Developing Event Based Middleware For Information Dissemination in Vehicular Ad-hoc Networks

A Thesis Submitted

in Partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy

in

Computer Science and Engineering

by

Tulika (Reg. No. 90703506)

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT

Thapar University

Patiala

November, 2013

CERTIFICATE

Certified that the work contained in the thesis titled "*Developing Event Based Middleware For Information Dissemination in Vehicular Ad-hoc Networks*", by *Tulika (Reg. No. 90703506)*, has been carried out under our supervision and that this work has not been submitted elsewhere for a degree.

(Dr. Deepak Garg)
Dept. of Computer Science & Engg.
Thapar University Patiala

(Prof. M.M. Gore)
Dept. of Computer Science & Engg.
Motilal Nehru National Institute of Technology Allahabad

November, 2013

Papers Published

- "A publish/subscribe communication infrastructure for VANET applications". In IEEE advanced information networking and applications (WAINA) workshops (pp. 442-446), March 22-25, 2011.

- "Publish/subscribe based information dissemination over VANET utilizing DHT", Journal of Frontier of Computer Science (Springer), Vol 6, pages 713 to 724(2012)
  Indexed in: Science Citation Index Expanded
  Impact Factor:0.298

- "Structured P2P Overlay of Mobile Brokers for Realizing Publish/Subscribe Communication in VANET", The Scientific World Journal, vol. 2014, Article ID 136365, 12 pages, 2014. doi:10.1155/2014/136365
  Indexed in: Science Citation Index Expanded
  Impact Factor: 1.7

Abstract

VANET (Vehicular Adhoc Network) can be defined as a distributed, self organizing communication network of moving vehicles and stationary road side info-stations. These vehicles are equipped with radio interfaces using which they are able to communicate with any existing infrastructure or among themselves. VANETs can be utilized to disseminate important information items which can assist in providing safe and comfortable driving experience.

Efficient and scalable information dissemination remains a major challenge in VANET as communicating nodes may dynamically leave or join the network and availability of any particular node cannot be guaranteed at any given time. There is need of a middleware that can provide asynchronous and decoupled communication mechanisms for robust application development.

In this thesis, we propose our approach for developing information dissemination middleware which utilizes publish/subscribe communication paradigm over structured P2P (Peer to Peer) overlay networks. Publish/subscribe communication paradigm is an attractive alternative for designing information dissemination applications in distributed and dynamic environment like VANET. This paradigm provides decoupling in time, space and synchronization between information producers, called publishers and information consumers, called subscribers.

Structured P2P networks are based on Distributed Hash Tables (DHT) and

provide high scalability, fault tolerance and self organization. While realizing publish/subscribe communication over DHTs, participating peers cooperate in routing and storing publications and subscriptions. It is ensured that matching publications and subscriptions meet in at least one peer node, termed as rendezvous node, which in turn notifies the subscriber.

Our approach has been guided by simulation studies performed to understand the trade-offs related to availability of infrastructure and traffic conditions. Depending upon the number of available road side info-stations, three designs have been proposed. In the first design, publish/subscribe framework is implemented over DHT of info-stations. Further, it is also investigated that even if info-stations are not connected to internet, the DHTs formed among them can provide acceptable performance. In second, DHT of city-buses run by public transport is utilized and in third, a 2-tier DHT is proposed which is the hybrid of the first and second design. These designs have been evaluated under different traffic conditions prevailing in urban and semi-urban area and under uniform and skewed vehicle distribution across the roads.

Acknowledgement

First and foremost, I bow my head humbly before the Almighty God for making me capable of completing my Ph.D. Thesis. With His blessings only I have been able to accomplish this challenge.

My first debt of gratitude must go to my supervisors Dr. Deepak Garg and Prof. M.M.Gore. Dr. Deepak Garg patiently provided the vision, encouragement and counseling necessary for me to proceed through the doctoral program and complete my dissertation. He was instrumental in helping me cranked out this thesis.

I express deep sense of gratitude and thankfulness to Prof. M.M. Gore who has helped me throughout my research work with patience and enthusiasm. I am much indebted to him for his inspiring guidance, affection and generosity throughout the tenure of my research work.

I am grateful to Dr. Seema Bawa and Dr. Maninder Singh, former and current Heads of Department of Computer Science and Engineering, Thapar University, Patiala for their constant support, wishes and motivation.

I am extremely grateful to the members of Doctoral Committee Dr. Maninder Singh (Chairman), Dr. Anil Kumar Verma and Dr. R.K. Sharma for their encouragement, insightful comments and valuable suggestions during these last five years.

I am thankful to honorable Vice Chancellor of Sam Higginbottom Institute of

Agriculture, Technology and Sciences, Rev. Prof. Rajendra B. Lal for giving me permission to pursue my research at Thapar University, Patiala. I would also like to thank Prof (Col.) Gurmit Singh, former Head of the Department of Computer Science and Information Technology for his genuine support and providing me necessary infrastructure and resources to accomplish my research work.

I owe a lot to my husband Dr. Mayank Pandey for always inspiring me with his passion for research. He was always there for guiding me throughout this long journey in each and every step. He has taught me the methodology to carry out the research and to present the research works as clearly as possible. It was a great privilege and honor to work and study under his guidance. I am extremely grateful for what he has offered me. I feel fortunate to have him in my life.

I am deeply indebted to my mother and father for their love, prayers, care and sacrifices for educating and preparing me for my future. Big thanks are due to my father who took so much pain to escort me from Faridabad to Patiala many times.

My mother-in-law and father-in-law deserve special mention for their desire to see me reach new heights in academics. I thank them for their unconditional support and prayers. Also, I express my thanks to my sister, brother, sister-in-law and brother-in-law for their constant encouragement, without which I could not have been succeeded.

I was blessed with my lovely daughter Sona during this research work. Special thanks for her whose innocent mischieves and heart stealing smile have been a source of motivation and a great stress buster for me.

Last but not the least, I wish to thank my friends and colleagues from the Department of Computer Science and Information Technology, SHIATS, Allahabad. I would specially like to thank Ms. Mudita Srivastava for her guidance and support during the hard time.

This Dissertation is Dedicated

To my beloved husband Mayank for being my pillar of strength and to my sweet daughter Sona for bringing so much bliss in my life.

Contents

List of Figures

List of Algorithms

Chapter 1

Introduction

Millions of people around the world die/get injured every year because of road accidents. A report published by Ministry of Road Transport and Highways (MORTH) [6] suggests that in India alone there are around 400,000 road accidents with 90,000 fatal accidents. Many people get stuck in traffic jam every day. Lots of time and money can be saved if drivers get information about traffic conditions beforehand. Existing Traffic Information Systems fail to provide accurate and timely information to drivers. Also, these systems need lots of infrastructure to provide any information to drivers which in turn results in high deployment and maintenance cost.

Some research efforts like DOLPHIN [83], DEMO2000 [85], FleetNet [32] have been made which is not solely dependent on infrastructure but still rely on some infrastructure to some extent. Recent researches are focusing on infrastructure less traffic information system such as CarTalk [72], CarNet [62], Network on Wheels [33] and Car2Car communication consortium [1].

It is expected that in near future all the vehicles will come equipped with wireless access point and they will communicate with each other forming a vehicular network. This type of vehicular network is termed as Vehicular Ad hoc Network (VANET) and can be considered as sub-class of Mobile Ad-hoc Network (MANET).

VANETs have been recently attracting ample attention from both research and industry communities. Recent advances in wireless communications technology and car industry made it possible to consider wireless ad-hoc networks or hybrid of wireless ad-hoc and infrastructure networks, providing connectivity among vehicles on the road. VANET is an emerging technology that can integrate ad-hoc network, wireless LAN and cellular technology to achieve communication between vehicles for increased efficiency and security.

VANETs can be defined as a distributed, self-organizing communication network of moving vehicles and stationary road side info-stations. Vehicles in VANET move along roads with different speeds, stoppage times and directions. Their movement trajectories are controlled unlike the mobile nodes in mobile ad-hoc network (MANET). However, forming a network of these moving vehicles is no less challenging than MANET as they move with different speeds which cause unpredictable changes in the network topology. These vehicles are equipped with radio interfaces through which they are able to communicate with any existing infrastructure or among themselves. Vehicles communicate directly with other vehicles within their radio range. To communicate with any info-station or other vehicles which are out of range, intermediate vehicles are used to forward the information.

Some of the distinguishing features of MANETs and VANETs are:

- **High Mobility and dynamic topology :** Because of the high speed of the vehicles in VANET, topology changes rapidly. As a result network gets partitioned more frequently.

- **Mobility pattern :** Vehicles move along some controlled trajectories (roads) which is different from MANET where mobile nodes may move in irregular manner.

- **Sufficient energy and storage :** In VANET, there is no power constraint as batteries get recharged simultaneously in moving vehicles. Nodes have ample computing power including both storage and processing since nodes are vehicles instead of small handheld devices/laptops of MANET.

- **Network density :** Density of communicating nodes is dynamic because traffic patterns in any city are dependent on time of day. Further, some areas of city may experience high traffic in comparison to other areas.

Two types of communications are possible in VANET, Vehicle to Infrastructure (V2I) communication and Vehicle to Vehicle (V2V) communication. In infrastructure based network, Road Side Units (RSU) are used which are also equipped with radio interfaces whereas in pure ad-hoc mode, communication is achieved among vehicles without using RSUs. Cellular networks such as 2G/3G/4G can serve as alternatives for information dissemination among vehicles. However, scarcity of bandwidth is being experienced more frequently as the number of subscribers grows rapidly. Taking this issue into account, the 802.11 working group of IEEE is standardizing 802.11p also known as Dedicated Short Range Communication (DSRC) [47] for vehicular communication. Major players in the automotive industry such as BMW, Toyota, and Nissan have already begun producing vehicles equipped with short range communication technologies.

Further, experiments conducted in [38] suggest that a vehicle traveling past a roadside access point with 802.11 a/b/g wireless interface at 80 km/h is able to transfer up to 50 MB of data, the equivalent of approximately 20 songs, 15 minutes of low quality video or 25 high quality digital photographs. This indicates that even if constant connectivity is not present and frequent changes in topology are possible, the data rate achieved is enough for common applications. The above arguments indicate that in near future short range communications between vehicles can be realized and network applications will be built over them.

Also, some interesting approaches have been proposed where RFID technology [69] [68] is utilized for locating a vehicle or traffic signal identification [49]. RFID assisted navigation system has also been proposed in [23]. The advantage of this option is that RFID do not require a power source which reduces the maintenance and thus the global cost of the system. In vehicular applications, RFID is typically used in a scenario where the tag is located on the vehicle and the reader is placed on the road. This research development unleashes a possibility where in near future technologies like 802.11p and RFID may be combined to provide improved information dissemination solutions for VANETs.

Developing routing protocol for VANET has been a huge research challenge. Various routing protocols have been proposed, studied and investigated in past few years. Considering VANETs as a specific class of ad-hoc networks, the initial attempt was to test and evaluate ad-hoc routing protocols. It has been found that the reactive routing protocols [65] [44] can be adapted relatively easily in VANET environment than proactive routing protocols [25][66] and cluster based routing protocols [79][77]. This is due to high speed mobility of vehicles and rapidly changing topology of the network. Further, recently many routing protocols specifically suited for VANETs [40][81][13][64][55][78] have been proposed which provided the required impetus to the research and development in this area.

1.1 Motivation

If we consider the traffic condition in any urban locality at any given time, we may observe that roads are often blocked for some duration due to accidents, ongoing maintenance work etc. If the vehicles running towards the affected road segment are not aware of these conditions, it may result in traffic jams. Using their communication interfaces, vehicles running on roads can cooperate among themselves to disseminate these events to all the vehicles going towards the direction of road blocks. This reduces the volume of stalled traffic due to road blocks and eventually helps in clearing the obstacles relatively swiftly.

VANET may be utilized to disseminate important information like precarious location (construction site, post crash obstacle, road condition etc) to the vehicles that can provide comfort of driving, better managed traffic and other benefits. Vehicles can get information about the current and expected traffic conditions (expected delays, better routes etc). Information regarding the location of the next intersection and the signal timing can also be disseminated so that the vehicle can notify the drivers about the optimal speed.

If the vehicle travels at optimal speed, the driver is not required to slowdown or stops the vehicle at intersections as the traffic signal is likely to be green. This reduction in number of halts may result in increased traffic flow and increased fuel economy for vehicles. An excellent survey has been presented in [14] which provides an insight that utilizing timely information dissemination through VANETs can assist in reducing fuel consumption which can lead to a greener environment.

Efficient and scalable information dissemination remains a major challenge in VANET environment. This information cannot be broadcasted as every vehicle does not need this information. It is believed that once an efficient message dissemination technique is designed and developed, a large number of applications can be designed over VANET. Robust information dissemination increases message delivery ratio, decreases transmission delay and minimize packet communication overhead. However, developing applications over such a dynamic and distributed environment is challenging due to unpredictable and continuous change of underlying network topology. Moreover, it is not possible to establish any central administrative authority in such environments.

Developing applications for distributed and dynamic network such as VANET, designers face problems such as intermittent connectivity, network partitions, scalability, resource sharing etc. All the applications for VANET have to deal with these similar kinds of problems. Solving these problems in each application individually is not feasible. So a middleware should be developed that can solve these problems and thus the designer does not have to write code for all the common hindrances faced by applications designed over VANET.

Middleware is a layer of software which sits above the operating system and network substrate, but below the application. It is like a glue code which assists in composing independent systems together and makes them work together. Essentially, it reduces the burden of developing distributed applications for developer. Different authors have classified middleware in a different ways. In [39], six types of middleware are described. These are event based and message oriented middleware, component based and mobile-agent middleware, peer-to-peer based middleware, tuple-space based middleware, data sharing based middleware and virtual machine based middleware.

In highly dynamic environments such as VANETs, there is a need of a middleware which can provide decoupling and asynchrony. Event based middleware [61] are most suited for such environments. Generally, they utilize variants of publish/subscribe communication paradigm [31] which is an elegant solution for information dissemination in situations where set of information providers and consumers can frequently change over time. The main strength of this paradigm lies in decoupling in time, space and synchronization between event producers, called publishers and event consumers, called subscribers. Another component, called broker, which acts as a mediator between publishers and subscribers, assists in creating a decoupled environment where publishers and subscribers are unaware of each other and can dynamically leave or join the system.

Some research attempts [67] [42] [26] [27] [17] have been made to utilize publish/subscribe style of communication for MANETs. In these efforts, there is no designated broker node. Instead, every node of network is considered as a broker. These approaches can be broadly classified as deterministic approaches, based on dynamic formation of routing structures and probabilistic approaches, based on gossip or flooding.

In [67] and [42], algorithms for building and maintaining a tree based routing structure involving all the nodes of MANET are presented. The subscriptions and publications are routed along this routing tree so that they can match on any node for generating notifications. The limitation of these approaches is that even for moderate mobility high overhead is involved in maintenance of routing trees. In [26] [27] and [17], some approaches are presented which utilize gossip or flooding among the nodes where each node acts as a broker. These approaches are unable to provide guaranteed delivery of messages. Further, there is an additional overhead due to redundant forwarding of same messages to the intended subscribers.

The major challenge in realizing publish/subscribe middleware for VANET like environment is the design of broker component. In such a dynamic and large scale settings, a centralized broker is of limited use as it cannot provide desired quality of service and scalability. On the other hand, considering every node as a broker may not be scalable due to higher overhead involved in routing and matching of publications and subscriptions. Essentially, the broker component has to be realized in a distributed manner where some nodes of VANET can acquire the role of broker to achieve scalable application development. Further, this distributed broker network should be able to handle frequent changes in topology due to vehicle movement and exhibit self-organizing capabilities without any central coordination.

Peer-to-peer (P2P) systems exhibit excellent self organizing capabilities and offer easily modifiable topology without any central control to handle the events of dynamic leaving and joining of brokers. P2P systems are essentially application level virtual networks with their own topology and routing procedures.

Self organization is one of the striking features of P2P systems. P2P systems shift the management of network links to participating peers. This aspect allows the network of different connected devices to be built on the fly without any central coordination. Further, P2P systems are capable of automatically adapting to the arrival, departure and failure of peers. P2P networks are broadly classified as unstructured and structured based on the process of links formation among peers.

In unstructured P2P networks, overlay links are established in unconstrained fashion. These networks do not conform to any fix topology. When a new peer joins the network it forms connections with other peers freely i.e. it selects arbitrary peers as neighbors. In these networks, a search request has to be flooded through the network to locate peers having desired information.

Generally, unstructured peer to peer networks provide loose guarantee for resource discovery and there is a possibility that resource is not found although it exists in network. Requests for popular contents are resolved easily as these contents are most likely to be held by several peers. However, if a search is made for rare resource available at very few peers, then there is no guarantee that it will be resolved. Further, flooding causes high amounts of signaling traffic in the network and hence such networks typically have less search efficiency and network scalability. Examples of such P2P networks are Gnutella [3], Freenet [2] etc.

Structured peer-to-peer networks based on Distributed Hash Tables (DHT) are most suited network substrate for distributed broker implementation as they provide high scalability, fault tolerance in addition to self organization. Implementations of structured P2P networks utilize consistent hashing mechanisms to assign unique identifiers for participating nodes, events and subscriptions from a universal identifier space. There is a predefined relation that maps event and subscription identifiers to node identifiers for identifying responsible nodes to store them. Examples of structured P2P systems are Chord [80], Pastry [75] etc.

1.2 Problem Statement and Objectives

This thesis focuses on the problem of designing an event based middleware for information dissemination in VANET. We aim to design a publish/subscribe communication framework over structured P2P network of brokers which is deployable and gives desirable quality of service in distributed and dynamic environment ofVANET. We do not seek to design any specific application. Instead, the focus is on developing a robust infrastructure which can act as a substrate for designing information dissemination applications for VANET.

The major objectives of this thesis can be summarized as:

1. To design an intelligent event based middleware for information dissemination in VANET.

    a. Design of algorithms and procedures to implement publish/subscribe communication paradigm.

    b. Design of DHT infrastructure of brokers for dissemination and matching of events and subscriptions.

2. To simulate and verify the proposed algorithms using suitable simulators with specific vehicular mobility models.

3. To survey and gather traffic conditions of Indian cities and incorporating them in the simulated environment for proving the correctness and reliability of middleware.

1.3 Contributions of the Thesis

The contributions of the thesis are listed as follows:

1. To understand the design challenges. The design challenges for information dissemination applications over VANET can be summarized as:

   Timeliness: Vehicles should be notified of obstacles or hazards, such as accidents, or traffic jam, with sufficient time for a new route to be chosen.

   High mobility with dynamic topology: System should be able to tolerate the high mobility of vehicles and consequent rapid topology changes. The mobility is constrained as vehicles usually move along controlled trajectories.

   Sparse and uneven distribution of vehicles: Should be able to cope up with network partitions which are very common in VANETs where vehicles are often distributed unevenly. The distribution of vehicles is dependent upon the deployment area (urban or rural) and time of the day (before office hours, after office hours, night hours, etc.).

   Minimal infrastructure: System should be able to perform effectively in scenarios where communication infrastructure is minimal. Further, it should be able to utilize moving vehicles for disseminating information by using multi-hop communication links in ad-hoc manner.

Scalability: System should be able to scale so that more vehicles can be added anywhere and at any time without affecting the performance.

2. Design of publish/subscribe framework over DHT of stationary brokers. Further, to understand the available tradeoffs between infrastructure deployment and ad-hoc deployment of DHT of stationary brokers. The proposed framework has following salient features:

   – It enables vehicles to publish events, or subscribe to relevant information and route it towards the nearest info-stations through other vehicles using multi-hop connections.

   – The DHT based overlay enables info-stations to act as designated placeholders for subscriptions and events having similar content based attributes.

   – It does not rely upon GPS (Global Positioning System) for location information. Info-stations assist in locating subscriber vehicles. Using the DHT methodology, every info-station takes responsibility to store location information of subscriber vehicles waiting for event notifications.

   – Notifications are routed towards only those vehicles that have previously subscribed. This is done by utilizing the overlay links of the info-station and vehicle-to-vehicle multi-hop connections.

3. Design of publish/subscribe communication over DHT of mobile brokers. The proposed distributed and reconfigurable broker infrastructure utilizes city-buses for the notification of publications to interested subscribers. These

buses are assumed to be potentially connected to the Internet and have underlying IP based communication channel among them, for example by utilizing infrastructure-based cellular communication, like UMTS (Universal Mobile Telecommunication System). These city-buses act as brokers for other vehicles which are in role of publishers or subscriber, and can form a Distributed Hashed Table (DHT) based P2P overlay for disseminating publications and subscriptions. These mobile brokers also assist in locating a vehicle for successful and timely transfer of the notifications.

4. Design of a 2-Tier DHT overlay of fixed and mobile brokers for publish/subscribe communication. The proposed 2-tier DHT is combination of the two distributed broker infrastructures discussed above. To form this two-tier DHT, the whole region is divided into sectors where each sector has its own unique identification number. The installation plan of info-stations is made in a way to ensure that every sector contains only one info-station. The info-station present in a sector takes the role of super-peer. All these super-peers present in different sectors are connected in DHT manner and form the upper-tier of two-tier DHT. These super-peers are logically connected to city-buses currently moving in their respective sectors which also form a DHT among themselves.

1.4 Organization of the Thesis

This thesis is organized in the seven chapters.

Chapter 2 presents the details of background technologies. More specifically, it provides description of publish/subscribe communication paradigm and Chord DHT.

Chapter 3 presents the detailed literature survey of state-of-art of this body of knowledge.

Chapter 4 presents the design and simulation of publish/subscribe framework over DHT of stationary brokers.

Chapter 5 presents the design and simulation of publish/subscribe communication over DHT of Mobile Brokers.

Chapter 6 provides the design and simulation of 2-Tier DHT of stationary and mobile Brokers for publish/subscribe communication.

Chapter 7 summarizes the contribution of this thesis and discusses future work.

Chapter 2

Background Technologies

This chapter gives a brief description of publish/subscribe communication paradigm [31] and Chord DHT [80] as they are utilized in the proposed framework for information dissemination in VANET. The chapter is structured as follows. Section 2.1 describes publish/subscribe communication paradigm, different broker architectures and strategies to specify publications and subscriptions. Section 2.2 presents the description of Chord protocol and its algorithms.

2.1 Publish/Subscribe Communication Paradigm

For a dynamic and distributed environments like VANET − where it is not clear in advance that who needs what information and at which moment − traditional communication abstractions [31] such as message passing, remote invocations, shared spaces etc. are undesirable due to their inherent coupled nature. Further, these abstractions are synchronous which means that communicating partners should be available at the same time. Recently, publish/subscribe communication paradigm has evolved as an attractive alternative for developing distributed applications where underlying interaction mechanisms are required to be flexible, asynchronous and highly dynamic in nature.

The publish/subscribe system consists of three basic components: publisher, subscriber and broker. The publishers are the sources of information called events or messages and the subscribers are the consumers of the events. The consumers subscribe to particular categories of event. This is also called event based system where messages are the basic communication mechanism. Whenever a message is published, it is the responsibility of broker to deliver the message to all interested subscribers. The system may consist of multiple publishers, subscribers and brokers. Figure 2.1 and 2.2 shows publish/subscribe system with single and multiple brokers respectively.

