Cooperative Caching In Vehicular Networks: Distributed Cache Invalidation Using Information Freshness

Master-Arbeit

Deepak Jayaram

KOM-M-0622

TECHNISCHE UNIVERSITÄT DARMSTADT

Fachbereich Elektrotechnik
und Informationstechnik
Fachbereich Informatik (Zweitmitglied)

Fachgebiet Multimedia Kommunikation
Prof. Dr.-Ing. Ralf Steinmetz

Cooperative Caching In Vehicular Networks: Distributed Cache Invalidation Using Information Freshness
Master-Arbeit
KOM-M-0622

Eingereicht von Deepak Jayaram
Tag der Einreichung: 22. März 2018

Gutachter: Prof. Dr.-Ing. Ralf Steinmetz
Betreuer: Tobias Meuser, M. Sc.
Florian Jomrich, M. Sc.

Technische Universität Darmstadt
Fachbereich Elektrotechnik und Informationstechnik
Fachbereich Informatik (Zweitmitglied)

Fachgebiet Multimedia Kommunikation (KOM)
Prof. Dr.-Ing. Ralf Steinmetz

Erklärung zur Abschlussarbeit gemäß § 23 Abs. 7 APB der TU Darmstadt

Hiermit versichere ich, Deepak Jayaram, die vorliegende Master-Arbeit ohne Hilfe Dritter und nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die Quellen entnommen wurden, sind als solche kenntlich gemacht worden. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Mir ist bekannt, dass im Falle eines Plagiats (§38 Abs.2 APB) ein Täuschungsversuch vorliegt, der dazu führt, dass die Arbeit mit 5,0 bewertet und damit ein Prüfungsversuch verbraucht wird. Abschlussarbeiten dürfen nur einmal wiederholt werden.

Bei der abgegebenen Master-Arbeit stimmen die schriftliche und die zur Archivierung eingereichte elektronische Fassung überein.

Darmstadt, den 22. März 2018

____

Deepak Jayaram

Abstract

Recent advances in vehicular communication has led to significant opportunities to deploy variety of applications and services improving road safety and traffic efficiency to road users. In regard to traffic management services in distributed vehicular networks, this thesis work evaluates managing storage at vehicles efficiently as *cache* for moderate cellular transmission costs while still achieving correct routing decision.

Road status information was disseminated to oncoming traffic in the form of cellular notifications using a reporting mechanism. High transmission costs due to redundant notifications published by all vehicles following a basic reporting mechanism: *Default-approach* was overcome by implementing caching at every vehicle.

A cooperative based reporting mechanism utilizing cache: *Cooperative-approach*, was proposed to notify road status while avoiding redundant notifications. In order to account those significantly relevant vehicles for decision-making process which did not actually publish, correspondingly virtual cache entries were implemented. To incorporate the real-world scenario of varying vehicular rate observed on any road, virtual cache entries based on varying vehicular rate was modelled as *Adaptive Cache Management* mechanism.

The combination of proposed mechanisms were evaluated for cellular transmission costs and accuracy achieved for making correct routing decision. Simulation case studies comprising varying vehicular densities and different false detection rates were conducted to demonstrate the performance of these mechanisms. Additionally, the proposed mechanisms were evaluated in different decision-making algorithms for both information freshness in changing road conditions and for robustness despite false detections.

The simulation results demonstrated that the combination of proposed mechanisms was capable of achieving realistic information accuracy enough to make correct routing decision despite false readings while keeping network costs significantly low. Furthermore, using QoI-based decision algorithm in high density vehicular networks, fast adaptability to frequently changing road conditions as well as quick recovery from false notifications by invalidating them with correct notifications were indicated.

Contents

List of Figures

List of Tables

List of Acronyms

ADAS Advanced Driver Assistance Systems 1
AOI Area of Interest 28, 31, 37, 39, 46, 47, 56, 57, 70
AP Access Point 9, 23, 24

BS Base Station 10

CAMs Cooperative Awareness Messages 5
CCI Cooperative Cache Invalidation 23, 24
CCN Content Centric Networking 7–9, 14
CDMA Code-division Multiple Access 5
CID ContentID 9
CN Caching Node 21, 22
COACS Cooperative and Adaptive Caching System 18, 21
CS Content Store 7, 15

D2D Device-to-Device Communication 5
DENMs Decentralized Environmental Notification Messages 5
DHT Distributed Hash Table 6, 14
DiCAS Distributed Caching and Adaptive Search 14
DSRC Dedicated Short Range Communication 3, 4, 6
DTN Delay Tolerant Networks 4, 29, 30

ECCI Enhanced Cooperative Cache Invalidation 23, 24

FIB Forwarding Information Base 7, 15

GFA Gateway Foreign Agents 23, 24
GPS Global Positioning System 17, 30, 45
GWSP Greedy Walk-based Selective Push 18

HA Home Agents 23, 24
HSPA High-speed Packet Access 5

ICN Information-Cetric Network 7–9, 14, 15
IR Invalidation Report 18–20, 22, 24
ITS Intelligent Transportation System 1, 3, 40
IVC Inter-Vehicle Capability 16

LTE Long Term Evolution 5, 29, 45

MANET Mobile Ad-hoc Networks 22
MIMO Multiple Input Multiple Out 5

NDN Named Data Networking 7, 9, 10, 14, 15

P2P Peer-to-Peer 6, 12–14, 17
PID PacketID 9
PIT Pending Interest Table 7–9, 15
POI Place of Incident 28, 31, 32, 35–37, 39, 40, 42, 43, 46, 47, 54, 59–61, 63, 68, 70, 72, 75
PST Pub/Sub Tree 6

QD Query Directive 21, 22
QoE Quality of Experience 10, 11
QoS Quality of Service 4, 5

RSU Road Side Unit 3, 7, 9, 10, 17, 40
RSUCs Road Side Unit Centres 10
RTT Round-trip Time 12

SDN Software Defined Network 10
SDNCs SDN Controllers 10
SSUM Smart Server Update Mechanism 18

TTL Time-to-live 2, 13, 16, 18, 20–23, 30, 31, 36, 37, 39, 42, 43, 54, 58, 78, 79
TTR Time-to-refresh 18

UIC Uniform Index Caching 14
UIR Updated Invalidation Report 18–20

V-NDN Vehicular Named Data Networking 9, 15
V2I Vehicle-to-Infrastructure 1, 3, 9, 40
V2V Vehicle-to-Vehicle 1, 3, 40
VADD Vehicle-Assisted Data Delivery 5
VAHS Virtual Ad-Hoc Server 16
VANET Vehicular Ad-hoc Networks 6, 7, 9, 15, 22, 37, 39, 78
VDTN Vehicular Delay Tolerant Networks 4, 5
VITP Vehicular Information Transfer Protocol 15, 16, 23

WANET Wireless Ad-hoc Networks 22
WAVE Wireless Access in Vehicular Environments 3

ZOR Zone of Relevance 17

1 Introduction

In this chapter, a preamble to this thesis work is presented. First, motivation for the chosen study domain is provided, followed by the specific use case considered in this study. With that, an existing problem with its impact, and thereby the necessity for an alternative efficient approach is briefly introduced. Later, a list of contributions of this work is highlighted. Finally, an outline regarding the organization of the thesis report is given.

1.1 Motivation and Contribution

Transportation system in modern society is plagued with many problems such as increased traffic congestion, CO_2 emissions, accidents and mortality. *Global Status Report on Road Safety - 2015* [WHO15] conducted by WHO, states that a staggering of 1.2 million people die each year with road injuries accounting it to be a leading cause of death globally. Predominantly, in many cases, accidents can be largely preventable, provided a novel vehicular communication exists. Road traffic not only exerts stress on road users but also accounts for increased CO_2 emissions. According to *Traffic Scorecard* report [INR16], conducted by INRIX, in Germany for the year 2016, a commuter in Darmstadt city spent 24 hours in an average traffic jam. The waiting time is much higher in bigger city such as Munich, averaging more than 48 hours. Furthermore, traffic congestion costs millions of Euros and expand GDP.

As part of making roads safer and efficient using technology, Intelligent Transportation System (ITS) was conceptualized. In ITS, the idea of vehicular network and communication using Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) is researched for practical feasibility in parallel with Advanced Driver Assistance Systems (ADAS). Vehicular networks are a promising approach to increase road safety, traffic efficiency, and comfort of road users. Applications of vehicular communication includes but not limited to safety and hazard warnings, Telematics for traffic management and infotainment services. These applications help provide number of services ranging from collision risk warning, traffic jam warnings, and road hazard warnings such as road works, car breakdown, bad weather conditions etc., For this, the network of vehicles relies on collaborative exchange of information of their surrounding recorded through on-board sensors to notify and alert other road users.

In this thesis, we concentrate on traffic management services in vehicular environment that reports road status and events from the affected road to the parties likely to be influenced by this. For example, alerts informing accident, traffic, congestion, and delay can be reported to the oncoming traffic. To that end, a real time, low latent and reliable information diffusion system with moderate communication range is desired.

In vehicular networks, communication can happen in an ad-hoc manner without a central entity or using cellular communication via a central entity. Although, ad-hoc network is cheaper in terms of setting up the infrastructure, but due to its dependency on other vehicles for information dissemination and with limited communication range makes it unsuitable for long distance traffic alerts that are generally sought.

In this work, to realize a desired information diffusion system for reporting traffic conditions, we employ communication using Publish/Subscribe mechanism via cellular infrastructure. This is because a Publish/Subscribe mechanism offers flexible filtering of information based on interested topics while being anonymous and low latent. We identify the scalability issue associated with the basic intuitive *Default-approach* of reporting as it burdens the cellular infrastructure with substantially high number of notifications that report the same information. Due to these redundant notifications, network congestion is caused that might result in application failure, information loss or performance degradation of cellular infrastructure that which already serves other mobile and internet applications.

Modern vehicles are not scarce of storage space, and hence as an alternative, we propose *Cooperative-approach* of reporting that makes efficient use of storage available at every vehicle to cache received notifications. This cache is used to avoid redundant notifications while keeping the network load at moderate levels and therefore scales well.

To overcome the inconsistency in decision-making associated with *Cooperative-approach* of reporting, we propose *Adaptive Cache Management* at every vehicle using virtual cache entries based on vehicular rate. To this end, a stateful entity with global knowledge for measuring vehicular rate is assumed, and the vehicular rate is sent as a publication attribute during publishing. As commonly seen in practice, vehicular rate observed on a road varies throughout the day over time. Accordingly, the proposed *Adaptive Cache Management* mechanism is designed to also deal with this varying vehicular environment.

Information quality is maintained by using Time-to-live (TTL) to invalidate any outdated notifications that are irrelevant. The proposed reporting approach in combination with cache management using virtual entries is evaluated for accuracy in different environments characterized by varying vehicular densities. Moreover, this combination is also assessed for performance in false detection environment where few of many vehicles can report a situation wrongly. The proposed system using QoI-based decision algorithm, upon comparison to other methods across different vehicular densities, adapts quickly to changing road conditions and any wrong notification is invalidated by the next correct notification.

An overview of contributions of this thesis work is highlighted below:

- Design and conceptualization of *Cooperative-approach* of reporting, that which eliminates redundant publications thus reducing transmission costs significantly.
- Despite significant reduction in actual cellular notifications, acceptable levels of accuracy required for decision-making besides false detections is achieved with *Adaptive Cache Management* using varying vehicular rate.
- Our proposal combining *Cooperative-approach* of reporting and *Adaptive Cache Management* achieves and maintains acceptable levels of information quality while sustaining cellular costs in a constantly changing vehicular environment.

1.2 Outline

The rest of this thesis report is organized as follows. In the succeeding Chapter 2, literature survey on communication mechanisms relevant to vehicular networks are discussed. In Chapter 3, we describe at first, the importance of caching and information freshness. Following that, methods for maintaining information freshness using caching are discussed.

In Chapter 4, the example scenario considered and evaluated in this thesis work is presented. Based on survey carried in Chapter 2 and Chapter 3, an appropriate communication mechanism for the scenario is chosen. We then introduce to the working of a basic and intuitive reporting method: *Default-approach*, which can be optimized; therewith introducing the optimization problem. In Chapter 5, we propose both *Cooperative-approach* for reporting and *Adaptive Cache Management* mechanism. Both these are applied in conjunction to address the optimization problem and essentially is the contribution of this work.

Chapter 6, introduces to the system and simulation components. Here, we also highlight the assumptions made that are required for the working of our approach. In Chapter 7, performance of *Cooperative-approach* in comparison to the *Default-approach* is assessed. We simulate and run both approaches with and without *Adaptive Cache Management* mechanism for different decision-making strategies in varying vehicular densities and false detection rates as required in vehicular networks. Finally, Chapter 8, concludes this work with the summary of evaluated results and interpretation, and presents possible future work taking this study further.

2 Background

To motivate this thesis, relevant background on existing communication designs in vehicular networks is discussed. Vehicular networks can be implemented both in infrastructure less ad-hoc manner or using cellular infrastructure alone, or by a combination of both. An overview of these approaches is discussed here. Following that, we outline different data dissemination paradigms and the ways in which they can be incorporated to assist vehicular communication.

2.1 Communication in Vehicular Networks

Communication among vehicles is conceptualized particularly for traffic efficiency, safety, and comfort applications. Using wireless protocols, a single-hop direct and a multi-hop communication between mobile vehicles are explored in an infrastructure less manner. Whereas, making opportunistic use of available cellular infrastructure, vehicular services with high data rate are also investigated. V2V communication challenges include high speed mobility of nodes in an unpredictable manner combined with delay constraints. For example, an emergency collision warning message must be exchanged immediately despite altering topology. Other traditional challenges of *Doppler Effect* and *Hidden Terminal* problems affect cellular communication's performance, and therefore, communication mechanisms should be designed to effectively handle these challenges too.

2.1.1 Vehicular Ad-hoc Networks using IEEE 802.11p

IEEE 802.11 provides a set of physical layer and media access control specifications for wireless communication. This standard is extended by adding amendments as in IEEE 802.11p, supporting wireless communication in vehicular networks.

IEEE 802.11p, which is an updated version of IEEE 802.11b that works on data link, physical layers and renders communication between high speed vehicles. It provides reliable data exchange between fast moving mobile nodes such as vehicles and also for communication with Road Side Unit (RSU) of ITS.

The work presented in [AL14] consists an overview of frequency band and physical specifications of IEEE 802.11p, which is considered under Wireless Access in Vehicular Environments (WAVE). Dedicated Short Range Communication (DSRC) protocol provides standards not just for V2V but also for V2I with distinct characteristics of high data transfers and low communication delay. In total, 7 channels are allocated in DSRC. First channel is used for safety communication, second for critical safety communication, third for high power public safety and the rest for either safety or non safety communications.

FEATURES	Japan	Europe	USA
Radio band	80 MHz	20 MHz	75 MHz
Data rate	1-4 Mbps	250 Kbps	3-27 Mbps
Communication range	30 m	15-20 m	1000 m (max)
Radio frequency	5.8 GHz	5.8 GHz	5.9 GHz

Table 2.1: Regional differences in DSRC specification (adapted from [BG13])

The standards for DSRC in each of the research contributing regions such as Europe, Japan, and USA varies and a comparison table summarizing it is given in Table 2.1. Standardization and consensus among research community and auto makers is yet to be achieved.

PARAMETERS	IEEE 802.11b	IEEE 802.11p
Channelbandwidth	20 MHz	10 MHz
Data rates	1-11 Mbps	3-27 Mbps
Slot time	20 μs	16 μs
SIFS time	10 μs	32 μs
Preamble length	96 μs (short), 192 μs (long)	32 μs
Air propagation time	<2 μs	<4 μs
CWmin	31	15
CWmax	1023	1023

Table 2.2: PHY values of IEEE 802.11b and IEEE 802.11p (adapted from [BG13])

Performance evaluation of IEEE 802.11p and IEEE 802.11b for vehicular networks in highway, urban, and rural scenarios under varying radio propagation environments is presented by [BG13] and summarized as in Table 2.2.

Although, IEEE 802.11p supports ad-hoc communications with benefits of low cost and straightforward deployment simply by embedding in vehicles, they suffer from scalability challenge. Drawbacks such as scalability issues, unbounded delay, intermittent connections, low capacity, and lack of Quality of Service (QoS) guarantees are discussed in [ACM12b].

Vehicular Delay Tolerant Networks:
When a direct end-to-end communication between source and destination does not exist, a multi-hop data communication utilizing intermediate nodes is possible in Delay Tolerant Networks (DTN). For a road network scenario, the exchange of information via intermediate vehicles in an opportunistic way is modelled as Vehicular Delay Tolerant Networks (VDTN) and is most suitable for applications with sparse vehicular traffic and acceptable delay constraints.

Intermediate vehicles forward data either in a store-and-forward or store-carry-forward manner based on the density of suitable vehicles available for forwarding. Since, intermediate vehicles buffer and forward data to destination, buffer management and routing strategies of these vehicles significantly affect the data delivery ratio and end-to-end delay. Objective of a DTN is to minimize end-to-end delay and increase delivery ratio. To this end, flooding and epidemic message forwarding are the most common and suitable strategies of data dissemination.

Flooding involves a vehicle forwarding data to every other vehicle in its transmission range irrespective of its direction to destination or delivery ratio associated with that vehicle. This method of dissemination does not require knowledge of topology and increases the probability of data delivery by increasing redundant copies in the network.

Forwarding strategies, to choose suitable forwarding vehicles, require vehicles to have some knowledge of the network which is changing constantly, and thus can reduce the count of redundant data copies in network. An outline of available routing strategies in VDTN and their corresponding scenarios to which they are suitable is given in the work of [BSB+14].

As intermediate vehicles buffer data for a long time until a suitable vehicle is found to forward, buffer management at vehicles is significant. Removing obsolete and redundant copies from the vehicular network increases delivery ratio and reduces delay as buffers can occupy and serve new requests. An obsolete message removal mechanism is proposed in [dNGCdLCV17] based on acknowledgement sent by the destination node once it receives the message. Vehicle-Assisted Data Delivery (VADD) protocol proposed by [ZC08], achieves better packet-delivery ratio with lowest data-delivery delay compared to other VDTN strategies. Predictable vehicle mobility limited by traffic pattern and road layout is the basis for their work, and hence constantly changing network knowledge of vehicles is assumed.

2.1.2 Cellular Communication in Vehicular Networks

Long Term Evolution (LTE)[1] is considered as promising approach for high speed vehicular communication supporting high data rate and large coverage service area. LTE not only supports safety applications requiring high data rate with low latency, but also infotainment services such as VoIP, browsing, media streaming which generally are QoS sensitive and have high bandwidth demands.

Safety applications require frequent periodic exchange of state information among vehicles to create awareness in neighbourhood. This is supported in LTE by Cooperative Awareness Messages (CAMs) and Decentralized Environmental Notification Messages (DENMs).

CAMs which are also known as beacons or heartbeat messages are short messages exchanged periodically via broadcast by every vehicle to others in its neighbourhood. They usually contain state information such as position, direction of travel, speed of that vehicle, brake status combined with timestamp and vehicle ID. Events occurring on roads such as accidents, traffic jams can be trigged using DENMs. These messages are usually to alert road-users and hence sent via broadcast to a larger set of interested nodes.

Both CAMs and DENMs are forwarded via broadcast to a specific geographic region: CAMs for immediate neighbourhood (awareness range) and DENMs for a region affected by notification event that is triggered. In this way, *Geocast*, where a set of notifications are transmitted only to those nodes satisfying geographical criteria with spatial and temporal relevance is achieved in LTE.

IP based voice, data, and signalling transmissions help for greater coverage area which was lacking in earlier cellular communications. It follows a simplified architecture with reduced number of network devices that assists in reduced round-trip time and latency which is suitable for delay-sensitive vehicular applications. Since LTE rely on cellular infrastructure, they offer wide coverage and supports to bridge network fragmentation while extending the connectivity. Market penetration is also high compared to IEEE 802.11p as LTE network interface can be installed on existing smart phones and navigation systems. However, since LTE follows a centralized architecture, frequent beacon messages can overload the serving of special base stations called Evolved Node Base stations (eNBs), which handle radio communication with multiple devices in the cell and carries radio resource management and handover decisions. Other related issues of upload/downlink traffic and idle state of non-active terminals are of concern.

To handle aforementioned issues, such periodic beacon signals should be distributed directly among vehicles without involving eNB. LTE-Advanced Device-to-Device Communication (D2D) communication by [LKMK15], directly route traffic between terminals in proximity (spatially located) to lighten the load of eNB. They provide improved throughput, delay as well as spectrum efficiency as compared with simple LTE. Supplementary communication techniques such as millimetre waves, Multiple Input Multiple Out (MIMO), low power nodes, carrier aggregation with D2D communication are offered in advanced LTE. LTE-A supports communication also in heterogeneous networks having variety of radio access technologies such as WiFi, High-speed Packet Access (HSPA), Code-division Multiple Access (CDMA), and is scalable with different cell sizes and power levels.

A comparative study between IEEE 802.11p and LTE for suitability in vehicular networking under various vehicular networking conditions and parameters settings is carried in the work by [HMF14]. Both standards are evaluated for parameters of delay, scalability, and mobility. The results show that LTE is a front runner in all aspects for vehicular communications as it offers high data rate and larger coverage area.

[1] `www.3gpp.org/dynareport/36300.htm`, accessed on: 01.02.2018

2.2 Publish/Subscribe in Vehicular Communication

Publish/Subscribe is an asynchronous communication paradigm for distributed systems. Messages are produced by producers and relevant messages are consumed by consumers. As Publish/Subscribe is asynchronous, neither producer nor consumer maintain state information for each other.

Producers, do not send messages to specific customers, rather the produced messages are categorized into topics. *Consumers*, express interest to particular topics in the form of subscriptions without knowledge of whoever the producer is. With the help of *Broker*, subscriptions are managed and forwarding of relevant topics to consumers happens.

A broker can manage several subscriptions from multiple subscribers and hence they support information filtering and tracking for providing better efficiency. Multiple brokers can be implemented to handle scalability challenges. Subscriptions are stored at brokers. When a publisher publishes to a nearby broker, it is forwarded to a tree of other brokers Pub/Sub Tree (PST) to find suitable matching subscriptions, and furthermore to notify relevant consumers. Brokers filter information based on topics (topic based filtering), and to achieve even finer granularity, content-based filtering can also be implemented. Any deletion or modification of message are also subsequently notified to subscribers.

The distributed, asynchronous, and anonymous Publish/Subscribe communication is suitable in vehicular communication where traffic efficiency services similar to accident, jam information, and other road services are expected. The messages associated with these services, instead of flooding across entire network, would be relevant only to those vehicles which are travelling in that direction, or interested in that road segment. For example, parking slot availability in a city center, which changes frequently, shall follow continuous retransmission with updated free spaces. This service would be of interest to only those vehicles which are in that locality. Hence, while publishing, particular aspects of spatial and temporal dimensions must be considered.

[PEW09] proposes data dissemination using Publish/Subscribe by visualizing vehicular network as a structured Peer-to-Peer (P2P) network. Apart from distributed communication, both Vehicular Ad-hoc Networks (VANET)s and P2P networks are self-organizing. Nodes join and leave abruptly in P2P network which resembles to vehicle mobility in VANETs. Both networks suffer from frequent topology change, where nodes or vehicles can go faulty. Although there is strong resemblance among both networks, unique characteristic such as high mobility, low latency, and communication despite obstacles such as buildings is a challenge in VANETs.

Another challenge is the implementation of a broker. Considering the importance of a broker in Publish/Subscribe, it should be implemented in a non-faulty manner and should be scalable. According to the work by authors, they consider P2P network as a structured entity because, Distributed Hash Table (DHT) in structured P2P networks supports routing of request to appropriate peer. Information items have keys which are mapped to identifiers (similar to address space of peers). Information is stored and requests are forwarded to that peer whose identifier matches keys. Whereas in unstructured P2P network, requests are forwarded through flooding as information is stored at randomized peers. Another reason to follow structured P2P network is how well it organizes in case of a peer fault. When a peer fails, neighbouring peer takes over identifier range and corresponding information without affecting services.

In the work by [PEW09], the road network is divided into segments which form individual structured P2P network interacting with each other. The segment sizes are adjusted according to communication range supported by DSRC to enable vehicles within a segment to communicate directly. Within a segment, each vehicle exchange their status and topology information frequently yet directly. The address space of each P2P network is limited to that segment and they overlap logically only at the borders of adjacent segments. Hence, when a vehicle moves from one segment to adjacent one, proper exchange of status should be taken care. Mapping between vehicle address and their segment identifiers is established. Queries are forwarded to those vehicles where information is stored based on associated identifiers. Segment boundaries and borders are known via digital maps which are installed on each vehicle. When a vehicle knows that it is crossing to a new segment, it announces its departure from older segment, and via heartbeat message it first contacts nearest vehicle in the new segment to obtain information required to join new segment topology.

The work also supports information which can be relevant to multiple segments. To handle segment-overlapping information, vehicle moving to a new segment, identifies these overlapping items and publishes them once it is connected to the topology of a new-segment. To avoid redundant forwarding of such items, an information type can be defined which enhances searching and can be marked as read to avoid redundancy. Subscriptions here are not maintained by a tree of brokers due to cost involved in checking information against every subscription in multiple brokers. Instead, it is assumed that information items of a particular topic are always stored in that node. A subscription is stored in a node which also stores relevant information. Any subsequent modifications affecting that topic can be tracked by contacting only that node. With this approach, subscriptions and notifications of a particular topic can be managed efficiently as exchange is limited to only one segment. To manage application level subscription and overlapping information, when a vehicle enters new segment, only subscriptions of that concerned vehicle is published in new segment followed by a deletion in old segment.

[ZJZ+12] designed a content based Publish/Subscribe system middleware - APUS for VANETs. As anonymous, asynchronous communication in Publish/Subscribe is supported by middleware, challenges to implement middleware in VANET have motivated this work. Properties other than rapid change of network and intermittent communication, services in VANETs are usually having spatial and temporal relevance. This relevance along with scalable, reliable, low latent event notifications should be handled by an ideal middleware. Their design explores available RSUs along the roads to efficiently relay notifications after matching subscriptions with every event that is generated. RSUs take role of a broker to keep track of subscriptions and to forward matching event notifications. Vehicles as subscribers subscribe for event within spatial-temporal relevance and receive them within pre-designated areas or regions where RSUs can serve.

2.3 Information-Centric Networks and Vehicular Communication

Information-Cetric Network (ICN) is a new networking paradigm, where communication is centred on request/reply of the content. In ICN, nodes exchange information based on the names of the content instead of the IP addresses of the communicating end points. Regardless of location of the content, any content identified by unique name can be retrieved from anywhere in network since nodes care only for content names instead of the host to reach for data. This shift from a traditional location-based network to a content-centric network for content dissemination is suitable when provider and/or consumer are mobile. In-network caching described in Subsection 3.1.1, results in availability of data at multiple points, providing additional performance.

Popular ICN architecture include Content Centric Networking (CCN) ([JST+09]), Named Data Networking (Named Data Networking (NDN))[2], DONA, PURSUIT, NetInf. NDN is an extension of CCN. Both NDN and CCN follow Interest/Data packet combination to obtain required content. However, minor architectural design changes induced in NDN reduces content search time and interest looping issues.

NDN follows a hierarchical naming scheme similar to file systems which supports context and relationship of contents across network. Request/reply in ICN happens via corresponding Interest/Data messages. Every participating node in ICN maintains three data structures for communication. Content Store (CS): stores content generated and received by other nodes. Forwarding Information Base (FIB): helps for forwarding incoming request to next-hop towards content provider. It maintains name prefix and outgoing interface for interested packets. NONCE list: records all NONCEs of pending entries of satisfied interest to avoid interest loop. These entries are removed after particular time period. Pending Interest Table (PIT): contains list of pending interests raised by that node. Each entry includes name

[2] `http://named-data.net/`, accessed on: 01.02.2018

prefix, NONCEs, interest-incoming interface of the received interests. The entries are maintained as long as they are not yet satisfied or until lifetime expires.

A interest message is uniquely identified by NONCE and content name. When a node receives interest message for content, it first checks its NONCE list to verify if that content has been recently served or not. If no entry is found, the interest is checked with PIT to see if it is already been forwarded. If it is a new interest, then the NONCE and name along with incoming-interface is stored as an entry in PIT and interest is forwarded. If an entry already exists, the new interest is tagged to the existing ones.

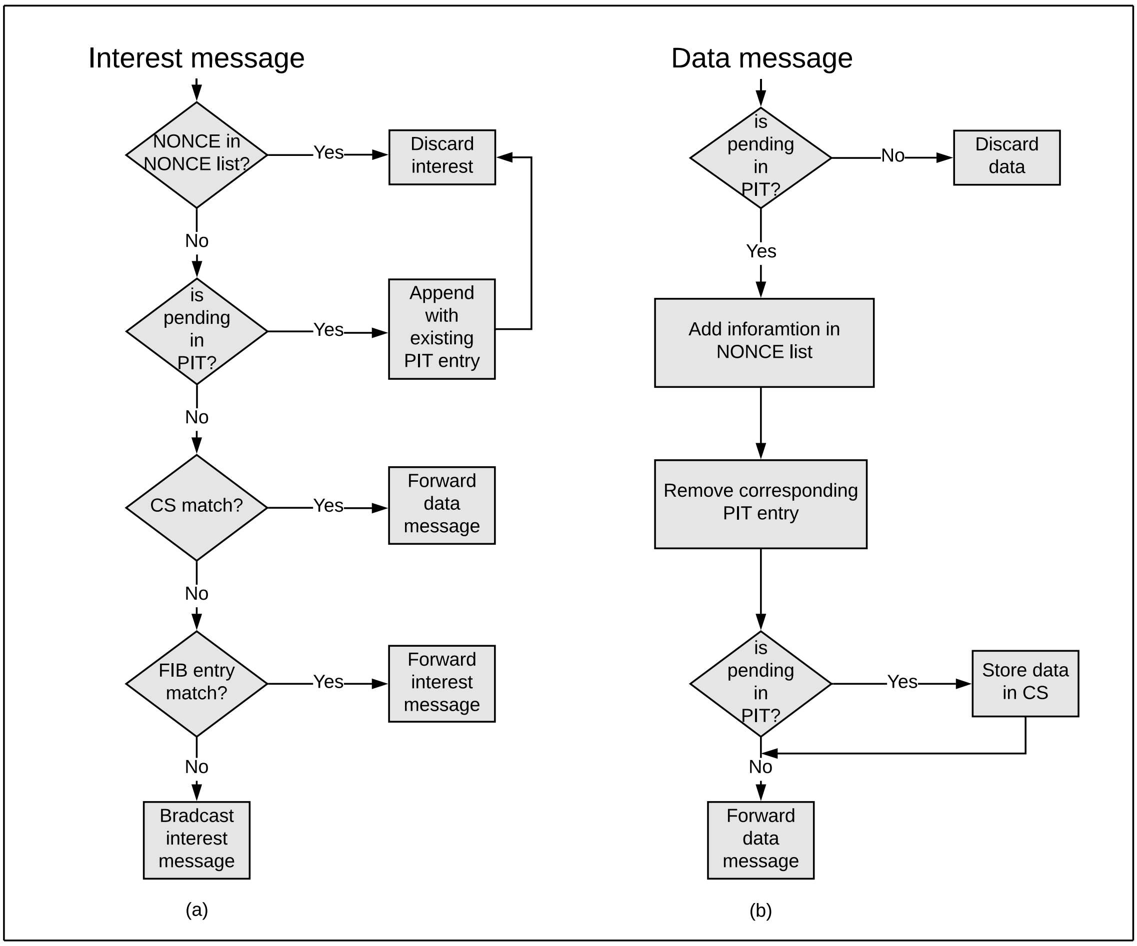


Figure 2.1: Operational decision of (a) Interest message and (b) Data message, (adapted from [ABK+17]).

When a node receives data message, it checks its PIT entries to find matching one. If no record is found, the data message received is dropped. With a matching entry for a received data message, it is forwarded, and corresponding entry is removed from PIT followed by adding NONCE in NONCE list. When a node receives already satisfied data message from a path with longer delay, interest loops occurs. This can be avoided by checking interest's record in NONCE list. The operational decision of interest and data message is as shown in Figure 2.1.

Applying CCN design for vehicular communication is challenging because ICN are not designed for wireless applications. Typically, in wired network, interests are not forwarded to same node from where they were received. Whereas in wireless networks, due to node failure they can be relayed if needed. But with in-network data caching and content replication features of ICN, location and time relevant vehicular applications can benefit. Transparent packet-level caching available at every node can be leveraged for vehicular networks where high mobility is implicit.

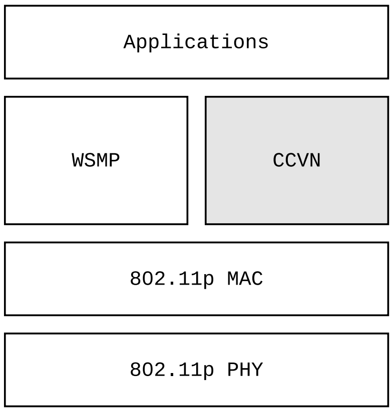


Figure 2.2: CCVN on top of IEEE 802.11p layers as a replacement for TCP/IP (adapted from [ACM12a]).

In the work by [ACM12a], a CCVN architecture is proposed that is based on CCN for applications with dynamic, short lived, intermittent connectivity features which are commonly found in VANETs. Design of CCN for VANETs deployed on top of IEEE 802.11p layers is as shown in Figure 2.2.

CCVN protocol runs on all devices including vehicles, RSUs where IEEE 802.11p is enabled. Every CCVN node is uniquely identified by NodeID irrespective of its location. Content is identified uniquely by a persistent name: ContentID (CID). Content can be made up of multiple packets where each packet has unique PacketID (PID). A packet belonging to a particular content is retrieved by a unique combination of CID and PID. Similar to interest and data messages, CCVN relies on Interest and C-Obj packets.

[BCG$^+$14] proposes ICN architecture for vehicular networks utilizing in-network caching where contents are pre-fetched and distributed to maximize probability of using required data. Vehicles move along a known path which is mapped with Access Point (AP). This sequence of known APs serves content delivery from a fixed content catalogue while vehicle is on the move. Content can be categorized into chunks based on their consumption period, and different chunks are delivered from different APs only when required. This design provides optimal content distribution with guarantee and at low latency.

A V2I scenario is considered where vehicles move in a predefined location containing multiple APs and vehicles download content from them. Hence, APs are connected in a tree topology forming parent-child relation, where parent (root) contains all chunks of all contents. A vehicle connects to first AP in its path and requests first chunk (*j*) for a content made of (*Sj chunks*). Every AP has content store capable of holding some chunks. If an interest receiving AP has been requested for a chunk in its content store, it forwards to the vehicle, and if not, it forwards interest to upstream AP. Similar forwarding happens along the hierarchy of APs until data is found. Vehicle keeps requesting a AP by interest message until it is satisfied or it moves to a new AP. In case, if it moves to a new AP, pending requests are requested again until entire content is retrieved.

A similar work is conducted in [dSCC16], where contents are pro-actively cached along the APs known by the vehicle's trajectory to its destination. [GPW$^+$13] proposed Vehicular Named Data Networking (V-NDN) based on NDN, where a vehicle can play role of data consumer or data producer or data forwarder or as a data mule. This data mule comes handy when there is no connection available and a vehicle can follow store-carry-forward methodology. It is designed for traffic efficiency services, and hence, V-NDN encodes geographical information in data names, so that all relevant interests can be forwarded to that location where desired data is produced. On the contrary to the working of PIT in NDN/CCN, this design optimizes by saving data message even if an entry in PIT is not found. For supporting dynamic mobile vehicles, this approach lets every node to cache received data regardless of whether a matching entry in PIT exists or that data is relevant in future or not.

