with other protocols in reducing overhead of control packets and high delivery of packets to destinations, and it reduced the end-to-end delay of routing packets.*
Keywords: Location service protocol, Vehicular ad hoc network, Grey prediction model, Location prediction

1. **Introduction.** Vehicular Ad hoc Networks (VANETs) integrate the capabilities of wireless networks to transportation systems. This may offer mobile users connectivity wherever they are and ability to exchange useful information among the mobile users. Hence, VANET creates efficient communication between cars (car to car). Applications of VANET are wide with various services.

Most of VANET's services and applications rely on the locations of the source and the destination of a message. For forwarding a message from source to destination, the position-based routing protocols are suitable for VANET environment [1]. This class of protocols needs, as a prerequisite, a location service that could find the location of a given ID of a destination in order for source to communicate with a destination using position-based routing [2]. Many location service protocols have been proposed. They differ in term of strategies used in selecting location servers and schemes deployed in updating locations. Majority of these protocols were designed for Mobile Ad Hoc Network (MANET) and sensor networks environments, for example, Grid Location Service (GLS)

[3], Hierarchical Location Service (HLS) [4], and quorum-based location service [5]. In general, a location service protocol for VANET should provide accurate location of the destinations, low control overhead, and minimum end-to-end delay.

The high mobility of VANET's nodes causes nodes move far distance from each other in short time which causes storing outdated locations on location servers which influence the efficiency of location service. Furthermore, quorum-based location service is well known to store some outdated information. This outdated location information degrades delivery of packet to their destinations. Frequent beaconing could keep nodes locations up-to-date, but it interferes with other data transmission and causes packets collisions, and retransmissions in addition to network congestion [6]. Nodes moving in a very high speed may cause frequent link breakage, which affects the performance of the network [7, 8]. Moreover, the delivery success ratio is affected by link lost [9]. In VANET, the high mobility of nodes influences the effectiveness of the conventional protocols and may degrade the performance of the network [10]. Therefore, a prediction could be a good solution here to overcome the problem of outdated locations and reduce control overhead in high mobility environment.

Predicting future locations of mobile objects has received a lot attention since a few decades ago. A mobile object could be a car, mobile robot, or aircraft that moves away from its neighbour nodes within wireless detecting range. When a mobile object stops sending its status to tracker for a period of time, the tracker will lose the track of that object. In object tracking, the information about the moving object's latest received position and speed is important. However, prediction models used in predicting locations of wireless mobile nodes are affected by factors such as noise input and acceleration. Hence, filter for the noise is needed.

Predicting future locations of mobile nodes could be a good solution in the case of high mobility environment such as VANET. Grey prediction model is widely used to efficiently predict future locations of mobile nodes in ad hoc environment and other applications [11, 12]. However, the accuracy of the model is affected by some factors such as irregular intervals of nodes location updates and variable speed of mobile objects. These factors corrupt the predicted value by the grey prediction model. In this paper, a prediction algorithm is proposed to overcome those problems. Using hybrid grey prediction model with noise filter, the algorithm predicts accurate locations of mobile nodes that move in high speeds. The integration between prediction algorithm and quorum-based location service protocol provides node with accurate locations of destinations in VANET urban environment.

For illustration purpose, an example is given here to show the effects of high mobility environment on grey prediction model. In urban areas, a node accelerates and decelerates based on traffic conditions, such as whether there is a traffic congestion or not, and based on road topology such as approaching an intersection. However, a node has to speed up in some open roads, which then increases its current speed. This makes distance moved by this node changes according to speed acceleration. It is well-acknowledged that constant speed means fixed moving distance over time. However, with acceleration, this distance will vary. A node sends location update packets towards its corresponding location server invoked over regular time intervals. However, when distance-based update scheme is used in location service protocol, this interval changes and the location packets sent reaches the location server irregularly. As a result, acceleration and irregular time intervals of sampled locations influence the accuracy of grey prediction model. For this reason, quasi-smoothness is used to check noise on grey prediction model and to eliminate the noise with Alpha-Beta-Gamma Filter. The proposed prediction algorithm is applicable in predicting the future locations of moving objects such as mobile nodes in ad hoc networks or mobile robots.

This paper proposes a new location service protocol called VQPLS designed especially for urban environment to avoid problems represented in acceleration and irregular time intervals updates. VQPLS also distributes the load on multi servers. The Vehicular Quorum Prediction-based Location Service protocol (VQPLS) uses two parameters: distance of a vehicle from centre point of an intersection, and its speed during the selection of the main location server. The construction of quorum group ensures the load distribution around dense intersection, which can minimise the load of packets on main location server. Furthermore, a prediction algorithm is proposed in this paper, and this algorithm predicts accurate locations of mobile nodes move with high mobility in urban area.

2. **Related Work and Problem Background.**

2.1. **Location service protocols for VANET.**

Many location service protocols have been proposed. They can be categorised based on the strategy used in selecting nodes as the location servers or the scheme employed for updating node's location. Furthermore, most of the protocols were designed for Mobile Ad hoc Network (MANET) environments [3, 4, 5]. Designing a location service protocol for urban environment in VANET is a challenging task and it requires careful design in order for it to perform well in dense area and high mobility environment, whereby the load on some nodes such as location servers need to be distributed efficiently for better protocol performance. Selecting a proper node as a location server may increase packet delivery ratio by minimising delay.

There are some limitations and challenges faced by the location service protocols that were initially designed for MANET, specifically when they are applied in VANET urban environment [13]. The early designs of these protocols considered open area and random node's mobility. The nature of nodes on urban areas such as junctions, roundabout, and corners forces nodes to move based on restricted mobility models and causes frequent topology change [14]. The high speed of VANET nodes mobility also causes MANET protocols to be unsuitable for VANET environment [15]. All these issues influence the performance of location service protocols in determining destinations locations accurately with low overhead. Proposed location service protocols for VANET environment such as Intersection Location Service (ILS) [16], Vehicle Location Service (VLS) [17] and Hierarchical Location Service with Road-adapted Grids for Vehicular Networks (HLSRG) [18] do not consider node's stability in the designed protocol and do not distribute load on multi severs. ILS and Responsible Section Location Service (RSLS) [19] protocols use intersection to be the place for some control nodes such as location servers or main forwarders, but they do not take into account the high nodes density on intersections that can cause load imbalance. In addition, the parameter usually used for selecting location servers is either distance of node to a specific location or to intersection as in RSLS and Map-Based Location Service (MBLS) [20], or simply based on node's ID such as in ILS. This however does not ensure the best stable node selection. The high mobility of VANETs nodes considers stability as an important factor in selecting responsible nodes as location servers. Design of RSLS protocol faces the same problem as ILS. The effects of control packet overhead and load balance are not examined in both ILS and RSLS. MBLS uses specified points to select location servers. It selects location servers based on waypoint, which cannot guarantee a good location in selecting a location server.

In this paper, a quorum-based location service protocol called Vehicular Quorum Prediction-based Location Service protocol (VQPLS) is proposed. The protocol takes the advantages of quorum system in replicating location information on many nodes to avoid location server failure. VQPLS is designed for an urban area topology and it utilises intersection to be the location of main node of the quorum. In addition, it exploits the speed and position of nodes inside intersection vicinity as criteria for best location server selection. The distance to centre point of intersection and node speed are set based on realistic data. This ensures most stable node is selected as main location server, which is responsible for the construction of the quorum of location servers. The Main Location Server (MLS) is first selected based on its lowest speed compared with neighbour nodes and its position should be within pre-specified distance to centre point of intersection. Then, MLS constructs a quorum by selecting number of nodes called Passing Location Servers (PLSs). The PLSs are nodes passing the intersection, selected based on their directions to act as location servers for the road segment branch out of the intersection. MLS selects one PLS for each direction at a time, and once the number of PLSs reaches below a threshold due to failure or is out of range, MLS reselects new PLSs. This design guarantees load distribution on dense intersection, and reduces packet overhead. Packet overhead occurs due to high number of reply packets handled by PLSs that surround the intersection. Additionally, PLS reduces number of hops a query may take to reach MLS.

2.2. **Location prediction with grey model.** Grey prediction model of grey theory called GM(1,1) is a quantitative prediction model. It is based on some theoretical treatment of the original data and establishment of grey models to discover and control the development rules of the system of interest so that scientific quantitative predictions about the future of the system can be made [21]. The grey action quantity in GM(1,1) is a value derived from the background values. It reflects changes contained in the data and its exact intention is grey. The grey action quantity realises the extension of the relevant intention. The existence of this grey action quantity distinguishes grey systems modelling from the general input-output modelling (or black box modelling), and is a test stone to separate the thoughts of grey systems and that of grey boxes.

Grey theory is useful for modelling systems with unidentified or a small number of sampling data. Hence, researchers used GM(1,1) to predict locations of moving objects as in [11], who proposed a location prediction model based on GM(1,1). GM(1,1) is also used to predict the motion of some random moving objects in image processing; an example is the usage of this method for detecting hand motions and tracking objects [22]. Many applications of GM(1,1) in various fields are discussed in [23]. This prediction model showed high accuracy in predicting positions but it needs at least four previous locations with their received timestamps to calculate the new position. Unlike linear functions, GM(1,1) uses historical data to predict the future locations. GM(1,1) then manipulates those data through mathematical equations and predicts the future location.

This model reduces the randomness of data through Accumulated Generation Operation (AGO). AGO reduces noise in the input series of data, starts a number of calculations, and ends with a predicted value. The time-series that the grey model depends on could be affected by some factors, which then lead to noisy predicted values. These factors are irregular location update and acceleration. Irregular location update of moving objects leads to irregular time-series data. Second factor is acceleration, which reflects the movement of mobile objects in reality that accelerate their speeds and this acceleration affects the accuracy of predicting locations using GM(1,1).

Grey prediction model is used in literature for tracking mobile nodes location. Wei et al. [24] proposed a location-based prediction protocol with three prediction algorithms. First, it uses GM(1,1) to predict a node's location in sensor network environment, and Kalman filter as the second algorithm. A combination of the two prediction technologies is proposed to avoid location error of using one prediction algorithm. However, some weight values were used to support hybrid between grey model and Kalman filter. Synchronisation is assumed between data saved on both sensor node and the sink. This synchronisation is for synchronising both nodes in order to produce very close or similar predicted location. Hybrid Grey model and Kalman Filter Data Aggregation (CoGKDA) does not take the acceleration of mobile node into account, which then affects the accuracy of grey model prediction. CoGKDA also does not study the effects of irregular time intervals of sending updates on time-series prediction model.

Wang and Liu [25] used grey prediction model in vehicular sensor network (VSN). It utilised GM(1,1) to predict step of tracking mobile node and in sending alert messages. The authors showed that grey model decreased the required time for prediction compared with linear prediction methods. The received signal power (four values) from cluster head node will be treated as input to the grey model. Some ranges of threshold have been set and used to trigger grey model. Whenever a mobile destination node is out of source node's range, it will generate a warning message to source node. Destination determines the location of a forwarder node that lies in range of source. This scheme does not study the effects of vehicle acceleration and irregular time intervals of collected data on grey model. Moreover, it has higher control packets overhead and it causes latency in packet transmission.

