Omnibone: An Efficient Service Data Circulation and Discovery Scheme in VANETs

Chyi-Ren Dow,* Yu-Hong Lee, Pa Hsuan, Yi-Tung Lee, and Shiow-Fen Hwang
Feng Chia University
Taichung, Taiwan
{crdow, p9840570, p9431820, m9738218, sfhwang}@fcu.edu.tw

Abstract

Due to recent developments in wireless communication networks, Vehicular Ad-hoc Networks (VANETs) technologies have received a lot of attention in the fields of information sharing and service discovery. However, due to the ever-shifting mobility of vehicle topology, vehicles moving along non-fixed routes may not find suitable next-hop vehicles. This paper proposes schemes to effectively circulate and discover service information with the aid of public transportation systems. Bus routes can be used to create a backbone structure on which data can be posted and circulated to avoid the broadcast storm problem. The proposed architecture can effectively disseminate and discover the required data through the traffic infrastructure and mobile vehicles. Experimental results demonstrate that the proposed scheme outperforms other schemes in terms of packet delivery ratio and end-to-end delay. Moreover, the overhead of this scheme is less than other schemes with an increasing of number of service requests.

Keywords: Vehicular ad-hoc networks, data circulation, service discovery, virtual backbone.

1 Introduction

Due to recent developments in wireless communication networks, Vehicular Ad-hoc Networks (VANETs) technologies have received a lot of attention in the information sharing and service discovery fields. In the research field of VANETs, data circulation refers to the spreading and transmission of service information into certain network areas, especially application-rich areas, making the information sharable to other users. Many data dissemination schemes first broadcast target messages to neighboring nodes, and then these nodes continue re-broadcasting the messages. Several schemes [1, 2, 3] attempt to reduce the number of broadcasts by adding restrictions on directions, probability-drops, or the number of forwarding nodes to avoid the broadcast storm problem. Most of these methods only broadcast data packets to specific areas, rather than the entire network, to reduce the number of transmissions.

Due to the ever-shifting vehicle topology, vehicles moving along non-fixed routes may not be able to find suitable next-hop vehicles. The use of a sharable backbone structure can minimize the number of retransmissions and preserve network connectivity. However, many issues must be addressed to deploy such a backbone for aiding data dissemination, including its establishment, maintenance, posting of service information, and service discovery. Since the VANET context is based on opportunistic networks, it is not possible to completely transfer a large amount of data in a short period. Thus, additional mechanisms are required to maintain the data transmissions. In VANETs, establishing and maintaining a dynamic virtual backbone [4] architecture requires a large amount of network resources. Thus, the use

Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications, volume: 3, number: 3, pp. 4-20
*Corresponding author: Department of Information Engineering and Computer Science, Feng Chia University, Taichung 407, Taiwan, Tel: +886-2451-7250(2090), Fax: +886-2451-6101, Email: crdow@fcu.edu.tw

of geographical information and road information can help establish a fixed backbone structure. Examples of this approach include RailRaod [1] and LBDD [5]. The advantages of the fixed route of public transportation systems should be included in design considerations.

Inspired by the concept of data circulation and virtual infrastructure, this study designs and develops Omnibone, an efficient VANET service data circulation and discovery scheme that uses public transportation systems to establish a backbone structure for data circulation. Omnibone constructs a virtual infrastructure along bus routes to restrict the circulation region and avoid redundant transmission. The grid architecture and grid header election schemes are used to create a backbone structure. In the Grid header election scheme, a vehicle with the longest stay duration is elected as a header in its grid. The grid header collects the information around the grid. When the grid header election completes, each grid header will exchange information and hold connection with other grid headers. Once established, buses, stops, and/or some vehicles elected as grid headers serve as conduct nodes, allowing mobile nodes to disseminate packets on the backbone. Data is collected and carried forward by buses as they travel, thus achieving data circulation. Stops are useful for indexing the amount of data on each bus, generating circulation metadata so that other buses/users can determine the existence and whereabouts of data chucks without actually accessing it. The proposed method enables effective data transmission and reception. The grid structure is used to limit the broadcast packets in the grid, and decrease the network load. The grid header collects metadata from other grid headers. This metadata can be used to provide discovery services for vehicles. This scheme avoids redundant transmission of user requests. This study also designs corresponding management methods to improve overall system performance.

The rest of this paper is organized as follows. Section 2 discusses related work, while section 3 describes data posting and discovery schemes. Section 4 presents experimental results. Finally, section 5 draws conclusions.

2 Related Work

This section describes related work, including the backbone-based approaches, data ferry, data circulation, rendezvous-based approaches, and service discovery schemes.

The backbone-based approaches use self-organization schemes to build a virtual structure [4] (such as a cluster, backbone, dominating set) over the physical network to assist the process of data posting. The DDB self-organization approach [6] maintains a dynamic and distributed communication architecture. In this scheme, each source node can be a leader, member, or gateway. The leader node ensures that all the members of the leadership communicate with other leaders of the nodes through the gateway link, while sink nodes send queries to the leader node and along the backbone to find the data. HCDD [7] is a hierarchical clustering scheme for data dissemination. Each clusterhead is combined into a cluster, sink query to the nearest clusterhead. The HCDD clusterhead notifies all the cluster heads, and finds the required information.

Traditional MANET [8] broadcast methods can be used to avoid redundant broadcast packets for each node, reducing the number of broadcasts. However, these dissemination schemes cannot be directly applied due to the ever-shifting nature of VANETs. Researchers have proposed various data dissemination protocols for VANET environments [4, 9, 7, 6, 10, 11, 12]. Some protocols avoid broadcast packets for each node, and may cause broadcast storm problems that decrease network performance. However, if there are many location-based services for users to find service information, users must re-send packets to a location and may cause a very high overhead. Hamida et al. [13] explored the possibility of using several WSN data dissemination protocols, and the use of virtual infrastructures for data dissemination. Since these approaches are based on fixed environments, they cannot be applied to VANETs directly.