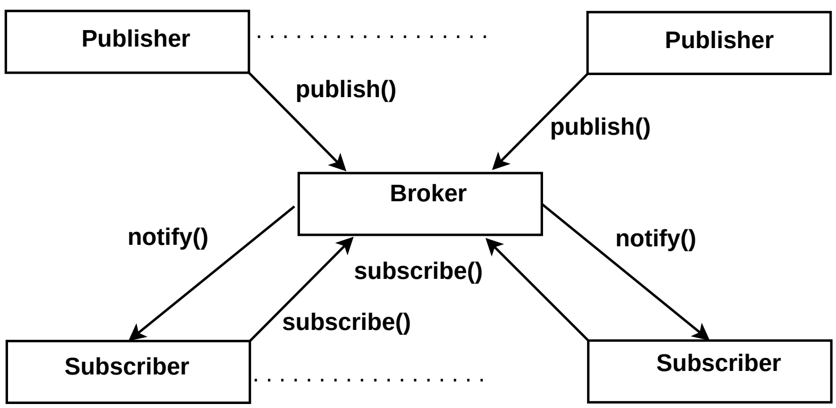


Figure 2.1: Publish/Subscribe system with single broker

2.1.1 Basic Interaction mechanism

The publish/subscribe interaction mechanism (Figure 2.1 and 2.2) relies on broker which acts as a mediator between the publishers and the subscribers. Subscribers register their interest with the broker using *subscribe*() primitive. These subscriptions are stored in the broker component and are not forwarded to the publishers. Subscribers can also use *unsubscribe*() primitive to unsubscribe or terminate the

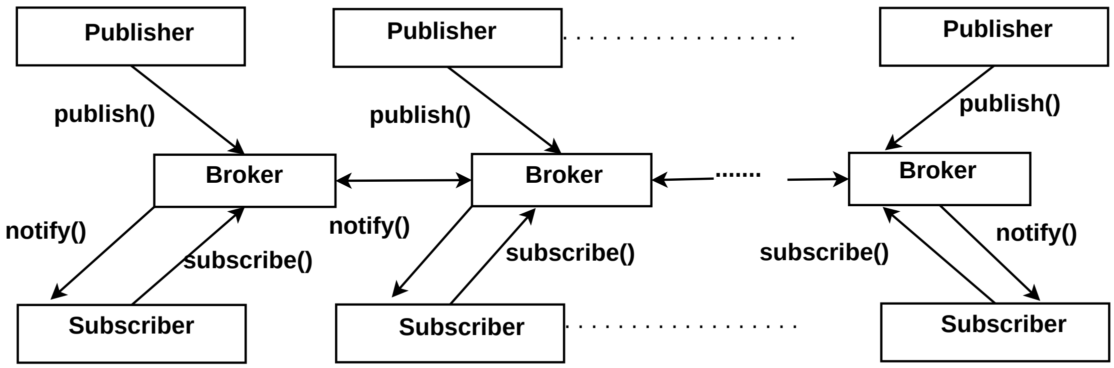


Figure 2.2: Publish/Subscribe system with multiple brokers

subscription. Publishers utilize *publish*() primitive when they wish to share any information. These publications are stored at the broker. The broker component uses matching algorithms to match the subscriptions and the publications and delivers the matching publications (also termed as notification) to the interested subscribers using $notify()$ primitive.

Clearly, publisher and subscriber do not communicate directly. The broker component acts as a mediator providing complete decoupling between them. There are three different types of decoupling facilitated through broker component; Time decoupling, flow decoupling and space decoupling. Decoupling in time means the event subscriber and event publisher need not active up at the same time. Decoupling in flow means sending and receiving does not block participants. Decoupling in space means the subscriber can move from one location to another without informing the publisher. Decoupling among communicating partners removes dependencies on each other reducing the requirement of synchronization and coordination. This provides the much needed flexibility for realizing applications in distributed and dynamic settings.

2.1.2 Classification

Publish/subscribe system can be broadly classified as topic-based, content-based or type-based depending on the manner in which a publication and subscription can be described.

- *Topic-based System*

In a topic-based system, publications follow some predefined topics or named logical channels. These topics are predefined statically based on some external criteria. Each topic can be viewed as separate logical communication channel or broker of its own. Subscribers in a topic-based system will receive all messages published to the topics to which they subscribe. Publication here does not depict the actual fine grained characteristics of published item. An external criterion statically defines the classes of messages to which subscribers can subscribe. For example, some topics (traffic information, parking space information etc.) and sub-topics (area wise traffic information, movie hall parking space etc) may be predefined to which vehicles can publish or subscribe. This scheme is easy to implement as matching process of publications and subscriptions just requires a simple comparison of two topics. Topic-based publish/subscribe is rather static and primitive, but can be implemented very efficiently.

- *Content-based System*

The content-based publish/subscribe provides fine-grained and more flexible alternatives to define subscriptions. The subscriber need not have to learn a set of topic names and their content before subscribing. Publications are not notified to subscribers according to any predefined set of topics. Instead, they are dynamically selected based on the properties of publications. These properties can be internal attributes of data structure carrying publications or the metadata of contents. The values of attributes or metadata can be different for different publications even though they belong to same topic. As a result, set of subscribers for a published item cannot be determined in advance and has to be dynamically figured out at the time of publication. Subscriptions are defined using logical formulas over name-value pairs. In name-value method, a subscription is defined dynamically as attribute-value pairs of properties. For example *(Movie Hall="some name", parking available= true or false, parking price<50)* can be a content based subscription. Content-based publish/subscribe is highly expressive, but requires sophisticated protocols that have higher runtime overhead.

- *Type-based System*

It is an extended version of topic based system with the features of object-oriented programming. Publications are generated as objects of specific types and subscribers are chosen based on their attributes and methods. From the point of view of expressiveness this system lies between topic based and content based system.

2.1.3 Broker Implementation

Considering scalability, usually the broker component is implemented in a distributed fashion. Broker process executes on different nodes of the network communicating with each other using underlying protocols. There are three popular methods to implement distributed brokers.

- Network Layer Multicasting
- Application Layer Broker Overlays

- Structured P2P Broker Overlay

Network layer Multicasting

In this method, the network layer multicast facility is utilized for communication among brokers. This method is most suitable for topic-based systems. It is a fast and easy to implement method with high throughput for many-to-many dissemination from one multicast group to another. Each multicast group may be assigned exactly one topic and subscribers can be simply attached to a group based on their interest.

However, for content based publish/subscribe where a group of subscribers can not be known in advance, network layer multicasting becomes insignificant. Some efforts [37][73][74] have been made to overcome this problem. In these efforts, subscribers with common subscriptions can be grouped together to utilize multicasting for dissemination of matching publications. Further, another design challenge is that network level multicast cannot guarantee reliable delivery. Though some approaches [34][87] have been proposed to achieve reliability over unreliable multicast channels but they suffer from high message overhead.

Application Layer Broker Overlays

This is the most common approach to implement distributed brokers in large scale publish/subscribe systems. Here the nodes executing broker process form virtual links between them over the underlying transport protocols. A broker connects another set of small brokers and thus that broker need not have the complete knowledge of the network. Examples of publish/subscribe systems using application level broker overlays are [9][20][28] etc.

This approach improves scalability but the major limitation of this approach is the problem related to topology creation and lack of self organization capabilities.

Generally, in these systems a fixed set of servers act as brokers and the overlay topology (ring, graph or tree) is defined by an administrator. This arrangement affects the scalability of system.

Ideally, for highly scalable system, the overlay should be self-organized and offer easily modifiable topology to handle the events of dynamic leaving and joining of brokers.

- *Structured P2P Broker Overlay*

As described earlier, structured P2P overlays are based on Distributed Hash Table (DHT) and exhibit efficient routing and self organization capabilities. Further, they also offer features like robust wide-area routing architecture, efficient search of data items and storage etc. Designing publish/subscribe systems over DHTs has recently gained popularity and resulted in several research proposals. For example [21][88] are topic-based designs whereas [18][82][84] etc. are content-based systems proposed over structured P2P overlays.

The distributed broker design for implementing publish/subscribe proposed in this thesis is based on Chord DHT [80]. Next section describes the Chord DHT in detail.

2.2 Chord

Chord [80] is one of the earliest, often cited, and very popular structured P2P overlay design. In this, m bit identifiers for both node and content are generated by using a *consistent hashing* technique [45], SHA-1. A node identifier is created by hashing its IP address while a content identifier (or key) is produced by hashing topic or attributes of content. Both node and content identifiers are taken from same identifier space.

The m-bit identifier space is represented as an *identifier-ring* modulo 2^m where m is the number of bits in the identifier. This is called a Chord ring. Further, taking the value of m large enough ($m = 160$), it makes the probability of generating duplicated identifiers for different nodes or contents very low.

A key with identifier k is stored on the first node whose identifier is equal to or follows k (clockwise) in identifier-ring. This node is termed as successor-node of a key with identifier k and denoted by *successor(k)*. Figure 2.3 shows a 3 bit ($m = 3$) identifier circle with 3 peers and 4 keys. Key 1 is stored at peer 2 as 2 is the successor of 1. Key 3 and 5 are stored at peer 6 and key 7 is stored at peer 0. The identifier circle is of 3 bits so it wraps around at $2^3 - 1 = 7$. That is why key 7 is stored at peer 0.

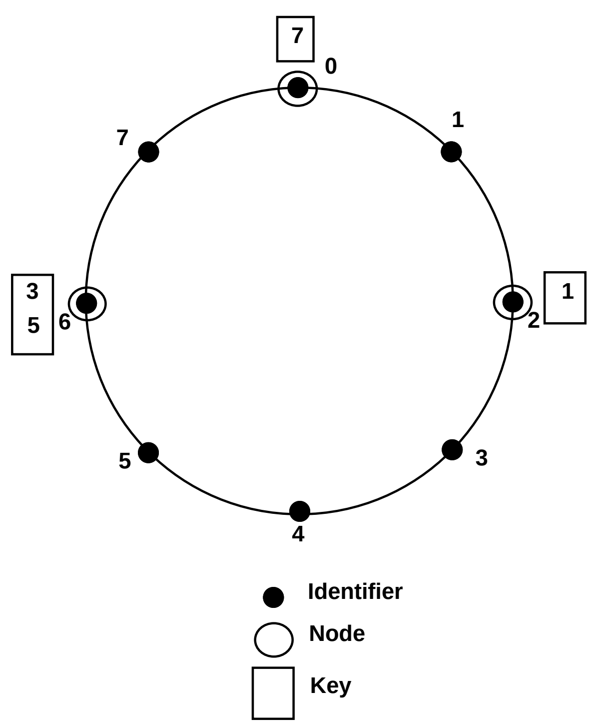


Figure 2.3: The Chord Ring

2.2.1 Routing

In Chord, each node maintains links to its successor-peer and predecessor-peer in the identifier ring. Routing is a process of passing lookup requests around the ring using successor-links. This process is terminated when a node is found which stores the desired identifier getting looked up. However, this simple routing strategy is inefficient. The worst case lookup time would be N for N nodes. To speed up the lookup process, in order to improve the routing performance, each peer maintains additional routing information as *finger-table.*

The finger-table is a routing table of at most m entries (for m-bit identifier space). The i^{th} entry in this finger table at any peer n contains identifier of a peer that succeeds n by at least 2^{i-1} on the identifier ring, where $1 \leq i \leq m$ and all operations are modulo 2^m. Figure 2.4 shows a finger table of node 0 having 3 entries. First entry in the finger table points to peer 2 as peer 2 is first peer $((0 + 2^{1-1}) \ \ mod \ \ 2^3) = 1)$ that succeeds 0. Second entry also points to peer 2. Third entry points to peer 6 as peer 6 succeeds peer 2.

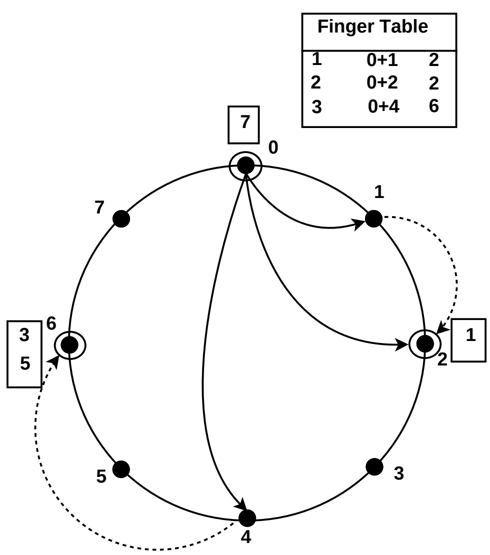


Figure 2.4: Chord Ring with Finger-Table

Algorithm 2.1 depicts the procedure to find the successor of any identifier using finger-table. If any peer n wants to find an identifier x then peer n calls $find\_successor(x)$ (line 2). If x is between n and successor of n then x can be found at successor of n (line 3,4). Otherwise, the largest predecessor of x is searched in the finger table of n by calling $closest\_preceding\_peer(x)$ ((line 6 and line 11-13). Subsequently, returned finger peer calls the procedure $find\_successor(x)$ (line 7). This process gets repeated recursively until the target peer is found.

Algorithm 2.1 Find Successor Procedure of Chord

1: // ask node n to find the successor of identifier x
2: $n.find\_successor(x)$
3: **if** $x \in (n, n.successor]$ **then**
4:     **return** $n.successor$
5: **else**
6:     $n' := closest\_preceding\_peer(x)$
7:     **return** $n'.find\_successor(x)$
8: **end if**
9: // search finger table for the largest predecessor of x
10: $n.closest\_preceding\_peer(x)$
11: **for** $i := m$ **downto** 1 **do**
12:     **if** $finger[i] \in (n, x)$ **then**
13:         **return** $finger[i]$
14:     **end if**
15: **end for**
16: **return** n

The finger-table leads to an efficient routing performance as it enables each peer to jump at power of 2 intervals around the identifier space. Generally, at least half of the distance can be covered between a peer and target identifier using finger table entries. Thus a lookup request can be routed between any two peers of the system in $O(\log_2 N)$ overlay hops. The formal proof for this is provided in [80].

2.2.2 Node Joining

If any node n wants to join the chord ring then it contacts any known node n' and requests to find its successor. This request is routed along the overlay. When the successor is found, then n is integrated in the ring by setting its successor link to point to the successor. Node n builds its finger table with the help of its successor. Algorithm 2.2 describes the peer joining procedure.

Algorithm 2.2 Peer Joining in Chord

// peer n joins through peer n'
$n.join(n')$
$predecessor := nil$
$s := n'.find\_successor(n)$
$successor := s$
$build\_fingers(s)$
// build finger table
$n.build\_fingers(s)$
$i_0 := \lfloor \log(successor - n) \rfloor + 1$
for each $i \geq i_0$ **do**
    $finger[i] = n'.find\_successor(n + 2^{i-1})$
end for

It may be noted that after the execution of Algorithm 2.2, the successor link and finger-tables of newly joined peer n are updated. However, few more tasks are required to be performed for the completion of joining process. These tasks are listed as follows:

1. Setting up the predecessor link of newly joined peer n.

2. Update of finger-tables, successor links and predecessor links of the existing nodes in overlay. These may have changed due to joining of n.

3. Transfer of contents for which the new node n is responsible, from its successor node.

The above tasks are performed utilizing several Chord periodic maintenance algorithms which are described in next subsection.

2.2.3 Chord Ring Maintenance

Due to dynamic leaving and joining, there is a need of periodic maintenance of the overlay structure. The periodic maintenance in Chord ensures that successor and predecessor links become eventually correct. Following periodic maintenance procedures are executed by every node in Chord.

- $stabilization()$:To learn about the new nodes that have joined.
- $check\_predecessor()$:To learn about the failure or departure of nodes.
- $fix\_fingers()$:To update the entries of finger-table that may have changed due to leaving or joining of nodes.

As described in previous section, a node n joins by setting its successor link to successor s. Still, the predecessor link of n and s as well as the successor link of predecessor of n are remained to be set. Chord sets these links by letting nodes periodically execute the stabilization process. This process also allows existing nodes to learn about new nodes that have joined the overlay in concurrent manner. Algorithm 2.3 describes the $stabilization()$ procedure.

For example, let n_p and n_s are two existing nodes in the overlay where n_s is successor of n_p. A new node n_{new} wants to join whose hashed identifier lies between n_p and n_s. New node initiates joining by setting its successor link to n_s. Subsequently, new node notifies its successor about itself after executing stabilization procedure (line 7 of Algorithm 2.3). After receiving the notification, node n_s sets node n_{new} as its predecessor (line 9-11 of Algorithm 2.3). When node n_p executes stablization

Algorithm 2.3 Stablization Procedure

1: // periodically probe the successor s of n and notify s about n
2: $n.stabilization()$
3: $x := successor.predecessor$
4: **if** $x \in (n, successor)$ // successor has changed due to new joining **then**
5: $\quad successor := x$
6: **end if**
7: $successor.notify(n)$
8: // n notify its successor s that n is the predecessor of s
9: $s.notify(n)$
10: **if** $predecessor = nil$ *or* $n \in (predecessor, s)$ **then**
11: $\quad predecessor := n$
12: **end if**

it sets node n_{new} as its successor (line 3-5 of Algorithm 2.3) and notifies n_{new} (line 7 of Algorithm 2.3). After receiving the notification n_new sets n_p as its predecessor (line 9-11 of Algorithm 2.3).

Nodes in Chord periodically checks the availability of their predecessors and successors. The presence (or absence) of successor can be figured out by $stabilization()$ procedure. Further, the procedure $check\_predecessor()$ runs at every node periodically setting the predecessor pointer to nil if the predecessor node has failed. This permits a node to accepts a new predecessor during execution of $notify()$ procedure (line 9) of Algorithm 2.3.

To increase the robustness of Chord against node failures, each node maintains a successor-list of some r nodes following it in the identifier ring. If the immediate successor of a node does not respond during successor stabilization, it is replaced by the next node in successor-list. The event of simultaneous failure of all the peers in successor-list is highly unlikely as even for the modest values of r, the probability of this event becomes very low.

Some minor changes are done in *stabilization*() procedure (Algorithm 2.3) when successor-list is maintained on nodes. While node n performs stabilization, the successor-list of its successor s is copied, its last entry is removed and s is appended as the first entry. The procedures for checking predecessor, maintenance of successor list and fixing successor in the event of node leaving are given in Algorithm 2.4.

Algorithm 2.4 Stabilization Procedure with Successor Lists

// periodically check for the availability of predecessor
$n.check\_predecessor()$
if $predecessor\ has\ failed$ **then**
    $predecessor := nil$
end if
// periodically update the successor-list
// Number of entries in successor-list is r
$n.fix\_successor\_list()$
$\langle s_1, ...s_r\rangle := successor.successor\_list$
$successor\_list := \langle successor, s_1, ...s_{r-1}\rangle$
//periodically update failed successor
if $successor\ has\ failed$ **then**
    $successor := next\ entry\ in\ successor\_list$
end if

Due to joining and leaving of nodes, entries in the finger-tables of existing nodes may become wrong. To tackle this, each node in Chord periodically refreshes the finger-table entries using $fix\_fingers()$ procedure. Algorithm 2.5 outlines the $fix\_fingers()$ procedure. Here m is the number of entries in finger-table and the variable $next$ is initialized to 0.

Algorithm 2.5 Fix-finger Procedure

$n.fix\_fingers()$
$next = next + 1$
if $next > m$ // m is the number of entries in finger-table **then**
    $next = 1$
end if
$finger[i] = n.find\_successor(n + 2^{next-1})$

Chapter 3

Literarture Survey

Survey of existing literature reveals that a number of research efforts have been proposed, where various information dissemination approaches for VANET like settings are discussed and analyzed. These research efforts can be roughly grouped as event based middleware approaches and P2P network based approaches. Further, details of some other approaches are also presented that utilize city transport system, cellular networks etc. for ferrying information among vehicles. The chapter is organized as follows. Section 3.1 presents the details of event based middleware approaches. Section 3.2 describes P2P networks based approaches and Section 3.3 provides details of other approaches.

3.1 Event Based Middleware Approaches

Developing applications for distributed network such as VANET, designers face problems related to distribution like heterogeneity, scalability, resource sharing and accessing information in this heterogeneous environment. All the applications for VANET deal with these similar kind of problems. Solving these problems in all applications is not feasible. So a middleware should be developed that can solve these problems and thus the designer does not have to write code for information access and resource sharing in every application of VANET.

Several middleware have been proposed for MANET environment [19][61][63] [35][24][41][60][57]. They can be roughly classified [39] as event based and message oriented middleware [61], component based and mobile-agent middleware [35][24], peer-to-peer based middleware [19], tuple-space based middleware [41], data sharing based middleware [60] and virtual machine based middleware [57].

In dynamic and mobile settings such as MANETs and VANETs, there is a need of a middleware which can provide decoupling (among communicating partners) and asynchrony. Event based middleware are most suited for such systems. They use variants of publish/subscribe paradigm (discussed in Chapter 2) which assists in creating decoupled and asynchronous communication environment.

STEAM [61] is an event based middleware for MANET. It is purely based on the publish/subscribe communication paradigm. STEAM was designed to support entities that communicate with each other when they are in close proximity. The consumers subscribe to a particular event types and the publisher publishes the events. It does not provide efficient resource discovery mechanism to support high nodes mobility. It addresses scalability when the mobile nodes are within range but problems can arise when it comes to distributed applications using entities which are not in close proximity.

EMMA [63] is a message oriented event based middleware for MANET. It uses Java Message Service (JMS) which is originally designed for semi mobile distributed systems. EMMA can use both point to point and publish subscribe communication system. If the mobile nodes are not in range then epidemic routing mechanism is used for message routing.

Clearly, middleware developed for MANETs cannot be used in VANETs because of the distinguishing characteristics of VANET discussed in Chapter 1. There are some existing research efforts [43][53][56][52][54], which have been targeted specifically towards development of middleware for VANETs.

In [43], an event based middleware called RT-STEAM has been developed. This is designed for VANET that often requires reliable communication that provides guaranteed real time message propagation. Unlike other event based systems RT-STEAM does not rely on a centralized event broker or look-up service. Events are filtered based on subject, content and/or proximity similar to STEAM. Further, like STEAM, RT-STEAM can also be used efficiently when the nodes are in close proximity. In VANET, vehicles are more likely to interact once they are in close proximity. For example, emergency braking notification from the other vehicle is received only when that vehicle is close.

In [53], [56], [52] and [54] an attempt is made to design a publish/subscribe middleware for vehicular networks that considers location and time in its design objectives. This middleware enables the application developers to easily publish notification in specific location by treating location as context. In these approaches, a hybrid setup is assumed where there are stationary info-stations and moving vehicles communicating in cooperative manner. These info-stations are assumed to be connected to the Internet for timely information spreading. Vehicles are assumed to be installed with navigation system and GPS (Global Positioning System) and they behave like mobile sensors that collect information about traffic condition, parking situation etc. Then with the help of GPS and navigation system a publish/subscribe middleware is used to disseminate information geographically.