Qiang et al. [26] has compared grey prediction model GM(1,1) with Kalman filter [27] for efficiency in tracking moving object. The authors used a very simple algorithm construction related to extracting historical data about moving object. The latest 5 locations saved about the mobile object were retrieved and used in grey model to predict the future steps. With the use of fresh historical data, the grey model showed relatively higher accuracy compared with Kalman filter. Additionally, the actual algorithm assumed that the object moves in a constant speed. This however does not represent the actual road environment.

Target tracking filter such as Alpha-Beta filter predicts the next position of a moving object but does not take into account the acceleration of the object. An extended version called Alpha-Beta-Gamma filter [28] was proposed. The filter considered acceleration in predicting future location of accelerated objects [29]. The three orders Alpha, Beta, and Gamma refer to the parameters used in prediction equations. These parameters need to be within optimal range of values as determined in [30]. This filter can also remove noise and predict more accurate future positions of mobile objects as in [31] than Alpha-Beta filter.

This paper proposes a hybrid algorithm that combines both GM(1,1) and $\alpha - \beta - \gamma$ filter in reducing noise produced by grey model when sampled historical data are collected over irregular time intervals. The algorithm reduces the effects of acceleration of nodes on grey prediction model accuracy. The algorithm utilises the quasi-smoothness test of grey model to check error of the prediction model. Once the detected error exceeds a threshold, the algorithm will filter the predicted data with $\alpha - \beta - \gamma$ filter.

3. Preliminaries.

3.1. Grey model concept.

Grey Model First Order One Variable GM(1,1) of grey systems theory is applicable in various applications [32]. The theory has been studied widely in the literature. Based on the time series data that represent the historical data, GM(1,1) can predict the future values of these data using some differential equations. It is efficient and able to make accurate prediction with a small number of historical data.

The input to GM(1,1) should be non-negative data. In this paper, the location of nodes $X^{(0)}$ are input to GM(1,1).

$$X^{(0)} = \left(x^{(0)}(1), x^{(0)}(2), \ldots, x^{(0)}(n)\right) \tag{1}$$

where $n \geq 4$.

By Accumulated Generating Operation (1-AGO), the new sequence of primitive data can be obtained as follows:

$$X^{(1)} = \left(x^{(1)}(1), x^{(1)}(2), \ldots, x^{(1)}(n)\right) \tag{2}$$

where

$$x^{(1)}(k) = \sum_{i=1}^{k} x^{0}(i), \quad k = 1, 2, 3, \ldots, n \tag{3}$$

The mean generated sequence of $X^{(1)}$ neighbours is specified by:

$$Z^{(1)} = \left(z^{(1)}(1), z^{(1)}(2), \ldots, z^{(1)}(n)\right) \tag{4}$$

where

$$z^{(1)}(k) = 0.5x^{(1)}(k) + 0.5x^{(1)}(k-1), \quad k = 2, 3, \ldots, n \tag{5}$$

Hence, the differential equation of grey system is

$$x^{(0)}(k) + az^{(1)}(k) = b. \tag{6}$$

The equation used for whitening is

$$\frac{dx^{(1)}}{dt} + ax^{(1)} = b \tag{7}$$

Equation (7) represents GM(1,1), which is called first order grey differential equation. The first order derivative of AGO of X data is represented in first 1 of GM(1,1). Meanwhile, the second 1 stands for 1 series considered with deferential equations of grey model.

The first order differential equation has three main parts; derivative $\frac{dx}{dt}$, background value $x^{(1)}$, and finally a and b parameters.

The a and b parameters $[a, b]^T$ are generated by:

$$[a, b]^T = (B^T B)^{-1} B^T Y \tag{8}$$

where

$$Y = \begin{pmatrix} x^{(0)}(2) \\ x^{(0)}(3) \\ x^{(0)}(4) \\ .. \\ .. \\ x^{(0)}(n) \end{pmatrix} \tag{9}$$

$$B = \begin{pmatrix} -z^{(1)}(2) & 1 \\ -z^{(1)}(3) & 1 \\ -z^{(1)}(4) & 1 \\ . & . \\ . & . \\ -z^{(1)}(n) & 1 \end{pmatrix} \tag{10}$$

and the time response sequence:

$$\widehat{x}^{(1)}(k+1) = \left[x^{(0)}(1) - \frac{b}{a}\right] e^{-ak} + \frac{b}{a} \tag{11}$$

Finally, the predicted value is obtained by:

$$\widehat{x}^{(0)}(k+1) = \widehat{x}^{(1)}(k+1) - \widehat{x}^{(1)}(k) = (1-e^{-a})\left[z^{(0)}(1) - \frac{b}{a}\right] e^{-ak} \tag{12}$$

3.2. **Quasi-smoothness check in GM(1,1).** One of the smart features of the grey model for prediction is the quasi-smoothness check [21]. It detects whether or not the prediction model will forecast accurate values. This is done by using the sequence data input to the model and the data generated by AGO, and by checking the results against a given threshold. If the result is equal to, or below the threshold, the prediction is accurate. Otherwise, noisy predicted data are produced.

Firstly, the model performs the quasi-smoothness check on $X^{(0)}$ data sequence using the following equation:

$$p(k) = \frac{x^{(0)}(k)}{x^{(1)}(k-1)} \tag{13}$$

This check is applied a number of times to validate the results; if some of the obtained results are < 0.5, then it satisfies the condition of quasi-smoothness. The second check is quasi-exponentiality using $X^{(1)}$:

$$\sigma^{(1)}(k) = \frac{x^{(1)}(k)}{x^{(1)}(k-1)} \tag{14}$$

If the results of this check are within the threshold [1, 1.5] range, then its quasi-exponentiality is satisfied.

Satisfying the above quasi check is important in establishing the GM(1,1) prediction model, and in checking whether results are accurate or not. This research uses this feature for testing the accuracy of GM(1,1). Based on the results obtained from the checking, the proposed algorithm decides whether to filter the data or not.

3.3. **Alpha-Beta-Gamma filter.** Many proposed target tracking filters such as Alpha-Beta filter predict the next position of a moving object without taking into account the acceleration of the object. The Alpha-Beta-Gamma filter considers acceleration and has ability to predict the next position and next speed of objects [29]. The three orders Alpha, Beta and Gamma refer to the parameters used in prediction equations. These parameters need to be within optimal range of values as determined in [30]. This filter can also filter noise data and predict accurate values for future positions of mobile objects.

Equation (16) is used for predicting next position:

$$x_p(k+1) = x_s(k) + T.v_s(k) + \frac{1}{2}.T^2.a_s(k) \tag{15}$$

Next speed of the object is predicted using Equation (17) below:

$$v_p(k+1) = v_s(k) + T.a_s(k) \tag{16}$$

The smoothed parameters, or also known as innovations, are calculated for position $x_s(k)$, speed $v_s(k)$ and acceleration $a_s(k)$:

$$x_s(k) = x_p(k) = \alpha.(x_0(k) - x_p(k)) \tag{17}$$

$$v_s(k) = v_p(k) + \frac{\beta}{T}.(x_0(k) - x_p(k)) \tag{18}$$

$$a_s(k) = a_p(k-1) + \frac{\gamma}{T^2}.(x_0(k) - x_p(k)) \tag{19}$$

The ranges of the three order parameters are modifiable within their ranges until accurate results are obtained as determined in [33] as follows:

$$0 < \alpha < 2$$
$$0 < \beta < 4 - 2\alpha$$
$$0 < \gamma < \frac{4\alpha\beta}{2 - \alpha}$$

4. **Proposed Location Service Protocol.** This section explains the design of Vehicular Quorum Prediction-based Location Service protocol (VQPLS), which considers urban environment topology, where selection of location servers is done near intersection centre point of roads. Centre point is defined as the intersection of roads. A node can be an eligible location server if its position is within intersection vicinity, moving at low speed or stopped, and closest to centre point. Main location server is selected among nodes at the intersection because it is the position where nodes usually stop and this will provide high probability of a stable node. The trade-off between node's distance to intersection centre point and slow speed values is practical for selecting most stable node. A stable node is the node that stays longer at intersection and close to centre that makes the node to be in range of most of other nodes near that vicinity. This stability is to reduce control overhead, thus increasing the performance of end-to-end communications. It is assumed that each node is provided with 802.11 communication abilities in order to communicate with each other. Additionally, each node knows its position through global positioning system (GPS) and knows its neighbours' location through beacons, which include ID, position, speed, and direction of node's mobility.

4.1. **Location server selection algorithm.** A node always checks the beacon it receives from neighbouring nodes to discover if any location server is within range. Beacon packet includes some useful information about each node as shown below, which enables location server to distinguish itself from other nodes. The ID of a node is used to distinguish a node among other nodes, while the position is required in enabling position-based routing protocol to forward message to destination. Speed is one of the criteria used in determining the best candidate for location server. Direction will be used to select other members of the quorum group for each direction. The LS field indicates the status of a node. The field is set to 1 if a node is a location server. 0 indicates it is a normal node. The *Flag* field determines the quorum group of location servers a node belongs to. *Role* determines the role of a location server whether main or passing location server. With the beacon information, nodes are able to know each other's information and the information is used during the nomination of the best node as an MLS, which is based on shortest distance and lowest speed.

$$BEACON = (ID, Position(X, Y), Direction, Speed, LS, Flag, Role, Timestamp)$$

VQPLS uses two parameters in selecting a location server, i.e., velocity of a node's movement and its distance from the centre point. The optimum parameters for selecting a location server are represented by:

$$BLS = distoptim + spdoptim \tag{20}$$

where *BLS* is the best location server, *distoptim* is the optimum distance of a node to centre point of an intersection, and *spdoptim* is the optimum speed of the node.

A node periodically checks its position to determine whether it is inside intersection vicinity or not as depicted in Figure 1. It also checks beacon it receives from neighbours.

