DYNAMIC CACHE INVALIDATION IN WIRELESS ENVIRONMENTS

A Thesis submitted in fulfillment of the requirement for the award of the degree of

DOCTOR OF PHILOSOPHY IN COMPUTER SCIENCE AND ENGINEERING

Submitted By

Rajeev Tiwari
(Registration No: 951103004)

Under the guidance of:

Dr. Neeraj Kumar

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT
THAPAR UNIVERSITY, PATIALA–147004 (PUNJAB–INDIA)

APRIL 2016

CERTIFICATE

I, Rajeev Tiwari, Regn.No. 951103004, hereby declares that the thesis entitled **"Dynamic Cache Invalidation in Wireless Environments"** submitted to the Department of Computer Science & Engineering at Thapar University, Patiala, Punjab, India is an authenticated record of my own work for the award of the degree of "Doctor of Philosophy" under the supervision of Dr. Neeraj Kumar. This report has not been submitted to any other institution for award of any degree.

Rajeev Tiwari

Rajeev Tiwari
951103004

Place: Patiala
Date: 29/8/2016

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Approved by:

Dr. Neeraj Kumar,
(Supervisor)
Assistant Professor,
Department of CSE,
Thapar University, Patiala.

Abstract

In this era of Internet and its related technologies such as 4G, and 5G, there is a need of fast response time for various queries raised by intermediate nodes and mobile terminals (MTs) from infra structured-based/less networks. Although there has been many efforts in the past to address this issue by using an Invalidation Report (IR)-based cache management schemes which reduce the bandwidth requirements and battery consumptions but, for high update rate on server, most of the existing approaches have long query latency due to large and fixed size of IRs, and broadcast time (BT) interval. In all such cases, IRs become unmanageable due to the poor utilization of available bandwidth and non-adaptive size of IRs. So, to address these issues, in this thesis, various cache invalidation techniques are proposed. In the proposed work, two types of environments are considered for validation of designed techniques. One is wireless environment without gateway cooperation and another is Internet based vehicular environment with cooperation from gateways.

In wireless environment without gateway cooperation, an adaptive Cache invalidation technique (ACIT) is proposed. In comparison to the previous approaches, the proposed scheme uses different threshold update rates for adaptive IR, and BT intervals. In the proposed scheme, only hot data updates in IR are recorded which results a less query delay and bandwidth consumption. The performance of the proposed ACIT scheme is studied in wireless environment by extensive simulations with respect to various metrics such as Average Query Response Time (AQRT), IR size, Number of Uplink Requests, and BT by varying update and query rates. The performance of the proposed scheme is evaluated by comparing it with other state-of-the-art schemes such as Update Invalidation Report (UIR) and Selective Adaptive Sorted (SAS). The results obtained shows that the proposed scheme is better than the existing schemes.

Another environment considered in the proposed work is Internet based vehicular ad hoc network (IVANET). It is a fast growing technology with an aim to provide uninterrupted services such as regular safety alerts, entertainment, and resource sharing to the on board passengers even on-the-fly. On board passengers avail all these services even on-the-fly either in dense urban regions, or on highway using an Internet connection. Reduction in delay with an increase in the accuracy for accessing all these services from anywhere are the most challenging tasks to be performed in this environment. In literature, it has been found that caching the most relevant content at some of the intermediate sites may increase the overall performance of the network in this environment. In this direction, an important issue is to invalidate the cache when the original data items are updated and these data items are irrelevant for the end users. So, to tackle these issues, a Cooperative Gateway Cache Invalidation (CGCI) scheme is proposed. The designed scheme has the advantages of cooperation of gateways in different regions along with the underlying location management scheme to reduce the number of broadcast operations, lesser query delay , and Uplink requests with an increase in the cache hit ratio. To test the effectiveness of the proposed scheme, we carried out extensive simulations by varying query arrival rate, object update rate, and cache size. Moreover, to evaluate the cost of query communication among the vehicles, an analytical model is also included in the proposed scheme. The results obtained confirm that the proposed scheme yields a reduction in the query delay with an increase in the cache hit ratio as compared to other state-of-the art existing schemes in literature. The proposed scheme includes stateful server, thats keeps track of updates of data and generates IR which is broadcasted to HA which in turn, unicast the updated data to GFA to the vehicles. The proposed scheme has taken benefits of cooperation of neighboring GFAs which store the hot data fetched from server on any query reply. The proposed scheme works faster with cooperation of GFAs so, queries are replied faster from cache or GFA which reduces the number of Uplink requests.

To evaluate the performance of CGCI, exhaustive simulations are performed by varying the parameters such as-update rate query response time, Uplink requests hit ratio, and broadcast time. Results obtained show the effectiveness of proposed CGCI scheme over other state-of-the-art schemes such as UIR, SAS, CCI, and ECCI. The results obtained shows that the proposed scheme is better than the other schemes such as CCI, EECI, UIR and SAS.

In summery we can say that proposed CIT techniques ACIT in first wireless environment has adaptive IR, and BT. It is compared with peer techniques UIR and SAS on basis of parameters such as AQRT, IR size, Number of Uplink Requests and BT with respect to varying query rate and update rates. Proposed technique has shown promising results in comparison to the peer techniques. Also, other proposed technique CGCI in other IVANET wireless environment is compared with UIR, SAS, CCI, and ECCI techniques based on parameters such as query response time, uplink requests, and higher hit ratio. Proposed technique CGCI has promising results in terms of hit ratio and Uplink requests in comparison to peer techniques.

Keywords: *Cache invalidation, Cooperative caching, Cache hit ratio, Wireless networks, Adaptive, Invalidation Report, Cache hit ratio, IVANET.*

ACKNOWLEDGMENT

First of all and the greatest importance, I would be thankful to GOD for all his blessings without which nothing of my work would have been carried out. I am highly indebted to Prof. (Dr.) Prakash Gopalan, Director, Thapar University, Patiala and Prof. (Dr.) R. S. Kaler and Dr. Susheel Mittal, Deputy Directors, Thapar University for providing me the opportunity to pursue my Ph.D. and course work. This thesis would not be possible without the continuous guidance from my supervisor Dr. Neeraj Kumar, Associate Professor, Computer Science and Engineering Department, Thapar University, Patiala. The extensive discussions with him have helped me to stay on the right track towards finishing this work. He was always patience and helpful whenever his guidance and assistance was needed. He also helped me by sparing his valuable time for reviewing my experimental setup, publications, reports and presentation of my work from time to time. He also helped me in shaping my initial work of thesis during synopsis submission and I would like to acknowledge him with deferential thanks. As my supervisor, he provided me with the encouragement and freedom to pursue my own ideas. He taught me how to do the research. In other words, he taught me how to make gold out of stones. He always remained tranquil while outlaying his edifying time on me. I am also grateful to him for spending several hours for reviewing the document for accuracy and providing valuable feedback during the several iterations that this document went through. Therefore, I want to show my deep appreciation to him. I would also like to convey my thanks to Dr. Deepak Garg, Head, Computer Science and Engineering Department for providing me the continuous feedback throughout my Ph.D. work. I am thankful to my doctoral committee members Dr. Anil Kumar Verma and Dr. Rinkle Aggarwal for their constructive comments and ensuring the progress of my research work in the right direction. Their constants opinions and ideas help me to reach at this stage of my course. The helped rendered by them is greatly accredited. My deep regards to Dr. O.P. Pandey, Dean of Research and Sponsored Projects for his enormous help in completion of my course work. I will go a miss if I forget to thank Dr. Seema Bawa, Dr. Maninder Singh, and Dr. Parteek Bhatia for their invaluable suggestions during various progress monitoring presentations. I wish to thank all the faculty and staff members of Computer Science and Engineering Department of Thapar University, Patiala for their co-operation and support.

Special thanks to V.C. Dr. Shri Hari, CEO Mr. Utpal Ghosh, Deans Dr. Kamal Bansal and Dr. Manish Prateek and Head Dr. Amit Aggarwal of UPES, for putting faith in me and giving me eminent encouragement to do my research work. They have always helped me at every twirl of my life when I am in a need.

My work would not have been possible without the support of my family so, I would like to pay my sincere gratitude to my father Mr. Surender Nath Tiwari, mother Mrs. Mala Tiwari for motivating me in every situation and don't let me down in my life. Finally, and most importantly, I would like to thank my wife Dr. Shuchi Tiwari for her support, encouragement, quiet patience and unwavering love upon which the past years of my life have been built. Her tolerance of my occasional moods is a testament in itself of her unyielding devotion and love, without her love it would have not been possible. Sincere thanks to Mr. Sanjeev Tiwari, elder brother who has been persuasive for my progress and encouragement all through my life. My hearties gratitude towards my little daughter Aanya, she has binded me emotionally for completing my work. Words cannot truly express my deepest gratitude and appreciation to all of the above. I am sorry, if I have forgotten someone and I cannot thank everyone enough for the involvement they have shown and the willingness they have expressed to take on the completion of tasks beyond their comfort zones.

Contents

List of Figures

List of Tables

LIST OF IMPORTANT ABBREVIATIONS

Acronyms	**Meanings**
2TierCoCS	Two-Tier Cooperative Caching Scheme
ACB	Asynchronous Call Back
ACIT	Adaptive Cache Invalidation Technique
ADD	Adaptive Data Dividing
ADU	application-level data unit
AIS	Adaptive Invalidation Strategy
AODV	Adhoc On-demand Distance Vector
AP	Access Point
AQRT	Average Query Response Time
N_{gfa}^{ap}	No of hops between AP & GFA
ARPA	Advanced Research Projects Agency
ARPANET	Advanced Research Projects Agency Network
AVI	Absolute Validity Interval
BS	Base Station
BSA	Basic Service Area
BSeq	Bit Sequencing
BSS	Basic Service Set
BT	Broadcast Time
C	Cold Data %age
$Cache_{MT}$	Cache of MT
$Cache_{V}$	Cache of Vehicle
CB	Call Back
CBSFI	Counter Based Stateful Invalidation Scheme
CBSLI	Counter Based Stateless Invalidation Scheme
CCI	Cooperative Cache Invalidation
CGCI	Cooperative Gateway Cache Invalidation
CHR	Cache History Register
CIP	Clients Internet Address
CIR	Current Invalidation Report
CIReg	Cached Item Register
CIT	Cache Invalidation Technique
CNs	caching nodes
CoA_{AP}	Care of Address of AP
COA_{GFA}	Care of Address of GFA
CoA_{MT}	Care of Address of Mobie Terminal
CoA_{S}	Care of Address of Server
COA_{V}	Care of Address of Vehicle
COOP	Cooperative Technque

$Count_{dx}$	Counter register associated with dx
CQ	Cold Query
C_{size}	Cache Size on MT
C_{th}	Counter threshold value
CU	Cold Update
CURP	Cache Update Request Packet
D	Total Data on Server
DBT	Dynamic Broadcast Time Techniques
DIRV	Different IRs with Variable Sizes
DREP	Data Reply Packet
DRP	Data Request Packet
DS	Distribution System
DSDV	Destination Sequence Distance Vector
DSM	Distribution System Medium
DSR	Dynamic Source Routing
DSRC	Dedicated Short Range Communication
DSS	Distribution System Service
DT	**Disconnection Time Duration**
DVD	Data Validity Defining Invalidation
DVFIR	Different IRs with Fixed Sizes
DVIBT	Variable Informed Broadcast Time Technique
DVIIR	Different IRs with Informed Contents
DVIR	Dynamic/Variable IR Techniques
DVUBT	Variable Uninformed Broadcast Time Technique
d_x	Data Page Id
ECCI	Enhanced Cooperative Cache
FCFS	First Come First Serve
FCPP	Flexible Combination of Push and Pull algorithm
FIP	False Invalid period
FVP	False Valid Period
GCI	Group Cache Invalidation Report
GFA	Gateway Foreign Agent
GIR	Group Invalidation Report
GPS	Global Positioning System
H	Hot Data %age
HA	Home Agent
HCI	Hybrid Cache Invalidation Report
HIR	History Invalidation Report
HQ	Hot Query
HU	Hot Update
I	Initial
IAVI	Invalidation by Absolute Validity Interval
IBSS	Independent BSS
IoT	Internet of Things
IR	Invalidation Report
IVANETs	Internet Based Vehicular Ad Hoc Networks
k	Constant Delay of each hop
L	Normal Mode broadcast time

l_{bt}	Last Broadcast Time
LDT	Large Disconnection Time Technique
LEAVE	Node Departure
LTE	long term evolution
LRU	Least Recently Used
LUR	Load Utilization Ratio
MANETs	Mobile ad hoc Networks
MB	Mode Bit
MBSS	Mesh BSS
MIPV6	Mobile IPV6
MQAT	Mean Query Arrival Time
MT	Mobile Terminal
MTS	Modified TS
n	Number of Packets
N_{gfa}^{s}	No of hops between server & GFA
N_{ap}^{v}	No of hops between vehicle & AP
N_{gfa}^{ap}	No of hops between AP & GFA
N_{ha}^{gfa}	No of hops between GFA & HA
NC	No Cache
NS-2	Network Simulator 2
OCI	Object Cache Invalidation Report
OIR	Object Invalidation Report
OTcL	Object Oriented Tcl
P2P	Peer to Peer
P2PCC	Peer-to-Peer Cooperative Caching
$Page(d_x)$	Data Page d_x
PCB	Probability-based Callback
PCC	predictive caching consistency
PDA	Personal Digital Assistants
PER	Poll Each Read
P_{hit}	Probability of finding page in cache
P_{miss}	Probability of page missing from cache
P_{GFA}^{hit}	Probability of hit of a queried data on GFA.
PUMA	Protocol for United Multi casting Through Announcements
QDs	Query Directories
QHWC	QoS-Aware Hierarchical Web Caching Scheme
QoS	Quality of Service
QP_s	Query Processing delay time at Server
QP_{bs}	Query Processing delay time at BS
QP_{ap}	Query Processing delay time at AP
QP_{mt}	Query Processing delay time at MT
Q_{rate}	Query Rate
QP_v	Query Processing delay time at Vehicle
QRT	Query Response Time
RIH	Reduced Improving Handling
RMA	Remainder
RSU	Road Side Unit
RTT	Round Trip Time

S	Data Server
SAS	Selective adaptive Sorted
SBPER	Server Based Poll Each Read
SBT	Fixed/Static Broadcast Time
SCI	Selective Cache Invalidation
SDT	Small Disconnection Time Techniques
SIR	Static/Fixed Size IR Techniques
STA	Communicating Station
SUDP	Server Update Data Packet
SVRP	Server Validation Reply
t_{bt}	Length of current broadcast interval
TORA	Temporally ordered Routing Algorithm
t_q, t_u	Time of query and update
TS	Time Stamp
ts_{curr}	Current Time stamp
TTL	Time To Live
U	Update
UDT	Unlimited Disconnection Times Techniques
UIR	Update Invalidation Report
UR	Uplink Requests
U_{rate}	Update Rate of current t_{bt}
U_{th}	Update rate threshold value
VANETs	Vehicular Ad Hoc Networks
VDT	Variable Disconnection Time Techniques
VINT	Virtual Internet Testbed
w	No of broadcast intervals for carrying history
W	Waiting
w_c	Wireless Communication Cost
wL	w Broadcast Times
WLAN	Wireless LAN
WM	Wireless Medium
WPAN	Wireless Personal Area Network
WWW	World Wide Web
τ_{GFA}	Round Trip Time from Vehicle to GFA
τ_S	Round Trip Time from Vehicle to Server

List of Publications

SCI/SCIE Journals:

1. Rajeev Tiwari, Neeraj Kumar, " An Adaptive Cache Invalidation Technique for Wireless Environments", Telecommunication Systems,62(1): 149-165 (2016).**(SCI-Thomson Reuters Rated Journal with Impact Factor 0.705)**.

2. Rajeev Tiwari, Neeraj Kumar, "Cooperative Gateway Cache Invalidation Scheme for Internet-Based Vehicular Ad Hoc Networks", Wireless Personal Communications, 85(4): 1789-1814 (2015). **(SCI-Thomson Reuters Rated Journal with Impact Factor 0.653)**.

International Conferences:

1. Rajeev Tiwari, Neeraj Kumar, " Dynamic cache invalidation technique in mobile environments", In proceedings of 1st International Conference on Next Generation Computing Technologies (NGCT-2015), Dehradun, India, 4th-5th September-2015, pp.320-324.

2. Rajeev Tiwari, Neeraj Kumar, " Performance comparison of cache invalidation techniques in mobile computing environment", In proceedings of 1st International Conference on Next Generation Computing Technologies (NGCT-2015), Dehradun, India, 4th-5th September-2015, pp.101-103.

3. Rajeev Tiwari, Neeraj Kumar,"Minimizing Query Delay using Co-operation in IVANET", In proceeding of Third International conference on Recent Trends in Computing, Delhi, Procedia Computer Science, Elsevier, vol. 57, pp. 84-90, 2015.

4. Rajeev Tiwari, Neeraj Kumar, "A Novel Hybrid Approach for Web Caching", In proceedings of 6th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS-2012), Palermo, Italy. pp. 512-517.

5. Rajeev Tiwari, Neeraj Kumar, "Dynamic Web caching: For robustness, low latency & disconnection handling", In proceedings of International Conference on Parallel Distributed and Grid Computing-2012 (PDGC-2012), Solan, India. pp. 909-914.

Chapter 1

Introduction

This chapter starts with history of communications, IEEE standard architecture used for wireless networks. Then, the need of caching and cache invalidation techniques are discussed followed by the classification of the traditional cache invalidation techniques with detailed discussion along with relative advantages and disadvantages. Then, motivation and problem formulation for the proposed scheme are given. In the next Section, objectives of the thesis along with the research contributions are discussed. Finally, the organization of thesis is discussed.

1.1 History of communication

A computer network, generally called as a network, is a collection of computers and other interconnected devices using channels that provides communication between users and allow them to share information and resources. Computer networks can be used for several purposes to facilitate communication, sharing hardware, information preservation, security, sharing of files, data and information [1, 2]. Computer networking is the engineering discipline, which is related to communication among computer systems or devices [1, 3]. In the recent years, there has been a tremendous technological advancements in wireless communication and networking. Advanced CPU technology and highly efficient batteries have made mobile terminals (MTs) such as laptops, personal digital assistants (PDAs) and web-enabled mobile phones smaller in size. In 1960s, the advanced research projects agency (ARPA) initiated the design of the advanced research projects agency network (ARPANET) for the United States

Department of Defense.

Networks are classified in two main categories as wired or wireless. A wireless network is very much flexible and provides a lot of freedom for mobility and connectivity. Wireless networks are categorized into Infra structured and Infrastructure less and are discussed as follows.

- Infrastructure less Wireless Networks

  In infrastructure less network, there is no need of existing infrastructure to create the network, e.g., mobile ad hoc networks (MANETs). In MANETs, network can be formed dynamically using collection of wireless nodes to share information without making use of pre-existing laid infrastructure of network. This part of communication technology supports truly pervasive computing because the contextual information is exchanged between MTs which may rely on rapid configuration of a wireless connections on-the-fly but not on any fixed network infrastructure. MANETs have emerged as the most popular networks to be used in the military and commercial applications, where fixed infrastructures are not available. They comprise of MT devices which use wireless transmission for communication without including any of pre-existing network infrastructure. In these networks, a collection of MTs form a temporary network without using any standard administration or support services. In Latin, ad hoc means *"for this"* extended to *"for this purpose only"* and temporary. A MANET is shown in Figure 1.1.

- Infrastructure Wireless Networks

  Mobile Communication is based on cellular architecture which is a special case of Wireless Networks. It depends upon good infrastructure support, in which MTs communicate with access points such as- base stations connected to the fixed network infrastructure. The success of infrastructure networks, like the Internet and wireless mobile/wireless personal area network (WPAN)/wireless local area network (WLAN) networks increased in such a way that they have now become an integrated part of our day-to-day life.

In wireless environment, there are networks which are laid on cellular architectures for better support. They have infrastructural support at upper hierarchy levels and

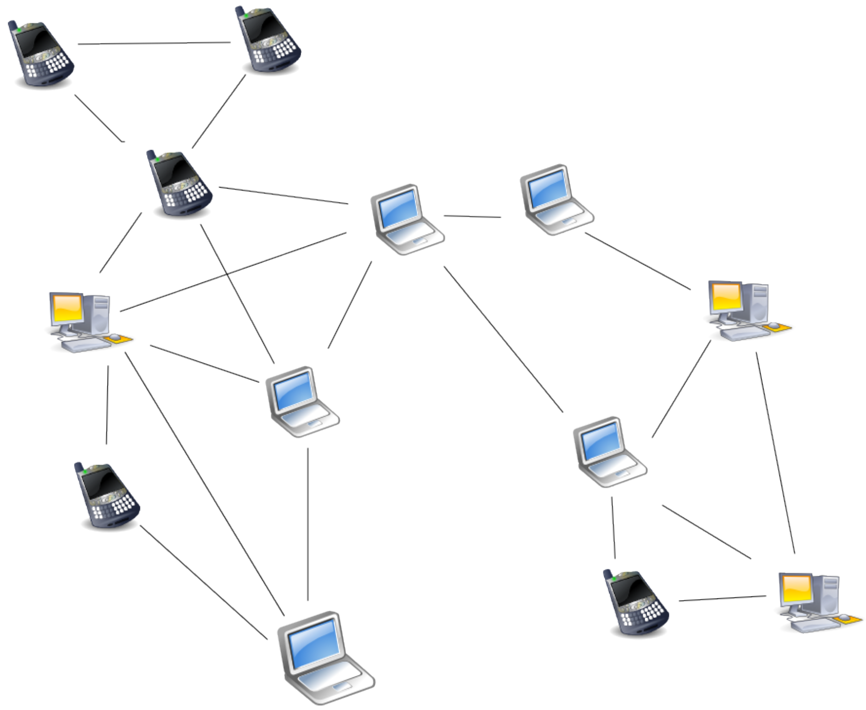

Figure 1.1: Mobile ad hoc network architecture

have ad hoc networks at lower levels. Devices in MANETs use IEEE 802.11 standards for communication, which is explained in next Section.

1.2 Components of the IEEE 802.11 architecture

This section explains the IEEE 802.11 architecture used in wireless communication for various devices. Every component is elaborated one by one. Following are the components of the given architecture.