Researchers have also proposed data dissemination protocols [14] for VANET environments that avoid broadcast storm problems and improve the network performance. However, these methods produce network overhead as the number of users increases. The Anycast-K [9] service discovery scheme can be used to find services, and is based on a virtual backbone to reduce unnecessary message transmissions. A hybrid approach for location-based service discovery scheme [15] is based on geocast addressing, and combines the advantages of proactive and reactive strategies. HarpiaGrid [16] is a geography-aware, grid-based routing VANET protocol that uses map data to generate a shortest transmission grid route. This approach effectively trades route discovery communication overhead with insignificant computation time. GPSR [17] makes greedy forwarding decisions using only information about neighbors in the network topology.

3 Our Proposed Schemes

This section describes the assumptions of the proposed framework, backbone establishment and maintenance, data posting, data circulation and service discovery schemes.

3.1 Network Environment Assumptions

The proposed framework adopts the following assumptions. A backbone $B = \{E,V,Vb,S,L,I,X\}$ is used to manage the service information. $E = \{e_1, e_2, ..., e_l\}$ refers to the set of full service information, and $e_i = \{t, p, d, m\}$, where t is the service type, p is the position of a service, d is the service descriptions, and m is the messages announced by the service owner. $V = \{v_1, v_2, ..., v_m\}$ refers to the set of vehicles in a backbone. These vehicles are equipped with a GPS and wireless transmission-enabled on-board unit (OBU). $Vb = \{vb_1, vb_2, ..., vb_n\}$ refers to buses in a backbone. An OBU and a GPS are equipped on each $vb_i \in Vb$, and bus Vb can process the complete services to metadata I, where $I = \{i_1, i_2, ..., i_y\}$ is a collection of service metadata i_y, $i_y = \{t, p\}$ which is published by the service owner. $S = \{s_1, s_2, ..., s_w\}$ refers to the set of bus stops in a backbone. Each $s_i \in S$ can receive service information from any $vb_i \in Vb$, and each s_i is included in one or more bus line L. $L = \{l_1, l_2, ..., l_x\}$ refers to the set of bus lines in a backbone, where $l_i \in L$ is an ordered subset of bus stop S. Moreover, metadata I is a subset of E. $X = \{x_1, x_2, ..., x_z\}$ refers to the set of centers, where x_i provides at least one metadata of $e_i \in E$.

Assume that bus Vb consists of storage devices but bus stop S does not. Moreover, bus stop S acts as a getaway to maintain the communication between bus Vb and center X. Figure 1 illustrates the system architecture. First, bus Vb collects full data E along bus line L and generates metadata I and sends it to bus stop S. Then bus stop S receives metadata I and forwards it to center X through the wired networks. When a bus connects with center X through bus stop S, the bus will get metadata I of full data E from center X. However, the bus can also get metadata from its neighboring buses. When the bus on the border of two bus lines, the bus can collect metadata from these bus lines. Thus, metadata can be effectively exchange and used to reduce the center's bandwidth loading.

3.2 Grid Header and Coordinator Election

This study assumes that a bus route can be used to define the data exchanging region. Previous research [16] uses grid architecture to determine the route and data exchanging area to decrease the number of

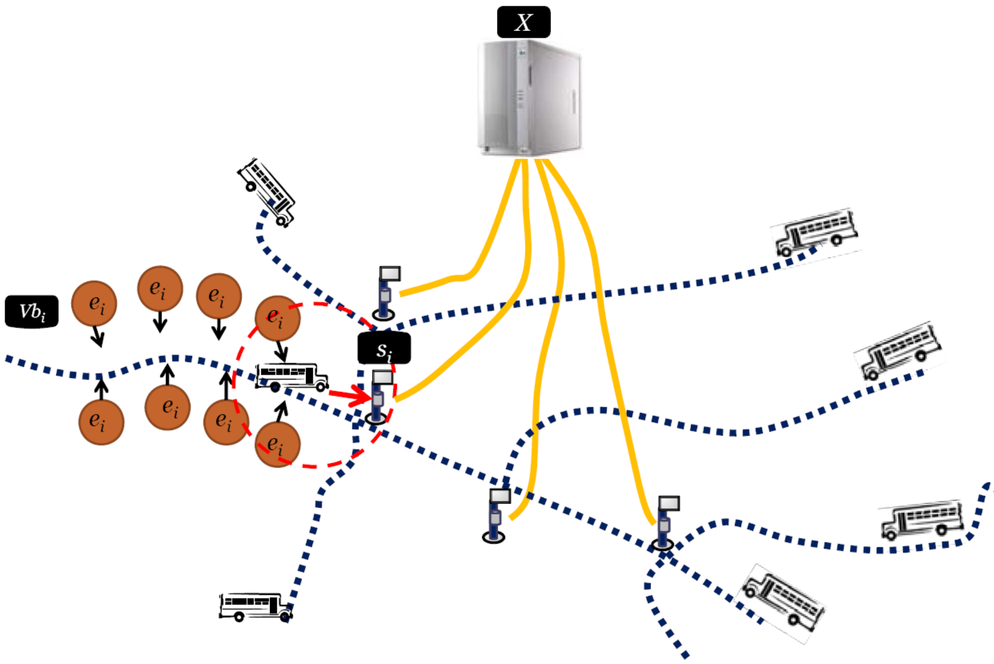


Figure 1: System Architecture

data transmissions. This study also uses the grid structure and combines it with the coordinator framework of data posting.

A grid header election scheme is used to find a potential grid header of the grid. The bus with the longest stay period should be the grid header to maintain the stability of network. The network stability of grid architecture is least change of the grid header. When the stay duration of the grid header is long, we can minimize the data hand-off time in a grid. Assume that the vehicle maintains its information, such as ID, stay duration, vehicle status, grid header, and grid ID. Moreover, each vehicle has a unique ID for identification. All vehicles periodically broadcast hello messages to notify the other of their stay duration. The stay duration estimation is

$$StayDuration(V_i) = \frac{\sum_{i=1}^{n} D_i/VSpeed_i}{n} \tag{1}$$

where D_i is the distance between the current position and the departure point, and n is the number of departure points. Departure points are road exits for vehicles to leave a grid. $VSpeed_i$ is the vehicle speed on the road, as most cities use speed limits to regulate the speed of vehicles. On the other hand, if the speed is zero, the node type may be an idle speed vehicle or an infrastructure. In this case, we do not compute equation (1), and we will directly set $1/StayDuration$ to zero in equation (2).