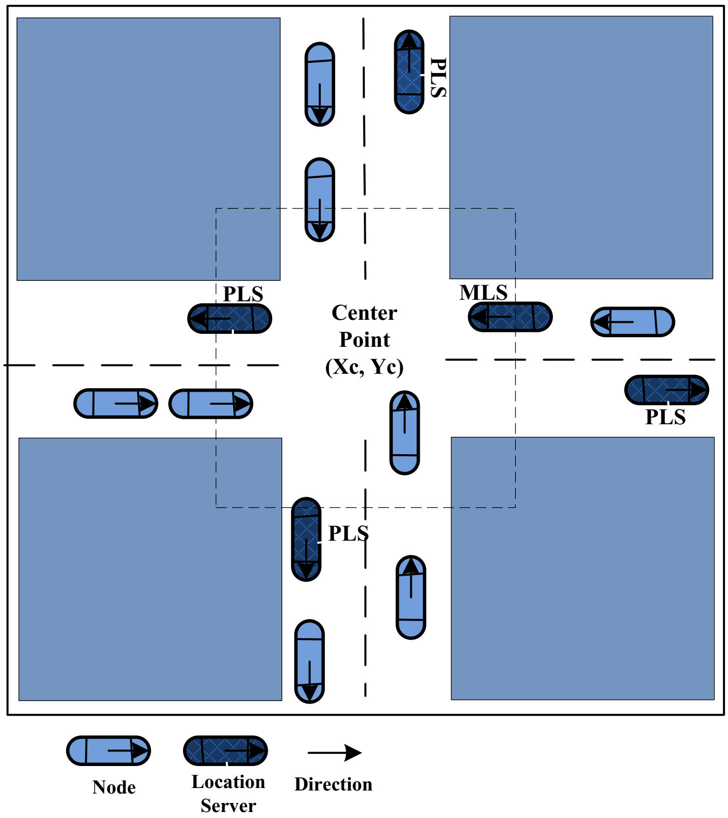

FIGURE 1. Selecting location server inside an intersection

If a node finds itself close to an intersection by using Equation (21) and it stops or moves with lowest speed compare with neighbours, then it will be nominated as a location server. Once it is nominated, it will change its status in the beacon to declare itself as a location server. Nodes close to this location server, which is one or two-hops away, need to send their updates to this LS. The location server is called MLS. This MLS will select some nodes to hold location information of nodes called as PLS.

$$Distance = \sqrt{(cpx\text{-}nx)^2 + (cpy\text{-}ny)^2} \tag{21}$$

where cp is centre point coordinates of intersection, and n is node's coordinates.

A criterion for selecting PLSs is basically based on their moving direction. MLS will select n number of PLSs, which is equal to number of different available directions in range of MLS. Both MLS and PLSs form one quorum of location servers for an intersection. MLS periodically checks the number of PLSs within its range. If the number is below the threshold, it determines the number of needed nodes and selects the number of new PLSs that equals to the number of needed nodes. Selecting new PLSs is based on hybrid method between unicast and multicast. If the number of available PLSs in range of MLS is 1 and 3 more PLSs are required, then a multicast is invoked by the MLS. On the other hand, if the number of PLS is only 1, then MLS just unicasts the locations of information table to that particular node. Checking the number of needed updates is represented in Equation (22) below.

$$NumUpdates = directionsnum - PLSnum \tag{22}$$

Number of directions ($directionsnum$) represents number of total directions, and PLS-num is number of PLSs in range of MLS. This determines whether multicast update or unicast is invoked:

$$\textit{Multicast Update if NumUpdates} > 2$$
$$\textit{Unicast Update if NumUpdates} \leq 2$$

VQPLS avoids broadcasting packets inside intersection vicinity to minimise packets collisions that can cause degradation in network performance.

PLS has two main functions. Firstly, it functions as a backup in case of MLS failure. Secondly, it acts as location server answering queries sent to MLS at intersection vicinity. These reduce overhead on MLS, and number of hops a query packet may take to reach MLS that has destination location. MLS updates PLSs of any new nodes joining the intersection and hash their nodes to MLS. Algorithm 1 shows the sequence of steps for selecting location server. However, the constructed quorum is terminated in case MLS moves away from intersection vicinity and finds no other node in range to pass the copy of locations table.

Algorithm 1 Location Server Selection Algorithm

```
while node inside intersection:
 if it is first node inside intersection then
     Nominate self as MLS
     Change Beacon's fields LS, flag, and Role
     Start constructing locations table
     Check neighbours' directions
    SelectingPLSsFunction(neighboursDirections)
    elseif it is MLS and moves away of intersection's vicinity then
        for i = 1 to last neighbour
            MLSSelectionMethod(currentMLSSpeed, currentMLSDistance)
            MLSSelectionMethod(neighbourSpeed, neighbourDistance)
        endfor
        return selected neighbour
      UnicastLocationsTable(newMLS)
      New MLS change Beacon's fields LS, flag, and Role
      Checks neighbours' directions
      SelectingPLSsFunction(neighboursDirections)
   endif
 endif
endwhile
```

4.2. **Location update.** The update packet contains node's ID, position, speed, direction, quorum group ID, and MLSs position. Intermediate nodes forward the packet through geographical forwarding scheme based on closest neighbour to destination. Once a PLS receives an update packet, it does not directly send this update packet to MLS. Instead, it collects the received update packets and sends it to MLS after a period of time. This process ensures less frequent sending of packets towards MLS and intersection vicinity. The node stops sending updates to its MLS once it becomes closer to another quorum group of location servers. It then hashes itself to the newly discovered MLS and starts to update its location to the new location server.

4.3. **Location query algorithm.** Source node sends a location query message to acquire location of destination node. Initially, source node is assumed to know the destinations' ID, and then it includes this ID in the query packet along with source ID, source position, and source location servers quorum information in addition to timestamps. The query message is sent to MLS that it is hashed to, if it is one hop away as depicted in Figure 2. Otherwise, the forwarder that receives the query will check their database of any cached up-to-date information entry of the destination. If there is an up-to-date location for destination, then a reply message is generated and forwarding towards MLS is stopped. The reply message is sent to query node. If no location is found for the destination, intermediate node keeps forwarding query packet to MLS or any near PLS. Once MLS or PLS receives this query, it looks up its own locations table. If there is an entry to the queried node, then the destination node resides on the same intersection. MLS or PLS replies latest received location of destination to source, then the source uses this location to communicate with the destination. If location is found but outdated, the location will be predicted using proposed prediction algorithm explained in Section 4.4. In case that MLS and PLS have no location information about destination, the query packet will be forwarded to surrounded quorum groups. The pseudo-code in Algorithm 2 describes the query algorithm.

Algorithm 2 Location Query Algorithm

1: A query packet sent in the network do:
2: **if** forwarder has up-to-date location **then**
3: got to step no. 20
4: **else** unicast Query Packet to nearest PLS or MLS
5: **endif**
6: **if** PLS finds up-to-date location **then**
7: got to step no. 20
8: **else** Predict outdated location and go to step no. 20
9: **elseif** no entry found on PLS locations table **then**
10: unicast Query Packet to MLS
11: **endif**
12: **endif**
13: **if** MLS finds up-to-date location **then**
14: got to step no. 20
15: **else** Predict outdated location and go to step no. 20
16: **elseif** no entry found on MLS locations table **then**
17: multicast query packet to surrounded quorums
18: **endif**
19: **endif**
20: Reply destination's location to query originator
21: Stop query process

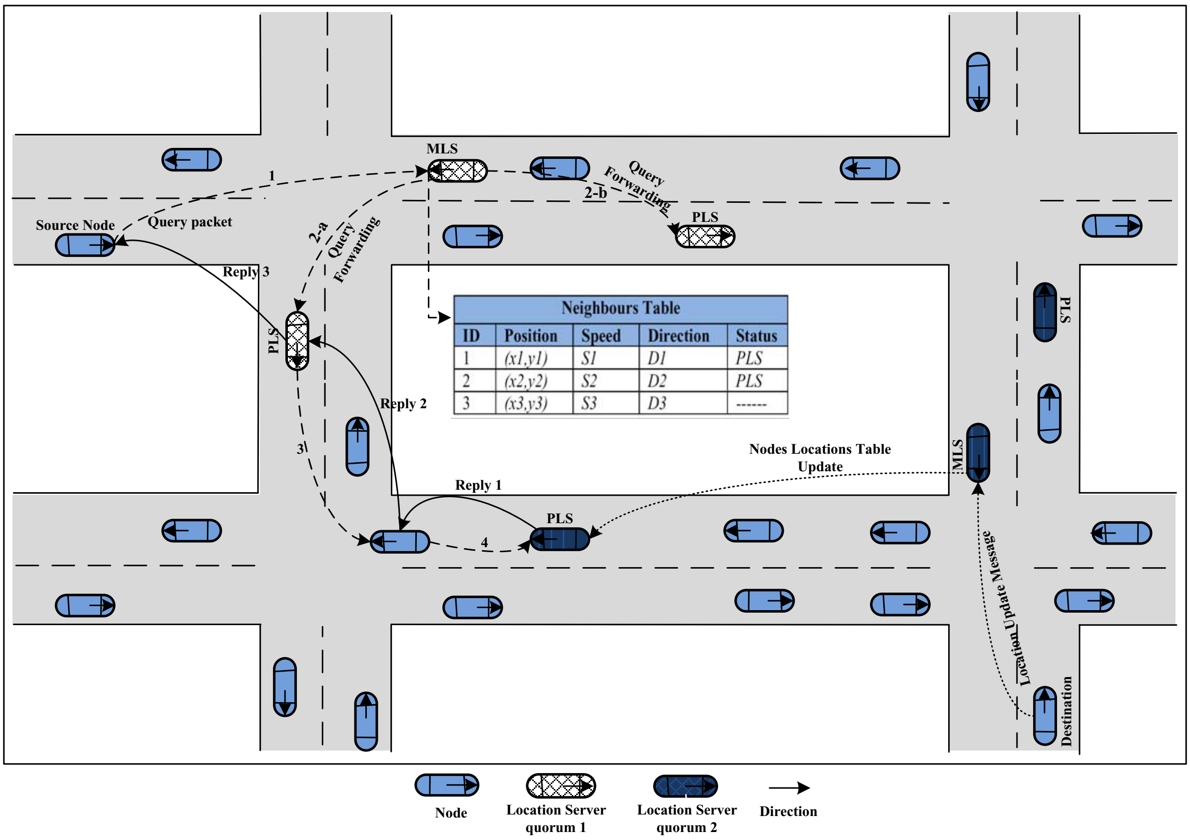

FIGURE 2. Querying of a destination's location

The query packet has TTL field that expires after a given time period to avoid looping over the network. If there is no reply to a query after a period of time equals to TTL, the query node will resend the query. However, if no reply is received after the second query, the query node gives up the query.

4.4. **Location prediction algorithm.** A hybrid of GM(1,1) and Alpha-Beta-Gamma Filter is proposed to improve the accuracy of grey prediction model called Grey Prediction-based Alpha-Beta-Gamma Filter (GP-ABGF). This hybrid is reasonable because of the effectiveness of grey theory model in forecasting more than one future data based on a few numbers of historical data. In addition, the hybridisation is able to overcome the noise of predicted data when sequence of data enter GM(1,1) is collected over irregular time intervals. It is also promising in predicting the data when the moving object accelerates due to the use of Alpha-Beta-Gamma Filter that is efficient in filtering noisy data when acceleration of moving objects is considered.

The prediction starts by entering four or more sequence of data $X^{(0)}$ using GM(1,1), and time response equation $\widehat{x}^{(1)}(k+1) = \left[x^{(0)}(1) - \frac{b}{a}\right] e^{-ak} + \frac{b}{a}$. The predicted data are calculated using Equation (13).