1.2.1 Typical 802.11 architecture

The IEEE 802.11 architecture consists of many components that work in coordination to provide a Wireless LAN(WLAN). The basic service set (BSS) is main building block unit of an IEEE 802.11 LAN. Figure 1.2 includes two BSSs, each BSS has communicating stations (STAs) as its own members. BSS is considered as the coverage region area in which STAs remains in communication with each other. This coverage area is known as Basic Service Area (BSA). Outside of this covering area no STAs can communicate directly with STAs inside the BSA. The independent BSS (IBSS)

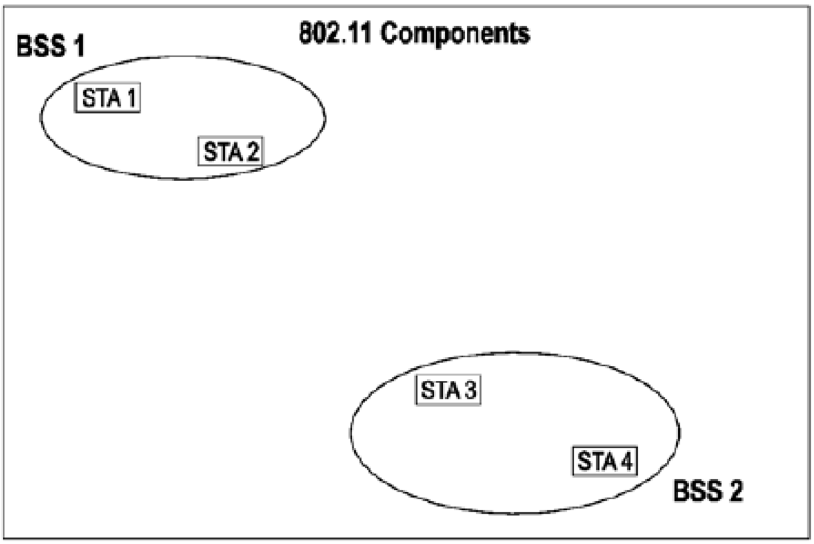

Figure 1.2: Typical 802.11 architecture

is an IEEE 802.11 LAN of basic type. A IBSS can have minimum two STAs. BSSs shown in Figure 1.2 are simple and lack other components which are shown in Figure 1.3. Two STAs can be representative of two IBSSs when they are residing in their BSAs. When two STAs communicate with each other directly, called as MANET and such networks can be created without pre-planning.

1.2.2 STA dynamic registration in a BSS

STA's can register on the fly i.e. dynamically in a BSS It means STAs may turn on, turn off, come within BSA range, and goes out of BSA). A synchronization method is used by STA to become a member of an infrastructure BSS or an IBSS. To become a member in a mesh BSS, beacons are broadcasted and synchronization maintenance procedure is used. When a STA gets associated with a BSS, it may access all the srvices of BSS. These associations are dynamic and makes use of the distribution system service (DSS). A mesh STA peers with other mesh STAs.

1.2.3 Distribution system (DS) concepts

For some networks, station-to-station sufficient distance is maintained but, for other networks, increased coverage region is required. An infrastructure BSS may also be extended to built with multiple BSSs. Such architectural component uses DS to interconnect infrastructure BSSs.

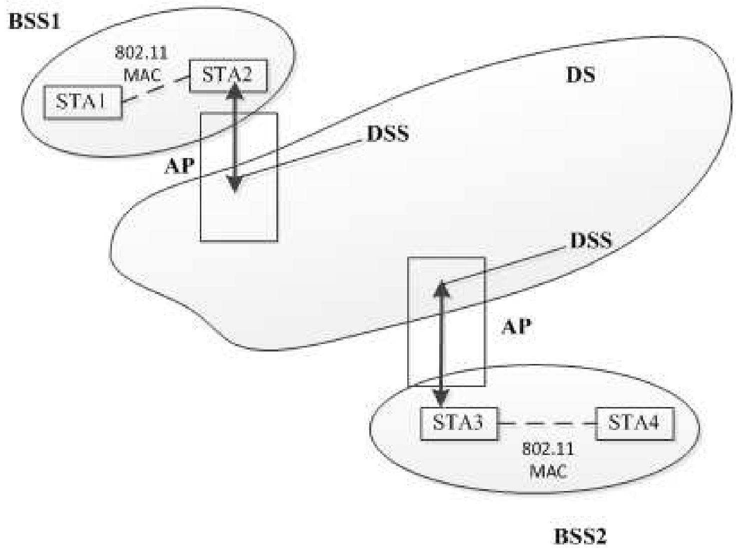

Figure 1.3: 802.11 IEEE standard components for infrastructure wireless networks

Wireless medium (WM) is separated logically in IEEE 802.11 from the distribution system medium (DSM). For implementation of some specific requirement characteristic, IEEE 802.11 LAN architecture is customized separately in BSS. The DS provides support to make rational MT connections providing the logical services to handle address to destination mapping. And also provides integration of multiple BSSs with a DSS. An access point (AP) is a component that enables associated STAs connectivity on DS through WM. Figure 1.3 shows DS, DSM and AP components for the IEEE 802.11 architecture.

Ap acts as an intermediary interface for data movement between BSS and DS. All APs are similer to STAs thus, they are network addressable entities. The addresses assigned to APs for communication to WM and DSM are may be different. Data sent by STAs associated with any of the AP's is always received at the uncontrolled port (used for processing) by IEEE 802.1X port access.

1.3 Wireless Environment

Now a days, Internet and related technologies for various applications such as health care, transportations, monitoring and surveillance usage is exponentially growing . During this era, various wireless standards have been developed to reduce the access

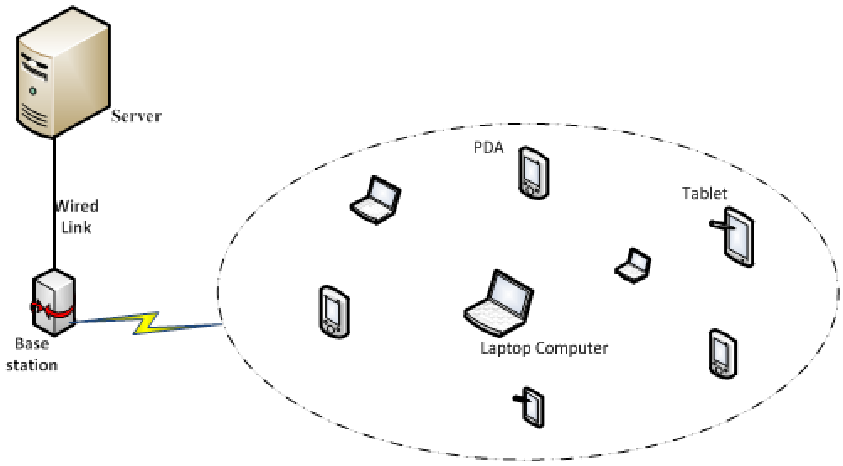

Figure 1.4: A typical wireless environment

delay of various services accessed by the end user. The popularity of different kinds of portable devices to access these services from the service providers has also increased with an aim to provide an efficient wireless environment where users can access and use various services with less delay. But, the access delay of these services are dependent on the type of network being used, users can access both wired/wireless networks as both these types of networks have different characteristics which differentiate them from one another.

In a wireless environment, servers connect to each another via a wireless network /wired network, where every client is connected to server by wireless channel and has smaller parts of the data in its cache. Such a wireless environment can work independently or has gateways for accessing the data from remote sites. When a server can communicate with clients via one or more gateways and peer clients communicate with one other in an ad hoc manner, then users can access data from any location with an ease. This type of environment is shown in Figure 1.4.

1.3.1 Standards Used in Wireless Communications

A wireless network has mobile terminals (MTs), APs, and base station (BS) to provide services to the end users. But, the MTs have high mobility, which leads to dynamic topological changes [4] and peers communicate with one another in a ad hoc manner. [5, 6, 7]. There exist a hierarchical architecture in this environment in which MTs are connected to an AP, which is connected to a BS, and finally, BSs are connected to server(S), which contains the overall data repository (D) for all the operations

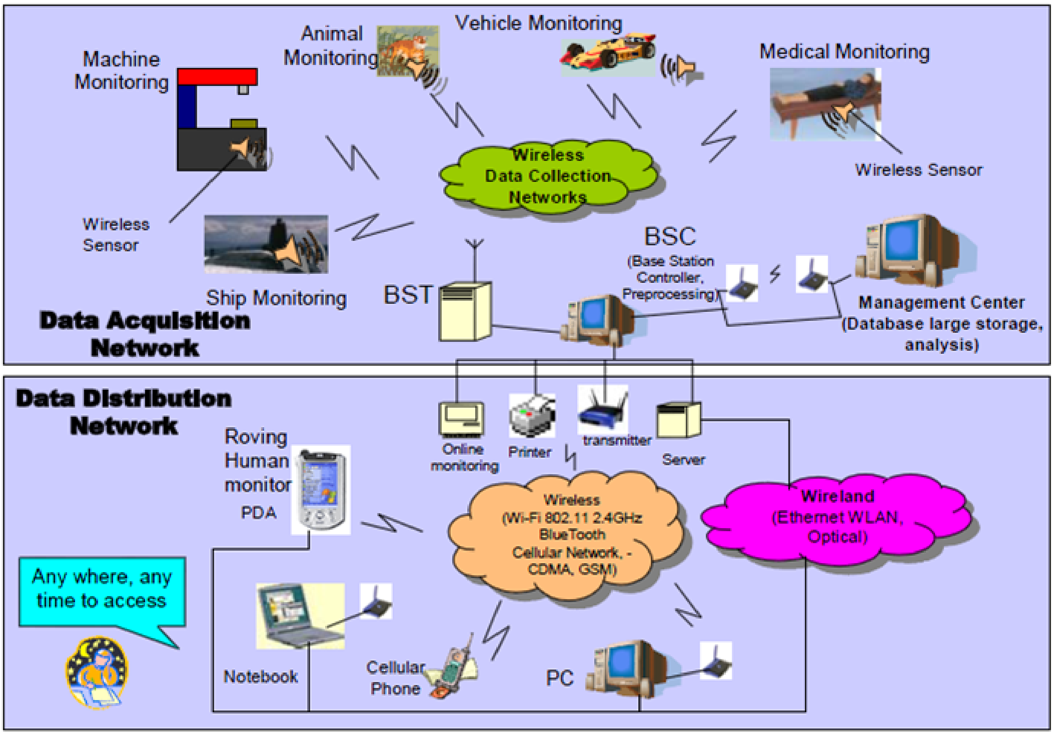

Figure 1.5: Application areas of wireless networks

performed in this environment. MTs are connected via unreliable low-bandwidth communication channels with the server having frequent disconnections due to power saving strategies and mobility [8, 9], i.e., this type of environment has a slow wireless link having MTs with limited battery power. Communication channels can be classified as two categories. First is Uplink channel which is used for communication of requests from MTs to server while other is downlink channel which is, utilized for sending or downloading data from the server to MTs. Uplink channel is limited with respect to the available bandwidth, so lesser Uplink traffic needs to be transferred from clients to the server. Some of the application areas of such type of networks are shown in Figure 1.5 below. In all subsequent Sections, we have taken MTs, mobile nodes and clients as same.

In Figure 1.5, it is clearly shown that using infrastructure based wireless network, user can access various services very easily.

1.4 Need of Caching and Invalidation

In a highly distributed environment of wireless environment, MTs have higher mobility. Due to their mobility, they have dynamic topology (connections and configurations change frequently) which changes constantly [4]. These MTs are connected via unreliable low-bandwidth communication channels with frequent disconnections. It may occur because of the following reasons:- the MT or client may switch off by itself (to save battery power), or a client may be disconnected involuntarily due to its own mobility in a wireless environment (hand-off, and wireless link failures). A wireless environment has a slow wireless links in which MTs have limited battery powers. Facilitation of an efficient wireless environment becomes complicated because of enormous growth, and development of different portable equipments connected to the wireless network. Wireless environments have different attributes from conventional wired networks such as limited bandwidth, slower Uplink channel than the downlink channel, and suffers from frequent disconnections between servers and MTs [8, 9]. MTs can voluntarily be in switch off mode to save battery power or may be involuntarily disconnected due to hand-offs or wireless link failures. Because of these features, such conventional techniques of data access may not suite for wireless environments.

On MT side caching of mostly used data is an effective way to minimize network traffic and query latency. But, maintaining data consistency between clients and servers is a big challenge every time so that clients need not answer the queries by invalid cached data items, which are updated by servers. Cache consistency maintenance means that whenever data items get changed on the server then the changes must also be reflected in the cache of the MTs. Figure 1.6 shows that for cache invalidation we need to have three locations-location 1, location 2 and location 3. Location 1 is the place where all the cache contents/data pages are places in the database repository, location 2 is the place of AP where a database for some pages can be placed, and finally at location 3, cache of each individual MT can be used as shown in Figure 1.6. We have considered cache invalidation of location 3 in this thesis because on MT side, caching of the most recently used data is an effective way to minimize network traffic and query latency due to which Uplink requests can be reduced.

From the above discussion, it is clear that access delay is one of the major challenges in accessing various services when MTs have high mobility in wireless environment. One of the solutions to reduce the delay is-Caching, in which On MT side caching of mostly used data is an effective way to minimize network traffic and query latency to enhance the network performance. It is an efficient technique to increase

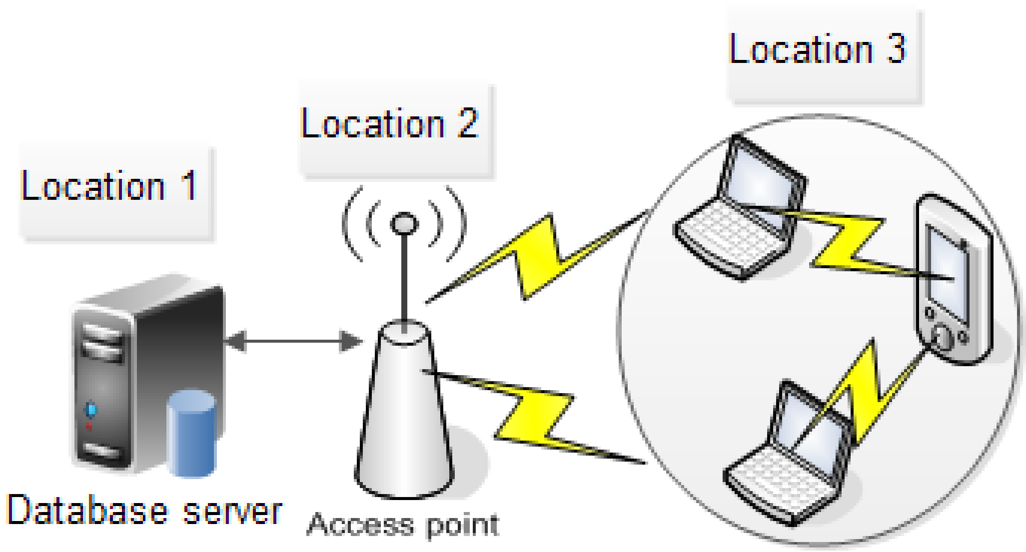

Figure 1.6: Generalized architecture for various locations for Cache Invalidation

the performance of network in which mostly accessed data items are placed either in the MTs memory or to the nearest APs. Due to cache, every time MTs need not ask for data from the remote servers which may cause a long access delay. If data is available in the cache of MTs, then they can respond quickly for any query related to it. However, data consistency in cache of MT must be maintained between MTs and servers to prevent MTs from answering to those queries of data which are outdated by servers, i.e., which are invalidated from the server. A data item is invalidated if a query to access the same has bee generated from the clients and it has been updated by the server and clients are still using the old copy of the data item.

1.5 Cache Invalidation

Cache invalidation is the technique in which most recent contents are broadcasted to the MTs from the server. As soon as contents are updated, these must be broadcasted to the MTs so that they can update their cache contents. But, due to highly dynamic nature of wireless environment, the frequent disconnection rate and high mobility can make MTs unable to receive updates from the server which results in cache invalidation. To validate the cache, MT either have to drop their whole content or "invalidate" the cache which is called as "Cache Invalidation". Cache Invalidation Technique (CIT) allows the MTs to restore their cache contents to a valid state and rejects the invalid cache contents. Valid state can be last consistent state or new consistent state as adopted by the technique. A good invalidation strategy should use minimum bandwidth of wireless channels and limited resources of the MTs. This saves bandwidth and reduces Uplink traffic considerably which makes query response fast. Faster responses are required in wireless networks such as vehicular ad hoc networks (VANETs), where vehicles (communicating nodes) communicate with one other smartly [10, 11, 12, 13, 14, 15].

1.6 Classification of Traditional Cache Invalidation Techniques

Traditional CITs are classified as:- Cache status should be maintained or not, who initiates cache validity (driven by client request or by servers asynchronously) and level of cache consistency (fully consistent, inconsistent or partial consistent).This classification is given in Figure 1.7. The detailed description of each approach is described as follows.

1.6.1 Maintains Cache or Not

In this categorization, depending upon cache state should be maintained or not on server, stateless and stateful approaches exists and are explained as follows:

i) **Stateless Approach[16, 17, 18, 19, 20]**
In this approach, a server may not maintain any cache data state ,i.e., which data pages are to be cached in which cache. If there is a request for web page on client machine, request is checked whether is present in cache or not. Whatever the status of page in cache is, all the requests are forwarded to upper level hierarchy servers for validation.
The most common approach in this category is Poll Each Read (PER)[16]. In this technique, at each request of a page, a client or MT uses a request message sent to the server to check validity of data at this time. When object is valid the server sends an acknowledgment to the client; otherwise, the object is sent by server to the client. Due to stateless server approach, server can update objects

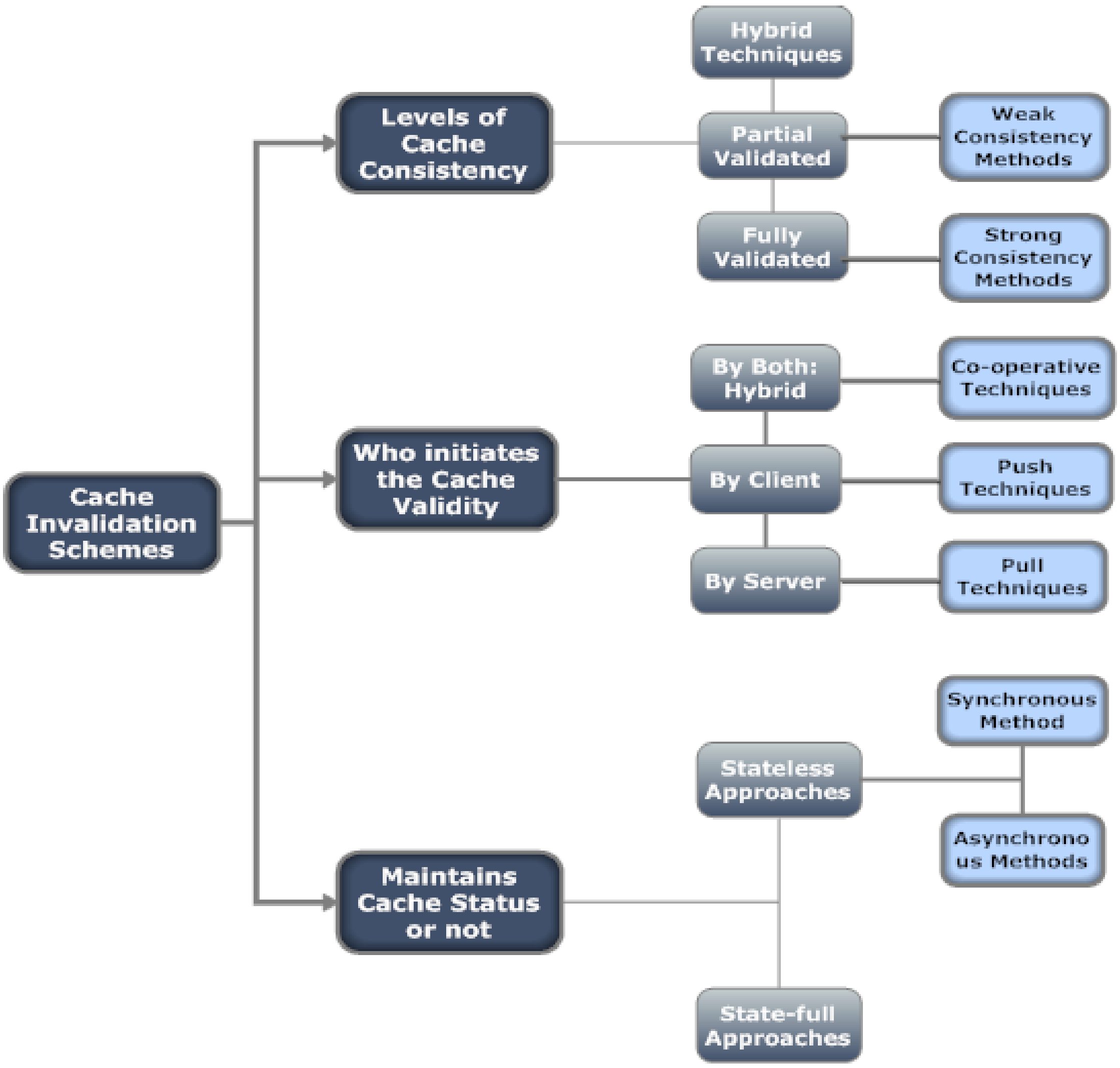

Figure 1.7: Classification of traditional CITs

as per need without notifying the MTs as well. PER uses a stateless server, i.e., the server does not keep track of what data objects are updated and are in use by which MTs. The major advantage of this technique is that it updates copy of data every time client sends a request. But, following are the disadvantages of this technique: It has more latency for queries. Also it has an extra over head of checking validity. With no real usage of caching.