The status denotes a vehicle's role. There are three types of status: N, H, and M. N denotes that the role of a node is non-determined. H denotes that a vehicle is a grid header, and M denotes a grid member. $Weight(V_i)$ is a function that returns a weight value of vehicle V_i. The weight value is combined with vehicle type ($VType$), stay duration and vehicle ID (VID). $VType$ is used to denote the vehicle type. The suitable vehicle has a smaller type number. $1/saty$ duration is used to find the vehicle with the longest stay duration. When $VType$ and $1/satyduration$ are the same, the lowest VID is used to identify the priority of vehicles. The weight value of a vehicle is

$$Weight(V_i) = \frac{1}{StayDuration} + \frac{VID}{I+1} + VType_i \tag{2}$$

where VID_i is the vehicle's ID number. I is the number of vehicles in the network. The middle part of the weight definition makes each weight value unique. $VType_i$ is a type value. For example, if $VType_i$ =0, the node is a bus stop. If $VType_i$ =1, the node is a bus. If $VType_i$ =2, the node is a passenger vehicle. Moreover, $GridMember(V_i)$ is a function that returns the neighbors of V_i in the same grid. $MinWeightNode(V)$ is a function that returns a minimum weight neighbor of the vehicle set, where V is the set of the neighbors of V_i. Since the grid header election scheme is event-driven, a specific procedure will be executed at a vehicle depending on the receipt of corresponding events. The following discussion describes the four event handling routines in this algorithm. (1) When a vehicle is in the initialization state or has just entered the grid, the onInit() routine will be executed. In both conditions, the role of the vehicle is set to N. (2) When a vehicle does not receive any hello packet from its grid header after several hello intervals, the vehicle sends a re-election message to its neighbors. Then, it sets its role to N and executes onInit() shown in Figure 2. (3) When the grid header sends the re-election message before it leaves the grid, the grid header sets its role to N and executes onInit(). (4) When a vehicle receives a re-election message, it sets its role to N and executes onInit().

For the onInit() algorithm, only the vehicle which does not determine its role can execute this algorithm. Lines 2$\sim$6 illustrate that if a grid header exists, the vehicles will be its member nodes. In line 7, $MinWeightNode(V)$ is used to locate a vehicle with minimum weight value in its located grid. Lines 8$\sim$14 denote that if vehicle V has minimum weight of its neighbors, it will become a grid header.

```
    Algorithm 1 : Grid header election
1   Procedure onInit()
2   If ∃Y, Y∈ GridMember(V) and Y.Status = H
3       V.Status := M
4       V.Gridhead := Y
5       Exit()
6   End If
7   W := MinWeightNode(GridMember(V))
8   If V=W
9       V.Status := H
10      V.Gridhead := V
11  Else
12      V.Status := M
13      V.Gridhead := W
14  End If
15  End Procedure
```

Figure 2: Grid Header Election Algorithm

3.3 Omnibone Establishment and Maintenance

The grid architecture and the coordinator scheme are used to create a backbone structure. As shown in Figure 3, the coordinator is selected from the grid header that has degree>2. The coordinator must build links when the grid header election completes. The coordinators in the virtual backbone consist of mobile and fixed ones. A mobile coordinator collects the data and metadata of other coordinators and transmits metadata to other coordinators. For example, a bus is a mobile coordinator that can replicate service data and carry it around the backbone. A fixed coordinator collects metadata from other coordinators and transmits metadata to other coordinators. For example, a fixed coordinator can be a bus stop

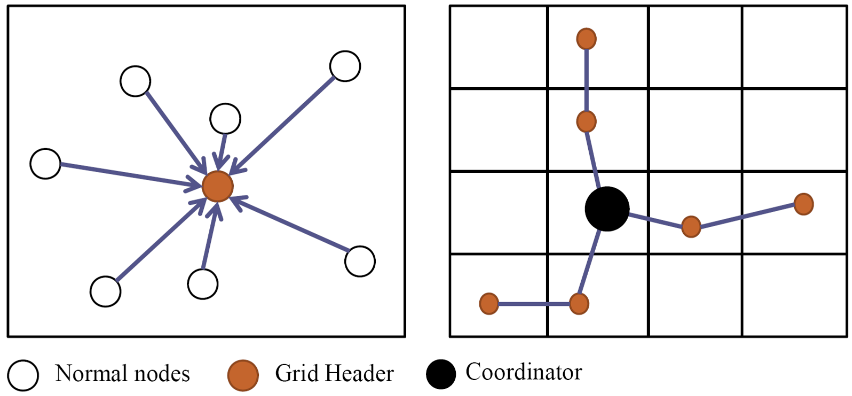


Figure 3: Grid Header and Coordinator Election

or roadside unit with communication capabilities. Unlike mobile coordinators, fixed coordinators may not carry the full service information along the bus route, but they can serve as access points to pass the service information to a service center.

Moreover, to establish the backbone between the coordinators, each selected coordinator compares its stay duration with others in the grid. If two nodes have the same weight value, the node with the lowest grid ID is selected as the coordinator. The link rule of the backbone is established in a clockwise direction to know the next transmission grid. If there is a communication gap, the grid header sends a query message to its neighboring grids. If the neighboring grid receives query messages from two grids, it becomes a temporary backbone grid. It will continue to query until the resumption of the neighboring grid. If no vehicle can be found, the local recovery mechanism is triggered to avoid packet re-routing. If even the local recovery mechanism fails due to a large topology gap, the backtracking method can be used to overcome any gap problems [16]. It is not necessary for the source node to re-transmit packets, thus reducing latency and packet lost.

3.4 Information Posting Scheme

Data posting often relies on broadcast and multicast schemes to publish information, but their methods may incur a large overhead. To circulate service information effectively, this study uses a map to create a backbone structure.

The data posting route includes all members of public transportation systems, and this study uses bus routes as the data posting routes. For example, the bus transportation system of Taichung City, Taiwan is used to disseminate data. The statistics of bus stops along the bus route are used to calculate the standard deviation. The backbone establishment is based on the calculated standard deviation σ, and the mean number of stations. The grid selection is based on a selection threshold $S_t = meannumberofstations + k\sigma$, where k is a user-defined multiplier. If a grid covers a road segment for which the number of bus stations of the road is greater than S_t, it will be a part of the backbone. This virtual backbone structure can help avoid data dissemination to the entire network.