GP-ABGF can go to $K+N$ of future locations of moving object, where N is the step-size of prediction. The accuracy of these predicted locations depends on the quasi-smoothness and speed, where the higher the speed, the noisier the prediction, and vice versa.

The smoothness of AGO's data sequence is tested to determine whether the model makes accurate prediction or otherwise. The prediction is accurate if the results are below threshold value of 0.5 for $p(k)$ and [1, 1.5] for $\sigma^{(1)}(k)$. Otherwise, the prediction is noisy and it needs to be filtered.

Alpha-Beta-Gamma Filter filters the noisy data produced by GM(1,1) due to acceleration or irregular intervals over collecting sample data sequence. Alpha-Beta-Gamma Filter uses $K + 1$ value produced by GM(1,1) to start its prediction. The filter will predict $K + 2 \cdots K + N$ by using the available information such as speed, acceleration, last position, and time using Equation (16).

Algorithm 3, GP-ABGF shows the steps and conditions that should be met whenever the filter needs to be used. The algorithm begins with data sequence and generation of AGO's sequence that will be used for the rest of grey model prediction steps. They are also used to test quasi-smoothness of the model. Once the quasi-smooth condition is satisfied, the data obtained from the model are accurate. If the condition is not satisfied, the prediction is noisy and it needs to be filtered. Alpha-Beta-Gamma Filter prediction step-size goes up to $K + N$. N is determined by time difference between time of prediction and final-time a location is received from the moving object. The speed also plays a major role in predicting $K + N$ future locations, where lower speed leads to large number of N and higher speed leads to low N. The value of N has limit due to high cumulative error especially when $N > 3$ and the speed is higher than 10 meter/second.

Algorithm 3 Location Prediction Algorithm

1: Prediction is invoked do:
2: Retrieve locations $X^{(0)} = (x^{(0)}(1), x^{(0)}(2), \ldots, x^{(0)}(n))$ with timestamps
3: Calculate the step-size value for prediction
4: N ← step-size
5: Invoke GM (1,1) to predict K+N
6:   for i = 0 to N
7:     GM(1,1) ← $X^{(0)}$
8:     $\hat{x}^{(0)}(k+1) = \hat{x}^{(1)}(k+1) - \hat{x}^{(1)}(k)$
9:     Return K+N
10:   **endfor**
11: Test the quasi-smoothness condition
12: **if** $p(k) < 0.5$ and $\sigma^{(1)}(k)$ is in [1,1.5] **then**
13:   K+N is accurate
14:   Go to step no. 26
15: **endif**
16: **elseif** $p(k) > 0.5$ and $\sigma^{(1)}(k)$ is not in [1,1.5] **then**
17:     K+N is noise
18:     Invoke the Alpha-Beta-Gamma Filter
19:   for i = 0 to N
20:     ABGF ← $\hat{x}^{(0)}(k+1)$
21:     $x_p(k+1) = x_s(k) + T.v_s(k) + 1/2 \, . \, T^2 \, . \, a_s(k)$
22:   Return K+N
23   **endfor**
24: Go to step no. 26
25: **endif**
26: Reply destination location to query originator
27: Stop prediction process

5. **Results and Discussion.** The evaluation of the proposed protocol is divided into two parts; the accuracy test of prediction algorithm and analysis of the proposed prediction-based location service protocol. The following subsections explain each experimental set-up.

5.1. **Results of prediction algorithm.** For evaluating the proposed prediction algorithm, SUMO simulator of moving objects is used to simulate car movements in urban area [34]. The nodes in SUMO move on roads of urban map provided. Speed and direction are managed by mobility model that SUMO simulator provides. Thus, nodes change their speed based on road topology such as low speed at intersections and dense areas.

The collected data of moving nodes are classified into three main categories: (1) collection of data on a moving car with constant speed over regular time intervals to show the accuracy of grey model, (2) data collection over regular time intervals with high speed acceleration to show acceleration effects on grey model and to evaluate efficiency of GP-ABGF algorithm, and (3) data collection over irregular time intervals with constant speed. Data with irregular time intervals are to show the effectiveness of irregular time intervals on grey prediction accuracy and also to show the GP-ABGF overcomes this effect.

The collected data are inputs to the proposed algorithm and for benchmarking. The results are compared with original GM(1,1) as well as original locations of moving cars. Original locations are used to show the accuracy of the predicted locations of GP-ABGF, while GM(1,1) is to show difference between original grey prediction model and proposed GP-ABGF algorithm. GP-ABGF is benchmarked with prediction model based GM(1,1) proposed in [11].

5.1.1. *GM(1,1) accuracy.* Table 1 shows the grey model prediction with 5 data sequence and constant speed of car's movement. It clearly shows the two axes that make the accuracy of grey model. The change on the X-axis is higher than Y-axis due to the movement of object occurred on X-axis with average speed of 10 m/s (meter per second).

TABLE 1. Accuracy of GM(1,1) prediction model

Sequence	X Real	Y Real	X GM(1,1)	Y GM(1,1)	K+N
1	501.69	520.49	501.69	520.49	
2	501.45	541.27	501.36	544.44	
3	501.33	563.85	501.44	564.12	
4	501.48	583.58	501.51	584.52	
5	501.64	605.02	501.59	605.66	
6	502.42	636.86	501.66	627.56	K+1
7	501.64	657.68	501.74	650.26	K+2

Figure 3 shows the predicted locations by grey model on both axes using Equation (13). It is worth mentioning that quasi-smoothness satisfied $p(k)$ less than 0.5 and $\sigma^{(1)}(k)$ around 1 of the data sequence and the prediction error is negligible, thus, filtering is not required. The Accumulated Generating Operation (AGO) diminishes the randomness in data and makes good accuracy in GM(1,1).

5.1.2. *Acceleration effects.* Acceleration has a big impact on the accuracy of grey model. The quasi-smooth check exceeds the specified threshold value, which indicates noise occurring in the prediction model. Value of $p(k)$ calculated by Equation (14) is around 1 and $\sigma^{(1)}(k)$ is about 2, thus GP-ABGF algorithm needs to filter the noisy data.

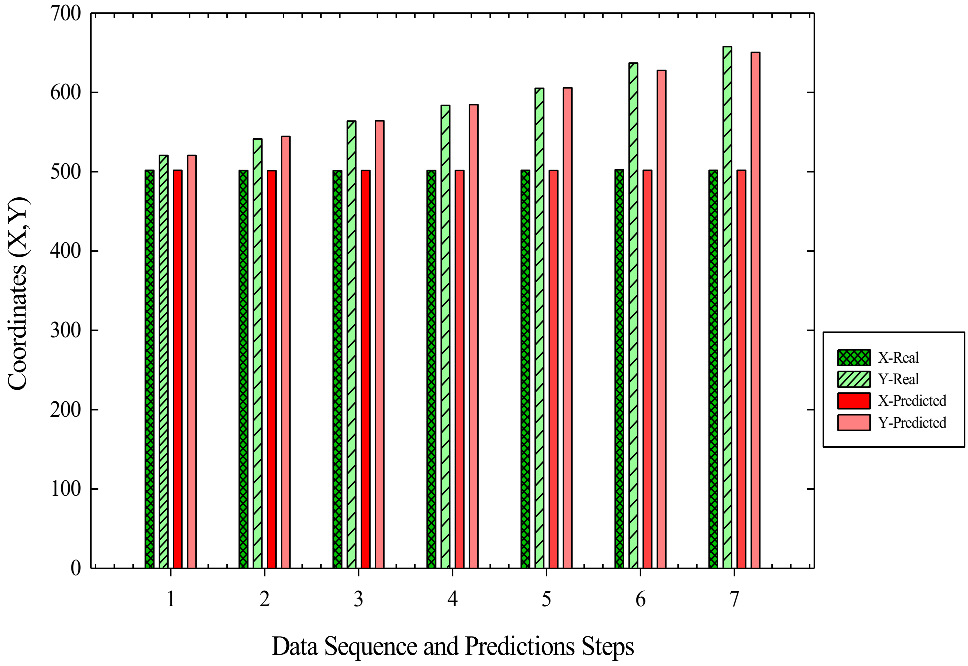

FIGURE 3. Accuracy of GM(1,1) prediction model

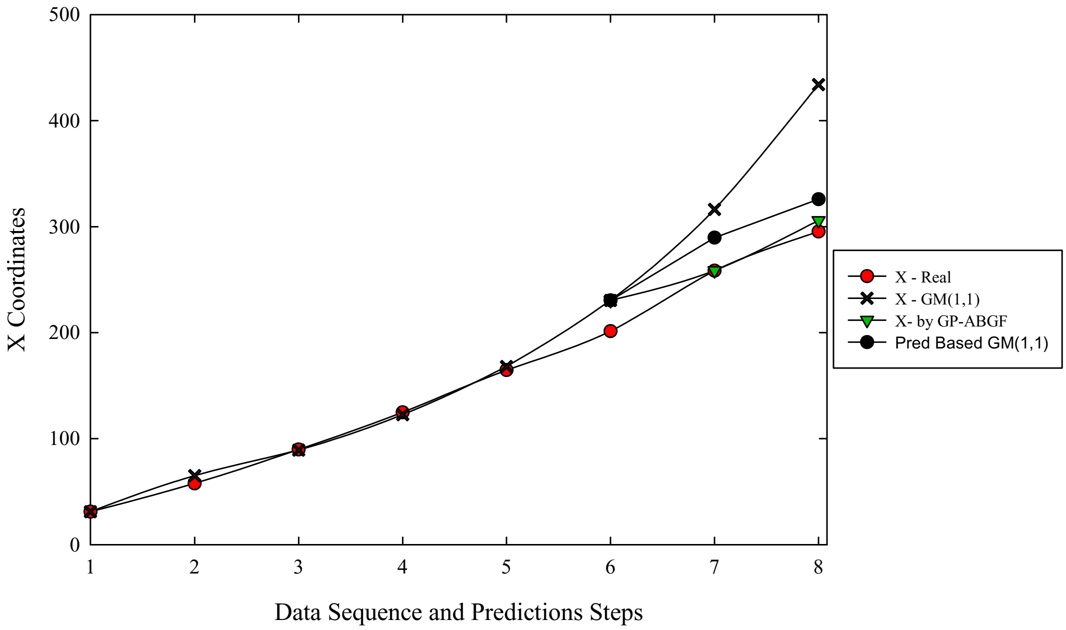

FIGURE 4. Accelerations effects and filtering noisy locations

The high speed affects value of N (number of future prediction steps). Thus, the prediction as shown in Table 2 is up to $K + 3$, whereby the car's speed reaches 19 m/s. The speed is considered high with the speed of cars movement is equivalent to 68.4 km/h in urban areas. Results depicted in Figure 4 show two main impacts on accuracy. Firstly, the high speed limits the number of future location that could be predicted as mentioned earlier, the higher the speed, the noisier it will be. The acceleration of nodes from 10 m/s to 19 m/s within few seconds influenced the accuracy of GM(1,1).