This technique is further sub classified into synchronous and asynchronous [15, 21, 22, 23, 24] modes as follows:

Asynchronous mode: In this mode there is no synchronization between request and validation from server. An invalidation report(IR) is broadcasted by server for a given data item as soon as this item changes its value. A client which is connected invalidates the cached copy of data and disconnected clients lose their cache content completely. Efficiency of this technique can be enhanced by making use of piggybacking extra information on invalidation message like timestamp and recent changes. Instead of a single IR, we get additional information regarding validity and changes. So, disconnected client can save its cache if it waits for the first asynchronous IR after reconnection, i.e., IR shows no change in data during disconnection period of client. IR Reports are sent after a random period so, waiting time varies. The sequences of operations are shown in Figure 1.8. Following are the disadvantages of this technique: It has more congestion due to broadcast of IR with usage of extra resources. But waiting times of different pages for validation varies. Also, it is not an informed approach, because it has more irrelevant IRs, and cache proxy servers

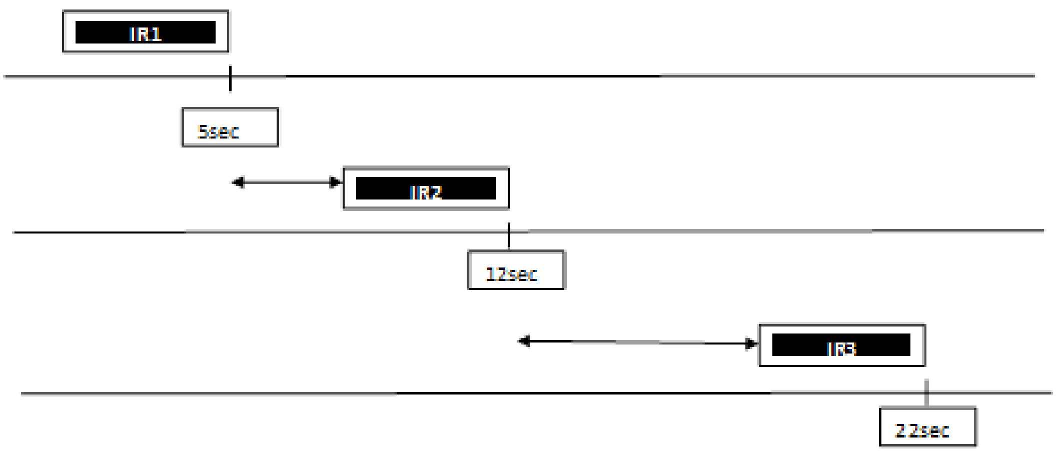

Figure 1.8: Asynchronous Cache Invalidation scheme

Synchronous mode: This mode of cache invalidation is based on periodical broadcasting of IRs of updated data by server, i.e., IRs are sent by servers after a fixed time slots. A client must first listen to IR to make sure that its cache has valid data or not. This may add some latency to query processing. If the cache is not consistent then the client generates a request for needed data to the server for refreshing its cache contents. Validity is rechecked after every broadcasted IR, and timestamp of updated data items is fixed to broadcast time and validity of cached data is checked after listening IR after T_i interval. If client cache's data is invalid then Uplink request is generated and timestamp value is updated by time of Uplink request using server clock. The sequences of operations are shown in Figure 1.9. Following are the disadvantages of this technique. It has an extra delay due to invalidation checks. Also irrelevant IRs may be sent to clients along with more bandwidth usage. Also it is not an instant method as cache has valid data which is still to be verified from the server.

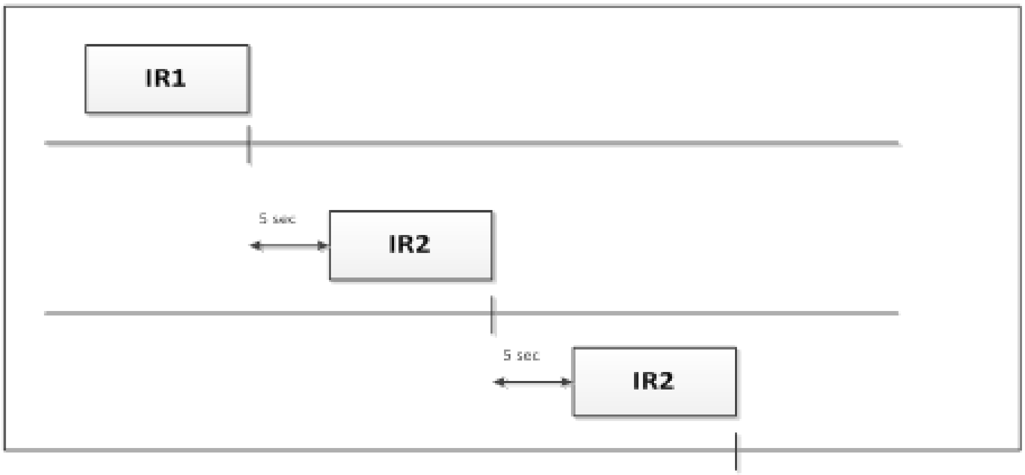

Figure 1.9: Synchronous Cache Invalidation scheme

ii) **Stateful approach [16, 17] :** These are the techniques where the server keeps track of all the data used by a particular device by keeping the registry with itself. In this approach, Call Back and Server Based Poll Each Read techniques are well known. These are explained as follows.

Call Back is the most common technique of stateful type that has different mechanisms for access and updating of data. A client checks if an object is in the cache when the client requests to access the object. If object is in the cache then cached copy is used for reply otherwise, client fetches object from the server.

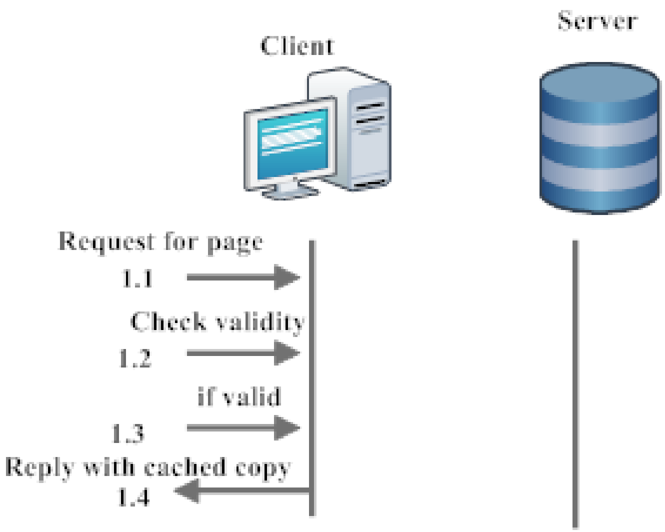

Figure 1.10: Call Back (CB) when requested page is in cache

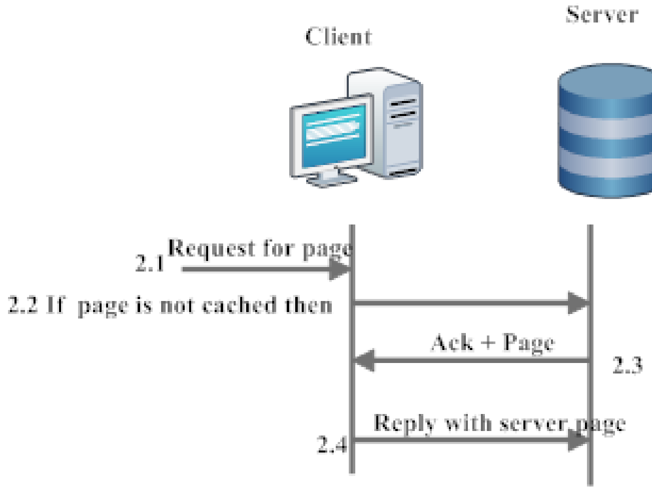

Figure 1.11: CB when requested page is in not in cache

When server updates an object, it informs the clients to remove object from the cache. The objects cached at the client are always valid and a client need not have to contact the server when an access happens unless the data object is not in the cache. Server keeps track of MT or cache server to cache the desired data item. On updating of data item, the server multicasts an IR to tell MTs about the cache data update. Figures 1.10,1.11 and 1.12 explains the sequence of operations in Call Back approach.

Another variant of CB is **Asynchronous Call Back (ACB)**, where on demand basis CB is used. Server waits for request from clients then sends IR report. **Server Based Poll Each Read (SBPER)** is one such technique which uses the

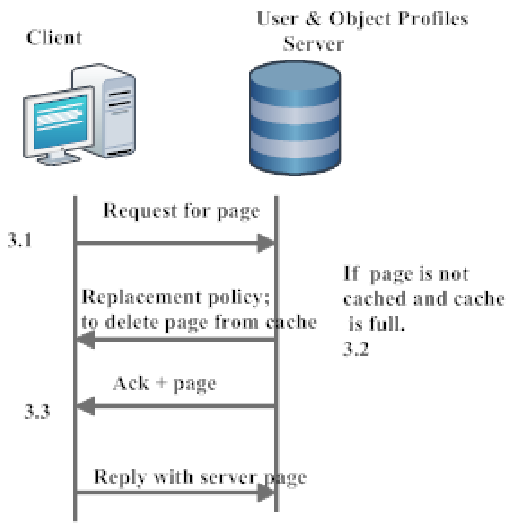

Figure 1.12: Call Back (CB) when requested page is not in cache and cache is full.

stateful server approach. Following are three events take place in this technique and events are described as follows:

- Cached Object is used after checking its validity.
- If there are no updates on server in this object since it is cached means, it is valid.
- When the object is un-cached and cache has some space then server sends object towards client and client caches it.
- If the object is un-cached and cache has no more space then some other object is replaced from cache using replacement algorithms.

The SBPER technique differs from traditional PER with respect to the following points.

- Firstly, PER is stateless and SBPER is a stateful technique.
- SBPER is initiated by server for validation but in PER, validation is initiated by the client
- SBPER updates and removes the pages from the cache of clients and clients do not know about it while in PER, clients initiates the validity and removes the pages from cache when cache is full.

It always forward the validated page to the client but it requires extra space which creates an extra overhead for maintaining profiles and updates.

1.6.2 Who Initiates the Cache Validity

The second traditional based cache categorization depends on the maintenance of cache by server or client and is classified in two parts as given below.

Push and Pull based: In push approach [21],IR is broadcasted by server and initiates the validity of cache. In pull approach [20], for a generated query on MT or client it is replied from a cached copy of data item. MT sends a query request packet upper hierarchy server to check and verify the validity of cached copy of data item before replying. Hybrid technique makes use of both the concepts [21]. Several push, pull [25], and hybrid schemes [21] have been proposed by considering the trade offs between query delay, and communication overhead.

1.6.3 Levels of Cache Consistency

The third traditional cache categorization is based upon the cache's data consistency which can be fully maintained every time (strong consistency) or is weakly maintained (weak consistent) or any combination of these two approaches [21, 26, 27]. Problem due to data replication occurs in such kind of techniques.

1.7 Discussion of Traditional Approaches:

The various traditional approaches as discussed above are compared using various parameters and a critical analysis is given in Table 1.1. The traditional approaches such as PER, CB and ACB are not sufficiently good in terms of optimization of cache size. These schemes suffer from long query delays, sluggish response to queries, and provide more congestion to network due to which network is not reliable with respect to services delivery. Barbra and Imelienski [23] proposed Time Stamp (TS) which has used above approaches and was found a promising solution in terms of strong consistency, average query delays, fast response time, and has less congestion. But, with more number of data items and with higher update rates, the size of IR becomes very large which further increases the query delays and increases the Uplink requests to the server which results in the network congestion. Table 1.1 describes the relative comparison of various traditional CIT based approaches using various metrics.

There are various types of cache management schemes for wireless environments

Table 1.1: Relative comparison of various traditional cache invalidation approaches

Approaches	PER	SB-PER	CB	ACB	TS	Hybrid Techniques	Co-operative Techniques
Stateful	No	Yes	Yes	Yes	No	Yes	Yes
Stateless	Yes	No	No	No	Yes	Yes	No
Pull	Yes	Yes	No	Yes	Yes	Yes	Yes
Push	No	No	Yes	No	No	No	No
IR Based	No	No	Yes	Yes	Yes	Yes	Yes
Piggy Backing	Yes	Yes	Yes	Yes	No	Yes	Yes
Time Stamping	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Broadcasting	Yes	No	Yes	No	Yes	No	No
Multicasting	No	Yes	No	Yes	No	Yes	Yes
Synchronous	Yes	Yes	Yes	No	Yes	No	No
Asynchronous	No	No	No	Yes	No	Yes	Yes
Strong Coherence	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Query Delay	Higher	Higher	Higher	Higher	Moderate	Moderate	Small
Round Trip Time (RTT)	Higher	Higher	Higher	Higher	Moderate	Moderate	Small
Congestion	Higher	Higher	Higher	Moderate	Moderate	Moderate	Small

which have been described in literature in Chapter 2, TS [23] is one such technique which records all updates pages for IR and broadcasts it to MT. It has fixed size IR. In Update Invalidation Report (UIR)[24] technique a smaller fixed size m UIRs are sent between BT times. SAS [28, 29] uses group Invalidation report (GIR) [28] which has longer delays and IRs. Many of these proposals were based on IR to handle frequent disconnections [23]. Periodically broadcasting of IRs is done by server, to inform about updates occurred during last time to the clients [16].

1.8 Motivation

Most of the above techniques [24, 23, 28] suffer from longer query delays and large bandwidth consumptions due to the longer and fixed sized IRs. So, in such as case, MTs can't reply for query till its cached data is validated by IR sent from server. Hence, Longer IRs consume more bandwidths and increases the query response time for MTs. Authors in [24] proposed an UIR based cache technique in which a server sends UIRs after every L/m^{th} seconds here L is broadcast interval and m is sub interval for UIRs. This may decrease query response time to the extent of hit occurrences, due to UIRs on MT. But the replication of UIR instead of IR consumes a lot of bandwidth and it also increases the congestion in the network. Also, it has been observed from the above discussion that the only update in data should not be criteria for it to be included in IR. It increases the size of IR and technique may loose its effectiveness. As hot data is data which is accessed more frequently by the clients, so including hot data in IRs sent from the server can reduce the size of IRs and at same time it can be an effective option to increase the hit ratio. So, authors in [30, 31] proposed a Dynamic and Adaptive Cache Invalidation Technique (ACIT) which was based on counter to identify a hot data item. For hot data identification, a counter is maintained for each data item on server and the correct information is maintained so that data item is cached on each MT but, it generates large overhead when update rates are more on server. Therefore, these approaches may not be suitable when the update rate is higher in the wireless environment.

1.9 Problem Formulation

In wireless environment, with an exponential growth in MTs such as laptops, PDAs, and smart phones users wants to have various services from different places across the globe. But, as wireless environment has limited bandwidth, frequent disconnections, and slower Uplink channel so, effective mechanism to transmit information from servers to various clients is required so that lesser Uplink traffic is generated. Caching the frequent data items is an effective technique to improve performance in such an environment but maintaining cache validity in an efficient way is one of the major challenges. Hence, efficient cache invalidation techniques are required to improve the performance of various services in wireless environments. without negotiating with the quality. To adapt to the most dynamic environment, a CIT techniques should have a strong mechanism for handling the disconnection of MTs using server and to deliver the continuous services with fast response time. Efficient cache invalidation scheme optimize the size of IR report due to which lesser bandwidth is required in wireless environment by the clients. As Uplink channel is slower in such an environment, so a CIT reduces the Uplink requests by validating the existing cache contents of the client. In such a case, only a few Uplink requests need to be sent. Adaptive nature of IRs and broadcasting time can make schemes more effective which can reduce the Uplink requests. This increases the hit ratio of cache. An efficient CIT scheme is an informed approach for carrying data in IRs which may be needed by most of clients. Hence, the current research work focuses on formulation of adaptive and informed cache invalidation algorithms by using state of database, nature of data and broadcasting time intervals.

1.10 Research Goals / Objectives

From the above discussion and analysis, following are the objectives of the research proposal in this thesis.

i) To study and review various CITs in literature for wireless environment.

ii) To design efficient caching algorithms for wireless environments.

iii) To verify and validate the proposed CIT-based algorithms and compare their performance with other existing algorithms.

1.11 Summery of Contributions

We propose efficient cache invalidation schemes for wireless environments with stateless servers having hot data at any instant. We have used a similar type of approach for identification of hot data as used in [32], but we have used a separate mechanism for checking update rate of data on server so that broadcast time for server can be regulated. The usage of adaptive size of IR reduces the bandwidth consumption in the proposed scheme. Moreover, we have sent IRs of hot data along with data itself. So, it increases the hit ratio, and reduces the query response time. The performances of the proposed schemes are analyzed by using extensive simulations which results a decrease in query latency, and bandwidth consumption.

1. To accomplish the first objective, a wide range of available proposals in literature have been studied. After cavernous mining from the various research proposals, it has been observed that there is still some gaps which need to be filled by designing adaptive techniques for CITs. Few such gaps are discussed in Chapter 2.

2. From the available literature, we have identified parameters such as- Dynamic IR size, hotness based on thresholds, and dynamic/discrete broadcast times. These parameters are used for designing new algorithms in the proposal and are explained as follows (Detailed explanation of following points are given in Chapters 4 and 5).

    - **Hot Data Maintenance**
      On the server side, data hotness is maintained which is accessed frequently by many MTs. So, we have identified the hotness of data in the proposed scheme. Hotness of the data is decided using two factors number of requests for a data item arrived at the server in last broadcast time interval and recent history of hotness for data.

    - **Adaptive IR Generation**

IR records data *ids* of hot updated data only for a time period of current broadcast time, i.e., t_{bt}. So, for time $(w-1)t_{bt}$ where w is the no of broadcast intervals for carrying historical changes, the data *ids* of such hot data with their time stamps are also included in IR. It keeps historical updates of data on server and also helps disconnected clients to validate their cache up to $w \times (t_{bt})$ times. This IR is broadcasted to BS. Because IR has only hot data interval of t_{bt} seconds so, size of IR is optimal and broadcasting it to BS doesn't consume more bandwidth.

IR contains information in 2 blocks- first block stores information of current broadcast time interval t_{bt}. This block contains information such as- Data *ids* of hot updated data, original $page(d_x)$, and time stamp. On the server, update rate (U_{rate}) of data pages is tracked which is used to set the mode of operation of scheme using value of Mode Bit (MB) as 1 or 0. In second block, data *ids* of last $(w-1)$ broadcast time, hot data *ids* with their timestamps are stored. The data of Block one is broadcasted to MTs because it is the hot updated data from the server and is required by MTs in near future. Then, Uplink and downlink requests are generated for it. So, it is beneficial for MT, if server sends updated hot data which is invalid in local cache of a MT. Then, MT pre fetches that data for future uses, which makes the down link bandwidth effective and reduces future Uplink and down link requests for that hot data.

Also, IR is divided in two parts on BS as- Current Invalidation Report (CIR) $(wL - t_{bt} < ts <= wL)$when time stamp t_s is between current broadcast time interval from initial to final value so ts exists lesser than $wL - t_{bt}$ and greater than wL, and History Invalidation Report (HIR) $(0 < ts < wL - t_{bt})$, when ts is older than current broadcast interval. Then BS, sends first block as IR to MTs and real hot updated data page d_x to MTs.

- **Adaptive Broadcast Time (BT) - Fast or Normal Mode Operations**

  As server keeps track of update rate of current BT interval as U_{rate}, so, a threshold value for update rate (U_{th}) is kept. After this update rate, BT is regulated to smaller intervals and MB is set to Fast Mode. If update rate is higher then, smaller BT is kept. Otherwise, earlier BT works fine. Due to the adaptive BT, the tuning size of IR is not increased considerably and it has an optimal value.

- **Disconnection MT Support**

  As MTs are limited with the battery life so, an energy efficient scheme is required to support the disconnected operation. For energy efficiency, MT can disconnect actively (voluntarily) or passively (Due to some failures such as link failure).

  For active disconnections upto t_{bt}***:*** An actively disconnected MT reconnects with AP can have its first query for reply, so, it listens to current IR from AP and current updated hot data list to validate its cache and pre fetches that data.

  For Passively Disconnected for Longer durations up to wL ***seconds.*** : It makes use of second block of IR, i.e., HIR report.

  The novelty factors of the proposed work are given as follows:

  - **Dynamic Invalidation Report:** Dynamic Invalidation report is generated as per the need of the MTs and hotness of data. So IRs are having variable sizes. So, such IRs do not lose their effectiveness and consume less bandwidth. Effectiveness is maintained because IRs are having only hot targeted data update log.
  - **Dynamic Broadcast Time**: As server keeps track of update rate and a threshold is kept. So, when update rate is higher than threshold then the mode bit is set as 1 and the proposed scheme switches to smaller BT. Otherwise BT is kept as L seconds.

To achieve second objective, we have designed a Dynamic Cache Invalidation Techniques for wireless environment(Infrastructure based wireless and Internet based vehicular ad hoc network environment). These schemes solve the issues of earlier CIT techniques having larger IRs, fixed BT, consuming more bandwidth, and more Uplink requests. But, the proposed techniques have dynamic and smaller IRs and BTs. So, they use lesser bandwidth and can carry smaller size

IRs. Cooperation among upper components of architecture like server, gateway foreign agent, AP, and BS is utilized to reduce Uplink requests.

3. Third objective is achieved by evaluating the proposed algorithms in various scenarios in comparison to different CITs with respect to various parameters. The new algorithms are compared with existing cache invalidation techniques such as UIR, SAS, Cooperative Cache Invalidation (CCI), and Enhanced Cooperative Cache Invalidation(ECCI) Techniques and are compared using various performance metrics such as- average query response time, IR size, and Uplink requests. Each metric is measured by varying parameters such as object update rate, and query arrival rate. The results obtained are illustrated in Chapters 4 and 5. The results obtained show that the proposed algorithms outperform the other algorithms with respect to various selected parameters.

1.12 Organization of Thesis

- The first chapter elaborates the details of the cache invalidation techniques and the historical background. Its significance for the improvement in quality of service in wireless environment is also discussed. Introduction gives the implication of stateless or stateful servers, invalidation reports, broadcast time and update rates. Then, we have given motivation for this work, formulate the problem and objectives of the proposed work. Then, chapter ends with contributions and organization of thesis.

- The second chapter summaries the various CITs for wireless environments. Classification of CITs is done on the basis of IR content type and static and dynamic nature of IR, broadcast time and disconnection time duration (DT). We have done rigorous study of many existing invalidation techniques. A comparison table is also made to compare their features. This chapter helps to find various gaps between existing techniques and proposed techniques. Then at the end, simulation environment of NS2 for web caching is discussed.

- In third chapter, we have given description of tool used for simulation in work NS2. A introduction of tools and its familiarizing details of environment is given.

Animator tool NAM of NS2 is described. Wireless model networking details are given, about MobileNode class to setup a MT and routing protocols description. Then how node movement is achieved using setdest.

- In fourth chapter, we have mentioned network and mobility models used in proposed techniques. Details of new proposed ”Dynamic Cache Invalidation in Wireless Environments” is given with algorithms and flowcharts. Data Structures used are also described. We have used default configurations for Broadcast Manager, Query, Cache, Cache manager and Uplink Queue. We have used history register to keep track of history for each data item and depending upon it MB is generated. Hot data selector for keeping tack of hotness and recency. Analytical and experimental work analysis are given for proposed technique. An analytical model is given to show variation of various parameters and their relationship with Average Query Response Time (AQRT). Experimental work is done using network simulator NS2, version 2.35. The proposed scheme is compared with existing cache object based and group based invalidation techniques using metrics such as AQRT, Invalidation Report Size, and Uplink requests with respect to query arrival rate, and update rates.

- In fifth chapter, another technique in wireless environment with mobility of the clients is proposed. Cooperative gateway cache invalidation in Internet based vehicular ad hoc network is elaborated in this Chapter. It has a cooperative gateway approach to invalidate the cache contents of vehicles. Chapter started with motivation for this work, our contributions, network and mobility models. Then, problem is formulated with objective function and associated constraints. Detailed design of the proposed technique is given with suitable algorithms, flowcharts, and sequence diagrams. An analytical model is also given for query response time and Uplink requests. Then, at the end of the chapter, performance evaluation of proposed technique is done using simulations. Proposed technique is compared with existing techniques such as CCI, SAS, UIR, and ECCI based on various metrics such as AQRT, Invalidation Report Size and Uplink requests with respect to query arrival rate and update rates.