By using the millimetre wave (mmWave) technique, the fifth generation (5G) networks can provide high bandwidth to satisfy the requirements for high quality media access and alleviate the traffic pressure. However, due to shorter transmission range of mmWave leads to suffer from frequent handoffs and connection delays which degrade the Quality of Experience (QoE). To overcome this issue, integration of fog computing is proposed in [GLL17]. They propose integration of Software Defined Network (SDN), cloud and fog computing in 5G in their network architecture with three logistical planes: data plane, control plane and application plane. The control plane includes Road Side Unit Centres (RSUCs) and SDN Controllers (SDNCs), where as data plane contain RSUs, Base Station (BS)s and vehicles. Application requirements and services are handled by application plane. [ABK+17] propose an architecture that combines SDN functionalities in vehicular networks to get relevant content using NDN.

3 Related Work

In this chapter, we introduce to two important concepts concerning our thesis, namely, Caching Systems and Information Freshness. First, we discuss caching approaches in distributed systems and vehicular networks. Following that, an overview of methods to maintain information freshness in mobile ad-hoc and vehicular networks are presented.

3.1 Caching Systems

Caching systems are meant to increase performance and achieve scalability. They take advantage of principles of temporal and spatial locality while accessing a data source. Instead of repeated access to a data source, which can be remotely located, caching systems store temporarily those frequently accessed data, thereby reducing the associated costs of repeated access. *Cache Hit*, is said when cache contains the data that a client is requesting for, and in contrast, a *Cache Miss* occurs when cache does not contain that requested data. Cache performance is measured as the percentage of successful accesses handled by it, i.e., *Cache Hit Rate*. The goal of a caching system is to achieve high cache hit rates as possible. The following sub-sections give an overview of caching schemes in distributed and vehicular networks.

3.1.1 Caching in Distributed Networks

Internet is a large distributed network where a single page request requires hundreds of lookups which should be served in a fraction of a second. Connection runs through number of firewalls, gateways before reaching the remote server. During early days of Internet, FTP was the major traffic with FTP file transfers on NSFNET. With increased penetration of Internet, more than 70% of today's traffic is constituted by HTTP traffic requiring fine QoE. Global scale large volume file sharing is supported by peer-to-peer applications such as Gnutella, KaZaA. Without scalability solutions similar to caching and load balancing, it would result in higher network load and unacceptable user experience.

Web Caching

The World Wide Web (WWW) is a large distributed information system with shared objects. With exponential growth of web users in the recent years, issues of server overloading and network congestion are increasingly affecting web users' experience. Web Caching is an efficient method to reduce loads on data servers as they scale the Internet. A proxy server where web caching is implemented, can serve the client instead of by a remote data server, thus reducing the network bandwidth usage as reported by [CDF+98]. Additionally, when a proxy server successfully serves a client, i.e., a *Cache Hit*, it reduces the user-perceived delay and consequently enhances the web users' experience. Other advantages associated are including but not limited to reducing data server's workload and serving the client temporarily in case of a server failure. The effectiveness of web caching depends on the architecture of caching system, proxy placement, and cooperation among proxy servers. Cooperative caches that cache non-identical data objects trust one other and assist each other to increase overall hit rate.

Hierarchical Caching:
Hierarchical caching proposed in the Harvest Project by [CDN+96] arranges a number of caches in a hierarchy, similar to a tree-like structure. The child-cache can query the parent-cache but not the other way round. In this way, the data is filtered down to a client's cache in browser, which generally is the bottom-level cache in the hierarchy. When a request is not served by client's cache or by any intermediate caches, data retuned by remote data server is cached at all the intermediate caches for future accesses.

Adaptive Web Caching by [MNR+98], a kind of hierarchical caching method, where multiple caches can join or leave a cache group dynamically to match the content demand is proposed. In this approach, a parent of a group of caches can be a child in another group or vice-versa, and thus being adaptive as per needs. Although, hierarchy in cache implementation reduces remote data server accesses greatly and provides scalability on demand, however, caches higher up the hierarchy become bottleneck with increased cache-miss at lower levels. Subsequently these cache-misses increase query delay as reported in [RSB01] and [TDVK98]. The hierarchical caching approach also stores redundant copies of a data at different cache levels which is unessential at most times.

Distributed Caching:
Distributed caching was proposed by [JH97] to overcome the shortcomings of hierarchical caching. Instead of storing the documents at both leaf- and intermediate-level caches, only leaf-level caches store the actual content. On the contrary, intermediate-level caches only have meta-data having information of which documents are stored at which leaf-level caches. When a cache-miss happens, leaf-level cache resolves the query from remote data server and updates its entry, followed by successively advertising this event to upper level caches. When data requested by a client is not served by its leaf-level cache, the immediate intermediate-level cache checks for availability with other leaf-level caches and acts as a pointer to the cached object.

Since, in this scheme, caching happens at low-levels, they offer better load sharing by erasing bottlenecks at upper levels. However, due to leaf caches servicing the requests, when the demand for a particular cached object is queried often, they become the performance bottlenecks. In this case, they need to serve their own clients and also the forwarded requests from other caches.

Hybrid Caching:
In hybrid caching, caches have the flexibility to cooperate and query with both parent and neighbouring caches, thereby increasing the chances of a cache-hit. In Internet Cache Protocol (ICP), by [WC97] that follows this method of caching, a document can be retrieved either by the parent or by neighbouring cache based on the lowest Round-trip Time (RTT). The work of [RSB01] shows shorter connection time with hierarchical caching and shorter transmission time achieved with distributed caching. The advantages of both schemes are exploited in a well configured hybrid caching.

The effectiveness of web caching depends on understanding demands of web users and to efficiently organize available caches. Cache coherence mechanisms to maintain consistency of cached object with that to the master copy in data server, along with anticipating the future possible request and pre-fetching them are other things that influence web caching significantly.

Peer-to-Peer Caching

P2P applications are organized in a decentralized manner where a peer can function as a client, a server, or as a router. A peer in any of these active roles also serve other clients unlike in a client-server model, where client only demands services. With high interconnections among the peers, a P2P model offers high scalability and fault-tolerance. For this reason, information is not located in a central entity but instead, it is partially distributed among all peers. Hence, a peer requesting a query may have to communicate with multiple peers to gather information. Thus, techniques of web caching may not be suitable for P2P applications where peers frequently connect and disconnect from the network. Caching approaches can be categorized broadly in P2P applications based on where a cache is placed.

Caching at every peer:
In the first approach, cache is implemented in every peer that stores both query strings and results which pass through them. *Gnutella*, an unstructured overlay infrastructure, works based on flooding, where a peer broadcasts the received query to directly connected peers. Gnutella caching method by [Mar02]

proposed specifically for Gnutella peers, identifies locality patterns, where a peer tends to submit similar queries frequently.

A query request and reply process in Gnutella architecture is shown in Figure 3.1. When a query request is received, a peer forwards that query to its immediate neighbours in order to get a reply. In turn, these neighbouring nodes forward that query to their neighbours as depicted. Upon receiving a query, a peer checks for a matching file and replies with matches if found.

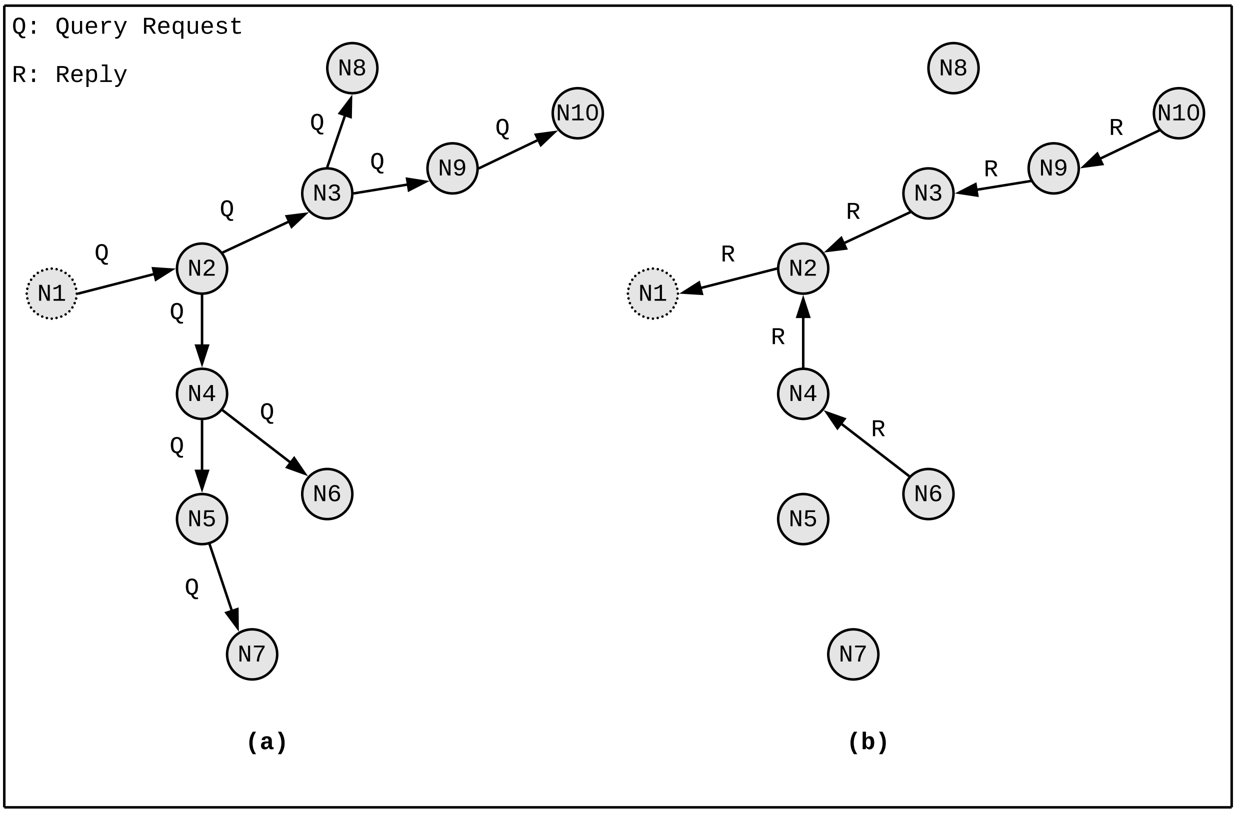


Figure 3.1: (a) Query Request and (b) Query Response in Gnutella (adapted from [Mar02]).

Query response are always forwarded back to the requester using reverse path of query requests. Since each peer floods query to neighbours, to avoid flooding, TTL for every query request is associated. Before flooding to neighbours TTL is decremented and when a peer receives request with TTL value of one, it ignores it. Within a valid TTL duration of a request, a peer can receive the same query request again, and this overhead is resolved by maintaining a record of already received requests with their unique identifications. Although, Gnutella peer-to-peer system have burst traffic pattern with peers joining and leaving the network frequently, caching the query responses even for a short time is proved significant as reported in their work.

Caching based on locality measurement:
In the second approach, instead of caching at each peer, a centralized server manages caching. The transparent query caching scheme proposed by [PH03], makes use of high locality measurements in the queries going through a gateway of an organization. Implementing caching at the gateway is proven more effective as P2P traffic and query traces of all peers can be tracked. This way, if a query is issued, it is more likely for the centralized server to receive such queries and can find similar queries stored in cache. Hence, cache hit rate of up to 60% and significant reduction in uplink query traffic is observed.

Caching at selected peers:
In the third approach, caching is implemented on selected peers instead of caching query results in all peers along inverse query path as followed in Uniform Index Caching (UIC). The redundant and duplicate caching along all peers decreases search efficiency of a peer. [WXLZ04] proposed distributed caching mechanism where query results are cached across only selected neighbouring peers. Distributed Caching and Adaptive Search (DiCAS) – protocol is proposed which encourages forwarding of query to the peers with high probability of serving them.

According to DiCAS, which is based on hashing, peers are logically organized into layers or groups, and each group is uniquely identified by a Group ID. For a query issued, hash code is calculated as: *query ID = hash (query string) Mod N*, where N is the number of groups. Following this, now the query flooding is restricted only to those peers in the group where Group ID equals Query ID. The experiment results presented in [WXLZ06] following DiCAS show significant reduction in network search traffic as search cost is highly reduced due to shrink in searching space and restricted flooding.

Caching based on DHT:
In structured P2P overlay infrastructure, DHT techniques are followed, where every peer and data object is assigned a unique identifier. A data object is mapped to the peer with closest identifier. This peer and content localization is used to direct and search queries. Although they increase search efficiency and provide high recall, designing sophisticated index for highly dynamic network is a daunting task. DHTs are more suitable for single-term lookups by hashing terms into keys, whereas to address efficient multi-term queries in structured overlays, Distributed Cache Table (DCT), proposed in [SA06] is suitable. It avoids maintaining index entries that are rarely used which occupy and increase search space by making use of query-driven strategy adapting to the query load.

Caching in Information Centric Networks

In ICN, the focus is on the content but not the content provider. This approach of decoupling content from the location of the content leads to revolutionary addressing mechanism using unified content names instead of host content addressing as in traditional IP based internet. Named Data Network (NDN) and CCN are popular ICN based architectures focusing on supporting efficient and scalable content retrieval. In such receiver driven content oriented applications, ubiquitous and transparent cooperative caches form the fundamental building blocks. Caching in ICN is categorized as in-network (on-path) and off-network (off-network) based on original content retrieval path.

In-network Caching:
In-network caching is based on caching repeated contents formed by their access history referred to as frequency. [WBW+11], compares coordinated and non-coordinated on-path caching for performance and cost metrics associated with coordination. Objects are cached carefully using compare content algorithm to avoid storing duplicates and thus neighbouring nodes hold non-similar objects. This scheme ensures probability of availability of data locally, and thus saving load on server and network bandwidth in terms of routing. However, the coordination costs associated in scanning the topology before caching an object degrades performance.

In non-coordinated on-path caching policy, as there is no need to ensure resilience against duplicate cached objects, the cost associated with searching is absent. But, with duplicate entries, when a cache miss happens, the objects need to be fetched from the server increasing the hop counts and bandwidth usage. [PCP14], proposed probabilistic on-path caching for a Publisher/Subscribe paradigm, where having cache close to the source would yield better performance as it servers multiple subscribers. Based on hop count between publisher and subscriber, the optimal cache placement is predicted.

Off-network Caching:
The pervasive off-path caching in [XVS+14], requires additional registrations for routing information

that is based on hash function. [AAH15], proposed off-path caching in ubiquitous ICN using fog computing with additional process as suggested in [XVS+14]. Cache deployment strategies for better content retrievals have been studied where positioning of caches provide higher cache hit ratio. Placing a proxy cache close to client nodes reduces bandwidth consumption and enhanced response time. However, it suffers when a client forwards malicious content to a neighbour requesting same content. Reverse proxy is an approach, where proxy cache is placed close to publisher in order to overcome malicious client forwarding.

3.1.2 Caching in Vehicular Networks

The characteristics of a vehicular network varies from that of a distributed network because of inherent property of node mobility. Mobile nodes lead to frequent partitioning of wireless network established on ad-hoc basis. Fundamental properties found in vehicular networks such as high mobility, intermittent connectivity, and ad-hoc demand caching mechanisms differing from traditional ones. In this sub section, we discuss some of the approaches followed for caching in vehicular networks.

Using Named Data Networks

[GPP+14], apply the architectural design of Named Data Networking (NDN) for caching in vehicular networks and to address VANET challenges. NDN, consists of three main entities, namely, data producers, data consumers, and routers. Every entity in this architecture can be realized to as a node in the network each having the following data structures as mentioned in Section 2.3, (1) CS, (2) PIT, (3) FIB.

CS is the cache storage for data packets which would be of interest to the node in future. PIT stores all those requests that have not been satisfied yet and hence have been forwarded. Forwarding is done according to the topology information available in FIB. When a new request's entry is found neither in CS nor PIT, then it is forwarded and considered as a new entry in PIT. Request packets are routed based on names towards producers and data packets are returned based on state information at each hop. If the returned data packet has no corresponding entry in PIT, then it is dropped. Both request and data packets are addressed based on naming alone and carry no IP addresses.

Applying similar concept as aforementioned, Vehicular Named Data Networking (V-NDN) by [GPW+13], assumes a car on a road in four different roles. A car can be a data producer or consumer or forwarder or a data mule.

As a forwarder, a vehicle transmits query (forwards) to its immediate one hop neighbour, where in as a data mule, the vehicle carries data for distances when there is no connectivity available and then it follows store-carry-forward approach. Unlike accepting only those data packets that have an entry in PIT as in NDN, in V-NDN, a vehicle might cache all the data packets irrespective of matching PIT entry and whether it requires that data or not. This leads to the benefit of faster information dissemination by opportunistically caching all data packets. Having more data mules increases probability of finding a requested packet in a surrounding locality as they carry a copy of it and thereby enlarging data spreading areas. They also increase data availability since they can still serve with data and pass it around even when the data producer is temporarily off line. The experimented results show, effective data dissemination with this in-network caching, and particularly with data mules in a relatively limited area, 66% of interest packets were replied in the local caches.

Using Location-Aware Services

An ad-hoc service infrastructure over VANET is proposed in [DFNI07]. This infrastructure is based on a stateless application-layer communication protocol called Vehicular Information Transfer Protocol

(VITP), which is originally a contribution by [DINI05]. The proposed approach takes advantage of short-range, inter-vehicle wireless communication to provide location-aware services such as traffic conditions, parking availability and other road side facilities.

A VITP peer is a component running on a computing device in Inter-Vehicle Capability (IVC) enabled vehicles which has access to a vehicle's on-board sensors. The proposed protocol specifies syntax and semantics for exchange of messages among VITP peers of all IVC enabled cars. Complementing the needs of vehicular networks, VITP peers establish on-demand, ad-hoc exchange of vehicular services by operating as clients, intermediaries, and servers.

Service requesters are interested in events around a particular geographic location and can be constrained within a set of roads and segments. Accordingly, a location can be tagged with unique two-identifier tuple [`road_id, segment_id`]. `Road_id` is the unique identifier of the road and `Segment_id` is the segment along that road. An interest in the event happening in this location is requested by a service request generated using this tuple.

Virtual Ad-Hoc Server (VAHS), which is a group of VITP peers that are moving within query's location area, computes replies to the service requested. VAHS is a best-effort approach, where the group is formed on-the-fly. No knowledge of VITP Peers that are active group members of VAHS is maintained, complementing stateless VITP protocol. An incoming query can be served by an intermediary peer if the query matches the semantics and the specification of target location of a matching entry exists in its cache. With a match, query is replied with the cached data and transaction is complete. If not, it is retransmitted to its target location. Due to this return condition, where either an intermediary or a peer from specified location completes the transaction, cache control headers are implemented in VITP messages as an extension which act as directives for a VITP peer caching decision. The effects of shorter and longer TTLs associated with query replies along with geographic routing with increased hops to target location are studied in [LPD10]. The work depicts with cache-based VITP increases information accuracy over 65% and reduces up to 12% of network overhead.

Using Peer-to-Peer Cooperative Caching

Vehicular networks are relevant and can be realized both in urban and sparse road networks where the density of vehicles vary significantly.

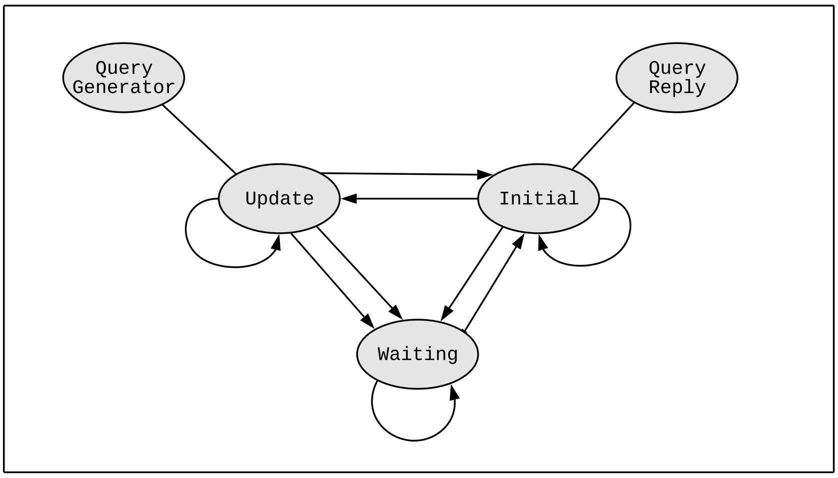


Figure 3.2: Markov chain model with *Initial*, *Waiting* and *Update* states, (adapted from [KL14]).

The work by [KL14], propose data dissemination in urban networks using peer-to-peer cooperative caching among vehicles and also with RSU. Due to higher density in these networks, to minimize load on wireless infrastructure with respect to exchanged messages, P2P approach using Markov chain model with three states is considered as shown in Figure 3.2.

The vehicles exchange information of roads and routing decisions in a cooperative manner and this data is stored locally. This data helps in serving future requests when a query from other vehicles is requested. If the data for a query is available in local cache, it is served at once to the client and receives a new query. If data is not available, it is passed to other vehicle and changes to waiting state. Once a reply is received, it is forwarded to client and updates its state to initial or updated to process a new query.

In this process, a vehicle's state may remain in same state or change to waiting or to update state based on probability. The probability is directly proportional to the time in waiting state and how frequently a data item is accessed in a given time interval. Cache replacement happens when cache is full and new entry needs to be added. Higher the time spent in waiting state, higher is the frequency of access and hence, less probability to replace this data item from cache. As vehicles not only exchange information but also exchange query with each other in peer-to-peer cooperative way in hope of resolving, the results depict reduction in congestion and query delay.

Another work [LS11] is based on pull-based geo-cast protocol. With the help of Global Positioning System (GPS), a request-reply geocast data dissemination protocol is used. Geo cache is implemented on a previous work by [LC09], where a map of congestion indexes for a sector of road is pro-actively and cooperatively implemented among each participating vehicle. The map is continuously updated using geo cast protocol. Road state information is stored locally which is collected by other vehicles in a peer-to-peer fashion. As communication of road events happen usually within the same geographical region, geocast protocols are efficient means to constrain flooding of queries within a particular area of interest.

Every query message carries Zone of Relevance (ZOR) value, typically representing the coordinates of that region a vehicle is interested in. Based on GPS coordinates, a vehicle decides to search its cache if its current coordinates lies within the perimeter of the ZOR represented in query message. To limit the number of broadcast messages, in this approach, first, a vehicle checks for matching entry in its cache and if available it is directly served to the client. If a matching entry is unavailable, the vehicle simply re-transmits the query to its one hop neighbouring vehicles. This happens iteratively until either a vehicle can resolve query or till the entire ZOR is flooded.

A vehicle receiving a reply from answering vehicle makes sure to cache it locally for a predetermined *cache_time* before forwarding it to query generator. This helps to reduce query delay for future requests involving same query. Once the *cache_time* expires, data is removed from cache. This value can be calibrated based on mobility, dynamic, and congestion degrees of the network. For a client S with H hops away from the answering node D, and with L being average length of request message sent between them at a transmitting rate of R, the *cache_time* is given as: *Cache_time* = (V / d) ; where V is average velocity of vehicle and d is given as *(2.V.H.L / R)*. The simulated results in their work show that due to caching, reduction in both the number of messages generated and subsequently sent via broadcast by intermediary vehicles is observed. Furthermore, the response time improved with the use of caching.

3.2 Information Freshness

Caching improves performance and scalability of a communication system as discussed in previous section. Cache consistency is an important aspect of any caching system and plays a significant role in serving the motive of a cache. No good a cache if the stored data is stale or not valid any more. In fact, in that case, a stale data causes high latency and delay leading to overhead than being beneficial. Freshness of a cache is based on the age of its content and signifies for how long the cached value can be used without validating with the original data source. Therefore, cache consistency and coherency policies are implemented for timely validations with original data and thus to maintain freshness of cached data. In the following section, we give a general overview of available cache consistency mechanisms in ad-hoc networks, and furthermore, we look at the corresponding policies available in vehicular networks.

3.2.1 Information Freshness in Mobile Ad-hoc Networks

Ad-hoc networks are self-configuring, wireless, infrastructure less communication networks where communicating parties need not necessarily be within a direct communication range. Due to frequent disconnections and mobility of nodes, traditional cache invalidation strategies are not efficient. In these networks, it requires nodes to forward both requests and replies between source and destination via multi-hop forwarding. Therefore, each node can be seen either as a router to forward or as a source or as a destination node. In general, cache consistency schemes can be classified as the Pull or the Push approach based on the initiator who initiates invalidation process.

In Pull based approach, a cache node is responsible for maintaining its consistency with the server. TTL is accompanied for every data item that is cached, where cache node considers data to be valid only as long as TTL is valid. Upon expiration of TTL, it is caching node's responsibility to query the server for updating. [FA13], suggested adaptive TTL based consistency scheme with features of piggybacking and perfecting to increase freshness and reduce query delay.

In Push based approach, the server initiates update by invalidation reporting. For this initiation, a server maintains state information about its clients and their corresponding stored data. When a data item is changed, server sends a corresponding invalidation report containing the updated value and current time stamp to the respective client. [MA10] proposed a server-based update mechanism called Smart Server Update Mechanism (SSUM). In this approach, a special node called *Query Directories* contain the queries requested so far and their corresponding caching nodes which store those data. Control mechanism is implemented to adapt data caching according to the popularity of data and its update rate at the server node. [ASM+08] proposed Cooperative and Adaptive Caching System (COACS), which is a mobile distributed system maintaining indexes of queries in separate nodes called *Query Directories*. The contents are stored in *Cache Nodes*. This approach suggests efficient cache invalidation in addition with replication and re-use schemes. [CZCX07] proposed periodic Invalidation Report (IR) based invalidation scheme, where server sends periodically a list of updated data items. While answering to a query, a node waits until it receives a IR notification to validate whether its stored value is valid or not. [Cao03] improvised periodic IR based invalidation method with Updated IR (Updated Invalidation Report (UIR)) reports occurring in-between successive IR's to decrease associated delay. [HXD+13] proposed a Greedy Walk-based Selective Push (GWSP) approach, where server sends updates to only those caching nodes that are more likely to serve queries when there are no more updates before Time-to-refresh (TTR) expires. In this case, server also maintains state information of cached data, TTR, and query access rate for each caching node.

IR based invalidation:
In the methods proposed by [HL98] and [WYC96], a IR based invalidation for mobile environments is conceptualized. In a IR based invalidation model, the server broadcasts IR for every *L* seconds, consisting: current time stamp (T_i), tuple set (d_x, t_x), so that $t_x > (T_i - w*L)$, where (d_x) is a data item with Id = x, (t_x) is recent update time stamp for (d_x), w = broadcast window size of invalidation. The formula signifies that, every IR broadcast contains update history of past *w* broadcast intervals.

To answer a query, client waits for next IR broadcast to validate its cached content. If it is valid, right away a reply is sent to requesting client, and if it is not valid, a query request to server is sent for updating. In Figure 3.3, when a query is received between $(T_{i-1}$ *and* $T_i)$, it waits until T_i to receive a IR report. Hence average latency associated is a sum of query processing time and half of IR broadcast interval. The advantage of having update history of past *w* intervals is that, even when a mobile client is disconnected during a IR broadcast, client still has the opportunity to validate its content and remain

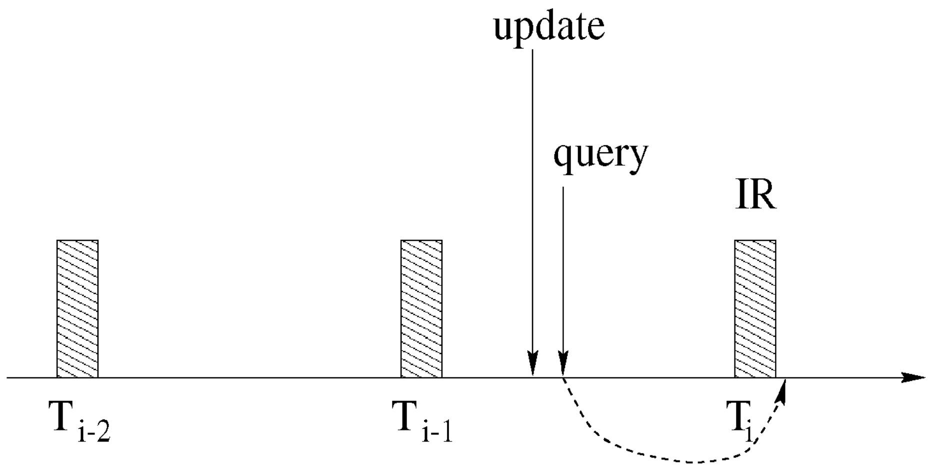


Figure 3.3: IR-based invalidation (adapted from [Cao03]).

updated as long as disconnected time is less than the product of *(w.L)*. If the disconnection time is more than or equal to this value, then client has to remove all its entry and start fresh by querying server.

However, there are inefficiencies related to this approach. First, client has to wait until next IR update from server to answer a query, due to which long query latency is associated. Second, when a *hot data item* (frequently accessed data) is updated at server, the clients have to query server for this update individually which significantly increases bandwidth.

UIR based invalidation:
To address the inefficiency of a straight forward IR invalidation, [Cao03] proposed a scalable IR invalidation for low-latency. UIR is proposed which contains only those data items that are updated since last IR. These UIR are sent between successive IR's.

In Figure 3.4, a UIR operation is shown. $T_{i,k}$ represents the time of the k^{th} UIR after the i^{th} IR. For a query received between *(T_{i-1} and T_i)*, it can answer the query at $T_{i-1,2}$ instead of T_i. Hence, client needs to wait only for the next UIR or IR, whichever arrives earlier to validate its cached data. If a valid cached copy of the requested data item exists, then client replies the data item immediately. Otherwise, a query is requested to the server.

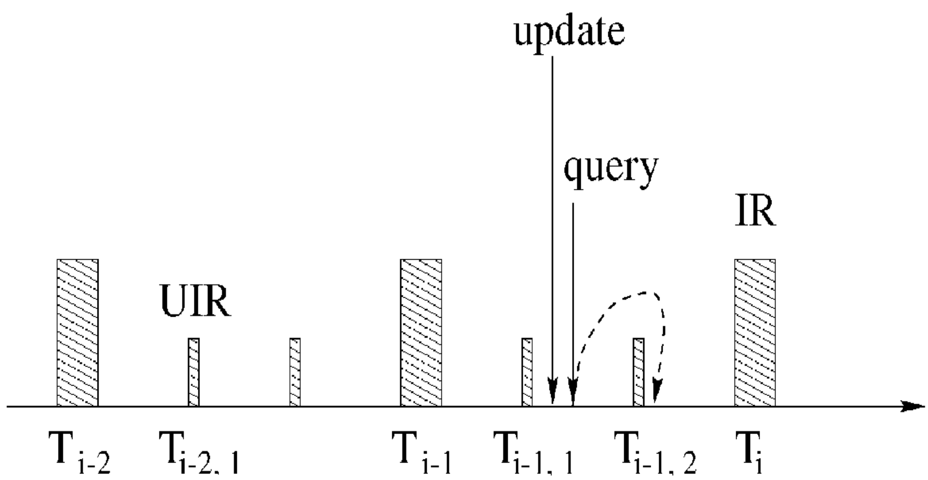


Figure 3.4: UIR-based invalidation (adapted from [Cao03]).

But, compared to previous approach, it is important that client is not disconnected during IR reports because UIR consists of only the recent updates only since last IR unlike IR-based invalidation. A client can confirm if it has missed any IR by comparing the time stamp associated with the UIR and the time stamp of the last received IR. However, the advantage is two fold. First, since it is UIR, instead of having complete update history, it contains only updated data items since last IR, and thus, saving large bandwidth and time associated for validation. Second, due to UIR between successive IR's, now clients waiting time to answer a query is greatly reduced.

Although, UIR based approach increases bandwidth utilization while decreasing query delay, but when there is a cache miss, still a query needs to be fetched from server. This overhead is overcome in [Cao02] where authors propose pre-fetching data which can be used for future queries. Based on pre-fetch-access ratio, caching is dynamically adapted for frequently accessed data. When a server broadcasts a data item for which an entry in client is invalid, an intelligent client would update this data, so that it saves uplink and downlink costs between server and client for future request associated to that item. In this process, server first broadcasts only the list of ID's of data items which are frequently accessed before broadcasting each of the updated data item. Following this, clients can verify this list with ID's of its cached items, and in successive broadcast from server, client updates only the required data item.

To identify list of frequently accessed items, server can be imagined as stateful or stateless entity. In stateful method, server maintains a counter for each data item and is incremented each time a query associated for it is received. But this method is not scalable due to involved computational costs, and in case of a server failure, has bad effects. In stateless method, server does not immediately answer to a client request, instead, waits until next IR broadcast and only then sends list of data item ID's which were requested between IR's. In other words, this list contains frequently requested items which are indeed frequently accessed items.

Cache consistency is maintained by adjusting the UIR based approach appropriately. Here, to deal with frequent updates, multiple broadcasts of these frequent data items are done between IR's and UIR's. [IVB97] proposed indexing techniques instead of caching for frequently updated items. In this method, broadcasts include index and data item together, and is sent every time it is updated. However, if the updated data item is large, this method scales badly. Hence, a data item is categorized as *hot* (frequent accessed) or *cold* (rarely accessed) data items based on their frequency of access. Indexing methods are effective for hot items, whereas IR based methods for cold items that are relatively large.

CacheData and CachePath:
[YC06] have proposed cooperative caching using *CacheData* and *CachePath* schemes. In the first scheme, an intermediate node caches the actual data and serves requests directly. A node, caches only popular data item which are frequently queried for. When a node receives query for data item (*di*) from more than one node, it assumes that as popular item, and decides to cache its value locally when it passes-by via a reply. As a rule, to avoid redundancy, a node does not cache if data requests for a data item are all from the same node.

In Figure 3.5, both N6 and N7 request (*di*) via N5, and hence, N5 assumes that (*di*) is popular and caches it locally. Future requests by N3, N4, or N5 can be served by N5 directly.

In *CachePath*, node caches the path to the actual node that has the data, and forwards queries to that node instead to the data source. In Figure 3.5, assume, node N1 is the requester for a data item (*di*) from N11. When N3 forwards the data (*di*) back to N1, instead of caching that data item, it only stores the information that node N1 has a copy of (*di*). In future, if N2 requests (*di*), N3 knows that the data center N11 is three hops away, whereas N1 is only one hop away. Hop count information between source and destination can be known by Dynamic Source Routing(DSR) or Ad-hoc On Demand Distance Vector (AODV) routing algorithms. Another advantage is that, a node need not cache path for all passing-by values, and it is done only if that node is close to the caching node that also has data other than the data source. As in Figure 3.5, though, forwarding nodes N4 and N5 are closer to N11 which is source, only N3 which is closer to caching node caches cache path to N1.

Due to bandwidth power constraints in ad-hoc networks, a weak TTL based consistency is considered. A node considers a cache copy (both CachePath and CacheData) as valid as long as TTL is valid and any requests for expired items are forwarded directly to data source in that case. As an optimization, a node can still keep list of ID's of invalidated items, so that, when a matching passing-by value is seen, it can

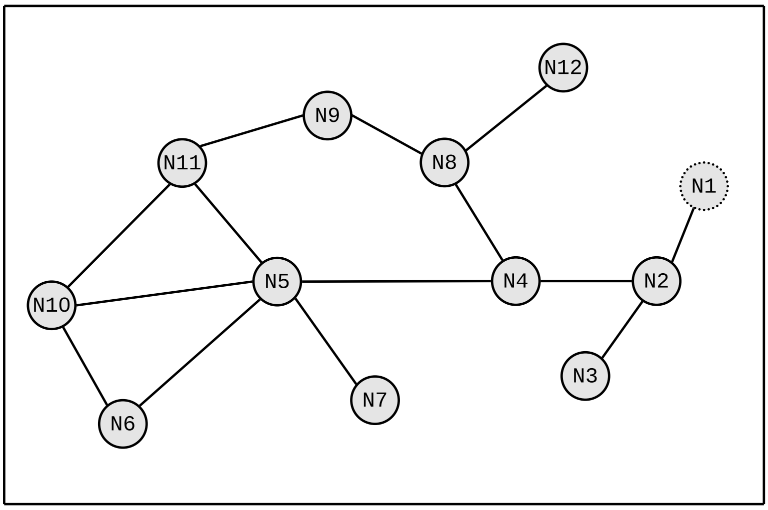


Figure 3.5: An Ad-hoc Network (adapted from [YC06]).

be updated immediately. But care is taken not to keep this list longer in cache and is removed when not refreshed for a duration of original TTL value.