The proposed middleware takes advantage of the information that can be extracted from the vehicles navigation systems (location, map, destination of the driver etc) to generate subscriptions. Navigation system decides if a vehicle is interested on receiving a specific notification or not. These approaches treat vehicles as mobile sensors that collect information about traffic conditions, accidents etc. Mobile vehicles transfer this information to the info-station on its way. All the info-stations are directly connected. A centralized system combines the gathered information and generates traffic warnings. Traffic warnings are sent to the nearest info-station and from there they are routed towards the affected road segment by vehicle-to-vehicle communication.

Authors of [22] have proposed an ACIS communication middleware, called ACME (ACis Middleware for Emerging VANET applications), which is broadly based on publish/subscribe paradigm and the peer-to-peer networks. One of the main goals of the project was to turn each moving car into a sort of mobile sensor such that it is able to observe and report about the traffic. All the vehicles have a street map where each road is dissected into fixed cells, each univocally identified by the geographic coordinates of its middle point. The size of each cell is such that direct 802.11p communications can take place between any two nodes within adjacent cells. In each cell some vehicle is designated as cell leader. The cell leaders are appointed to collect traffic data of their respective cells. This paper has not discussed the details of publication, subscription and matching methods. Also they are dependent on positioning devices like GPS receiver, GSM cell id etc.

3.2 Peer-to-peer Approaches

Several approaches [29][50][51][12][11][16][70][71][76] have been proposed which mostly utilize and augment concepts of popular P2P file sharing mechanisms like BitTorrent [7] already deployed over the Internet. Generally, in these approaches a file of interest is divided into number of pieces and peers with fractions of file form a swarm like overlay among them. This mechanism where a group of cooperating peers perform parallel downloads of pieces of a file is called swarming.

Spawn [29] is a file swarming infrastructure dependent protocol for VANET. It is a simple cooperative strategy for content delivery in VANET. This uses a mixture of centralized and decentralized approach for peer discovery. Vehicle can download the file from the gateway if it is in the range of gateway otherwise it uses gossip for content availability. Message is sent from one peer to another peer. A peer who is not interested in the message, simply forwards it. Here proximity driven piece selection strategy is used to select a peer which has information to transfer where proximity is estimated based on hop count.

CarTorrent [50] is a BitTorrent [7] like file swarming protocol designed for VANET which uses cooperative P2P paradigm. It is the first system which is tested on a real VANET testbed. For a given file CarTorrent client disseminate information via $k - hop$ limited scope broadcasting. Each message is forwarded until it reaches to nodes located $k - hop$ away from originator. Thus, peers can gather statistics such as local topology and piece availability which are then used to select a piece/peer that is preferably close in proximity.

CodeTorrent [51] is a P2P network coding based file swarming protocol for VANET. CodeTorrent is based on single hop data pull model. Multi-hop routes are never used and thus are not required to be maintained explicitly by any layer in the protocol stack. Data is propagated through the overlay network of peers of common interest. A node that wants to share a file (seed node) broadcasts the file description to its $1-hop$ neighbor. If any of the $1-hop$ neighbors is interested in file, nodes will apply the idea of network coding and will exchange coded frames instead of the file pieces, where a frame is a linear combination of file pieces. When the interested node collects n independent frames, it can decode and recover the entire file. As the file description is broadcasted, it also reaches to the vehicle which is not interested. This results in unnecessary messages and consumptions of resources.

VANETCODE [12] is another cooperative content distribution scheme which is also based on the concept of network coding. Here some stationary gateways are supposed to be installed along the road at regular intervals of about 2 to 10 miles. The content at the gateway is divided into smaller blocks and the nodes linearly encode their constituent blocks. The encoded blocks are then shared amongst the neighboring nodes. Since each node can act as a router, the network coding can be done at the network layer without hindering the upper layer protocols. The information is broadcasted to one-hop neighbors like CodeTorrent. VANETCODE eliminates the need of peer selection, content selection and neighbor discovery, which take up significant time and resource in other cooperative downloading mechanism proposed for VANETs. However, VANETCODE has some limitations. First, the gateways are the only sources of data and it is expected that all data of interest to be existing and replicated at all gateways. Second, there is no sharing of the content stored in different vehicles. Third, the system does not scale as adding or modifying data to all gateways is a challenging issue.

Zipper [11] is a zero infrastructure Peer-to-Peer system for VANET which is used for multimedia streaming. ZIPPER does not require any pre-installed infrastructure along the road so it can be implemented with no extra cost. In ZIPPER, files are stored as a collection of blocks and a vehicle may not possess the entire set of blocks of a certain file as it may be in the process of downloading. The vehicle can send the query request only to its one-hop neighbors where the query is processed. When the nodes are able to send a block of data they use Traffic Adaptive Packet Relaying (TAPR) [10] routing protocol to deliver the content to the query sender node. An updated query is propagated to next hop mobile nodes to search for remaining blocks.

Cooperative and infrastructure-free peer-to-peer system (Coffee)[16] is proposed for multimedia file sharing between mobile vehicles. The query requester node forms a virtual group using trajectory prediction to form a structured overlay network. Each mobile node sends a HELLO packet periodically to its one-hop neighbors to indicate its presence in the network. It is assumed that the mobile nodes are aware of the future mobility positions of their neighbor nodes and this information is integrated and sent in the HELLO packet. When a mobile node wishes to send a query request on the network, Direction-based Geocast Routing protocol (DG-CastoR) [15] protocol is used to route the query packet between mobile nodes that can meet the requester node during a period of time. A large multimedia file is partitioned based on tree structure represented by levels which is well described in the paper. They also try to improve the data availability on the network, by proposing a reactive on-demand data replication strategy.

In [70] and [71], an approach is proposed where the city is divided into several segments and each segments forms a separate and interacting Chord DHT based peer to peer network of moving vehicles. They have assumed that each vehicle knows its position, direction and velocity. This information is provided by sophisticated devices such as in-car sensors and navigation systems. Further, the vehicles are equipped with digital maps which assist in triggering the event of crossing the segment border. Vehicle can leave one segmented DHT and join some other while moving. Their approach proves that no knowledge about the network segmentation is required for information dissemination. Though their approach looks feasible but there are certain limitations. The authors have not provided any supporting simulation results. Moreover, authors have assumed that within a segment vehicles can directly communicate to each other i.e. all are in communication range of each other.

In such settings, all the vehicles can overhear the messages within a segment and thus the utilization of DHT formation in such small segments needs investigation.

PeerTIS had been proposed in [76]. Here all the vehicles are equipped with devices having Internet connections. These vehicles form a structured peer to peer overlay network over the Internet using cellular Internet access to realize scalable information sharing. In near future, 3G/4G network is going to be used by more number of users which will reduce the available bandwidth per person. Using the same technologies for vehicular communications also may add burden to already scarce bandwidth. However, only relying on 802.11p based ad-hoc communications for information dissemination may not provide the intended performance. It is required to make balanced use of both technologies for the desired quality of service.

3.3 Other Approaches

This section presents some other research efforts which utilize cloud systems, city transport system and cellular networks etc. for information dissemination in VANET.

PAVAN [36] utilized the existing cellular network to broadcast a file description to all vehicles in the current area. If a driver is interested in the file, a route should be discovered and maintained between it and the owner of the file. Scalability is again an issue here. As the number of vehicles transmitting their file descriptions increases, the cellular network, which supports only a few tens of kilobits per second, will not be able to carry such a load.

In [59], an Intelligent Transport System (ITS) Information Platform (IIP) is proposed which act as a common data management and communication platform facilitating easy application development. This paper describes two levels of IIP, one residing on the cloud which is referred as the IIP Cloud Component and the other residing on the clients which is referred as the IIP Client Component. The IIP Cloud Component provides publish/subscribe functions to both traffic management facilities and mobile nodes. The IIP Client Component provides mobile nodes with publish/subscribe functions which allow them to communicate with the IIP Cloud Component as well as with each other. The Cloud Interface allows mobile nodes to access the IIP Cloud Component through the Internet access points. The Cloud Interface cannot support direct mobile peer to peer applications. A mobile P2P interface is also suggested as a complementary communication method. However, any specific peer to peer protocol is not proposed. Authors have also not provided any supporting simulation results or evaluation.

Further, some approaches [58][30] have been proposed recently which utilize public transport (city-buses) for information ferrying between other ordinary vehicles. However, these approaches do not utilize publish/subscribe communication paradigm.

In [58], a two tier architecture is proposed where the upper tier of buses constitute a mobile backbone for data delivery while the low tier is composed of ordinary cars and passengers. The authors have argued that city-buses in their approach can be considered to behave like wireless mesh routers and establish direct connections among them. Though the presented approach is promising, but there are some limitations. How the mobile city-buses - with dynamically varying distances among them - can behave like wireless mesh-routers, and maintain direct connections is not clear. Also, for location services buses use GPS and has digital street map of the city with bus line information.

In [30], an approach is presented where the information dissemination flow is restricted to follow the routes of buses only. The authors have proposed a grid based virtual backbone of city buses. City-bus with the longest stay duration is elected as the grid leader which stores the information around the grid. There are many grid leaders in the city and all are connected to each other using ad-hoc communication links between each other. All the vehicles in the backbone are assumed to be equipped with a GPS device for location services. The presented approach heavily relies on the durability of ad-hoc connections formed between city-buses (grid leaders). The approach is feasible only when traffic density is high and distribution of vehicles is uniform across the region. Authors have not evaluated their approach under different traffic conditions which limits the applicability of the proposed approach.

Chapter 4

Pub/Sub over DHT of Info-stations

This chapter presents the design of publish/subscribe communication framework over DHT of stationary brokers. Here the info-stations installed at every major intersection of city are utilized as stationary brokers and act as rendezvous points for related publications and subscriptions. Further, these info-stations also assist in locating vehicles that have subscribed for information. Both infrastructure deployment and ad-hoc deployment of info-stations are considered and performance of proposed framework is evaluated under different traffic conditions of two Indian cities (Delhi and Allahabad) as a reference for urban and semi-urban areas.

The chapter is structured as follows. Section 4.1 describes the system model. This section also describes the effect of skewed vehicle distribution on overlay structure. Section 4.2 presents the description of algorithms utilized in proposed framework. Section 4.3 provides the details of simulation environment and results collected. Section 4.4 presents the concluding discussion.

4.1 System Model

We assume a city based scenario where the info-stations are installed at major positions (e.g., important intersections) of the city. These info-stations act as rendezvous points for publications and subscriptions by forming a DHT structure among them. These info-stations and the vehicles are equipped with Omni-directional antenna and they have fixed transmission range. Each vehicle and info-station has its unique identification number. We assume that the info-stations have a transmission range of 500m whereas vehicles have a transmission range of 200m. The minimum distance between two info-stations is assumed to be 3 kilometers. Info-stations broadcast their ids periodically for the vehicles moving around them. Vehicles can send publications/subscriptions to info-stations directly if they are in their range. Otherwise, publications and subscriptions are forwarded hop by hop by utilizing other vehicles moving on the road.

Info-stations act as the ultimate meeting points or brokers for publications and subscriptions and they forward matching publications to interested subscribers. Each vehicle and info-station has its own unique identification number. Info-stations form a Chord like DHT structure.

Chord is utilized to create hashed ids of info-stations to form a logical overlay ring. The published and subscribed messages get hashed content ids. These content ids are termed keys. Both content and node ids are chosen from same identifier space which is taken big enough to avoid node and content ids mapping to the same hash value. This overlay of info-stations is used to form rendezvous points for publication and subscriptions. We assume following two deployment scenarios:

- *Infrastructure deployment*: Info-stations are connected to the Internet. The underlying communication among them is through the IP based Internet.

- *Infrastructure less deployment*: Info-stations are not connected to the Internet. The underlying communication among them is multi-hop via vehicles.

Further, we assume that vehicles are not equipped with GPS (Global Positioning System) or navigation systems in our approach.

4.1.1 Skewed vehicle distribution

It is believed that distribution of vehicles running around any area is not uniform. It is dependent on time of day, with some popular hot spots in particular area. For example, during morning and evening hours, traffic density is maximal due to crowds of office goers. Similarly, in night hours traffic density is minimal. Further, some areas can be considered as hot spots during overall low traffic density times. We observe that around movie theatres, airports, railway stations, night clubs, etc., traffic density is more during off hours.

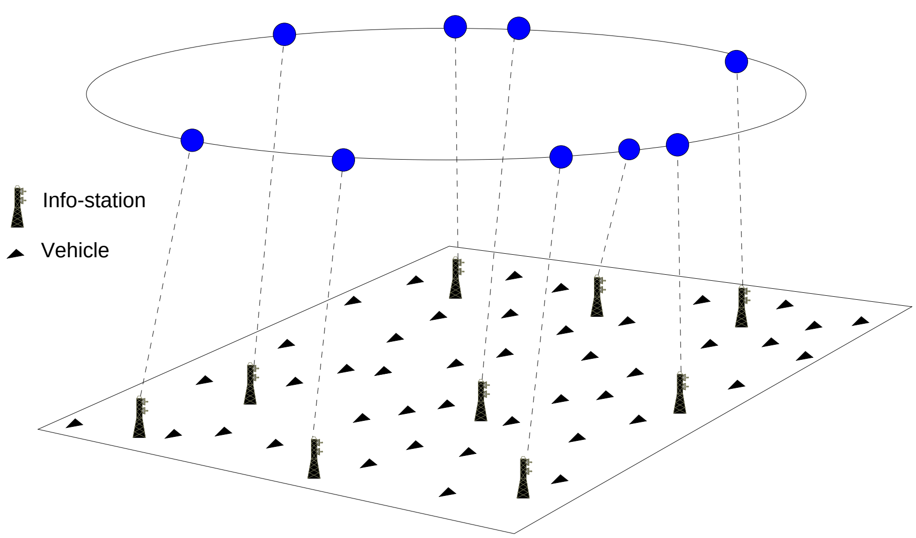


Figure 4.1: DHT of info-stations

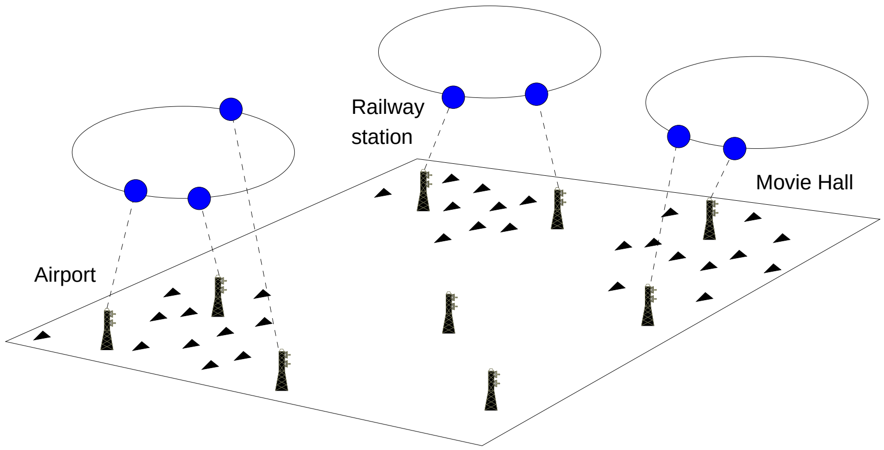


Figure 4.2: Splitting of DHT

This skewed distribution of vehicles has an interesting effect on the DHT of info-stations when we consider infrastructure-less deployment. In high density situations, all the info-stations are able to run the DHT maintenance algorithm and, as a result single DHT of info-stations is formed. On the other hand, under skewed distribution and low density, this single DHT is broken up into smaller DHTs as only those info-stations which are surrounded by vehicles can maintain their neighborhood information. This split of DHTs is depicted in Figure 4.1 and Figure 4.2.

4.2 Description of algorithms

In this section we present the algorithms for publication, subscription, location determination, and notification delivery used in our approach. Each vehicle has the following data structures to manage publications and subscriptions and to provide location services:

- Subscription-Table: to store active subscriptions
- Publication Table: to store active publications

- Forwarding Table: to store others publications and subscriptions for further forwarding.

- Vehicle id: Identification number of the vehicle.

- Last info-station time stamp: Identifier of the last info-station passed and the time stamp of this event.

Info-stations also has Subscription Table, Publication Table, Info-station id, and Forwarding Table. Additionally, they maintain a Location Table which stores the location information of vehicles they are responsible for.

4.2.1 Publishing, subscribing and matching

In our approach, the publish primitive is defined as Publish (publication specification, TTL) where TTL is the time for which a publication is considered to be active. Subscribe primitive is defined as Subscribe (subscription specification, subscriber_id, TTL) where subscriber_id is the identification of the vehicle which subscribes and TTL is the time for which a subscription is valid. Algorithm 4.1 provides the procedures for publishing and subscribing.

Vehicles willing to publish or subscribe send the publication/subscription to the nearest info-station through other vehicles running between itself and the info station. Vehicles forward the publication/subscription to one hop neighbors in range. Vehicles which are running ahead of the vehicle (that is publishing or subscribing) in the same direction and the vehicle running in opposite direction carry the publication or subscription and send it again to their one hop neighbor in their ranges. This process continues till the publication or subscription reaches an info-station. If traffic is dense, then publication/subscription is transferred to the info-station with less delay. On the other hand, if there are very few vehicles on road, then there may be substantial delay in sending the publication/subscription to info-station.

Algorithm 4.1 Publication and Subscription

```
1: // Procedure Publish(publication, TTL)
2: Store Publication in Publication Table
3: if Publication is active // Checking TTL value then
4:    if any vehicle in 1-hop range then
5:       Send Publication
6:    else
7:       Keep on moving for some time
8:       Goto Step3
9:    end if
10: else
11:    Discard the Publication
12: end if
13: // Procedure Subscribe(vehicle − id, subscription, TTL)
14: Store Subscription in Subscription Table
15: if Subscription is active // Check TTL then
16:    if any vehicle in 1-hop range then
17:       Send Subscription
18:    else
19:       Keep on moving for some time
20:       Goto Step15
21:    end if
22: else
23:    Discard the Subscription
24: end if
```

In this process of hop-by-hop transfer of publication and subscription towards info-stations, it may be possible that some of the vehicles (through which the publication/subscription is forwarded) receive matching publications or subscriptions. In this case, in addition to forward the publication/subscription, they can also act as rendezvous point for publications and subscriptions. Once the publication/subscription reaches an info-station, it is sent to the rendezvous node by utilizing the DHT structure connecting the info-stations.

Algorithm 4.2 provides procedures which outline forwarding and matching processes for moving vehicles.

Algorithm 4.2 Forwarding and Matching

```
// Procedure Forwarder(publication/subscription/notification, vehicle − id)
Case: (notification)
if notification for itself then
   Accept
else
   Goto Step18
end if
Case: (subscription)
//Match with Subscription Table and Forwarding Table
if match=TRUE in Subscription Table then
   Accept as notification
else if match=TRUE in Forwarding Table then
   Build Notification
   Goto Step18 to notify concerned subscriber
else
   Goto Step18 to forward publication further
end if
if Publication/Subscription/Notification active // Check TTL then
   if any vehicle in 1-hop range then
      Forward Publication/Subscription/Notification
   else
      Keep on moving for some time
      Goto Step18
   end if
else
   Discard Publication/Subscription/Notification
end if
```

Each vehicle, after receiving a publication or subscription from other vehicles first performs the matching operation with pre-stored publications and subscriptions. At this instant, each vehicle assumes the role of broker. For example, every received publication is first checked with the subscriptions in the subscription table and forwarding table. If the received publication matches with any subscription of subscription table then it is accepted as a notification by the vehicle receiving the publication. If the publication matches any subscription in forwarding table then the corresponding subscriber is notified by the vehicle itself. If none of the cases mentioned above occurs, then the publication is forwarded to other vehicles in the one hop range. This process is repeated until a matching subscription is found at vehicles on the way or the publication reaches at the nearest info-station.

When a publication or subscription reaches any info-station it is routed to a rendezvous info-station by utilizing the DHT routing substrate. These info-stations are connected to each other through virtual links provided by the underlying transport layer. We predefine a specific attribute schema for describing publications and subscriptions. Each publication and subscription specification has some attributes associated with it. These attribute names are hashed to find the rendezvous info-station of publications and subscriptions. Consequently, publications and subscriptions related to the same attributes are routed towards the same info-station. The publications and subscriptions are matched with each other at rendezvous info-stations and corresponding subscribers are notified with the help of vehicles passing info-stations.

4.2.2 Tracking subscriber location

As vehicles move and subscribe simultaneously, the location of subscribed vehicles will very likely be different when notifications are ready to be delivered. Vehicles may issue subscription in one region of a city and at the time of notification they might be in any other region. In our approach, we have not assumed GPS enabled vehicles. Instead, subscribed vehicles and info-stations work in a cooperative manner to locate vehicles.

Location information of subscribed vehicles is maintained in the DHT of info-stations in a distributed fashion. Every info-station is responsible for storing the location information for a set of vehicles. The vehicle id is hashed to discover the info-station which stores the location details of a vehicle. Each vehicle broadcasts its id and direction at the time of passing an info-station. That info-station hashes the vehicle id to find out which info-station is responsible for the location base of the vehicle and updates the location information at that info-station. In this way, the location of every subscribed vehicle is up to date at info-stations.

It should be noted that the system is not required to maintain the location details of all the vehicles. It would be very costly overhead to regularly update the locations of all vehicles all the time. The location information is maintained only for vehicles that have subscribed to some events and are thus expecting notifications. The information consumers are not interested in knowing the id or location of information providers. They are only interested in getting the desired information. On the other hand, subscribed vehicles must be located to forward the notifications to them at a given time.

The info-stations are installed at major intersections of city in our approach. As mentioned earlier, info-stations periodically broadcast their ids to vehicles moving past them. This assists vehicles as they know that they are nearing info-stations and certain actions should be taken. Roads leading away from info-stations are marked with direction tags (North, South, East, and West) as depicted in Figure 4.3. Vehicles can inform the info-stations that which direction they are coming from and going to.

How this direction information is generated can be vehicle or application specific. At an info-station, vehicle can either turn left, right or go straight. Smart vehicles can record their steering movement while they are passing info-stations to know what direction is taken.

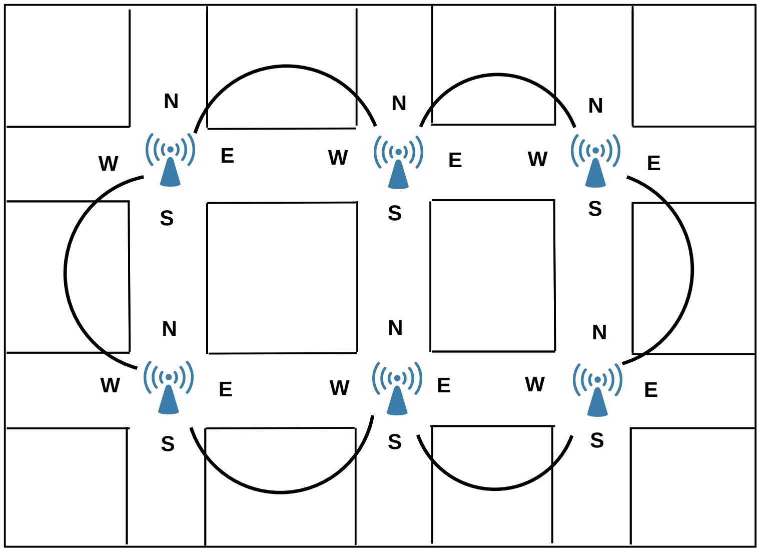


Figure 4.3: Tracking vehicle location

The direction information with timestamp is stored at the info-station responsible for the location of the vehicle. This last info-station crossed and direction information is also stored in the vehicles. For example, suppose a notification is ready at an info-station for any vehicle. The info-station enquires about the location of vehicle by contacting the info-station maintaining the location of a vehicle via DHT lookup. The returned info-station reports the last info-station passed data for that vehicle. The notification is forwarded to that last passed info-station using the underlying DHT routing substrate which uses the direction information to forward the notification by using other vehicles as carriers that are passing the info-station and moving in that direction. If a notification is ready at any vehicle (matching publications and subscriptions on vehicles met on the way), then the notification is first transferred to nearest info-station which forwards it to the target info-station.