- Sixth chapter concludes with detailed findings of the proposed dynamic algorithms in wireless environment. Results obtained in chapters 4 and 5 concluded that the new proposed techniques perform better than the existing benchmark schemes used in comparison. This validates and verifies the proposed techniques for cache invalidation in wireless environment. Lastly, we have also provided directions for future research work in this area.

Chapter 2

Literature Survey

A number of research proposals exist in literature related to caching for efficient data dissemination for highly MTs. This chapter explores various existing state-of-the-art existing proposals in literature with their relative advantages and disadvantages.

2.1 Classification of Cache Invalidation Techniques

Cache Invalidation Techniques are classified in two categories as shown in Figure 2.1. It shows the classification of CIT techniques in two different categories. First category consists of IR Content Type Approaches and second category consists of IR-BT-DT with static and dynamic nature. These are described in details in the next Section.

2.2 IR-Content Type Techniques

The first category of CIT is based upon IR contents and various sub categories in this category are explained as follows.

2.2.1 Object Cache Invalidation (OCI)

The server sends an IR of objects, containing information of the updated data item objects, that has been updated from the last broadcast time. The OCI is used in TS[23], UIR[33], Counter Based Stateful Invalidation Technique (CBSFI)[30, 31], Reduced Improving Handling (RIH)[34, 35], Counter Based Stateless Invalidation Technique (CBSLI)[36, 37], Selective Adaptive Sorted(SAS)[28, 29], Adaptive Data Dividing

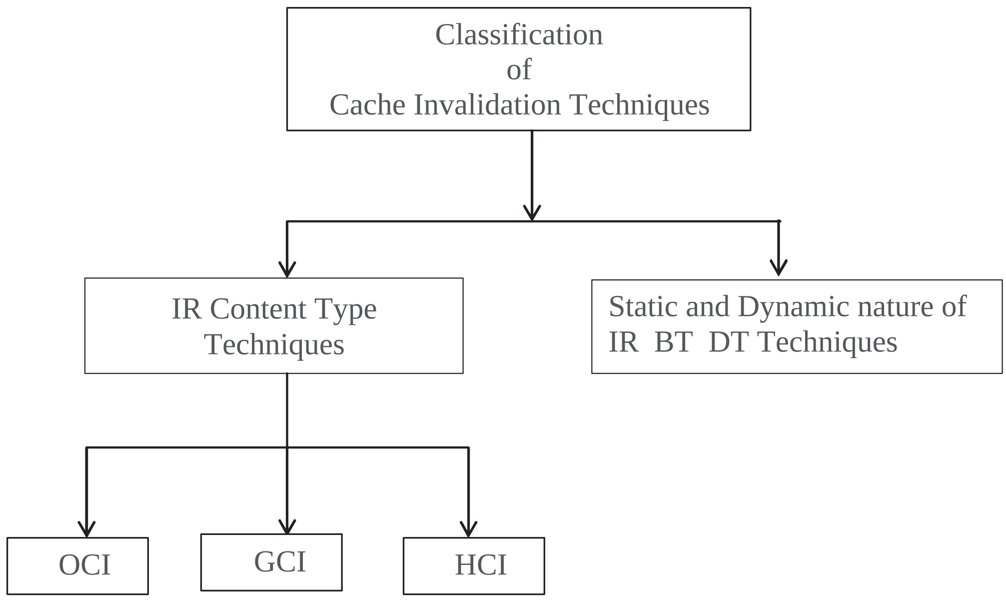

Figure 2.1: Classification of Cache Invalidation Techniques

(ADD) [38], Data Validity Defining (DVD)[38, 39, 40], CCI[41], ECCI[41]. All these techniques are explained in details in the coming Sections.

2.2.2 Group Cache Invalidation Report (GCI)

Data items in the server are categorized into disjoint data groups and then a Group IR is broadcasts from server. It has invalidation information of a complete disjoint group rather than a data item thus reducing the size of IR report. GCI is a type of GIR)[19], which is explained in the coming Section.

2.2.3 Hybrid Cache Invalidation Report (HCI)

HCI is a combination of OCI and GCI in which the two reports are broadcasted by server ie. GCI and OCI [20, 42]. It is used in HCI, Selective Cache Invalidation (SCI)[43], and SAS[28, 29]. All these techniques are explored in the coming Section. Table 2.1 describes the classification of CIT based upon the contents of IRs.

Table 2.1: Classification of CIT as per their contents of IR

Sr.No	IR Content Type	Techniques
1.	OCI	TS[23], UIR[33], CBSFI[30, 31], RIH[34, 35], CBSLI[36, 37], SAS[28, 29], ADD[38], DVD[38, 39, 40], CCI[41], ECCI[41]
2.	GCI	GCI, GIR[19]
3.	HCI	SAS[28, 29], HCI[20, 42], SCI[43]

2.2.4 Discussion on IR-Content Type Techniques

As IR report size is very large in OCI techniques, **so they carry longer delays in query response to MT.** OCI techniques have all the updated pages information so they are quite good in invalidation and they don't provide any false invalidation. They can support very small disconnection times for MTs. On the other hand GCI techniques have smallest IR sizes but they exhibits false invalidations of cache data of MTs which is not acceptable as shown below in Table 2.2. Due to longer update time coverage they can provide long disconnection times to MTs which is of no use against false invalidations. The false invalidations can be minimized by HCI techniques. Being dynamic and adaptive in IR sizes, SCI and SAS provide lesser query delays and can support longer disconnection times for MTs, while SCI has longest IR size so is unfavorable for wireless environments. Another alternative for reducing size of IRs is to keep optimal size of cache [44]. SAS has dynamic small and adapted IR due to which it can provide least query delays and faster response. Concept of dynamic IR was introduced in SAS [29], which supports longer disconnection times for MTs. Table 2.2 provides the detailed comparison of IR contents type techniques based upon various parameters.

Table 2.2: Comparison of IR Content Type Techniques

Techniques	Report Size	False Invalidation	Query Delay	Disconnection Time
OCI	Large	Nil	Long	Least
GCI	Small	Yes- More Prone	Long	Long
SCI[43]	Largest	Less	Less	Longer
SAS[29]	Dynamic-Smaller	Nil	Lesser	Longer

2.3 Categorization Based Upon Static and Dynamic Nature of IR, BT, and DT

This categorization is based upon the contents of IR, BTs and DTs with respect to their static and dynamic nature. The detailed description of these techniques is categorized in Figure 2.2.

2.3.1 Invalidation Report Based Techniques

All the techniques that are having IR based techniques for maintaining cache invalidation are categorized under this category. There are various techniques that have a single or many IRs in which some are having static and others are having dynamic contents. By keeping dynamic contents of IR, we can have smaller size of IR which results less congestion in the network. These techniques are discussed in [45, 46, 47, 48, 49, 50] having shown dynamic nature of contents, and IR. These are discussed as follows.

1. Static/Fixed Size IR Techniques (SIR): CIT that have fixed size IRs or have static IRs are TS [23], Bit Sequencing(BSeq) [18], GIR[19], DVD[38, 39, 40], CCI[41], and ECCI[41].

2. Dynamic/Variable IR Techniques (DVIR): Techniques having more number of IRs and adaptive IRs are discussed as follows.

    i) Different IRs with Fixed Sizes (DVFIR): It includes all those techniques which are having more than one IRs but sizes of IRs are fixed. Popular techniques in this category are UIR [33] and RIH [34, 35] HCI [20, 42], and SCI[43].

    ii) Different IRs with Variable Sizes (DIRV): It includes all those techniques which are having more than one IRs and IRs may be of different sizes. Techniques using DIRV are as follows- IAVI[51], SAS, and ADD.

    iii) Different IRs with Informed Contents (DVIIR): It includes techniques having different IRs but content of IRs are adaptive and informed depending upon some knowledge parameters such as type of data handling as hot or cold. Techniques using DVIIR are CBSFI, and CBSLI. A very useful contribution towards dynamic web content generation is given in [52] and another alternate approach of effectiveness of IR using hot data is to use an optimal cache replacement for higher hit ratio is explained in [53] for predictive cache replacements.

CIT techniques depending upon IRs have issues with size of IR. As size of IR reports increases considerably, there exists a congestion in downlink channel. So, static sizes of IRs use the fixed sizes even if there are less updates of data pages on server, however,still the same size IRs are sent to MTs. Dynamic sizes as per need of users, i.e., considering hotness of data, update information of only hot data is included in IRs. So size of IRs is can be reduced. This approach is taken in proposed techniques to have small size of IRs.

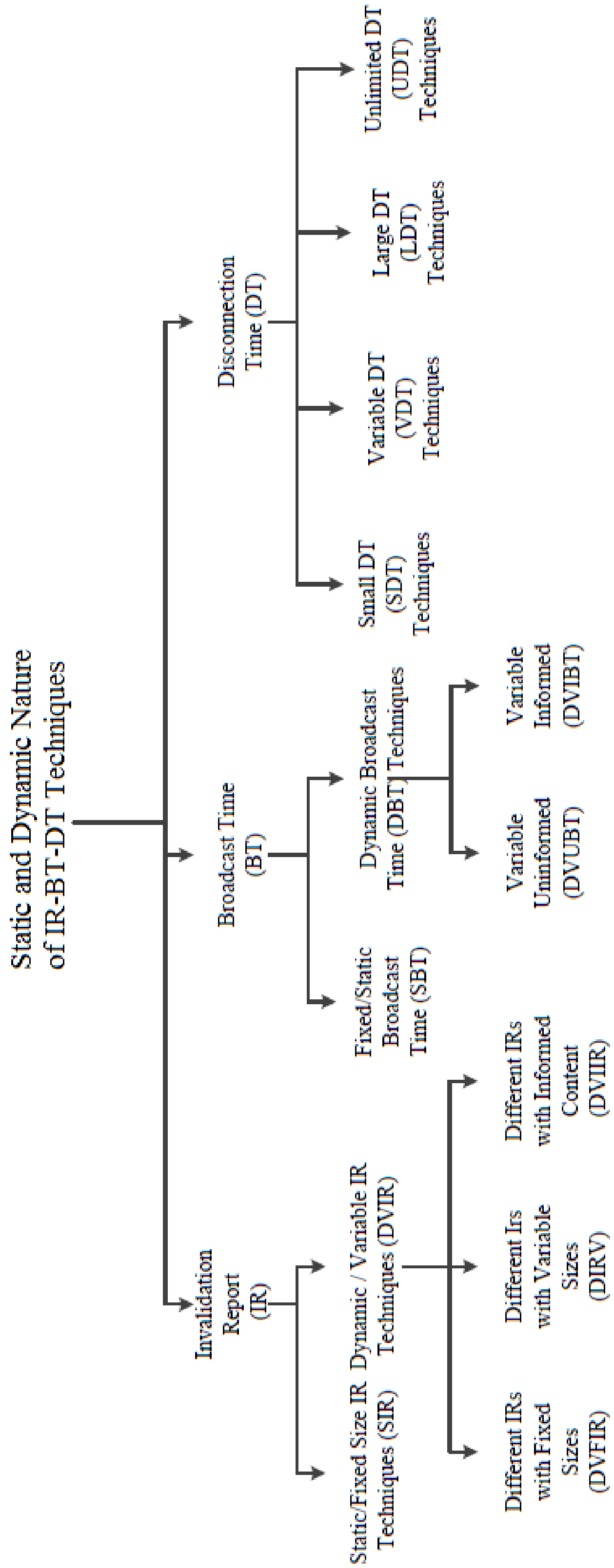

Figure 2.2: Dynamic IR, BT, and DT Techniques categorization

2.3.2 Broadcast Time Based Technique

In this category, IRs are sent at fixed or variable broadcast time intervals. These are sub classified as follows:

1. **Fixed/Static Broadcast Time (SBT):** It includes all those techniques in which IRs are broadcasted after a fixed time unit. Techniques using SBT are TS[23], BSeq[18], GIR[19], HCI[20], CBSFI[30, 31], SCI[43], DVD[39], CCI[41], and ECCI[71]

2. **Dynamic Broadcast Time Techniques (DBT):** These cache invalidation techniques have variable broadcast time for their IRs and are described as below.

   - **Variable Uninformed Broadcast Time Technique (DVUBT):** It include those techniques which are having discrete broadcast time. The discrete time is not increased or decreased on basis of information/knowledge regarding update rate. Popular techniques in this category are UIR, RIH, and SAS

   - **Variable Informed Broadcast Time Technique (DVIBT):** It includes those techniques which are having variable broadcast time due to informed knowledge regarding update rate of traffic on the servers. In this category, popular techniques are -DVIBT, IAVI, and ADD.

Keeping fixed BTs may make size of IRs longer and unmanageable due to higher update rate of data on server. So technique may loose its effectiveness due to longer size of IRs. So, there is a need of dynamic BT, which is used in the our proposed scheme depending upon update rate of data on server. BT is reduced for higher update rates and remains normal otherwise.

2.3.3 Disconnection Time Techniques

All those techniques which support small, longer, variable and unlimited disconnection time of clients are included in this category. These techniques are explained as follows.

1. **Small Disconnection Time Techniques (SDT):** All techniques which can support a small or fixed quantum of disconnection time of MTs and still they

can validate the cache of MTs fall under this category and are described as- TS, IAVI, CBSFI, and CBSLI.

2. **Variable Disconnection Time Techniques (VDT):** All techniques which can support variable disconnection time for MTs and still able to validate their cache after reconnection with the APs fall in this category. It includes techniques such as- GIR, HCI, UIR, SCI, RIH, SAS, and ADD.

3. **Large Disconnection Time Technique (LDT)** All techniques which can support longer disconnection times for MTs fall in this category.

4. **Unlimited Disconnection Times Technique (UDT):** Techniques which can support unlimited disconnection times for MTs and still they can invalidate the cache contents of MTs after reconnection fall in this category. Techniques under this category are- DVD, CCI, and ECCI.

Large disconnection type techniques are suitable for MTs, so that they can remain in power saving mode to save battery power and can still invalidate their cache after reconnection. So it is limitation of techniques with smaller disconnection times. In proposed technique, disconnection time for MTs is increased.

2.4 Detailed description of CITs

Data caching is a concept of storing the copy of frequently accessed data in neighboring nodes or locations for rapid access. Once the data is stored in the local cache, the future requests fetch the data from the local cache rather than from the original source. This reduces the average access time of the data and effectively improves the system performance. Data caching methods are broadly classified into two types- data caching in wired networks and data caching in wireless networks.

The World Wide Web (WWW) is a huge distributed information system, shared objects can be used. The popularity of the Internet and WWW continues to grow since the most of the popular applications running on the Internet and WWW have an exponential growth in size, which results in network congestion and server overloading. With rapid growth of the WWW, its usage is inexpensive now, and accessing information is faster than by using any other mean. WWW has documents which can be used by wide variety of users, for business, entertainment, news, education,travel, sports, shopping, research, stock market, shopping, weather forecasting, maps, and multimedia. Internet capacity increases by 60 percent per year, as most of services are moving onto web, the demand of bandwidth will increase sharply in near future. De to such rapid migration of applications and services WWW would become congested soon which may result in reduction of efficiency of any solution in this environment.

One expensive way is to update the network and server bandwidth to match the client demand. An alternative and effective less expensive way is caching, which reduces the network bandwidth usage and server workloads, by migrating the server files closer to clients, so that future demands can be full filled from clients instead of server. Caching of popular objects at close to client locations is identified as one of the effective solutions to alleviate web service bottlenecks and to reduce Internet traffic, which improves the scalability.

This section provides the detailed description of various strategies used for caching and are described as follows. Table 2.3 provides the categorization of various CIT techniques used in literature.

2.4.1 Time stamp based Technique (TS)

Barbara and Imelienski [23] proposed a cache coherence mechanism for wireless environments. The proposed technique was periodic and having synchronous broadcast of IRs similar to CBs by a server. The broadcasting Timestamp [23] strategy was the first object-based, stateless server, synchronous IR-based invalidation strategy. In this strategy, at every time interval equal to L seconds, IR is broadcasted by server. IR contains information of the data items that are updated in the last w broadcast intervals as shown in Figure 2.3.

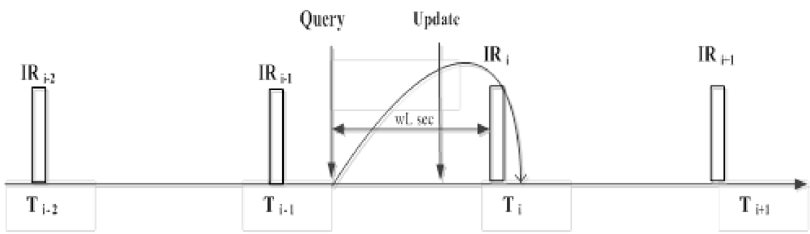

Figure 2.3: Broadcast Time Stamp Technique

Table 2.3: Various CIT Techniques

Sr.No	Year wise	Authors	Technique
1	1994	Barbra and Imielinski	TS [23]
2	1996	K.L. Wu, P.S. Yu and M.S. Chen	MTS [54]
3	1997	Jin Jing & Elmagarmid	BSeq [18]
4	1998	Qinglong Hu and Dik L.	AIS[55]
5	1999	Cai & Tan	GIR[19]
6	1999,2004	Tan,Yuli	HCI [20, 42]
7	2000	C.H. Yuen,E.Chan	IAVI[51]
8	2000,2003	G.Cao	UIR[33]
9	2001,2002	G.Cao & C.Das	CBSFI[30, 31]
10	2001	Tan, cai and Ooi	SCI[43]
11	2005	N.Chand, R.C, Joshi, M.Misra	RIH[34, 35]
12	2008	Denko MK and Tian	COOP[56]
13	2009	Chao-Chin Wu	CBSLI[36, 37]
14	2010	Haidar Safa, Hassan Artail	SAS[28, 29]
15	2011	Po-Jen & Yu-Shin Chiu	ADD[38]
16	2011	Po-Jen & Y.S. Chiu, F. & Lim	DVD[38, 39, 40]
17	2012	Suno-Lim, Chita R Das	CCI[41]
18	2012	Suno-Lim, Chita R Das	ECCI[41]
19	2014	Neeraj Kumar, Jong H.Lee	P2PCCI[57]
20	2015	Neeraj Kumar,	QHWC [58]
21	2015	Parvathy P Ra, Ajish Kumar	2TierCoCS[59]

The Figure 2.3 shows that at time T_i, the server broadcasts an invalidation report IR_i has a list of (O_{id}, T^r_{id}) pairs where, O_{id} is the id of the updated data item and T^r_{id} is the timestamp of data item update such that $T_i - wL < T^r_{id} < T_i$. Time of last broadcast T_{lb} is also stored on MT. Along with it, for each cached data item O_i, id and the timestamp (T^c_{id}) is stored in IR. When MT receives the IR_i report at T_i, depending upon state of MT, following possibilities may occur:

- If MT has disconnected before the last broadcast time, i.e., T_{lb} $(T_{lb} < T_i - wL)$ then it drops its entire cache because the DT is more than wL seconds and all updates may not be covered in the IR_i.

- If $(T_i < T_{lb} < T_i - wL)$, i.e., disconnection interval is less than wL seconds, then MT starts invalidation process. For all cached data objects, it checks timestamp value with the IR_i timestamp value. If $(T^r_{id} > T^c_{id})$, then data item is marked as invalid otherwise, it is marked as valid

IRs carry information about all the data items that have changed during the last wL seconds and all MTs perform caching of recently accessed data reports. A query is answered only after receiving an IR and an Uplink requests are made for data items that have been marked as invalid. This technique ensures that user always gets the current information but has a long latency. This is because the client must wait till the broadcast of the next IR before it answers the query. No state needs to be maintained at the server about the data items being cached by each user due to the broadcast nature of the media. This technique handles disconnections of a MT up to a maximum duration of wL seconds. It supports smaller DT so it falls in SDT categories. Reports are broadcasted in every wL seconds. This access latency makes this strategy unsuitable for most of the applications. In addition, the cache has to be dropped every time a disconnection lasts longer than wL seconds, reducing the hit rate of the cache. During each broadcast time, MTs accepts the queries and store them in a First Come First Serve (FCFS) queue and after receiving IR_i updates the status of whole data of cache as valid or invalid. Queries are replied in the same order as they arrived in this technique. Following are the advantages and disadvantages of this technique.

In this technique, MTs remain disconnected and invalidate their cache after reconnecting but query delay is more because query reply waits for the next IR to come. In higher update rate environment, size of IR becomes unmanageable and technique fails with respect to these requirements. So, larger size of IR is an issue in this technique which is reduced in proposed technique by including the hotness concept.

2.4.2 Bit Sequencing (BSeq) Adaptive Invalidation Technique

In this technique[18], data items are not updated frequently on the database server. Server broadcasts report on the periodical basis. Mainly it works in three steps. Firstly, it uses a bit sequence naming technique to refer the data items in the report. Each bit in a bit sequence represents a data item in the database. Secondly, it uses an update aggregation time stamp technique to group the data items and gives only one time stamp for a group of data items instead of individual. Third, it is a hierarchal structure of Bit Sequences (BSeq) technique to link a set of bit sequences so that this structure can be used by MTs for different disconnection times. It optimizes the IR size to keep its effectiveness. Figure 2.4 shows various operations performed in this technique. It manages the cache and query reply as follows.

If servers has N data items then they periodically broadcasts the IRs to answer a query that a MT listens to the next IR and checks for validation of its cache contents. If requested page is valid then servers answers immediately otherwise, MT queries the server for that page. Follwoing optimizations are used in the technique.

Optimization procedure

a) Bit Sequence Naming: Each bit in a BSeq denotes a data item in the data base. Each bit position in a BSeq is used as an index to a data item in the data base. So, instead of using number of bits to represent data *id* in IR, it represents a data item by a single bit'1' in a BSeq which means that data item is updated since last broadcast time and '0' means no updates in that data item.

b) Update Aggregation: One Timestamp is used for updated items in one BSeq. One timestamp specifies the time of all the updated data items in one BSeq and it is the most recent timestamp of the updated data item. So, it effectively saves $(D - 1 \times WORDLENGTH)$ bits only for timestamps.