[ASM+08] is an improvisation to the [YC06] approach. They propose COACS which is a distributed indexing scheme that makes use of query directives to locate nodes storing data items. Nodes, in this approach, can take up either of the two roles: a Query Directive (QD) or a Caching Node (CN). QD's task is to cache all received queries, and CN stores data items which are replies to queries. When a node requests data that is not cached locally, it queries data source. Upon receiving reply, the requesting node becomes a CN storing the data item, and a nearest QD caches query with its corresponding CN where that data item can be found.

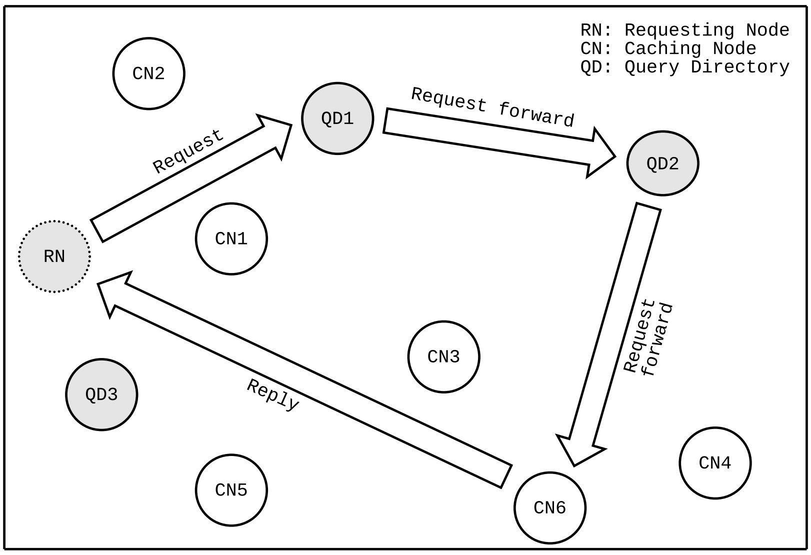


Figure 3.6: An exampe of a COACS scenario (adapted from [ASM+08]).

Figure 3.6, shows how a requested query when not found in nearest QD (QD1) is forwarded to QD2 where a cache hit of CN is found (CN6). Although, *CachePath* and *CacheData* schemes look similar to this approach, however, they do not offer functionality of searching across all intermediary caching nodes. Hence QD are significant nodes which maintain distributed index of previously searched queries and their corresponding CN. This design has an advantage in maintaining cache consistency too. Instead of previous IR broadcasts to all CN, in this method, IR can be sent to QD's that know CN's for corresponding data items. Hence, QD's disseminate IR reports to only concerned CN. In this way, better hit ratio and lower delay is achieved.

Although, [ASM+08] has better consistency mechanisms, but since it is a Push-based server scheme, it does not scale well. With frequent disconnections with CN in ad-hoc networks, it incurs higher costs on server. [FA13] proposes pull-based approach of [ASM+08] with few modifications. The CN's monitor the TTL data and update frequency of data items, and are responsible for initiating validation with server. CN estimates item's expiry time based on duration of successive update interval. Piggybacking and pre-fetching is also considered to maintain higher cache consistencies. With the principle of piggybacking, expired TTL values are grouped for validation requests for update from server, whereas, unexpired or soon-to-expire frequent data items are pre-fetched from server. Comparison of content caching and pre-fetching techniques for mobile applications is outlined by [SKZ17]. Probability based pre-fetching based on user access history is proposed which has a good prediction rate leading to moderate cache hit ratio for small size caches.

[FAAK+15] is an extension of [FA13] that studies QD election procedure and replication of indexes in QD. An extended adaptive -TTL (EX- TTL) is proposed by [FK11], in which a CN broadcasts a copy of its cached item and its corresponding TTL to 1-hop neighbour for future requests, wherein consistency is maintained by implementing hash table with data item as key and its corresponding TTL as value.

Cooperative caching with pre-fetching for Wireless Ad-hoc Networks (WANET)s is proposed in [Den07]. In this work the author has considered clustered architecture for supporting localized caching. Cluster Head, Data Source, Caching Agent, Mobile Host constitute a cluster. Moderate levels of computation are required in electing the cluster head among mobile hosts considering network and mobility characteristics.

A data item from data source is stored in caching agent which should be optimally located within the cluster for reducing latency of access in the cluster. Based on mobility of hosts and latency of network, this location in terms of hops can be adjusted. Larger hop lengths to caching agent are acceptable for a stable network with lower mobility of hosts. Neighbour hosts store different data items increasing cache hit ratio within the cluster. Hence, they are cooperative, trying to serve a request locally. Hosts cooperate each other for data caching and pre-fetching a data item which is about to expire soon known via TTL. Information freshness is maintained via cache replacement based on TTL leading to weak consistency. Based on the metrics combination of frequency and latency of access associated with each cached data, cache replacement policy is employed. This method increases cache hit ratio irrespective of network traffic. Although the method is valid for multiple data source and works for any routing protocol, but, since they consider cluster architecture and cache are stored in different hosts for optimization, network partition costs are high. When a caching agent moves to other cluster then query has to be served through inter-cluster communication increasing latency. To overcome this they use duplication of data items to reduce network partition effect.

3.2.2 Information Freshness in Vehicular Networks

Caching scheme considered for Mobile Ad-hoc Networks (MANET)s would not be suitable for VANETs due to fast mobility of nodes in the later. Vehicular networks are prone to frequent disconnection and random walk. Therefore, significance to query delay and cache hit despite high mobility and faster updates is of concern. In this type of network, it is unlikely that adjacent vehicles carry similar data for optimization reasons, and hence having IR based invalidation broadcast will be irrelevant to most vehicles. Consequently, a traditional IR based invalidation would be ineffective in vehicular networks. In this section, cache invalidation strategies in vehicular networks are discussed.

Time-to-live (TTL)

TTL based cache invalidation for vehicular computing is proposed using VITP in [DINI05]. In VITP, caching is supported by cache-control headers included in the messages that are exchanged among VITP peers. Associated TTL values are compared to decide if stored value is valid or not. An extension of this work in [LPD10] evaluates scenarios with different TTL values.

TTL = 0 seconds which signifies no caching and TTL = 200 seconds signifying indefinite traffic information cached in VITP peers is also analysed. With longer TTL values, traffic information dissemination is better across the network with increasing probability of finding requested data in a nearby vehicle. However, longer TTL values degrade the accuracy of information since higher chances that queries are resolved by local caches (which are not updated in high mobility network) instead of data source. Therefore, the accuracy of cached data depends on estimated TTL values, which in turn depend on rate of change of road network. Hence, adaptive TTL yields better results which consider vehicle density, road lengths, traffic jam and other real scenarios.

Location-Based Invalidation

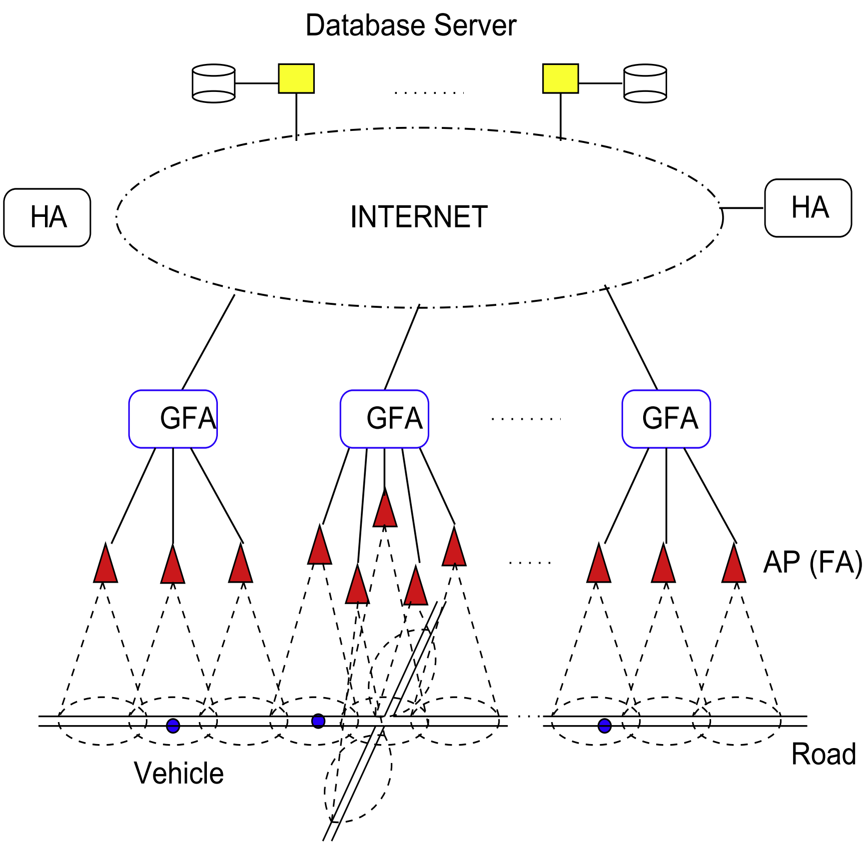


Figure 3.7: A hierarchical network model for IVANET (adapted from [LYD09]).

[LYD09] proposed Cooperative Cache Invalidation (CCI) scheme and its enhancement, Enhanced Cooperative Cache Invalidation (ECCI) based on location-aware hierarchical model. As shown in Figure 3.7, this model consists of location management agents such as APs, Gateway Foreign Agents (GFA), Home Agents (HA), and data server(s). This hierarchy facilitates scalable invalidation only to happen within specific-locations instead of flooding. Cooperative approach between servers and location management agents is supposed to exist where tight coupling of list of data items cached and current vehicles within a location exists. The location aware agents are hierarchically structured as shown in Figure 3.7.

APs are first point of contact for a vehicle to communicate, and multiple vehicles can be present in a single AP. A set of geographically closely associated AP's are grouped under one GFA. When a vehicle moves from one AP to adjacent AP, it correspondingly un-registers and registers respectively. This event is also recorded in GFA and HA. Both server and network agents maintain list of cached data items and current location of vehicles in their region. Due to this, instead of flooding IR broadcasts, a server can simply forward IR report to a relevant HA. The HA's compute furthermore to forward this report only to corresponding GFA for update where a vehicle has cached this data item. Instead of forwarding IR report by a GFA to corresponding vehicle, it waits for validation request pro-actively by vehicle. Advantage of this is that, either a cached vehicle would not need the updated item or it would be moving to a different AP's where that traffic information is irrelevant to it. Another benefit following this model is the non interdependency of cache update rate with query rate of vehicles. Due to this, a blind IR flooding incurring high traffic costs are overcome in this work.

In CCI scheme, whenever a data item is updated by server, it generates IR report containing mapping between latest value and vehicle in which it is stored. Instead of flooding this IR, as done traditionally, in this case it forwards to HA. HA maintains table to track vehicles that are currently located under which GFA, and can direct IR only to those affected GFA. Since multiple GFA's are grouped under a HA, it is important to know that based on matched GFA in the IR report sent by server, multiple IR can now be created by HA. And hence, size of individual IR created by HA can vary from the one initiated by server. In this way, communication happens only in relevant GFA and APs.

HA pro-actively updates vehicle handoffs among different GFA and forwards IR to new GFA under which the affected vehicle is currently located in. In cases where a query can be answered by adjacent vehicle, the vehicle having that value first validates with GFA via AP. If cached copy is valid, the vehicle resolves query. If not, GFA validates it with server and vehicle is updated subsequently after receiving reply from server. In ECCI, when a reply is sent back from server after a validation, GFAs' cache these values for serving future requests from vehicles and can avoid forwarding query to server.

Other location based invalidation is proposed in [CSLH10]. It follows update and query schemes to maintain consistency. In update scheme, at intersections, vehicles exchange a set of newest n – data items recently updated which are stored in each cache along with location information of that vehicle. Based on time stamps of exchanged data items, an item with latest time stamp is considered fresh. In query scheme, if a vehicle V1 needs to find a route to V2, it locally floods this query to search for a vehicle V3 which has location information of V2 irrespective of how old this data is. Once this query is received by that vehicle V3, it replies with information of V2 to V1 considering newest locality of vehicle V1. While forwarding request and reply, if any intermediary vehicles have newest information about V2, they forward the query to that vehicle dropping the information sent by V3.

Randomized Invalidation

In [FCC05], vehicles cooperatively exchange information which is cached locally. They get information from gateways connected to internet and which are forwarded using broadcast along the road. Every vehicle in this ad-hoc network generates request for some information at random time. The query is resolved by corresponding vehicle which has cached that item and replies to originator along the route query was received. The query generator caches the reply for some random time, and again requests, if needed in future. This pull based method reduces duplicate replies and aims to achieve higher information dissemination.

Probabilistic Estimation

Algorithm proposed by Hamlet [FMCC09], works on probabilistic estimation of content stored in neighbouring nodes. It is a distributed caching scheme, where every node independently decides what to store and how long to store. Each node estimates the content store in adjacent nodes, so as to store uncommon items in its cache, leading to content diversity where a query generator finds required data locally. Nodes exchange cached information in a peer-to-peer fashion. Upon receiving requested information, that node independently decides the validity of that item based on perceived presence of the same data item in its proximity. For this estimation, it is assumed a node can oversee request and reply packets and hence nodes work in collaboration. This obtained probability performs well for both low and high popular items

To summarize, most of cache consistency schemes in ad-hoc networks consider the following design aspects in their approach: Whether cache state is maintained or not? (stateless versus. stateful), Who initiates cache validity? (pull versus. push) and the level of consistency required? (strong, weak, hybrid).

In mobile environments, both stateful and stateless approach can be valid. When clients do frequently disconnect, a stateless approach is more efficient as client can move to another cell and overhead of state maintenance in server is reduced, but with stateless approach, client pro-actively has to validate its consistency with server. In strong consistency schemes such as, [KKG^{+}00] and [CXS06], queries are resolved either by an updated cache value or directly from data source. Weak consistency as in [WDCK04], a query need not necessarily be resolved by an updated value. Though this saves query delay and communication cost but staleness of data will have higher impact. Hybrid consistency scheme in [CZCX07] suggests that a node can flexibly specify required consistency levels for every data item based on trade-off among communication cost, staleness of data, query delay, complexity, and scalability.

Traffic safety and comfort applications in vehicular networks have different properties. Safety applications such as inter collision, lane change assistance, collision risk, and overtaking warnings demand strong latency constraints. They have high-data rate and might be of interest to small group of nodes. In these scenarios a push based approach is more suitable. Whereas, for comfort applications, such as, congestion status, traffic information, parking services, a pull based scheme is more suitable. This is because these applications are delay-tolerant without strict latency constraints, but requires this dissemination to a group of large nodes spanning relatively long distances from the source. Generally, push based scheme compels all the clients to update the data even when clients are not expecting to use them. This might lead to unnecessary bandwidth and computation costs if the client is not interested in frequent updates. Requiring to specific applications and as per their needs, a combination of discussed design aspects are considered.

4 Scenario Representation and Problem Description

This chapter introduces to the sample road scenario that is examined in this work. First, we illustrate the scenario considered with its properties and assumptions. Second, we state the objective we are trying to achieve in the whole. Third, we evaluate the suitability of different communication mechanisms for the objective in hand before choosing an appropriate one. Later, reporting process followed by vehicles to report a road condition is mentioned. Following that, we discuss the inefficiency of a basic reporting approach using chosen communication mechanism, and the need for an alternative reporting. This need is essentially the optimization problem.

4.1 Scenario Representation

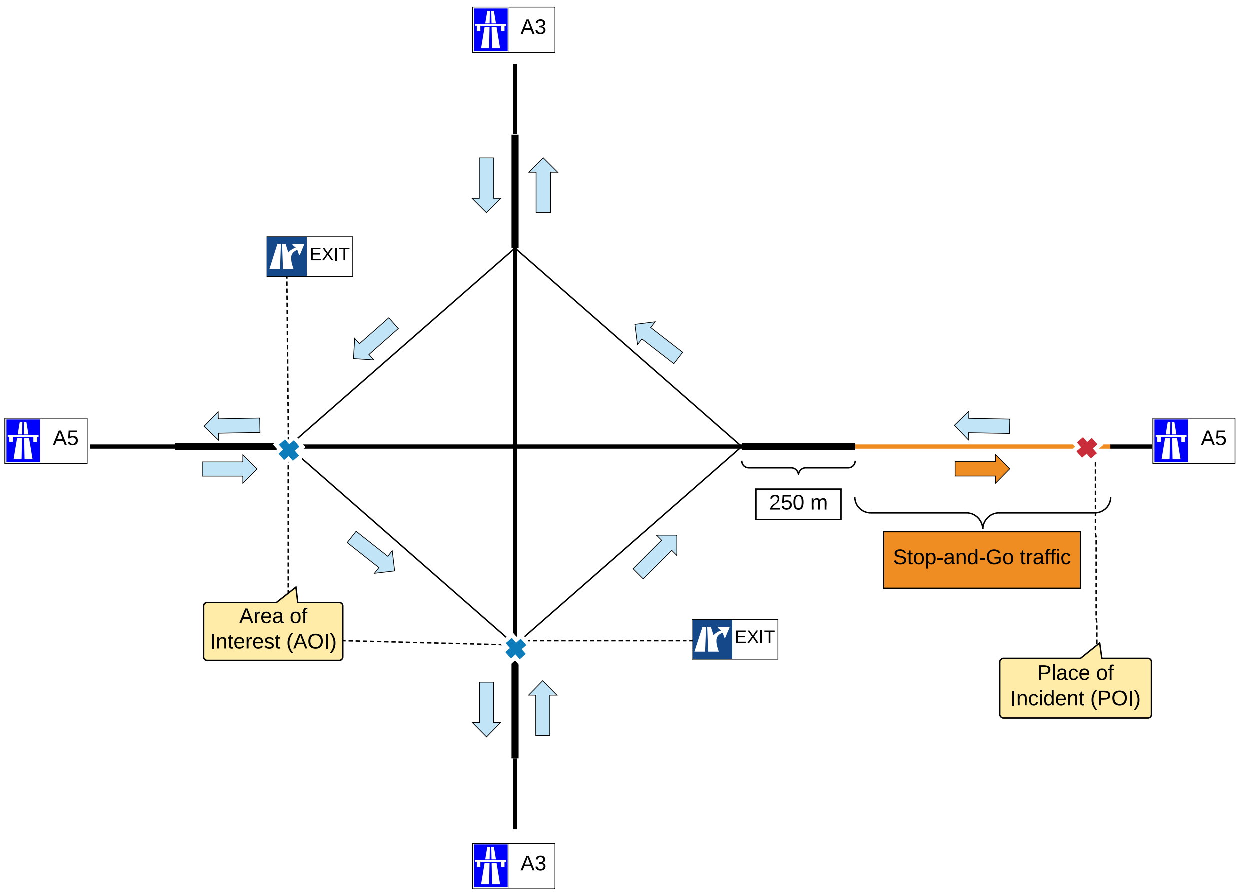


Figure 4.1: Overview of an Expressway-interchange: the scenario.

The road scenario is visualized in Figure 4.1 and is a representation of a typical Expressway-interchange that is commonly found on Expressways. We assume a simple diamond-interchange of two Expressways namely **A3** and **A5** as it is unambiguous and straightforward. As seen in figure, along each direction of travel, just before the interchange, there is an Exit ramp. Each Exit connects to the other Expressway which is orthogonal via an Entrance ramp that is located after the interchange. At the Exit ramps, vehicles wishing to reroute to other Expressway utilize these Exits appropriately, and those willing to continue on the same Expressway simply continue without rerouting. The opportunity to reroute at Exit ramps plays an important role for decision-making in context of traffic management and is discussed in future sections.

We assume an accident at a point after the Entrance ramp along the Expressway A5 well beyond the opportunity to interchange and reroute. This is marked by symbol ✖ in the Figure 4.1. We call the place where accident has occurred as *Place of Incident (POI)*. Due to the accident, which has now blocked one of the traffic lanes, though, traffic has not come to a complete standstill, it is moving at a reduced pace than normal. The stretch of slow traffic is seen lasting from POI to a couple of hundreds of meters behind it, as commonly experienced in a real-world scenario. This length of slow traffic is highlighted in Orange colour in the representation. The vehicles in this coloured region have no option for rerouting other than waiting for the traffic flow to resolve to normalcy. Hence, this is the reason for considering an accident at a point only after the possibility to reroute. This kind of slow traffic movement characterizes to *stop-and-go* traffic frequently experienced along Expressways near interchanges especially during rush hours.

A stop-and-go traffic is characterized by short period of movements with regular interruptions leading to frequent stopping and restarting. Although, this is not a complete halt of traffic, yet the traffic movement is not efficient and optimized, and hence is considered as abnormal. Any of the following, including but not limited to accident, construction work, maintenance work or just too much traffic can be the causes for stop-and-go traffic, and generally the time for resolving it to normalcy is indeterministic.

The possible Exits for the oncoming traffic towards POI is represented as *Area of Interest (AOI)*, i.e., First, along A3, by taking an Exit from A3 towards A5, or second, for vehicles in A5 coming from other side, continuing to travel along it. In this scenario, we have two such AOI, one each near the Exit along A3 and A5 represented by symbol ✖ in the Figure 4.1.

Under the mentioned circumstances, the significance of these AOI is that, if the vehicles approaching AOI are notified timely about the prevailing stop-and-go traffic ahead due to an accident, then they can consider taking available Exit appropriately. That is, oncoming traffic from A3 willing to reroute to A5 towards POI can avoid taking the Exit ramp and instead continue along A3. On the contrary, oncoming traffic from A5, instead of continuing on the same Expressway can avoid getting stuck at POI by rerouting to A3 using Exit ramp. With this change in travelling, an alternative route to the destination can now be contemplated instead of getting stuck in a jam, and hence will be an efficient traffic management.

4.2 Objective

Thereby, the motive of the communication system is the real-time dissemination of road status and events for the sake of traffic efficiency to the oncoming traffic well before reaching the event locations. Furthermore, this dissemination system should enforce information freshness as vehicles rely on them for making travelling decisions.

To this end, the communication system should also communicate dissolving of an existing traffic jam so that the new road status can be considered and decision can be made appropriately. Additionally, the entire process of data dissemination should consider in terms of network overload and should be scalable to match varying vehicular densities.

4.3 Choosing Appropriate Communication Mechanism

In Chapter 2, different communication mechanisms for vehicular environments were discussed. Choosing the right communication mechanism is critical to meet our objective mentioned in Section 4.2. Three such properties are significant in selecting the required communication mechanism. First, the communication should happen from POI to AOI which are distantly located from each other (more than couple of hundreds of meters as generally observed in real-world). Second, the road events should be updated to the vehicles in AOI in real-time without delay so as to maintain information freshness. Furthermore, it should also consider communicating any change of status of those events. Third, the communication system should be scalable. In the following we discuss and evaluate different choices available for their appropriateness bearing these design constraints.

Bearing the design constraints, since our scenario is characterized by strict latency constraints to maintain information freshness, we consider a push-based approach for data dissemination instead of pull-based scheme. Furthermore, with pull-based approach vehicles which are the clients have to frequently poll for updates. Generally, update intervals are non deterministic in vehicular environments due to high mobility, and therefore this approach will be inefficient. Additionally, any change in status of events that are already disseminated cannot be easily updated in pull-based scheme. It involves self-initiated frequent polling by clients, and thus leading to communication overhead and scalability problems. On the contrary, updates can be efficiently managed with push-based approach as they maintain state information of the clients and can be initiated only when there is an actual update concerning to a client exists.

Ad-hoc Communication: A one-hop ad-hoc communication with beacon messages using 802.11p is typically used within a vehicle's immediate surroundings for safety and warning messages. For this purpose, a vehicle sends beacons up to 10 times per second causing huge load on the communication channel as shown in the work of [SRA10]. Also, since the same control channel is used for sending beacon messages by all vehicles, [LYAW12] showed the challenge of congestion in control channel during high density of vehicles that affects the performance significantly. Moreover, considering the short communication range of 802.11p it would be inefficient for long distance data dissemination as required in our case. A multi-hop DTN is another option. Although, long-distance data dissemination is not unrealistic in DTN but it is not an efficient way for our scenario. This is because, DTN rely on availability of intermediate vehicles acting as data mules for data delivery between source and destination. This dependency on other vehicles leads to reliability that is not guaranteed. For selecting a vehicle as a data mule, global knowledge of paths taken by each vehicle should be known well before. If a predictable path is unavailable, then DTN simply follows flooding mechanism to increase data delivery which is inefficient. Also, since redundant copies can exist within network to increase data delivery; the buffer management in intermediate vehicles is of concern. Although, redundant message removal mechanism in DTN using acknowledgement sent by a destination vehicle is proposed in [dNGCdLCV17], it is still an overhead to remove those copies from all intermediate vehicles across the entire network. Furthermore, if an update is issued to an already disseminated notification, tracking and updating it is not realized in this mechanism. Due to the discussed inefficiencies associated with DTN, we found it not suitable for our scenario.

Cellular Communication: Generally, ad-hoc communication with single-hop or multi-hop is not reliable as they are dependent on other vehicles for communication. In places of sparsely populated areas similar to rural roads or empty roads especially during night time, data dissemination in real-time using ad-hoc is not possible. On the contrary, although ad-hoc might be suitable for densely populated areas such as highways and urban roads, but they scale poorly and are limited to their intrinsic short communication ranges.

We found it suitable and realistic to make use of cellular communication for our scenario due to large communication range compared to ad-hoc communication. Moreover, given the acceptable delay in cellular communication with 3G/LTE, realising fast data dissemination is simplistic. In real-world, modern cars are already equipped with interfaces to communicate with these technologies and hence cellular based data dissemination was practical for our scenario. Choosing a push-based dissemination using cellular communication made sense. Dissemination using flooding method was also evaluated. However, considering the issues of redundancy, contention and the associated transmission costs referred together as *broadcast storm* problem mentioned in [NTCS99], we ignored it.

In light of above discussions, appropriateness of Publish/Subscribe approach was evaluated for our scenario where long distance communication while maintaining information freshness was required. As discussed in Section 2.2, Publish/Subscribe is an asynchronous and anonymous communication where publishers and subscribers are unknown to each other. It enables communication in an application-independent way abstracting underlying communication system. Since a broker-system manages the task of matching publications with subscriptions, updates are managed without overhead. Therefore reliability is ensured without need of any form of buffer management as it is in DTN. Since, with Publish/Subscribe, locally detected events can be disseminated by vehicles in real-time even without the knowledge of those interested vehicles, it clearly matched our design requirements.

Although *Channel, Content* and *Topic* based Publish/Subscribe methods exits, content based Publish/-Subscribe mechanism is selected as it is efficient to represent content using discrete attributes. It is required because a traffic jam can be due to an accident, construction work, maintenance work, or simply due to high vehicular density experienced during rush hours. Using the discrete attributes, the reason for a jam can be easily representable. Additionally, many traffic jams can be reported on multiple such Expressway-interchanges along the journey and these should be uniquely identified to make correct rerouting decisions only at the appropriate interchange. Moreover, although, for our study we have considered a traffic management scenario, it can be later applicable for warning and environmental updates such as weather, slippery road due to snow, hydroplaning, or other related notifications. All these cases can be represented discretely using content based Publish/Subscribe as it enables containment and aggregation capabilities. A broker-system relies on these capabilities that offers flexibility for matching publications to subscribers.

4.4 Premise and Reporting Process

In this section, initially, we describe the premise assumed with respect to vehicles for their functioning with the chosen Publish/Subscribe method via cellular communication. Later, the reporting process to alert a road condition with the chosen mechanism is explained in parallel with our scenario.

4.4.1 Premise

Nowadays, modern vehicles are not inadequate of storage space. Therefore we assume vehicles to have sufficient storage space, modelled as *cache* in our design for storing all publications. To this end, we model cache as an array, and its elements are arranged based on published time associated with received notifications. The necessity for this storage is that during a vehicle's journey from start till reaching its destination, multiple traffic jams can be reported at many roads along its route. With storage of publications, it enables a vehicle to make rerouting decision to avoid traffic-jam based on its cached contents that are relevant for that particular road.

For every road, by default, we assume every vehicle to have a default cache value with an entry equals `False`. An entry with `False` signifies a `no-traffic-jam` condition. This assumption is logically correct, because if otherwise, a vehicle would assume `traffic-jam` always for a road. With respect to our scenario, once the vehicle has travelled past a road, corresponding cached values can be removed as they are no more required nor relevant any more.

Broadly, a publication can contain the publisher identifier, the published time, the publishing location (location from where the publication was issued), the value of the publication and the associated TTL, if any. Subsequently, we adapt this in our study for better representation. In our scenario, a publisher would be aware of its own identifier, time, location from GPS and the detected road status value from its sensors. Henceforth, all publications in this work would contain these attributes.

In real-world scenario maintenance- or construction work on roads are generally scheduled and hence are known before. Thereby, time until which the road traffic is affected can be determined. We represent this case in our scenario in the form of TTL associated with events. Once the TTL of a notification expires, then this notification is no more valid and will be removed from the cache. As our scenario is stop-and-go traffic, we assume TTL associated with all publications to be the same. Since the cache contents are arranged according to their corresponding published time, this means, if an entry is removed due to expired TTL, then all the entries before that should also be removed. Furthermore, it ensures decisions made will be based only on notifications that are valid in the cache(i.e., with valid TTL).

As we use cellular network for communication, we assume a central server as broker. In practice, this server can be a dedicated back-end server or can be a hierarchically organized cellular network. Furthermore, discussion on design and assumptions of a vehicle's communication capability, road segment visualisation, syntax of a subscription and publication, mapping of publications from a particular road segment to the concerned AOI for decision-making, and other details are explained in Chapter 6.

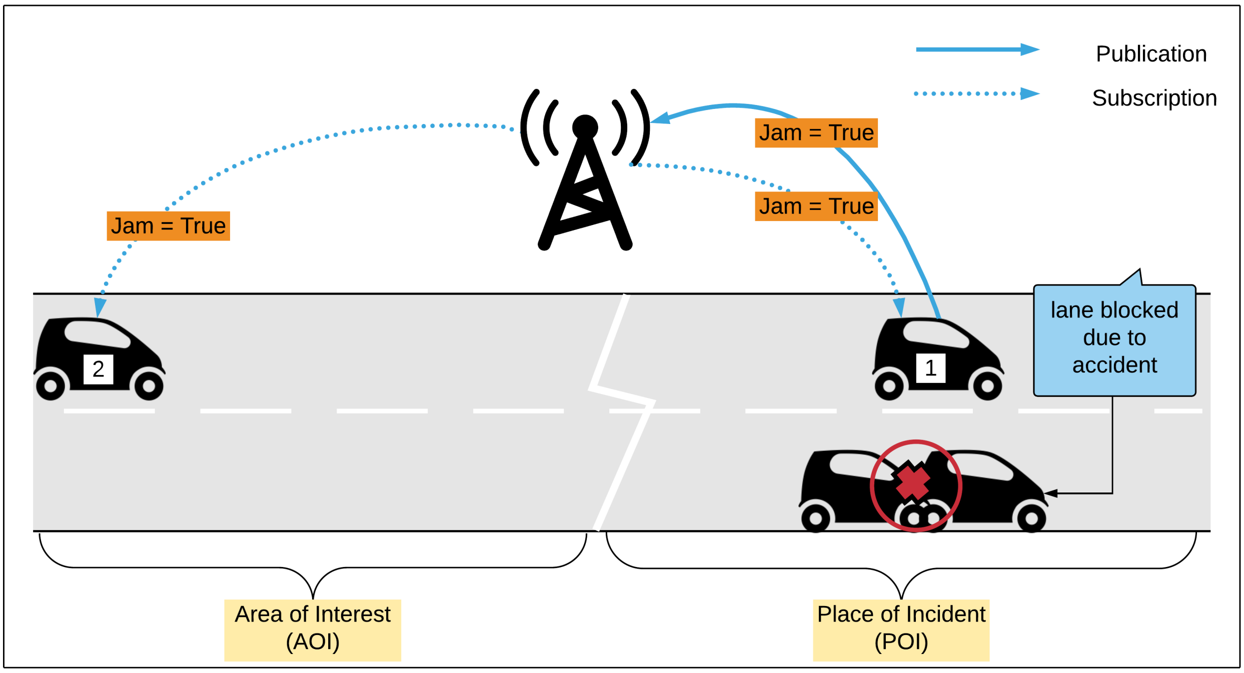


Figure 4.2: Event notification using Publish/Subscribe mechanism over cellular communication.

4.4.2 Reporting process

The reporting of a road status or event is as following. Any vehicle upon reaching the POI senses the accident via it's on-board sensors and notifies others of the prevailing jam by sending notifications in the form of a publication. The broker in the Publish/Subscribe system then forwards these notifications in the form of subscriptions to other subscribed vehicles including the published vehicle, which is also a subscriber. Since the publishing vehicle does the task of reporting, it is assumed that other subscribed vehicles are behind the publishing vehicle with respect to location of POI. As stop-and-go traffic is assumed in our scenario, a vehicle publishes only once from a POI and multiple publications after some time from the same vehicle for the same POI is not considered.

Although, the stretch of slow traffic seen is lasting from POI to a couple of hundreds of meters behind, a vehicle publishes only when it reaches POI. The reason for this is explained as follows. Assume, a stop-and-go traffic was seen for hundred of meters, and now it starts to dissolve starting from the head of the jam as it should be. Due to this, all those vehicles in the head of the jam would now report `no-traffic-jam`, whereas vehicles still at the tail of the same jam from the same location are still experiencing `traffic-jam`. To avoid this inconsistent behaviour, we allow publisher to publish only once. As our scenario is not a complete halt of traffic, any vehicle in this stretch will eventually have to reach POI. And anyway, as dissolving of this traffic jam should happen starting first at POI with clearance of the accident, the assumption made is appropriate.

The reporting process is depicted in Figure 4.2. In the figure, an accident has occurred allowing only one lane for traffic movement. Here, Vehicle-1 and Vehicle-2 are both subscribers. As Vehicle-1 arrives first at POI, it is the publishing vehicle (Publisher). Vehicle-1 sends publication with $Jam = True$ upon detecting accident. Vehicle-2, which is yet to arrive at POI is notified about this jam well before its arrival via cellular entity. The possibility of more than one publishing vehicle reporting at the same time is also considered in our study. This can happen when many vehicles reach POI at same the time and send publications individually. Since the publications are sent by different vehicles, it is not a duplicate and regarded to be valid.

4.5 Optimization Problem: Default-approach

A basic and intuitive approach of reporting is that every vehicle reaching the POI sends publication reporting the event. The issue with this approach is that all the publications reported are having the same road status information making them redundant. All notifications published are from the same POI having same information except for the publisher and the published time which is based on a publishing vehicle's arrival at the POI.