In our design, subscribed vehicles are not required to be located in a fine grained manner. Our vehicle tracking approach is rather a coarse-grained one. It provides details of the info-stations between which a vehicle can be found at any given time. Once vehicle is located between two info-stations, then other vehicles moving in the direction of the target vehicle can be utilized to forward notifications.

4.2.3 Notification delivery

Notifications are delivered using the location information of subscriber. They are forwarded towards target info-stations which in turn forward them towards the target vehicle. This process is explained by the following example. Suppose we learn that a target subscriber vehicle is between info-stations A and B at a given time (has passed A and is moving towards B). The notification is routed towards A which forwards it through vehicles moving towards info-station B. There may be a situation that more than one vehicle acts as a carrier for this notification. There can be two cases; (1) the notification is delivered to the target vehicle but it is still being forwarded towards B (because more than one vehicle was carrying it) (2) notification is not delivered to target vehicle (perhaps because the target vehicle has already passed B at the time the notification arrived). These two cases are handled by our approach in following manner:

- In the first case, when the subscriber vehicle passes B, it informs B that it has received notifications (with their id) while it was in between A and B. Thus when duplicate notifications reach at info-station A, they are discarded and not forwarded further.

- In the second case, B knows which direction the target vehicle has moved in. Consequently, when a notification reaches B (and the subscribed vehicle has not acknowledged the receipt of the notification), it is forwarded towards the target subscribed vehicle.

4.3 Simulation environment and results

We have simulated our approach by using Oversim [5] over OMNeT++(Objective Modular Network Testbed in C++) [4] and Traffic control Interface (TraCI) client for OMNeT++/INET framework. Our approach for simulating the peer to peer overlay network of publishers, subscribers, brokers and info station is depicted in Fig.4.4.

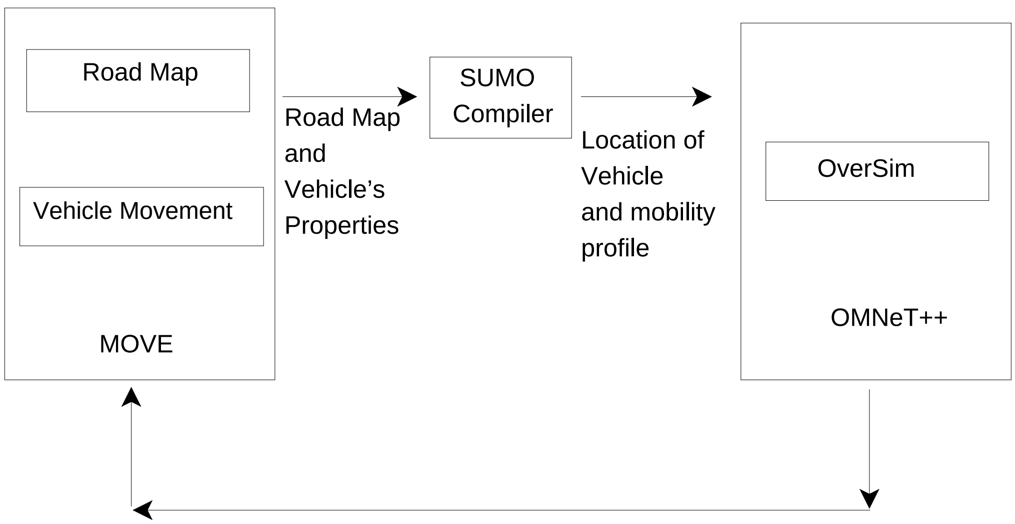


Figure 4.4: Traffic and network simulators

OverSim [5] is an open-source simulation framework for peer to peer overlay networks to be used over OMNeT++/INET [4] simulation environment. TraCI (Traffic Control Interface) client [86] uses SUMO (Simulation of Urban Mobility) [48] with OMNET++ to simulate vehicle to vehicle communications. SUMO utilizes MOVE (MObility model generator for VEhicular networks) [46] as the mobility simulator. The two components of MOVE are a road map editor and a vehicle movement editor. Roadmap editor is used to generate a road map from TIGER (Topologically Integrated Geographic Encoding and Referencing) database or Google Earth files. Movement of vehicles can be generated automatically or manually using the Vehicle

Movement Editor. It specifies the properties of each vehicle such as vehicle speed, duration of trip, origin and destination of vehicle, vehicle departure time, etc. The road map and the vehicle's properties are sent to the SUMO compiler which generates a trace file for the network simulator. This trace file includes the location of each vehicle at every time instant for the entire simulation time and their mobility profiles. This trace file is used in the simulations on OMNeT++. We have performed many sets of experiments to evaluate the performance of our approach in different settings. In the following subsection, we describe, in detail, the different simulation settings and process of traffic simulation in our approach.

4.3.1 Traffic simulation

We have simulated traffic scenarios in two cities of India as our reference. To simulate the traffic of an urban area we have chosen South Delhi whereas for semi-urban area we have chosen Allahabad. The following steps were taken to develop the map and simulate traffic using SUMO:

Fixed nodes and edges: To simulate the traffic we use the Google Earth maps of South Delhi and Allahabad. We mark the major roads and use all the major intersections as fixed nodes (info-stations). We also mark the edges between the fixed nodes.

XML files for Coordinates of nodes and edges : We formulate the coordinates of all fixed nodes and also identify the edges connecting these nodes. In general, we convert the given information into three files, which are used by SUMO as input data. Two of the files contain the network information, to be converted into node and link information in SUMO. Figure 4.5 depicts the map of south Delhi which is taken as reference. Further, Figure 4.6 depicts the snapshot of traffic simulation for the map of south Delhi.

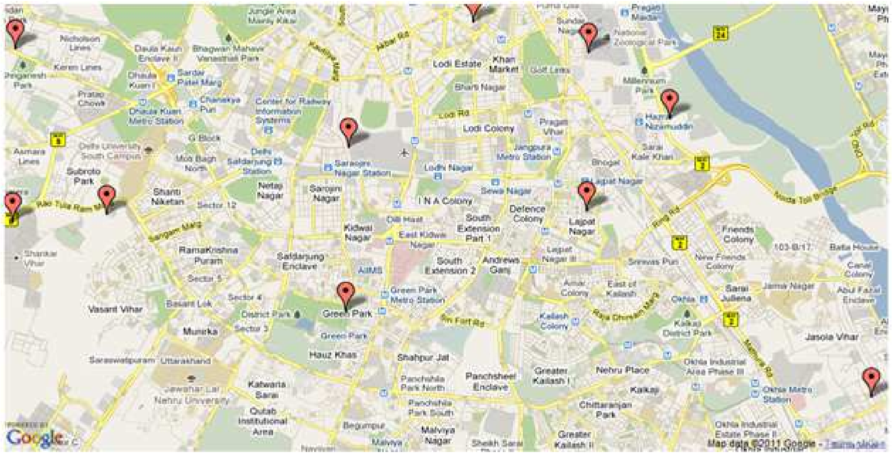


Figure 4.5: Map of South Delhi showing fixed nodes and edges

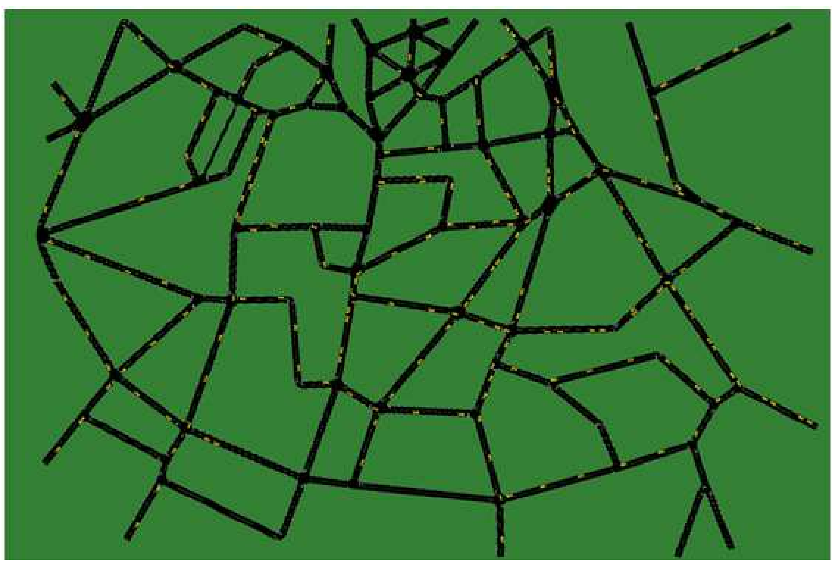


Figure 4.6: Snapshot of the traffic simulation for South Delhi

We divided a simulated day into three traffic types; 07:00-10:00 and 17:00-19:00 (peak traffic), 10:00-17:00 and 19:00-00:00 (moderate traffic), and 12:00 AM to 7:00 AM (low traffic). The simulation settings for vehicles are summarized as follows:

South Delhi (Urban Area): This is a high vehicle density scenario with an organized four-lane road network. Maximum 1000 vehicles may be present concurrently. Maximum speed of vehicles is 60 Km/hour and near intersections (info-stations) the average speed is 30 Km/hour. The vehicle density is dependent upon time of day. Peak traffic density, 800 to 1000 vehicles, moderate traffic density, 300 to 500 vehicles, and low traffic density, 100 to 300 vehicles..

Allahabad (Semi-urban Area): This is a somewhat low vehicle density scenario with a less organized two-lane road network. A maximum of 600 vehicles can be present concurrently. Maximum speed of vehicles is 50 Km/hour and near the intersections (info-stations) average speed is 20 Km/hour. The vehicle density is again dependent upon time of day. Peak traffic, 400 to 600 vehicles, moderate traffic, an average of 300 vehicles, and low traffic, an average of 100 vehicles.

In both of the above settings, publishers and subscribers are distributed randomly among all the vehicles. The maximum percentage of subscriber vehicles is 10% whereas maximum 20% act as publishers. Vehicles utilize a predefined set of matching publications and subscriptions from a file to generate at fixed rate. Simulation results are collected for both the deployment scenarios with two further variations: (1) info-stations are connected to the Internet and (2) info-stations exchange information through vehicles moving between them. In the next subsection we present our simulation results and their analysis. Further, the distribution of the number of vehicles is also of two types: in one set of experiments, vehicles are uniformly distributed across the map whereas in other set vehicle distribution replicates real life situations where some roads near hot spots like metro and rail stations, shopping malls, airports, and some popular junctions experience heavier traffic than other roads.

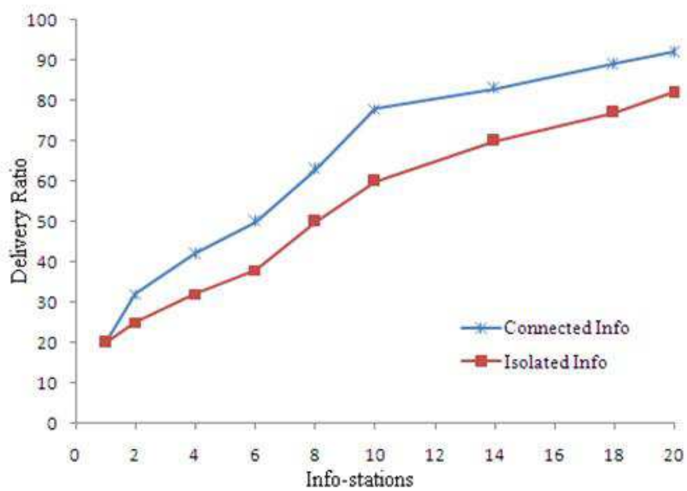
4.3.2 Simulation results

This subsection presents the simulation results. Two sets of results are presented, one with uniform distribution of vehicles and other where the traffic is skewed around hot-spots. These results are gathered for both South Delhi (urban area) and Allahabad (semi-urban area), and for both deployment scenarios; info-station with the Internet connectivity (Connected) and without connectivity (Isolated). In all results, the delivery ratio expressed as a fraction of subscribers successfully receiving the notification with respect to the number of info-stations.

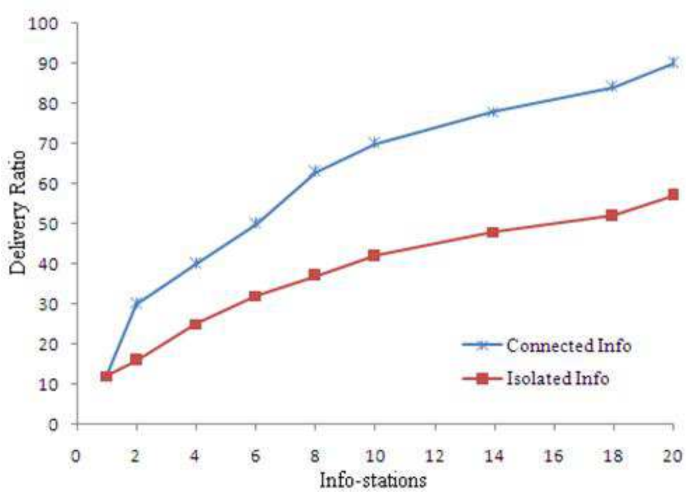
▪ *Uniform distribution of vehicles*

Figure 4.7 and 4.8 depicts the results where a uniform distribution of vehicles is considered. Figure 4.7(a) and 4.7(b) depicts the results obtained for high and low densities of vehicles respectively, in the South Delhi setting. In Figure 4.7(a), it can be observed that when vehicle density is high, the performance of isolated info-station case is somewhat comparable to the connected info-station case. It may also be noted that as the number of info-stations increases, the delivery ratio increases and reaches a maximum of 95% and 89% for connected and isolated info-stations respectively. This suggests that if vehicle density is high then with more info-stations, we can achieve a similar delivery ratio with isolated info-station to the connected info-station case.

The situation is completely different when the density of vehicles is low. Here, the connected info-station scenario performs much better than the isolated info-station case. It may be observed in Figure 4.7(b) that the maximum delivery ratio is not degraded much for the connected info-station case in low density but for isolated info-stations it is reduced by almost 50%. This is due to partitioning of the DHT overlay as there are very few or no vehicles moving between them.



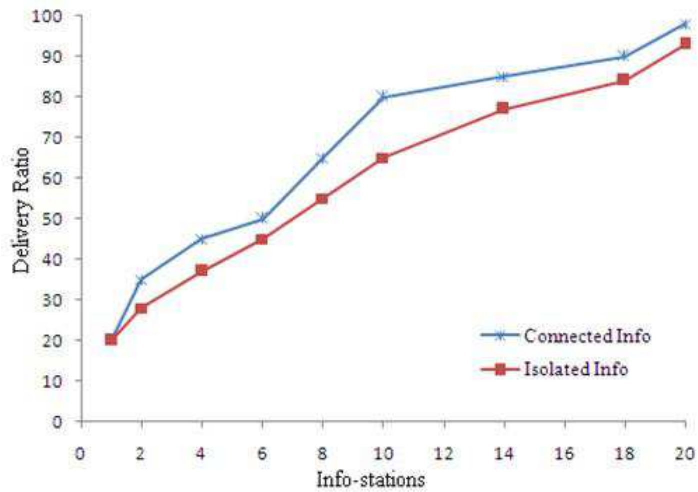
(a) High vehicle density



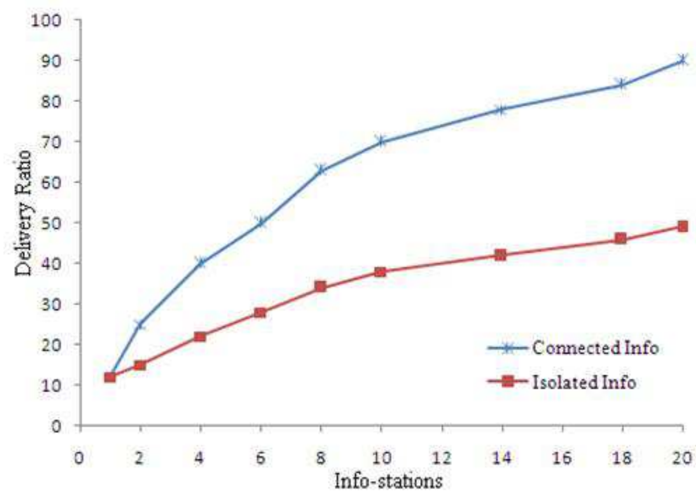
(b) Low vehicle density

Figure 4.7: Delivery Ratio, Urban Area, Uniform distribution of vehicles

Figure 4.8(a) and 4.8(b) depicts the results obtained for high and low density of vehicles respectively, in the Allahabad setting. Here, a similar pattern of results is obtained. In the case of connected info-stations very similar performance is recorded as in the South Delhi setting for both high and low densities whereas for isolated info-stations, performance is much reduced.



(a) High vehicle density



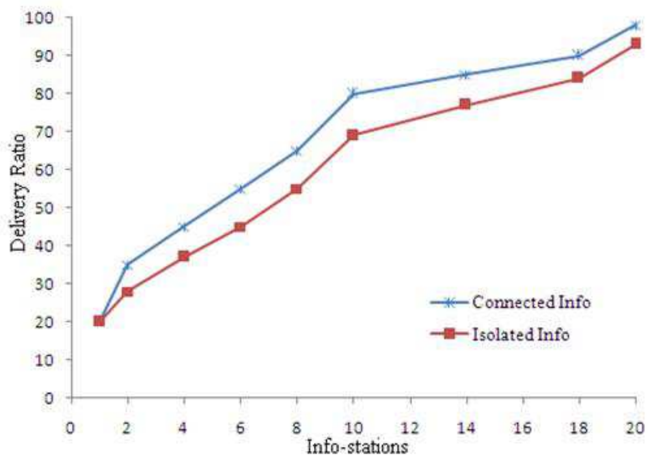
(b) Low vehicle density

Figure 4.8: Delivery Ratio, Semi-Urban Area, Uniform distribution of vehicles

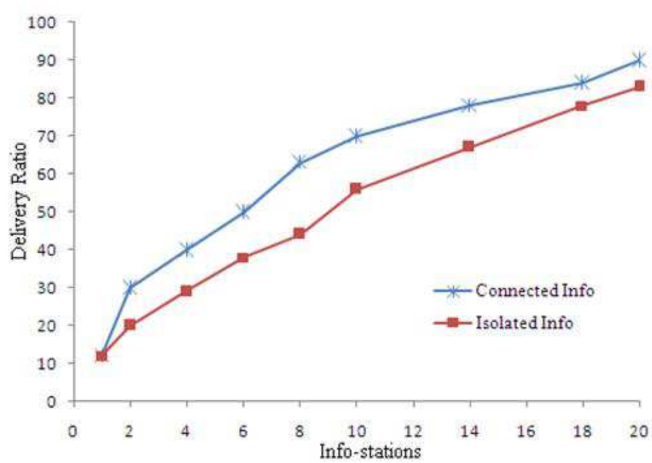
Skewed distribution of vehicles

Figure 4.9 and 4.10 presents the results for the skewed distribution of vehicles. Figure 4.9(a) and 4.9(b) show the results for urban area whereas Figure 4.10(a) and

4.10(b) provide the results for semi-urban settings. We have marked some areas in the map as hot spots with a vehicle distribution such that roads and info-stations around them experience higher traffic density.

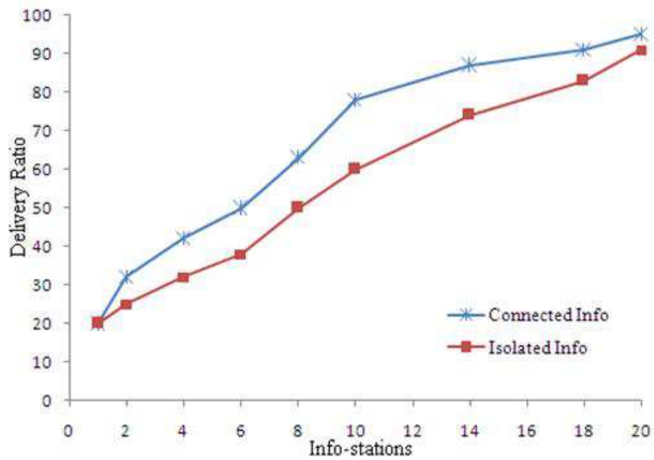


(a) High vehicle density

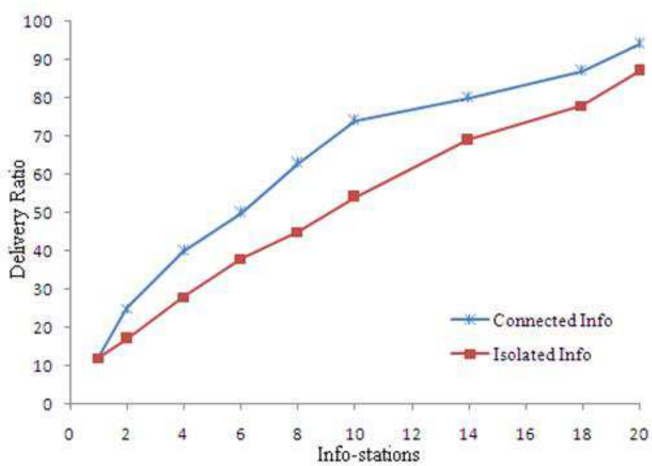


(b) Low vehicle density

Figure 4.9: Delivery Ratio, Urban area, Skewed distribution of vehicles



(a) High vehicle density



(b) Low vehicle density

Figure 4.10: Delivery Ratio, Semi-urban area, Skewed distribution of vehicles

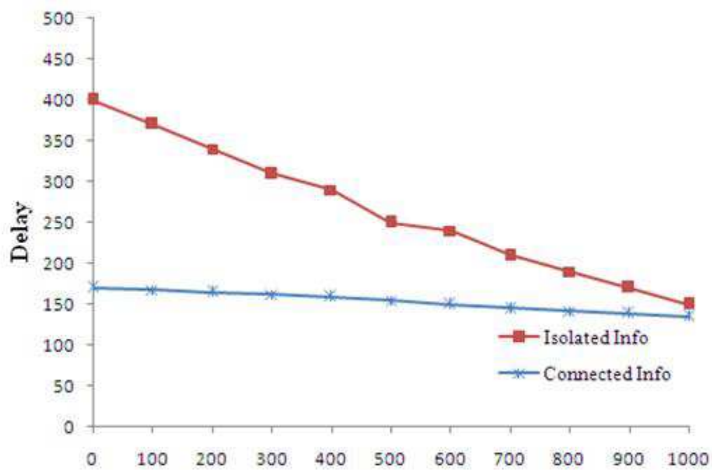
Here, we recorded substantial improvement in the performance of the isolated info-station scenario, both for high and low densities. The improvement is more remarkable in the case of low density, both in the case of urban and semi-urban areas where we observed that delivery ratios almost double with respect to the uniform distribution setting.