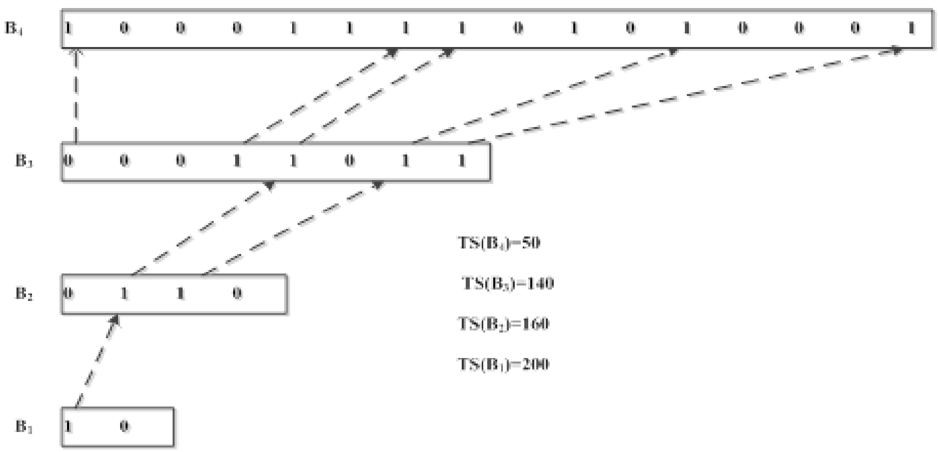

Figure 2.4: Bit Sequencing Time stamping Technique

c) Hierarchal Structure of Bit Sequences: To adapt to variable disconnected clients, hierarchal bit sequence is used in this technique. Multiple levels ($log(N)$ levels) of BSeq are used with different timestamps and sizes are used together for N data items in a BSeq of N bits. Total size of these bit sequences can be only

$2N+bTlog(N)$ where b_T is the size of each time stamp. In this hierarchal structure, the highest level BSeq has N bits, and $N/2$ bits are set to 1 which represents the most recent $N/2$ updates of data items. If more than $N/2$ data items are updated then most recent $N/2$ updated data items are marked in BSeq of highest rank which is represented by B_n. The next $BSeq_{n-1}$ can have only $N/2$ bits in which k_{th} bit denotes the k_{th} '1' bit upper bit sequence, i.e., in B_n and so on up to B_1.

Due to adaptive nature of IRs, this technique is quite effective and allows variable and longer DT. Due to longer IR size with higher update rates, this technique fails and may incur false invalidations.

2.4.3 Modified TS (MTS)

Wu [54] proposed a modified version of TS technique. After reconnection it includes cache validity checks. The disadvantages of this technique is more requirement of Uplink bandwidth and an update history window of the past W broadcast intervals ($W > w$ in $GCORE$), which incurs the problem of lesser DT similar to the TS technique, i.e., total drop of cache content occurs, when disconnection time is more than broadcast time interval.

The effectiveness of the algorithm depends on the update pattern on the data items. So, update rate is monitored in proposed technique to meet effectiveness of technique.

2.4.4 Adaptive Invalidation Strategy(AIS)

Hu *et al.* [55] proposed an adaptive invalidation strategies (AIS) which combine the TS and BSeq techniques. Depending upon update and query patters and disconnection time, server broadcasts adaptive IRs. This scheme has three adaptive cache invalidation reports. These IRs are broadcasted by the server according to the update and query rates/patterns and client disconnection time with lesser Uplink cost.

This scheme uses different IR reports depending upon update patters but have not considered hotness of data on server, due to which size of IR may become larger and scheme may lose its effectiveness. So dynamic size of IR is required.

2.4.5 Group Cache Invalidation (GCI)

A group-based [19], CIT strategy reduces the size of IR which in turn reduces the communication cost. It classify the data items on the repository of server into disjoint groups called data groups and each group has a unique identifier, G_{id}. At each time interval of L seconds, the server broadcasts GIR, in which data group id is the report's entity instead of a data item. Figure 2.5 shows the sequence of operations in GIR technique. At time T_i, the server broadcasts the report GIR_i, which is a list of (G_{id}, T^r_{id}) pairs where, T^r_{id} is the time stamp of the most recent data item of the group's data items, e.g., Data group $G1$ contains four data object items $(O1, O2, O3, O4)$. Both $O1$ and $O2$ are updated in the interval$[T_{i-1}, T_i]$ and $O3$ is updated at $T_i - 3$. The time stamp of the data group $G1$ is equal to T_i as recent time stamp is assigned to whole group. In this technique there is a fixed size GIR, so it is a kind of SIR technique. The cache management and query response are performed as follows.

On receiving GIR_i report through MT, group cache validity is checked, G_{id}. When time stamp of the group is greater in report than time stamp of group in MT, then all cache contents of the group are dropped. Otherwise current time stamp is set as T_i. Queries are stored in a list of ordered pairs (Qid, G_{id}) and after checking validity queries are answered. If cache miss occurs for requested data then Uplink request is sent to server for requested data to retrieve it. So, contents of cache are marked as valid or invalid only when MT has disconnected from time 0 to L, so, it supports variable disconnection times but up to a fixed upper limit as L seconds and hence falls in VDT category. Figure 2.5 describes the various sequences of operations to be performed in this technique.

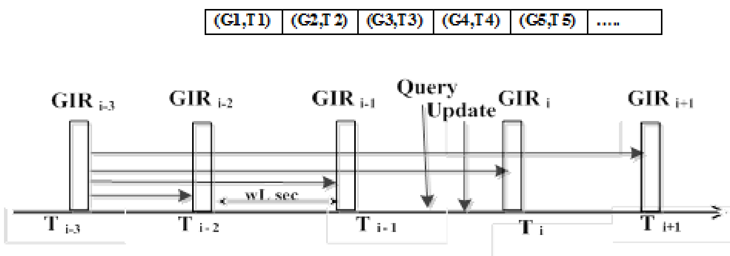

Figure 2.5: Group Invalidation Report strategy (GIR)

Main advantage of this technique is adaptive BT and unlimited DT while due to

GIR false invalidation exist in this technique.

2.4.6 Hybrid Cache Invalidation (HCI)

Tan and Yuli [20, 42] proposed a Hybrid Cache IR based technique. It combines the advantages of OIR and GIR to reduce the false invalidations and permits the clients longer disconnections and invalidate their cache in an effective manner. The server broadcasts two IRs after every time interval equal to L seconds. The first report is GIR_i containing the information of group data with one most recent time stamp and second report is OIR_i which contains the information about the data items. This technique falls in DVFIR and SBT. A hybrid cache technique is also proposed in [60].

This technique has lesser false invalidations and supports longer DT. But, due to longer IR size, it consumes more bandwidth.

2.4.7 Invalidation by Absolute Validity Interval (IAVI)

This technique [51] works on a parameter *"Absolute Validity Interval" (AVI).* MTs can invalidate the AVI if current time from last update time is greater than the AVI. With this self invalidation mechanism, the server sends an IR report. It makes use of real time parameters such as update arrivals rates and update intervals which varies in a predefined manner and can be estimated. Based on these properties, AVI is defined to estimate the average life span of a data item. AVI needs not to be same with the life span of data item. Only a suitable approximation is needed to invalidate the data items through server by sending IR generated on basis of it. So, it uses DIRV with informed contents in IR and sends it at different broadcast times so, it acts as a DVIBT. Figure 2.6 describes various sequences of operations in this technique.

Depending upon AVI, two parameters have been used in this technique which are described as follows.

False Valid Period (FVP): It is used when AVI overestimates the validity period of data item

False Invalid period (FIP): AVI underestimates the valid period of data item called as FIP and is shown in Figure 2.6.

The main advantage of this scheme isthat it has smaller IR size and it consumes lesser bandwidth, higher hit ratio. But, it works only for data having usability trends.

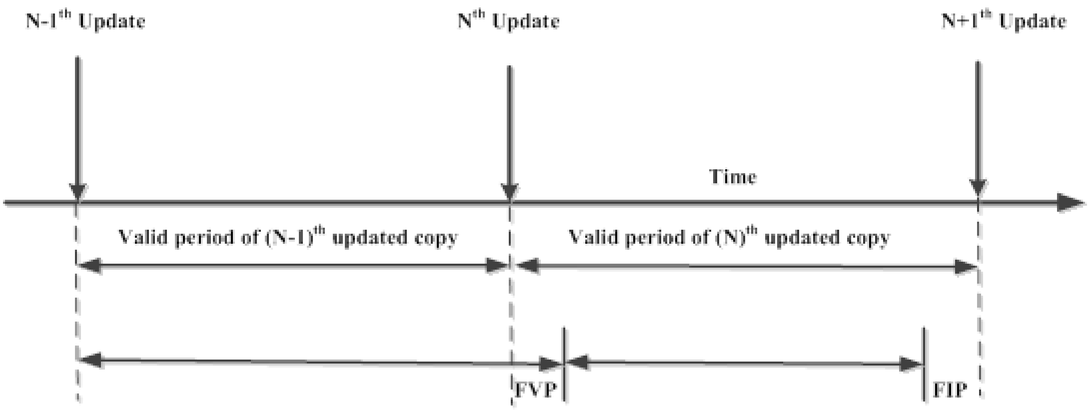

Figure 2.6: AVI Model

2.4.8 Update Invalidation Report based Cache Invalidation (UIR)

Cao *et al.* [33] introduced a new type of IR report called Update Invalidation Report (UIR) along with IRs so, it is a kind of DVFIR. In UIR, the broadcast interval L is divided into $(m-1)$ sub intervals $(m > 0)$ and UIR is broadcasted after every L/m seconds as shown in Figure 2.7. The server broadcasts an IR_i at T_i that contains the history of changes in last wL seconds where, $w > 0$. IR is an OIR based technique having data id and time stamp $< Oid, T^r_{id} >$ such that $Ti - wL < T^r_{id} < Ti$. At time $T_{i,k}$, server broadcasts the k_{th} UIR report denoted by $UIR_{i,k}$, so it has different broadcast times for UIRs and hence falls in the category of DVUBT. This UIR is a list of data items identifiers which have been changed after last broadcasted IR. Cache management and query response are performed as follows. UIR is dependent upon IR reports. So, when a client has not received last IR and receives the UIR then, it ignores the UIR report and wait for the next IR report for answering the queries. Otherwise, client starts invalidating all the data items cached which matches in UIR. If cached data is valid, client answers the query otherwise, sends an Uplink request for the invalid data. To preserve the power, server does not reply the client but it creates a list of pages and then replies to client after next IR as shown in Figure 2.7.

This technique has faster response to query if data is valid. Moreover disconnected clients are allowed upto wL seconds, while more IRs are broadcasted so more bandwidth is used. Also if one IR is lost then this technique may not perform well.

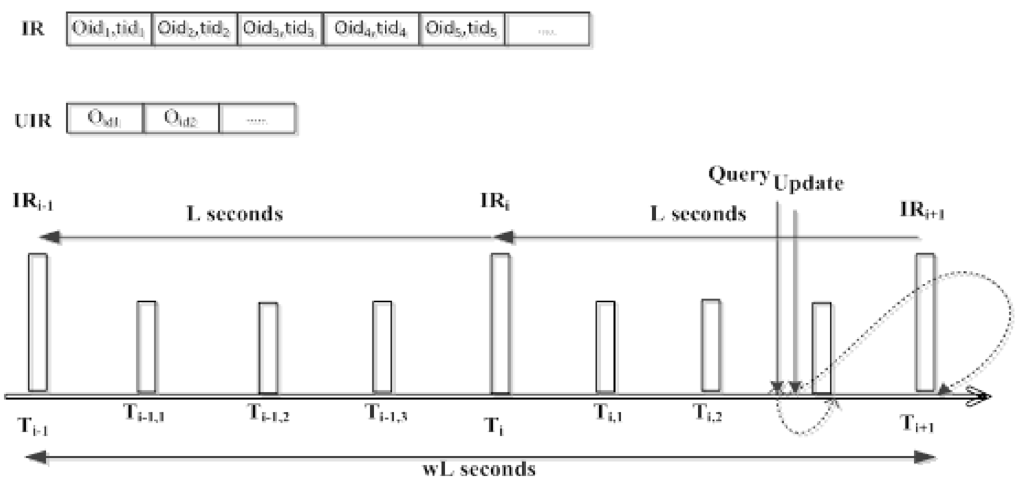

Figure 2.7: Updated Invalidation Report Strategy (UIR)

2.4.9 Counter Based Stateful Invalidation Technique (CB-SFI)

It is a stateful technique that uses a counter to identify the hot data[30, 31]. The server broadcasts the message to all the clients by updating of hot data items and hence falls in the category of DVIIR in which clients pre-fetches the data that are most likely to be used. Server manages a counter associated with each data item of each client. When client requests the data, the counter is increased by one and if counter reaches equal to or more than a threshold value then it is considered as hot data. Client informs the server when it deletes the data from cache, so that counter is decreased by one. A cached item register (CIReg) is associated in server with each client to maintain the state of the client. CIR stores the *id* of data items cached to handle server failure, client failure, and disconnections. During hands-off, this information of client is transferred between two servers.

Hot data changes are recorded in the IRs so it becomes more effective technique. But, it is difficult to maintain the cache state which generates an extra overhead and consumes lot of efforts.

2.4.10 Selective Cache Invalidation (SCI)

Cai *et al.* [43] is a synchronous, stateless server, IR-based technique which has hybrid cache invalidation strategy to minimize the false invalidations while having suitable long disconnections for MTs. It works with two IRs GIR_i and OIR_i. Server broadcasts SCI (IR) periodically after every L seconds so it is SBT. GIR is a triplet (Gid, T^r_{id}, Pid) where, P_{id} is the pointer to the first element of group in OIRi. OIR is a list (Oid, T^r_{id}). GIR contains information of group data which is changed from $T_i - WL$ to $T_i - wL$ such that $W > w > 0$. OIR contains the information of data items update between $T_i - wL$ to T_i as shown in Figure 2.8. As it has two different IRs with fixed sizes so it falls in DVFIR.

Cache management and query response are generated as follows. MT receives the SCI report. If node has disconnected before the history of changes covered in report, then client drops all the cache content. Otherwise, scans the GIR_i for the requested data in a query. For each cached data, client matches the timestamp of the GIR_i , with the cached timestamp of the data . If $(T^r_{id} > T^c_{id})$ then MT drops all the group cached data. Otherwise sets the T^c_{id} to T_i and scans the OIR_i report. MT continues scanning the data in OIR till it meets a partition symbol and compares the cached data time stamp with T^c_{id} and T^r_{id} mark the data as invalid or valid and sets T^c_{id} to current report timestamp T_i. Finally, MT starts answering the query with validated data and sends Uplink request to server for invalidated data as shown in Figure 2.9.

It handles the clients with longer and variable disconnections so VDT. It has faster response to query due to selective tunning of IR. But the report size is larger, so more bandwidth consumption and false invalidation exists.

Figure 2.8: Selective Invalidation Report

2.4.11 Reducing Improving Handling (RIH)

This technique is an enhancement of UIR technique [34, 35]. UIR sends the list of requested pages after next IR and RIH sends the list of requested pages by broadcasting from server in next UIR/IR whichever is earlier. If client is disconnected for some time then in UIR, it drops all the contents of cache of client but this technique being a stateful approach keeps state of disconnected client so it can validate the contents

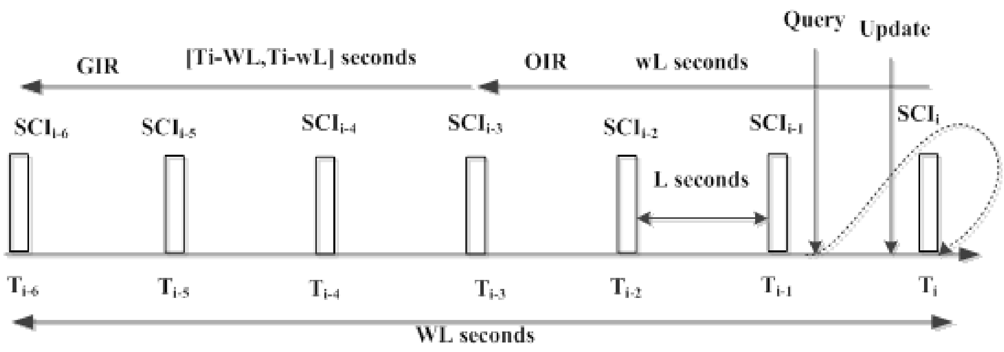

Figure 2.9: SCI Invalidation Report Strategy

of the disconnected client as well.

2.4.12 Counter Based Stateless Cache Invalidation Technique (CBSLI)

Counter Based Cache Invalidation Technique [36, 37] is a stateless technique, so there is no need to maintain cache state of clients, but it maintains the counter for the data items whenever a data item is requested by client. This counter counts only the current requests of a data item during the current broadcast intervals and assigns them in categories of hot or cold as per the threshold values. So, only hot data updated during interval are broadcasted as IR to clients. Now, clients may pre-fetch that data and piggyback the *id* of that hot data with the next request of data to maintain the counter accurately. For the next broadcast interval, counters are again reset to zero. Hot data is kept in IRs and is broadcasted to clients which increases the effectiveness of technique so it falls in DVIIR and SBT categories. Due to static BT, it supports only smaller DTs so it falls in SDT.

2.4.13 Selective Adaptive Sorted Cache Invalidation Technique (SAS)

This technique is a stateless server based HCI technique [28, 29] which makes use of both GIR and Object Invalidation Report(OIR) [61], so it is a type of DIRV technique. In this technique, both reports cover the same history of changes that may be from wL to WL seconds. Server may send a report of covering history of changes of wL initially but as per the need, client can demand larger history changes up to size WL in next report. Hence, it has variable BT, so it falls in DVUBT category. Data items in a GIR and OIR are sorted in descending order as per their timestamps as shown in Figure 2.10. Here, MTs may not validate the same data item twice because it starts scanning the reports with the most recent data items updated.

It is has selective, adaptive and sorted IR, so lesser response time for queries and

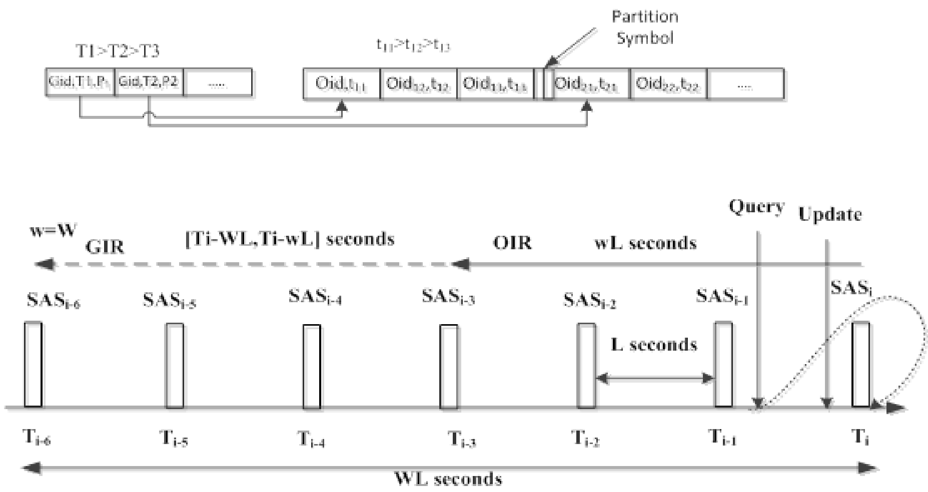

Figure 2.10: SAS Invalidation Report Strategy

smaller size of IRs. It also has lesser false invalidations and due to HIR, longer IRs may exist in some cases.

2.4.14 Adaptive Data Dividing Technique (ADD)

This technique [38] reduces the latency time for query replies. It has two sub techniques which are known as- Adaptive Broadcast Interval and Hot/Cold Query/Update. These are described as follows:

Adaptive Broadcast Interval Technique- This technique divides the size of database into three different ranges as lower, medium and higher. During one broadcast interval, if the numbers of updates are in lower range (lesser number of updates), then broadcast interval is increased. When updates are in medium range, then current broadcast interval is suitable and when updates are in higher range, then broadcast interval is reduced to get the faster response of queries and negotiates with the contents of the IR so that replies can be sent in lesser time. It falls in DVIBT and DIRV.

Hot/Cold Query/Update Technique- In this technique, data is divided into 5 different categories as Hot Query (HQ), Hot Update (HU), Cold Query (CQ), Cold Update (CU) and Remainder (RMA). According to groups, each group of data is assigned a suitable broadcast interval. The server broadcasts five different IRs with different broadcast intervals with priorities $HQ > HU > RMA > CQ > CU$. It is a type of DIRV and DVIBT technique.

It has a variety of BT intervals, so extra overhead to maintain all group categories.

2.4.15 Data Validity Defining Technique (DVD)

This technique [38, 39, 40] is efficient in maintaining data validity after disconnection. A MT can be disconnected in two ways: one is voluntarily for saving battery life called actively and other is due to factors such as media breakdown, lost connection, and failures called passively. These are described as follows.

For active disconnections, after receiving a query after reconnection, client sends timestamp of start and end disconnections along with the *id* of queried data item. Then server, sends the recovery message with the IR report for the client of that disconnected interval along with the queried data. This provides unlimited disconnection times so it falls in UDT technique.

For passive disconnections, after reconnection from a passive disconnection, client sends the checked message to neighboring clients. If they have the valid copy of the data by matching with the last IR, they reply with a confirm message to client. This intercommunication is suitable than Uplink request to server because this is a low power broadcasting. There are some other techniques for energy efficiency and disconnection handling which are also equally good such as [62, 63, 64, 65, 66, 67, 68, 69].

2.4.16 Peer-to-Peer Cooperative Caching(P2PCC)

A cooperative caching technique for data dissemination in urban vehicular communications[57] has been proposed by Kumar and Lee. This P2PCC technique is described using Markov chain model, it has three states, as shown in Figure 2.11.

Its a cooperative approach as queries are shared among neighbor vehicles for answering. Every state of vehicle has a certain probability to pass a query from state to state. The different states can be initial, waiting, update and reply. Jump from one state to another depends upon probability of each state. A query generator is associ-

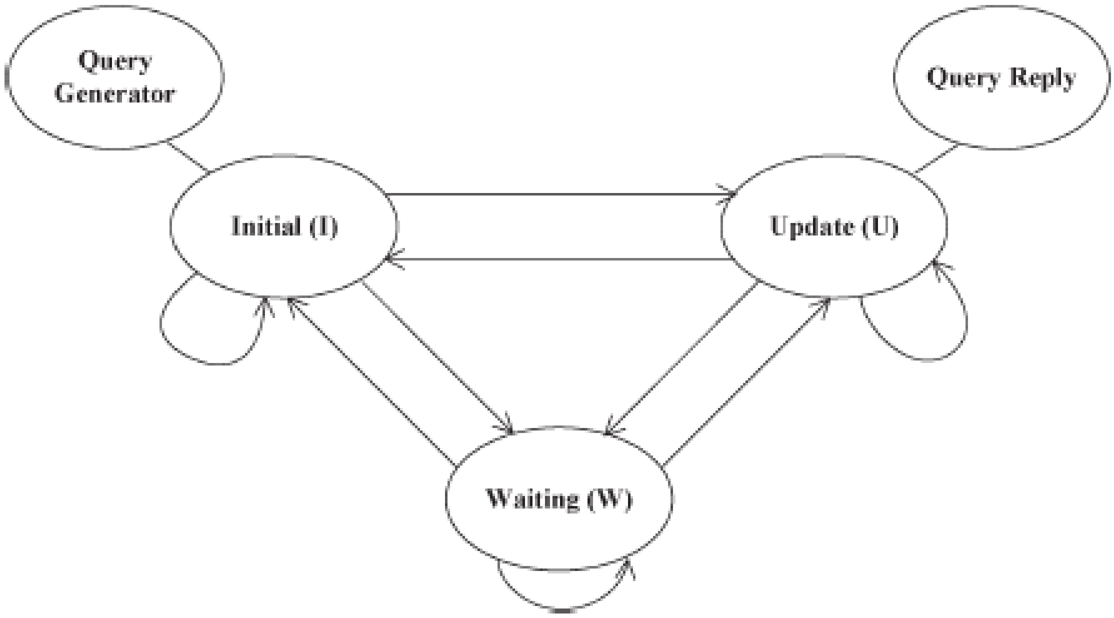

Figure 2.11: Markov chain model with three states: Initial (I), Waiting (W), and Update (U)

ated in initial state, a request is generated, if requested data is available in cache then query is passed to vehicle/client and query is replied and process is over. However, if requested data is not available with vehicle then query is shared among other vehicles until reply from other vehicles cache is received and process goes into waiting state. When query result is received on vehicle then it enters into update state or initial state to receive new query request.