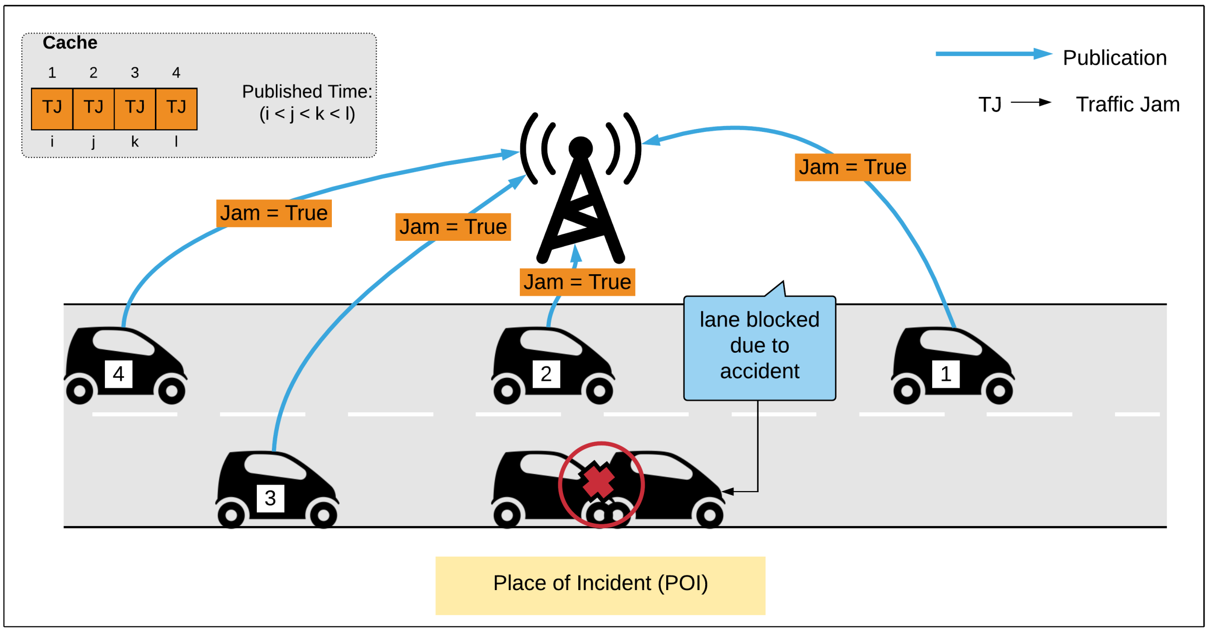


Figure 4.3: Event notification by all vehicles: *Default-approach*.

For example, as depicted in Figure 4.3, an accident has caused leading to only one of the two lanes operational. As Vehicle-1 is the first publisher arriving, say at $time = t_i$, it senses the traffic jam due to an accident and sends publication to alert oncoming traffic. Similarly, the next approaching publisher, Vehicle-2, does the same process to alert oncoming traffic, except publishing at a different time, $time = t_j$, $j > i$. The same procedure is followed by vehicles numbered 3 and 4. In this scenario, we have five different publications reporting same information concerning the same event. Subsequently, as each publication is sent to every subscriber (that includes publisher too), now we have 16 notifications forwarded by the broker (4 notifications each for each of the 4 vehicles who are subscribers). Consequently, now the cache in each vehicle contains all the four publications from four individual vehicles with the same *Traffic-Jam* value. In total, the broker is involved in the burden of serving 20 notifications (4 publications + 16 subscriptions).

This redundant way of reporting the same information by every vehicle incurs unwanted load on cellular infrastructure. We call this approach of reporting as a *Default-approach*. As number of notifications forwarded by broker is proportional to number of publications and number of subscribers at a given point in time, this approach is not scalable. The overhead caused by this approach varies with the density of vehicles. In a dense network such as Expressway or Urban scenario especially during rush hours this approach scales badly.

Not to forget, a cellular infrastructure also bears the load of serving other mobile and internet applications. Furthermore, the high costs involved in setting-up and operating a cellular infrastructure is quite significant. Relating to the issues of scalability, redundant publications and subsequently the number of notifications forwarded by broker, we see a need for alternative efficient approach for reporting. As an alternative, we propose *Cooperative-approach* of reporting in Chapter 5 that makes efficient use of caching available at every vehicle to keep the network load at moderate levels.

5 Adaptive Cache Management in Distributed Networks

This chapter describes our proposal to optimize the problem described in previous chapter. First, we propose and explain the working of *Cooperative-approach* of reporting considered by publishers to avoid redundant publications. Second, we highlight the inconsistency in decision-making associated with *Cooperative-approach*. Third, to overcome this inconsistency, we propose *Adaptive Cache Management* at every vehicle using vehicular rate. Later, we discuss how constantly changing vehicular rate as seen in real-world is considered and adapted in our approach. Finally, we contemplate how the combination of *Cooperative-approach* of reporting and *Adaptive Cache Management* adapts to constantly changing vehicular environment.

5.1 Avoiding Redundant Publications

In this section, we describe the working of our proposal: *Cooperative-approach* to skip and avoid redundant publications. Initially, we explain the decision-flow algorithm that every vehicle at POI follows before publishing. Based on this a vehicle decides either to publish or not publish. Following that, we apply this approach of reporting to our scenario and compare it with the *Default-approach*. Later, we analyse and explain the reliability issue for decision-making associated with *Cooperative-approach*.

5.1.1 Cooperative-approach

To avoid redundant publications having same information by all vehicles, we propose *Cooperative-approach*. In this approach, while reporting, we exploit the spatial and temporal correlation of an event. As discussed in previous chapter, notifications received due to subscriptions are stored in cache for the purpose of decision-making. We utilize this information efficiently to avoid redundant publications.

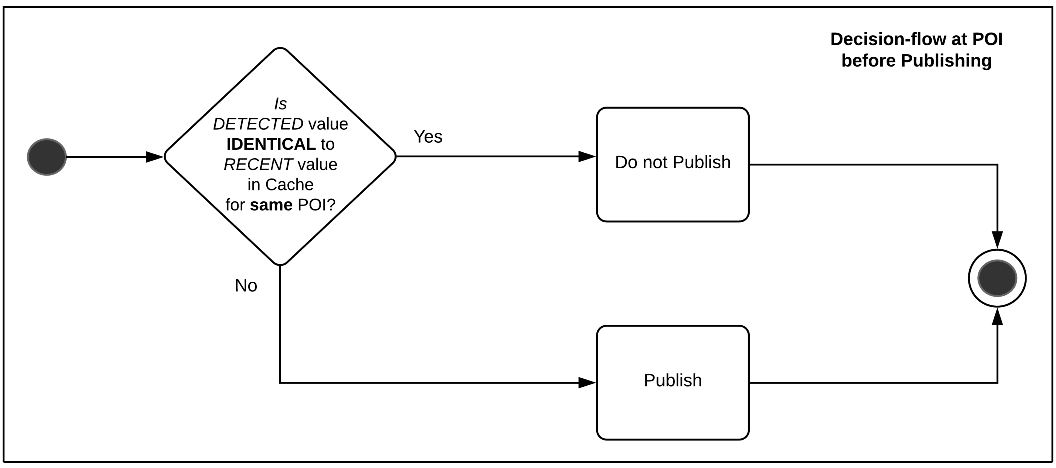


Figure 5.1: Decision-flow for publishing: *Cooperative-approach*.

The *Cooperative-approach* modelled as a decision-flow algorithm is seen in Figure 5.1. Any vehicle before reporting an event at a particular POI with a publication, first, checks the value of recent entry in its cache for the same POI. If that value is same as the value detected by the vehicle's on-board sensors, then that vehicle refrains from publishing. On the contrary, if the detected value is different from the recent value in cache, then the detected value is simply published as a publication.

We call this way of reporting by vehicles at POI as *Cooperative-approach*. This is because, every vehicle before publishing verifies cooperatively with its latest cache entry to get the recent publication that was published. As mentioned in Subsection 4.4.1, since cache is modelled as an array and arranged based on published time of publications, last entry in cache reflects the most recent publication.

The rationale behind comparing detected value by a vehicle to the recent entry in its cache is that the recent entry in cache is the reflection of the recent publication. Subsequently, the recent publication is the most recent road status detected and published by a vehicle that was recently at POI.

We also thought of comparing detected value with majority of values stored in cache. This would have resulted in inconsistency because of the following reasons. First, a vehicle that has recently joined the system can have fewer entries in cache compared to a vehicle that has been there on a road since a long time. Furthermore, any comparison under these condition using majority of values in cache will lead to different behaviour for both these vehicles. Additionally, there is no guarantee that a majority of values in cache would always reflect the recent road status since it would be considering just the majority of entries. Due to this erratic and unpredictable nature, this option was discarded.

Whereas comparing with latest cache entry, it guarantees to reflect the recent road status. This is also true even if TTL is associated with notifications. As TTL associated with a notification will be same across all caches wherever it is stored, and upon expiry it will be removed from all caches at the same time, the behaviour is consistent and predictable. Due to these reasons, we chose comparing with latest entry in cache that is consistent and provides latest road status adhering to the information freshness required for decision-making.

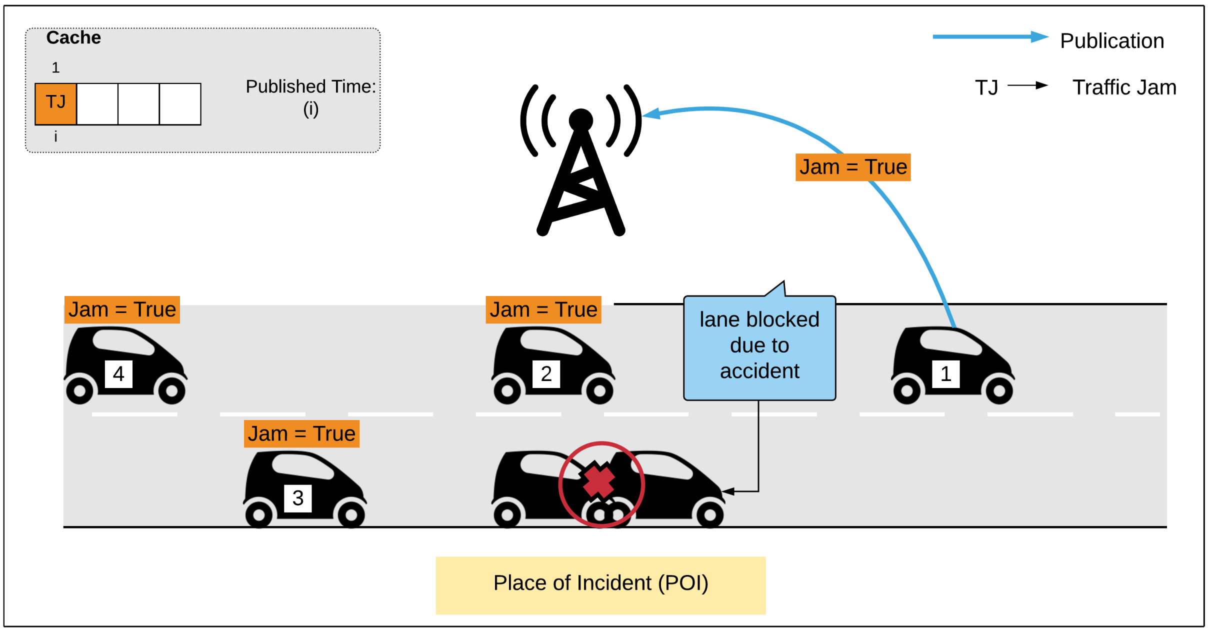


Figure 5.2: Event notification based on: *Cooperative-approach*.

We apply the *Cooperative-approach* method of reporting for our scenario. As depicted in Figure 5.2, Vehicle-1 is the first vehicle to reach POI. When it arrives at POI, there is only the default cache entry with a value `False` signifying `no-traffic-jam`. Upon detecting a `traffic-jam` through its sensors and as there is no entry in cache regarding this detected jam, it publishes a jam-notification to alert other vehicles. Subsequently, vehicles numbered 2, 3, and 4 who are the subscribers and are already on the road edge receive this alert to update their cache with this information. Although, now these vehicles also detect the same jam information, however they all restrain from individually publishing it after following the decision-flow algorithm.

Now, we have only one publication reporting road status, although four vehicles detect the same. In comparison to the *Default-approach*, instead of serving 20 notifications by broker, in *Cooperative-approach*, in total we have just 5 notifications (1 publication + 4 subscriptions). And subsequently, now the cache in each vehicle contains only one entry with a *Traffic-Jam* value matching the single publication sent by that single publisher (Vehicle 1). And this single entry in cache still reflects the road status as expected.

5.1.2 Reliability Issue with Cooperative-approach

Though, *Cooperative-approach* overcomes the challenge of network overload compared to *Default-approach*, it exhibits reliability issues. Since, only one vehicle at POI is actually publishing, now there is no quantification of number of those vehicles which have detected the same road status and have refrained from publishing. The significance of these vehicles is that it gives a count of all those vehicles which have implicitly consented to the recent published value as to be true. This count plays an important role for vehicles at AOI as they rely on their cache containing all publications for decision-making.

In Figure 4.3, due to publications by all vehicles, the cache at subscribed vehicles includes all the publications. And therefore, inherently, each individual entry in cache represents a vehicle seen at POI. On the contrary, due to *Cooperative-approach*, in Figure 5.2, the cache only has one entry from the single publication, although four vehicles are seen at POI.

As significant numbers of entries are missing in cache due to *Cooperative-approach*, trusting just one publication for decision-making causes reliability issues. Whereas, in *Default-approach*, as every vehicle at POI was publishing, the count of all vehicles at POI would be reflected in the number of cache entries and reliability was not an issue.

5.2 Addressing Reliability via Adaptive Cache Management

The significance of addressing reliability issue associated with *Cooperative-approach* is critical as it affects greatly the decision-making process. To this end, without actual publication, we had to account the missing entries in cache for those vehicles which refrained from publishing. In the following section we discuss the design of adaptive caching to overcome this issue.

5.2.1 Ensuring Reliability with Virtual Cache Entries

In order to account in cache those vehicles that were prevented from publishing to avoid redundant publications, we propose adding virtual cache entries. This idea is based on *Redundant Aggregation*, commonly used in information fusion for wireless sensor networks to avoid network costs. Accordingly, as mentioned in the work of [MLN$^+$17], instead of transmitting similar information, aggregation of those similar information can be done to avoid network costs while maintaining accuracy and reliability. To this end, each of the aggregated information have same values for attributes as that of the latest information that was transmitted .

In our work, the virtual cache entries are added based on vehicular rate. *Vehicular Rate* is the number of vehicles moving through a point in some duration, expressed in time units. As discussed in Subsection 4.4.1, a publication contains the publisher identifier with the published time, publishing location, published value and associated TTL, if any. Apart from these basic publication attributes, we propose to also include the observed vehicular rate by the publisher during the time of publishing.

In VANETs with 802.11p and beacon messages, a vehicle can opportunistically measure the number of vehicles around it. Based on the number of beacon messages that a vehicle processes from other vehicles around it, a vehicular rate per time can be obtained. With this rate now known to publisher, it is also included in the publication as an attribute to support *Adaptive Cache Management* implementing virtual entries.

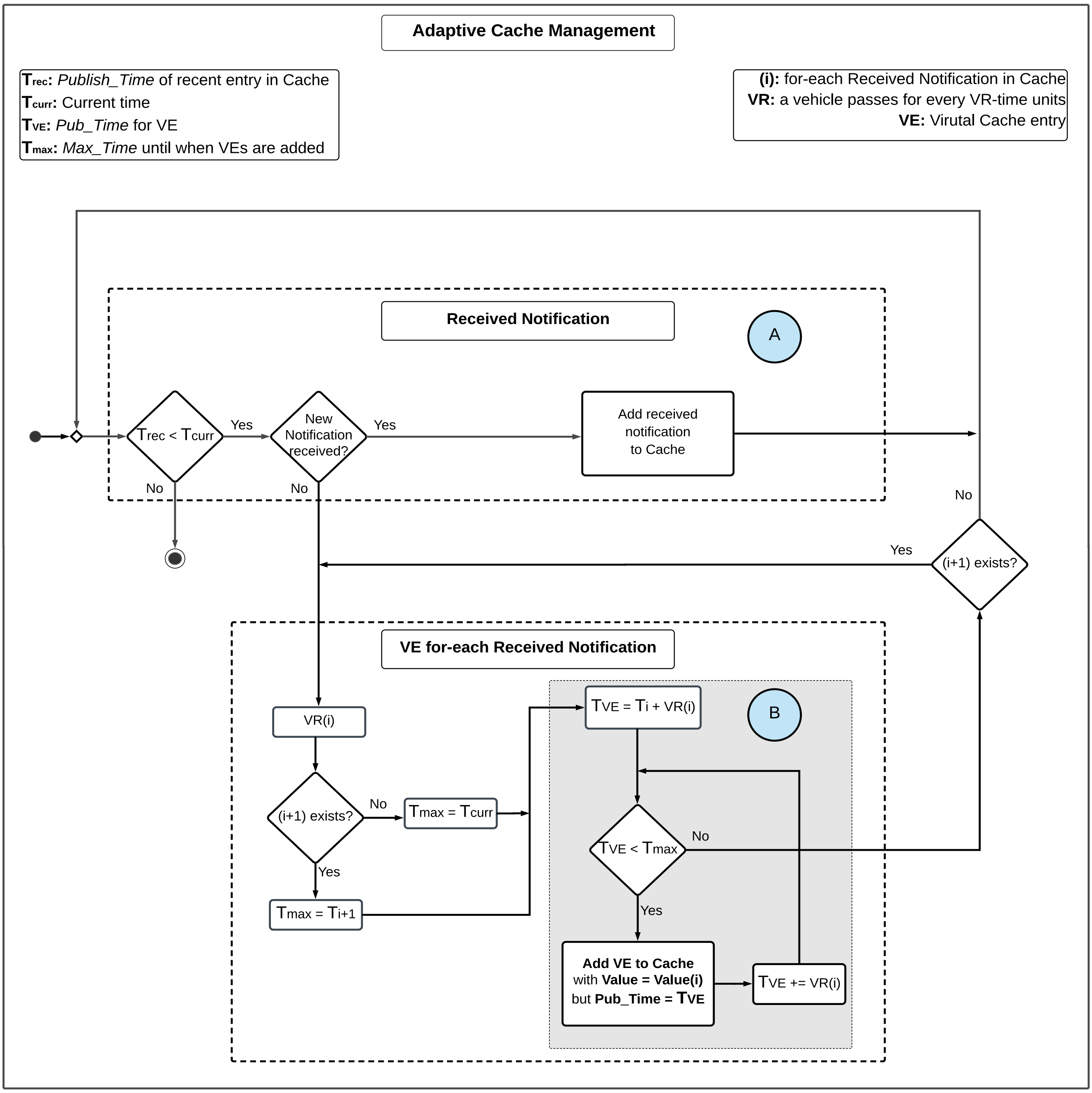


Figure 5.3: *Adaptive Cache Management* using virtual cache entries.

Adaptive Cache Management: The process of adding virtual entry is as shown in Figure 5.3. If a notification is explicitly received via cellular communication, it is simply added to the cache as shown in **Block-A** in figure. Therefore, without any virtual entries added, the cache would simply contain all the notifications received in the form of subscription.

The flow for adding virtual entry starts provided at least one entry reflecting the original notification exists in cache. In other words, virtual entries are added only if the vehicle is subscribed and has received atleast one notification that is still valid in cache. The reason for this assumption is because the required vehicular rate is sent only with original notification, and to add virtual entries, this rate can only be obtained provided the original notification still exists in cache. As any expired cache entries are removed automatically, it is assured that no virtual entries with expired TTL are added.

Virtual entries are calculated for each original notification that are valid in cache. Hence, during the iteration through each of the received notifications that are valid, virtual entries would be added with a published time starting immediately after the published time of the current original notification (marked as $\boldsymbol{T_i}$).

Based on vehicular rate obtained at ***(i)***, marked as [***VR(i)***] (i.e., vehicular rate observed when this original notification was published by publisher), the published time for each of the virtual entry added can be uniformly distributed. The published time of a virtual entry (marked as $\boldsymbol{T_{VE}}$) is a summation of the published time of current original notification and the vehicular rate obtained at ***(i)***.
It is given as: $\boldsymbol{T_{VE} = T_i + VR(i)}$.
Subsequently, for the next virtual entry to be added, its published time would be incremented by vehicular rate. That is: $\boldsymbol{T_{VE} = T_{VE} + [VR(i)]}$. This way all the virtual entries added for a notification would have different published times distributed uniformly based on vehicular rate.

Furthermore, these virtual entries are replicated with the attributes and their corresponding values from current original notification ($\boldsymbol{T_i}$) as followed in *Redundant Aggregation*. Hence they all have the same published location and published value as that of the original notification except for different published time. To this end, virtual entries for current notification are added until a time (marked as $\boldsymbol{T_{max}}$), which is either the published time of the *next-received-notification* entry in cache (marked as $\boldsymbol{T_{i+1}}$) or until current-time (marked as $\boldsymbol{T_{curr}}$). In the former case, the loop for adding virtual entries will continue with the same procedure with the following *next-received-notification* entry, whereas in later case, it will terminate. This process is shown in **Block-B**.

Block-B signifies the number of virtual entries added for each original notification which is dependent on $\boldsymbol{T_{max}}$ and vehicular rate. *Current-time* signifies the present system time at which a vehicle at AOI wants to make rerouting decision based on its cached contents. Hence, any virtual entries, if added, should be only until this time.

5.2.2 Challenges of Varying Vehicular Rate

In real-world, using 802.11p and beacon messages in VANETs, vehicles can measure vehicular rate around its surrounding. But the communication range of these technologies is limited to couple of hundred of meters. Due to this, rate observed by a publishing vehicle accounts only for vehicles coming behind it that were detected by its limited communication range during its publishing time.

As virtual entries are being added based on this rate until next actual publication (which can be at times only after long time), change of vehicular rate until next publication, if witnessed any, is neglected. Consideration this practically probable scenario is again crucial for ensuring reliability during rerouting decisions.

In Figure 5.2, only Vehicle-1 publishes the road status with a Traffic-Jam value. Lets assume that it also sends observed vehicular rate accounting other three vehicles coming behind it in the publication. Subsequently, using *Adaptive Cache Management*, this rate is considered for adding virtual entries until next publication. However, now suppose, this vehicular rate decreases significantly almost to zero, and for the worst case the next publication is only after long time, then accordingly, this approach is still adding more virtual entries than it is supposed to be. Ideally, in this case, the number of virtual entries added should have been lessened. However, since outdated rate is still considered, though no vehicles are passing through POI, vehicles at AOI, accounted falsely by adding more virtual entries. Given the possibility of varying vehicular rate between successive publications and subsequently to incorporate it, in the next section we discuss how we measure this rate more accurately.

5.3 Dealing with Varying Vehicular Rate and Changes in Road Conditions

This section explains adaptation of our method for varying vehicular rate. First, we describe how the rate is continuously measured to be used only during publishing. Following that we describe how a change in road status is rightly reported by our method as expected.

5.3.1 Estimation of Vehicle Passing Rate

Practically, in real-world, varying vehicular rate can be estimated and achieved. If all vehicles travelling on road are subscribed by default, then a cellular base station which can also be the broker, can provide this information. As broker forwards notifications to subscribers, a stateful broker can assist providing number of vehicles that have travelled between the time of two publications. However, this approach will cause unwanted load on the broker. A publisher will have to first get the observed rate from the broker and then encapsulate this rate with its publication. This publication is sent again to broker via cellular communication which causes overload and scalability issues. As an alternative, the role of RSUs is substantial.

Recent research studies in ITS emphasize the benefits of using RSUs. With collaborative sensing from V2V and V2I, an almost accurate estimation of vehicular density can be achieved as reported in the work of [SBF+15]. The benefit of interconnecting a group of deployed RSUs is analysed in the study [RSNT14]. This interconnection helps to provide low-latent, non-safety services which were thought to be unrealistic before.

Although, in our study, to restrict the scope, we do not consider existence of a RSU to measure vehicular rate, instead, we assume a global knowledge consensus. We assume an independent stateful entity with global knowledge that keeps track of all vehicles which have passed through POI since the beginning. This entity maintains and provides the number of vehicles that have travelled over duration of time irrespective of road status and vehicular rate. Once a publisher decides to publish, the number of vehicles that have passed through in the duration between previous publication and current-time can be obtained. With this information, vehicular rate:- a vehicle passes for every x-time units can be measured. This value is sent along with the publication as discussed before.

5.3.2 Adapting to Varying Vehicular Rate

The process of adding virtual entries accounting varying vehicular rate is as shown in Figure 5.4. This is extended and adapted from the Figure 5.3. **Block-A** represents the flow for adding all the received notifications to cache. **Block-B** represents the time until which virtual entries are added. Both these blocks remain unchanged except for the decision from where the vehicular rate for-each loop iteration is considered.

With global knowledge, the exactly observed vehicular rate including variations since the time of previous publication is now available. This is sent with latest publication. Hence, for adding virtual entries for an original notification, the rate is obtained from its immediate *next-received-notification*. This obtained rate gives the exact number of those unpublished vehicles that have passed through POI between the published time of current-(i)th publication and next-(i+1)th publication. If there is only one original notification in cache or when the loop iteration has reached end-to the recently received notification, then there would be no (i+1)th publication to obtain vehicular rate from. In this case, we use the rate observed at current publication for adding virtual entries, as that gives the most recently observed vehicular rate.

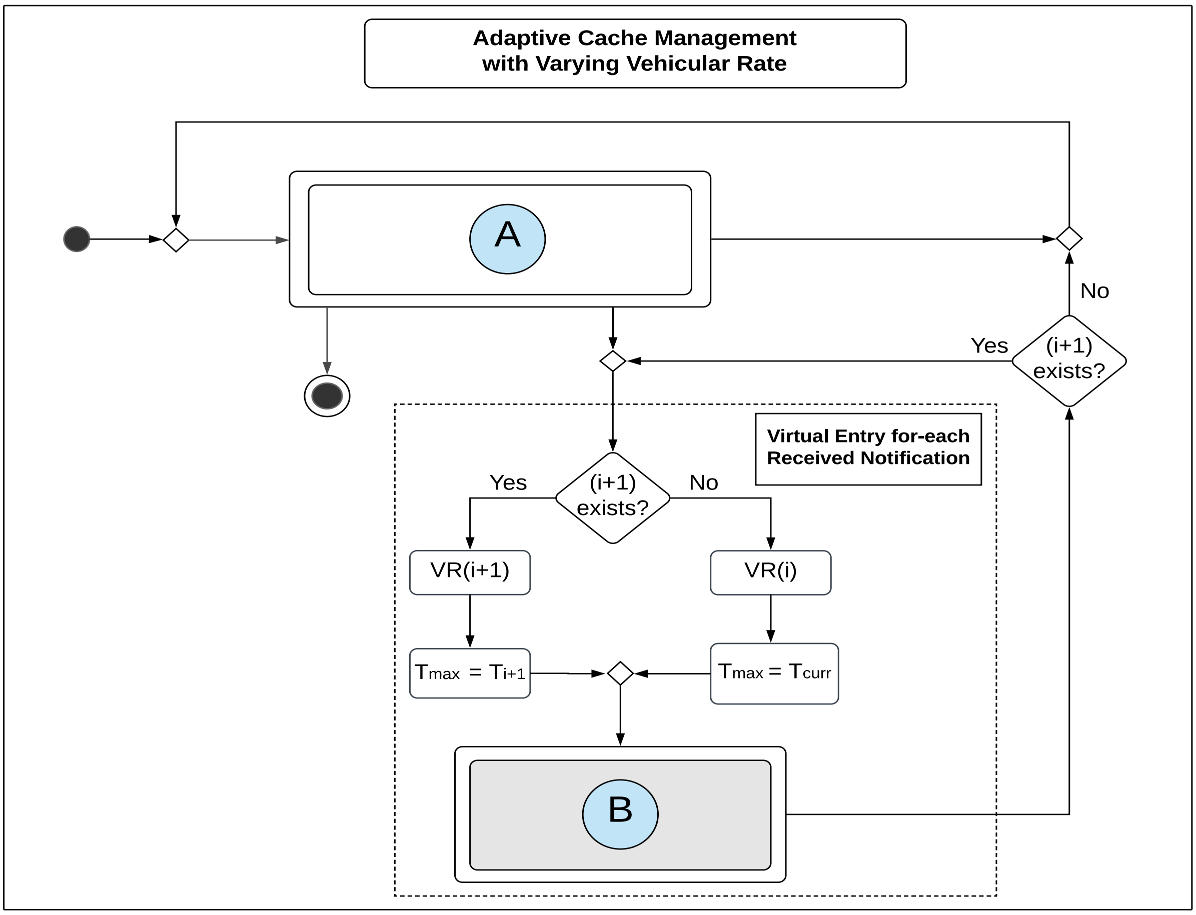


Figure 5.4: Contemplating varying vehicular rate using *Adaptive Cache Management*.

5.3.3 Adapting to Changes in Road Status

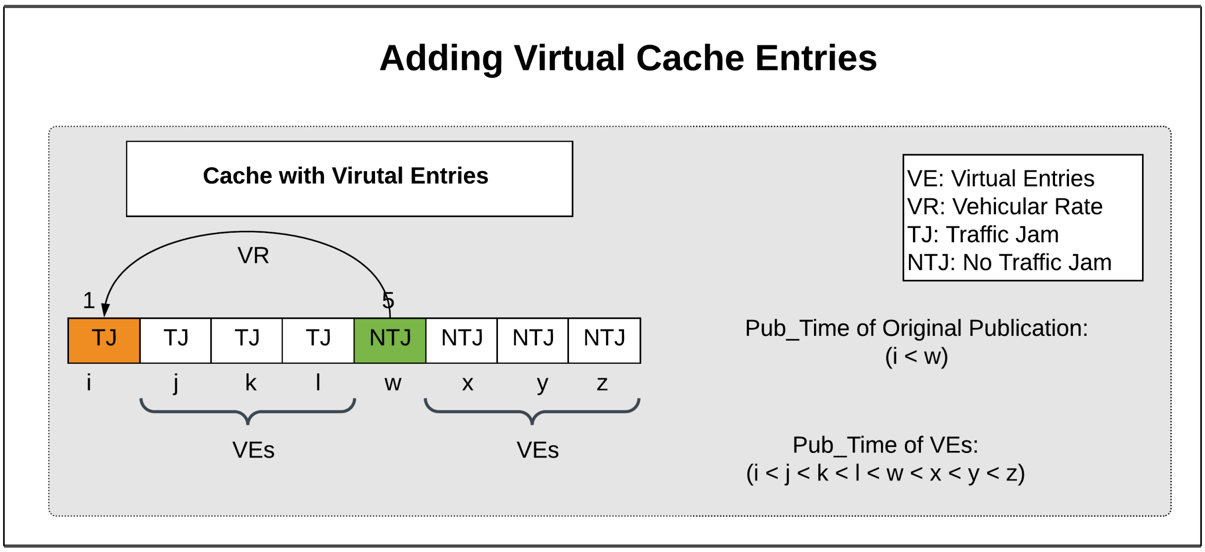


Figure 5.5: Cache with virtual entries.

A change in road status occurs when a traffic-jam starts or an existing jam dissolves. Both these changes should be immediately notified to maintain information freshness. In our scenario, although vehicles refrain from publishing when detected value is same as recent publication, they immediately publish if it is different otherwise.

If an existing traffic-jam dissolves, the first vehicle reaching after this change of road state, would detect this new road status and publish with a value `NTJ: no-traffic-jam`. On the contrary, when a normal traffic movement experiences traffic, first vehicle that arrives since this new state, would publish with a value `TJ: traffic-jam`.

Figure 5.5 gives an overview of adding virtual entries to cache. With respect to Figure 5.2, only Vehicle-1 reports about `TJ: traffic-jam` at published time = (i). Successive vehicles numbered 2,3 and 4 refrain from publishing due to *Cooperative-approach*. Assume after some minutes, the road condition changes to `NTJ: no-traffic-jam`. Then according to *Cooperative-approach*, the first vehicle arriving after this change event must publish the new road status. Accordingly, Vehicle-5 that arrives after the change event, detects a `NTJ: no-traffic-jam` which is contradicting from its recent cache value that which is still representing `TJ: traffic-jam` and therefore publishes at time = (w). While publishing this new condition, Vehicle-5 also sends the vehicular rate observed until its publication to support adding virtual entries. Based on this rate, vehicles numbered 2, 3 and 4 are accounted in cache in the form of virtual entries. However, it is significant to note these virtual entries all have different published times that lie between the published times of two original publications. Until the next original publication is received (which is in future and not deterministic), vehicular rate observed by Vehicle-5, which is the latest, is considered to add further virtual entries that is required for decision-making.

5.4 Improvised Cooperative-approach

In the *Cooperative-approach* discussed before, publishing will happen by a vehicle at POI either when there is a change in road status or the TTL associated with the recent notification expires. Although, upon TTL expiration, a new publication is sent by the immediate next vehicle at POI, but the time of arrival of this immediate next vehicle at POI can not be known before.

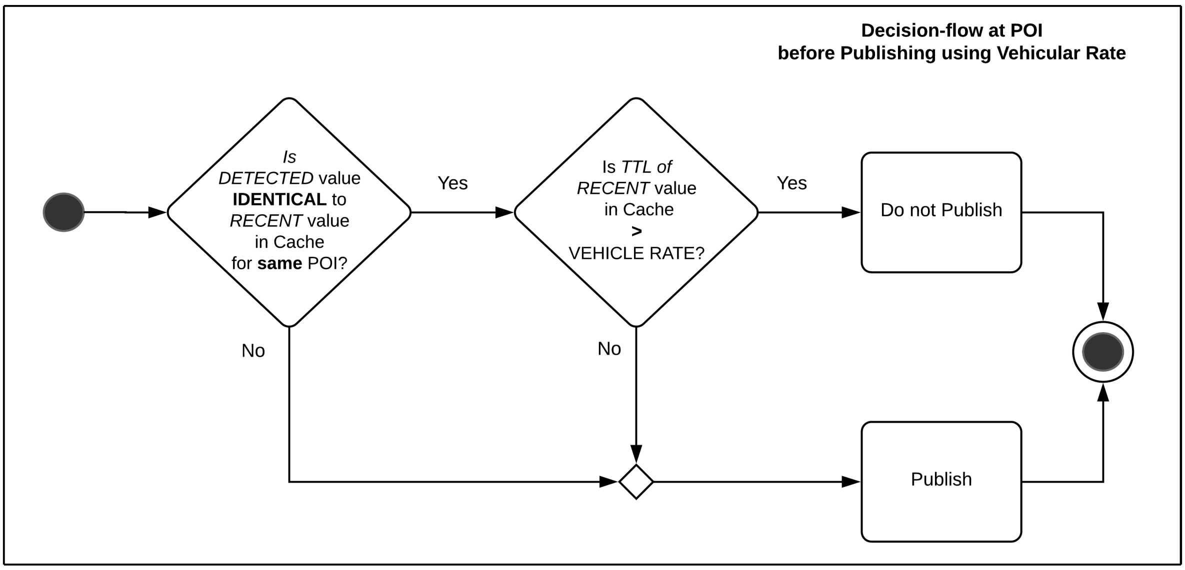


Figure 5.6: *Cooperative-approach* utilizing TTL and vehicular rate.

Imagine a real-world scenario, where TTL of the recent notification is still valid and a vehicle which is at POI decides not to publish since its detected value is same as the recent notification. However, assume the recent entry in cache expires immediately after this vehicle passes through, then this entry must be removed from cache. Now there is no entry in cache and the next publication would not happen until the next vehicle reaches POI. Then, in this scenario, there is no option to know the road status due to empty cache. To overcome this challenge, we consider TTL and vehicular rate to further optimize *Cooperative-approach*. The other advantage of using vehicular rate apart from for adding virtual entries can be seen in Figure 5.6.

To increase information freshness, before publishing, a vehicle compares the TTL of recent entry in cache with the observed vehicular rate at POI. If this TTL is less, then we allow the vehicle to publish. For instance, assume a vehicular rate such that for every 10 seconds only one vehicle passes through the POI (realistic during non-rush hours), and recent cache entry would be invalidated in 3 seconds due to its associated TTL. Now, with additional comparison implemented before publishing, the vehicle decides to publish to maintain information freshness instead of waiting till the next vehicle as it would have been done following previous flow.