3.5 Information Circulation Scheme

The data circulation mechanism is used to disseminate the service information to the entire backbone structure. When disseminating service information, routing protocols are used by the nearest coordi-

```
Algorithm 2 : Data Posting and Circulation
Procedure DDInit
    For each vb_i do
        CollectServices(vb_i,e_i) along l_i
        vb_i generates I
        CenterConnection(vb_i, s_i, X)
        Forward I to s_i
        For vb_i encountering vb_i-1 do
        If (e_i =complete data ) then
            vb_i generates I
            List(vb_i,e_i) and send I to vb_i-1
        Else
            Exit()
        For vb_i encountering v_i do
            Wait()
        If vb_i on l_i
            CenterConnection(vb_i, s_i, X)
            Response (I)
            Exit()
    End
End Procedure
```

Figure 4: Data Dissemination Algorithm

nators on the backbone, which may be the mobile coordinator for buses Vb or a fixed coordinator for stops S as well as the forwarding coordinator for normal vehicles V. The algorithm is shown in Figure 4. When a service e_i posts data to a bus vb_i, the full data can be collected by vb_i and its metadata can be forwarded to other s_i. Because Vb moves along the bus routes, the bus routes can collect the full data on buses and forward the metadata to other bus stops. The difference is that the fixed coordinators may not carry the full service information along the bus route due to the limitation of its system resources. However, S provides a stability platform and they can disseminate data to the coordinators. When data is disseminated to the backbone, the data is circulated through coordinators in the grid and the selected coordinator forwards data to other coordinators. S and Vb are used to send and receive packets in the virtual backbone in order to increase transmission efficiency and reduce transmission cost.

The following shows three main functions used for posting and circulation;

- `CollectServices()` : collects ei to forward full data or List(vb_i, e_i) to the next hop vb_i-1. In the beginning, each e_i is disseminated to L and vb_i would collect E along l_i.
- `List()` : generates I from vb_i and each vb_i can process e_i to i_i. When vb_i encounters s_i, vb_i will send message I to s_i and then s_i receives I and forwards it to X through the wired networks.
- `CenterConnection()` : connects between vb_i and X through s_i. vb_i obtains I (service metadata) stored in other vb_i-1 through s_i. However, each vb_i on the L has E (full data) along the passing l_i and has I from s_i which has service metadata of other l_i-1.

3.6 Information Discovery Scheme

When users need services, they may query the virtual backbone to find the nearest service. The exploration and acquisition are the last two phases. If the required message is in the coordinator of the first

```
   Algorithm 3 : Data Exploration and Acquisition
1  Procedure SDInit
2      For each vb_i do
3          If(vb_i receives Request(v,e_i) from vb_i and e_i ∈ vb)
4              Response(v,e_i)
5          Else
6              vb_i sends query to neighbor grids
7              If (neighbor vb_i has e_i)
8                  Forward Request(v,e_i) to vb_i
9              Else
10                 CenterConnection(vb_i, s_i, X)
11                 vb_i sends query to s_i
12                 If (center X has e_i)
13                     Forward Request(v,e_i) to X
14                 Else
15                     Forward Request(v,e_i) to source location of e_i
16                 End If
17             End If
18         End If
19         If(vb_i receives a query from vb_i)
20             Response (I, vb_i)
21         End If
22     Next
23 End Procedure
```

Figure 5: Services Discovery Algorithm

query, it can immediately acquire the required information. If the required information is in its neighbor coordinators, the required message is forwarded to neighbor coordinators and explores the required information. Another case is to query general vehicles because they do not store messages and forward messages to the neighboring grid. When requesting the required information, the data is forwarded through the coordinator.

Figure 5 shows the service discovery algorithm. When a user wants to request data, SDInit will be executed. The exploration message is sent to its grid header. Lines 3∼4 illustrate that if a grid header receive the request and it has the full data, that full data will be sent to the user. Lines 5∼11 illustrate that if the grid header does not have full data, a query message will be sent to its neighboring grids. When the grid header receives the query result, and one of its neighboring grids has full data, the user request will be sent to that grid. Otherwise, a query message will be sent to the center. Lines 12∼16 illustrate that if the center has full data, the user request will be sent to the center to get the full data. The worst case is that if the user cannot find full data from any grid header or the center; in this case, the request will be sent to its source location.

The service also provides an advertisement received by the coordinator that has the full data. If there are emergency, traffic and other text messages, due to a smaller amount of text data, users can ask the coordinator and exchange among various coordinator points. The coordinator can obtain information on a smaller amount of data rather than the location of data sources used to acquire and efficiently find the necessary services and to reduce the required time.

In practical applications, our backbone structure is built on the major route of transportation and covering most of the points of interest in the urban environment. The backbone can transmit the traffic information, traffic accident notifications, business advertisement, and other information. For example, a department store wants to promote the commodity. The department store creates coupon information and sends it to the grid header. The grid header stores the coupon information and transmits the metadata data to the center through the infrastructure. When a user requests the coupon information, the request is received by the nearest grid header of the backbone through multihop routing. If the grid header has the coupon information, the user can get it directly. If there is no information, the grid header will first send request metadata to neighboring grid headers to find the coupon information. If the coupon information is not found, the grid header will request metadata from the center. Finally, the user can get the full coupon data from the backbone.

4 Experimental Results

Experiments were conducted to evaluate the efficiency of the Omnibone scheme. This section summarizes the experimental results, including the comparisons of different routing protocols and our scheme.

4.1 Simulation Design

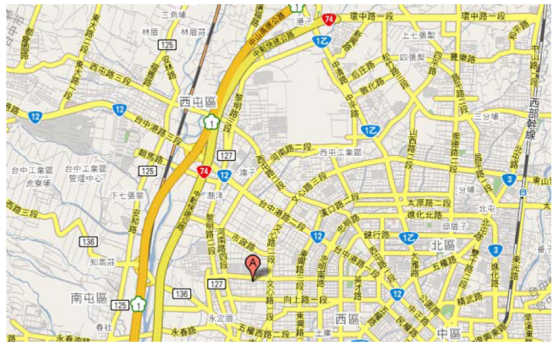
This section assesses the effectiveness of the virtual backbone, and performs experimental analysis to confirm the validity of the proposed scheme. The experimental evaluation uses the NS-2 [18] simulation platform. The simulation traffic model was generated using SUMO [19] and the map data of Taichung City (Figure 6 (a)). Latitude and longitude coordinates were obtained by Google Maps [20], and the SUMO traffic simulator was then used to generate traffic and road information. As shown in Figure 6 (b), five POIs (Point of Interest), including Feng Chia Shopping Town, Yizhong Shopping Mall, Electronics Bazaar, Sogo Department Store, and Shin Kong Mitsukoshi Department Store are located in our simulation environment.