TABLE 2. Accelerations effects and filtering noisy locations

Sequence	X Real	Y Real	X GM(1,1)	Y GM(1,1)	X GP-ABGF	X PredBasedGM(1,1)	K+N
1	31.18	498.51	31.18	498.51			
2	57.77	498.84	65.08	498.6			
3	89.74	498.07	89.28	498.44			
4	124.79	498.51	122.5	498.28			
5	164.57	498.07	168.06	498.11			
6	201.38	498.84	230.58	497.95	230.58	230.58	K+1
7	258.4	498.40	316.35	497.79	275.84	289.6200	K+2
8	295.43	498.95	434.02	497.63	323.0	325.9300	K+3

Figure 4 shows that the filter cleared the noise of grey model accuracy starting from $K+2$, which is at Step 7 and overcame the challenged factors; high speed and acceleration. It could be concluded from Figure 4 that the acceleration affects the accuracy of GM(1,1), whereas it is clear that the value at Step 6 is predicted first by grey model, which is not very accurate compared with the actual location. However, the filtered location at Steps 7 and 8 are more accurate and close to real locations of the mobile node. On the other hand, the prediction model based on GM(1,1) accuracy was influenced by acceleration, which led to inaccurate prediction of moving objects.

5.1.3. *Irregular time intervals effects.* Grey model is time series prediction model that depends on regular intervals between sampled data sequence. Any changes on these intervals could affect the accuracy of grey model. Some of the collected data in this research were sampled based on irregular time intervals in order to test irregular intervals effects on the accuracy of grey prediction model. A node may report its location over irregular time intervals in mobility environment such as VANET, where nodes follow restricted mobility that may push the node to send location over irregular intervals of time to their correspondent location server.

Table 3 represents the the results of data collected over irregular time intervals, where the speed is almost stable around 10 m/s. The difference between predicted data produced by grey model and the real data is large on X-axis and this indicates the effect of irregular time series. However, $\alpha-\beta-\gamma$ Filter mitigates the corrupted and noise in predicted data and makes the prediction more accurate where $K+2$ of grey model is 926.84 although the real data is 905.31. The location predicted by $\alpha-\beta-\gamma$ Filter is 900.4, which is close to real location. $K+3$ and $K+4$ show interesting results where the difference between grey

TABLE 3. Irregular time intervals effects and filtering noisy locations

Time	X Real	Y Real	X GM(1,1)	Y GM(1,1)	X GP-ABGF	X PredBasedGM(1,1)	K+N
2	556.48	498.57	556.48	498.57			
7	610.05	498.20	629.52	498.23			
14	692.49	498.37	680.15	498.3			
18	734.77	498.29	734.86	498.57			
23	787.69	498.48	793.97	498.44			
28	851.70	498.59	857.84	498.51	857.84	857.84	K+1
33	905.31	498.37	926.84	498.58	900.4	880.90	K+2
37	947.22	498.16	1001.40	498.65	944.4	925.30	K+3
42	1003.64	499.02	1081.95	498.72	988.4	969.50	K+4

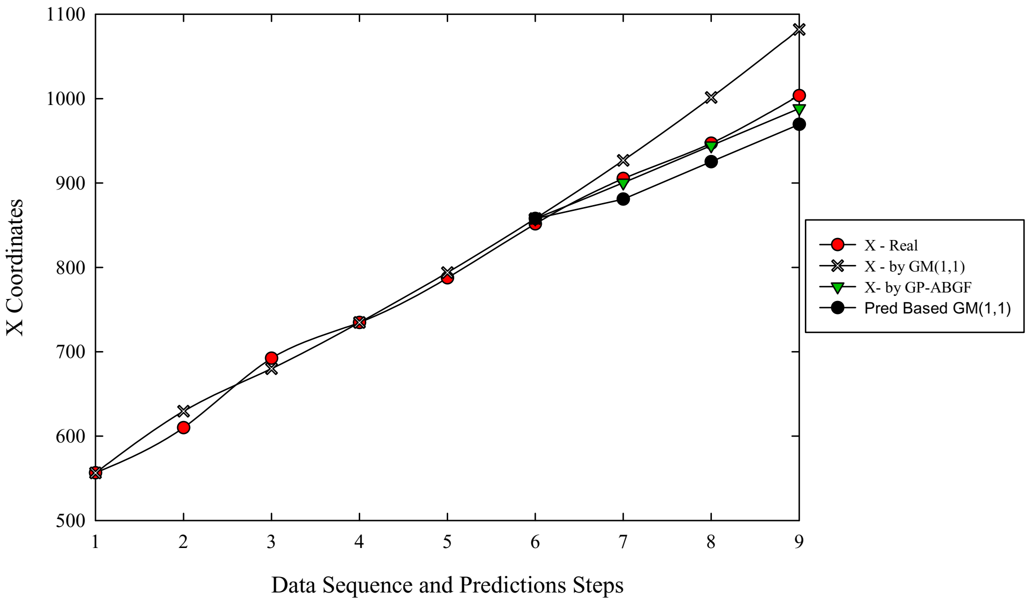

Figure 5. Irregular time intervals effects and filtering noisy locations

predictions and real data are very small as shown in Figure 5. This accuracy is achieved because of stable speed of the moving object, and high efficiency of filtering using $\alpha-\beta-\gamma$ Filter that applies Equation (16). The proposed filter filtered the big noise and returned the difference to an acceptable range. Nodes in VANET send their location update packets to location servers based on time or distance. In distance-based scheme, time between sending packets of update is irregular. It influences prediction and introduces noise, thus GP-ABGF filters out this noise and produces accurate locations. Enhanced prediction-based GM(1,1) algorithm was designed based on regular intervals between sampled data and this could not be in high mobility environment such as VANET. As a result, predicted locations of enhanced prediction-based GM(1,1) algorithm are inaccurate.

5.1.4. *Relative errors check.* Relative errors check is for checking errors between predicted and filtered values by comparing the accuracy of filtered results and GM(1,1) as shown in Table 4.

The quasi-smoothness of the noisy predicted data is $p(k) = 0.59$ and $\sigma^{(1)}(k) = 1.59$. Even though there is only a slight difference between the thresholds, the effects are huge on the predicted values.

Figure 6 shows the calculated relative errors using filtering and prediction with noise. The relative error of filtering is around 1.51 and below along all the steps, whereas for GM(1,1) the error goes to 7.8, which is higher than filtering. On the other hand, prediction-based GM(1,1) relative errors reach 3.4, which is better than the original

Table 4. Relative errors of predicted and filtered locations

Sequence	Error of GM(1,1)		Error of GP-ABGF		Errors of Pred Based GM(1,1)	
	Errors	Relative Errors (%)	Errors	Relative Errors (%)	Errors	Relative Errors (%)
	ε(k) = X Real - X GM(1,1)	Δk = |ε(k)| / X Real	ε(k) = X Real - X GP-ABGF	Δk = |ε(k)| / XReal	ε(k) = X Real - X Pred	Δk = |ε(k)| / XReal
6	-6.14	0.72	-6.14	0.72	-6.14	0.72
7	-21.53	2.3	4.91	0.54	24.41	2.6
8	-54.18	5.7	2.82	0.29	21.92	2.3
9	-78.31	7.8	15.24	1.51	34.14	3.4

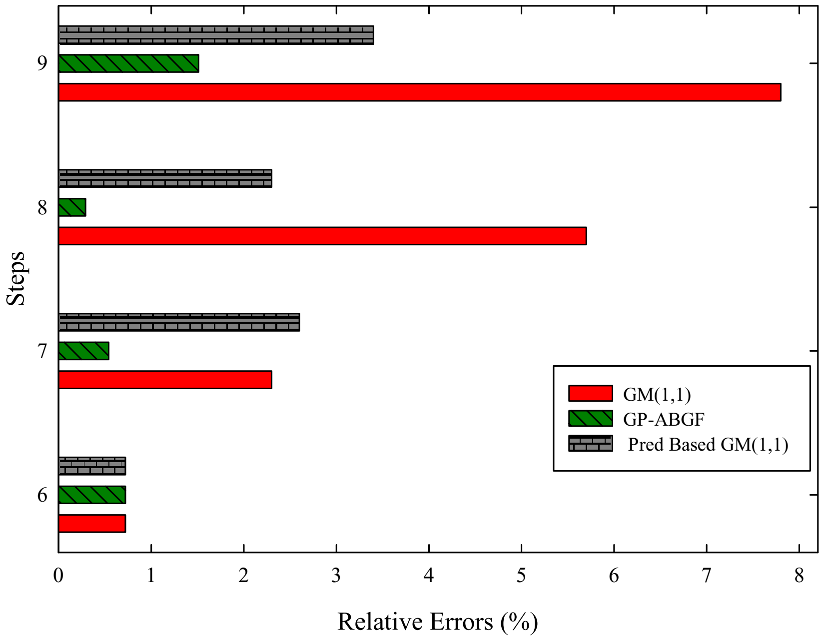

FIGURE 6. Relative errors of predicted and filtered locations

GM(1,1), however, GP-ABGF performs better. This proves the high accuracy that could be achieved by filtering noise of GM(1,1) when the $\alpha - \beta - \gamma$ Filter is used. These differences in relative errors are important especially when forecasting data that do not accept high error. Moreover, this validates the correctness of mathematical modelling used in this work in producing accurate prediction.

5.2. **Location service protocol results.** The performance measures of VQPLS protocol are evaluated using Jist/SWANS simulator [35]. Simulated road scenarios for an urban area are extracted from TIGER/Lines map files [36] representing down-town Chicago, with a dimension of an area of around 800 square meters. Street mobility model is used to simulate mobility of nodes on provided map roads. Each node is assigned a destination point on the map and moves towards the point with specified maximum velocity. Once a node reaches its destination point, it pauses for a short period of time and moves again towards another destination point. Number of query originator nodes and their destination nodes are set to 10% of total number of nodes. A node originates and sends a query packet towards its intersection vicinity in order to retrieve destination node's location for starting a communication.

Benchmarking VQPLS is done against two location service protocols; Self Organizing Location Server (SOLS) [37] and ILS [16]. SOLS is a quorum-based location service protocol, while ILS is hash-based VANET location service protocol. The metrics used to evaluate the performance of the proposed protocol are age of old locations returned by queries, control overhead, end-to-end delay, and ratio of packets delivery. These metrics are measured against speed of vehicles and density of vehicles. Parameters for the simulation set-up are summarised in Table 5.