Using markov model probability of different states is computed.A queried data item is looked in the neighboring vehicle's cache. Different attributes of cache query are set like vehicle ID, start and end times. It has considered two processes one is cache query and other is cache update or replacement. For any generated query, a queue with upper and lower threshold is used to keep track of generated queries for same or different data using data ID. For acceptance of new query, queue length has to be with in lower and upper limit of thresholds. Now process one starts, firstly data s looked into local cache if missed then broadcasted query to other vehicles and waits for acknowledgment in a fixed time interval. If no vehicle replies then cal, is dropped. Now, the second process starts after getting acknowledgment, request is forwarded to road side unit (RSU), which in turn is sent to demanding vehicle with its own ID.

Otherwise,vehicle enters into the update state ti get update of update time parameter. If the queue size is more than the maximum threshold size, then again vehicle enters into the waiting state creates some spaces in its cache by doing replacements.

Major limitation of technique is it works in an environment where geographical-based data is required for set of vehicles. Due to this, the performance of the proposed scheme degrades in different network scenarios.

2.4.17 QoS-Aware Hierarchical Web Caching Technique (QHWC)

Kumar and Lee [58] proposed a Hierarchical Web Caching technique. This hierarchical model consists of RSU connected with APs which are located on houses, government buildings or university buildings. Vehicles are at lowest hierarchy which are moving with some velocity. These vehicles are connected to RSUs. Vehicles inside range of AP and RSu can communicate directly but for those nodes which are outside the range can relay their data using intermediate vehicle nodes. This technique takes inputs, users provide ratings to videos, depending upon this, priority of video is increased or decreased. a cost function is used to change priorities dynamically depending upon feedbacks of users. Vehicles can request for video on the move, suitable video is located and pulled from server. The vehicle's location is stored in Clients Internet Address (CIP) variable. Parameters such as Load Utilization Ratio (LUR), downloading and uploading cost, frame length, and mean waiting time are computed. Video streaming quality is computed using parameters flow rate and effective transmission across each channel. If the requested web page found in the cache, it is a hit and the numbers of service requests decreased by 1, and the cache content as stored in the CIP. In other cases, reply was sent to proxy server at level 2 and the associated cost of sending query to proxy server is calculated. The invalidation and migration costs are also computed. Web page was added into cache and cache contents was sent to CIP.

It is a technique in which data can be relayed by the intermediate nodes towards AP. It is for larger data set, i.e., videos. So it has a demand of higher bandwidth.

2.4.18 Cooperative Technique(COOP)

Denko *et al.* [56] proposed a cooperative caching technique, in which the data accessibility depends on the Time To Live (TTL) value. When the TTL is small, pre-fetching provides higher data accessibility. When the TTL is large, most of the cached items remain valid during the simulation, so some of the invalid items need to be pre-fetched, which reduces the effect of pre-fetching. It reveals that the integration of pre-fetching and caching improve the system performance when TTL is small. There is another pre-fetching algorithm [70], where the pre-fetching rules are defined in such a way that when there occurs a cache miss then, the pre-fetch set for the cache missed item are generated which reduces query latency.

2.4.19 Cooperative Caching Technique(CCI)

Lim *et al.* proposed CCI techniques [41] work by cooperation among different level components located at different levels and neighbors. So a variety of Cooperative techniques were proposed in literature [71, 72, 73, 74, 75]. Lim *et al.* [41, 76, 77] proposed most significant techniques for cooperative caching. Among these techniques, cooperative stateful approach [41] is the most viable technique, whose architecture is given in Figure 2.12. Many other effective techniques were also proposed by researchers for Internet based Vehicular Ad Hoc Networks (IVANETs) [78, 79, 80]. The architecture proposed by the authors has a server and agents for location management which works together in coordination for cache invalidation maintenance operations. A list of accessed data items by vehicles is maintained at server and other network agents. Here vehicles act as MTs. As it was a stateful approach so it provides unlimited disconnection times for MTs and is of type of UDT technique. Unlike the most of stateful techniques, the vehicles current location is not tracked by server for IR broadcast. As broadcast time is fixed, so it is SBT and SIR based techniques. Thus, on a data item update, server sends an IR to the Home Agent (HA) rather than directly broadcasting IR to multiple cells, this may increase congestion. Then, HA judiciously filters and does redistribution of IR to appropriate Gateway Foreign Agent (GFA), where GFA can reply queried data page after checking its validity. In this technique, GFAs do not broadcast IR to individual vehicles but replies a vehicle's validity using on demand. The achievements of this technique are as follows:

(i) HA and GFA both maintains location of vehicle, server need not to track vehicle's current location.

(ii) Assumption,since adjacent vehicles have lesser chances to find common data, so same IR broadcasting to vehicles is not efficient in IVANETs.

The detailed architecture is shown in Figure 2.12.

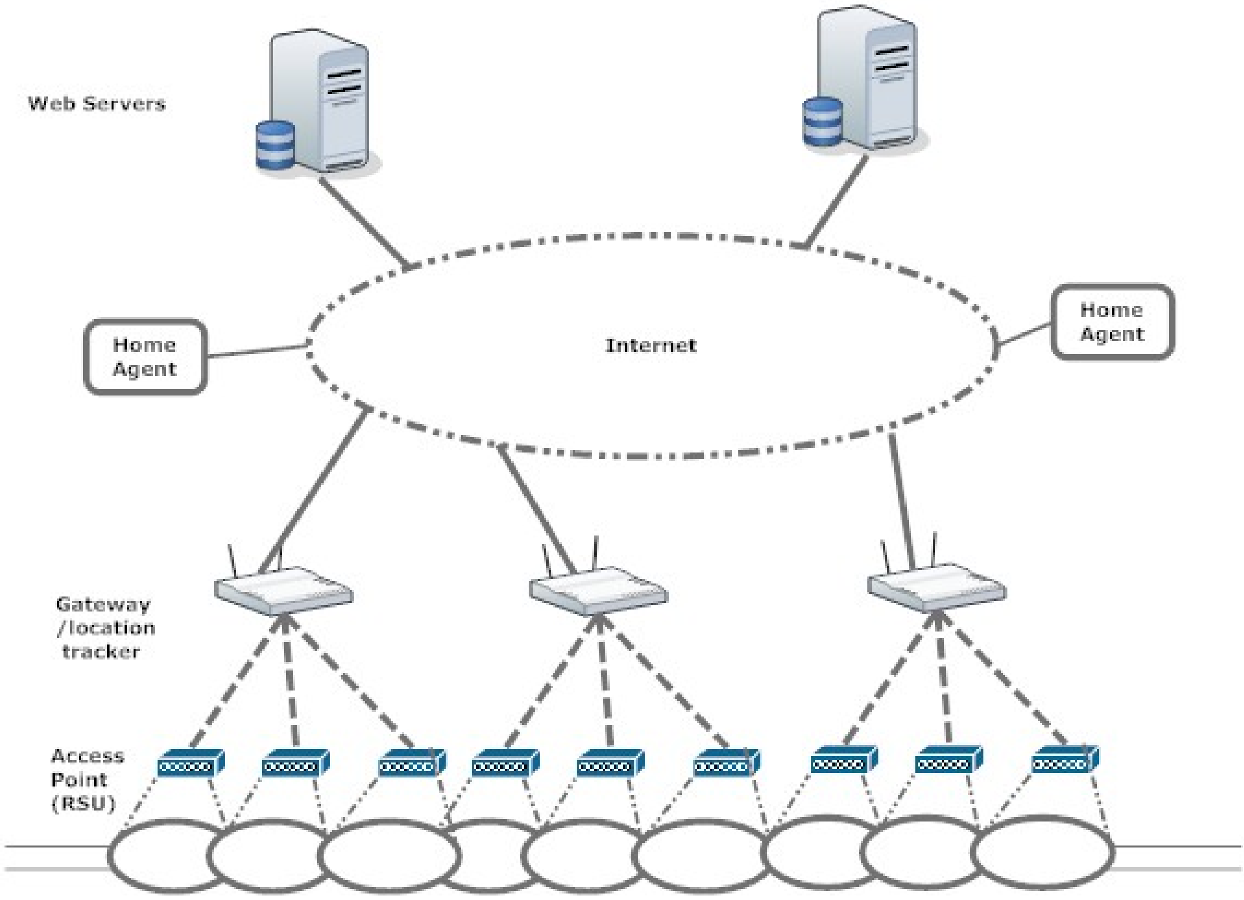

Figure 2.12: CCI Invalidation Strategy

Authors proposed a state-aware cooperative approach coordination of server and location management agents is used for cache invalidation. Server generates IR and sends to HA instead of sending it directly to vehilces. Then HA redistributes IR and sends them to corresponding GFAs depending upon vehicles *id*.

This technique is integrated with a mobile IP based location management. In this method, when a vehicle movement in own region is tracked by GFA, vehicle sends the location update to the GFA. Also, when out side region movement is done by vehicle then, it sends a location update to HA, so that any requested packets can be channelized via GFA and AP.

It has multicasting so saves bandwidth. Also handles handsoffs effectively, but extra infrastructure such as GFA and HA is required.

2.4.20 Enhanced Cooperative Cache Invalidation (ECCI)

Lim *et. al* proposed an enhanced version of CCI technique [41] and is called as Enhanced CCI (ECCI). In this technique, the GFAs are used for prefteching data and to answer future raised queries. Updated data is piggy backed along with acknowledgment packet from the server. If GFA catches the packet and stores the updated page within its cache and if it maintains the list of updated pages then in future, it serves the requests by checking validity at GFA and replied quickly to minimize the query delay.

2.4.21 Two-Tier Cooperative Caching Technique(2TierCoCS)

VANETs face problems in data availability due to frequent disconnections and large network traffic. Cooperative caching is an appropriate solution for the problem of poor data availability, which enable nodes to locally store data items and share it with other vehicles. Parvathy *et.al* [59] proposed two-tier cooperative caching technique for VANETs (2TierCoCS), which address is the issues of cache discovery, cache admission control, and cache consistency. In the proposed technique, caching is done at two levels of architecture, vehicles and RSU so that hit ratio can be improved. The cooperative caching technique proposed by authors improves the data availability in VANETs, by reducing query latency and bandwidth usage, by avoiding the need for data from the Internet data server.

Vehicles in VANETs performs a local cache search first. If it is a miss, the vehicle sends a Data Request Packet(DRP) to the RSU, which performs a popular cache search. If it is a cache hit, Data REP packet containing data ID, data item and TTL are sent to the requesting vehicle. Otherwise, RSU searches its Neighbor Index Cache for the vehicles carrying requested data. If it finds such one, DRP packet is forwarded to the corresponding vehicle and data retrieved is sent as a DREP packet to the requesting vehicle. Otherwise, DRP packet is forwarded to the data server and retrieved data is returned to the RSU as DREP packet, which is in turn forwarded to the requesting vehicle. Following operations are performed in this technique.

Firstly server is the only entity which updates the data, in 2TierCoCS. Mainly there are two operations on the server- receive DRP packets, and Cache Update Request Packet (CURP) messages. When server receives a DRP packet from RSU, it retrieves required data from database, assigns a TTL and pack it in DREP packet along with the TTL, and sends to the RSU. The Server receives CURP packet from a RSU to prefetch data items in CURP, if it has been changed on the server. When server receives a CURP packet, then for each dataID in the CURP, it checks whether the data had changed on the server. If so it adds the updated data item and assigned TTL in server update data (SUDP) packet and send to RSU. Otherwise server adds only the updated TTL and packs it in server validation reply (SVRP) message and sends it to RSU.

RSU keeps accounting of Neighbour Index Cache; a Request count table and a Waiting List. Request count is used to find hot data. Waiting List is used to keep track of pending requests on RSU. When vehicle sends CIP packet containing cache status, RSU adds each dataID in CIP to the Neighbour Cache Index with the vehicleID and TTL. After each N seconds, RSU polls its popular cache to check whether there are items that are expected to be expired in next N seconds and prepare a CURP message which consists of data IDs of that data items and send to server. RSU receives either a SVRP or SUDP packet as the response to the CURP packet. If it is SVRP, TTL of data items in SVRP is updated in popular cache. Otherwise, if it is a SUDP message, the data item itself is updated in the popular cache. When RSU receives a DRP packet from a vehicle, first it updates the request count of corresponding dataID, and then checks whether the requestID is in the waiting list. Receiving DRP with same requestID means that the requested data item is not available in the zone and so RSU forwards the DRP to the data server to fetch data. If it is a new request ID, RSU performs a popular cache search then, Neighbour cache Index is searched in case of cache miss, or forwards the DRP to the server in case of Neighbour Cache Index miss. In case of popular cache hit, it returns a Data Reply Packet (DREP) to the requesting vehicle. If a matching entry is found on Neighbour Cache Index, DRP packet is forwarded to the corresponding vehicles carrying the data. Also, when a DRP packet is received, then update request count table is updated and sorted as per the request count. If the newly requested item is one of the maximum items, then a

DRP packet is forwarded to the server to fetch data and place in popular cache. When DREP packet from server is received, the data item is added to the popular cache, if the destination is RSU. Otherwise, the DREP packet is forwarded to the destination vehicle. A vehicle sends node departure LEAVE message when it leaves the range of the RSU. RSU delete entries corresponding to that vehicle, in its Neighbour Cache Index, after it receives the LEAVE message. There are a large number of cooperative technique for cache invalidation which are explored in [80].

2.4.22 Additional Techniques

Apart from the above techniques, there are many other techniques proposed such as- Location-dependent cache invalidation cache consistency as described in [81, 82]. Probability-based Callback (PCB), which applies a probabilistic approach to send the updated data or invalidation reports to the clients is described in [83]. In[84], a predictive caching consistency (PCC) approach uses different consistency level of data in cache. PCC uses the prediction methods on data update and query raised depending upon the history of data update and query raised pattern. Then it decides to pull, push or directly server the query with data. Better the prediction better is performnace but it cant achieve the strong consistency. A general consistency model called Probabilistic Delta Consistency was proposed recently by the authors and a flexible combination of push and pull algorithm (FCPP) was used to provide the necessary consistency to meet the user's requirements [85, 86]. Li *et al.* [87] explored probabilistic consistency further and proposed algorithms to handle the special case of frequently off line devices in a wireless environment. Also, there exists an approach proposed by Lim [88] known as user defined consistency sensitive cache invalidation. In this technique, user ranks data as per its usage and then its consistency is maintained on priority basis. Hassan Arteil *et al.* [89] has given a smart mechanism for server update for consistency in mobile environments.Special designated nodes are there called query directories (QDs), all the queries are submitted on these node only and uses these queries to locate the data (responses) from the caching nodes (CNs)ie. among themselves. This cache invalidation technique is server-based. All updates occur on server and maintains update rate. Technique handles disconnections of QD and CN nodes from network and maintains the cache consistency or drops the contents. Average response time and node bandwidth utilization are found to determin the gains of applying it in MANET environment [90, 91, 92]. Using NS2 simulations several parameters are measured, such as- average data request response time, cache update delay, hit ratio, and bandwidth utilization in [93, 94, 95]

Based upon the above discussion for all the techniques, Table 2.4 describes the categorization of Cache invalidation upon IR, BT and DT.

Table 2.4: Cache Invalidation Techniques by Nature of IR, BT and DT

Sr.No	techniques	IR Category	BT Category	DT Category
1	TS [23]	SIR	SBT	SDT
2	MTS [54]	SIR	SBT	VDT
3	BS [18]	SIR	SBT	VDT
4	AIS [55]	DVIR	DVUBT	VDT
5	GCI [19]	SIR	SBT	VDT
6	HCI [20]	DVFIR	SBT	VDT
7	IAVI [51]	DIRV	DVIBT	SDT
8	UIR [24]	DFIR	DVUBT	VDT
9	CBSFI [30, 31]	DIIR	SBT	SDT
10	SCI [43]	DVFIR	SBT	SDT
11	RIH [34, 35]	DFIR	DVUBT	VDT
12	CBSLI [36, 37]	DIIR	SBT	SDT
13	SAS [28, 29]	DIRV	SBT	VDT
14	ADD [38]	DIRV	DVIBT	VDT
15	DVD [39, 40]	SIR	SBT	UDT
16	P2PCC [57]	SIR	SBT	VDT
17	QHWC [58]	DVFIR	SBT	UDT
18	COOP [56]	DVFIR	VBT	SDT
19	CCI [41]	SIR	SBT	UDT
20	ECCI [41]	SIR	SBT	UDT
21	2TierCoCS [59]	SIR	SBT	UDT

2.5 Discussion of Dynamic IR, BT & DT Techniques

For a futuristic technique of cache invalidation with respect to various required parameters, Table 2.5 categorizes the dynamic IR techniques with various attributes. Size of IR must be optimal, i.e., it should not be much larger so that it exhibits much

Table 2.5: Comparison of Dynamic IR, BT and DT Techniques for Futuristic use

Dynamic IR-Techniques	**Futuristic Attributes**
DIRV	Small IR, Lesser Query Delay, No False Invalidation
DVIIR	Variable and Smaller IR, Having Hot Data, More Effective, Lesser Congestion, High Hit Ratio
Dynamic BT - Technique	Futuristic Attributes
DVUBT	Faster Responses, Lesser Query Delay, Approximate Size of IR
DVIBT	Faster Responses, Lesser Query Delay, Smaller IR, Update Rate Considerations, Lesser Congestion
DT - Techniques	Futuristic Attributes
VDT	Can have a range of DT
LDT	Can have larger DT
UDT	Can have Unlimited DT

delay during query processing and also not too small such that it loses its effectiveness. To increase the effectiveness, contents of IR should be included on basis of hot data/ frequently used data.

In this context, DVIR and DVIIR the technqiues which may be used in future. In these techniques IRs are broadcasted to MTs over a finite interval of time. If update rate of data items is more, then IRs must be broadcasted frequently and when update rates of data items is small then IRs must be broadcasted after a long duration by keeping a balance between size of IR, congestion, and bandwidth used. Based upon these parameters, DVUBT and DVIBT may be used in future technology. Also, MT has limited battery lives, so they need to save their battery life by switching them off voluntarily. So, a longer disconnection times must be supported by techniques. DT techniques such as VDT, LDT and UDT can be used with respect to these requirements. So, futuristic techniques would be a mixture of the above discussed techniques. A cooperative and adaptive caching for IR and BT techniques are also discussed in DCIM [96, 97, 98, 99]. Then, a cache Invalidation technique with link adaptation and downlink traffic is proposed in [100]. Table 2.5 describes the comparison of dynamic IR, BT and DT for future use with various attributes.

2.6 Comparative analysis, discussion and future scope

Table 2.6 provides the detailed comparison with analysis of various invalidation techniques by selecting various parameters. From the discussion above, it can be concluded that earlier techniques such as TS and BS are good for validity of data but these are very slow and time consuming. Also, these techniques work with larger IR which further looses the effectiveness of techniques as upper channel has less bandwidth which can be congested quite easily so these techniques may not be used in future. Then, Group based techniques like GIR, HCI and SCI are good for faster response and can provide longer disconnection times but suffer from false invalidation of data in cache of MTs. So, these may not fit in dynamic nature of wireless environments for future use. Then, UIR provides faster response and strongly consistent data with suitable longer disconnection times. Also, CBSFI, CBSLI, are techniques with longer disconnection times which is most desirable but CBSFI is a stateful technique so overhead of maintaining cache is not desirable as illustrated in Table 2.6. SAS, ADD, DVD, and CCI are the techniques which are near to the future techniques as they have Informed IRs with suitable contents. Also, sizes of IRs are adaptive in these so that congestion can be minimized at the channel. These techniques also have variable size of BT as well. So, when update rate is high they can send IRs at longer intervals keeping the effectiveness of techniques without generating congestion in the network. Most importantly, these can provide the unlimited disconnection time for MTs. So, when MT needs to switch off voluntarily it can do so to save battery and has strong consistency mechanism. To conclude, we can say that these techniques don't suffer from false invalidations which reduce the various network resources such as bandwidth and memory which enhance the performance of the network. Table 2.6 provide a detailed comparison of various invalidation techniques based upon various performance metrics.

Table 2.6: Comparison table of various existing techniques

Protocols	Year	Report Type	Report Size	Server Type	Disconnection Time	False In-validation	Query-Response	Band-width Used
TS[23]	1994	IR(OCI)	Fixed	Stateless	Small	No	Slow	More
BS [18]	1997	Bit Sequence	Fixed-Longer	Stateless	Longer	——	Common	Much
MTS [54]	1997	Bit Sequence	Fixed-Longer	Stateless	Longer	——	Common	Much
AIS [55]	1998	Bit Sequence	Fixed-Longer	Stateless	Longer	——	Common	Much
GIC [19]	1999	GIR	Fixed-Small	Stateless	Unlimited	Yes	Fast	Much
UIR [24]	2000,2003	IR+UIR	Fixed-Longer	Stateless	Long	Yes	Common	More
SCI [43]	2001,2002	IR+GIR	Longer	Stateless	Longer	Yes	Fast	Much
CBSFI [30, 31]	2001	IR(OCI)	Small	Stateful	Unlimited	No	Fast	Little
RIH [34, 35]	2005	IR+UIR	Small	Stateless	Longer	No	Fast	Much
CBSLI [36, 37]	2009	IR(OCI)	Longer	Stateless	Longer	No	Fast	Little
SAS [28, 29]	2010	IR(OCI)	Adaptive-Longer	Stateless	Longer	No	Fast	Little
ADD [38]	2011	IR(OCI)	Adaptive-Longer	Stateless	——	No	Very Fast	Little
DVD [39, 40]	2011	IR(OCI)	Longer	Stateless	Unlimited	No	Common	Much
P2PCC [57]	2014	IR(OCI)	Longer	Stateful	Unlimited	No	Common	More
QHWC [58]	2012	IR(OCI)	Longer	Stateful	Unlimited	No	Common	More
COOP [56]	2012	IR(OCI)	Longer	Stateful	Unlimited	No	Common	More
CCI [41]	2012	IR(OCI)	Longer	Stateful	Unlimited	No	Common	More
ECCI [41]	2012	IR(OCI)	Longer	Stateful	Unlimited	No	Fast	More
2TierCoCS [59]	2012	IR(OCI)	Longer	Stateful	Unlimited	No	Common	More

Detailed analysis and relative comparison of various techniques have also been provided after careful selection of various parameters. From the analysis, it can be concluded that most of the techniques may not be a candidate for future caching mechanisms in wireless environment. Only SAS, ADD, DVD, CCI are the techniques which can be considered as potential candidate as future techniques as they have adaptive size of IRs which can avoid the congestion on the link and channel. Also, they have variable size of BT. When update rate of data is more they can send IRs at shorter intervals and when update rate is high they can send IRs at longer intervals keeping the effectiveness of techniques which avoids the congestion in the network. Most importantly, these techniques also provide the longer or unlimited disconnection time for MTs. After careful analysis from the existing proposals, there is a need of a new categorization which has adaptive IR size, BT and DT intervals.