4.2 Performance Evaluation of Omnibone Scheme

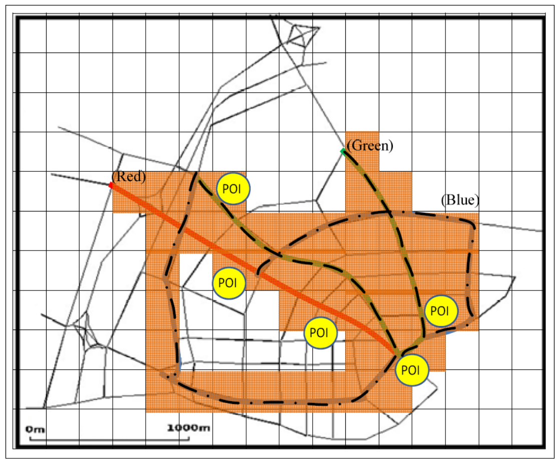
Figure 7 summarizes the parameters for the simulation environment for the Omnibone scheme experiments. A node was randomly selected as a service provider or a sender.

These experiments compare the proposed scheme with the HarpiaGrid and GPSR routing protocols. This study evaluates four important performance metrics:

- `Packet delivery ratio` : defined as the total reply services to a sender over the total request services. A higher value for this metric is better.
- `End-to-end delay` : representing the average time from the sending request packet to receiving reply packets per request.
- `Hop count` : defined as the distance from user node to the source.



(a)



(b)

Figure 6: Map Data and SUMO Traffic Simulation Generate for Simulation

Parameters	**Value**
Simulation Time	200s
Map Size	Google MAP 3150m×2200m
Mobility Model	Random
Vehicles Speed	30~50 km/h
Vehicles Number	200
Number of Buses	4~40
Bus Speed	40km/h
Buses Headway	15s
Number of Bus Stops	70
Transmission Range	250m
CBR Rate	600 Kbits/sec
Data Packet Size	512 bytes

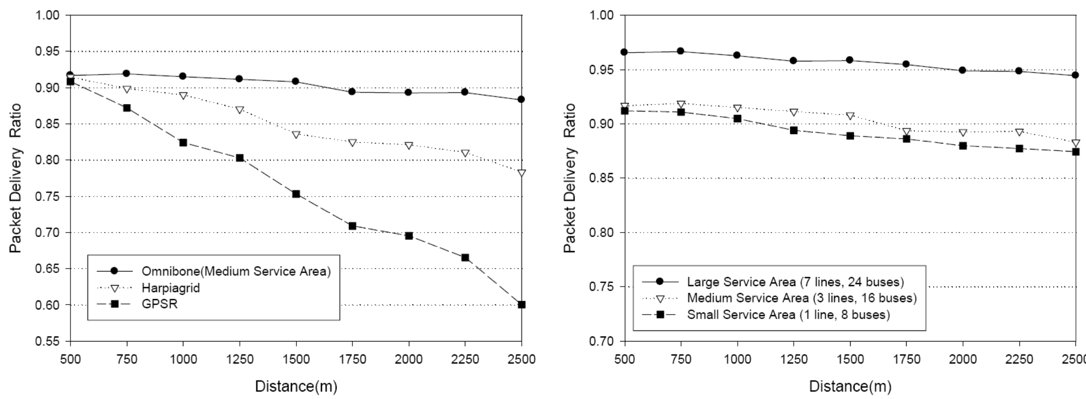
Figure 7: Simulation Parameters for the Omnibone Scheme

- `Normalized overhead` : defined as the total control messages over the delivery ratio. A lower value for this metric is better.

4.3 Simulation Results

Experiments were conducted to evaluate the efficiency of the proposed scheme. This section summarizes the experimental results, including the comparisons of different routing protocols with the proposed scheme.

4.3.1 Packet Delivery Ratio



(a) (b)

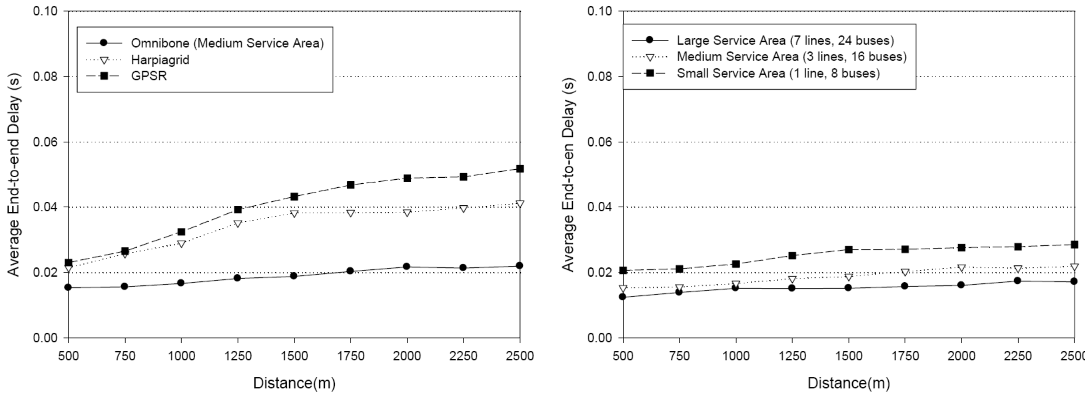
Figure 8: Packet Delivery Ratio vs. Distance for Routing Protocols and Service Areas

Figure 8 (a) shows how distance influences the packet delivery ratio. As the distance increases from 500m to 2500m, the packet delivery ratio decreases because the longer distance causes a higher packet lost. Figure 8 (a) shows that Harpiagrid and GPSR are more sensitive to distance than Omnibone because long distance causes a high packet lost. The proposed scheme uses the backbone structure to reduce the exploration distance, providing effective data packet circulation and discovery.