To summarize, although there is some computational effort involved in measuring the vehicular rate, but its role is significant to meet the objective without causing overhead on cellular network. First, using vehicular rate, we add virtual entries, thereby still accounting for the vehicles which did not publish to avoid redundancy. Second, since a publishing vehicle knows this rate, it can make use of this opportunistically to enforce information freshness as seen in Figure 5.6.

Adhering to *Cooperative-approach*, publishing at POI should happen only for any of the following reasons:

- The publishing vehicle is the first vehicle to report the road status after an event. This will be the first publication for that POI.
- A change in road status such as a traffic jam which existed earlier has now dissolved. Hence, a new publication with the new status should be made possible to maintain information freshness.
- If TTL is associated with published notifications, then, once the validity of TTL expires, a new publication reflecting the road status must be allowed. Otherwise, there would be infinitely long living events, violating information freshness.

In Chapter 4, we briefly described the appropriateness and thus the selection of Publish/Subscribe mechanism for our scenario. In this chapter, we discuss the design and implementation of this communication mechanism. First, we discuss the assumptions related to vehicles and Publish/Subscribe semantics. Second, we discuss the realization of the Publish/Subscribe system using `Bypass.KOM`. Third, we introduce to the simulation environment used to evaluate our methods mentioned in Chapter 5.

6.1 Requirements and Assumptions

This section discusses the communication ability assumed for every vehicle. Later, we depict how and why a route is conceptualized as a set of smaller road segment called as road edge. Finally, we discuss the selection of a subscription scheme and their corresponding semantics for our scenario.

6.1.1 Road Edge

In our design, we assume each vehicle to be enabled with a cellular communication interface such as 3G/LTE to communicate with a central entity such as broker via cellular communication. A Wi-Fi communication interface is also assumed to communicate with other vehicles in an infrastructure less manner. Wi-Fi supports ad-hoc communication with a limited communication range, whereas cellular supports long range communication. Any vehicle that is not enabled with these technologies would not be involved in communication process and hence we have not considered those kinds in our study.

We also assume, every vehicle is aware of its absolute current location, target location which is its destination, and the route to its destination using GPS. In real-world, this can be related to a vehicle being equipped with GPS enabled Map Navigation. In our study, each vehicle has an identifier to uniquely distinguish itself among other vehicles. Hence, when a vehicle decides to publish, it can obtain locally the current location using its GPS, its own identifier, current system time and detected value from its on-board sensors as mentioned in the Premise-Subsection 4.4.1.

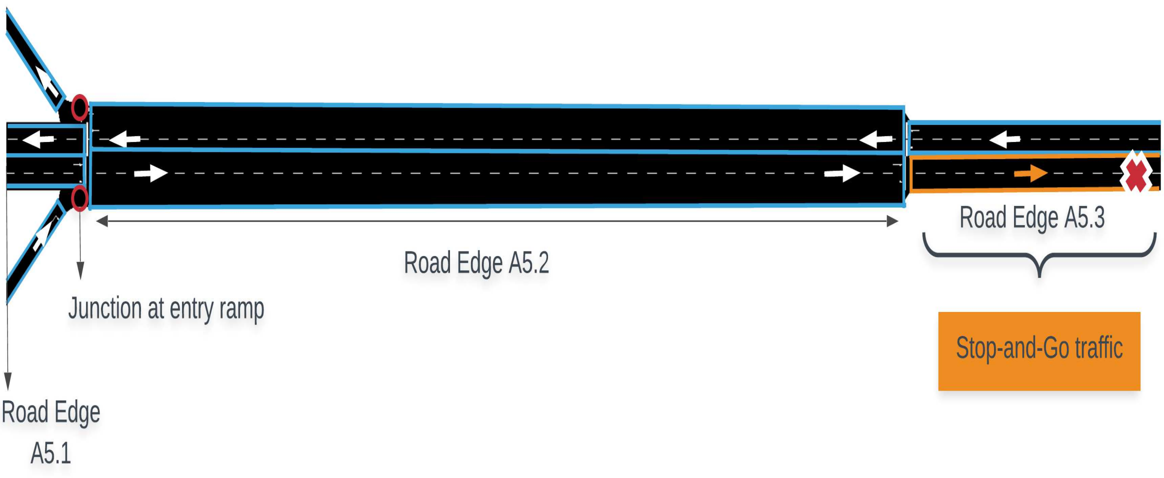


Figure 6.1: Overview of Road-Edges.

For efficient filtering schemes offering flexibility, we model the route as a set of ordered road edges as shown in Figure 6.1. This way, any publication reported can be tagged to a particular road edge with which only concerned vehicles subscribed to that road edge can be notified. To this end, all road edges are identified with a unique identifier.

Figure 6.1 represents a set of road edges. Each block is a unique road edge with its identifier. The Orange coloured block is the affected road edge. The POI and stop-and-go traffic depicted in our scenario in Chapter 4 is seen on the road edge A5.3. Any publications made in our study are from this road edge. To avoid complexity, we have considered stop-and-go traffic only on a single road edge and do not consider multiple POIs. To this end, in our study, all vehicles travelling on the road edge A5.3 are subscribed by default to traffic notifications from this road edge.

6.1.2 Subscription and Publication semantics

With respect to filtering scheme for subscription semantics, we consider attribute-based instead of channel-based. Channel-based subscriptions limit subscription only to a particular topic such as traffic-jam or bad weather. But it fails in offering granularity required to know the exact location for a particular event.

Generally there should be a generic way for subscribing to interested events by vehicles. Matching to our requirement, attributed-based filtering offers expressiveness to subscribe to a topic: `traffic-jam` for all the road edges which lie under a vehicle's interested route.

A subscription semantic is a combination of topic and the filter, i.e., (`topic,filter`), where filter is a set of attribute, its value and an operator, i.e., (`attr,value,operator`). Accordingly, with a `traffic-jam` topic and attribute-based filtering, a subscription for the road edge A5.3 in our case is represented as (`'traffic-jam','road-edge','A5.3','='`). On the contrary, a publication from the road edge A5.3, without being aware of the interested subscribers would be represented as (`road-edge,status`) = (`'A5.3','True'`) with status set to true for a traffic-jam on this edge or as (`'A5.3','False'`) for a no-traffic-jam, where status is set to false.

The reason to have a separate publication even in case of `no-traffic-jam` is to reflect the case of returning to normalcy after a existing jam dissolves. This change in event status is very important to maintain information freshness and for decision by vehicles at AOI.

6.2 Jam Detection Process

This section describes how a publishing vehicle detects and decides locally a given condition on a road edge as jam or not. This jam detection process is run on every vehicle irrespective of the road edge or the road status it is currently travelling on.

The jam detection algorithm shown in Figure 6.2, extended on a thesis work titled '*Hybrid Communication in Vehicular Networks*[1]', runs on each vehicle. In our system, every vehicle on a road edge decides whether the road condition as a jam or not by periodically observing its own speed. This fixed time interval for periodic observation of $\Delta\ \mathbf{t_{jam}}$ can be adjusted based on the computation overhead and required level of information freshness.

In our scenario, we focus on the road edges A5.1, A5.2 and A5.3. An accident and consequently, a stop-and-go traffic jam is seen on the road edge A5.3. Vehicles whose routes include above road edges to their destination are considered and discussed in our study. The flow model on other lanes does not have any effect on our algorithm as it should be ideally.

Vehicles have the option to choose subscription either for all the road edges on their route till destination or only to parts of the route. When subscription is only to parts of the route, it can be visualized as

[1] Master-Thesis Title: Hybrid Communication in Vehicular Networks: Combining the Advantages of Cellular and V2V Communication, Author: Sophie Schönherr, KOM-M-0599, KOM - TU Darmstadt. Date: 01. December 2017

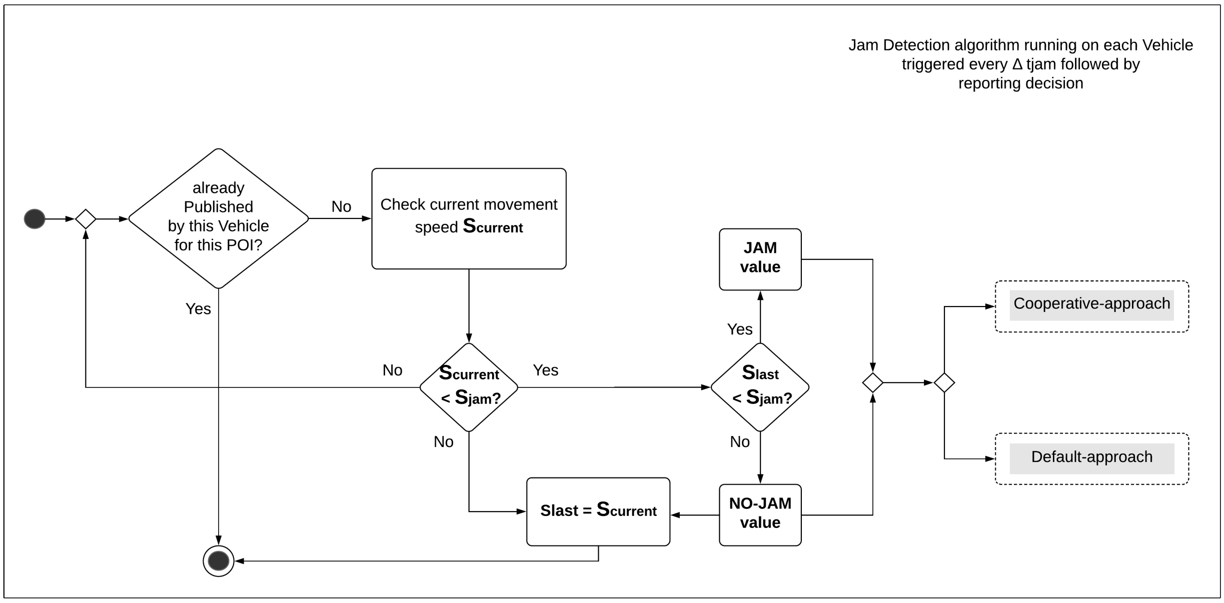


Figure 6.2: Jam detection algorithm, adapted and extended from the work '*Hybrid Communication in Vehicular Networks*'.

('traffic-jam','A5.1','='), ('traffic-jam','A5.2','='), ('traffic-jam','A5.3','='). As vehicles periodically observe for their current location with an interval of Δ $\mathbf{t_{observe}}$ to check their current road edge, it can delete any notifications in its cache for a road edge it has successfully travelled. In this way, as and when it moves along its road edges, it subscribes to next *n*-road edges and clears cache for the deprecated edges.

A vehicle takes the role of a subscriber when moving at AOI and a publisher when moving at POI. If a vehicle observes its current speed (marked as $S_{current}$) and speed known from previous observation (marked as S_{last}) to lie below a particular threshold speed (marked as S_{jam}), it decides the road condition to be a `traffic-jam` on that road edge. On the contrary, if the observed speeds are above the threshold speed, a `no-traffic-jam` state is detected.

Upon detecting the road condition, if a *Default-approach* of reporting is followed, then it is simply published. Otherwise, with *Cooperative-approach*, it checks with the recent value before publishing and follows accordingly. The publication for a detected jam would be represented as ('A5.3','True').

Since it is a stop-and-go traffic, and for the reasons discussed in Subsection 4.4.2, we consider only single publication from a vehicle for that road edge and not multiple publications. Since those vehicles that are not subscribed will not receive any notification, and hence, in our study, all vehicles travelling on the discussed road edges are inherently subscribed.

For consistency and accuracy with respect to simulation parameters, our vehicular behaviour is modelled in a pseudo-random way and hence reproducible with the same setting. This helps to compare performance of different approaches in simulation environment.

6.3 Design of Publish/Subscribe system using `Bypass.KOM`

As befitting to our requirement of location-based Publish/Subscribe, we use the `Bypass.KOM` proposed by [Ric17] that enables transition in event brokering. It offers distinct type of event mechanisms that includes filtering schemes for location-based Publish/Subscribe and locality-aware data dissemination mechanism. In this section we give an overview of components involved in `Bypass.KOM` and discuss the benefits of using this design. Later, we discuss the assumptions considered in order to match our scenario.

`Bypass.KOM` – is a modular Publish/Subscribe middleware framework that offers distinct transition mechanisms for location-based filtering and locality-aware event dissemination. The framework offers adaptive Publish/Subscribe enabling transition to react to changing environmental conditions. This feature is suitable for mobile entities especially for vehicular networks. Since Publish/Subscribe is an anonymous distributed communication paradigm, semantics of subscription and publication plays a major role in disseminating only relevant data.

The filtering of subscription and publication can be based on Channel-based, Topic-based or Content-based. Each of these filtering schemes provides their own level of granularity in expressiveness of messages, and therefore involves corresponding broker-overhead. Based on trade-off between broker-overhead and granularity required, transition among the location-based filtering schemes can be achieved using this framework.

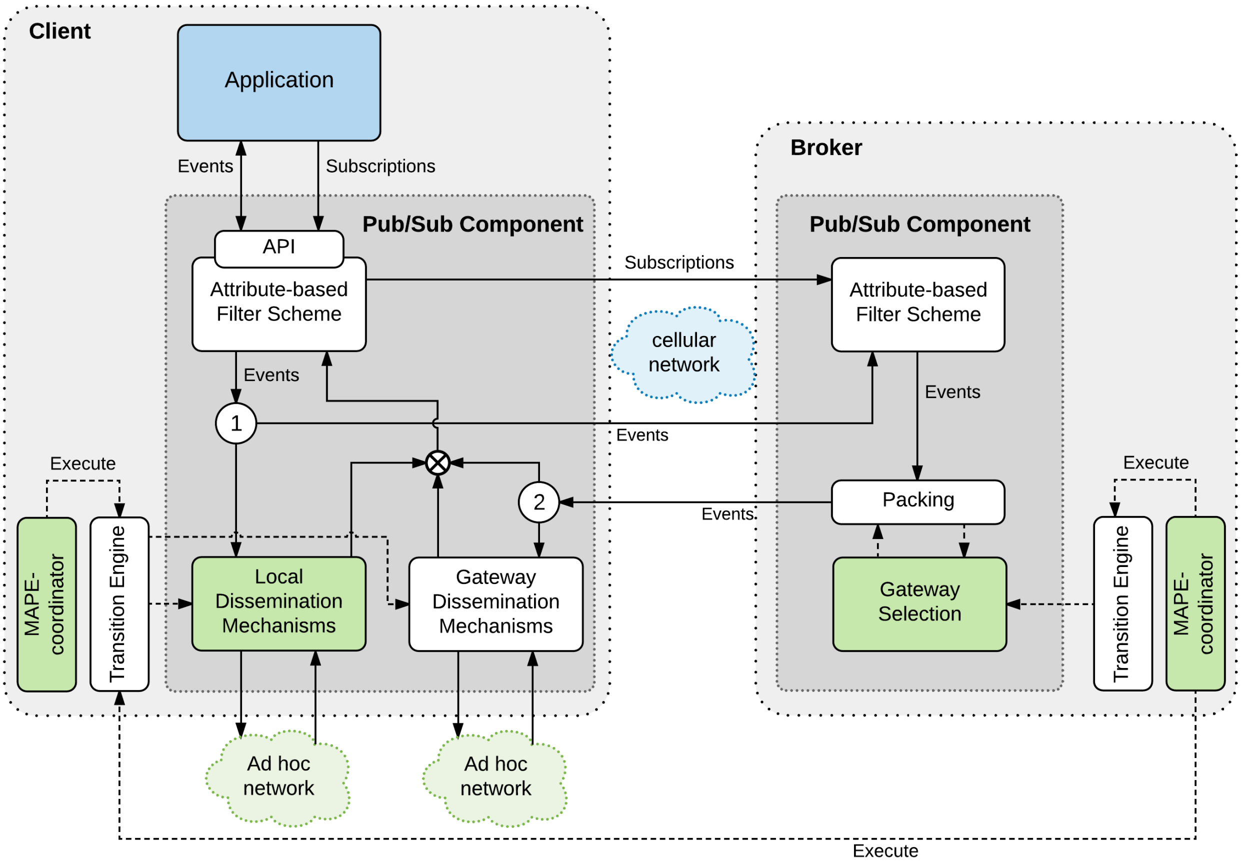


Figure 6.3: The Publish/Subscribe system component adapted from the work '*Hybrid Communication in Vehicular Networks*', extended from `Bypass.KOM`.

`Bypass.KOM` also offers transition for locality-aware event dissemination. A centralized broker via cellular network provides long distance data dissemination whereas brokers distributed throughout ad-hoc network initiate event dissemination locally. The transition among centralized broker network or decentralized peer-to-peer broker network can be selected based on trade-off of communication coverage and transmissions costs.

As transition-mechanisms are enabled; any upcoming or future filters can be utilized on the run without much effort. This feature is particularly beneficial in vehicular settings, as a vehicle can subscribe not just to traffic alerts (e.g., congestion, delay, accident, broken vehicle, construction or maintenance work) but also to other relevant road side service utilities (e.g, location and price-list of service and gas stations, location and menu at restaurants).

The overview of components involved in `Bypass.KOM` is given in Figure 6.3. A client communicates via API to interact with `Bypass.KOM` framework. The client subscribes to events based on its current location with a chosen filtering scheme. This subscription is sent to broker via cellular communication and the broker registers this subscription with the client's current location and chosen filtering scheme. On the contrary, events produced by clients (publishers) are disseminated to broker by the chosen dissemination mechanisms at the client.

Although, originally this framework was realized for social mobile applications, however, to match the requirements of high speed vehicular networks, it was extended in a thesis work titled '*Hybrid Communication in Vehicular Networks*[2]'. Here, an attribute-based filtering scheme is used for subscription matching for the reasons discussed earlier. The components of this hybrid system, extended based on original `Bypass.KOM` is represented in Green color and those implemented directly is represented in Blue color in the Figure 6.3.

The designed hybrid system offers dissemination using not only pure ad-hoc network or cellular-network but also a combination of both based on scenario and transmission costs trade-off. For our study, we consider dissemination using only pure cellular network since dissemination is not dependent on other vehicles as it would be in ad-hoc network. To this end, as seen in figure, for individual transmission, all subscribed vehicles are selected as gateways and no dissemination is performed using *gateway dissemination mechanism*.

6.4 SUMO - Traffic Simulator

To emulate the road conditions to a real-world scenario, we use SUMO traffic simulator environment. SUMO – *Simulation of Urban Mobility* is a free and open source traffic simulation suite developed by German Institute of Transportation Systems [DLR18].

It is beneficial for modelling road environments that includes vehicles, multitude of public transportation systems, and pedestrians for various tasks. Tasks such as route finding, visualization, network import, and emission calculation can be simulated with real-world road conditions similar to traffic-jam, road blockage, slow traffic due to construction work, maximum speed in construction site of a road, highways with interchanges and many others.

It provides API for remote deployment and interaction with its environment using TraCI (*Traffic Control Interface*). We make use of TraCI to connect to remote SUMO and to emulate our proposed method on vehicles.

For each simulation in SUMO, the following components are required.(a) Configuration file (*.sumocfg), (b) Settings file (*.settings.xml), (c) Networks file (*.net.xml), (d) Route file (*.rou.xml). The configuration and setting files define the settings to run simulation within the SUMO-GUI, whereas the route and network files represent the kind of road network to run in the simulation.

A road edge can be a single-lane or multi-lane with different speed restrictions as in real-world scenario. In our case, as seen in Figure 6.1, road edge A5.3 operates in a two-lane traffic movement under normal conditions, namely `(A5.3_0)` and `(A5.3_1)`. Since we wanted to have a stop-and-go traffic, we blocked one of those two lanes, allowing only one lane to be operational as seen in Figure 4.2. This was implemented by setting the speed of the blocked lane to a value close to zero. If this value is set to exactly zero, SUMO interprets that lane for the entire stretch of road edge to be completely unusable which should not be the case in our scenario. The other operational lane was set to a lower speed to emulate stop-and-go traffic. `('EdgeId','LaneId','Speed') : ('A5.3','(A5.3_1)','1.0')` and `('A5.3','(A5.3_0)','12.0')`

2 Master-Thesis Title: Hybrid Communication in Vehicular Networks: Combining the Advantages of Cellular and V2V Communication, Author: Sophie Schönherr, KOM-M-0599, KOM - TU Darmstadt. Date: 01. December 2017

6.5 Integration with SIMONSTRATOR.KOM

We use SIMONSTRATOR.KOM [RSRS15], a Java-based simulation framework for distributed systems, for simulation as it already implements the Publish/Subscribe middleware Bypass.KOM used by vehicles in our work.

SIMONSTRATOR.KOM is a modular component-based simulation framework for distributed systems supporting heterogeneity. It is also a prototyping platform offering different run time environments to integrate with existing simulators and test beds. It supports integration with PeerfactSim.KOM [SGR+11] run time environment to remotely access the described SUMO – traffic simulator via TraCI- API.

PeerfactSim.KOM is a discrete-event simulation engine to investigate P2P systems in a large scale where peers communicate among each other by exchanging messages. Hence it supports running experiments with thousands of peers as required in a vehicular environment with varying density of vehicles.

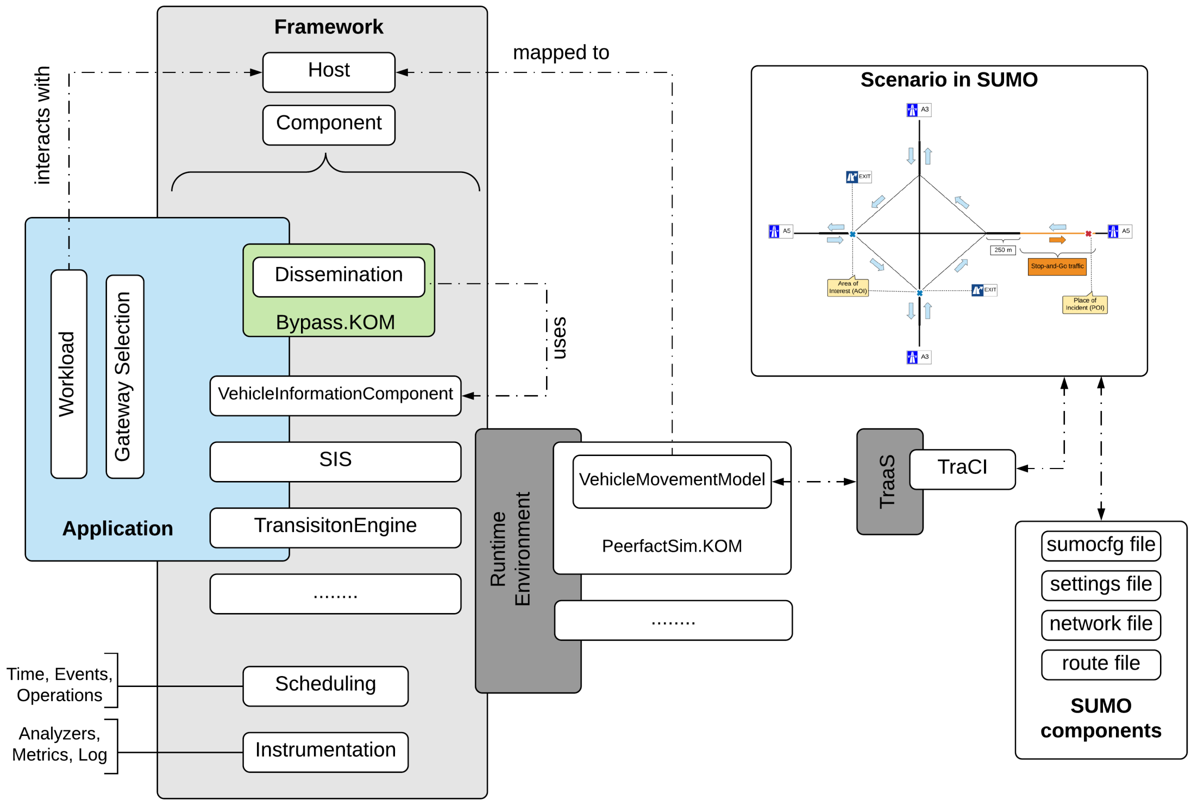


Figure 6.4: Overview of integrated components from SIMONSTRATOR.KOM platform (adapted from [RSRS15]) with SUMO.

An overview of components of SIMONSTRATOR.KOM and its integration with SUMO is given in Figure 6.4. The *scheduling* component provides representations for absolute and relative time with respect to events and operations. The *instrumentation* layer captures data from the simulations carried for evaluation and debugging. Metrics and logs against which a simulation is to be analysed are visualized here in a modular way.

Every individual entity in the framework is modelled as a host component in this framework. It acts as a container for the component where the functionality of a host is initialized with. In our case, for the simulation, every host in SIMONSTRATOR.KOM is a corresponding vehicle in SUMO. And as every host implements a component, correspondingly, every vehicle implements our proposal – *Adaptive cache management* described in Chapter 5.

A host also implements the dissemination mechanisms from `Bypass.KOM`, i.e., using Publish/Subscribe. In our case, this dissemination mechanism is implemented in the workload component, which includes the implementation of either event notification using *Default-approach* or using our proposal *Cooperative-approach*.

TraCI is encapsulated by *TraCI as a Service* (TraaS)[3] to enhance the ease of communication with SUMO. To this end, behaviour and state information from any one particular vehicle out of many vehicles running in the simulation can be carried with ease.

As any simulation carried is with an objective to evaluate the effect of some parameter in some settings, a number of settings files would be needed. As `SIMONSTRATOR.KOM` is designed with loose coupling of components, it is simple to run only those settings required in the scope of carrying that simulation. This architecture with low coupled component-based design in parallel with generic interface for different run time, made it suitable to implement and deploy our method in it.

6.6 Traffic Flows

To include real-world traffic behaviour where traffic changes over time throughout a day, different workloads with varying traffic characteristics are created. We consider four different type of vehicles namely *fast-cars*, *cars*, *trucks*, and *trailers* with their corresponding maximum speeds of 180 km/h, 130 km/h, 100 km/h, and 80 km/h. Different traffic flows are configured where each traffic flow is instantiated with one of the following vehicle distributions, namely `80-180`, `80`, `100`, `130`, and `180`.

A vehicle distribution contains the kind of vehicles it is composed of. For example, `100` type distributions compose only trailer type vehicles whereas 80 type distributions are composed of only truck type vehicles. A composition of all the above mentioned vehicles is configured in `80-180` type distribution as seen in Listing 6.1[4].

The percentage of combination of different vehicles can be configured. As seen in the listing, for `80-180` type distribution, it is composed of 10% trailers, 30% trucks, 50% cars, and 10% fast-cars. 70% of all vehicles configured in each type distribution travel between 80% to 100% of their corresponding maximum speeds. Due to this, traffic flows with different average movement speeds can be configured.

Subsequently, in order to have varying vehicular densities, the probability of a traffic flow can be set to a value between 0 (least) and 1 (highest). The probability of the flows in the listing is set to $pFlow_n$, which signifies the average rate at which a vehicle is spawned in SUMO for every $1/pFlow_n$ seconds. In our simulation, *typedist* and $pFlow_n$ have real values ranging between 0 and 1.

Listing 6.1: Vehicle Type Distributions and Flows for a Workload

```
<routes>
<vTypeDistribution id="80-180">
<vType id="trailer" probability="0.1" maxSpeed="22.22" length="15"
accel="1.1" decel="0.4" speedDev="0.9" speedFactor="0.1"/>
<vType id="truck" probability="0.3" maxSpeed="27.8"
accel="1.3" decel="0.4" length="10" speedDev="0.9" speedFactor="0.1"/>
<vType id="car" probability="0.5" maxSpeed="36" length="5"
accel="2.9" decel="7.5" speedDev="0.9" speedFactor="0.1"/>
<vType id="fast_car" probability="0.1" maxSpeed="50" length="5"
accel="2.9" decel="7.5" speedDev="0.9" speedFactor="0.1"/>
</vTypeDistribution>

<vTypeDistribution id="80">
```

[3] www.traas.sourceforge.net/cms, accessed on: 01.02.2018

[4] Master-Thesis Title: Hybrid Communication in Vehicular Networks: Combining the Advantages of Cellular and V2V Communication, Author: Sophie Schönherr, KOM-M-0599, KOM - TU Darmstadt. Date: 01. December 2017

```
<vType id="trailer" probability="1" maxSpeed="22.22" length="15"
accel="1.1" decel="4" speedDev="0.9" speedFactor="0.1"/>
</vTypeDistribution>

<vTypeDistribution id="100">
<vType id="truck" probability="1" maxSpeed="27.8" length="10"
accel="1.3" decel="4" speedDev="0.9" speedFactor="0.1"/>
</vTypeDistribution>

<vTypeDistribution id="130">
<vType id="car" probability="1" maxSpeed="36" length="5"
accel="2.9" decel="7.5" speedDev="0.9" speedFactor="0.1"/>
</vTypeDistribution>

<vTypeDistribution id="180">
<vType id="fast_car" probability="1" maxSpeed="50" length="5"
accel="2.9" decel="7.5" speedDev="0.9" speedFactor="0.1"/>
</vTypeDistribution>
<flows>
<interval begin="0" end="1000">
<flow id="1" type="typedist" probability="pFlow1" from="-A5.start" to="A5.4"
departPos="base" departLane="random" departSpeed="max"/>
<flow id="2" type="typedist" probability="pFlow2" from="A5.start" to="-A5.4"
departPos="base" departLane="random" departSpeed="max"/>
<flow id="3" type="typedist" probability="pFlow3" from="-A3.start" to="A3.4"
departPos="base" departLane="random" departSpeed="max"/>
<flow id="4" type="typedist" probability="pFlow4" from="A3.start" to="-A3.4"
departPos="base" departLane="random" departSpeed="max"/>
<flow id="5" type="typedist" probability="pFlow5" from="-A3.start" to="A5.4"
departPos="base" departLane="random" departSpeed="max"/>
<flow id="6" type="typedist" probability="pFlow6" from="-A5.start" to="-A3.4"
departPos="base" departLane="random" departSpeed="max"/>
<flow id="7" type="typedist" probability="pFlow7" from="A3.start" to="-A5.10"
departPos="base" departLane="random" departSpeed="max"/>
<flow id="8" type="typedist" probability="pFlow8" from="A5.start" to="A3.4"
departPos="base" departLane="random" departSpeed="max"/>
</flows>
</routes>
```

7 Evaluation

In this chapter, we evaluate our two proposals namely, *Cooperative-approach* of reporting and *Adaptive Cache Management* mechanism supporting varying vehicular rate. As depicted in Figure 6.4, we use `SIMONSTRATOR.KOM` platform for our evaluation in which the road scenario discussed in Figure 4.1 is configured on SUMO.

To emulate real-world scenario, where traffic flow varies throughout a day over time, we have configured different workloads. Each of the workloads offers a separate vehicle-flow model, comprising of different vehicles including fast-cars, cars, cars with trailers, and trucks all travelling at different maximum speeds. The proportion and distribution of these vehicles within each vehicle-flow model provides different vehicular densities and vehicular rate.

We evaluate our two proposals for cellular transmission costs and accuracy achievable for decision-making. This chapter is furthermore organized as follows. First, in Section 7.1, we provide the simulation settings, parameters and metrics used for evaluation. Second, in Section 7.2, we evaluate reporting approaches for their transmission costs. Third, we evaluate in Section 7.3 for accuracy-levels achievable with different combination of reporting and caching mechanisms. Later, the impact of vehicular density on transmission costs and its associated accuracy is analysed in Section 7.4. In the end, in Section 7.5, we evaluate how these different combinations perform using frequently used decision-making algorithms followed by a proof-of-concept for optimization.

7.1 Evaluation Setup

Simulation setup with default parameters, metrics along with combination of reporting and caching methods used in evaluation are discussed in this section. First, we provide the simulation parameters along with representation of changing road conditions that which is set up in simulation. Following that, we explain the combinations of reporting and caching mechanism that are evaluated for. Lastly, we mention the evaluation metrics against which the combinations are evaluated.

7.1.1 Simulation Parameters and Configuration

Simulation Parameters:
For our simulation, we assume vehicles to be subscribed to all those road edges that lie on their route towards destination. Thus, when vehicles enter a new road edge, they simply unsubscribe from the previous deprecated road edges and delete those corresponding cache entries. For this to happen, as explained in Section 6.2, vehicles keep checking for every $\Delta\ \mathbf{t_{observe}}$ time interval to check their current road edge.

In our study, we have considered jam only on the road edge A5.3. To this end, all vehicles willing to travel on this road edge in near future are subscribed for traffic-alerts from this road edge by default. For activating jam on A5.3, we decrease allowed speed on that road edge to a value close to zero at times whenever we want to have a `traffic-jam` scenario. Correspondingly, vehicles during a `traffic-jam` scenario, travel at speeds less than 15 km/h, which is the threshold speed assumed for the vehicles to detect a jam. In other case, when a `no-traffic-jam` is required, there is no restriction of allowed speed, and hence vehicles travel at their normal speeds (i.e., greater than 15 km/h). The jam detection algorithm triggers on each vehicle every $\Delta\ \mathbf{t_{jam}}$ time interval to check whether road condition is jam or not as shown in Section 6.2.

To be realistic, the latency of cellular communication is set at 100 ms with a variance of 25 ms. The cellular coverage is assumed for all vehicles seen at any time in the simulator without any transmission errors. Therefore, any publication sent at a particular time is received by all subscribers seen

PARAMETER	SYMBOL	UNIT	DESCRIPTION
Observation Interval	$\Delta\ \mathbf{t_{observe}} = 1$	s	Time interval for which a vehicle checks if it is in a new Road Edge
Jam Threshold Speed	$\boldsymbol{S_{jam}} = 15$	km/h	Threshold Speed below which vehicles detect road condition as `traffic-jam`
Jam Detection Interval	$\Delta\ \mathbf{t_{jam}} = 10$	s	Time interval for which a vehicle checks if it is in a `traffic-jam`
Jam Location	A5.3	none	Road Edge where accident has occured - POI
Jam Duration	$\mathbf{T_{TJ}} = 6.66$	min	Duration of each `traffic-jam` scenario
Time-to-live	$\mathbf{TTL} = 400$	s	Validity of a notification from the time of its publication
Simulation Duration	$\mathbf{T_{sim}} = 30$	min	Total duration for each simulation
Cellular Latency	$\mathbf{l_c} = 100$	ms	Latency of cellular connection

Table 7.1: Overview of parameters for each simulation

in the simulator at that time. The cellular upload bandwidth concerning to publications and cellular download bandwidth concerning to forwarding notifications are both assumed to match and serve our requirements. This assumption is independent of our objective and does not have negative impact on our evaluation. The duration of each simulation in our evaluation is 30 minutes. An overview of these parameters are summarized in Table 7.1.

Workload and Traffic flow:
For evaluating our methods in simulations with different vehicular densities, different workloads by varying respective traffic flows are created in SUMO.