Chord algorithm used in ILS protocol keeps old information and thus returns old locations. Quorum systems also return old data such as in SOLS protocol. The age of old locations returned by queries is depicted in Figure 7. Old locations returned by SOLS are higher than other protocol as they reach 25 seconds old. On the other hand, in ILS, age of queries is around 17 seconds. VQPLS shows significant reduction in age of locations

Table 5. Simulation parameters

Parameter	**Value**
Simulation area	800m $*$ 800m
Number of vehicles	200
Transmission range	100m
MAC protocol	IEEE 802.11 DCF
Simulation time	300s
Beacon Interval	1 beacon/second
Vehicles velocity Ranges	4, 6, 8, 10, 12 and 14 meter per second

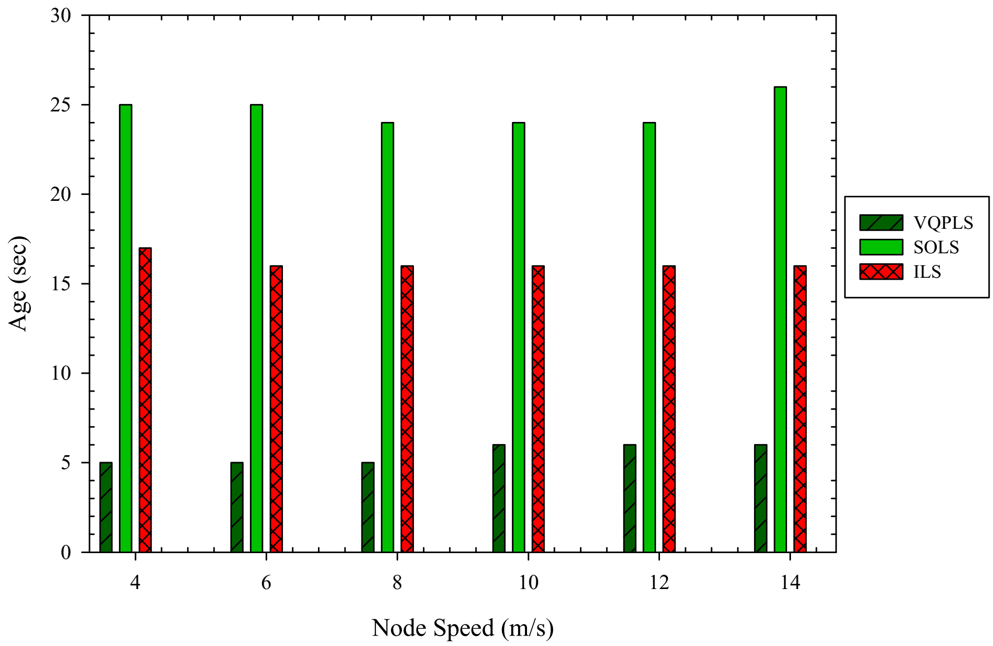

Figure 7. Age of old locations returned by queries

stored at location servers. In VQPLS, prediction overcomes old data by predicting the next location. Since the proposed GP-ABGF algorithm has high efficiency in returning next locations of nodes for number of step-size, the age of old locations returned by VQPLS is small. However, they have no effects on overhead and delivery of packets.

5.2.1. *Effects of vehicles mobility.* This test examines the effects of varying speed of nodes on the metrics such as packets delivery ratio, control overhead, and end-to-end delay.

Results on Figure 8 show that VQPLS delivers more packets compared with other protocols over different speeds of movement. The ratio of packets delivery ratio for VQPLS is about 90 percent while other protocols deliver much less percentage of packets especially ILS. The first significant reason of the increase in packets delivery ratio is due to assigning location servers based on intersections in addition to distributing PLSs on road segments surrounding the intersections. Another reason is reduction of contention on medium at the intersection because load is distributed among MLS and its PLSs, in addition to the use of prediction that produces accurate locations of destinations. Packet delivery ratio for ILS is lower than VQPLS because all queries are sent directly to intersection location server. On dense VANET, this causes high packet collision that degrades delivery of packets to their destinations. On the other hand, SOLS achieves packet delivery better than ILS.

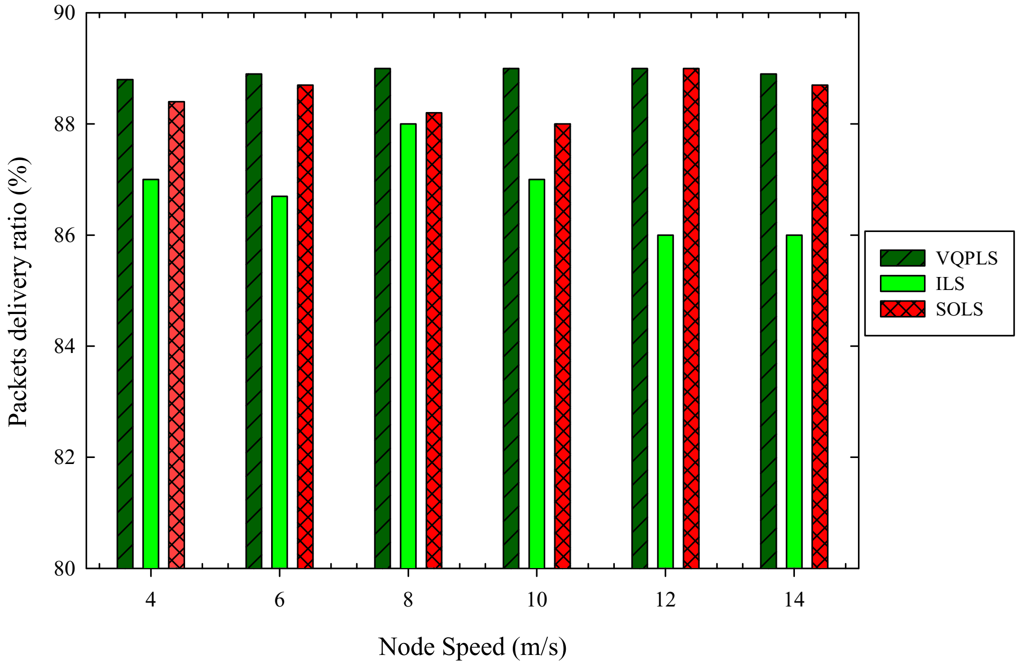

FIGURE 8. Packets delivery ratio

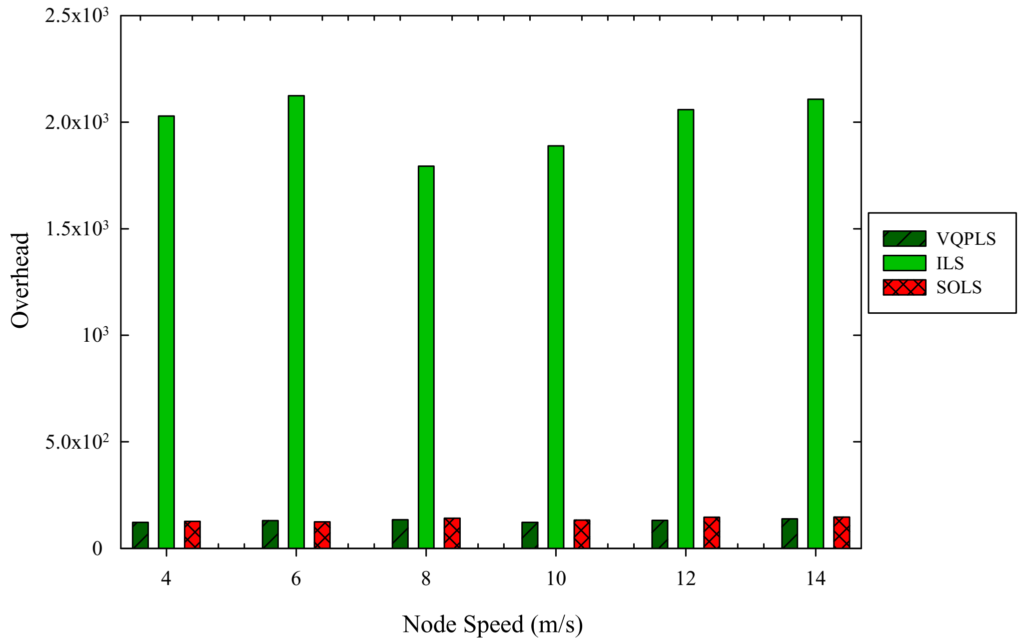

FIGURE 9. Control packets overhead per server

Figure 9 depicts the overhead that represents control packets overhead per server per minute. The results show that the overhead of VQPLS is lower among other protocols as expected from load distribution results. This is because inside the intersection vicinity, the number of control packets such as beacons, update locations, locations table contents updates, and queries is high. As a result, selection of PLSs for different directions to function as location servers reduces the load on intersection vicinity. Selecting MLS based on distance to centre point and low speed makes this MLS serves for longer time and thus providing stability in quorum management. If selecting MLS is just based on

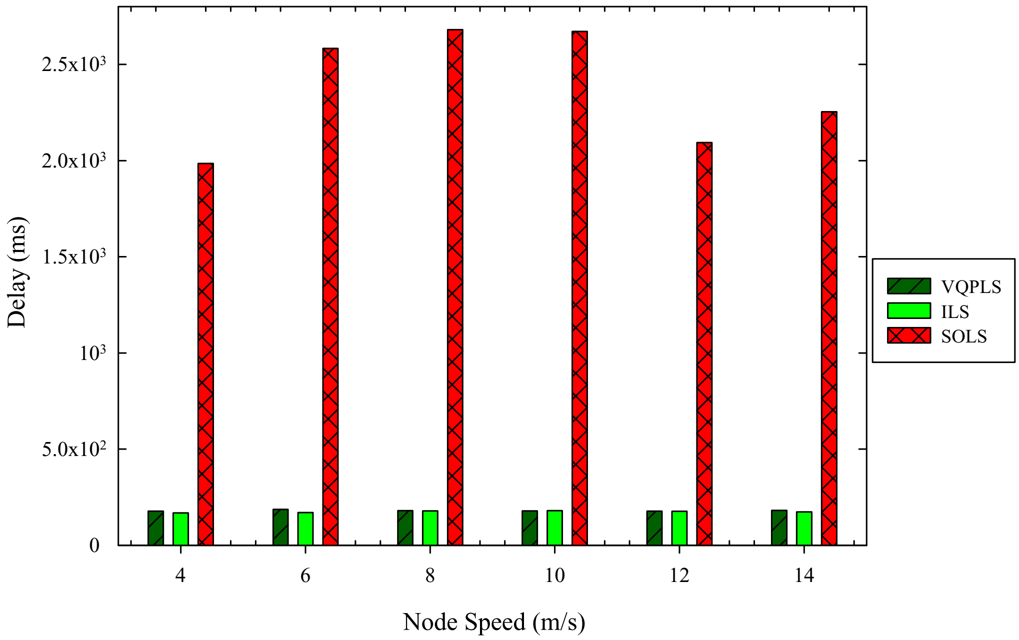

FIGURE 10. Average end-to-end delay

close distance to centre point, this leads to high frequency of selecting MLSs during simulation time, which requires high frequency of locations table exchange. Nevertheless, overhead slightly increases with the increase in mobility, because high mobility makes a PLS goes out of MLS range in short time, and MLS needs to reselect new PLS and sends a copy of locations table to it. On the contrary, ILS manages too many control packets for selecting successors and predecessors. ILS selects location server based on ID that does not ensure stable location server. Therefore, ILS suffers from high overhead and it does not distribute load on multi servers. Master node in SOLS protocol broadcasts duplication of nodes locations to slaves and this leads to overhead. But here, it is not so much because SOLS does not select location servers based on intersection. Instead, it selects home location of location servers in places with low number of nodes and then low overhead, but this will cost high delay of routing packets.