The experiments in this study consider a network with different service areas. These service areas are large, medium, and small service areas. There are twenty four buses and seven bus lines in the large service area, sixteen buses and three bus lines in the medium service area, and eight buses and one bus line in the small service area. Figure 8 (b) shows that the packet delivery ratio for the large service area with 500m is 96.6%, while the packet delivery ratio for the small service area with 2500m is 87.4%. In other words, the large service area has a higher packet delivery ratio. The large service area can reduce the exploration distance. The packet delivery ratio decreases as the distance increases.

4.3.2 End-to-End Delay

Experiments of the end-to-end delay for different sizes of service areas were conducted and for different distances between the user and service location. The performance was evaluated for the distances from 500m to 2500m in our simulation environment. Figure 9 (a) shows that the end-to-end delay of the proposed scheme outperforms other protocols. This is because the discovery mechanism catches the nearest services on the backbone. The average end-to-end delay is close to 0.02 seconds. The average end-to-end delay of GPSR and Harpiagrid increases with the increasing of distance. This is because



(a) (b)

Figure 9: End-to-end Delay vs. Routing Protocols and Size of Service Areas

GPSR and Harpiagrid must take time to forward packets to destination on road segments. Figure 9 (b) shows how the different sizes of service areas influence the average end-to-end delay. The large service area performs better than the medium and small service areas due to the number of duplicates on the backbone. The large service area has more buses and bus lines than the other two service areas. Furthermore, the coverage area of routes in the large service area is wider, making it easier for users to acquire the service information.

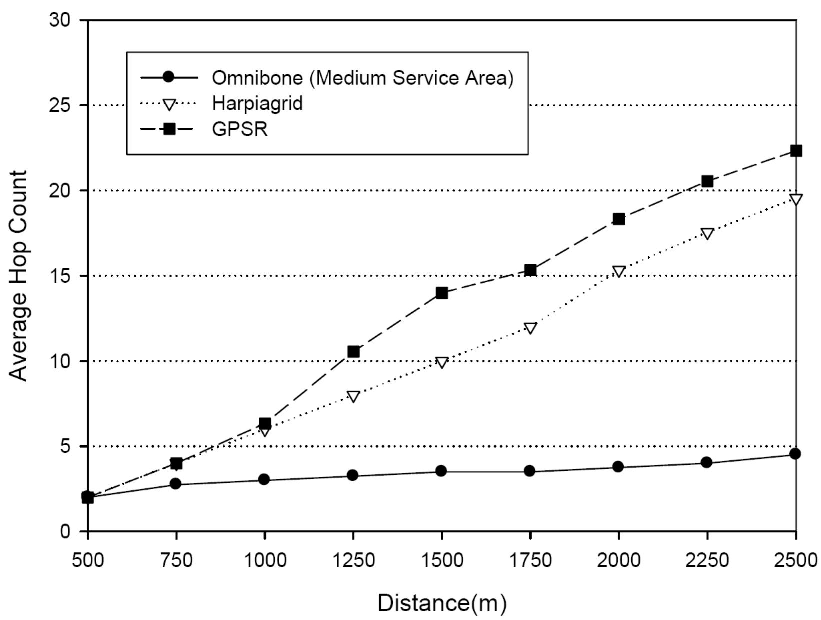


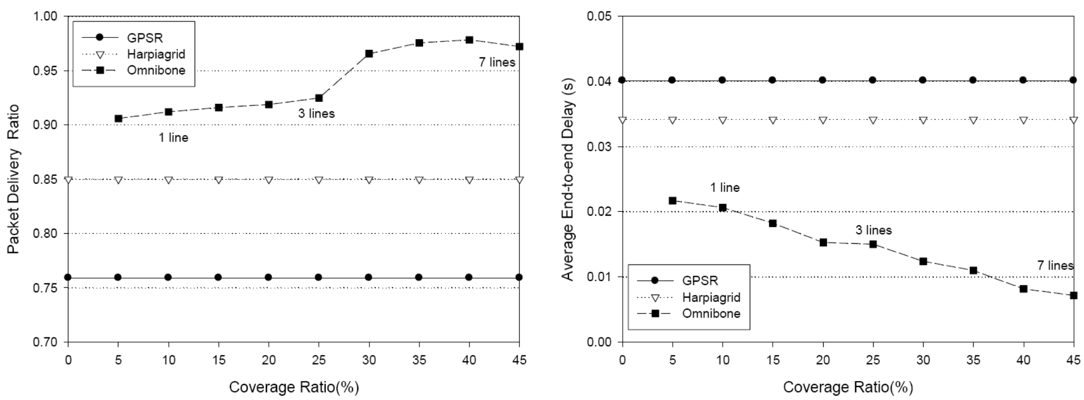
Figure 10: Average Hop Count vs. Distance for Different Routing Protocols

4.3.3 Hop Count

The average hop count is the mobile path for a source-destination session, averaged over all source-destination sessions. Figure 10 shows that the proposed scheme has shorter average hops than other protocols. There are two reasons for this result: (1) Omnibone prefers to link backbone routes over forward progress when discovering services, and (2) unlike GPSR, which consistently selects the route to destination, there are many duplicates for easier acquiring services.

4.3.4 Performance in Different Coverage Ratio

This section increases the number of buses and study how many buses can increase the packet delivery ratio or decrease the average end-to-end delay of Omnibone. The coverage ratio is set to 10% of one line (red route), 25% of 3 lines (red and green in Figure 6 (b)) and 45% of 7 lines (red, green, and blue), and the backbone is set as well-distributed to obtain experiment data.