TYPE	WORKLOAD	DESCRIPTION	DENSITY
Speed	`80`	Trailers (max. speed 80 km/h)	-
	`100`	Trucks (max. speed 100 km/h)	-
	`130`	cars (max. speed 130 km/h)	-
	`180`	Fast-cars (max. speed 180 km/h)	-
	`80-180`	10% trailers, 30% trucks, 50% cars, 10% fast-cars	-
Spawn Rate	`VLOW`	$\text{pFlown}_n = 0.05$, $\text{pFlown}_m = 0.01$	Very Low
	`LOW`	$\text{pFlown}_n = 0.1$, $\text{pFlown}_m = 0.02$	Low
	`MED`	$\text{pFlown}_n = 0.3$, $\text{pFlown}_m = 0.06$	Medium
	`HIGH`	$\text{pFlown}_n = 0.9$, $\text{pFlown}_m = 0.18$	High

Table 7.2: Overview of Speed and Spawn Rate configuration for workloads

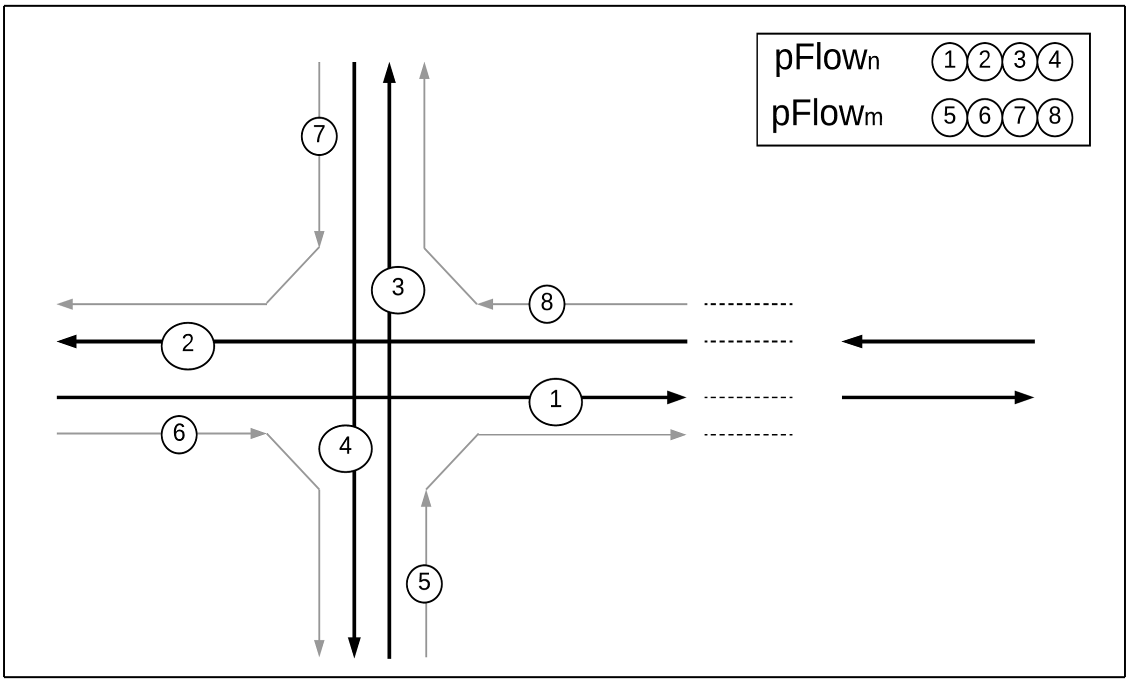


Figure 7.1: Major and Minor Flows modelled for vehicles in SUMO

As seen in Figure 7.1 that depicts our scenario, flows with ids = `1, 2, 3, and 4` are major traffic flows and other flows with ids = `5, 6, 7, and 8` are minor traffic flows. Four different workloads are considered for simulation namely `VLOW`, `LOW`, `MED`, and `HIGH`. Each of these workloads correspond to *verylow*, *low*, *medium*, and *high* vehicular densities that they offer. Each workload is a combination of a vehicle's spawn rate configuration and its speed configuration. This is represented in Table 7.2. As outlined in Section 6.6, `80-180` type distribution corresponds to a combination of vehicles in a manner that comprises 10% trailers, 30% trucks, 50% cars, and 10% fast-cars. To this end, in our simulation, possibilities of vehicles overtaking is valid.

Change of Road Events over Simulation Duration:
In order to evaluate the adaptability of *Adaptive Cache Management* mechanism to varying road conditions, we create an alternating `no-traffic-jam` and `traffic-jam` conditions over the entire simulation duration.

This set up can be seen in the Figure 7.2. Simulation starts at `t = 0` min and ends at `t = 30` min. There are two periods of `traffic-jam` observed in total, each lasting for a duration of 400 seconds. The first `traffic-jam` occurs at `t = 5` min. This jam is resolved at `t = 11.66` min. Similarly, second `traffic-jam` occurs at `t = 18.33` min and resolves at `t = 25` min. The simulation ends at `t = 30` min. When the simulation starts, the road condition is in `no-traffic-jam` state. To indicate this, every vehicle, by default, at the start of simulation is assumed to have a cache value of false.

7.1.2 Combinations of Reporting and Caching Mechanisms

In this subsection, an overview of the combinations of reporting and caching mechanisms that are evaluated is provided. Reporting-approaches discussed in previous chapters signify when and how a road status should be reported by a vehicle to alert other vehicles. With assistance from timely reporting of an event, a vehicle can avoid traffic jams by choosing alternative routes, if any available.

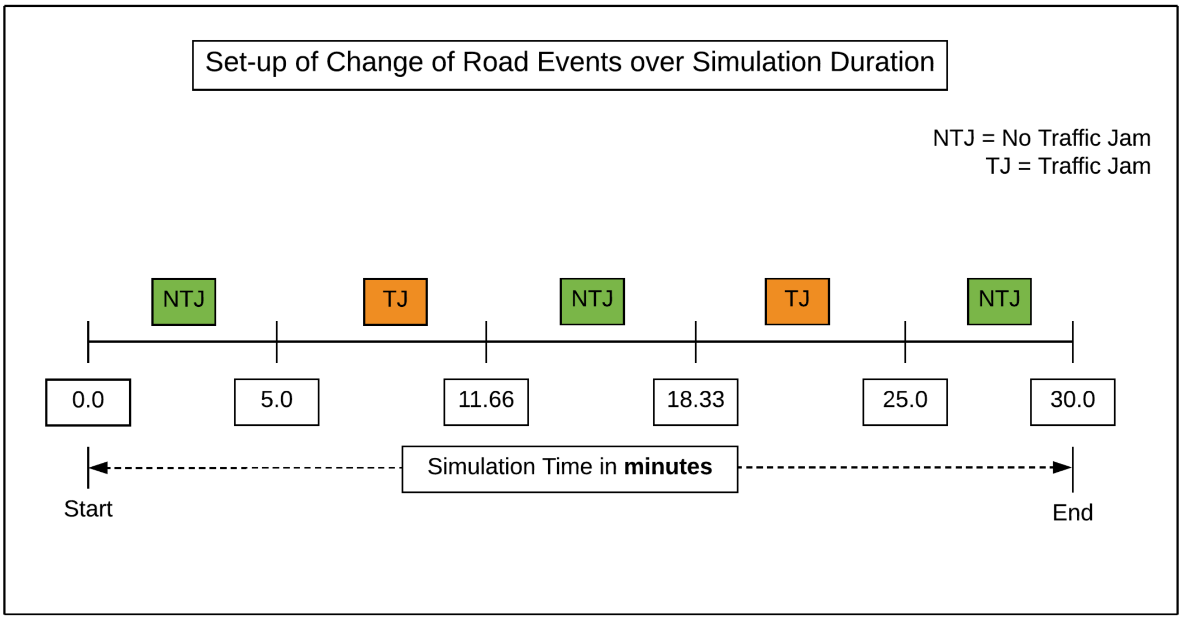


Figure 7.2: Change of road events over simulation duration on Road Edge A5.3.

Default-appraoch (explained in 4.5), follows a simple method of reporting where all vehicles report their detected road status. Whereas in *Cooperative-approach* (explained in 5.1.1), a vehicle verifies with its recent cache entry to decide whether to report or not with a publication. In our evaluation, we evaluate both these approaches for their associated cellular transmission costs.

A vehicle at AOI can avoid travelling towards a road that is in a traffic jam state by making a rerouting decision, provided an alternative route is available. A subscribed vehicle receives notifications about road state of a particular road it is interested in, and stores all these notifications in its cache. To make rerouting decision (whether to continue towards that interested road, or reroute and avoid), a vehicle relies on its notifications that were received and cached. Better the number of cache values that mirror the current situation of the road, better are the chances for vehicles to make a correct decision. Therefore, for decision-making, the role of cached values, and thus the number of received notifications is significant. Due to this associated importance, *Adaptive Cache Management* with virtual entries was proposed in Chapter 5. This designed caching mechanism adapts to variations in vehicular rate to reflect the real-world scenario of varying vehicular rates observed throughout a day.

REPORTING METHOD	CACHING MECHANISM	SYMBOL
Default-approach	simple without virtual entries	Default(none)
Cooperative-approach	simple without virtual entries	Co-op.(none)
Cooperative-approach	Adaptive Cache Management(ACM) considering constant vehicular rate(VR)	Co-op.+ACM(const.VR)
Cooperative-approach	Adaptive Cache Management(ACM) considering varying vehicular rate(VR)	Co-op.+ACM(vary.VR)

Table 7.3: Overview of combination of reporting and caching methods evaluated

Overall, to understand the importance of choosing a reporting approach combined with an appropriate caching mechanism, we construct different combinations of discussed methods. The overview of these methods is summarized in Table 7.3. Each combination is evaluated for higher decision accuracy while keeping the transmission costs low.

The caching mechanism named *simple without virtual entries* represents cache management where only received notifications are maintained in cache. In this case, any decision-making process happens purely based on received notifications alone, and hence no virtual entries are added. This is visualized in Block-A of the Figure 5.3. Since, with *Default-approach*, irrespective of road status all vehicles publish, and correspondingly all vehicles are accounted in cache, there is no need to add virtual entries. The second combination seen in the table is for *Cooperative-approach* but with no virtual entries. To understand and analyse the impact of not accounting the skipped publications via virtual entries, this combination is crucial. In the third combination, we use *Adaptive Cache Management* with a constant vehicular rate of 1 vehicle/second. To examine the necessity of accurate vehicular rate measurement for *Adaptive Cache Management*, this combination is selected. Although, any constant vehicular rate can be chosen, but to be realistic as possible without high deviation from vehicular rate known from workloads, 1 vehicle/second is chosen. In the final combination, we evaluate for *Cooperative-approach* of reporting combined with *Adaptive Cache Management* considering varying vehicular rate.

7.1.3 Evaluation Metrics

In order to evaluate the efficiency and advantage of one approach over the other for the studied scenario, following metrics are used.

(1) Cellular Transmission Cost:
As our data dissemination system uses cellular network to alert vehicles using publications and subscriptions, keeping the network load to as minimal as possible is also one of our goals. Considering this, we found it necessary and important to use the following metrics:

- Cellular Upload - Number of Publications sent to cellular entity by Publishing Vehicles.
- Cellular Download - Number of Notifications forwarded by cellular entity to Subscribed Vehicles.

(2) False Detection Rate:
As commonly seen in real-world, faulty measurements and inaccurate reading from vehicle's sensors is possible. An accurate information is that which is reliable due to error-free attributes. Not all cars detect a road condition correctly and accurately at all times. This inaccuracy issue affects the decision-making process significantly as each vehicle inherently derives decision based on its cache values that includes false notifications too.

Assume, although there is no jam observed on a road, out of five vehicles, three vehicles are reporting a road status wrongly as traffic jam. With these wrong notifications in cache, when a vehicle at AOI makes a Majority-based decision, it is bound to believe the road status as traffic-jam, though it is not. With this incorrect decision, a vehicle might reroute to an alternative route leading to inefficient traffic management. To incorporate the possibility of false detections in our simulation, we make use of following different false detection rates. With each of the following, the number of vehicles reporting correctly or wrongly is statistically distributed.

- 0% False Detection Rate = 100% Accuracy
- 1% False Detection Rate = 99% Accuracy
- 5% False Detection Rate = 95% Accuracy
- 20% False Detection Rate = 80% Accuracy

With 0% false detection rate, all vehicles report correctly representing the actual road scenario that existed at the time of publishing. This is the best case scenario without false detections. The worst case scenario we have considered is with 20% false detection rate. Here, most of the cars report wrongly compared to other rates shown. A worst value below 20% is not considered as otherwise it would mean almost as always a sensor would detect wrong road status, and it is as good as not trusting that reading.

(3) Accuracy:
Accuracy provides the degree to which an information correctly describes an event or situation. Accuracy for our study can be formalized as the ratio of cache entries with correct road status to the total number of cache entries. In order to evaluate the performance of different methods at different false detection rates and their adaptability to false measurements this metric is significant. Accuracy gives the distribution of *True Positives* and *False Negatives* in a set of data. As in our scenario, cache can contain wrong values due to false detection, the metric accuracy, provides the percentage of correctness of its cached entries.

It is given as: `P = [correct events in cache / total events in cache]`.
In our simulation, accuracy observed at a given time is the summation of individual accuracy observed by each vehicle that were seen in simulation from the start of simulation till the time of measurement.

(4) Decision-Making Algorithm:
The methods listed in Table 7.3 are evaluated for accuracy over following decision-making algorithms.

- ***Majority-based:*** In our scenario, since a road status is in a binary state, i.e., either `traffic-jam` or `no-traffic-jam`, a majority voting is applicable. Based on majority of values, the decision is made.
- ***Newest-based:*** In this algorithm, decision is made based on the recent value in cache alone and therefore all other cache entries are insignificant.
- ***QoI-based:*** QoI-based algorithm is a weight based algorithm that considers age of the information. Freshness-based majority voting decision for vehicular environment proposed by [MWL+18] considers the TTL associated with every notification. The idea is that longer the time has elapsed since the notification was received, the less accurate it becomes with respect to current road status. In our simulation, all notifications have the same TTL value. Hence, an old notification has lesser significance than the recent notification, but still the impact of old information is not completely neglected as it would be in the Newest-based algorithm.

Analysis of different methods mentioned are evaluated for these metrics to measure the effectiveness and benefits of one over the others. In the following sections, graphs obtained by multitude of simulations for different methods over mentioned metrics are analysed.

In the following sections 7.2 and 7.3, we evaluate different methods for the metrics transmission costs and accuracy using the workload MED. This is because, this workload provides a moderate combination of fast-cars, cars, trailers, and tucks offering moderate vehicular density. Later, in section 7.4, impact of different vehicular densities is provided.

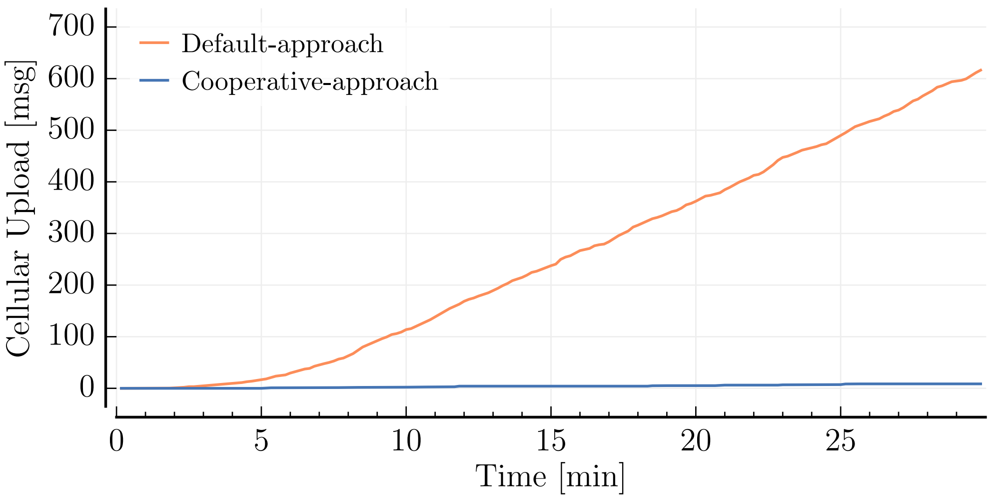
7.2 Evaluation of Reporting Approaches for Transmission Costs

Transmission costs accounts number of messages handled by a broker both during the process of cellular upload and cellular download. The two reporting approaches, Default-approach and Cooperative-approach are compared for the number of messages generated by each.

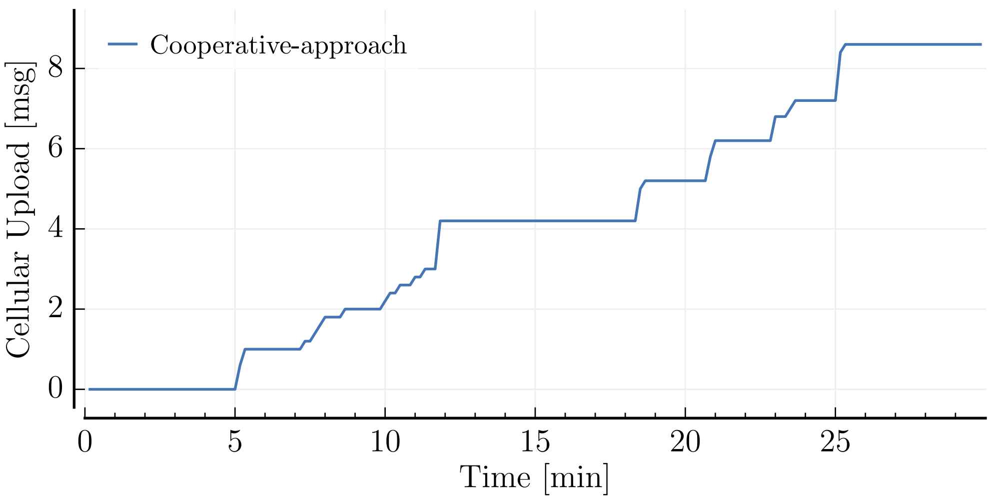
7.2.1 Cellular Upload

Cellular Upload at 0% False Detection Rate:
The number of publications published by all publishing vehicles constitute cellular upload messages. In Figure 7.3a, number of publications for both reporting approaches over the entire simulation duration in a 0% false detection rate is provided. The difference in overall number of publications published by Default-approach and Cooperative-approach is evident here. The significant difference in publications can be accounted to the publishing-algorithm followed by each approach before publishing.

In Default-approach, every vehicle passing through POI is a publishing vehicle. On the contrary, in Cooperative-approach, a vehicle verifies with its recent cache value concerning that POI and publishes if and only if the detected value is different.



(a) Default- and Cooperative-approach



(b) Cooperative-approach

Figure 7.3: Cellular Upload costs in 0% False Detection Rate Environment.

Every vehicle, by default, has a cache value of false indicating no-traffic-jam. As represented in Figure 7.2, when the simulation starts, the road condition is in no-traffic-jam state. Only at the 5^{th} min, the first change of road condition occurs changing to a traffic-jam state. However, since in Default-approach, every vehicle publishes irrespective of whatever value is detected, plot for this approach can be seen increasing even before appearance of first traffic jam. With continuous arrival of vehicles, publications starts right from first vehicle until last vehicle arriving at POI, and therefore change of road state has no effect on publishing algorithm in Default-approach.

On the contrary, first publication for Cooperative-approach occurs only at 5^{th} min when the change of road state occurs. Until this time all vehicles travelling with default cache value detect the same road state reflected in their cache and do not publish. Similarly, at all the other times when a new road state occurs (i.e., 11.6^{th}, 18.3^{th}, and 25^{th} min), there is a publication happening to reflect the new road status. The first vehicle arriving at POI during the time when a new road state occurs, this vehicle decides to publish upon checking with its recent cache entry. Since, this new road state is not yet observed in its cache, it publishes immediately. This way, although with less publications, any change of road states are correctly reported in Cooperative-approach. This can be seen in a step-like plot in Figure 7.3b where publications happen at the times when change of road state occurs.

However, on observation in Figure 7.3b, there are publications happening apart from the times when a new road state occurs. These publications occur at roughly 7^{th}, 21^{th}, and 23^{th} minute and all happen during the traffic-jam periods. The reason for these publications, although at these times no change of road state occurred, can be related to unaware-vehicles.

Unaware-vehicles are those vehicles that are unaware of the prevailing traffic-jam state because they were not existing when the publication corresponding to that traffic-jam happened. Due to this, unaware-vehicles which do not have the traffic-jam notification continue to still believe a no-traffic-jam state due to default cache value. Upon their arrival at POI, they detect this already existing traffic-jam as a newly changed road state and choose to publish. This situation where unaware-vehicles publish, happens only during a traffic-jam state since the default value false represents to a no-traffic-jam state as explained in Subsection 4.4.1.

Cellular Upload at 20% False Detection Rate:
The number of publications observed in an environment with 20% false detection rate is seen in Figure 7.4. In comparison to Figure 7.3a, Default approach has same plots for 0% and 20% false detection rates.

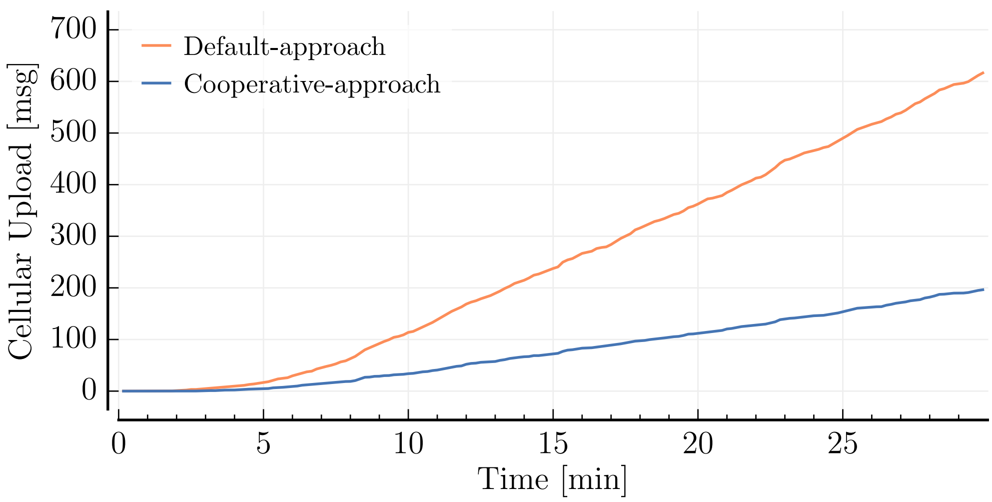


Figure 7.4: Cellular Upload costs in 20% False Detection Rate Environment.

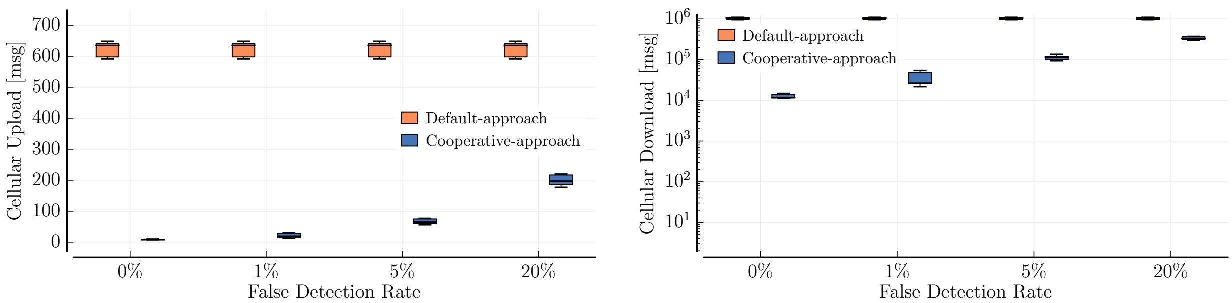
As all vehicles arriving at POI publish in this approach, false detection rates have no effect on number of publications. But for Cooperative-approach, as false detection rates increases to 20%, there is a gradual increase in the number of publications. This is because of two interdependent reasons. As there are vehicles with false detection, they involve in publishing the wrong road status after comparing to their recent cache value. These publications are wrong traffic alerts contradicting current road situation. However, any immediate vehicle that reports correctly, will contradict this recent wrongly updated traffic alert by sending a publication. This publication has correct road status value, and thereby negating the previous wrong alert. This process highlights the advantage of verifying detected value with a recent cache entry in Cooperative-approach, so that the effect of wrong notification is negated by an immediate correct notification.

Cellular Upload over 0% to 20% False Detection Rate:
The box-plot involving publications for different false detection rates is given in Figure 7.5a. For Default-approach, as mentioned above, irrespective of false detection rates all vehicles publish. Hence, there is no change from low to high false detection rates. With respect to Cooperative-approach, as false rates increase, the number of publications sent gradually increase so as to maintain correct information. However, it is critical to notice the significant difference in number of publications generated by both these approaches. Even in a 20% false detection rate environment, number of publications in Default-approach is as high as three times that of Cooperative-approach. As all vehicles publish in Default-approach, the number of publications sent is directly related to the number of vehicles that were seen at POI.

7.2.2 Cellular Download

The number of notifications received by all subscribed vehicles constitute cellular download messages. A publication published is forwarded by broker to all subscribed vehicles that exist at that time in the system. This includes the publishing vehicle too.

Cellular Download over 0% to 20% False Detection Rate:
The box plot for cellular download over different false detection rates is given in Figure 7.5b. As in Default-approach, irrespective of road state and false detection rate, all vehicles publish, this leads to very high number of subscriptions forwarded. Whereas with Cooperative-approach, this number is significantly very low. Due to false detections as publications increase in Cooperative-approach, correspondingly number of cellular download messages increases too, though this number is less compared to Default-approach.



(a) Cellular Upload **(b)** Cellular Download

Figure 7.5: Transmission costs over a 0% to 20% False Detection Rate Environments.

Every publication must be forwarded by broker to alert all the subscribed vehicles. Therefore, number of notifications forwarded by broker for a single publication is directly proportional to the number of subscribed vehicles. In this regard, every publication has immense effect on the load borne by broker. This can be seen in exponentially high number of notifications forwarded by broker in Default-approach.

The transmission costs associated in both reporting approaches in terms of publications and their corresponding notifications forwarded can be summarized in the two box plots Figure 7.5a and Figure 7.5b. It can be noted that compared to Default-approach, Cooperative-approach involves significantly less number of cellular messages and thus causes less overload on the cellular entity.

7.3 Evaluation of Caching Mechanisms for Accuracy using QoI-based Decision

Accuracy gives the percentage of correct information available in a vehicle's cache. In our study, since based on values available in cache, a vehicle must make a rerouting decision, accuracy plays a significant role in making the right decision.

A high percentage of correct information during a `traffic-jam` period means that vehicles are aware of this condition and can avoid this road edge by rerouting. On the contrary, high percentage of correct information during a `no-traffic-jam` period signifies that vehicles decide to continue on this road edge without concern of traffic-jam or congestion.

Any decision made wrongly due to incorrect cache values lead to inefficient traffic management. In this regard, in this section, we evaluate different methods for the accuracy metric at various false detection rates. Despite false readings and wrong publication, which combination of methods mentioned in Table 7.3 achieve a better accuracy is also evaluated. The overall accuracy achieved for a given simulation is the summation of accuracy over all the vehicles seen from simulation start till end. This way, accuracy achieved at a particular simulation time by the set of all vehicles that were seen until that time is obtained.

For decision-making process, a vehicle can choose any of the decision-making algorithms mentioned in Subsection 7.1.3. In our scenario, for a given simulation, we have evaluated for accuracy achieved using one of the decision-making algorithms running on all the vehicles.

In this section, we discuss the simulation results obtained for accuracy metric based on QoI-based algorithm proposed in [MWL+18]. The reason for this selection is that QoI-based algorithm is a weight-based model that considers the age of the information. In a distributed vehicular environment where the road status changes over time and false reporting due to false detections is possible, QoI-based algorithm provides both quicker adaptability to new road conditions and speedy recovery over false values. As depicted in the Figure 7.2, our simulation is characterized by alternating road conditions. To evaluate performance of different methods in these changing environment besides false detection rates, QoI-based algorithm is best suited.

7.3.1 Accuracy at 0% False Detection Rate

Accuracy achieved for different combination of methods at 0% false detection rate is as shown in Figure 7.6. As observed, although for Cooperative-approach, three different combination of caching mechanisms are evaluated, there are only 2 plots seen each for *Default(none)* and *Co-op.+ACM(vary.VR)*. All the combinations which follow Cooperative-approach (i.e., *Co-op.(none)*, *Co-op.+ACM(const.VR)* and *Co-op.+ACM(vary.VR)*) are represented in Blue color. There are two reasons for this behaviour. First, in all of these methods, Cooperative-approach for reporting a road status is followed, and therefore the number of publications and the publishing times are the same. Second, all the vehicles are reporting correct road status with 0% false detection. Since all publications are correct and QoI-based decision is applied, the effect of virtual entries and *Adaptive Cache Management* is not observed here.

As the first change of road state occurs only at 5^{th} min, until then, all the vehicles have a 100% accuracy reflecting the current no-traffic-jam state. Once at 5^{th} min, when the traffic-jam appears for first time, the accuracy for both the plots drop. The accuracy drops due to the reason that not all vehicles in the system have information about this new traffic-jam state. This drop in accuracy can be observed through out the simulation whenever the road state changes (i.e., at 5^{th}, 11.66^{th}, 18.33^{th}, and 25^{th} min).

Plot for the method - *Default(none)*, achieves overall a better accuracy than that of any method involving Cooperative-approach. This is because all the vehicles arriving at POI publish and subsequently are accounted explicitly in the cache without need of virtual entries. In this method, accuracy drops exactly at the times when a change of road state is observed. Once there is a drop due to new road state, the accuracy increases as more vehicles report and subsequently more vehicles become aware of this new

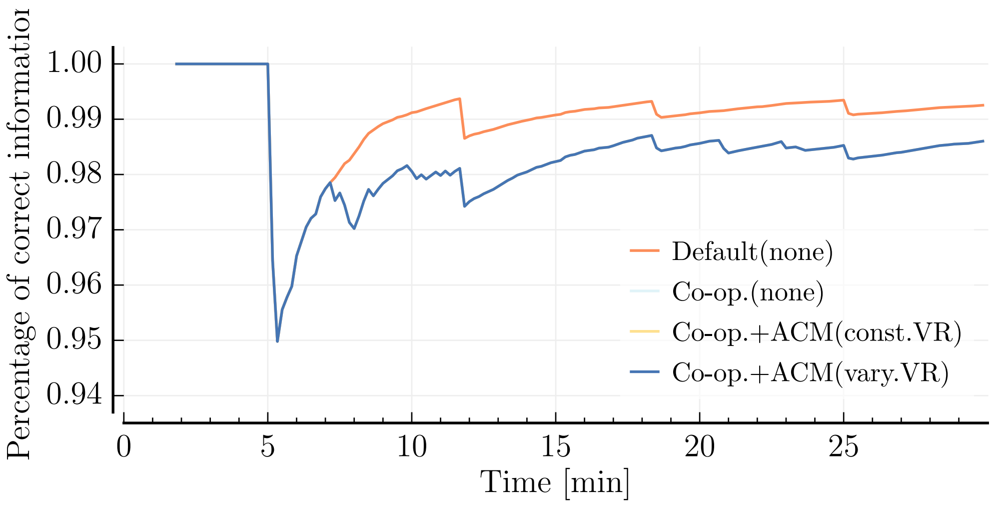


Figure 7.6: Accuracy achieved using QoI-based decision in 0% False Detection Rate Environment.

condition with continuous publications. This gradual increase in accuracy is seen until the time when next change occurs and following that accuracy drops again as expected.

Plot for the method involving Cooperative-approach behaves similarly by a drop in accuracy at the times when road state changes. However, apart from at these times, there is a drop observed at 7^{th}, 21^{th}, and 23^{th} minute. These drops are caused by the unaware-vehicles explained in subsection 7.2.1. These vehicles do not have traffic-jam notification which means they are not informed of the current road condition leading to an overall drop in accuracy.

The cost of unaware-vehicle in the system due to Cooperative-approach is observed here. This situation however doesn't occur in the method *Default(none)* as all continuously arriving vehicles publish and subsequently there are no unaware-vehicles. Unaware-vehicles are seen only in traffic-jam period. The first unaware-vehicle is seen at 7^{th} minute, and this is the point where plot for Cooperative method drops below than Default method. Similar to Default method, the plot for Cooperative method also increases gradually after a drop, however drops in Cooperative methods occur for both change in road state and due to unaware-vehicles.

7.3.2 Accuracy at 20% False Detection Rate

Accuracy achieved for different combination of methods at 20% false detection rate is as shown in Figure 7.7. As the false detection rate is at 20%, now there are vehicles reporting wrongly due to false readings, contradicting the actual road condition. As a first observation in contradiction to the previous Figure 7.6, now all the evaluated methods using Cooperative-approach are plotted and have a distinct behaviour due to false detections. This is discussed further.

As vehicles report wrongly, though the first change of road state occurs at 5^{th} min, accuracy for all methods drops even before this time. This is because of wrong notifications sent by vehicles which detected a traffic-jam falsely though it was not the case. Due to the combined effect of change of road condition at 5^{th} min and possibility of wrong reporting, accuracy for all methods drop heavily at 5^{th} min till 70%. Whereas in 0% false detection rate, accuracy dropped moderately at exactly 5^{th} min from 100% till 95% as observed in Figure 7.6.

The need of *Adaptive Cache Management* with virtual entries for accuracy is evident and can be observed in the plots of methods involving Cooperative-approach. The method *Co-op.(none)*, follows Cooperative-approach of reporting and a simple caching mechanism. This cache contains only received notifications without virtual entries which are required to account those vehicles that did not publish. Due to combined effect of not accounting vehicles which did not publish and wrong notifications in cache, accuracy drops as simulation progresses and is the worst among all methods.

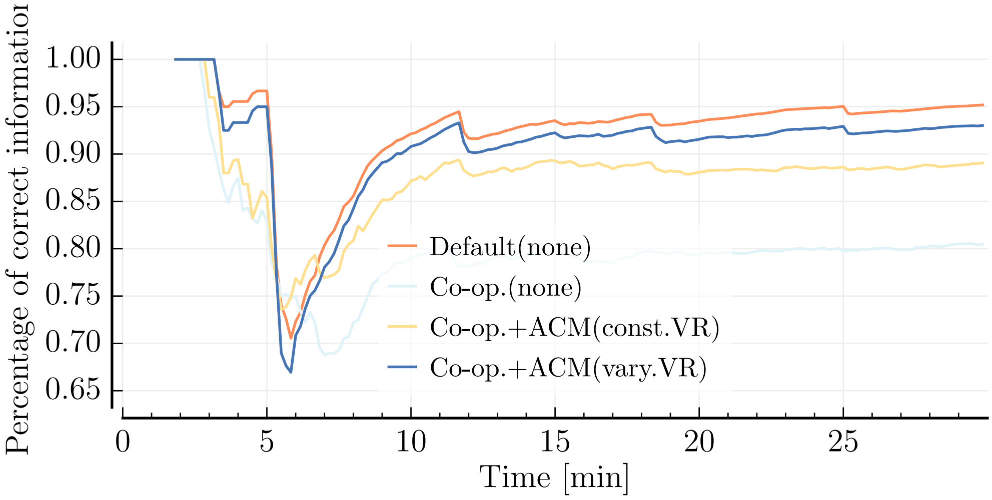


Figure 7.7: Accuracy achieved using QoI-based decision in 20% False Detection Rate Environment.

Vehicular rate in *Adaptive Cache Management* is used as a scale for adding virtual entries to compensate for the missing entries of those vehicles which did not publish. For the method *Co-op.+ACM(const.VR)*, a constant vehicular rate of 1 vehicle/second is considered. Although, this method gives better accuracy than *Co-op.(none)*, it is still not the efficient. This is because of the constant vehicular rate used for accounting the vehicles who did not publish. For adding virtual entries, every time this fixed rate is considered even if this is not the observed rate actually. This under-estimation or over-estimation of virtual entries leads to drop in accuracy. As observed, between 3^{th} and 5^{th} minute, the accuracy for this method drops below that of methods *Default(none)* and *Co-op.+ACM(vary.VR)* due to inaccurate vehicular rate estimation. This drop and the overall accuracy achieved throughout simulation signifies the necessity of accurate vehicular rate estimation for accuracy.