In Figure 10, the end-to-end delay of VQPLS is lower than SOLS. This is because the selection of location servers is based on intersection that allows packets forwarding of over well-connected chain of nodes until they reach a PLS or MLS of the specified quorum group. Meanwhile, SOLS shows higher end-to-end delays as it uses specified regions within the network area and assigns a number of nodes as location servers. As a consequence, packet routing suffers from high number of hops required to reach the specified location server due to using perimeter mode of routing packets to avoid void areas. ILS utilises Chord algorithm that manages packets forwarding and minimizes the delay, but with cost of outdated information stored by Chord algorithm.

5.2.2. *Effects of vehicles density.* In this section, the effects of vehicles density (number of nodes) in urban environment are analysed. The average speed of nodes is set to 10 m/s. Beacon intervals are set to 1 beacon per second, and location update scheme is distance-based. The three parameters: delivery ratio, overhead and end-to-end delay shown in Figures 11, 12 and 13 reflect the performance of VQPLS protocol versus other benchmarking protocols in term of number of nodes. Each point in the graphs represents average of 10 simulation runs. Number of vehicles in the simulation is varied; 100 nodes

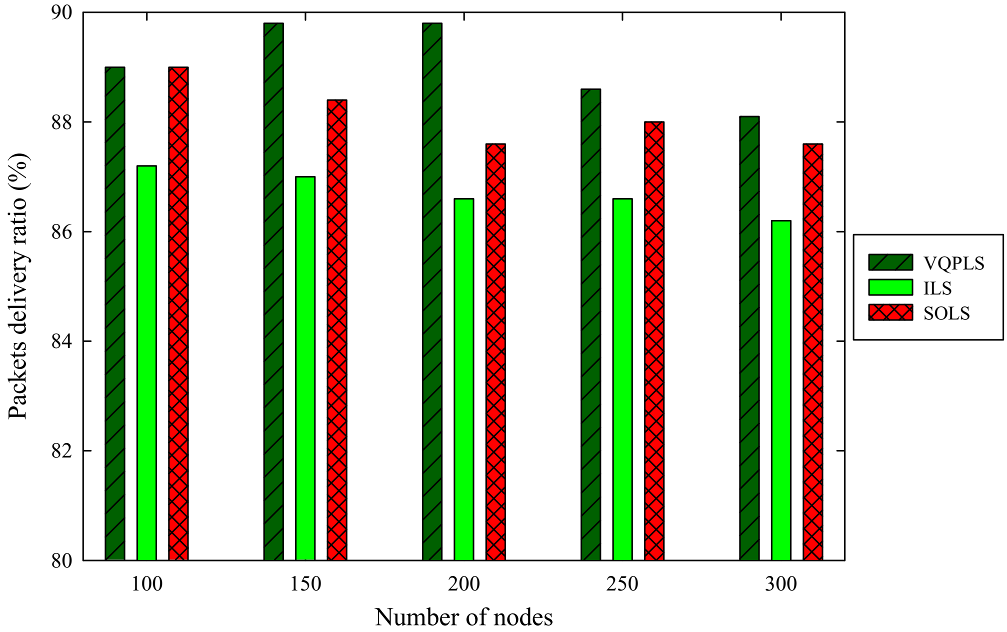

Figure 11. Packets delivery ratio with 10 m/s speed

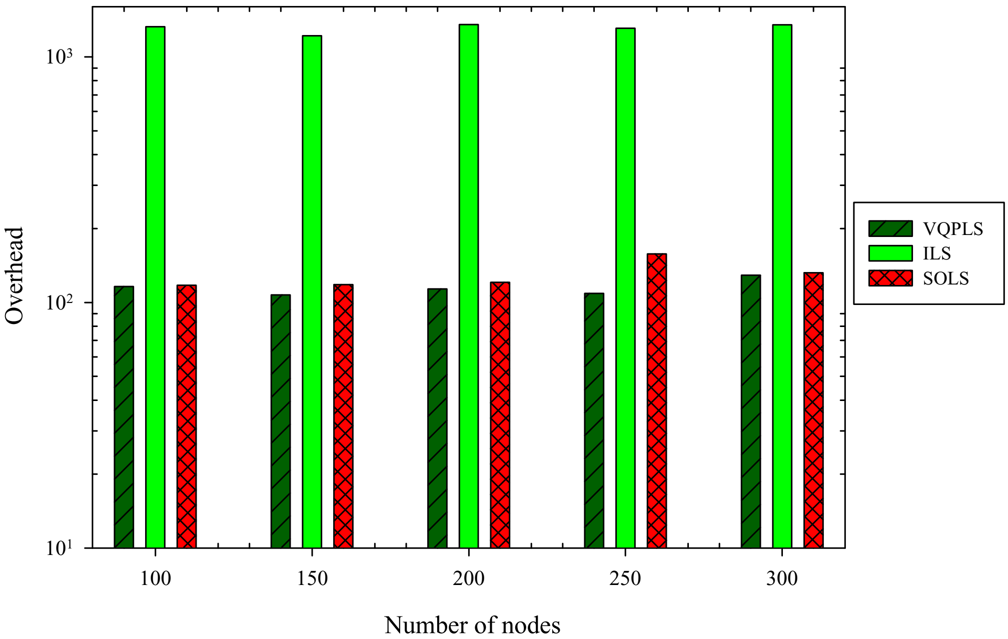

Figure 12. Control packets overhead per server with 10 m/s speed

(which represent low density), 200 nodes (which represent high density), and 300 nodes to correspond very high density over 800 square meter area. These densities reflect the situation of urban area during day time (i.e., rush hours) while at other times, it is less dense. These varying nodes densities are to test the scalability of VQPLS, and the effect of overhead and delivery of packets in addition to showing the contention effects on wireless medium.

The average delivery ratio of packets to destinations of VQPLS is higher than other protocols as depicted in Figure 11. This reflects its design efficiency, which is based on number of location servers of each individual quorum group. With high density of vehicles (e.g., 250 and 300 nodes), delivery of packets becomes lower. This is because of the high contention on the medium for transmission that causes high packets collisions. The contention on medium is high due to large number of packets exchanged between nodes such as query, beacons, location update, and locations table contents exchanged between MLS and its PLSs. High number of nodes in wireless environment would increase the connectivity, thus increasing the delivery ratio. However, high density of nodes also means high contention between these nodes, and thus high packet drops and collisions between packets. These are the reasons behind lower packets delivery in VQPLS when nodes density increases. In general, VQPLS packets delivery is better than other protocols, and this proves the efficiency of quorum in delivery and fault tolerance mechanism it offers. The prediction algorithm has a role in keeping high delivery due to accurate locations it provides, thus increasing the chance of delivering packets to their destination correctly. Generally, VQPLS is better than SOLS and ILS in delivering packets to destinations with low and high number of nodes.

Figure 12 shows very scalable overhead of VQPLS versus different number of vehicles. The load distribution of answering queries by PLSs makes the intersection area receives lower number of queries and generates lower number of replies packets compared with that of ILS protocol. Consequently, location prediction by VQPLS reduces the overhead on intersection area with different number of vehicles. ILS produces high overhead because the number of nodes becomes higher in intersection vicinity, which increases the number of sent control packets to location server. This is the reason behind the high overhead. Another reason is that ILS selects location servers based on node's ID that does not ensure stability of selected location server. This leads to high frequency of changing in the assigned location server. Unlike ILS, SOLS protocol shows accepted overhead due to its quorum-based design. Thus, it uses many location servers for answering queries. However, the disadvantage of using SOLS is its high delay and old age of queries.

Figure 13 depicts the end-to-end delay where VQPLS shows scalability and has low delay over all nodes densities. The low delay in VQPLS is reasonable due to load distribution on multi servers around dense areas. The highly dense area causes contention

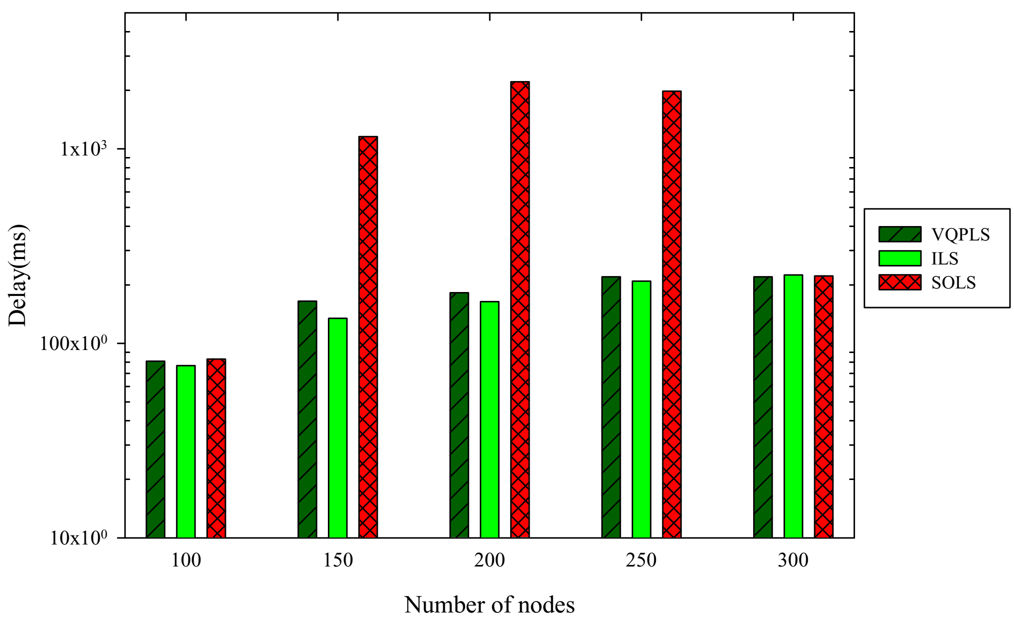

Figure 13. Average end-to-end delay with 10 m/s speed

on the medium for transmission, which causes high packets collisions and high back-off time delay due to waiting for free medium. ILS shows relatively low delay because it uses Chord algorithm, which directs packets that assist in routing packets within short time. However, this is not without disadvantage because the high packets overhead on location servers and stale information are managed by Chord algorithm. On the other hand, the increase in number of nodes increases the delay in SOLS. However, SOLS shape of the graph at 200 nodes reaches the peak of curve that could be due to selecting home region in a dense area. This leads to high contention between nodes due to high migration and duplication process between quorum members and this increases the end-to-end delay. In testing nodes with density 300, the delay of SOLS is declined.