The reason behind this remarkable improvement is justifiable. We have taken several snapshots of running simulation to closely observe the DHT formation process among the info-stations. In the Chord DHT protocol, every node periodically runs a stabilization procedure to discover node joining or leaving. In this stabilization procedure, nodes ask their successor to tell the id of its predecessor. In isolated info-station case, the underlying multi-hop communication among vehicles is the only way to send these stabilization messages in DHT layer. If there are no vehicles between two neighbors of the DHT ring, then this results in timeout of the DHT heartbeat messages. Nodes which are not able to reply the stabilization messages due to underlying intermittent connectivity are considered to be out of the DHT ring. As the density of vehicles is skewed around a few clusters of the info-stations, we observe formation of small DHT rings of info-stations around the hotspots. Further, publications and subscriptions happen around those hot spots only due to the presence of most of the vehicles around them. Consequently, we observe increase in delivery ratio as notifications are routed easily towards subscribers due to good connectivity through dense vehicle population and notifications have to travel lesser distance to reach to subscriber.

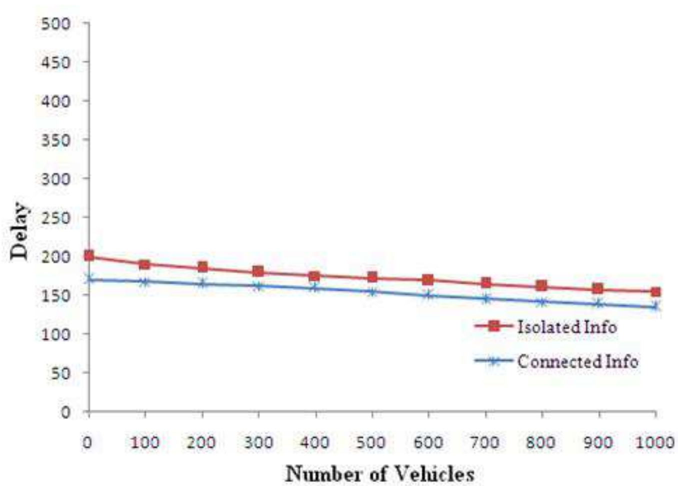
- *Delay in notification delivery*

Figure 4.11(a) and 4.11(b) depict the simulation results for delay in delivering notifications to interested subscribers. These results are collected for the urban scenario where delay in notification delivery is plotted against vehicle density. Figure 4.11(a) provides the results for uniform distribution of vehicles whereas Figure 4.11(b) shows the results for a skewed distribution. It may be noted that when info-stations are potentially connected to the Internet then vehicle density causes very little effect on delay in notification delivery. In the case of isolated info-stations, when uniform distribution of vehicles is considered then delay increases when vehicle density decreases.

On the other hand, when vehicle distribution is skewed then the performance of system is remarkably satisfactory. This is due to both publishers and subscribers may be found in the vicinity of hot spots where smaller DHTs get formed due to vehicle concentration around them.



(a) Urban area, Uniform Distribution



(b) Urban area, Skewed Distribution

Figure 4.11: Delay in Notification Delivery

4.3.3 Discussion

The approach closest to ours is presented in [53]. Although the proposed approach is effective, it depends heavily upon GPS, navigation system and digital maps of cities. They also require highly sophisticated vehicles equipped with these devices. Also, in these approaches info-stations are assumed to be connected to the Internet and a centralized server is required to gather information to build notifications. This requires the pre-deployment of infrastructure to make the design workable. Further, it is desirable to store the relevant information zone-wise and maintain given zone for vehicles to be notified. The subscription has to be routed towards a zone to get matching notifications. This is achieved through the network of info-stations assumed to be connected to the Internet.

We argue that for different application scenarios, the proposed solution in [53] has limitations even if this extensive infrastructure is there. For example, in any urban locality there are parking spots for cinemas, bars etc. Some parking spots are near to points of interest and some are several kilometers away. If vehicles are only notified once they arrive in the vicinity of a point of interest that parking is not available then it would be a significant obstacle for drivers to find alternative parking. If drivers can get information in a timely manner, they can choose another parking location or another cinema altogether. Further, if info-station infrastructure is not there, then vehicles maintain the information by working in collaboration (ad-hoc persistence). In other words, vehicles crossing the affected area or moving nearby keep on transferring the desired publication to each other to maintain it there. This is hugely dependent on vehicle density. For instance, if a vehicle having desired information crosses the area and does not find any other vehicle for some time going towards that area, then some important notifications will be missed.

Finally, the approach in [53] assumes that there is a centralized server attached to the Internet which collects information and then generates notifications. These notifications must be generated by a centralized entity because they also include the location information of persistence area where they must be stored, routed and maintained for a given time. They require infrastructure of info-stations because the publications have to be routed to the centralized server in a timely fashion, notifications have to be generated and then must be forwarded towards the affected area in time. Their approach cannot be solely dependent on an ad-hoc mode of dissemination.

Our approach does not assume that vehicles are fitted with sophisticated devices like GPS-navigation systems. Instead of generating notifications in centralized manner, the underlying DHT is used to route and store related publications and subscriptions close to rendezvous info-stations. In our approach, we relax the requirement of info-stations to be connected to the Internet. Info-stations are just wireless devices with an Omni-directional antenna and some memory. They are not required to have any infrastructure to talk to other info-stations. Instead of the continuous transfer of notifications from one vehicle to another, these low cost info-stations can store the information for a given time in a given area. For [53], one cannot find info-stations everywhere (because that approach assumes that info-stations are connected to the Internet). For our approach, it is just as simple as putting a stationary node into ad-hoc mode, which is less expensive to install and maintain. Our contributions can be listed as follows:

– Results are gathered for two scenarios; info-stations connected to the Internet and isolated info-stations where links are formed in an ad-hoc manner through vehicles moving among them.

– For the connected info-stations case, our approach is somewhat similar to [53] with respect to performance issues like delivery ratio and delay in delivery. This is justifiable as when the info-stations are not leaving and joining dynamically and the link between info-stations are stable then whether they communicate and build notifications in the DHT manner (as in our approach) or by any other manner (as assumed in [53]), there would not be much difference performance wise. This is also evident in our results as compared to performance achieved by [53] in similar simulation scenarios.

– The distinguishing feature of our approach is the solution provided for the scenario where info-stations are not connected. In this case there is an enforced dynamism in the DHT structure of info-stations due to dynamism of ad-hoc links formed by moving vehicles between them. This dynamism is dependent upon the density of vehicles which changes according to the time of day. Accordingly, the DHT of info-stations can be split or different DHTs can be merged together. Interestingly, this merge and split does not have a major impact on the performance when the skewed distribution of vehicles is used in our simulations which models the density of vehicles near hotspots as it happens in real life. This is due to the fact that the vehicles that need information and vehicles that provide information are found to be close to these hotspots.

Chapter 5

DHT of City-buses

This chapter presents the architecture of distributed mobile brokers which are dynamically reconfigurable in form of structured P2P overlay and act as rendezvous points for matching publications and subscriptions. We have taken city-buses in urban settings to act as mobile brokers whereas other vehicles are considered to be in the role of publishers and subscribers. These mobile brokers also assist in locating a vehicle for successful and timely transfer of notifications.

The chapter is organized as follows. Section 5.1 presents the overview of the proposed approach. Section 5.2 presents the system model and assumed scenario. Section 5.3 provides the protocol details. Section 5.4 describes simulation environment and presents the obtained results and Section 5.5 concludes the paper.

5.1 Need of Mobile Brokers

As already discussed in Chapter 3, the existing research efforts [53][56] [52][54] made towards Publish/Subscribe based information dissemination in VANET environment utilize road side info-stations (assumed to be connected to the Internet) to provide infrastructure services whereas ad-hoc vehicle-to-vehicle communication is used to extend the range of these info-stations in areas where infrastructure is not available. In these approaches, either the info-stations are considered to be in the role of brokers or any random vehicle can dynamically become broker at any given time. However, there are some limitations in these approaches which require attention for further scope of improvements:

- For the successful operation of proposed architectures, lots of pre-installed infrastructure is needed. Moreover, these info-stations have to be distributed evenly to cover the entire region under consideration. Otherwise, there may be a situation when information publisher is far away from the interested subscribers and there is no info-station near publisher and/or subscriber. In this situation, getting a suitable broker in time where related publication and subscription can meet is difficult as only vehicle-to-vehicle communication is possible. This may result in loss of important notifications or delayed notifications.

- The strategy of choosing any random vehicle to act as broker may also affect the desired quality of service. As the mobility patterns of the vehicles are unpredictable, there can be a situation where vehicles acting as brokers may take abrupt turn (start moving in opposite direction of subscriber) or may halt which may lead to loss of (or delayed)notifications.

In urban region like Delhi in India, there are lots of city-buses being run by DTC (Delhi Transport Corporation) on various pre-designated routes. These routes are also shared by other vehicles. It is observed that at any given time these buses are distributed across the city in such a manner that they cover almost every part of city. At any route, different buses depart from bus-stops every ten minutes. Depending on the time of day which decides the density of traffic, these buses run with certain average speed and on every route a bus can be spotted after a certain distance. These city-buses can be utilized to act as brokers for other vehicles which are in role of publishers or subscriber. Further, Distributed Hashed Table (DHT) of these city-buses may be formed for disseminating publications and subscriptions.

5.2 Assumed Scenario

Quite similar to the Chapter 4, we have assumed a VANET in urban settings. Each vehicle is equipped with a wireless network interface running IEEE 802.11p with fixed range through which they are able to communicate with each other. The IEEE 802.11p is also known as dedicated short range communication (DSRC) [47] and is an approved amendment to the IEEE 802.11 standard to add wireless access in vehicular environments. The transmission range of vehicles is assumed to be 200m in our approach. These vehicles are assigned with unique identification number. Further, some designated vehicles (city-buses) are potentially connected to the Internet and have underlying IP based communication channel among them, for example by utilizing infrastructure-based cellular communication like UMTS (Universal Mobile Telecommunication System). These city-buses act as brokers for other vehicles that are in role of publishers or subscriber at any given time. It may be noted that apart from city-buses no other normal vehicle is equipped with Internet connectivity.

Figure 5.1 depicts the assumed scenario where city-buses which have fixed timings and generally constant average speeds act as mobile brokers. These mobile brokers act as rendezvous points for publications and subscriptions. These are connected to the IP based Internet and form Chord [80] like DHT overlay among them. The IP address associated with each city-bus is hashed to find out their logical

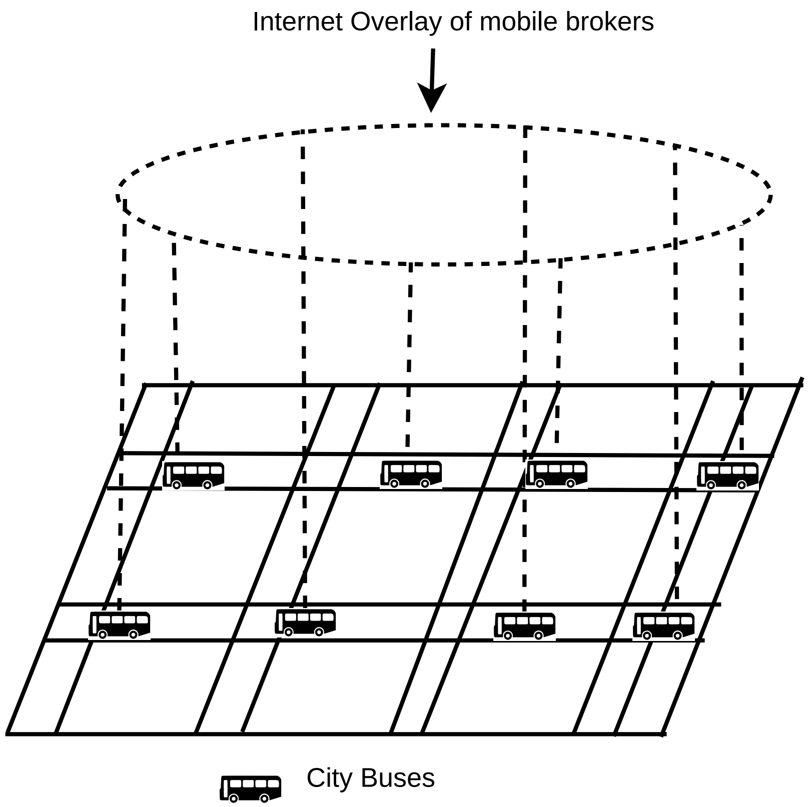


Figure 5.1: DHT of Mobile Brokers (City Buses)

placement in overlay ring.

In the proposed approach, a vehicle takes the role of publisher when it generates any information. The information can be categorized under set of predefined topics like *Traffic Diversions*, *Parking Space* or *Road Jams* etc. Similarly, a vehicle may act as a subscriber and express its interest in receiving information related to these topics.

The related publications and subscriptions are routed by underlying Chord DHT of city-buses to meet in at least one broker termed as rendezvous broker. The associated topic names with publications and subscriptions are hashed to get their identifiers. Subsequently, they are routed to and stored at the broker which is the immediate successor of their identifiers. The rendezvous broker performs matching of arriving publications and subscriptions. Matching publications are routed from rendezvous broker towards a broker (using DHT routing mechanism) that is nearest to the current location of subscriber-vehicle. This broker notifies the matching publication to intended subscriber-vehicle using ad-hoc vehicle-to-vehicle communication.

5.3 Protocol Details

The proposed protocol design consists of various procedures. These procedures include:

- DHT formation of mobile brokers.
- Publication/subscription routing and storage.
- Locating the subscriber vehicle.
- Notification routing and delivery.
- Opportunistic delivery of notification.

As discussed in the approach presented in Chapter 4, common data structures maintained at each ordinary vehicle are *Subscription-Table*, *Publication-Table* and *Forwarding-Table*. These three data structures are used here also. Apart from these, *Last-mobile-broker* is also used which stores the identification of last city-bus (with the time-stamp) from which the contact has been established. Similarly, city-buses also maintain *Subscription-Table* and *Publication-Table*. Additionally, the city-buses also maintain a *Location-Table* which stores the location information of subscriber vehicles they are responsible for.

The publish primitive in our approach is defined as:

$$Publish(publication\_specification, TTL, max\_hop)$$

Here, TTL is the time for which a publication can be considered to be active. The subscribe primitive is defined as:

$$Subscribe(Subscription\_specification, subscriber\_id, TTL, max\_hop)$$

Here *subscriber_id* is identification number of the vehicle which has issued the subscription and TTL is the time for which the subscriber should get the matching notification with respect to its subscription.

It may be noted that both publish and subscribe primitive include *max_hop* in their description. This represents the maximum number of hops a publication or subscription may take when they get forwarded in ad-hoc fashion from vehicle-to-vehicle. In other words, this also represents that at most in these many hops any mobile broker may be reached. This is required to prevent continuous vehicle-to-vehicle flooding of subscriptions and publications.

5.3.1 DHT Formation of Mobile Brokers

The proposed P2P broker overlay is based on Chord DHT [80]. We assume the presence of a bootstrap server for initialization purpose only. It maintains a list of few city-buses recently joined in the system and provides their IP addresses to newly arriving city-buses. The city-bus IP addresses can be hashed to a universal identifier space using SHA-1. The uniqueness of city-bus IP addresses and consistent hashing mechanism of SHA-1 ensures that there is less probability of collisions while assigning hashed identifiers.

A city-bus attempts to become a part of broker overlay after it starts its journey for a particular route i.e. when it starts from its first stop. Similarly, it detaches itself from the overlay when it reaches its last stop and halts for a stipulated time. The DHT application running on city-buses sends hashed value of vehicle identifier to bootstrap server. Bootstrap server keeps record of few mobile brokers which are already in operation and details of these few currently active brokers are provided to a city-bus willing to join. This newly arrived city-bus forwards the joining request to one of the existing active broker using its Internet interface. Subsequently, the joining request is routed using DHT routing mechanisms and new city bus is logically attached to broker overlay at appropriate place. The process of mobile-broker joining is summarized in Algorithm 5.1.

Algorithm 5.1 Joining Procedure of Mobile Brokers

1: Newly arrived mobile-broker (MB_{new}) sends request to bootstrap server.
2: Bootstrap server replies with the details of an existing mobile-broker (MB_{exist}).
3: MB_{new} requests MB_{exist} to find the successor of MB_{new} in overlay.
4: MB_{new} is attached by setting its successor link to point to the successor.
5: MB_{new} sets its predecessor link to nil.
6: MB_{new} builds its finger table with the help of its successor.

After the execution of above steps, newly joined mobile broker MB_{new} has its successor link and finger tables updated. Still, some tasks are required to be performed for the correct operation of overlay structure:

- Setting up the predecessor link of newly joined broker MB_{new}.
- Updating fingers, successor links and predecessor links of existing brokers in overlay, affected by joining of MB_{new}.
- Transfer of content objects to MB_{new} from its successor broker.

These tasks are handled by Chord DHT overlay maintenance procedures described in Chapter 2. Every node in DHT executes a *stabilization*() procedure periodically. In this procedure, node sends a request to its successor, and asks about the predecessor of its successor. In this manner, two adjacent nodes are able to know if any new node has joined between them. Further, every node executes a fix-finger procedure periodically to update its finger-table.

When a mobile broker leaves the network, then also some procedures have to be followed to make the DHT structure intact. The leave operation is triggered when a city-bus is no longer useful in disseminating information. This can happen when it reaches its last stop of route or it gets halted due to any other reason for more than a threshold time limit. Each node in DHT periodically runs a *check_predecessor*() procedure. If there is no response from the predecessor then node sets its predecessor as nil. Thereafter, the *stabilization*() procedure can set up the predecessor and successor links of existing nodes accordingly. The details of *check_predecessor*() are provided in Chapter 2.

5.3.2 Publication/Subscription Routing and Storage

Generally, the primary goal of vehicles is to transfer their publications or subscription to the nearest mobile broker. For this, vehicles utilize vehicle-to-vehicle forwarding mechanism over ad-hoc communication links established among them. Every city-bus periodically broadcast a control message in its proximity to inform vehicles of its presence. Once the publications or subscriptions reach at any mobile broker they are routed to their respective rendezvous broker through the Internet level DHT formed among them. Algorithm 5.2 outlines the steps performed by a mobile broker to route publications or subscriptions.

For example, suppose a vehicle issues a publication which has following specification:

Topic= Road Jam; Loc= MG Road; Time to clear= 1/2 hr; Timestamp=1200 hrs

Algorithm 5.2 Routing and Storage of Publication/Subscription

1: For every publication (P) or subscription (S) received:
2: Extract the value of topic attribute (t) from their specification
3: $k = hash(t)$
4: Use finger-table to know the closest successor mobile-broker of k.
5: Send P or S to the closest successor mobile-broker in the DHT.
6: Repeat steps 4 and 5 until P or S reaches at the immediate successor of k.

As this publication reaches any mobile broker, the value of topic name is hashed and using this hashed publication identifier, the publication is routed recursively from broker to broker. Finally, it is routed to and stored at the broker whose identifier is immediate successor of publication identifier.

Now suppose there is a vehicle that is travelling towards some destination and MG road comes in its route. Before embarking, the vehicle may issue a subscription which has following specification:

Topic=Road Jam; Route= Airport-MG Road-Nehru Place; Timestamp=1155 hrs

Clearly, the hashed subscription identifier is going to be equal to the hashed publication identifier described earlier. This is due to the consistent hashing mechanism of SHA-1 which creates same hashed values if the topic names are same. Consequently, the subscription is also routed to the mobile broker which stores the publication of same topic name.

This mobile-broker acts as a rendezvous point and performs matching operation of publication and subscription looking at the values of other attributes too. In this case, a notification will be generated because MG Road is coming in the route of subscriber vehicle. Further, the timestamp value is also indicating that the subscriber should divert its route for the desired driving comfort and to avoid further traffic congestion on MG Road.

5.3.3 Locating the Subscriber Vehicle

The major challenge in the proposed approach is to locate the subscriber at the time of notification delivery. As the vehicles subscribe for items while moving, there may be a situation that they issue subscription in one region of city and at the time when notification is ready, they are in another region. For successful reception of notification it is essential to locate the subscriber at any given time. In our approach, the location information of subscriber vehicle is maintained and updated in distributed manner over DHT of city-buses. Algorithm 5.3 outlines the steps performed by each subscriber vehicle for updating its location.

Algorithm 5.3 Locating Subscriber Vehicle

Extract TTL value from subscription specification
Repeat until TTL expires OR notification received:
if Last-mobile-broker == broadcasted identifier of city-bus (CB_{ID}) **then**
    No Location Update
else
    // Trigger Location Update
    Last-mobile-broker= broadcasted identifier of city-bus (CB_{ID})
    $k = hash(Subscriber - VehicleIP)$
    send k and location update request to the city-bus (CB_{ID})
    City-bus uses DHT routing to update location in the Location-Table at the immediate successor k.
end if

As already mentioned before, every vehicle maintains a variable *Last-mobile-broker* which stores the identification of last city-bus (with timestamp) from which the contact has been established. As vehicle move, it comes in proximity of other city-buses. This change of location can be known by comparing the stored city-bus ID with the broadcasted city-bus ID. If they do not match then location update is triggered. The value of *Last-mobile-broker* is modified and $hash(Vehicle - IP)$ is calculated. This hash value and location update request is forwarded to the city-bus in the proximity. Thereafter, the contacted city-bus uses DHT routing substrate to update the location in the *Location-Table* at the city-bus which is the immediate successor of $hash(Vehicle - IP)$. As a result, certain city-buses in DHT are made responsible for maintaining location details of a set of subscriber vehicles. This process gets repeated until the desired notification is received or subscription is invalid due expiration of TTL.

The location information is maintained only for those vehicles that are in role of subscriber and their subscriptions are active. To know the location of any subscriber vehicle, the *Location-Table* maintained at the immediate successor of $hash(Subscriber - VehicleIP)$ is looked up in the DHT of city-buses. It may be noted that in our design, subscriber vehicles are not required to be located exactly in a smaller region. Our approach roughly locates the vehicle between two city-buses on a specified route.

5.3.4 Notification Routing and Delivery

As already discussed, publications and subscriptions related to same topics are routed and stored at same mobile-broker. Further, other attributes are utilized for their fine grained matching. Each arriving publication and subscription is matched with the subscriptions and publications already stored in *Subscription-Table* and *Publication-Table* respectively. In these tables, subscriptions and publications are listed topic wise.

As publishers and subscribers are purely decoupled, publications and subscriptions can be generated at any time and in any order. Successful notifications strongly depend on ordering of occurrence of publications and subscriptions, the time instant they reach at broker and their lifespan. In our approach, both publications and subscriptions have definite time span. These time spans are provided as TTL values in descriptions of publications and subscriptions. Even if the subscription arrives at the rendezvous broker after the publication, subscriber can be notified if subscription-lifespan and publication-lifespan intersect with each other. Algorithm 5.4 outlines the steps performed to deliver the notification to subscriber vehicle.