2.7 Gaps Analysis

As discussed in the above section, cache invalidation techniques are widely investigated from different aspects and many research proposals have been reported in the literature but still following are the research gaps that needs further investigation:

I) Few of caching algorithms are stateful and others are stateless. Being a stateful there is an extra overhead of maintaining the cache and client state which continuously consumes more efforts. Also, stateless techniques are limited with the disconnection time of clients (wL seconds).

II) Size of IRs is a major concern. Techniques reported in the literature have longer IRs which consumes more bandwidth for broadcasting of IRs. So, it should be minimized and if required can be increased for making effectiveness of IR and invalidating the cache completely.

III) Most of the techniques have covered the updates of data items of last wL seconds and broadcast the report after every L seconds which leaves the techniques to check for redundant updated pages resulted an increases in the response time of the queries.

IV) Almost all the techniques have a fixed interval for broadcasting. So, when update rate is very high then due to same broadcast interval, size of IR may become large and limited bandwidth for wireless environments may also become bottleneck for various applications. Hence, technique can loose its effectiveness.

V) It has been observed that the size of IRs is fixed to carry updated data items, but recent proposals have used dynamic and adaptive IR reports to utilize them optimally in limited resources environments.

VI) Only "updates in data" should not be the only criteria for being a content of IR. If data is much of use and is updated only then it should be included into IR, i.e, if hot data is updated then it must be included into IR while cold data can be ignored to includes in IR which optimizes the size of IRs and keep the effectiveness of the techniques.

VII) Proposed techniques should include mechanism to invalidate the cache of disconnected clients for a longer time.

So, this thesis focuses on formulation of a Dynamic cache invalidation technique. The factors and natures of proposed work are the adaptive and cooperation by using state of database, nature of data, need of broadcasting time intervals, size of IRs and cooperation among different components of hierarchical architecture. So, a dynamic cache invalidation is proposed in two wireless environments. Tn this thesis first environment is the cellular type of environment where we have access point, base station, server and all mobile devices, discussed in chapter 4. The adaptive cache invalidation technique (ACIT) is proposed and simulated in comparison to the existing techniques. Second environment is an Internet based vehicular ad hoc network (IVANETs), where at lowest hierarchy vehicles reside. Then at one upper level access point, and gateway foreign agents exist. Then, home agent and server resides. It is given in chapter 5. A cooperative gateway cache invalidation technique (CGCI) is proposed and simulated in comparison to existing techniques. Both techniques are presented in coming chapters with their implementation, simulations, comparisons with existing techniques, and results achieved.

The next chapter 3 presents the network simulator tool used for evaluation of proposed work.

Chapter 3

Tool Used

3.1 Introduction to Network Simulator

There are many tools available for simulations of network topologies. But the proposed algorithms are simulated using the Network Simulator 2 (NS-2) tool [101, 102, 103]. NS-2 is an open-source discrete event simulator designed for conducting networking research. NS-2 is an event driven packet level network simulator. It is a part of the VINT (Virtual Internet Testbed) project. It was a collaboration of UC Berkeley, AT&T, XEROX PARC and ETH. Network simulator first version was offered in 1995 and second came in 1996 which has a provision of scripting language called Object Oriented Tcl (OTcl). It is an open source software package for platforms. It is widely used and acceptable in industry and academia for simulations of network for communications. NS-2 is an object-oriented, discrete event driven network simulator developed at UC Berkely written in C++ and OTCL. It has implementations of network protocols such as TCP and UPD. It has various traffic source behaviors such as FTP, Telnet, Web, CBR and VBR. Along with it, router queue management mechanism is also there to offer a working of buffer such as RED,Drop Tail, and CBQ. For identifying shortest path in routing, algorithms such as Dijkstra is there. It has multicasting and MAC layer protocols for network simulations [104, 105, 106, 107, 108]. NS-2 also has full support for cache invalidation in terms of Page Pool Class, Web client and Web Server, and Cache Class and Invalidation. NS-2 includes an animation tool for viewing the results of simulation, called Network Animator (NAM). NAM is an animation tool based upon Tcl/TK for viewing of network simulation traces, packet movements in any real world packet trace topology designed. To use NAM a network topology is produced. It should have informations like nodes, links, as well as packet traces. The trace file is generated by compiling the code using NS-2. When a trace file is produced after recording all the event occurrence, packet movements etc., it is animated by NAM. It takes input as trace file, creates desirable network topology, starts a window, and then pauses when first packet is in the trace file. NAM offers an interactive user interface to control many parameters during animation. NS-2 has many uses described as follows.

1. Exiatng network protocols can be evaluated.

2. New network protocol can be evaluated before its use.

3. In real life situations large scale experimentation can be done.

4. Various IP networks in different environments can be simulated.

Figure 3.1 represents the general layout of network animator under network simulator NS-2. The topology can be created using node objects in NAM or a layout for nodes

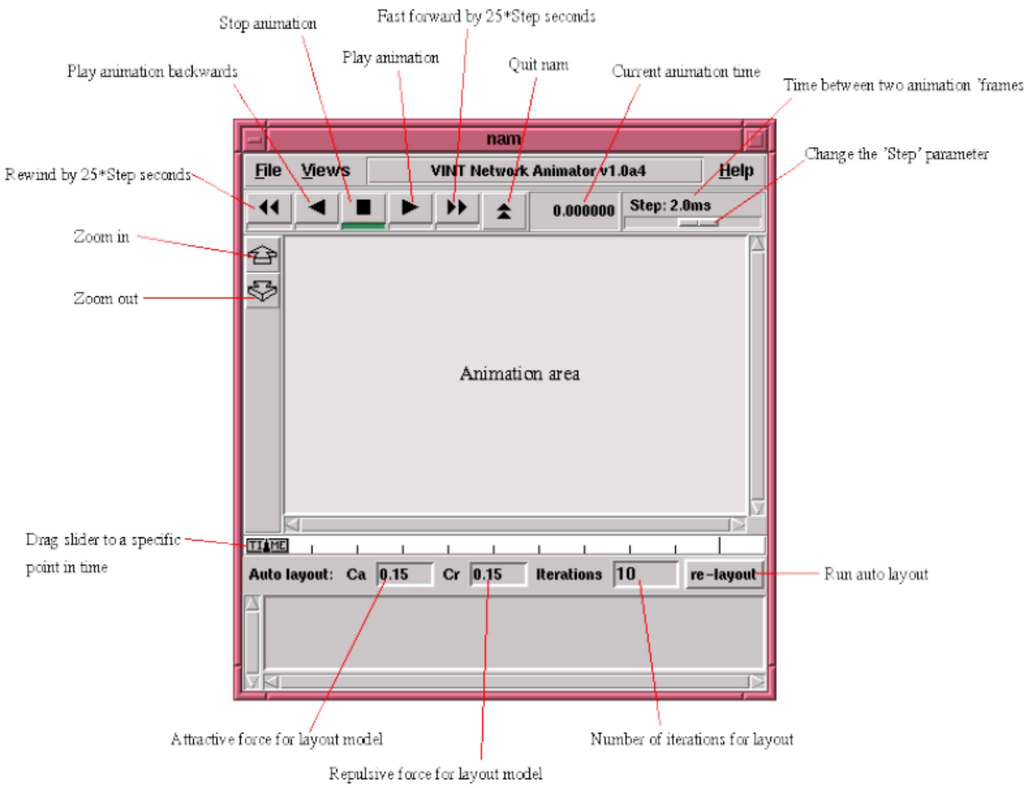

Figure 3.1: Network Animator

can be used. In NS2, NAM provides three layout methods.

First, layout of nodes is defined by mentioning the link's orientation. An orientation of link is the angle between edge and a horizontal line, in the interval of 0 to 2π degree. Orientation can be choosen by keeping one refrence node and then place other node and length of lik required.

Second, layouts can be generated automatically. Such provisions are there in NAM. For this, graph layout algorithm is used. In this graph algorithm graph with nodes connected with link is modeled. There are parameters to set during layout setting, first is 'Ca' for attractive force constant to control link form among nodes. second is 'Cr', for repulsive force between nodes and third is number of iterations i.e. number of times automatic layout procedure runs.

Third, Cartesian system layout $< x, y >$ coordinate whch is used to depict temporary wireless links in wireless topologies. Here x and y coordinate values indicate where nodes are placed in a plane coordinate system.

3.1.1 Modeling Wireless Networking in NS-2

Wireless model [102] has functionality of MobileNode as a class to have features of simulations of multi hop network,and wireless LAN networks. This C++ class is inherited from base class Node. Thus a MobileNode is Node object with functionalities for a wireless and mobile node, has mobility in designed topology. It can receive and transmit signals to and from a wireless channel. A MobileNode, has routing mechanisms, the routing protocols Destination Sequence Distance Vector (DSDV), Dynamic Source Routing (DSR), Temporally ordered Routing Algorithm (TORA), Adhoc On-demand Distance Vector (AODV), and Protocol for Unified Multicasting through Announcements (PUMA) allowing channel or neighbor nodes access in MobileNode.

The mobility means node movement, updated position information periodically, and confining into topology boundary, are implemented in C++ while having network components like classifiers, dmux , LL, Mac, and Channel are within MobileNode are implemented in OTcL. Its whole library can be found at ns/mobilenode.cc,h, ns/tcl/lib/ ns-mobilenode.tcl, for routing support ns/tcl/ mobility/dsdv.tcl, ns/tcl/ mobility/dsr.tcl,and tora implementation is at ns/tcl /mobility/tora.tcl. All given wireless example scripts can be tracked in ns/tcl/ex/wireless-test.tcl and ns/tcl/ex /wireless.tcl. Examples of topology of 3 nodes, and of 50 nodes can be executed through $ns tcl/ex/ wireless.tcl.

The routing protocols supported in ns 2 are DSDV, DSR, TORA, AODV, and PUMA. The syntax to create a mobile node is given below. set mnode [$opt(rp)-create-mobile-node $id] where $opt(rp) defines one of routing protocol among "DSDV", "AODV", "TORA", "DSR" or "PUMA" and *id* is id of mobilenode. The new API is described as follows:

```
$ns_ node-config -adhocRouting $opt(adhocRouting)
-propInstance [new $opt(prop)]
-phyType $opt(netif)
-channel [new $opt(chan)]
-topoInstance $topo
-ifqType $opt(ifq)
-ifqLen $opt(ifqlen)
-macType $opt(mac)
-wiredRouting OFF
-llType $opt(ll)
-agentTrace ON
-antType $opt(ant)
-routerTrace OFF
-macTrace OFF
```

The above syntax configures a mobilenode with primitive values of routing protocol, network stack, channel, topography, propagation model, with wired routing turned on or off. Setting the status of tracing at router,MAC, and agents.

3.1.2 Creating Node movements

The mobile node is created to have a movement in three dimensions in a topology. However the third dimension-Z axis is not used. The mobile node is assumed to move in a two dimensional plan so z axis is taken as zero. There are two ways to offer movement in a mobile node. First method, explicitly setting start and end positions of mobile node, is done by scenario file. Second method is to starting position and future position is set usin following syntax:

$node set X_ $< x1 >$

$node set Y_ $< y1 >$

$node set Z_ $< z1 >$

$ns at $time $node setdest $< x2 >< y2 >< speed >$ At $time seconds, the node starts moving from $(x1, y1)$ its initial position, towards coordinates of destination $(x2, y2)$ with predefined speed.

3.2 Web cache application support in NS-2

3.2.1 Application data use in NS-2

Application level data can be passed among applications by a mechanism from applications to transport agents. There are three major components involved first application-level data unit (ADU), second a common interface to pass data between applications, and third is mechanisms to pass data between applications and transport agents. ADU is a Packet unit here. Agent packs user data into an array of data area of ADU in nspacket. For having new ADU, inheritance is used from the base class. In derived class, more data can be added using new pack($char * b$) and a new constructor is used to initialize new data members from loaded array.

3.2.2 Transmitting user data over UDP

For all agents, a new derived class can be created inherited from UDP class, add a new function send(int nbytes, char *userdata). It facilitates sending user data from an application to agents. Process followed for sending data is: Each agent connects to agent wit a pointer, at run time pointer can be casted to an AppConnector and then give a call to $AppConnector :: process\_data()$. Further recv() function is overridden to contain user data, and a new version of send() function is written to pass user data in small packets. Then HttpApp is attached to HttpInvalAgent using $Agent :: attachApp()$.

3.3 Web cache based classes available in ns2

For web cache, ns2 provides main three classes: client (browser), server, and cache. They have same feature, i.e., HTTP protocol, so they are inherited from base class Http. An HTTP object may want to maintain multiple concurrent HTTP connections, but an Application contains only one agent_. HTTP objects need to send real world data, can be done by TcpApp instead of Agents. A class TclObject is used for all common attributes of HTTP objects and for managing HTTP connections and set of pages. In coming sections functionalities of HTTP, HTTP client, cache and server functioning is discussed.

3.3.1 HTTP connections management

Every HTTP connection has a TcpApp attached object. Http maintains a hash of TcpApp objects, which shows all active connections. It allows make and break of connections at run time. A OTcl interface used for a connection establishment, tear down and sending data through it.

3.3.2 Managing web pages

To manage set of pages OTcl interfaces are used given by HTTP. All pages are handeled by class PagePool and its family classes. Every HTTP object may have customized PagePool class as sub classes of it depending upon need of different pages. For example, need of browser for a page is different, it may need a PagePool only to use a request stream, so its PagePool can have a list of URLs only. But, for a cache, page size, modification time etc. are significant parameters so it needs different page pool. Every page contains several attributes, which are represented in OTcl as a list of the following (hnamei, hvaluei) pairs: "modtimehvali" (page modification time), "size hvali" (page size), and "age hvali".

3.3.3 Page pools

A server page has information like name, size, modification time, and lifetime. A PagePool and its subclasses are utilized by servers to generate page information.

Caches need to describe which pages are in storage, and by the clients have to generate a request stream for an URL, such taskes are completed by using PagePool.

3.4 Web client creation

Class Http/Client is used to depict the functioning of a client/web browser. It generates a series of page requests, where request intervals are randomly distributed and page *IDs* are also generated randomly. It is a OTcl class which is inherited from Http. Following are the functionalities of the web client.

Creating a client: Client is created and has a cache and connects it to a web server. A client can have a single cache but can connect to multiple servers

Configuring generation of requests: A PagePool is used by HTTP/Client to generate a request using a random variable then another random variable is used to have a consecutive gap between two request generations.

3.5 Web server creation in ns2

Class Http/Server is used to have a functionality of HTTP server. Attach a PagePool to created server and wait:

set server [new Http/Server $ns $node] ;

attach $server to $node

$server set-page-generator $pgp

An created server waits for incoming requests after simulation starts. Clients and caches have a connection method to connect to server and also server has its own conncet method to get connected to a cache or client.

In coming chapter 4 represents the detailed design of the proposed ACIT technique in wireless environment, algorithms, flowcharts, detailed design, analytical model and results obtained.

Chapter 4

Adaptive Cache Invalidation Technique- ACIT

In a wireless environment, with an exponential growth in mobile devices, user wants to have various services on-the-fly. But wireless environment unlike wired network is limited by narrow bandwidth, frequent disconnections, and slower Uplink channel. So efficient techniques to transmit information from servers to various clients are required which are generating lesser Uplink requests. Caching the frequent data items to some of the intermediate nodes is an effective and efficient way to deal with above cited problems as Uplink requests have not to be propagated to the server. But, maintaining the cache validity is one of a major challenge in improving the performance of the various services in wireless environments.

In wireless environment, MTs have high mobility, which leads to constant topological changes [4]. In hierarchical architecture, MTs are connected to an AP, which is connected to a BS, and finally, BSs are connected to server, which contains the overall data repository for all the operations to be performed in this environment. MTs are connected via unreliable low-bandwidth communication channels with the server having frequent disconnections due to power saving strategies and mobility [8, 9], i.e., this type of environment has a slow wireless link having MTs with limited battery power. But, due to highly dynamic nature of wireless environment, the frequent disconnection rate and high mobility can make them unable to receive updates from the server which results in cache invalidation. To validate the cache, MTs either have to drop their whole content or "invalidate" the cache which is called as "Cache

Invalidation".

To adapt to the dynamic environment, a CIT needs to have a strong caching mechanism based upon the cache's data consistency which can be fully maintained every time for handling the disconnection of MTs from server and to deliver the continuous services at fast response time. An efficient CIT can optimize the size of IR report due to which less bandwidth is required in wireless environment by the clients. Also the Uplink Channel is slower in such environments. So a CIT reduces the Uplink requests by validating the existing cache contents of the client. Only a few Uplink requests are to be sent. Hence, adaptive IRs and broadcasting time can make schemes more effective which reduces the Uplink requests and increases the hit ratio of caches. An efficient CIT is an approach for carrying data in IRs needed by most of the clients, so that response time for clients query decreases considerably.

This chapter aims on formulation of a Dynamic CIT. The dynamic factors considered in the proposed work are adaptive and cooperation by using state of database, nature of data, need of broadcasting time intervals, size of IRs and cooperation among different components of hierarchical architecture. Hence a dynamic CIT is proposed in this chapter. We have considered access point, base station, server and MTs for testing the performance of the designed technique. So proposed ACIT[1]technique is presented with designed algorithms and obtained results are compared with existing techniques in literature.

4.1 System Model

This section illustrates the network and mobility models used in the proposed technique.

4.1.1 Network and Mobility Model

A wireless environment can be categorized into three groups [109, 110] as follows:

- Pure cellular: In this architecture, gateways/Access points are deployed at every

[1]The contents of this chapter have been peer reviewed and published in Rajeev Tiwari, Neeraj Kumar, " An Adaptive Cache Invalidation Technique for Wireless Environments", Telecommunication Systems,62(1): 149-165 (2016).**(SCI-Thomson Reuters Rated Journal with Impact Factor 0.705)**

intersection. It supports infrastructure- based services.

- Pure ad hoc: It contains only MTs communication without any centralized control.
- Hybrid: It is combination of pure ad hoc and pure cellular.

In this technique, we have considered a hybrid architecture in which APs and BS are deployed in such a manner so as to connect MTs to clients in different network domains. AP and BS are statically connected to Data Server as shown in Figure 4.1. We have also assumed that entire data resides on server and data updates take place on the server only. So, BS and AP facilitate MTs to communicate with clients in different network domains to access various services to exchange data. So, AP and BS act as gateway and router for MTs. The entire data locates at server and on updation of data, server periodically broadcasts an IR report down in network to BS, which in turn, sends IR to lower level device such as- AP and MTs. Finally, MTs fetch IR and validate their cache.

We have considered static components such as- Server S, BSs, and APs. Although MTs have mobility but, we have not considered any hand offs mechanism in the proposed scheme as it is provided by standard Mobile IPV6 (MIPV6) protocol, i.e., we have restricted MTs mobility in the range of their corresponding APs because, mobility is not a significant parameter for cache invalidation of MTs. We have taken assumption that MTs are continuously connected to their APs.

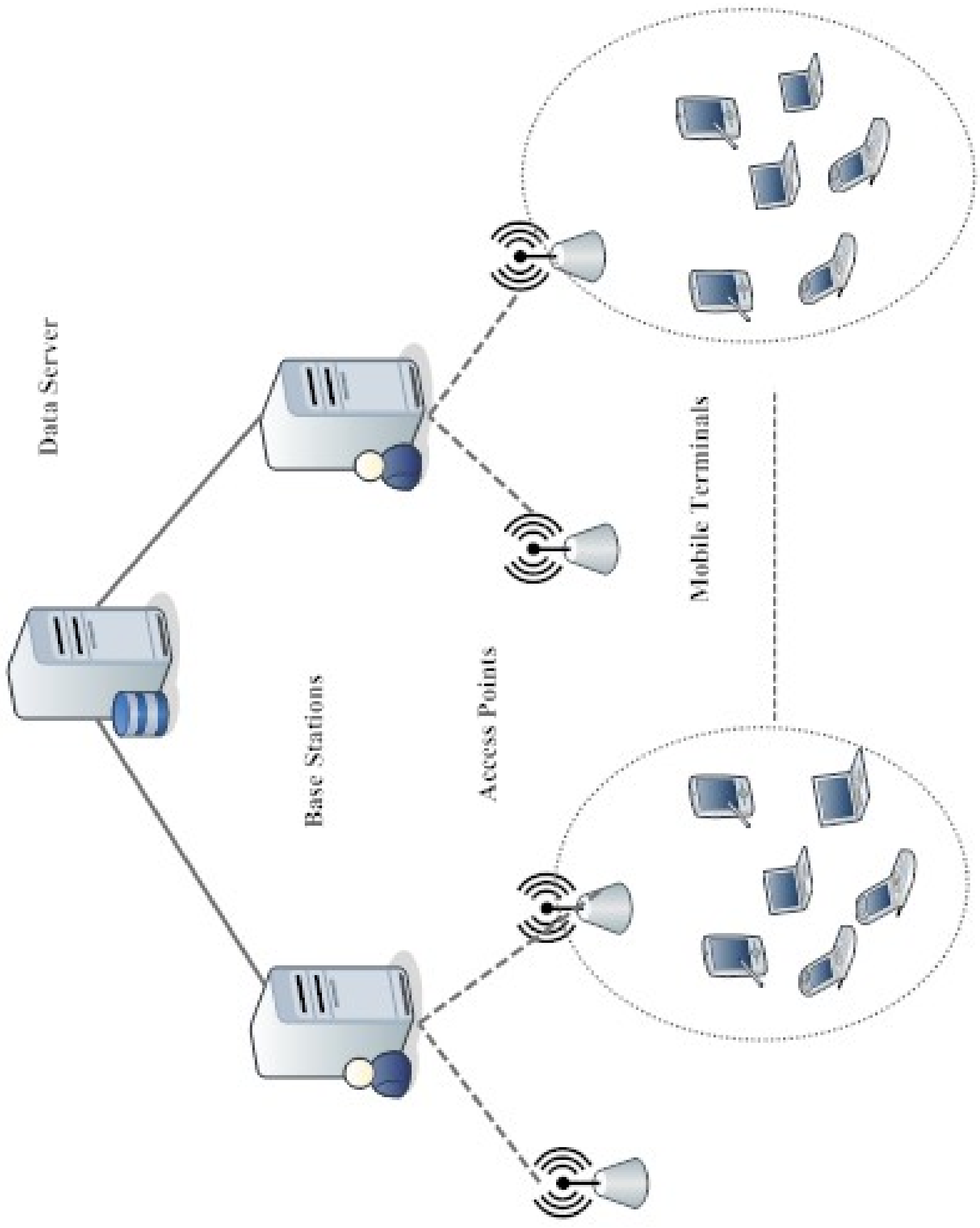

Figure 4.1: Network Model Used

4.1.2 Problem Formulation

There are various challenges for validating the cache consistency of MTs in a wireless environment. This section explores pitfalls in the existing approaches and then using the existing pitfalls, problem of cache invalidation is formulated. In a MT initiated cache invalidation approach, MT sends Uplink requests to server to invalidate the cache. So, as the number of MTs increases, it generates more Uplink requests which results in generation of congestion to the Uplink channel and congestion of AP. For stateful server approach [16], exact state of clients cache is also maintained on the server. So, on any update of data page on the server, it sends an IR message to client so that it replies to the queries instantly. Hence, such a technique has better query response time but, it has an extra overhead of maintaining cache state on server. Hence a lot of power is consumed in this process. In stateless server schemes [36], there is no extra overhead for maintaining exact state of client's cache as IR is broadcasted to MTs. When it is broadcasted asynchronously, then on every update of data, server broadcasts an IR to MTs which consumes most of bandwidth and keeps MTs active every time so, MTs consume more energy. Contrary to the above, when IR is broadcasted synchronously [23], then after fixed time interval, all updates are sent to MTs through IR so, for small duration between these events, MTs can sleep and save power consumption. But, IR sizes became large if update rate of data is more so, larger IR consumes more downlink bandwidth. Most of the group-based Invalidation schemes suffer from false invalidation of cache contents as they have large size of their GIRs which consume more bandwidth. In counter based stateless schemes [31], IR was effective in terms of invalidating cache contents, as it uses hot updated data only in IR but, their broadcast time are fixed, i.e., they need to be adaptive depending upon the update rate of data.