6. **Conclusions and Future Work.** VQPLS protocol efficiently utilises the nodes located at intersection to be chosen as a quorum of location servers. VQPLS protocol chooses a node close to centre point with lowest speed compared with all neighbour nodes to be nominated as main location server (MLS). This MLS elects some other nodes called passing location servers (PLS) to form the whole quorum group. VQPLS uses PLSs for fault tolerance in high mobility environment, distributes the load on MLS and on intersection dense with vehicles, and reduces number of hops a query may take to reach MLS. Simulation results show the high performance of VQPLS against other location service protocols. The low overhead with high delivery ratio of packets with VQPLS outperformed other protocols as well as very low end-to-end delay. All tests of nodes speed variant proved scalability of VQPLS and its high attitude in urban environment.

Sampled data sequence with quasi-smoothness below a specified threshold can lead to accurate prediction with grey model prediction. This smoothness can be affected by two main factors: acceleration and irregular time intervals. The empirical evaluation shows that high speed affects accuracy and limits the future steps that can be predicted. Additionally, the acceleration influences the accuracy of grey model. However, the stable speed of moving objects is proven to have better accuracy even when collected data are in irregular time intervals. The inaccuracy of grey model due to acceleration and irregular time intervals was mitigated by GP-ABGF algorithm using $\alpha - \beta - \gamma$ filter. GP-ABGF takes into account the acceleration of moving objects and filters noisy data, and minimises error in prediction with $\alpha - \beta - \gamma$ filter. Prediction with high accuracy is important for many applications such as positions of mobile robots and positions of mobile nodes in ad hoc networks. The high accuracy of prediction means high reliability on the system.

Overall, the robust exploitation of urban topology and nodes features (speed and distance to intersection center point) made VQPLS performs well and routes packets with reduced load at intersection vicinity. The GP-ABGF prediction algorithm used with VQPLS protocol helps in reducing control packets overhead and produces accurate locations of destinations that increase packets delivery ratio.

An avenue of future work is that input data with high noise could affect the accuracy of GP-ABGF algorithm when the value of quasi-smoothness check is very high and over the threshold. However, adjusting $\alpha - \beta - \gamma$ parameters values may lead to producing more accurate values and this needs some considerations in future research. The prediction algorithm is used by location servers only, it is good to make normal nodes able to predict neighbours moved out of range, this may increase the efficiency of the protocol. In future work, these addressed issues will be improved.

Acknowledgment. This work is partially supported by Research Management Centre (RMC), Universiti Teknologi Malaysia (UTM), under project's vote no. 77271. The authors also gratefully acknowledge comments and suggestions of Prof. Dr. Jeffrey Yi-Lin Forrest from Slippery Rock University, (USA), which have helped in improving the prediction algorithm.

REFERENCES

[1] J. Nzouonta, N. Rajgure, G. Wang and C. Borcea, Vanet routing on city roads using real-time vehicular traffic information, *IEEE Transactions on Vehicular Technology*, vol.58, no.7, pp.3609-3626, 2009.

[2] M. Kåsemann, H. Hartenstein, H. Füßler and M. Mauve, Analysis of allocation service for position-based routing in mobile ad hoc networks, *Proc. of the 1st German Workshop on Mobile Ad-Hoc Networking, GI Lecture Notes in Informatics*, pp.121-133, 2002.

[3] J. Li, J. Jannotti, D. S. J. De Couto, D. R. Karger and R. Morris, A scalable location service for geographicadhocrouting, *MobiCom'00: Proc. of the 6th Annual International Conference on Mobile Computing and Networking*, pp.120-130, 2000.

[4] W. Kieß, H. Füßler, J. Widmer and M. Mauve, Hierarchical location service for mobile ad-hoc networks, *SIGMOBILE Mob. Comput. Commun. Rev.*, vol.8, no.4, pp.47-58, 2004.

[5] I. Stojmenovic and B. Vukojevic, A routing strategy and quorum based location update scheme for ad hoc wireless networks, *Technical Report TR-99-09, Computer Science, SITE*, University of Ottawa, 1999.

[6] M. Heissenbttel, T. Braun, M. Wlchli and T. Bernoulli, Evaluating the limitations of and alternatives in beaconing, *Ad Hoc Networks*, vol.5, no.5, pp.558-578, 2007.

[7] D. Son, A. Helmy and B. Krishnamachari, The effect of mobility-induced location errors on geographic routing in mobile ad hoc and sensor networks: Analysis and improvement using mobility prediction, *IEEE Transactions on Mobile Computing*, vol.3, pp.233-245, 2004.

[8] S. Sharma, V. Alatzeth, G. Grewal, S. Pradhan and A. Helmy, A comparative study of mobility prediction schemes for GLS location service, *Proc. of IEEE the Vehicular Technology Conference*, pp.3125-3129, 2004.

[9] E. Amar, S. Boumerdassi and E. Renault, *Predictive Scheme for Location Service in Mobile Ad-Hoc Networks*, Springer-Verlag, Berlin Heidelberg, 2009.

[10] M. Alam, M. Sher and S. Husain, Integrated mobility model (IMM) for VANETs simulation and its impact, *International Conference on Emerging Technologies*, pp.452-456, 2009.

[11] Y. Xiao, H. Zhang and H. Wang, Location prediction for tracking moving objects based on grey theory, *The 4th International Conference on Fuzzy Systems and Knowledge Discovery*, vol.2, pp.390-394, 2007.

[12] G.-D. Li, S. Masudaand, D. Yamaguchi and M. Nagai, The optimal GNN-PID control system using particle swarm optimization algorithm, *International Journal of Innovative Computing, Information and Control*, vol.5, no.10(B), pp.3457-3469, 2009.

[13] Y.-J. Chang and T. L. Shih, Intersection location service and performance comparison of three location service algorithms for vehicular ad hoc networks in city environments, *The 3rd International Symposium on Wireless Pervasive Computing*, pp.562-565, 2008.

[14] B. Ramakrishnan, Dr. R. S. Rajesh and R. S. Shaji, An intelligent routing protocol for vehicle safety communication in highway environments, *Journal of Computing*, vol.2, no.11, pp.65-72, 2010.

[15] H. Menouar, M. Lenardi and F. Filali, Movement prediction-based routing (MOPR) concept for position-based routing in vehicular networks, *Vehicular Technology Conference, IEEE*, pp.2101-2105, 2007.

[16] Y.-J. Chang and S.-Y. Wu, Intersection location service for vehicular ad hoc networks with cars in manhattan style movement patterns, *Proc. of the 5th International Conference on Ubiquitous Intelligence and Computing*, pp.284-296, 2008.

[17] X. Bai, X. Ye, J. Li and H. Jiang, VLS: A map-based vehicle location service for city environments, *IEEE International Conference on Communications*, 2009.

[18] G.-Y. Chang, Y.-Y. Chen and J.-P. Sheu, A region-based hierarchical location service with road-adapted grids for vehicular networks, *Proc. of the 39th International Conference on Parallel Processing Workshops*, pp.554-561, 2010.

[19] G. Zhang, W. Chen, L. Hong and D. Mu, A novel location service for urban vehicular ad hoc networks, *IEEE International Symposium on Microwave, Antenna, Propagation and EMC Technologies for Wireless Communications*, pp.513-516, 2009.

[20] M. Boussedjra, J. Mouzna, P. Bangera and M. Pai, Map-based location service for VANET, *International Conference on Ultra-Modern Telecommunications Workshops*, pp.1-6, 2009.
[21] D. Liu, I. Stojmenovic and X. Jia, A scalable quorum based location service in ad hoc and sensor networks, *IEEE International Conference on Mobile Ad Hoc and Sensor Systems*, pp.489-492, 2006.
[22] J. Wen and Y. Zhan, Vision-based two hand detection and tracking, *Proc. of the 2nd International Conference on Interaction Sciences*, pp.1253-1258, 2009.
[23] S. Liu, J. Forrest and R. Vallee, Emergence and development of grey systems theory, *Kybernetes*, vol.38, no.7/8, pp.1246-1256, 2009.
[24] G. Wei, Y. Ling, B. Guo, B. Xiao and A. V. Vasilakos, Prediction-based data aggregation in wireless sensor networks: Combining grey model and Kalman filter, *Computer Communications*, vol.34, no.6, pp.793-802, 2011.
[25] Y.-F. Wang and L.-L. Liu, Grey target tracking and self-healing on vehicular sensor networks, *EURASIP Journal on Wireless Communications and Networking*, vol.2007, no.63503, pp.1-14, 2007.
[26] Q. Fu, Y. Xiao and H. Yin, Object tracking algorithm based on grey innovation model GM(1,1) of fixed length, *Proc. of the 2009 WRI World Congress on Computer Science and Information Engineering*, vol.6, pp.615-618, 2009.
[27] R. E. Kalman, A new approach to linear filtering and prediction problems, *Transactions of the ASME Journal of Basic Engineering*, vol.82 (Series D), pp.35-45, 1960.
[28] H. Simpson, Performance measures and optimization condition for a third-order sampled-data tracker, *IEEE Transactions on Automatic Control*, vol.8, no.2, pp.182-183, 1963.
[29] C. Wu, P. Lin, Z. Han and S. Li, Simulation-based optimal design of $\alpha - \beta - \gamma - \delta$ filter, *International Journal of Automation and Computing*, vol.7, no.2, pp.247-253, 2010.
[30] D. Tenne and T. Singh, Characterizing performance of $\alpha - \beta - \gamma$ filters, *IEEE Transactions on Aerospace and Electronic Systems*, vol.38, no.3, pp.1072-1087, 2002.
[31] T. Tenne and D. Singh, Analysis of $\alpha - \beta - \gamma$ filters, *Proc. of the 1999 IEEE International Conference on Control Applications*, vol.2, pp.1342-1347, 1999.
[32] S. Liu and J. Forrest, Advances in grey systems theory and its applications, *Proc. of 2007 IEEE International Conference on Grey Systems and Intelligent Services*, Nanjing, China, 2007.
[33] D. Tenne and T. Singh, Optimal design of $\alpha - \beta - \gamma$ filters, *Proc. of 2000 American Control Conference*, vol.6, pp.4348-4352, 2000.
[34] D. Krajzewicz, G. Hertkorn, Christian Rössel and P. Wagner, SUMO (simulation of urban mobility): An open-source traffic simulation, *Proc. of the 4th Middle East Symposium on Simulation and Modelling*, pp.183-187, 2002.
[35] R. Barr, SWANS – Scalable wireless ad hoc network simulator user guide, *Cornell Research Foundation*, http://jist.ece.cornell.edu/swans-user, 2004.
[36] U. C. Bureau, *Tiger/Line 2009*, http://www.census.gov/geo/www/tiger/.
[37] G. Owen and M. Adda, SOLS: Self organizing distributed location server for wireless ad hoc networks, *International Journal of Computer Networks & Communications*, vol.1, no.1, pp.18-31, 2009.