Algorithm 5.4 Notification Routing and Delivery

1: Extract $Subscriber - IP$ from subscription specification
2: $k = hash(Subscriber - IP)$
3: Examine the *Location-Table* at the mobile-broker that is immediate successor of k.
4: Retrieve the location information (ID of mobile-broker) of subscriber from Location-Table
5: Use DHT to route the notification to mobile-broker found in Location-Table.
6: Use ad-hoc vehicle to vehicle routing to forward notification from mobile-broker to subscriber-vehicle

To forward a notification, a query is sent via DHT lookup procedure to the mobile-broker which currently maintains the location details of a given vehicle. This mobile-broker gives the identifier of last-mobile-broker contacted by the subscriber-vehicle. The notification is forwarded to that last-mobile-broker using underlying DHT routing mechanism. This last-mobile-broker forwards the notification utilizing ad-hoc routing between other vehicles towards the target subscriber-vehicle.

5.3.5 Opportunistic Delivery of Notification

As discussed earlier, vehicles which need to publish or subscribe forward their publications or subscriptions to their neighbors in 1-hop communication range. These neighbors further forward all the received publications and subscriptions to their 1-hop neighbors. This process continues till the publications or subscriptions reach at a mobile-broker. This hop-by-hop transfer of publications and subscriptions towards mobile-brokers enables the other forwarder vehicles to act as opportunistic brokers at any given time. This process is quite similar to the steps outlined in Algorithm 4.2 presented in Chapter 4 where the hop-by-hop transfer of publications and subscriptions continues till they reach at any info-station.

Each ordinary vehicle maintains *Subscription-Table* and *Publication-Table* for storing its own active subscriptions and publications. Further, these vehicles also maintain *Forwarding-Table* for storing subscriptions and publications of other vehicles to forward them further. After receiving publication, subscription or notification for forwarding, each vehicle matches them with already stored publications and subscriptions. Every received publication is matched with the subscriptions in *Subscription-Table* and *Forwarding-Table*. If the received publication matches with any subscription of *Subscription-Table* then it is treated as notification for the vehicle itself. If received publication matches with any subscription in *Forwarding-Table* then the notification can be generated by the vehicle and forwarded to the subscriber-vehicle. Similarly, opportunistic notifications can be generated by ordinary vehicles when they receive and notification or subscription for forwarding.

5.4 Simulation Setup and Results

Similar to simulation setup explained in Chapter 4, we have simulated our approach using Oversim [5] over OMNET++/INET [4]. We have used Chord module of Oversim for simulating DHT of city-buses. To generate realistic vehicle movements we have utilized MOVE (MObility model generator for VEhicular networks) [46]. MOVE is built on top of micro-traffic simulator SUMO (Simulation of Urban MObility) [48]. To establish communication between SUMO and OMNET++/INET we have utilized TraCI (Traffic Control Interface) [86].

We have chosen OMNET++ because it provides the implementation of the IEEE

802.11p standard which is a recommended protocol for vehicular environment. The important reason behind choosing MOVE is that it provides a GUI for simulating the movement of city-buses. Using this GUI the routes of buses, their departure time, speeds, inter-bus interval etc. can be defined easily to simulate mobility pattern of city-buses. Further, MOVE provides interfaces for realistic road map generation from real world map databases like TIGER (Topological Integrated Geographical Encoding and Referencing) database or Google Earth. TraCI provides a TCP based client-server architecture. MOVE/SUMO acts as TraCI server whereas OMNET++/INET act as TraCI client to exchange commands using TCP connections between them.

We have compared our design with an approach closest to ours presented in [53][56] [52][54]. Here, two reference schemes are discussed namely infrastructure persistence and ad-hoc persistence. In infrastructure persistence, notifications are stored on road-side info-stations and subscriptions are routed towards them. These info-stations are assumed to be connected to a central server which collects publications and issues notifications. In ad-hoc persistence, vehicles collaborate with each other to store publications, and subscriptions are flooded towards them in hop-by-hop manner. Essentially, publish/subscribe paradigm is implemented over infrastructure of info-stations in one scheme and over ad-hoc network of moving vehicles in another. Henceforth, the info-station scheme and the ad-hoc scheme are referred as ***Comparison-Scenario-1*** and ***Comparison-Scenario-2*** respectively.

5.4.1 Simulation Parameters

We have performed simulation taking into reference the South Delhi area. This area has organized four lane road network. We have utilized the Map Editor provided by MOVE which generates real world map for simulation from Google Earth KML

(Keyhole Markup Language) files. Further, we have used Vehicle Movement Editor of MOVE to specify the properties of vehicles like vehicle speed, duration of trip, origin and destination of vehicle, vehicles departure time etc. The simulation parameters are provided in Figure 5.2.

1. Traffic Type		
Vehicle Density	**Time of the Day**	**No. of Vehicles**
Peak Traffic	07:00-10:00 and 17:00 19:00	800 to 1000
Moderate Traffic	10:00-17:00 and 19:00-00:00	300 to 500
Low Traffic	00:00-7:00	100 to 300
2. Vehicle Properties		
Maximum speed of ordinary vehicle = 65 Km/Hour		
Maximum speed of City Bus = 45 Km/Hour		
City Bus pause time at bus stop = 3 minutes		
No. of City Bus = 15% of total vehicles		
Maximum number of Publishers at a time: 20% of total vehicles		
Maximum number of Subscribers at time: 40% of total vehicles		
3. Ad-hoc Communication Properties		
Protocol used: 802.11p		
Transmission Range of Vehicles: 200 meters		
Identification Broadcast period of City Bus: 5 seconds		
4. DHT Communication Properties		
Protocol used: Chord, Underlying Protocol: TCP/IP		
Fix-Finger Period: 5 minutes		
Stabilization Period: 2 minutes		
5. For Comparison Scenario-1		
Central Server: one		
No. of City Buses : zero		
No. of info-stations: variable		
Transmission Range of info-stations: 200 meters		
Protocol used (for ad-hoc communications): 802.11p		
Protocol used (for Infrastructure communications): TCP/IP		
6. For Comparison Scenario-2		
No. of City Buses : zero		
No. of info-stations: zero		
Protocol used (for ad-hoc communications): 802.11p		
Protocol used (for Infrastructure communications): TCP/IP		

Figure 5.2: Simulation Parameters

We have simulated the vehicle densities according to realistic traffic situations. For example, during morning and evening hours, traffic density is more due to large number of office goers. Similarly, in night hours traffic density is low. Further, some areas can be considered as hot spots during overall low traffic density period. It is observed that around movie halls, airports, railway stations, night clubs etc. traffic density is much more during off hours too. So the distribution of vehicle in the city is normally not uniform. This means that at some places traffic density is high whereas at other places it is low. This is termed as skewed-distribution of vehicles in our approach.

The maximum speed of normal vehicles is 65 km/hour whereas the maximum speed of city-buses is 45 km/hour. The stoppage time of city-buses at bus stops is three simulated minutes. Number of city-buses is 15% of total ordinary vehicles at any given time. These buses run according to their fixed routes. The stoppage time of city-buses at bus stops is three simulated minutes. The routes of these buses in simulation are specified according to their actual routes information collected from Delhi Transport Corporation [8].

All the ordinary vehicles and city-buses have wireless interfaces running IEEE 802.11p protocol. The transmission range of each vehicle is set as 200 meters. City-buses also have wireless Internet interface through which they form Chord DHT among them. The parameters for Chord DHT are provided in Figure 5.2. City-buses send heartbeat messages at every 5 seconds to broadcast their presence to nearby vehicles.

Comparison-Scenario-1: For this scenario, static info-stations are created at major intersections in the simulated area of South Delhi. These are also created using Vehicle Movement Editor of MOVE by setting the maximum speed of vehicle (which act as info-station) as zero. In addition to the wireless interface through which info-stations interact with other vehicles, these are also connected with a central server using TCP/IP interface. The number of city-buses is set to zero in this scenario.

Further, the subscription specification in this scenario also includes location of vehicle, future navigation route and destination.

Comparison-Scenario-2: For this scenario, both number of city-buses and number of info-stations are set as zero. Other simulation parameters are similar to *Comparison-Scenario-1*.

In our approach and in both the other scenarios, any vehicle randomly can take the role of publisher or subscriber. Generally, in realistic situations publishers are less in number than the subscribers. It is assumed that the maximum number of publishers can be 20% of total vehicles whereas maximum 40% can be in role of subscriber. Publications and subscriptions are generated by vehicles at fixed rate by randomly choosing from a predefined set. This set contains matching publication and subscription specifications. Further, the rate of subscription is set to be higher than rate of publication.

5.4.2 Simulation Results

This subsection presents the simulation results. Results are obtained and compared for our approach, *Comparison-Scenario-1* and *Comparison-Scenario-2* with respect to following parameters:

1. Notification delivery ratio: This is the ratio of the number of subscriptions for which the successful notifications are received to the total number of subscriptions issued by the subscribers during a given time interval.

2. Notification delay: This is the time required to deliver a notification successfully to intended subscriber.

Further, these results are collected and compared under different simulation settings such as skewed and uniform distribution of vehicles in the simulated region.

- *Evaluation of our approach against* Comparison-Scenario-1

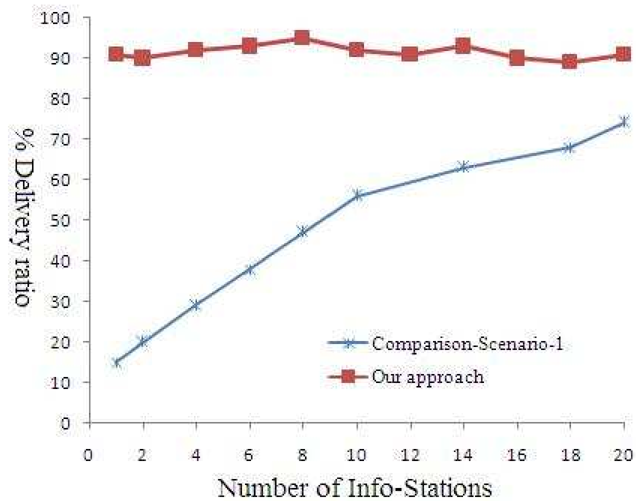
Figure 5.3 and 5.4 depicts a set of simulation results where our approach is compared with *Comparison-Scenario-1* with respect to delivery ratio. Here x-axis and y-axis represents the number of info-stations and delivery ratio respectively. The delivery ratios shown in results are for successful notifications. As discussed earlier, the subscription and publication both have their associated validity durations. The notification is discarded in route if the validity period is over.

Figure 5.3(a) shows the results for peak traffic conditions whereas in Figure 5.3(b), results are presented for moderate traffic conditions. Figure 5.4(a) and 5.4(b) show the results for low traffic conditions. In Figure 5.4(a) vehicles are uniformly distributed across all the roads in simulation area while in Figure 5.4(b), their population is more skewed towards some hot-spots in city such as bars, movie theatres etc. It may be noted that in all the traffic conditions, our approach performs better than *Comparison-Scenario-1*.

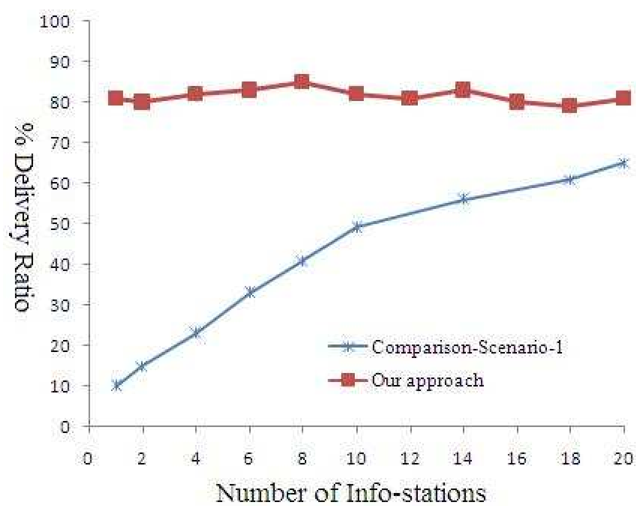
In peak traffic conditions our approach provides delivery ratio of 89-94% whereas under same conditions *Comparison-Scenario-1* gives maximum delivery ratio of 74% when 20 info-stations are there in simulated region. This delivery ratio goes down to 47% and 56% when there are 8 and 10 info-stations. Similar patterns may be observed under moderate and low traffic conditions.

Under moderate traffic conditions, our approach gives delivery ratio of 79-85% whereas maximum delivery ratio for *Comparison-Scenario-1* comes as 65% when number of info-stations are 20. Under low traffic conditions with uniform distribution of vehicles, our approach performs between 65-72% whereas maximum delivery ratio for *Comparison-Scenario-1* is 45%. Similarly, under low traffic conditions with skewed distribution of vehicles, our approach gives delivery ratio between 60-67%

compared to maximum delivery ratio of 37% achieved in *Comparison-Scenario-1*.

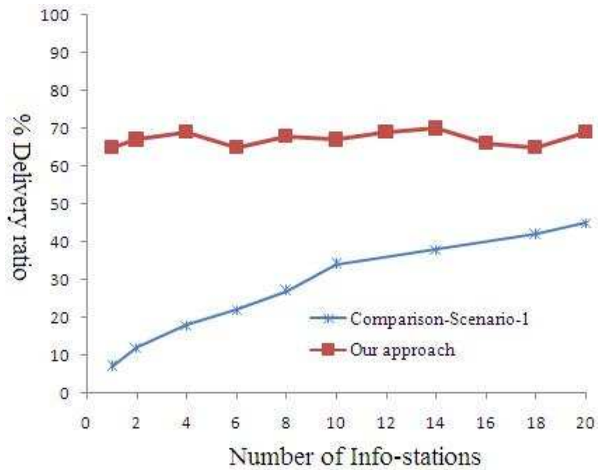


(a) Peak Traffic

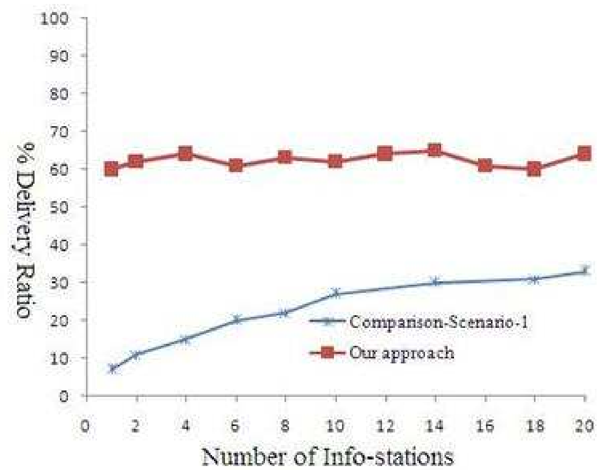


(b) Moderate Traffic

Figure 5.3: Delivery Ratio, Our Approach Vs Comparison Scenario-1



(a) Uniform Distribution

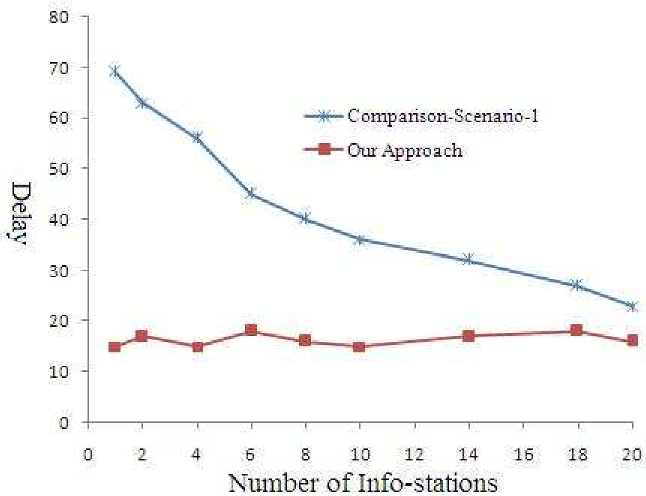


(b) Skewed Distribution

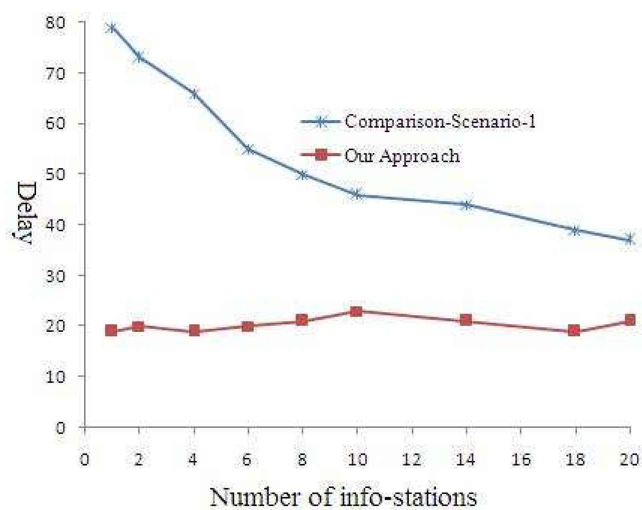
Figure 5.4: Delivery Ratio, Low Traffic, Our Approach Vs Comparison Scenario-1

Figure 5.5 and 5.6 depicts a set of simulation results where our approach is compared with *Comparison-Scenario-1* with respect to delay in notification delivery. Here x-axis and y-axis represents the number of info-stations and notification delay respectively.

Figure 5.5(a) shows the results for peak traffic conditions whereas in Figure 5.5(b), results are presented for moderate traffic conditions. Figure 5.6(a) and 5.6(b) show the results for low traffic conditions where under uniform and skewed distribution respectively. It may be noted that in all the traffic conditions, our approach performs better than *Comparison-Scenario-1*. Further, it may be observed that to reduce the notification delay, *Comparison-Scenario-1* requires more info-stations.

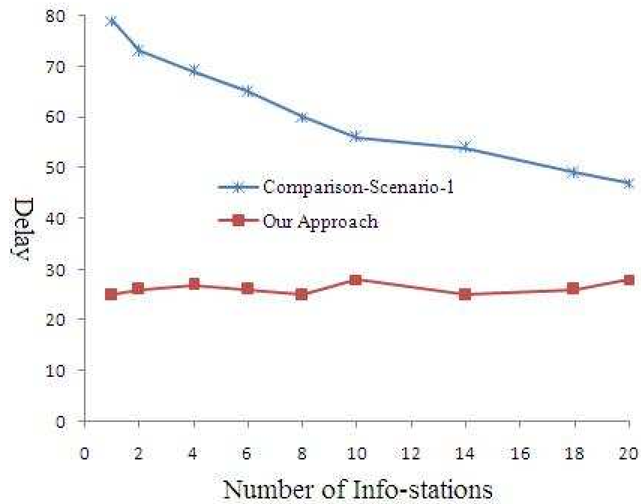


(a) Peak Traffic

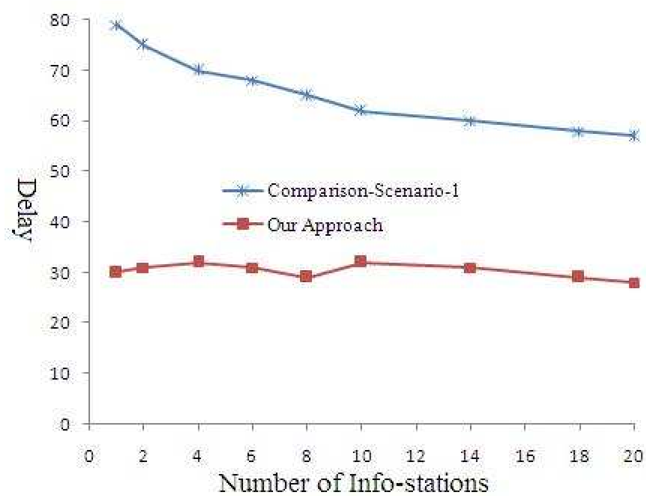


(b) Moderate Traffic

Figure 5.5: Notification Delay: Our Approach Vs Comparison Scenario-1



(a) Uniform Distribution



(b) Skewed Distribution

Figure 5.6: Delay,Low Traffic,Our Approach Vs Comparison-Scenario-1

In peak traffic conditions, our approach provides notification delay of 15-18 seconds whereas under same conditions *Comparison-Scenario-1* gives minimum delay of 23 seconds when number of info-stations are 20. This delay is increased to 40 and 36 seconds when there are 8 and 10 info-stations. Under moderate traffic conditions, the notification delay in our approach 19-23 seconds while minimum delay in *Comparison-Scenario-1* is 37 seconds. Under low traffic conditions (uniform distribution), the notification delay in our approach is in between 25-28 seconds while for *Comparison-Scenario-1*, the minimum delay is 47 seconds. Similarly, under low traffic conditions (skewed distribution), the notification delay in our approach is in between 30-34 seconds while for *Comparison-Scenario-1*, the minimum delay is 57 seconds.

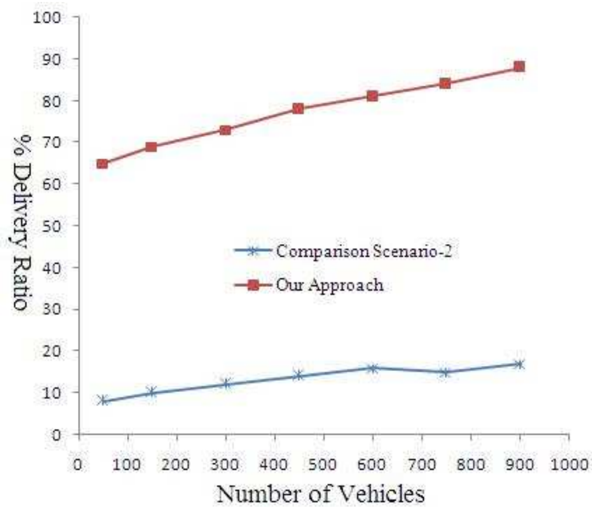
Observations: It can be observed from Figure 5.3, 5.4, 5.5 and 5.6 that the approach presented in *Comparison-Scenario-1* relies heavily on the number of info-stations. It performs better with more number of info-stations but performance goes down drastically when the number of info-stations is less. This suggests that a lot of pre-installed infrastructure is required to make this approach applicable with desired quality of service.

Further, even when the number of info-stations is substantially large, the performance of *Comparison-Scenario-1* is low compared to our approach. The reason is that vehicles have to move in the transmission range of stationary info-stations to transfer publications or subscriptions. In the case of city-buses acting as brokers, both ordinary vehicles and city-buses are mobile and transfer delay can be reduced to a large extent if they move towards each other.

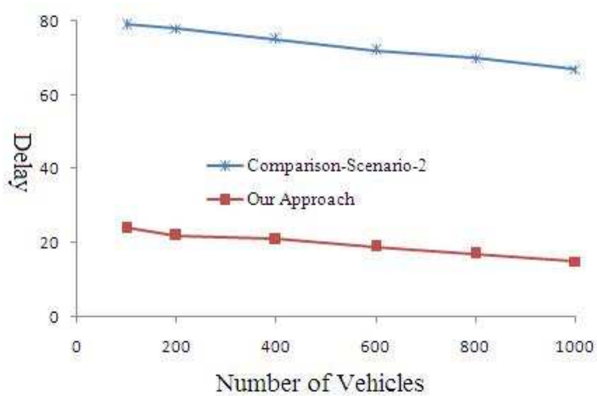
The performance of *Comparison-Scenario-1* degrades further at moderate and low traffic densities. This is because when the number of vehicles is less, it results in a longer delay before a vehicle can find another vehicle in its transmission range.