For the method *Co-op.+ACM(vary.VR)*, where accurate measurement of vehicular rate is calculated between successive publications, the overall accuracy achieved is better than that of the method *Co-op.+ACM(const.VR)*. However, even with accurate vehicular rate measurement, the accuracy is still lower compared to the method *Default(none)*. This is because of the following reasons. First, in *Co-op.+ACM(vary.VR)*, the accurate measurement of vehicular rate between second recent and most recent notification is sent only with the most recent publication. Until the most recent publication is made, the rate observed between the previous two publications is considered for adding virtual entries as discussed in Section 5.3. This means, there can be under-estimation or over-estimation of rate for the recent entry in cache until next notification is received. Second reason for the drop is an interdependent cause. If in case, an over-estimation of rate is being done for the recent entry which is actually a wrong notification, then the drop is further increased. Third, if the recent entry is a wrong notification and the next correct notification is received only after a long time, until then virtual entries for a wrong notification are being added with an over-estimation. This scenario would lead to a worst drop in accuracy. This can be regarded as the cost of a wrong notification, and subsequently cost of keeping a wrong notification until a next correct notification is received. Eventually, if accuracy is calculated in this window of time before receiving accurate rate from next correct publication, it performs slightly lower than *Default(none)*. However, both *Default(none)* and *Co-op.+ACM(vary.VR)* achieve an overall accuracy of well over 90% in a 20% false detection rate environment.

7.3.3 Accuracy over 0% to 20% False Detection Rates

The box-plot for accuracy of all methods across false detection rates is shown in Figure 7.8. As the false detection rate increases the performance of both *Co-op.(none)* and *Co-op.+ACM(const.VR)* drops poorly. The significance of *Adaptive Cache Management* is evident here with worst accuracy levels achieved for the method *Co-op.(none)* without virtual entries at 20% false detection rate. Likewise, importance of accurate vehicular rate measurement for adding virtual entries can be observed with overall performance achieved by *Co-op.+ACM(vary.VR)* method. Fixed rate of 1 vehicle/second in the method *Co-op.+ACM(const.VR)* either leads to under-estimation or over-estimation of actual vehicular rate and thus it performs below method *Co-op.+ACM(vary.VR)*.

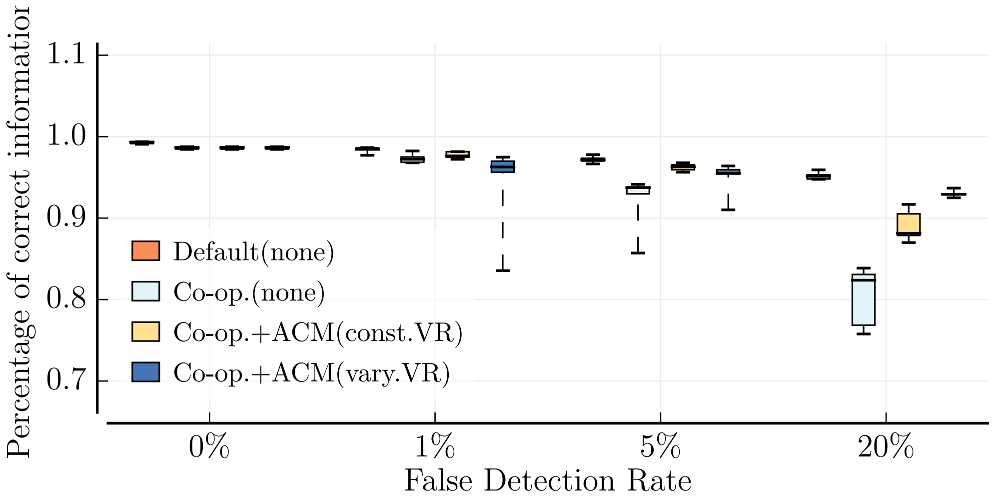


Figure 7.8: Accuracy achieved using QoI-based over a 0% to 20% False Detection Rate Environments.

For *Co-op.+ACM(const.VR)* method, at 1% false detection rate, the effect of whiskers on the confidence interval can be observed. Although, at this false detection rate, a wrong notification happens rarely, this variability can be for following reasons. A vehicle published wrongly and the next correct publication happened only after a long time. Until this time, the wrong publication was over-estimated by borrowing vehicular rate observed between last two publications leading to drop in accuracy.

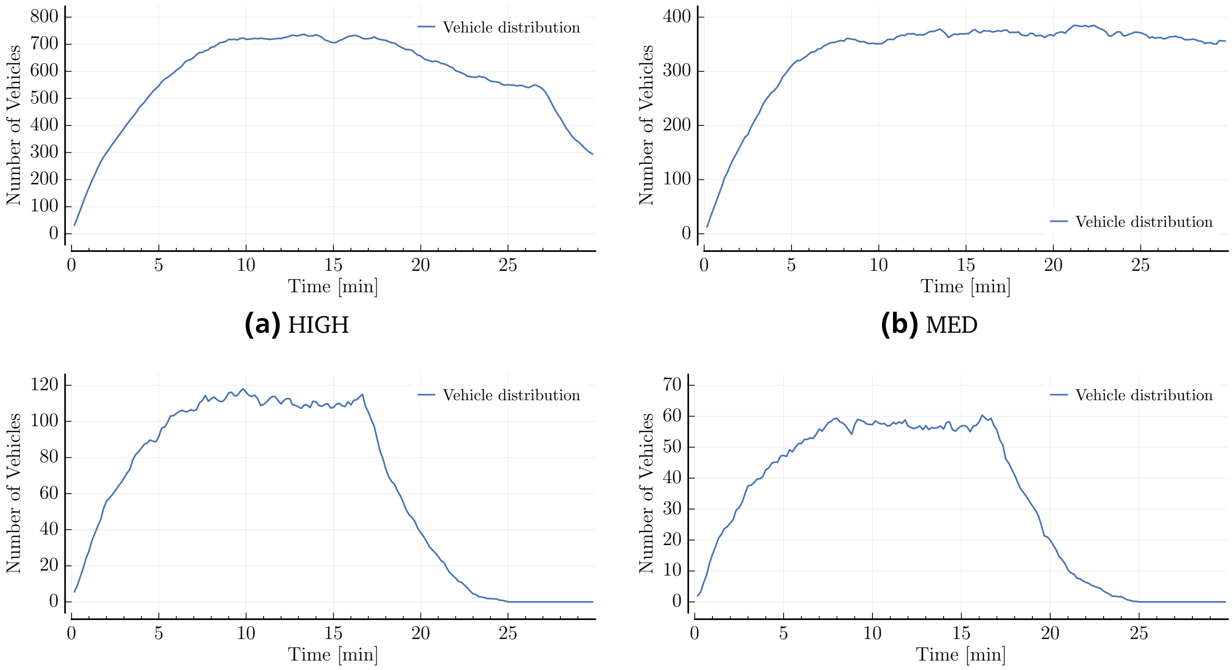
In a false detection environment, if a wrong notification is either over-estimated or not contradicted immediately by a correct notification, a decrease in accuracy is observed. The drop will be much further if over-estimation of this wrong notification happens for a long time because the next correctly publishing vehicles arrives only later. One can observe the consistent performance of *Co-op.+ACM(vary.VR)* falling just below the best achievable accuracy observed with *Default(none)* even at high false detection environment.

7.4 Impact of Vehicular Density on Transmission Costs and Accuracy

In order to evaluate the influence of vehicular density on the performance of methods, workloads offering different vehicle spawn rate are created. Simulations are carried out with these workloads to measure both transmission costs and accuracy achieved over different false detection rates.

7.4.1 Workloads and Vehicle Distribution

Four different workloads namely, `HIGH`, `MED`, `LOW`, and `VLOW` are created which correspondingly offer vehicular densities in the order of high, medium, low and very low. This configuration is set up as explained in Table 7.2, where by spawning vehicles at different rates, varying densities is obtained.



(a) HIGH **(b)** MED **(c)** LOW **(d)** VLOW

Figure 7.9: Vehicle distribution for different workloads.

According to fundamentals of traffic flow [MDM05], *Flow-density curve* gives the relationship between density and flow rate observed on a stretch of road in a given time interval. Some of its characteristics are discussed in the following. When there are no vehicles on the road, density will be zero and hence vehicular rate will be zero. As and when vehicles increase to appear on the road, density and rate will gradually increase only until the saturation point, referred to as maximum-flow point is reached. At this point onwards, an increase in traffic density would result in decrease of flow rate until it eventually becomes to zero leading to a complete halt of traffic. In our study, all the mentioned workloads vary with densities and thus flow rate is also varied as observed in Figure 7.9.

The workload `HIGH` has highest number of vehicles whereas `VLOW` has the least. In order to also incorporate the real-world scenario of night times with almost no vehicles, we configured in the workloads `LOW` and `VLOW`, after 17^{th} min, to have a drastic drop in number of vehicles. On the contrary, to reflect traffic variations observed during the day times, in the workloads `HIGH` and `MED`, a gradual decrease and continuous flow were correspondingly configured.

In this section we have chosen only two of the following combination of methods. One, *Default(none)* method, other *Co-op.+ACM(vary.VR)* method. From the previous evaluation section, it was evident that only these two methods perform better in terms of accuracy among all the combination of methods irrespective of false detection rates. Thus, evaluation on impact of vehicular density is carried for only these two best methods for both transmission costs and accuracy metrics.

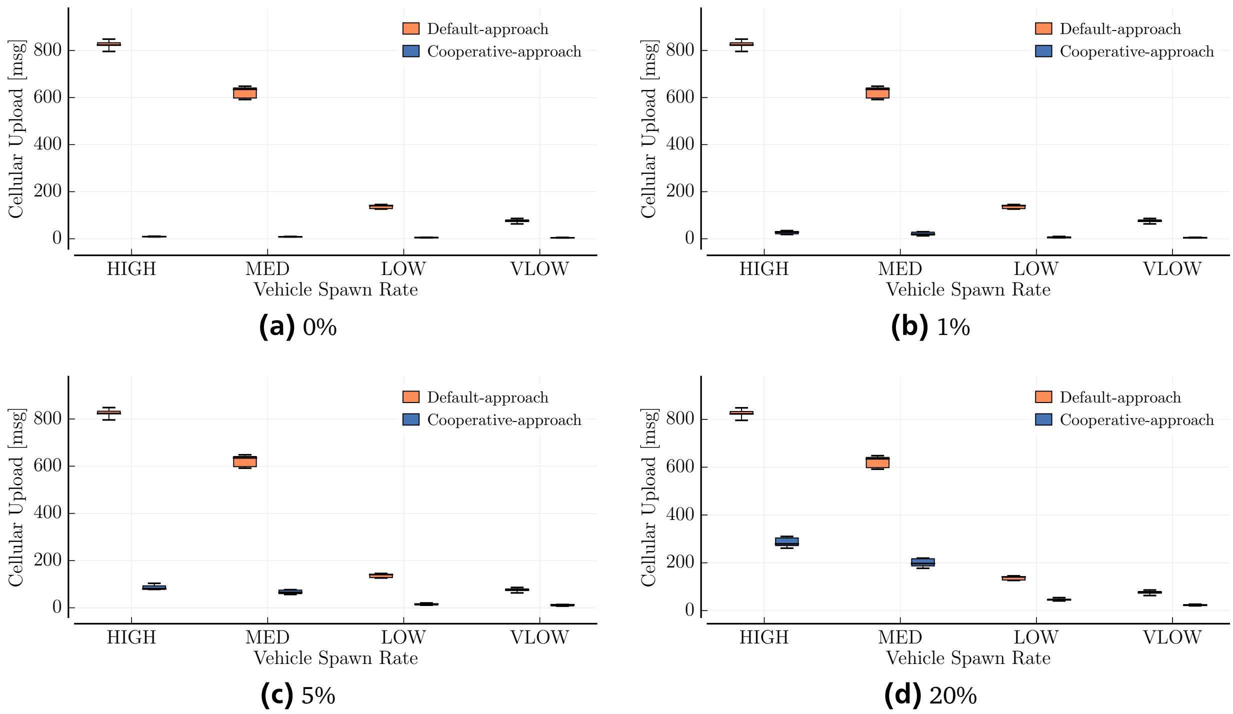
7.4.2 Transmission Costs

In the following subsection, we evaluate different workloads for their transmission costs that include both cellular upload and download costs.

Cellular Upload:
Figure 7.10 gives an overview of cellular upload costs associated for their corresponding workloads at different false detection rates. Cellular upload relates to the publication costs which depends on the reporting approach being followed.

As vehicular density decreases from `HIGH` till `VLOW`, it is observed that the number of publications for both methods also decrease. This behaviour is due to the reason that the number of vehicles travelling through POI is less in `VLOW` compared to `HIGH` workload. Thus, there are less vehicles on road now to report an event leading to infrequent reporting of traffic alerts though road state has changed.

The influence of false detections on number of publications associated with Cooperative-approach can also be observed with an increase in publications in 20% false detection environment compared to other lower false detection environments. Additionally, since vehicles following Default-approach simply publish irrespective of road condition and false detections, correspondingly the number of publications associated with this approach remain same across different false detection rates.

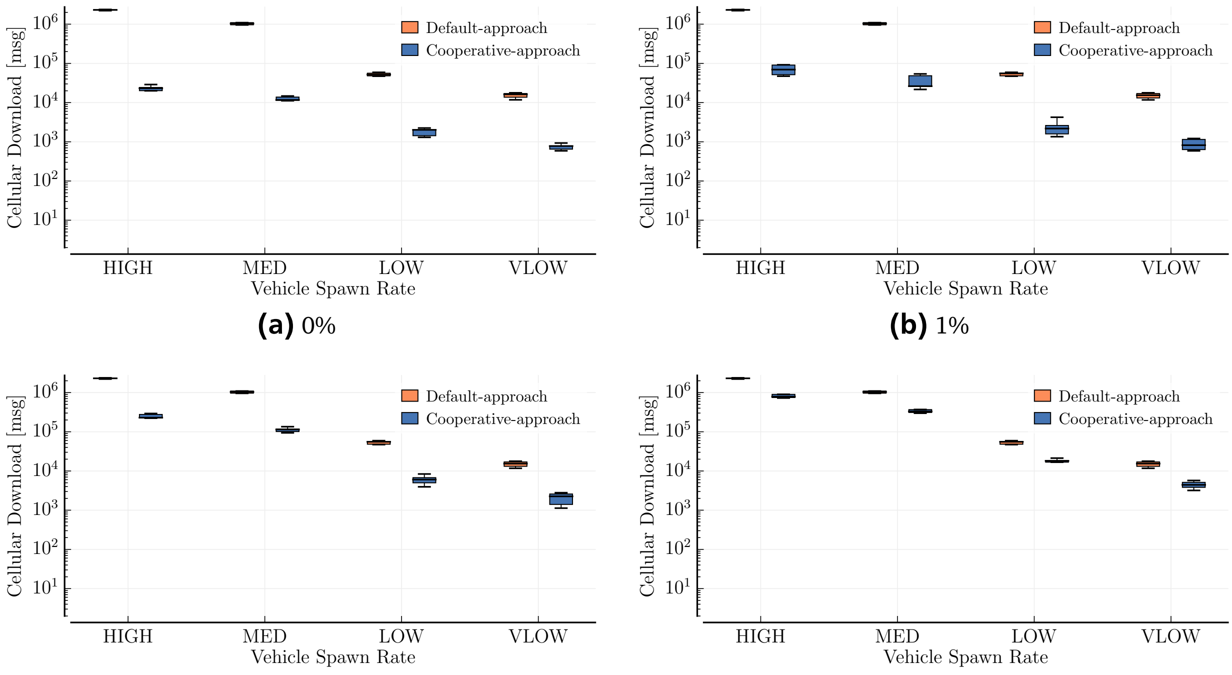


(a) 0% **(b)** 1% **(c)** 5% **(d)** 20%

Figure 7.10: Cellular Upload costs for each Workload over different False Detection Rate Environments.

Cellular Download:
Figure 7.11 gives an overview of cellular download costs associated for their corresponding workloads at different false detection rates. Cellular download messages relates to forwarding of notification by broker for every publication to all subscribers. This number is directly dependent on number of subscribed vehicles.

Similar to cellular upload costs, cellular download costs too decrease as vehicular density decreases from HIGH till VLOW. Less the number of subscribed vehicles on the road, lesser are the number of publication and consequently notifications to be forwarded by the broker. At lower densities the transmission costs borne by the broker is less compared to at higher densities. Alike cellular upload costs, the difference in cellular download costs between Default-approach and Cooperative-approach decreases with increase in false detection rates.



(a) 0% **(b)** 1% **(c)** 5% **(d)** 20%

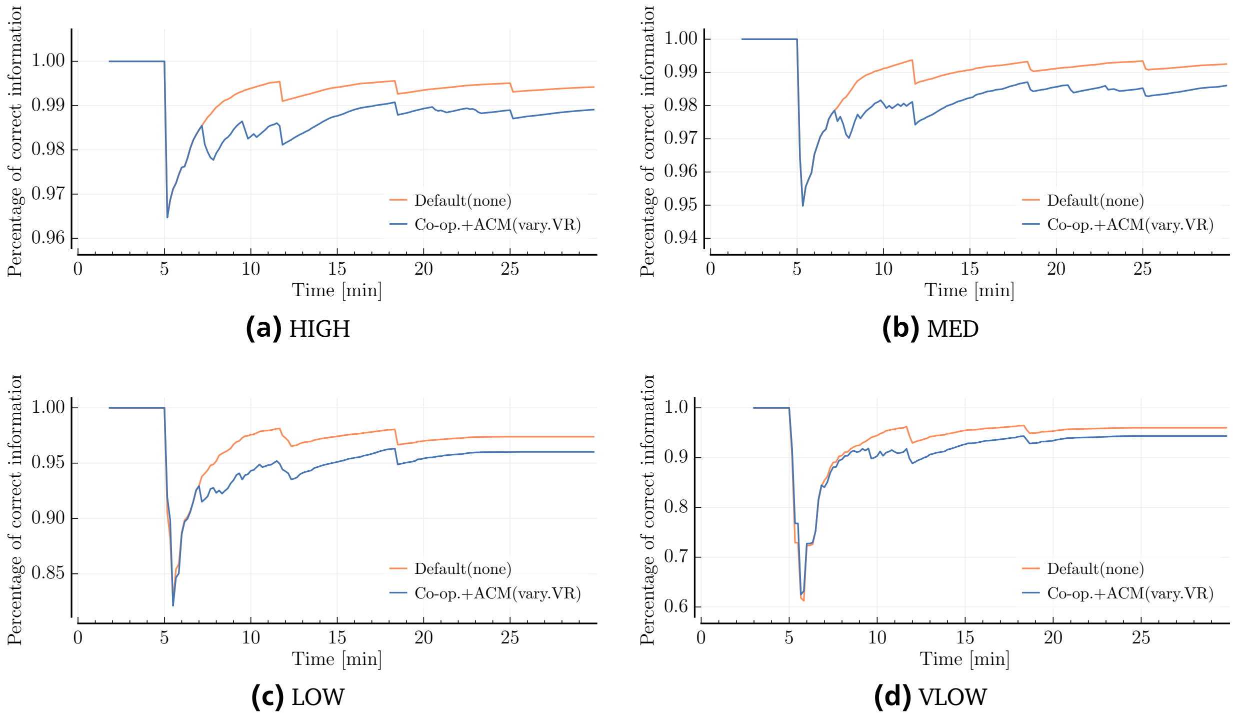
Figure 7.11: Cellular Download costs for each Workload over different False Detection Rate Environments.

In both the box plots for cellular upload and cellular download across different vehicular densities, the number of cellular messages for Cooperative-approach increases as the false detection rate increases. This is due to the combination of wrong publication and an immediate correct publication that happens based on availability of vehicles to negate this wrong publication.

7.4.3 Accuracy

In this subsection, performance of accuracy achieved over different workloads is presented. First, we highlight how the plots appear for different workloads at 0% false detection rate. Later, an overview across all the false detection rates is provided.

Accuracy at 0% False Detection Rate:
In Figure 7.12, overview of accuracy achieved over the simulation duration in a 0% false detection environment is provided. The overall accuracy achieved decreases from HIGH to VLOW workloads. Though, 0% false detection rate is considered, the workload with highest vehicular density achieves accuracy better than other workloads. This is because of continuous availability of vehicles to report change of road conditions that occurs at 5^{th}, 11.6^{th}, 18.3^{th}, and 25^{th} min and adapts quickly to achieve higher accuracy. This better performance is observed for both *Default(none)* and *Co-op.+ACM(vary.VR)* methods. For the method *Co-op.+ACM(vary.VR)*, overall accuracy in HIGH workload is around 99% whereas in VLOW workload it falls short below 95%.



(a) HIGH **(b)** MED **(c)** LOW **(d)** VLOW

Figure 7.12: Accuracy achieved using QoI-based decision in 0% False Detection Rate Environment for different workloads.

Another critical observation is adaptation of the methods to change of road conditions at different vehicular densities. Since this is a 0% false detection rate and thus all vehicles are correctly reporting a road situation, accuracy drops only at the times when road state changes. However, the depth of the drop observed in each workload is different. At 5^{th} minute, where road condition changes to a traffic-jam state, accuracy drops to 97% in workload `HIGH`, whereas in `VLOW` workload this drop is much further till 60%. This observation signifies the need of publishing vehicles to report new road states. In lower density vehicles as publishing vehicles appear infrequently at POI due to limited density, the publishing vehicle to report the new road state is appearing after long time compared to in higher density environment. In these conditions, more vehicles at AOI can make a wrong decision and travel towards a road which is in traffic jam state since they did not receive any alert.

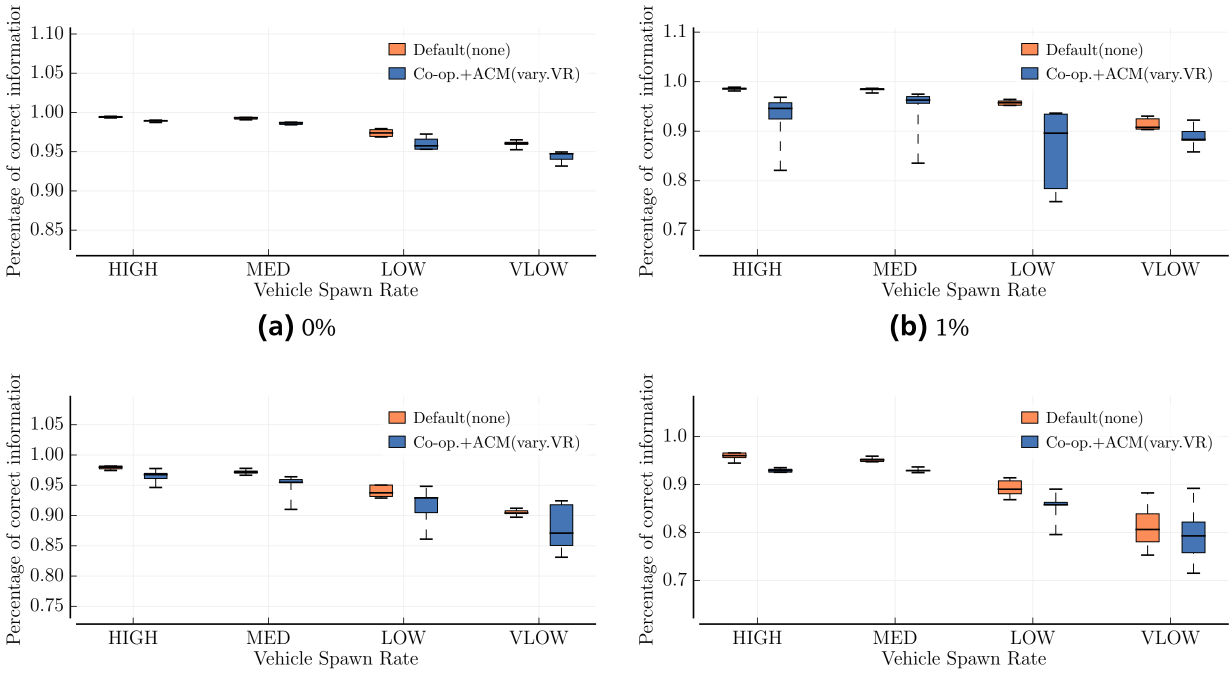
Another observation is that once the accuracy drops at 5^{th} minute, it gradually picks up in a short duration of time at higher density workloads than other workloads. This is because in lower density workloads as there are infrequent publications, the time required for all subscribed vehicles to be aware of a change of road state is high.

With respect to vehicular density in workloads `LOW` and `VLOW` in Figures 7.9c and 7.9d, the number of vehicles decreases rapidly after 18^{th} minute. This decrease in vehicles travelling on road has impact on accuracy. It can be observed in their corresponding Figures 7.12c and 7.12d for accuracy, that after 18^{th} min there are no variations in the plots though at 25^{th} min the road condition changes from no-traffic-jam to traffic-jam. As there are almost no vehicles when this event occurs, neither there is reporting nor drop in accuracy is seen as it should be. On the contrary, since there are still significant number of vehicles travelling, in Figures 7.12a and 7.12b for workloads `HIGH` and `MED`, the change of road condition is appropriately recorded with a drop in accuracy at 25^{th} min.

Accuracy over 0% to 20% False Detection Rate:

In Figure 7.13, box plot for accuracy across all workloads are given for different false detection rates. As vehicular density decreases the accuracy achieved by both methods decreases gradually. This signifies the necessity of continuous vehicles for reporting in a frequently changing road environment. At 20% false detection rate, in the workload `VLOW`, both methods achieve low accuracy levels due to the combination of both infrequent vehicles and wrong notifications.

Due to associated whiskers, the range for the method *Co-op+ACM(vary.VR)* is bigger at 1% false detection rate. Although, at this false detection rate, a wrong notification happens rarely, however, this variability outside the quartile interval can be for following reasons. A vehicle published wrongly and the next correct publication happened only after a long time. Until this time, the wrong publication was over-estimated by borrowing vehicular rate observed between last two publications leading to drop in accuracy.



(a) 0% **(b)** 1% **(c)** 5% **(d)** 20%

Figure 7.13: Accuracy achieved using QoI-based decision for each Workload over different False Detection Rate Environments.

To summarize, as vehicular density decreases, there are less number of vehicles to report a change of road condition. Due to this, the overall accuracy achieved decreases in lower density traffic. This also has impact on the recovery time required after a drop is observed due to change of road state. With availability of continuous vehicles in frequently changing environment, higher impact of correct notifications while lowering the impact of false notifications can be achieved. This helps in making the correct decision with higher probability. Additionally, information freshness can also be related to vehicular density. In high density traffic, adaptability to changes in road conditions is much faster and thus better information freshness is achieved in lesser duration.

7.5 Evaluation using Different Decision-Making Algorithms

In this section we evaluate different methods described in Table 7.3 for accuracy using different decision-making algorithms. Until now, evaluation for accuracy was using QoI-based algorithm. This is because of the weighted majority considered by QoI-based that accounts the age of the notification as well. This feature is beneficial in vehicular environments such as our scenario where temporal dynamics of an event is significant for quicker adaptability to new road conditions.

A Majority based decision-making algorithm applies simple majority based resolution. Since in our scenario the road condition is represented in a binary value either with a traffic-jam or a no-traffic-jam state, majority based decision-making can be employed to make a rerouting decision. On the contrary, with the Newest-based decision-making, a vehicle always makes rerouting decision based on the latest value in its cache. In this case, all other cache entries except recent entry are irrelevant.

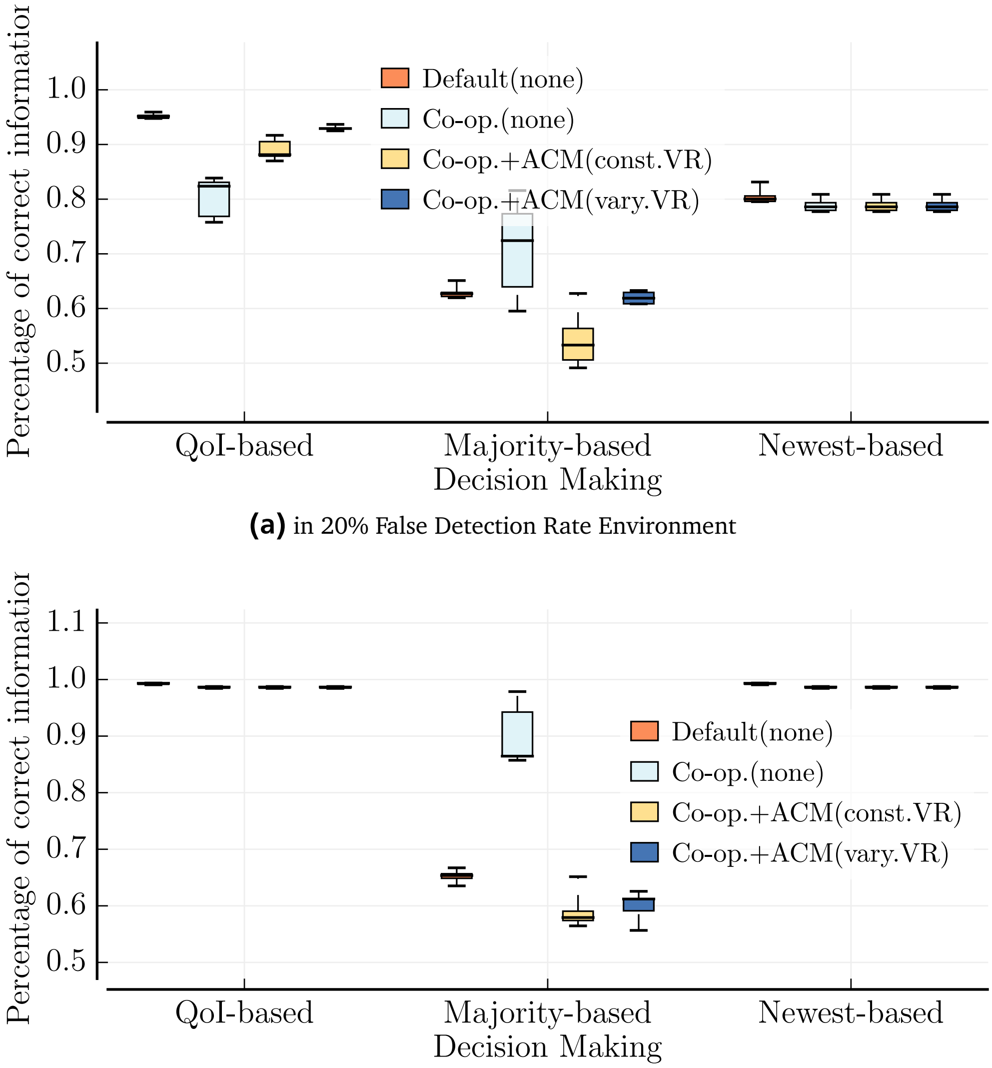
Accuracy at 20% False Detection Rate:
Figure 7.14a gives an overview of accuracy achieved using different decision-making algorithms in a 20% false detection environment. Considering the dynamic nature of vehicular environment with frequent changes in traffic behaviour, it can be observed that QoI-based decision algorithm performs better than other algorithms. As discussed in previous evaluations using QoI-based decision-making, the methods *Default(none)* and *Co-op.+ACM(vary.VR)* performs better than the methods *Co-op.(none)* and *Co-op.+ACM(const.VR)*. It can be noted that besides poor detection accuracies, method *Co-op.+ACM(vary.VR)* consistently achieves accuracy almost as close as the method *Default(none)* which achieves best accuracy levels due to redundant publications.

With respect to Newest-based algorithm, since decision is based only on the latest entry in cache, all the methods involving Cooperative-approach of reporting perform the same. However, there is a slight degradation in accuracy achieved by methods involving Cooperative-approach when compared with Default-approach. This slight degradation in performance can be related to the cost of unaware-vehicles. As unaware-vehicles which have not received any notification of the existing traffic-jam state, only upon their arrival at POI they decide to publish and update their cache. In comparison to QoI-based algorithm, Newest-based performs lower due to slow adaptation to changing road status in false detection environment.

In contrast to QoI-based and Newest-based algorithms, Majority-based algorithms achieve the least accuracy. It is lower than other two algorithms as neither does it regard the age of the information as QoI-based, nor does it consider the newest information which can give the recent road status. Instead, Majority-based is dependent on simple majority of values for decision-making which relies also on outdated information. Due to this characteristic, method *Co-op.(none)* though without accounting vehicles which did not publish and henceforth has the least cache entries among all, performs better than the method *Default(none)* which actually accounts all the publishing vehicles. However, in comparison between the methods *Default(none)* and *Co-op.+ACM(vary.VR)*, it can be noted, they both achieve almost the same performance in Majority-based decision-making.

Accuracy at 0% False Detection Rate:
Figure 7.14b gives an overview of accuracy achieved using different decision-making algorithms in a 0% false detection environment. Without any false detections and false reporting, QoI-based and Newest-based algorithm achieve the same performance levels across all methods. This is because all the changes of road state are correctly reported without any false notifications. This is also the reason for observing no difference in performance for all the methods that follow Cooperative-approach.

It is evident that irrespective of any decision-making algorithm at any false detection rate, *Co-op.+ACM(const.VR)* performs badly. This is due to considering a constant vehicular rate of 1 vehicle/second leading to either under-estimation or over-estimation of virtual entries. Leaving out this method, in comparison between *Co-op.(none)* and *Co-op.+ACM(vary.VR)*, one can choose *Co-op.+ACM(vary.VR)* method for higher accuracy across all algorithms except Majority-based. Though for Majority-based at



(a) in 20% False Detection Rate Environment

(b) in 0% False Detection Rate Environment

Figure 7.14: Accuracy achieved using different decision-making algorithms.

0% false detection rate there is significant difference in accuracy between these two methods, but it is very trivial in a 20% false detection rate. Due to this, choosing *Co-op.+ACM(vary.VR)* method with virtual entries over the method *Co-op.(none)* without virtual entries is advantageous as it gives better accuracy in other decision algorithms across low and high false detection rates.

Choosing between the methods *Default(none)* and *Co-op.+ACM(vary.VR)* is a trade off decision. This is because both these methods achieve the best possible accuracy across any false detection rates over other methods. Only that *Default(none)* performs slightly better than *Co-op.+ACM(vary.VR)*. However, in comparison to the load exerted on broker by Default-approach and Cooperative-approach, the method *Co-op.+ACM(vary.VR)* is clearly most suitable in high density vehicular networks that keeps transmission costs minimal.

Given the significantly low number of cellular upload and download costs associated with Cooperative-approach (Figure 7.5) and the accuracy levels achieved with *Co-op.+ACM(vary.VR)* across varying false detection rates, which is just as good as *Default(none)*, the method *Co-op.+ACM(vary.VR)* is clearly best suited for high density vehicular environments where road conditions change often.

False decisions in our study happens due to two reasons. One is due to false reporting by vehicles which detect a road situation wrongly. Another reason is due to slow adaptations to new road states. In our simulation results, it is observed that Newest-based decision algorithm performs well whenever a change in road condition occurs. However, the same algorithm does poorly to recover from a wrong notification in high false detection rates. Therefore, Newest-based algorithm adapts quickly to changes but is not robust to false detections, and accordingly is best suited in an environment where all vehicles report correctly. Majortiy-based algorithm adapts slowly to changes and is weak to false detections as well. On the contrary, QoI-based algorithm, adapts quickly to changes as well as recovers fast besides wrong notifications. Since vehicular environment is characterized by frequent changes in road conditions and can include falsely reporting vehicles, QoI-based algorithm is reliable and preferred for decision-making.

7.6 Proof-of-Concept to Minimize Influence of Unaware-Vehicles

Unaware-vehicles are those vehicles that are unaware of the prevailing traffic-jam as they were not existing when the publication corresponding to the traffic-jam was published. These vehicles can be observed predominantly in the methods that employ *Cooperative-approach* since it entails less publications avoiding redundancy. Unaware-vehicles have not received any notification since the time they are observed in the system and have only default cache value representing a no-traffic-jam. Therefore, upon their arrival at POI, they detect an existing traffic-jam that has occurred recently as a new road event and publish it exclusively.