(a) (b)

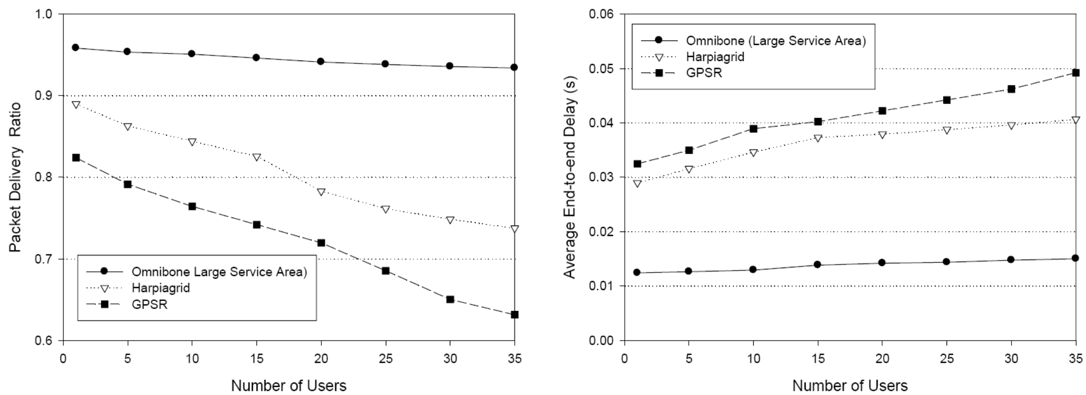
Figure 11: Packet Delivery Ratio and End-to-end Delay vs. Coverage Ratio

In Figure 11 (a), the packet delivery ratio of Omnibone increases when the coverage ratio increases from 5% to 45%. This is because more buses and bus lines on the bone structure increase the probability of a request packet finding the required information. As shown in Figure 11 (a), Omnibone outperforms Harpiagrid and GPSR because the Omnibone structure has duplicated circulation, which causes a higher probability to find required information.

As shown in Figure 11 (b), the average end-to-end delay of Omnibone at different coverage ratios is much lower than GPSR and Harpiagrid protocols. The end-to-end delay of Omnibone is lower than 0.025s from source to destination. This is because more buses in bus lines cause more duplicate circulation on the bone structure, which in turn decreases the request time of service information.

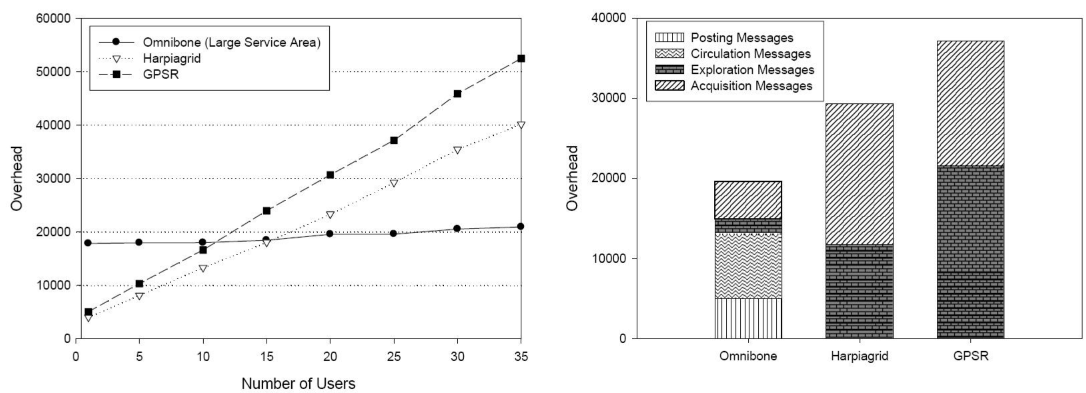
4.3.5 Performance in Different Number of Users

Figure 12 shows the performance for packet delivery ratio and average end-to-end delay for different numbers of users. When the number of users increases, the packet delivery ratio and average end-to-end delay in Omnibone scheme almost remain the same. This demonstrates that when the number of users increases, Omnibone can provide the same performance and does not increase the burden of this structure. However, the Harpiagrid and GPSR routing protocols have lower performance because more users cause more search time and delay in obtaining information.



(a) (b)

Figure 12: Packet Delivery Ratio and End-to-end Delay vs. Number of Users



(a) (b)

Figure 13: Overhead vs. Number of Users and Different Protocols

4.3.6 Overhead

The previous experiments show that there is very high performance for packet delivery ratio and small average end-to-end delay in the proposed scheme. A large number of buses and backbone routes may incur a large control overhead. However, nothing could be further from the truth is the lower control overhead of our scheme. Figure 13 (a) shows that although the overhead of the proposed scheme for few users is higher than that of the other protocols, the overhead is much lower than other protocols when the number of users increases. Figure 13 (b) demonstrates the overhead of various schemes for 25 users, in which Harpiagrid and GPSR have only exploration and acquisition overhead. The proposed scheme needs posting and circulation messages to build the backbone structure. However, when more users use the backbone, the posting and circulation messages do not need to increase. Experimental results show that this scheme can effectively increase packet delivery ratio and decrease end-to-end delay with an acceptable control overhead.

5 Conclusions

This paper proposes an efficient service circulation and discovery scheme that uses public transportation systems to establish a virtual backbone in VANETs. This virtual backbone uses bus routes to restrict the posting region and avoid redundant transmissions. This work also presents a grid header election scheme to establish and maintain the backbone. Experimental results show that the Omnibone scheme has higher performance for packet delivery ratio and end-to-end delay than other protocols. Furthermore, the proposed scheme outperforms other schemes in terms of the control overhead when more people share the backbone for data discovery. In the future, we will investigate techniques to use this backbone structure to achieve greater sharing of services and applications, such as the P2P (Peer-to-Peer) applications in a VANET distributed shared service system.

6 Acknowledgments

The authors would like to thank the National Science Council of the Republic of China for financially supporting this research under Contract No. NSC99-2221-E-035-045-MY2.

References

[1] J. H. Shin, J. Kim, K. Park, and D. Park, "RailRoad: Virtual Infrastructure for Data Dissemination in Wireless Sensor Networks," in *Proc. of the 2nd ACM International Workshop on Modeling Analysis and Simulation of Wireless and Mobile Systems (PE-WASUN'05), Montréal, Québec, Canada*. ACM, October 2005, pp. 168–174.

[2] S. Teshima, T. Ohta, E. Kohno, and Y. Kakuda, "A Data Transfer Scheme Using Autonomous Clustering in VANETs Environment," in *Proc. of the 10th International Symposium on Autonomous Decentralized Systems (ISADS'11), Kobe, Japan*. IEEE, June-July 2011, pp. 477–482.

[3] L. Wischhof, A. Ebner, and H. Rohling, "Information Dissemination in Self-organizing Intervehicle Networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 6, no. 1, pp. 90–101, March 2005.