Consequently, hop-by-hop ad-hoc transfer takes relatively longer time.

- *Evaluation of our approach against Comparison-Scenario-2*



(a) Delivery Ratio



(b) Notifiation Delay

Figure 5.7: Our Approach Vs Comparison-Scenario-2

Figure 5.7(a) and 5.7(b) depicts simulation results where our approach is compared with *Comparison-Scenario-2*. In Figure 5.7(a), the comparison is shown with respect to delivery ratio. Here, x-axis and y-axis represents the number of vehicles and delivery ratio respectively. In Figure 5.7(b), the comparison is depicted with respect to notification delay. Here, x-axis and y-axis represents the number of vehicles and delay in notification delivery respectively.

It may be noted that our approach performs lot better than pure ad-hoc approach of *Comparison-Scenario-2*. The delivery ratio in our approach is 65-89% compared to 8-17% of *Comparison-Scenario-2*. Similarly, notification delay in our approach is in between 15-24 seconds compared to 79-67 seconds of *Comparison-Scenario-2*. This is due to the strategy of random assignment of broker role to any vehicle in this scenario. The unpredictable mobility behavior of an ordinary vehicle in role of broker (which cannot be controlled by any external entity) results in notification loss and delay in delivery.

Chapter 6

2-Tier DHT of Info-stations and City-buses

The approaches described in Chapter 4 and 5 are quite feasible to implement. However, there are certain important trade-offs which has to be given consideration for their successful deployment. In the assumed scenario of Chapter 4, if number of info-stations is less then there would be bigger distance among them. In that case, there is a chance that hop-by-hop ad-hoc transfer of publications and subscriptions towards info-stations are more dominant instead of through DHT based routing. This may result in loss/delay of important notifications. Consequently, lot of pre-installed infrastructure is required for achieving a desired quality of service.

On the other hand, in the system explained in Chapter 5 (city-buses as brokers) does not require any availability of pre-installed infrastructure. However, the quality of service in this approach is largely dependent on the number of buses, their routine and their distribution across the city. It may be observed in a busy city like Delhi that buses are not able to follow their routines strictly due to dynamic and ever changing traffic conditions. There can be a situation where some buses move in a cluster having less distance with each other whereas there is a substantial distance between some of the buses. This uneven distribution of buses may hamper the performance of system.

Further, on major routes connecting important destinations of city, number of buses are generally more and any vehicle may locate a city-bus in few hops. But this is not the case on the roads which connects urban areas which are far from each other in a big region. For example, National Capital Region (NCR) of Delhi includes the townships of Noida, Faridabad, Ghaziabad and Gurgaon. In Delhi and Noida number of buses and bus-stops are more with respect to the total distance covered. However, the number of buses and bus-stops are found to be less when one wants to travel from Delhi to Noida connecting road. In this situation, it may be difficult to find city-buses which are close to each other at any given time.

The limitations of approach presented in Chapter 4 which require lot of infrastructure and the approach presented in Chapter 5 with no infrastructure, has given motivation to think about a hybrid scenario which has both fixed info-stations and city-buses working as brokers. This chapter presents the design of publish/subscribe framework over two-tier DHT formed by info-stations and city-buses.

The chapter is organized as follows. Section 6.1 provides the system description. Section 6.2 presents the details of algorithms utilized in proposed framework. Section 6.3 presents simulation details and results.

6.1 System Description

Similar to Chapter 5, a VANET in urban settings is considered where each vehicle is equipped with a wireless network interface running IEEE 802.11p with fixed range. The transmission range of vehicles is assumed to be 200m. The city-buses are assumed to be connected to the Internet by utilizing infrastructure-based cellular communication, like UMTS. The important distinction from the system model of Chapter 5 is that these city-buses are also assumed to be equipped with GPS device. The info-stations are present at major locations of city. These info-stations are also connected to the Internet. The underlying communication among them is through the IP based Internet.

Figure 6.1 depicts the design of two-tier DHT formed by info-stations and city-buses. To form this two-tier DHT, the whole city region is divided into sectors where each sector has its own unique identification number. The installation plan of info-stations is made in a way to ensure that every sector contains only one info-station. The info-station present in a sector takes the role of super-peer. All these super-peers present in different sectors are connected in DHT manner and form the upper-tier of two-tier DHT. These super-peers are logically connected to city-buses currently moving in their respective sectors which also form a DHT among themselves. A city-bus while entering a sector joins the lower-tier DHT formed there. Similarly, city-buses leave the lower-tier DHT when they move to another sector. It may be noted that city-buses are equipped with GPS and they can know about their current position while on move and ascertain that in which sector they are currently moving.

6.2 Description of Procedures

The common data structures maintained at info-stations and vehicles are same as described in Chapter 4 and 5. These are *Subscription-Table*, *Publication-Table*, *Forwarding-Table*, *Last-mobile-broker* etc. Further, publish and subscribe primitive are also same as described in Chapter 5.

$$Publish(publication\_specification, TTL, max\_hop)$$

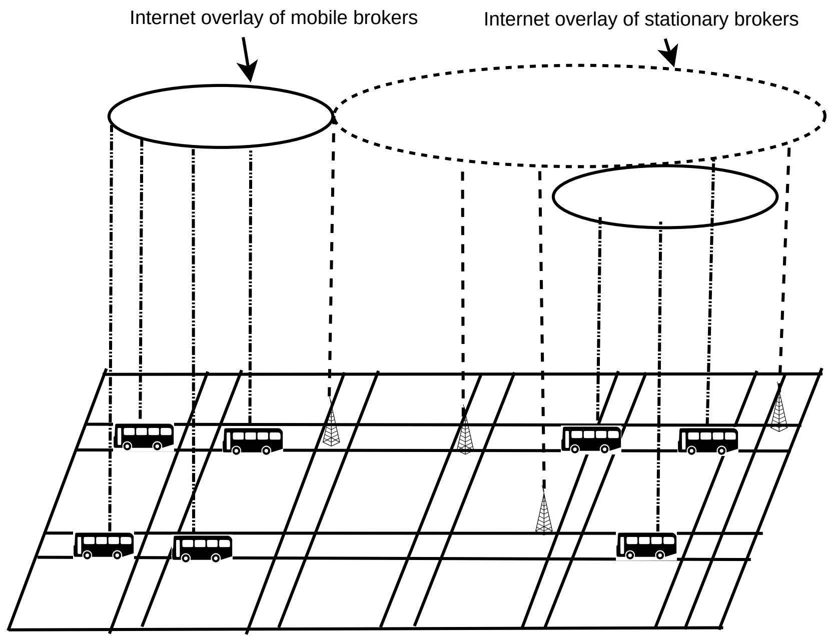


Figure 6.1: 2-Tier DHT of Fixed and Mobile Brokers

$$Subscribe(Subscription\_specification, subscriber\_id, TTL, max\_hop)$$

6.2.1 Formation of 2-tier DHT

The proposed 2-tier DHT consists of both stationary info-stations and moving city buses. As already explained, the upper-tier DHT contains info-stations whereas the lower-tier is formed by moving city buses. Further, each info-station in upper-tier DHT acts as supper-peer for a set of city-buses which also form DHT among them. These city-buses continuously change their association from one super-peer to another as they move on their route. These mobile peers exhibit a special kind of churn condition. In addition to normal dynamic leaving and joining, peers also change their point of logical attachment in the overlay network. Clearly, in this case there is a need that the assigned hashed identifiers of city-buses also include information about their respective super-peers. This is required at the time of routing the notifications towards intended subscribers.

The installation plan of info-stations is made in such a manner that each info-station is responsible for a certain geographical region of city. These regions or sectors are identified by their unique identification number. In the proposed two-tier DHT, the identifier spaces to find the hashed identifiers of peers at different tiers are independent from each other. Figure.6.2 depicts the identifier assignment scheme for any city-bus.

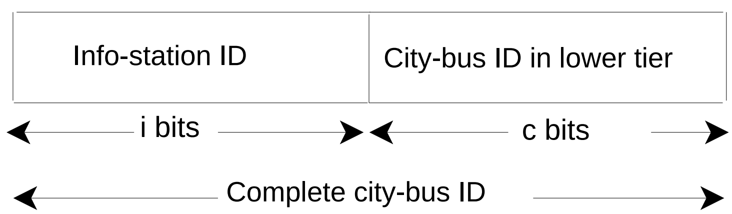


Figure 6.2: Identifier Assignment in 2-tier DHT

The first i bits represent super-peer (info-station) identifier and next c bits represent city-bus identifier. Super-peer or info-station identifiers are created by hashing the unique identification number of the sector where they are installed. The identifiers for city-buses joined at lower-tier are created by hashing their unique vehicle identification number. Further, super-peer identifier is appended before city-bus identifier to form complete city-bus identifier $(i + c)$ bits to identify a city-bus with its point of attachment in two-tier DHT.

Due to the mobility of buses, the shape and size of lower-tier DHTs keeps on changing dynamically. When a city bus enters a different sector it is informed about this event through a GPS device installed on it. Thereafter, the p2p application triggers the process of joining in new lower-tier DHT. The joining process of city-bus can be summarized in following steps:

1. Sends joining request to bootstrap server. Bootstrap server replies with the

details of any active super-peer.

2. Sends request to the super-peer informed by bootstrap server. This request includes the complete city-bus identifier as depicted in Figure 6.2. The first i bits of this identifier are created by hashing the sector identification number, which is also the new super-peer identifier.

3. The requested super-peer utilizes hashed identifier of new super-peer to route the joining request in upper-tier towards the target super-peer.

4. The new super-peer under which city-bus has to join, utilizes last c bits of complete city bus identifier to logically place and attach the city-bus in its lower-tier DHT.

In the proposed two-tier DHT, both stationary super-peers and mobile peers maintain successor list, predecessor information and finger table similar to Chord. It may be noted that super-peer is part of both upper and lower-tier. Consequently, it maintains two successor lists, two pointers for predecessor nodes and two finger tables for upper and lower-tier. Further, the frequency of periodic stabilization of successor, predecessor and finger table is lot more in lower-tier than upper-tier.

6.2.2 Publication and Subscription Routing and Installation

As discussed before in Chapter 4 and 5, vehicles that are willing to publish or subscribe must first of all transfer their publication or subscription towards any mobile or stationary broker. Both city buses and info-stations periodically broadcast control messages in their 1-hop range to inform vehicles of their presence. Publications and subscriptions are forwarded in hop-by-hop manner till they reach at any city-bus or info-station as described in Algorithm 4.2 presented in Chapter 4.

Only stationary brokers at the upper-tier (super-peers) of two-tier DHT act as rendezvous brokers. If any publication or subscription reaches at city-bus, it is routed towards the super peer utlizing the routing mechanism of lower-tier DHT. Thereafter, the topic names associated with publications and subscriptions are hashed to the identifier space of super-peers. These hashed identifiers are utilized by upper-tier DHT to route them to the super-peer which act as rendezvous broker for the desired topic names. The routing process of publications and subscriptions in upper-tier DHT is quite similar to the steps outlined in Algorithm 5.2 presented in Chapter 5.

As publishers and subscribers are purely decoupled, publications and subscriptions can be generated at any time and in any order. Successful notifications strongly depend on ordering of occurrence of publications and subscriptions, the time instant they reach at broker and their lifespan. In our approach, both publications and subscriptions have definite time span. These time spans are provided as TTL values in descriptions of publications and subscriptions by respective publishers and subscribers. Even if the subscription arrives at the rendezvous broker after the publication, subscriber can be notified if subscription-lifespan and publication-lifespan intersect with each other.

6.2.3 Locating Subscribers and Notification Delivery

Location information of the subscriber vehicle is maintained by the cooperation of both city-buses and info-stations. In this approach, the location database of subscriber vehicles is stored in distributed manner at info-stations (super-peers) only. The vehicle identification is hashed to the identifier space of super-peers, and this hashed identifier is utilized to find the super-peer responsible for the location base of any given vehicle. When a subscriber vehicle comes in the vicinity of any city-bus, it requests to update its location. The updated location information in form of complete city-bus identifier (as depicted in Figure 6.2) is forwarded to current super-peer. Subsequently, the current super-peer routes this location update to the super-peer in upper-tier responsible for the location base of subscriber vehicle.

When a notification is ready for any subscriber, it has to be forwarded to intended subscriber vehicle. The hashed identifier of subscriber vehicle assists in finding the super-peer where the location of subscriber is continuously getting updated. It may be noted that location information contains the complete hashed identifier of the city-bus, nearby which the subscriber can be found. This hashed identifier consists of the super-peer identifier also. Hence the notification can be routed utilizing the upper-tier DHT to the super-peer of the sector where the subscriber vehicle may be moving. The super-peer routes the notification using the lower-tier DHT towards the city-bus near which subscriber is last located. Finally, the city-bus uses vehicle to vehicle ad-hoc communication to forward the notification towards target subscriber.

Again, it may be noted that in our design subscriber vehicles are not required to be located exactly in a smaller region. Our approach roughly locates the vehicle in such a manner that it can be found on a road segment between two info-stations or between two city-buses on a specified route. Subsequently, vehicle to vehicle ad-hoc transfer is utilized to send notification towards target vehicle.

6.3 Simulation Results

As explained in Chapter 5, simulation is performed utilizing Oversim [5], OMNET++/INET [4], MOVE [46], SUMO [48] and TraCI [86]. The simulation parameters such as vehicle densities, speed of vehicles etc. are also set as described in Figure 5.2 of Chapter 5. Additionally, whole simulation area is divided in to 10 sectors and each sector has an info-station placed at some major intersection of roads. As described earlier, info-stations are simulated Vehicle Movement Editor of MOVE by setting the maximum speed of vehicle (which act as info-station) as zero. Further, to analyze the impact (on the performance) of number and distribution of city-buses, the percentage of city-buses is varied and their routes are set accordingly.

The performance of the two-tier DHT proposed in this chapter is compared with the single tier DHT of city-buses explained in Chapter 5 based on notification delivery ratio and delay in notification delivery. The simulation results are recorded for the following two situations:

- High number of city-buses which are evenly distributed across all routes.
- Less number of city-buses which are unevenly distributed across all routes.

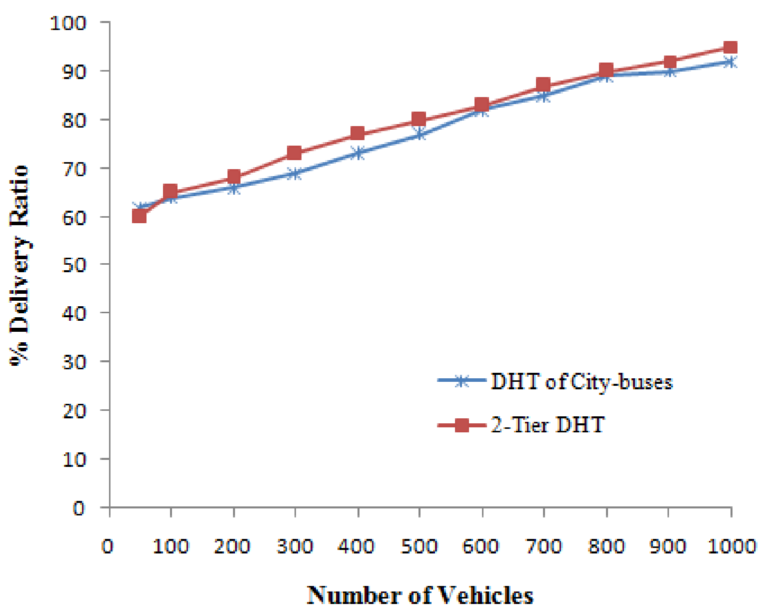


Figure 6.3: Delivery Ratio: High number of buses, Even bus distribution

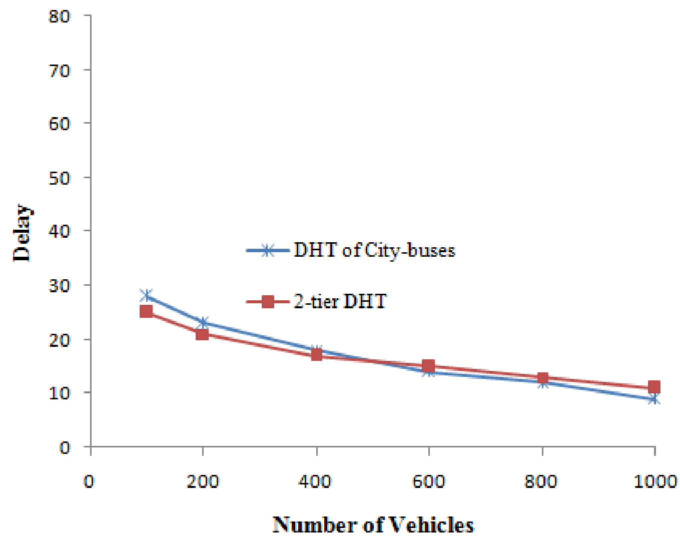


Figure 6.4: Delay: High number of buses, Even bus distribution

Figure 6.3 and 6.4 depicts the simulation results for the situation where number of city-buses is high (15% of total vehicles) and they are evenly distributed along all routes. Here the x-axis represents the number of vehicles. The y-axis represents delivery ratio in Figure 6.3 and notification delay in 6.4 respectively. For these results normal vehicle distribution is taken as uniform.

It may be observed that both 2-tier DHT approach and DHT of city-buses perform equally when the number of city-buses are relatively more and their distribution is inform. Further, as the number of vehicles increase, performance both in terms of delivery ration and notification delay improves for both the approaches.

Figure 6.5 and 6.6 depicts the simulation results for the situation where number of city-buses is less (5% of total vehicles) and they are unevenly distributed along all routes. Here also, the x-axis represents the number of vehicles. The y-axis represents delivery ratio in Figure 6.5 and notification delay in 6.6 respectively.

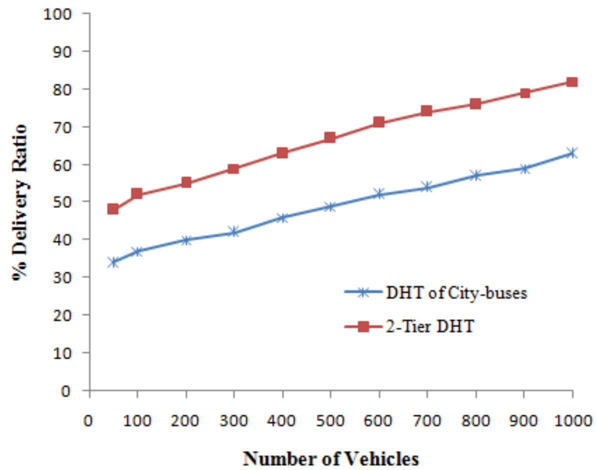


Figure 6.5: Delivery Ratio: Less number of buses, Uneven bus distribution

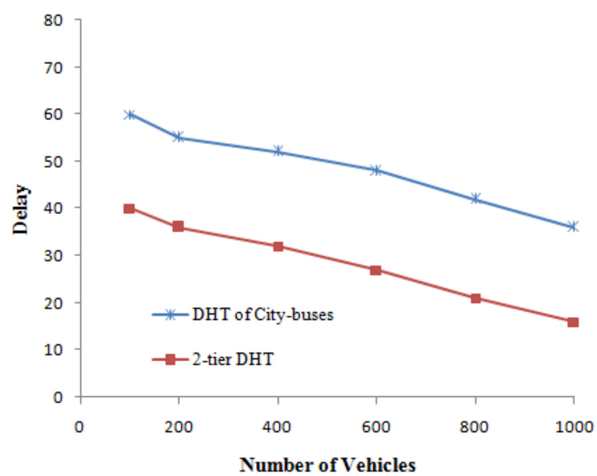


Figure 6.6: Delay: Less number of buses, Uneven bus distribution

It may be noted that under these conditions, The performance of the 2-tier DHT approach is lot more better than the approach which utilizes only city-buses as brokers. Even for the peak traffic densities, the trend remains same. This is due to the fact under uneven bus distribution some buses may be travelling in clusters with minimum distance among them whereas some buses may be far apart and there is relatively larger distance among them. If the separation among city-buses is more then publications and subscriptions may have to travel towards rendezvous broker mostly by utilizing vehicle-to-vehicle ad-hoc links. This results in loss of and (delay in) notifications.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

In this thesis, we have presented our attempt towards designing an information dissemination middleware for vehicular ad-hoc networks consisting of moving vehicles and stationary info-stations. The proposed framework has been guided by simulation studies which have been conducted to find a balance between trade-offs related to infrastructure availability and, timely and reliable delivery of information to interested vehicle. Three approaches have been presented based on the availability and unavailability of infrastructure. These approaches utilize the publish/subscribe communication paradigm and a DHT based structured P2P networks.

In the first approach, we have proposed a publish/subscribe communication framework over Chord like DHT of info-stations. Generally, vehicles act as publisher and subscriber whereas info-stations are in the role of brokers. We have utilized DHT routing substrate for information dissemination among info-stations. We do not use GPS to locate vehicles on the road. The location of vehicles is determined by message exchanges among info-stations in a cooperative manner. The simulation studies have been performed to understand the trade-offs between infrastructure deployment and ad-hoc deployment of info-stations. In the infrastructure deployment, info-stations are connected to the Internet and communication is achieved through the underlying IP based network whereas in the ad-hoc deployment, info-stations are not connected to the Internet and communication is acieved through the ad-hoc links formed among vehicles. We have used traffic scenarios for two Indian cities (South Delhi and Allahabad) as a reference for urban and semi-urban areas. The simulation results suggest that even when the info-stations are not connected, the performance of our approach is acceptable. This is true even for the situation when vehicle density is low and the skewed distribution of vehicles is considered.

In the second approach, we have proposed a publish/subscribe based information dissemination framework over DHT of mobile brokers. We have utilized city-buses run by public transport system as mobile brokers. These city-buses form DHT based P2P overlay among them using UMTS based Internet interfaces. These city buses also assist in locating subscriber vehicles for successful delivery of notifications. We have performed detailed simulation studies and compared our approach with stationary infrastructure based approach and a pure ad-hoc approach presented in [53]. A realistic scenario of South Delhi road network is simulated. Further, real routes of city-buses run by the Delhi Transport Corporation are incorporated in simulation to obtain the results. The simulation results suggests that this approach is better compared to both infrastructure based approach and pure ad-hoc approach.

Third approach presents a hybrid architecture where a 2-tier DHT of both fixed and mobile brokers used in first and second approach is formed. Whole city is divided into sectors and each sector is assigned a unique identification number. In each sector of the city an info-station is pre-installed. All the info-stations form an upper-tier of DHT ring and are designated as super-peers. Each super-peer is logically connected to lower-tier DHT ring city-buses moving in that sector. Simulations results suggest that this approach is better than the second approach when number of city-buses is less and their distribution is uneven across all routes.