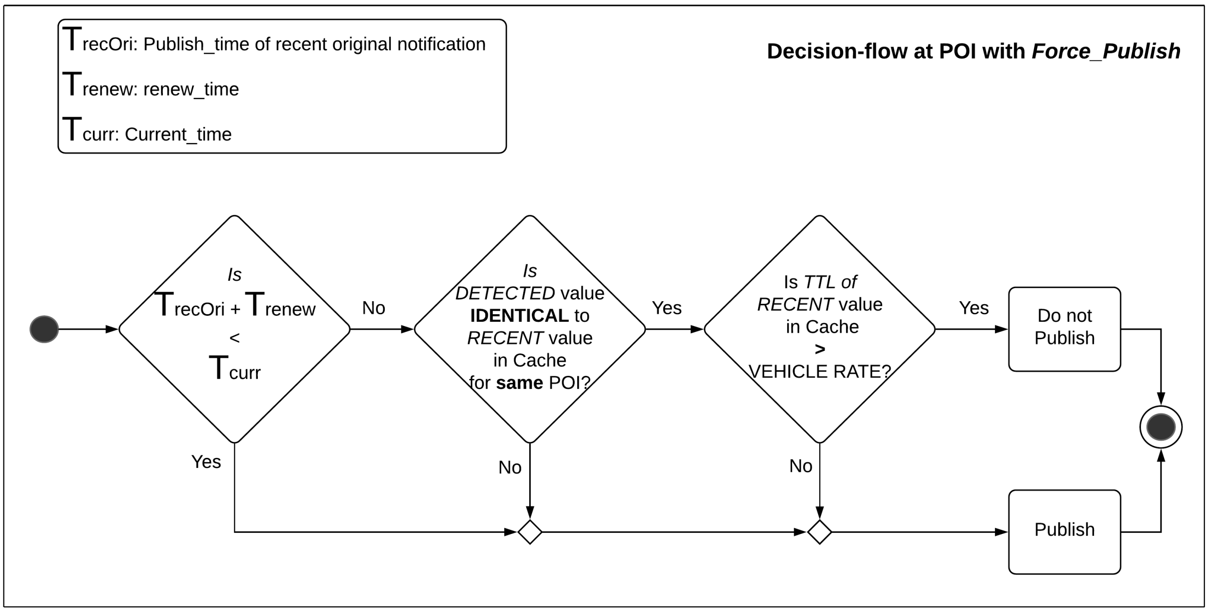


Figure 7.15: *Cooperative-approach* with *Force-Publish*.

It was observed in the Figure 7.3b, due to unaware-vehicles occurring at times 7^{th}, 21^{th}, and 23^{th} minute, publications were recorded although there was no change of road conditions at these times. Furthermore, drop in accuracy were observed correspondingly at these times in Figure 7.6. Additionally, in Figure 7.14b, though all vehicles reported correctly in a 0% false detection environment, however, due to the effect of unaware-vehicles, both in QoI-based and Newest-based algorithms, *Cooperative-based* methods performed slightly lower than the method *Default(none)*.

To address this issue of dropping in accuracy due to unaware-vehicles, we propose *Force-Publish*, compelling a vehicle to publish even though there is no change of road condition. To mitigate unnecessary transmission costs that would arise out of frequent republishing and at the same time to avoid effect of unaware-vehicle due to infrequent republishing, we apply a variable *renew_time*.

Renew_time is used to adjust the frequency of republishing. As seen in Figure 7.15, whenever the summation of *renew_time* (marked as $\boldsymbol{T_{renew}}$) and published time of recent original notification (marked as $\boldsymbol{T_{recOri}}$) is less than current time (marked as $\boldsymbol{T_{curr}}$), then publishing happens impulsively irrespective of change in road status so that vehicles become aware of the existing road situation.

In vehicular environment with high density traffic, value of *renew_time* can be increased to avoid frequent republishing, whereas in low density traffic, this value needs to be lowered to accommodate for the already infrequent flow of vehicles.

In our simulation to evaluate *Force-Publish* in a 0% false detection environment, we have used `MED` workload which entitles medium traffic density, and hence correspondingly, value of *renew_time* is set to 1 minute.

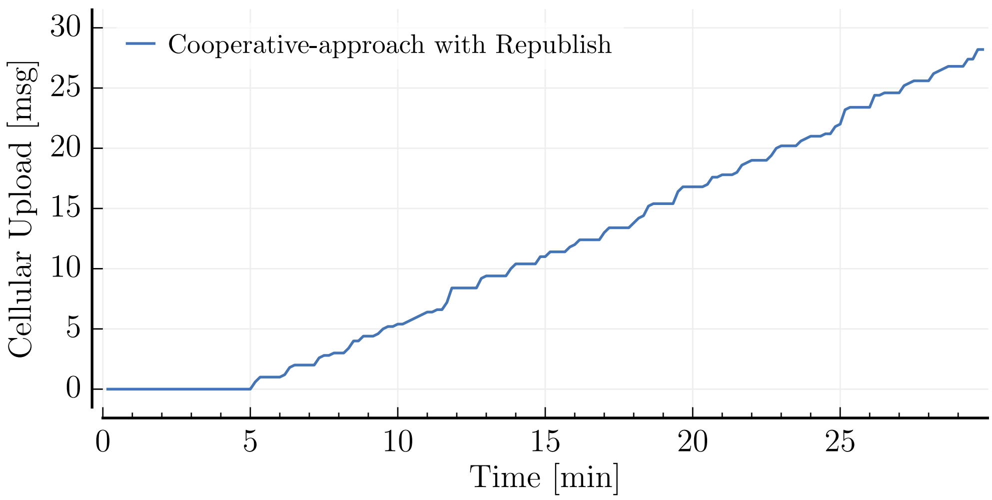


Figure 7.16: Cellular Upload messages in 0% False Detection Rate Environment using *Cooperative-approach* with *Force-Publish*.

To highlight the impact of *Force-Publish*, we carried out simulation with *Force-Publish* employed only on the method *Co-op.+ACM(vary.VR)-Republish*. Whereas the methods *Co-op.+ACM(const.VR)* and *Co-op.(none)* do not employ *Force-Publish* which lets the influence unaware-vehicles to be seen.

As observed in Figure7.16, now there are frequent publications happening due to *Force-Publish*. In comparison to Figure 7.3b which involved unaware-vehicles, number of publications have increased from 9 to 29. However, its positive impact on accuracy can be observed in Figure 7.17 with an increase of accuracy achieved for the method *Co-op.+ACM(vary.VR)-Republish*.

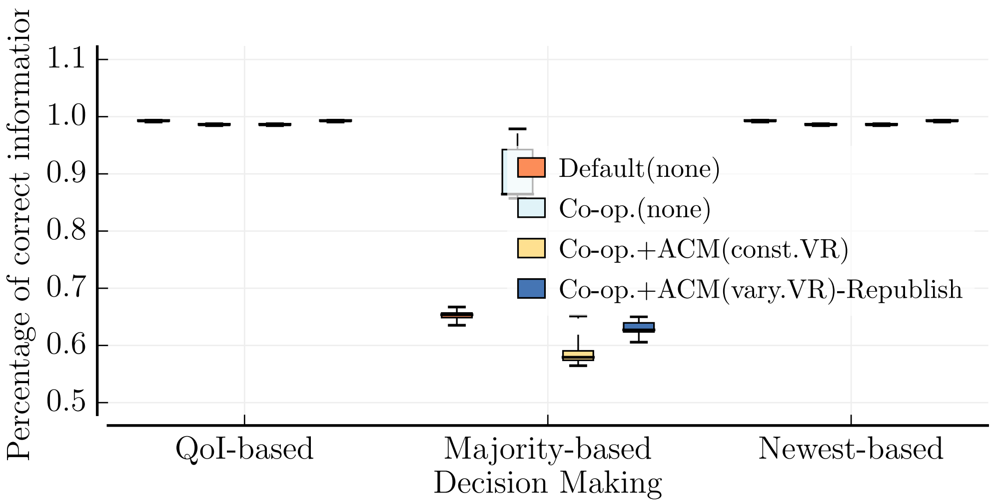


Figure 7.17: Accuracy achieved using *Cooperative-approach* with *Force-Publish* in 0% False Detection Rate Environment across different decision-making algorithms.

With *Force-Publish* feature, the method *Co-op.+ACM(vary.VR)-Republish* achieves better accuracy than other two methods following Cooperative-approach. The accuracy achieved is exactly equivalent to what the method *Default(none)* achieves but at significantly lower publication costs. Since this is a 0% false detection rate, this increased accuracy level can be seen in both QoI-based and Newest-based algorithms. Additionally, although trivial, one can observe increase in accuracy with *Force-Publish* also for Majority-based algorithm in comparison to Figure 7.14b.

8 Conclusion and Outlook

Traffic alerts in vehicular applications reporting road status and events to vehicles likely to be influenced establish an efficient road traffic management service. Thereby a data dissemination system for real-time reporting of events that which is scalable is required. In this thesis, a Publish/Subscribe event dissemination using cellular communication was employed. An attribute based filtering scheme that enhances flexible filtering of topics was utilized.

To avoid excessive load on broker caused by redundant publications in *Default-approach*, *Cooperative-approach* was proposed. A vehicle compares its detected value with recent cache value before publishing. If the comparison yields same value then the vehicle refrains from publishing and thus a redundant publication is prohibited. On the contrary, if the comparison yields a different value which is due to a change of road state, the vehicle publishes this new state thus maintaining the information freshness as expected.

The insufficient accuracy for decision-making process due to skipping of redundant publications in *Cooperative-approach* was a challenge. This was depicted in the evaluation of the method *Co-op.(none)*, which achieved the worst accuracy signifying the necessity to account those vehicles which did not actually publish. To address this challenge, *Adaptive Cache Management* using virtual entries based on vehicular rate was proposed. Using vehicular rate observed during publishing, vehicles which did not publish to avoid redundant publications were also accounted to achieve better accuracy for decision-making.

To realize the importance of accurate vehicular rate measurement, evaluation using a constant vehicular rate of 1 vehicle/second was carried out as represented by the method *Co-op.+ACM(const.VR)*. Although, this method did perform better than the method without virtual entries, however it suffered from under or over estimation of vehicular rate than it actually was. This was overcome by accurate vehicular rate measurement using global knowledge.

The method *Co-op.+ACM(vary.VR)* adapts to varying vehicular rate observed in real-world and thus achieving better accuracy than other two discussed methods. To match real-world use case of false detections and false reporting by vehicles, evaluations were carried at different false detection rates. Our proposed method *Co-op.+ACM(vary.VR)*, achieved better accuracy over other methods for decision-making even at high false detection rate.

The method *Co-op.+ACM(vary.VR)*, at different false detection rates, achieved accuracy levels almost as same as the method *Default(none)* which is the best possible due to publication by all vehicles. Moreover, this performance was achieved at a significantly lower transmission costs compared to the method *Default(none)* which had redundant publications leading to excessive load on broker.

In our proposal following *Cooperative-approach* and *Adaptive Cache Management*, an additional payload cost of 4 bytes (float value) with each publication is involved to accommodate vehicular rate as a publication attribute. However, in light of high transmission costs borne by cellular network due to redundant publications, it is rather efficient to comprehend this small payload cost that is associated only with less number of publications that do happen.

The influence of vehicular density was also studied. Using vehicular densities ranging from very low to high levels, accuracy achieved in each case was examined. In high density road environments better accuracy levels were achieved. This is because of the necessity of publishing vehicles to report and reflect a newly changed road condition. It was also observed that in very high false detection rates scenario, accuracy achieved were better in high density road environments. This was for the reason of availability of vehicles to negate the effect of a wrong publication by a correct publication in decision-making.

False decision happens for a combination of false detections and slow adaptations to new road conditions. Therefore, for evaluation, QoI-based algorithm that considers both freshness and accuracy for information validation in vehicular networks was applied. Vehicles using QoI-based algorithms achieved best accuracy levels by adapting fast to new road conditions and recovering quickly in case of false detections across varying vehicular densities.

Our work demonstrates that information quality can be achieved to moderate levels that is enough to make efficient routing decisions while sustaining network performance. It highlights that large amounts of unnecessary transmissions may be removed without affecting accuracy required for decision-making when combination of our two contributions namely *Cooperative-approach* and *Adaptive Cache Management* using varying vehicular rate is applied. This method is well suited for traffic management in high density road networks similar to Urban and Expressway-interchange environments where traffic jams are frequently reported and rerouting options are available.

8.1 Outlook

In this section, based on outcome from this thesis work, possible directions for future-work are provided. Each of the following topics discussed can also be investigated in combinations to explore and extend this study further.

8.1.1 Local Dissemination using VANET

In our work, using *Cooperative-approach* redundant publications are avoided. This reduces significantly number of messages transmitted between the publishing vehicles and the broker. However, based on number of subscribed vehicles, the broker has to forward notifications still to each one of them. This process is done by broker for every publication that is issued leading to high transmission costs.

An alternative efficient approach for alerting subscribed vehicles for each publication would be using a Hybrid communication system proposed in '*Hybrid Communication in Vehicular Networks*[1]'. According to this work, dissemination system is adaptive to choose either a long distance cellular communication or VANET based local dissemination considering vehicular density and the required offloading from cellular network. In case of local dissemination, notification is sent by broker to only one of the vehicles in ad hoc network and that vehicle assumes the role of broker for local dissemination. To investigate furthermore reduction in transmission costs while achieving better accuracy, implementation of *Cooperative-approach* and *Adaptive Cache Management* in *Hybrid Communication in Vehicular Networks* can be studied.

8.1.2 Sparse Vehicular Networks

With our approach, as vehicular density decreases, the accuracy achieved also decreases. This is due to infrequent vehicles to report new road conditions and late arrival of correctly reporting vehicles to negate a wrong publication that was sent previously.

Although, our approach does not fail to perform in a sparse vehicular network, however, with fewer vehicles it would be efficient to let all vehicles publish. Given the trade-off between transmission cost and computation required in accurate measurement of vehicular rate for virtual entries, every vehicle can follow *Default-approach* of reporting. This is also supported by the fact of rare occurrence of traffic in sparse networks such as rural roads, where cellular network can bear the associated load without much burden. This approach is furthermore efficient if the sparse vehicular network involves vehicles with higher false detection rates. This is because in sparse networks if a vehicle wrongly notifies, by following *Adaptive Cache Management*, virtual entries for this wrong notification would be added either until next correctly publishing vehicle arrives or until the TTL of that notification expires. Any vehicle making decision in this window of time is bound to make a wrong decision.

[1] Master-Thesis Title: Hybrid Communication in Vehicular Networks: Combining the Advantages of Cellular and V2V Communication, Author: Sophie Schönherr, KOM-M-0599, KOM - TU Darmstadt. Date: 01. December 2017

To avoid the influence of unaware-vehicles, we proposed *Force-Publish* using *renew_time*. However, this variable that signifies the retransmission interval is dependent on vehicular density. A lower value leads to frequent retransmissions and a higher value allows appearance of unaware-vehicles. It is ideal to retransmit when there is a significant rate of change of vehicular rate and hence, an investigation in this direction can be carried out to analyse the trade off between transmission costs and accuracy.

In our study, we have considered a push based Publish/Subscribe mechanism for event reporting. However, in a sparse network, pull based mechanism can be investigated. Due to fewer vehicles and correspondingly a subscribed vehicle might have not received notification as it was not in cellular range during a publication, this vehicle might make a wrong decision. To overcome this, road status can be stored in broker and an interested vehicle which arrives no matter how late, can request the broker for making a decision. Any outdated information at the broker will be invalidated using TTL as it was done earlier and hence consistency is also maintained.

8.1.3 Adaptive TTL

The significance of TTL associated with every publication is crucial for decision-making. QoI-based algorithm uses weighted majority decision based on age of the information. TTL associated with publications is used to determine age of each publication and due to which QoI-based algorithms exhibits better information freshness. Additionally, any outdated cache entries with expired TTL are removed automatically for cache management as they are insignificant in decision-making.

Combined with these advantages of TTL, an adaptive TTL can be realized based on vehicular density. Whenever there is significant change of vehicular density that effects vehicular rate, an increase or decrease in TTL for similar events can install better information freshness. Assume a scenario, where a vehicle publishes wrong notification and the next vehicle publishing correctly is only after a long time. In this window, vehicles believe road status reported by previous wrong publication as correct either until the time a correct publication happens or till the time it is removed from cache which happens only upon expiry of its TTL. To avoid keeping wrong information in cache for long time, adaptive TTL based on vehicular density can be examined. This would also be beneficial in sparse networks where besides fewer vehicles, the next correctly publishing vehicle might arrive after long time.

Another interesting investigation can be estimating TTL with regard to duration of a road situation. If road status changes frequently, then continuous flow of vehicles is needed both to report these changes and for faster adaptation to newer road conditions. In this case, if shorter TTL is employed, then more publications will be sent frequently to report conditions. On the contrary, if longer TTL is employed, then it's effect on decision-making is not removed until its expiry and a new event would have occurred within this time. Based on historical observation of congestion, traffic-jam occurring on a road combined with traffic behaviour during rush hours, TTL can be adapted to achieve best possible information freshness.

8.1.4 Other Notification Topics

For our work, a single stretch of stop-and-go traffic was considered, however multiple jams on multiple road edges along the route to destination must be investigated. This can also be in the direction of combining historical data consisting average life time of traffic jam observed on a road edge during rush hours. Other road users' services involving topics such as gas stations, service stations, restaurants can be combined with traffic services to enhance road users' travelling experience.

Bibliography

[AAH15] Ibrahim Abdullahi, Suki Arif, and Suhaidi Hassan. *Ubiquitous Shift with Information Centric Network Caching Using Fog Computing*, pages 327–335. Springer International Publishing, Cham, 2015.

[ABK+17] S. H. Ahmed, S. H. Bouk, D. Kim, D. B. Rawat, and H. Song. Named data networking for software defined vehicular networks. *IEEE Communications Magazine*, 55(8):60–66, 2017.

[ACM12a] Marica Amadeo, Claudia Campolo, and Antonella Molinaro. Content-centric networking: Is that a solution for upcoming vehicular networks? In *Proceedings of the Ninth ACM International Workshop on Vehicular Inter-networking, Systems, and Applications*, VANET '12, pages 99–102, New York, NY, USA, 2012. ACM.

[ACM12b] Marica Amadeo, Claudia Campolo, and Antonella Molinaro. Enhancing ieee 802.11p/wave to provide infotainment applications in vanets. *Ad Hoc Netw.*, 10(2):253–269, March 2012.

[AL14] AM Abdelgader and Wu Lenan. The physical layer of the ieee 802.11 p wave communication standard: the specifications and challenges. In *Proceedings of the World Congress on Engineering and Computer Science*, volume 2, pages 22–24, 2014.

[ASM+08] H. Artail, H. Safa, K. Mershad, Z. Abou-Atme, and N. Sulieman. Coacs: A cooperative and adaptive caching system for manets. *IEEE Transactions on Mobile Computing*, 7(8):961–977, Aug 2008.

[BCG+14] F. Bruno, M. Cesana, M. Gerla, G. Mauri, and G. Verticale. Optimal content placement in icn vehicular networks. In *2014 International Conference and Workshop on the Network of the Future (NOF)*, pages 1–5, Dec 2014.

[BG13] BE Bilgin and VC Gungor. Performance comparison of ieee 802.11 p and ieee 802.11 b for vehicle-to-vehicle communications in highway, rural, and urban areas. *International Journal of Vehicular Technology*, 2013, 2013.

[BSB+14] Nabil Benamar, Kamal D. Singh, Maria Benamar, Driss El Ouadghiri, and Jean-Marie Bonnin. Routing protocols in vehicular delay tolerant networks: A comprehensive survey. *Computer Communications*, 48:141 – 158, 2014. Opportunistic networks.

[Cao02] Guohong Cao. Proactive power-aware cache management for mobile computing systems. *IEEE Transactions on Computers*, 51(6):608–621, Jun 2002.

[Cao03] Guohong Cao. A scalable low-latency cache invalidation strategy for mobile environments. *IEEE Trans. on Knowl. and Data Eng.*, 15(5):1251–1265, September 2003.

[CDF+98] Ramón Cáceres, Fred Douglis, Anja Feldmann, Gideon Glass, and Michael Rabinovich. Web proxy caching: The devil is in the details. *SIGMETRICS Perform. Eval. Rev.*, 26(3):11–15, December 1998.

[CDN+96] Anawat Chankhunthod, Peter B. Danzig, Chuck Neerdaels, Michael F. Schwartz, and Kurt J. Worrell. A hierarchical internet object cache. In *Proceedings of the 1996 Annual Conference on USENIX Annual Technical Conference*, ATEC '96, pages 13–13, Berkeley, CA, USA, 1996. USENIX Association.

[CSLH10] G. Y. Chang, J. P. Sheu, T. Y. Lin, and K. Y. Hsieh. Cache-based routing for vehicular ad hoc networks in city environments. In *2010 IEEE Wireless Communication and Networking Conference*, pages 1–6, April 2010.

[CXS06] Hui Chen, Yang Xiao, and Xuemin Shen. Update-based cache access and replacement in wireless data access. *IEEE Transactions on Mobile Computing*, 5(12):1734–1748, Dec 2006.

[CZCX07] J. Cao, Y. Zhang, G. Cao, and L. Xie. Data consistency for cooperative caching in mobile environments. *Computer*, 40(4):60–66, April 2007.

[Den07] Mieso K. Denko. Cooperative data caching and prefetching in wireless ad hoc networks. *IJBDCN*, 3(1):1–15, 2007.

[DFNI07] M. D. Dikaiakos, A. Florides, T. Nadeem, and L. Iftode. Location-aware services over vehicular ad-hoc networks using car-to-car communication. *IEEE Journal on Selected Areas in Communications*, 25(8):1590–1602, Oct 2007.

[DINI05] Marios D. Dikaiakos, Saif Iqbal, Tamer Nadeem, and Liviu Iftode. Vitp: An information transfer protocol for vehicular computing. In *Proceedings of the 2Nd ACM International Workshop on Vehicular Ad Hoc Networks*, VANET '05, pages 30–39, New York, NY, USA, 2005. ACM.

[DLR18] Deutsches zentrum für luft- und raumfahrt - dlr. sumo - simulation of urban mobility. `www.dlr.de/ts/sumo`, 2018. Accessed on: 01.02.2018.

[dNGCdLCV17] Elenilson da Nóbrega Gomes, Carlos Alberto V. Campos, Sidney C. de Lucena, and Aline Carneiro Viana. A message removal mechanism for delay tolerant networks. In Rachid El-Azouzi, Daniel Sadoc Menasche, Essaïd Sabir, Francesco De Pellegrini, and Mustapha Benjillali, editors, *Advances in Ubiquitous Networking 2*, pages 43–55, Singapore, 2017. Springer Singapore.

[dSCC16] Vitor Borges Coutinho da Silva, Miguel Elias M. Campista, and Luís Henrique M. K. Costa. Trac: A trajectory-aware content distribution strategy for vehicular networks. *Vehicular Communications*, 5(Supplement C):18 – 34, 2016.

[FA13] K. Fawaz and H. Artail. Dcim: Distributed cache invalidation method for maintaining cache consistency in wireless mobile networks. *IEEE Transactions on Mobile Computing*, 12(4):680–693, April 2013.

[FAAK+15] Kassem Fawaz, Abdalla Artail, Rasha Al-Khansa, Hassan Artail, and Haidar Safa. Replication enabled distributed cache invalidation method: Replication enabled distributed cache management system for wireless mobile networks. *Wirel. Commun. Mob. Comput.*, 15(13):1711–1728, September 2015.

[FCC05] Marco Fiore, Claudio Casetti, and Carla-Fabiana Chiasserini. On-demand content delivery in vehicular wireless networks. In *Proceedings of the 8th ACM International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, MSWiM '05, pages 87–94, New York, NY, USA, 2005. ACM.

[FK11] N Sabiyath Fatima and Dr P Sheik Abdul Khader. A hybrid cache invalidation technique for data consistency in manet. *International Journal of Computer Applications*, 16(5):p40–44, 2011.

[FMCC09] M. Fiore, F. Mininni, C. Casetti, and C. F. Chiasserini. To cache or not to cache? In *IEEE INFOCOM 2009*, pages 235–243, April 2009.

[GLL17] X. Ge, Z. Li, and S. Li. 5g software defined vehicular networks. *IEEE Communications Magazine*, 55(7):87–93, 2017.

[GPP+14] G. Grassi, D. Pesavento, G. Pau, R. Vuyyuru, R. Wakikawa, and L. Zhang. Vanet via named data networking. In *2014 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pages 410–415, April 2014.

[GPW+13] Giulio Grassi, Davide Pesavento, Lucas Wang, Giovanni Pau, Rama Vuyyuru, Ryuji Wakikawa, and Lixia Zhang. Acm hotmobile 2013 poster: Vehicular inter-networking via named data. *SIGMOBILE Mob. Comput. Commun. Rev.*, 17(3):23–24, November 2013.

[HL98] Qinglong Hu and Dik Lun Lee. Cache algorithms based on adaptive invalidation reports for mobile environments. *Cluster Computing*, 1(1):39–50, May 1998.

[HMF14] Zeeshan Hameed Mir and Fethi Filali. Lte and ieee 802.11p for vehicular networking: a performance evaluation. *EURASIP Journal on Wireless Communications and Networking*, 2014(1):89, May 2014.

[HXD+13] Guangjie Han, Huihui Xu, Trung Q. Duong, Jinfang Jiang, and Takahiro Hara. Localization algorithms of wireless sensor networks: A survey. *Telecommun. Syst.*, 52(4):2419–2436, April 2013.

[INR16] Inrix 2016 traffic scorecard – germany: Ein leitfaden zur stausituation in deutschland. `www.inrix.com/resources/inrix-2016-traffic-scorecard-germany/`, 2016. Accessed on: 01.02.2018.

[IVB97] T. Imielinski, S. Viswanathan, and B. R. Badrinath. Data on air: organization and access. *IEEE Transactions on Knowledge and Data Engineering*, 9(3):353–372, May 1997.

[JH97] Dean Povey John and John Harrison. A distributed internet cache. In *In Proceedings of the 20th Australian Computer Science Conference*, pages 5–7, 1997.

[JST+09] Van Jacobson, Diana K. Smetters, James D. Thornton, Michael F. Plass, Nicholas H. Briggs, and Rebecca L. Braynard. Networking named content. In *Proceedings of the 5th International Conference on Emerging Networking Experiments and Technologies*, CoNEXT '09, pages 1–12, New York, NY, USA, 2009. ACM.

[KKG+00] Anurag Kahol, Sumit Khurana, Sandeep K. S. Gupta, Eep K. S. Gupta, and Pradip K. Srimani. A strategy to manage cache consistency in a distributed mobile wireless environment. *IEEE Trans. Parallel and Distributed Systems*, 12:686–700, 2000.

[KL14] N. Kumar and J. H. Lee. Peer-to-peer cooperative caching for data dissemination in urban vehicular communications. *IEEE Systems Journal*, 8(4):1136–1144, Dec 2014.

[LC09] A. Lakas and M. Cheqfah. Detection and dissipation of road traffic congestion using vehicular communication. In *2009 Mediterrannean Microwave Symposium (MMS)*, pages 1–6, Nov 2009.

[LKMK15] J. Liu, N. Kato, J. Ma, and N. Kadowaki. Device-to-device communication in lte-advanced networks: A survey. *IEEE Communications Surveys Tutorials*, 17(4):1923–1940, Fourthquarter 2015.

[LPD10] N. Loulloudes, G. Pallis, and M. D. Dikaiakos. On the evaluation of caching in vehicular information systems. In *2010 Eleventh International Conference on Mobile Data Management*, pages 218–220, May 2010.

[LS11] A. Lakas and M. Shaqfa. Geocache: Sharing and exchanging road traffic information using peer-to-peer vehicular communication. In *2011 IEEE 73rd Vehicular Technology Conference (VTC Spring)*, pages 1–7, May 2011.

[LYAW12] H. Lv, X. Ye, L. An, and Y. Wang. Distributed beacon frequency control algorithm for vanets (dbfc). In *2012 Second International Conference on Intelligent System Design and Engineering Application*, pages 243–246, Jan 2012.

[LYD09] S. Lim, C. Yu, and C. R. Das. Cooperative cache invalidation strategies for internet-based vehicular ad hoc networks. In *2009 Proceedings of 18th International Conference on Computer Communications and Networks*, pages 1–6, Aug 2009.

[MA10] K. Mershad and H. Artail. Ssum: Smart server update mechanism for maintaining cache consistency in mobile environments. *IEEE Transactions on Mobile Computing*, 9(6):778–795, June 2010.

[Mar02] E. P. Markatos. Tracing a large-scale peer to peer system: An hour in the life of gnutella. In *Cluster Computing and the Grid, 2002. 2nd IEEE/ACM International Symposium on*, pages 65–65, May 2002.

[MDM05] Sven Maerivoet and Bart De Moor. Traffic flow theory. *arXiv preprint physics/0507126*, 2005.

[MLN+17] Tobias Meuser, Patrick Lieser, The An Binh Nguyen, Doreen Böhnstedt, and Ralf Steinmetz. Adaptive information aggregation for application-specific demands. In *Proceedings of the 1st International Balkan Conference on Communications and Networking*, May 2017.

[MNR+98] Scott Michel, Khoi Nguyen, Adam Rosenstein, Lixia Zhang, Sally Floyd, and Van Jacobson. Adaptive web caching: towards a new global caching architecture. *Computer Networks and ISDN Systems*, 30(22):2169 – 2177, 1998.

[MWL+18] Tobias Meuser, Martin Wende, Patrick Lieser, Björn Richerzhagen, and Ralf Steinmetz. Adaptive decision making based on temporal information dynamics. In *Vehicle Technology and Intelligent Transport Systems (accepted for publication)*, Mar 2018.

[NTCS99] Sze-Yao Ni, Yu-Chee Tseng, Yuh-Shyan Chen, and Jang-Ping Sheu. The broadcast storm problem in a mobile ad hoc network. In *Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking*, MobiCom '99, pages 151–162, New York, NY, USA, 1999. ACM.

[PCP14] I. Psaras, W. K. Chai, and G. Pavlou. In-network cache management and resource allocation for information-centric networks. *IEEE Transactions on Parallel and Distributed Systems*, 25(11):2920–2931, Nov 2014.

[PEW09] Vivian Prinz, Robert Eigner, and Wolfgang Woerndl. 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*, IWCMC '09, pages 431–436, New York, NY, USA, 2009. ACM.

[PH03] S. Patro and Y. C. Hu. Transparent query caching in peer-to-peer overlay networks. In *Proceedings International Parallel and Distributed Processing Symposium*, pages 10 pp.–, April 2003.

[Ric17] Björn Richerzhagen. *Mechanism Transitions in Publish/Subscribe Systems - Adaptive Event Brokering for Location-based Mobile Social Applications*. PhD thesis, Darmstadt University of Technology, Germany, 2017.

[RSB01] Pablo Rodriguez, Christian Spanner, and Ernst W. Biersack. Analysis of web caching architectures: Hierarchical and distributed caching. *IEEE/ACM Trans. Netw.*, 9(4):404–418, August 2001.

[RSNT14] A. B. Reis, S. Sargento, F. Neves, and O. K. Tonguz. Deploying roadside units in sparse vehicular networks: What really works and what does not. *IEEE Transactions on Vehicular Technology*, 63(6):2794–2806, July 2014.

[RSRS15] Björn Richerzhagen, Dominik Stingl, Julius Rückert, and Ralf Steinmetz. Simonstrator: Simulation and prototyping platform for distributed mobile applications. In *Proceedings of the 8th International Conference on Simulation Tools and Techniques*, SIMUTools '15, pages 99–108, ICST, Brussels, Belgium, Belgium, 2015. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

[SA06] Gleb Skobeltsyn and Karl Aberer. Distributed cache table: Efficient query-driven processing of multi-term queries in p2p networks. In *Proceedings of the International Workshop on Information Retrieval in Peer-to-peer Networks*, P2PIR '06, pages 33–40, New York, NY, USA, 2006. ACM.

[SBF+15] Julio A. Sanguesa, Javier Barrachina, Manuel Fogue, Piedad Garrido, Francisco J. Martinez, Juan-Carlos Cano, Carlos T. Calafate, and Pietro Manzoni. Sensing traffic density combining v2v and v2i wireless communications. *Sensors*, 15(12):31794–31810, 2015.

[SGR+11] D. Stingl, C. Gross, J. Rückert, L. Nobach, A. Kovacevic, and R. Steinmetz. Peerfactsim.kom: A simulation framework for peer-to-peer systems. In *2011 International Conference on High Performance Computing Simulation*, pages 577–584, July 2011.

[SKZ17] Afiq Aisamuddin Mohd Shariff, Norliza Katuk, and Nur Haryani Zakaria. An overview to pre-fetching techniques for content caching of mobile applications. *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, 9(2-12):37–43, 2017.

[SRA10] Ghassan Samara, Sureswaran Ramadas, and Wafaa A. H. Al-Salihy. Safety message power transmission control for vehicular ad hoc networks. *CoRR*, abs/1007.3058, 2010.

[TDVK98] Renu Tewari, Michael Dahlin, Harrick M. Vin, and Jonathan S. Kay. Beyond hierarchies: Design considerations for distributed caching on the internet. Technical report, IN PROCEEDINGS OF THE 19TH INTERNATIONAL CONFERENCE ON DISTRIBUTED COMPUTING SYSTEMS (ICDCS, 1998.

[WBW+11] Sen Wang, Jun Bi, Jianping Wu, Zhaogeng Li, Wei Zhang, and Xu Yang. Could in-network caching benefit information-centric networking? In *Proceedings of the 7th Asian Internet Engineering Conference*, AINTEC '11, pages 112–115, New York, NY, USA, 2011. ACM.

[WC97] D. Wessels and K. Claffy. Internet Cache Protocol (ICP), version 2. RFC 2186, Internet Engineering Task Force, September 1997.

[WDCK04] Z. Wang, S. K. Das, H. Che, and Mohan Kumar. A scalable asynchronous cache consistency scheme (saccs) for mobile environments. *IEEE Transactions on Parallel and Distributed Systems*, 15(11):983–995, Nov 2004.

[WHO15] Global status report on road safety 2015. www.who.int/violence_injury_prevention/road_safety_status/2015, 2015. Accessed on: 01.01.2017.

[WXLZ04] C. Wang, L. Xiao, Y. Liu, and P. Zheng. Distributed caching and adaptive search in multilayer p2p networks. In *24th International Conference on Distributed Computing Systems, 2004. Proceedings.*, pages 219–226, 2004.

[WXLZ06] Chen Wang, Li Xiao, Yunhao Liu, and Pei Zheng. Dicas: An efficient distributed caching mechanism for p2p systems. *IEEE Transactions on Parallel and Distributed Systems*, 17(10):1097–1109, Oct 2006.

[WYC96] Kung-Lung Wu, P. S. Yu, and Ming-Syan Chen. Energy-efficient caching for wireless mobile computing. In *Proceedings of the Twelfth International Conference on Data Engineering*, pages 336–343, Feb 1996.

[XVS+14] G. Xylomenos, C. N. Ververidis, V. A. Siris, N. Fotiou, C. Tsilopoulos, X. Vasilakos, K. V. Katsaros, and G. C. Polyzos. A survey of information-centric networking research. *IEEE Communications Surveys Tutorials*, 16(2):1024–1049, Second 2014.

[YC06] Liangzhong Yin and Guohong Cao. Supporting cooperative caching in ad hoc networks. *IEEE Transactions on Mobile Computing*, 5(1):77–89, January 2006.

[ZC08] J. Zhao and G. Cao. Vadd: Vehicle-assisted data delivery in vehicular ad hoc networks. *IEEE Transactions on Vehicular Technology*, 57(3):1910–1922, May 2008.

[ZJZ+12] F. Zhang, B. Jin, W. Zhuo, Z. Wang, and L. Zhang. A content-based publish/subscribe system for efficient event notification over vehicular ad hoc networks. In *2012 9th International Conference on Ubiquitous Intelligence and Computing and 9th International Conference on Autonomic and Trusted Computing*, pages 64–71, Sept 2012.