[4] K. Almahorg and O. Basir, "Simulation-based Performance Comparison of VANETs Backbone Formation Algorithms," in *Proc. of the 12th IEEE/ACM International Symposium on Distributed Simulation and Real-Time Applications (DS-RT'08), Vancouver, British Columbia, Canada*. IEEE, October 2008, pp. 236–242.

[5] E. B. Hamida and G. Chelius, "A Line-based Data Dissemination Protocol for Wireless Sensor Networks with Mobile Sink," in *IProc. of the IEEE International Conference on Communications (ICC'08), Beijing, China*. IEEE, May 2008, pp. 2201–2205.

[6] J. L. Lu and F. Valois, "On the Data Dissemination in WSNs," in *Proc. of the 3rd International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob'07), White Plains, New York, USA*. IEEE, October 2007, pp. 58–58.

[7] C. J. Lin, P. L. Chou, and C. F. Chou, "HCDD: Hierarchical Cluster-based Data Dissemination in Wireless Sensor Networks with Mobile Sink," in *Proc. of the 2006 International Conference on Wireless Communications and Mobile Computing (IWCMC'06), Vancouver, British Columbia*. ACM, July 2006, pp. 1189–1194.

[8] H. Alshaer and E. Horlait, "An Optimized Adaptive Broadcast Scheme for Inter-vehicle Communication," in *Proc. of the 61st Vehicular Technology Conference (VTC'05-Spring), Clarion Hotel, Stockholm, Sweden*, vol. 5. IEEE, May-June 2005, pp. 2840–2844.

[9] C. R. Dow, P. J. Lin, S. C. Chen, and S. F. Hwang, "An Efficient Anycast Scheme for Discovering K Services in Cluster-based Mobile Ad-hoc Networks," *Wireless Communications and Mobile Computing*, vol. 11, no. 9, pp. 1287—1301, September 2011.

[10] R. Panayappan, J. M. Trivedi, A. Studer, and A. Perrig, "VANET-based Approach for Parking Space Availability," in *Proc. of the 4th ACM International Workshop on Vehicular Ad-Hoc Networks (VANET'07), Montréal, Québec, Canada*. ACM, September 2007, pp. 75–76.

[11] Y. Y. Sung and M. J. Lee, "Road-based broadcast message dissemination approach in VANETs," in *Proc. of the 2011 International Conference on Information Networking, Kuala Lumpur, Malaysia*. IEEE, January 2011, pp. 143–147.

[12] D. Yong and X. Li, "SADV: Static-node-assisted Adaptive Data Dissemination in Vehicular Networks," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 5, pp. 2445–2455, June 2010.

[13] E. B. Hamida and G. Chelius, "Strategies for Data Dissemination to Mobile Sinks in Wireless Sensor Networks," *IEEE Wireless Communications*, vol. 15, no. 6, pp. 31–37, December 2008.

[14] T. Camp and B. Williams, "Comparison of Broadcasting Techniques for Mobile Ad-Hoc Networks," in *Proc. of the 3rd ACM International Symposium on Mobile Ad-Hoc Networking and Computing (MobiHoc'02), Lausanne, Switzerland*. ACM, June 2002, pp. 194–205.

[15] N. Klimin, W. Enkelmann, H. Karl, and A. Wolisz, "A Hybrid Approach for Location-based Service Discovery in Vehicular Ad-Hoc Networks," in *Proc. of the 1st International Workshop on Intelligent Transportation (WIT'04), Hamburg, Germany*, March 2004.

[16] K. H. Chen, C. R. Dow, and Y. S. Lee, "HarpiaGrid: A Reliable Grid-based Routing Protocol for Vehicular Ad-Hoc Networks," *Journal of Information Science and Engineering*, vol. 26, no. 3, pp. 817–832, May 2010.

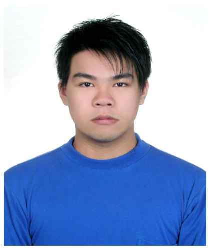
[17] B. Karp and H. T. Kung, "GPSR: Greedy Perimeter Stateless Routing for Wireless Networks," in *Proc. of the 6th Annual ACM/ IEEE International Conference on Mobile Computing and Networking (MobiCom'00), Boston, Massachusetts, USA*. ACM, August 2000, pp. 243–254.

[18] "The ns-2 network simulator," http://www.isi.edu/nsnam/ns.

[19] "SUMO-Simulation of Urban Mobility," http://sumo.sourceforge.net/index.shtml.

[20] "Google Map Server Browser," http://maps.google.com.

Chyi-Ren Dow was born in 1962. He received the B.S. and M.S. degrees in information engineering from National Chiao Tung University, Taiwan, in 1984 and 1988, respectively, and the M.S. and Ph.D. degrees in computer science from the University of Pittsburgh, USA, in 1992 and 1994, respectively. Currently, he is a Professor in the Department of Information Engineering and Computer Science, Feng Chia University, Taiwan. His research interests include mobile computing, ad-hoc wireless networks, agent techniques, fault tolerance, and learning technology.



Yu-Hong Lee was born in Miaoli, Taiwan, in 1985. He received the M.Sc. degrees in Institute of Applied Information Technology from Ling Tung University, Taiwan, in 2009. Currently, he is a candidate for the Ph.D. degree in information engineering from Feng Chia University, Taiwan. His research interests include vehicular ad-hoc networks, telematics, and sensor web techniques.

Pa Hsuan was born in 1981. He received the B.S. and M.S. degrees in information engineering from Feng Chia University, Taiwan, in 2003 and 2005, respectively. He is currently a candidate for the Ph.D. degree in the Department of Information Engineering and Computer Science, Feng Chia University, Taiwan. His research interests include ad-hoc wireless networks, anycasting, operating systems, and embedded systems.

Yi-Tung Lee was born in Taichung, Taiwan, in 1983. He received the M.Sc. degrees in Information Engineering and Computer Science from Feng Chia University, Taiwan, in 2010. His research interests include vehicular ad-hoc networks, telematics.

Shiow-Fen Hwang was born in 1963. She received the B.S., M.S. and Ph.D. degrees in Applied Mathematics from National Chiao Tung University, Taiwan, in 1985, 1987 and 1991, respectively. Currently, she is an Associate Professor in the Department of Information Engineering and Computer Science, Feng Chia University, Taiwan. Her research interests include interconnection networks, mobile computing, and computer algorithms.