Several interesting conclusions can be derived after performing detailed simulation analysis. The performance of first approach is acceptable (when skewed distribution of vehicles is considered) for ad-hoc deployment of info-stations even if the vehicle density is low. This is because of the fact that most vehicles may be found near or around the hot spots under skewed distribution of vehicles. Consequently, many small DHTs are formed due to vehicle concentration around them. As publishers and subscribers are from these vehicles only, notifications are routed easily towards subscribers due to good connectivity through dense vehicle population and notifications have to travel lesser distance to reach to subscribers.

Further, the obtained results suggested that if only city buses are acting as brokers then the performance of system can be nearly equal to the first approach considering that vehicle density is high and city buses are evenly distributed across the city. However, if vehicle density is low and distribution of city buses is uneven then the performance goes down. This provided a conclusion that full infrastructure based broker network is good but not a pragmatic solution as it needs lot of pre-planning and expenditure to install them. City buses as brokers are easy to implement in ad-hoc manner but they provide different quality of service in different traffic conditions. The third approach appeared more deployable as this hybrid setup with both fixed info-stations and city buses as part of broker network provides better performance than approach one and approach two in all traffic conditions. Also, in this approach the expenditure to install info-stations is less in comparison to first approach as only small number of info-stations (one per sector) is required.

7.2 Limitations and Future Directions

There are few limitations which exist in the work presented in this thesis. These limitations may lead to many possible research directions in which the present work can be extended. In this section we list the limitations and of some of the most obvious future directions.

In the present framework, vehicles publish and subscribe to information on variety of topics. It would be interesting to think of developing an ontology for the vehicular network environments. Then, it would be possible to gather different publications under the semantically related concepts of ontology. Subsequently, Instead of matching the topic names for notification, semantic matching can be performed. Firstly, the coarse grained matching can be performed based on the concepts and, if matched then fine grained matching can be performed based on other attributes of publications and subscriptions.

The present framework has been evaluated using simulation. Simulations are useful to easily evaluate the performance on large scale. Though the attempt has been made to set the simulation environment as realistic as possible, it will be interesting to investigate the performance of the framework by actually implementing it. The actual implementation can be on small scale using few info-stations and cars moving in a University campus.

Finally, in VANETs, it is necessary to ensure that the information in the network is reliable. Warning messages received by the driver affect his/her decision. Any wrong message can waste driver's time, lead to accidents etc. One of the important issues is related to the security, privacy and trust of the disseminated information which is required to be investigated to ensure truthful and trustworthy dissemination of information.

References

[1] Car to Car communication consortium. www.car-to-car.org.

[2] Freenet Website. http://freenetproject.org.

[3] Gnutella Protocol Specification 0.6. http://rfc-gnutella.sourceforge.net.

[4] Omnet++ Website. http://www.omnetpp.org.

[5] Oversim Website. http://www.oversim.org.

[6] Road Safety Report. http://morth.nic.in/writereaddata/sublinkimages/road_safety_sundar_report4006852610.pdf.

[7] The BitTorrent Website. http://www.bittorrent.com.

[8] The Delhi Transport Corporation Website. http://www.dtc.nic.in.

[9] The Gryphon Project. http://www.research.ibm.com/distributedmessaging/gryphon.html.

[10] Abuelela, M., and Olariu, S. Traffic-adaptive packet relaying in vanet. In *Proceedings of the fourth ACM international workshop on Vehicular ad hoc networks* (New York, NY, USA, 2007), VANET '07, ACM, pp. 77–78.

[11] Abuelela, M., and Olariu, S. Zipper:: a zero-infrastructure peer-to-peer system forvanet. In *Proceedings of the 3rd ACM workshop on Wireless multimedia networking and performance modeling* (New York, NY, USA, 2007), WMuNeP '07, ACM, pp. 2–8.

[12] Ahmed, S., and Kanhere, S. S. Vanetcode: network coding to enhance cooperative downloading in vehicular ad-hoc networks. In *Proceedings of the 2006 international conference on Wireless communications and mobile computing* (New York, NY, USA, 2006), IWCMC '06, ACM, pp. 527–532.

[13] Ali, S., and Bilal, S. An intelligent routing protocol for vanets in city environments. In *Computer, Control and Communication, 2009. IC4 2009. 2nd International Conference on* (2009), pp. 1–5.

[14] Alsabaan, M., Alasmary, W., Albasir, A., and Naik, K. Vehicular networks for a greener environment: A survey. *Communications Surveys Tutorials, IEEE 15*, 3 (2013), 1372–1388.

[15] Atechian, T., and Brunie, L. Dg-castor for query packets dissemination in vanet. In *Mobile Ad Hoc and Sensor Systems, 2008. MASS 2008. 5th IEEE International Conference on* (2008), pp. 547–552.

[16] Atechian, T., Torbey, Z., Bennani, N., and Brunie, L. Coffee: Cooperative and infrastructure-free peer-to-peer system for vanet. In *Intelligent Transport Systems Telecommunications,(ITST),2009 9th International Conference on* (2009), pp. 510–515.

[17] Baldoni, R., Beraldi, R., Cugola, G., Migliavacca, M., and Querzoni, L. Structure-less content-based routing in mobile ad hoc networks. In

Pervasive Services, 2005. ICPS '05. Proceedings. International Conference on (2005), pp. 37–46.

[18] Baldoni, R., Marchetti, C., Virgillito, A., and Vitenberg, R. Content-based publish-subscribe over structured overlay networks. In *Proceedings of the 25th IEEE International Conference on Distributed Computing Systems* (Washington, DC, USA, 2005), ICDCS '05, IEEE Computer Society, pp. 437–446.

[19] Bisignano, M., Calvagna, A., Modica, G. D., and Tomarchio, O. Expeerience: a jxta middleware for mobile ad-hoc networks. In *Peer-to-Peer Computing, 2003.(P2P 2003). Proceedings. Third International Conference on* (2003), IEEE, pp. 214–215.

[20] Carzaniga, A., Rosenblum, D. S., and Wolf, A. L. Design and evaluation of a wide-area event notification service. *ACM Transaction on Computer Systems 19* (August 2001), 332–383.

[21] Castro, M., Druschel, P., Kermarrec, A.-M., and Rowstron, A. Scribe: A large-scale and decentralized application-level multicast infrastructure. *IEEE Journal on Selected Areas in Communications 20*, 8 (2002), 1489 – 1499.

[22] Caviglione, L., Ciaccio, G., and Gianuzzi, V. Architecture of a communication middleware for vanet applications. In *Ad Hoc Networking Workshop (Med-Hoc-Net), 2011 The 10th IFIP Annual Mediterranean* (2011), pp. 111–114.

[23] Cheng, W., Cheng, X., Song, M., Chen, B., and Zhao, W. W. On

the design and deployment of rfid assisted navigation systems for vanets. *IEEE Trans. Parallel Distrib. Syst. 23*, 7 (July 2012), 1267–1274.

[24] Chetan, S., Al-Muhtadi, J., Campbell, R., and Mickunas, M. Mobile gaia: a middleware for ad-hoc pervasive computing. In *Consumer Communications and Networking Conference, 2005. CCNC. 2005 Second IEEE* (2005), pp. 223–228.

[25] Clausen, T., and Jacquet, P. Optimized link state routing protocol (olsr) RFC.

[26] Costa, P., Migliavacca, M., Picco, G. P., and Cugola, G. Epidemic algorithms for reliable content-based publish-subscribe: An evaluation. In *Proceedings of the 24th International Conference on Distributed Computing Systems (ICDCS'04)* (Washington, DC, USA, 2004), ICDCS '04, IEEE Computer Society, pp. 552–561.

[27] Costa, P., and Picco, G. P. Semi-probabilistic content-based publish-subscribe. In *Proceedings of the 25th IEEE International Conference on Distributed Computing Systems* (Washington, DC, USA, 2005), ICDCS '05, IEEE Computer Society, pp. 575–585.

[28] Cugola, G., Di Nitto, E., and Fuggetta, A. Exploiting an event-based infrastructure to develop complex distributed systems. In *Proceedings of the 20th international conference on Software engineering* (Washington, DC, USA, 1998), ICSE, IEEE Computer Society, pp. 261–270.

[29] Das, S., Nandan, A., and Pau, G. Spawn: a swarming protocol for vehicular ad-hoc wireless networks. In *Proceedings of the 1st ACM international*

workshop on Vehicular ad hoc networks (New York, NY, USA, 2004), VANET '04, ACM, pp. 93–94.

[30] Dow, C.-R., Lee, Y.-H., Hsuan, P., Lee, Y.-T., and Hwang, S.-F. Omnibone: An efficient service data circulation and discovery scheme in vanets. *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA) 3*, 3 (9 2012), 4–20.

[31] Eugster, P. T., Felber, P. A., Guerraoui, R., and Kermarrec, A.-M. The many faces of publish/subscribe. *ACM Computing Survey 35* (June 2003), 114–131.

[32] Festag, A., Fussler, H., Hartenstein, H., Sarma, A., and Schmitz, R. FLEETNET: BRINGING CAR-TO-CAR COMMUNICATION INTO THE REAL WORLD. *Computer 4*, L15 (2004), 16.

[33] Festag, A., Noecker, G., Strassberger, M., Lbke, A., Bochow, B., Torrent-moreno, M., Schnaufer, S., Eigner, R., Catrinescu, C., and Kunisch, J. now network on wheels: Project objectives, technology and achievements.

[34] Floyd, S., Jacobson, V., Liu, C.-G., McCanne, S., and Zhang, L. A reliable multicast framework for light-weight sessions and application level framing. *IEEE/ACM Transaction on Networking 5* (December 1997), 784–803.

[35] Frey, H., Gorgen, D., Lehnert, J., and Sturm, P. Selma: A middleware platform for self-organizing distributed applications in mobile multihop ad-hoc networks. In *Western Simulation MultiConference WMC'04, San Diego, California, USA* (2004).

[36] Ghandeharizade, S., Kapadia, S., and Krishnamachari, B. Pavan: a policy framework for content availabilty in vehicular ad-hoc networks. In *Proceedings of the 1st ACM international workshop on Vehicular ad hoc networks* (New York, NY, USA, 2004), VANET '04, ACM, pp. 57–65.

[37] Guimarães, M., and Rodrigues, L. A genetic algorithm for multicast mapping in publish-subscribe systems. In *Proceedings of the Second IEEE International Symposium on Network Computing and Applications* (Washington, DC, USA, 2003), NCA '03, IEEE Computer Society, pp. 67–75.

[38] Hadaller, D., Keshav, S., Brecht, T., and Agarwal, S. Vehicular opportunistic communication under the microscope. In *Proceedings of the 5th international conference on Mobile systems, Applications and Services* (New York, NY, USA, 2007), MobiSys '07, ACM, pp. 206–219.

[39] Hadim, S., Al-Jaroodi, J., and Mohamed, N. Trends in middleware for mobile ad hoc networks. *JCM 1*, 4 (2006), 11–21.

[40] Harsch, C., Festag, A., and Papadimitratos, P. Secure position-based routing for vanets. In *Vehicular Technology Conference, 2007. VTC-2007 Fall. 2007 IEEE 66th* (2007), pp. 26–30.

[41] Herrmann, K. Mesh mdl " a middleware for self-organization in ad hoc networks. In *Proceedings of the 23rd International Conference on Distributed Computing Systems* (Washington, DC, USA, 2003), ICDCSW '03, IEEE Computer Society, pp. 446–.

[42] Huang, Y., and Garcia-Molina, H. Publish/subscribe tree construction in wireless ad-hoc networks. In *Proceedings of the 4th International Conference*

on Mobile Data Management (London, UK, UK, 2003), MDM '03, Springer-Verlag, pp. 122–140.

[43] Hughes, B., Meier, R., Cunningham, R., and Cahill, V. Towards real-time middleware for vehicular ad hoc networks. In *Proceedings of the 1st ACM international workshop on Vehicular ad hoc networks* (New York, NY, USA, 2004), VANET '04, ACM, pp. 95–96.

[44] Johnson, D. B., Maltz, D. A., and Broch, J. Ad hoc networking. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2001, ch. DSR: the dynamic source routing protocol for multihop wireless ad hoc networks, pp. 139–172.

[45] Karger, D., Lehman, E., Leighton, T., Panigrahy, R., Levine, M., and Lewin, D. Consistent hashing and random trees: distributed caching protocols for relieving hot spots on the world wide web. In *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing* (New York, NY, USA, 1997), STOC '97, ACM, pp. 654–663.

[46] Karnadi, F. K., Mo, Z. H., and Lan, K.-C. Rapid generation of realistic mobility models for vanet. In *WCNC* (2007), pp. 2506–2511.

[47] Kenney, J. Dedicated short-range communications (dsrc) standards in the united states. *Proceedings of the IEEE 99*, 7 (july 2011), 1162 –1182.

[48] Krajzewicz, D., Hertkorn, G., Rössel, C., and Wagner, P. Sumo (simulation of urban mobility); an open-source traffic simulation. In *Proceedings of the 4th Middle East Symposium on Simulation and Modelling* (2002), (MESM2002), pp. 183–187.

[49] Lee, E.-K., Yoo, Y. M., Park, C. G., Kim, M., and Gerla, M. Installation and evaluation of rfid readers on moving vehicles. In *Proceedings of the sixth ACM international workshop on VehiculAr InterNETworking* (New York, NY, USA, 2009), VANET '09, ACM, pp. 99–108.

[50] Lee, K., Lee, S.-H., Cheung, R., Lee, U., and Gerla, M. First experience with cartorrent in a real vehicular ad hoc network testbed. In *2007 Mobile Networking for Vehicular Environments* (2007), pp. 109–114.

[51] Lee, U., Park, J.-S., Yeh, J., Pau, G., and Gerla, M. Code torrent: content distribution using network coding in vanet. In *Proceedings of the 1st international workshop on Decentralized resource sharing in mobile computing and networking* (New York, NY, USA, 2006), MobiShare '06, ACM, pp. 1–5.

[52] Leontiadis, I. Publish/subscribe notification middleware for vehicular networks. In *Proceedings of the 4th on Middleware doctoral symposium* (New York, NY, USA, 2007), MDS '07, ACM, pp. 12:1–12:6.

[53] Leontiadis, I., Costa, P., and Mascolo, C. A hybrid approach for content-based publish/subscribe in vehicular networks. *Pervasive Mob. Comput. 5* (December 2009), 697–713.

[54] Leontiadis, I., Costa, P., and Mascolo, C. Extending access point connectivity through opportunistic routing in vehicular networks. In *Proceedings of the 29th conference on Information communications* (Piscataway, NJ, USA, 2010), INFOCOM'10, IEEE Press, pp. 486–490.

[55] Leontiadis, I., and Mascolo, C. Geopps: Geographical opportunistic routing for vehicular networks. In *World of Wireless, Mobile and Multimedia*

Networks, 2007. WoWMoM 2007. IEEE International Symposium on a (2007), pp. 1–6.

[56] Leontiadis, I., and Mascolo, C. Opportunistic spatio-temporal dissemination system for vehicular networks. In *Proceedings of the 1st international MobiSys workshop on Mobile opportunistic networking* (New York, NY, USA, 2007), MobiOpp '07, ACM, pp. 39–46.

[57] Levis, P., and Culler, D. Mate: A tiny virtual machine for sensor networks. In *International Conference on Architectural Support for Programming Languages and Operating Systems, San Jose, CA, USA* (Oct. 2002).

[58] Luo, J., Gu, X., Zhao, T., and Yan, W. Mi-vanet: A new mobile infrastructure based vanet architecture for urban environment. In *Vehicular Technology Conference Fall (VTC 2010-Fall), 2010 IEEE 72nd* (2010), pp. 1–5.

[59] Ma, S., Wolfson, O., and Lin, J. Iip: an event-based platform for its applications. In *Proceedings of the Second International Workshop on Computational Transportation Science* (New York, NY, USA, 2010), IWCTS '10, ACM, pp. 1–6.

[60] Mascolo, C., Capra, L., Zachariadis, S., and Emmerich, W. Xmiddle: A data-sharing middleware for mobile computing. *Wirel. Pers. Commun. 21*, 1 (Apr. 2002), 77–103.

[61] Meier, R., and Cahill, V. Steam: Event-based middleware for wireless ad hoc network. In *Proceedings of the 22nd International Conference on Distributed Computing Systems* (Washington, DC, USA, 2002), ICDCSW '02, IEEE Computer Society, pp. 639–644.

[62] Morris, R., Jannotti, J., Kaashoek, F., Li, J., and Decouto, D. Carnet: a scalable ad hoc wireless network system. In *Proceedings of the 9th workshop on ACM SIGOPS European workshop: beyond the PC: new challenges for the operating system* (New York, NY, USA, 2000), EW 9, ACM, pp. 61–65.

[63] Musolesi, M., Mascolo, C., and Hailes, S. Emma: Epidemic messaging middleware for ad hoc networks. *Personal Ubiquitous Comput. 10*, 1 (Dec. 2005), 28–36.

[64] Naumov, V., and Gross, T. Connectivity-aware routing (car) in vehicular ad-hoc networks. In *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE* (2007), pp. 1919–1927.

[65] Perkins, C., Belding-Royer, E., and Das, S. Ad hoc on-demand distance vector (aodv) routing RFC.

[66] Perkins, C. E., and Bhagwat, P. Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers. *SIGCOMM Comput. Commun. Rev. 24*, 4 (Oct. 1994), 234–244.

[67] Picco, G. P., Cugola, G., and Murphy, A. L. Efficient content-based event dispatching in the presence of topological reconfiguration. In *Proceedings of the 23rd International Conference on Distributed Computing Systems* (Washington, DC, USA, 2003), ICDCS '03, IEEE Computer Society, pp. 234–.

[68] Piramuthu, S. Protocols for rfid tag/reader authentication. *Decision Support Systems 43*, 3 (2007), 897–914.

[69] Piramuthu, S. Adaptive framework for collisions in rfid tag identification. *Journal of Information and Knowledge Management 07*, 01 (2008), 9–14.

[70] Prinz, V., Brocco, M., and Woerndl, W. Peer-to-peer algorithms for c2c communication systems. In *Proceedings of the 6th International Conference on Advances in Mobile Computing and Multimedia* (New York, NY, USA, 2008), MoMM '08, ACM, pp. 376–381.

[71] Prinz, V., Eigner, R., and Woerndl, W. Cars communicating over publish/subscribe in a peer-to-peer vehicular network. In *Proceedings of the 2009 International Conference on Wireless Communications and Mobile Computing: Connecting the World Wirelessly* (New York, NY, USA, 2009), IWCMC '09, ACM, pp. 431–436.

[72] Reichardt, D., Miglietta, M., Moretti, L., Morsink, P., and Schulz, W. Cartalk 2000: safe and comfortable driving based upon inter-vehicle-communication. In *Intelligent Vehicle Symposium, 2002. IEEE* (2002), vol. 2, pp. 545–550 vol.2.

[73] Riabov, A., Liu, Z., Wolf, J. L., Yu, P. S., and Zhang, L. Clustering algorithms for content-based publication-subscription systems. In *Proceedings of the 22 nd International Conference on Distributed Computing Systems (ICDCS'02)* (Washington, DC, USA, 2002), ICDCS '02, IEEE Computer Society, pp. 133–141.

[74] Riabov, A., Liu, Z., Wolf, J. L., Yu, P. S., and Zhang, L. New algorithms for content-based publication-subscription systems. In *Proceedings of the 23rd International Conference on Distributed Computing Systems* (Washington, DC, USA, 2003), ICDCS '03, IEEE Computer Society, pp. 678–.

[75] Rowstron, A. I. T., and Druschel, P. Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems. In *Proceedings*

of the IFIP/ACM International Conference on Distributed Systems Platforms (London, UK, 2001), Middleware '01, Springer-Verlag, pp. 329–350.

[76] Rybicki, J., Scheuermann, B., Koegel, M., and Mauve, M. Peertis: a peer-to-peer traffic information system. In *Proceedings of the sixth ACM international workshop on VehiculAr InterNETworking* (New York, NY, USA, 2009), VANET '09, ACM, pp. 23–32.

[77] Safa, H., Artail, H., and Tabet, D. A cluster-based trust-aware routing protocol for mobile ad hoc networks. *Wirel. Netw. 16*, 4 (May 2010), 969–984.

[78] Saleet, H., Langar, R., Naik, K., Boutaba, R., Nayak, A., and Goel, N. Intersection-based geographical routing protocol for vanets: A proposal and analysis. *Vehicular Technology, IEEE Transactions on 60*, 9 (2011), 4560–4574.

[79] Singh, M. P., and Gore, M. M. A new top-down hierarchical multi-hop routing protocol for wireless sensor networks. In *Proceedings of the 9th international conference on Distributed computing and networking* (Berlin, Heidelberg, 2008), ICDCN'08, Springer-Verlag, pp. 428–433.

[80] Stoica, I., Morris, R., Liben-Nowell, D., Karger, D. R., Kaashoek, M. F., Dabek, F., and Balakrishnan, H. Chord: A scalable peer-to-peer lookup protocol for internet applications. *IEEE/ACM Transaction on Networking 11* (February 2003), 17–32.

[81] Sun, S., Kim, J., Jung, Y., and Kim, K. Zone-based greedy perimeter stateless routing for vanet. In *Information Networking, 2009. ICOIN 2009. International Conference on* (2009), pp. 1–3.

[82] Terpstra, W. W., Behnel, S., Fiege, L., Zeidler, A., and Buchmann, A. P. A peer-to-peer approach to content-based publish/subscribe. In *Proceedings of the 2nd international workshop on Distributed event-based systems* (New York, NY, USA, 2003), DEBS '03, ACM, pp. 1–8.

[83] Tokuda, K., Akiyama, M., and Fujii, H. Dolphin for inter-vehicle communications system. In *Intelligent Vehicles Symposium, 2000. IV 2000. Proceedings of the IEEE* (2000), pp. 504–509.

[84] Triantafillou, P., and Aekaterinidis, I. Content-based Publish/Subscribe over Structured P2P Networks. In *Proceedings of the 4th International Workshop on Distributed Event-Based Systems* (2004).

[85] Tsugawa, S., Kato, S., Tokuda, K., Matsui, T., and Fujii, H. A cooperative driving system with automated vehicles and inter-vehicle communications in demo 2000. In *Intelligent Transportation Systems, 2001. Proceedings. 2001 IEEE* (2001), pp. 918–923.

[86] Wegener, A., Piórkowski, M., Raya, M., Hellbrück, H., Fischer, S., and Hubaux, J.-P. Traci: an interface for coupling road traffic and network simulators. In *Proceedings of the 11th communications and networking simulation symposium* (New York, NY, USA, 2008), CNS '08, ACM, pp. 155–163.

[87] Whetten, B., Montgomery, T., and Kaplan, S. M. A high performance totally ordered multicast protocol. In *Selected Papers from the International Workshop on Theory and Practice in Distributed Systems* (London, UK, 1995), Springer-Verlag, pp. 33–57.

[88] Zhuang, S. Q., Zhao, B. Y., Joseph, A. D., Katz, R. H., and Kubiatowicz, J. D. Bayeux: an architecture for scalable and fault-tolerant wide-area data dissemination. In *Proceedings of the 11th international workshop on Network and operating systems support for digital audio and video* (New York, NY, USA, 2001), NOSSDAV '01, ACM, pp. 11–20.