So, for CIT, cache size is kept fixed with every MT, IR need to be of appropriate size, with less consumption of bandwidth so that query delay can be minimized. Hence, the objective function for the above problem is formulated as given in Eq. (4.1)-

$$Minimize\{AQRT\}$$

Subject to the following constraints

$$\begin{cases} P_{hit} + P_{miss} = 1 \\ P_{hit} \geq P_{miss} \geq 0 \\ 0 < C_{th} \leq count_{dx} \\ 1 \leq CHR \leq 2^{\frac{w}{2}-1} \end{cases} \tag{4.1}$$

For minimizing AQRT, we proposed an objective function given above subject to constraints as defined above where, $count_{dx}$ is greater than and equal to threshold counter value of count and Cache History register (CHR) is less than $2^{\frac{w}{2}-1}$ by considering the higher hit ratio with respect to miss ratio.

4.2 Details of ACIT

Based upon the above issues and challenges, this section illustrates the proposed solution to the problems discussed above. In the architecture discussed in Section 4.1.1, every MT has cache memory with size of 200 data items. A server has all the data D so that all updates of data items are done only on the server. Server is a stateless and synchronous in nature.

- So, a hybrid cooperative scheme is proposed with adaptive IRs and BT depending upon the update rate and hotness of the data items on the server. Hot data is identified using similar kind of technique as proposed in [32], which in turn, provides faster query response time with less bandwidth consumption.

- The proposed scheme also has disconnection time of 0 to wL seconds., i.e., upto w BT intervals.

- Proposed scheme improves bandwidth utilization by reducing the report size and the number of Uplink requests sent to the server.

- The report size is made dynamic and adaptive as per the needs of MTs by using the hotness of the data items on the server.

- Moreover, the proposed scheme overcomes with the problems of more bandwidth usage and higher query response time of existing Update Invalidation Report (UIR) [24] and Selective Adaptive Sorted (SAS) [28] techniques.

The detailed description of the proposed scheme is illustrated in the following subsections.

4.2.1 Design and Structure

For managing the hotness of data on server, we have used a counter register with every data, $count_{dx}$ for counting the number of accesses of d_x in t_{bt}. So, we identify the hotness of data using similar kind of technique as used in [32]. A counter is maintained to check that how many requests for a data item have been received at the server in last unit interval of time. Then, control history register of w bits is maintained to record historical hotness of data. Then, IRs of only that hot updated data is created along with data pages d_x for t_{bt} which is sorted as per time stamps. Similarly, report of last $(w-1)$ BT intervals is maintained without data pages. So, IR is divided in to two parts as- History Invalidation Report (HIR) of last $(w-1)$ BT, and Current IR (CIR) of t_{bt} seconds. This IR is sent to BSs, which divides this IR in two parts as HIR, and CIR. Finally, HIR is kept with BS and CIR is sent down to MTs using AP.

4.2.2 Hot Data Maintenance

On the server side, data hotness is maintained which is accessed frequently by many MTs. So, we have identified the hotness of data. Hotness of the data is decided using two factors- how many requests for a data item have arrived at the server in last BT interval and recent history of hotness of data. We define the following terms for hotness of data.

- ***Threshold Counter***: $count_{dx}$ value is compared with a counter threshold value c_{th}. When any data's counter value is larger or equals to c_{th} then, it is included in current hot data list, i.e., if $(count_{dx} >= C_{th})$ then data item is considered as hot.

- ***Historical Recency***: Data item remains hot during recent past $(w/2)$ intervals. As the hotness of data remains hot for some period, so we have used Control History Register (CHR) of length w bits to keep track of hotness of d_x of last $(w/2)$ BT intervals. So, if data was hot recently, it is kept in hot category. In CHR, every bit shows that it was hot (1) during that broadcast interval or not (0). So, if it was hot during recent half of time interval, we gave that data a second chance to become hot, and push a 0 bit in its history control register and use a mask of 0000011111 to check about recent hotness in CHR. If after masking, we get a value between 1 to 31 then, we keep data in hot list for t_{bt}. Hence, a less value indicates more recently data item listed in hot data list. It helps to keep the recency (locality of reference) for the hot data item. We illustrate this technique using following two examples.

Example 1: Consider a data counter is not upto threshold value with $w = 10$ units, L is 20 seconds and CHR is 1111010000. Check if data will be included in hot data list or not?
Sol: Mask of 0000011111 is used to check about the recent hotness of the data item. CHR is masked by mask register value as- 1111010000&0000011111 = 0000010000, it is numeric 16, and is less than 31 so, we keep this data in hot list and push another 0 in CHR by shifting it towards left. Now, for the next BT if d_x is not frequently accessed then, it will be out of hot data list and as a result after masking, the value is not between 1 to 31, i.e., there is no recent hotness of the data item.

Example 2: Consider a data counter is not upto threshold value with $w = 10$ units, L is 20 seconds and CHR is 1111100001. Check if data will be included in hot data list or not?
Sol: Mask of 0000011111 is used to check about the recent hotness of the data item. CHR is masked by mask register value as. 1111100001 & 0000011111 = 0000000001. It is numeric 1 which is less than 31 so, it is kept in hot data list so another 1 is pushed in CHR by shifting it towards left. As value is very close to 1 so, this is very recent hot data. Hence, it has more chances to be included into hot data list.

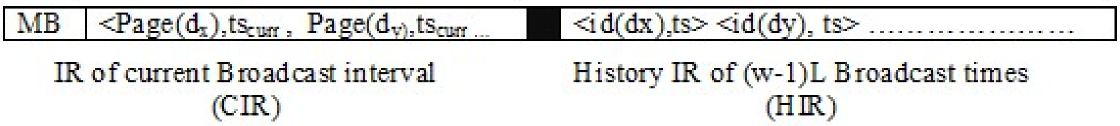

Figure 4.2: Invalidation Report (CIR+HIR)

4.2.3 Adaptive IR Generation

IR records data id_s of hot updated data only for a time period of current BT, i.e., t_{bt}. So, for time $(w-1)t_{bt}$, the data id_s of such hot data with their time stamps are also included in IR. It keeps historical updates of data on the server and also helps disconnected clients to validate their cache upto $w*(t_{bt})$ times. This IR is broadcasted to BS. Because IR has only hot data of interval of t_{bt} seconds so, size of IR is optimal and broadcasting it to BS will not consume more bandwidth.

IR contains information in two blocks- first block stores information of current BT interval t_{bt}. This block contains information such as- data id_s of hot updated data, original $page(d_x)$ and time stamp. On the Server, update rate U_{rate} of data pages is tracked which is used to set the mode of operation of scheme using value of MB as 1 or 0. In second block, data ids of last $(w-1)$ BT, hot data ids with their time stamps is stored. These ids are sorted using their time stamps values, so that while checking of most recent changes at client side, only unchecked and recent missed IR information can be checked firstly. When time stamp of MT exceeds time stamp of data items of IR, it will not check that further so, it is sorted. This sorting of data id_s in IR reduces the validation time to larger extent.

The data of block one is broadcasted to MTs because, it is the hot updated data from the server and is required by MTs in near future. Then, Uplink and downlink requests are generated for it. So, it is beneficial for MT, if server, sends updated hot data which is invalid in local cache of a MT. Then, MT pre fetches that data for future uses, which makes the down link bandwidth effective and reduces future Uplink and downlink requests for that hot data.

Also, IR is divided in two parts on BS as- CIR $(wL - t_{bt} < ts <= wL)$, and HIR $(0 < ts < wL - t_{bt})$. Then, BS sends first block as IR to MTs and real hot updated data page(d_x) to MTs. Its IR is shown in Figure 4.2.

4.2.4 Adaptive BT - Fast or Normal Mode Operations

As server keeps track of update rate of current BT interval as U_{rate}, so, a threshold value for update rate, U_{th} is kept. After this update rate, BT is regulated to smaller intervals and MB is set to Fast Mode. If update rate is higher then a smaller BT is kept. Otherwise, earlier BT works fine, i.e., L works by setting mode bit as 0 (NORMAL) to keep smaller query delay. Due to this adaptive BT, the tuning size of IR does not increase considerably and it remains optimal.

Following advantages are achieved using this strategy:

- Tuning of BT on update rate gives smaller size of IR.
- Congestion on the network reduces during high update rates.
- Query reply is faster as BT is reduced.

4.2.5 Disconnection MT Support

As MTs are limited with the battery life so, an energy efficient scheme is required to support the disconnected operation. For energy efficiency, MT can disconnect actively (voluntarily) or passively (due to some failures such as link failure). Following scenarios are considered for disconnected MTs.

For active disconnections up to t_{bt}: An actively disconnected MT reconnects with AP waits for reply with respect to the query. So, it listens to current IR from AP and current updated hot data list and validates some portion of its cache and prefetches that data.

For Passively Disconnected for Longer durations up to wL seconds. : As data history of last w broadcast times are recorded and is broadcasted at BS so, if any MT has missed some IR in sequence that demands for history IR from AP which in turn, asks the same from BS and validates the cache of reconnected active MT. When a MT is disconnected, it records its disconnection time and its reconnection time on AP. So, after reconnection when MT makes first query for data then, it listens to current IR from AP and current updated hot data list to validate the portion of its cache. After that, it pre fetches that hot update data and asks for validation of its cache by sending its start and end time of disconnection from BS along with the requested data if not validated yet. Then, server sends all the recorded changes of $(wL-t_{bt})$ interval to MT using which MT validates its cache and serves for subsequent requests.

4.3 Modules and List of Data Structures Used

We have kept the client with default configurations having cache, manager, and query generator modules. We have used Broadcast Manager, Query, Uplink Queue with their basic functionalities. For example, broadcast manager broadcasts the IR, query module uses MT query to be answered from server, and Uplink queue contains all the upward traffic from lower components towards server. When cache is full, Least Recently Used cache replacement algorithm is used instead of an advanced replacement techniques [111]. We have deployed other modules also such as- *update manager module, history register manager*, and *hot data selector* for IR to enhance the functionality of server. Following are the functionalities of these modules included in the simulation model.

- *Update Manager:* It keeps track of update rate of data pages and then matches this rate with a threshold value.
- *History Register Manager:* To keep track of history register for every data item and manages it for every t_{bt}.
- *Hot data Selector:* It selects only hot updated data for inclusion in IR.

Depending upon MB value, t_{bt} is set for next broadcast time. Then, IR is generated and broadcasted to MT. Depending upon disconnection time of MT, a hit or a miss of data item occurs. A miss occurs if data item is not found in cache or data item is marked as invalid in cache. Then, data item request is sent to an Uplink and brought from server to cache.

4.4 Algorithms and Flowcharts

To show the flow of sequences of various steps in the proposed scheme, algorithm for server side, Base station side, and MT side are illustrated in this section. The flowcharts of server is presented in Figure 4.3.To understand cache validation process another flowchart is created that depicts validation process in Figure 4.4 and finally query reply flowchart is given in Figure 4.5.

Algorithm 1 Server side sequence of operations

1: **Input**$t_{bt} = \{L, 1\}$, For Normal Mode MB=0 and Fast Mode MB=1, $L > 1$ //Working Mode Bit w: Broadcast Time Window & $w = n * t_{bt}$. Server Keeps Track of Updation of Data and Update Rate U_{rate} of current t_{bt} & counter of data items.
2: **Output:** IR and Requested Data Page.
3: Setting Mode of Operation
4: **if** $U_{rate} >= U_{th}$ **then**
5:   Set MB=1 & $t_{bt} = 1$ //Fast Mode- Adaptive BT
6: **else**
7:   $MB = 0$ & $t_{bt} = L$ //Slow / Normal Mode
8: **end if**
9: On Updations, Checking for Hot Data.
10: **if** (d_x is Updated) **then**
11:   **if** $count(d_x) >= C_{th}$ **then**
12:     Include in CIR as (d_x, t_x^r)
13:     Maintain CHR by pushing 1
14:   **else**//Checks its CHR Value
15:     **if** $((CHR <= w/2)AND(CHR > 0))$ **then**
16:       Include it in CIR as (d_x, t_x^r)
17:       Maintain CHR by pushing 0.
18:     **else**
19:       Push 0 in CHR.
20:     **end if**
21:   **end if**
22: **end if**
23: For IR Generation
24: Generate IR after t_{bt} and send to BS with data pages.
25: For Data Request
26: Receive Data Page Request
27: Send Data Page to BS.

Algorithm 1 describes the complete sequence of operations of Server side. As input in line number 1, server requires data such as MB bit, BT, and BT window w. Then, server generates IR and response of any requested data. From lines no 3 to 8, the working and setting up of MB is shown. Depending upon if $(U_{rate} >= U_{th})$, the mode bit is set (Fast Mode) or reset (Slow/Normal Mode). Then, lines 9 to 22 describe how the hotness of data is checked along with the counter and cache history register value. Lines 10 and 13 are used for checking of current hotness based on threshold value of counter. Lines 15 to 17 are used for checking the recency of hotness up to $(w/2)$ last intervals. In lines 23 and 24, IR is generated and in 25 and 27, data request is full filled and sent to BS.

Algorithm 2 Base station side sequences of operations

1: **Input:** IR from Server, DRP from AP
2: **Output:** Requested Data Page, request to server
3: Receive IR from Server.
4: Set its t_{bt} as MB bit.
5: Separate its CIR & HIR
6: Broadcast CIR with d_x and pages to APs.
7: Receives HRP request from AP.
8: Sends HIR to AP.
9: Receives DRP from AP as $\{< d_x, CoA_m >, CoA_{AP}\}$
10: **if** Page Found **then**
11:     Replies with Data Page.
12:     Forward Request to Server as $\{<< d_x, CoA_m >, CoA_{AP} >, CoA_{BS}\}$
13: **end if**
14: Receives Data Page from Server as $< page, CoA_m >$
15: Sends it to AP.

Algorithm 2 describes the working of BS component in the proposed technique. From lines 3 to 6, BS sets t_{bt} as MB and judiciously divides IR into CIR and HIR. Then, BS sends CIR with data and keeps HIR for disconnected MTs, which misses some recent CIRs. Lines 7 and 8 show history request packet demands and response by BS to AP. Then, in lines 9 to 11, request for query data handling from AP is depicted and is sent to upper level to server. In lines 12 and 13, response to requested data page is sent using AP.

Algorithm 3 Access Point side sequences of operations

1: **Input:** CIR, Request to server
2: **Output:** HRP,Data Page
3: Receives CIR from BS
4: Broadcast IR to MTs
5: Receives HRP request from MT
6: Forwards HRP to BS
7: Receives HIR from BS
8: Sends/Unicasts HIR to MT
9: Receives DRP from MT
10: Sends Request to BS
11: Receives Data Pages from BS
12: Sends it to MT.

Algorithm 3 describes the steps taken by AP. In lines 3 and 4, CIR is received on AP and is broadcasted by AP to MT. In lines 5 and 6, request for HRP is handled.

In lines 7 and 8, a HIR is received from BS and it is sent to MT. Then, in lines 9 and 10, received request for data from MT is forwarded to BS. Lastly in lines 11 and 12, received data is sent to MT.

Algorithm 4 MT side sequence of operations

1: **Input:** Query, CIR
2: **Output:** DRP, Query Reply.
3: MT connects to AP at time t_i
4: **if** $(wL - L <= t_i <= wL)$ **then**//Active Disconnection
5:    Wait for next CIR & receive pages
6: **else**
7:    **if** $(1 <= t_i < wL - L)$ **then**//Missed some IR
8:       Call HIR from AP
9:    **else**//Passive Disconnections
10:       Flush Cache Contents and Receive IR with Data.
11:    **end if**
12: **end if**
13: Query generated for d_x
14: **if** $(d_x \in Cache_{MT})$ **then** //Hit Occurs
15:    Reply with $Page(d_x)$
16: **else**
17: Miss Occurs
18:    Generate DRP (d_x, CoA_{MT}) and Send to AP.
19: **end if**
20: Receives Data Packet from AP and Replies Query
21: Receives HIR from AP
22: Validate Its Cache.

Algorithm 4 illustrates the detailed sequence of operations at MT side. MT queries for data, receives CIR and looks for data request packet or query reply. In lines 3 to 12, there is a check on reconnection of MT at t_i to confirm that whether it was active disconnection or passive disconnection in case MT has missed some recent IRs broadcasted by server. In lines 13 to 20, steps for query handling are shown. If hit occurs with a reply and miss then generated DRP is forwarded to AP. Lines 21 to 22 show that on arrival of requested data from the server, a send request is replied with respect to the generated query from the clients. Moreover, for passive disconnection, a HIR is received from BS for validation of the cache.

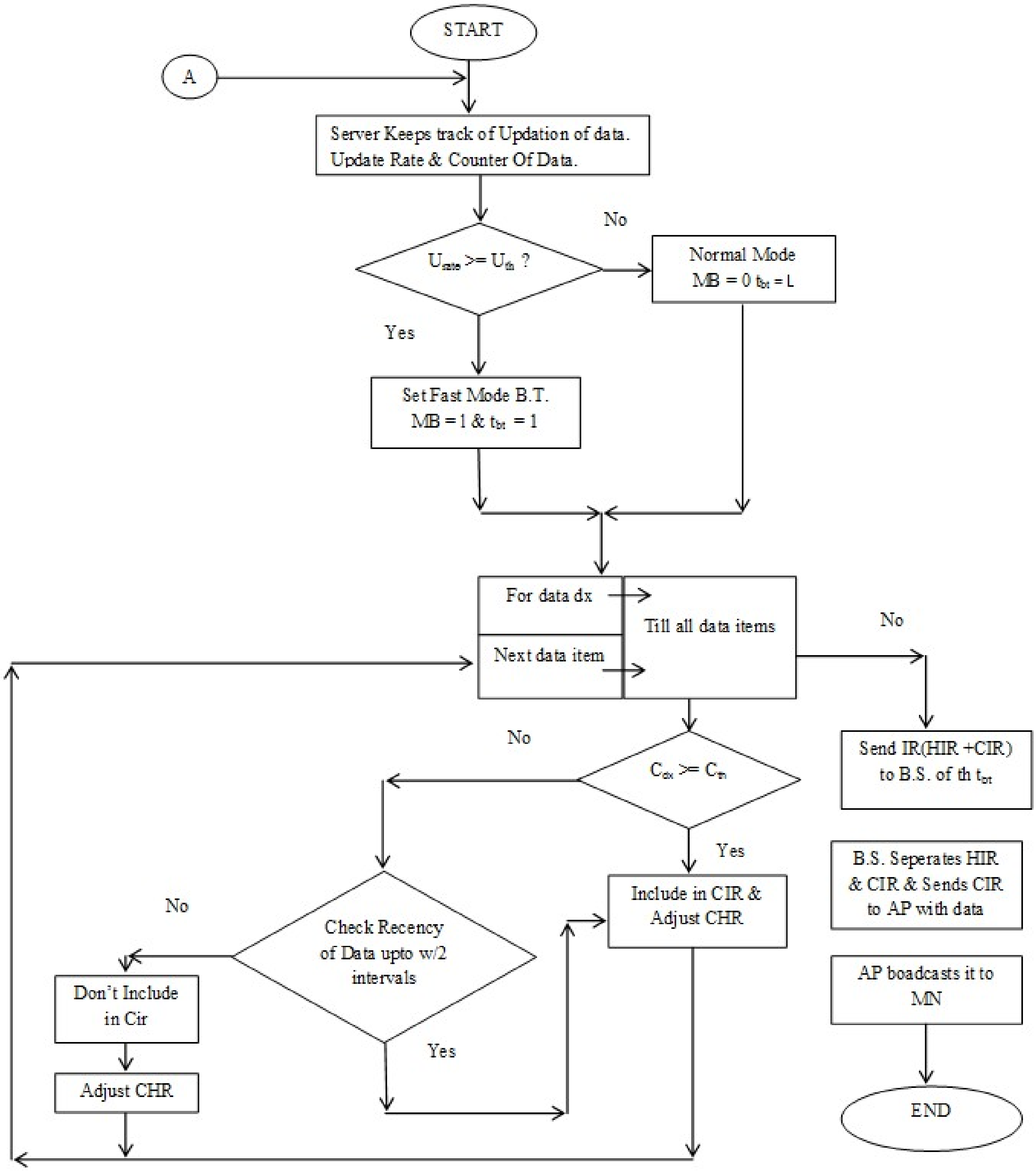

Figure 4.3: Server flowchart

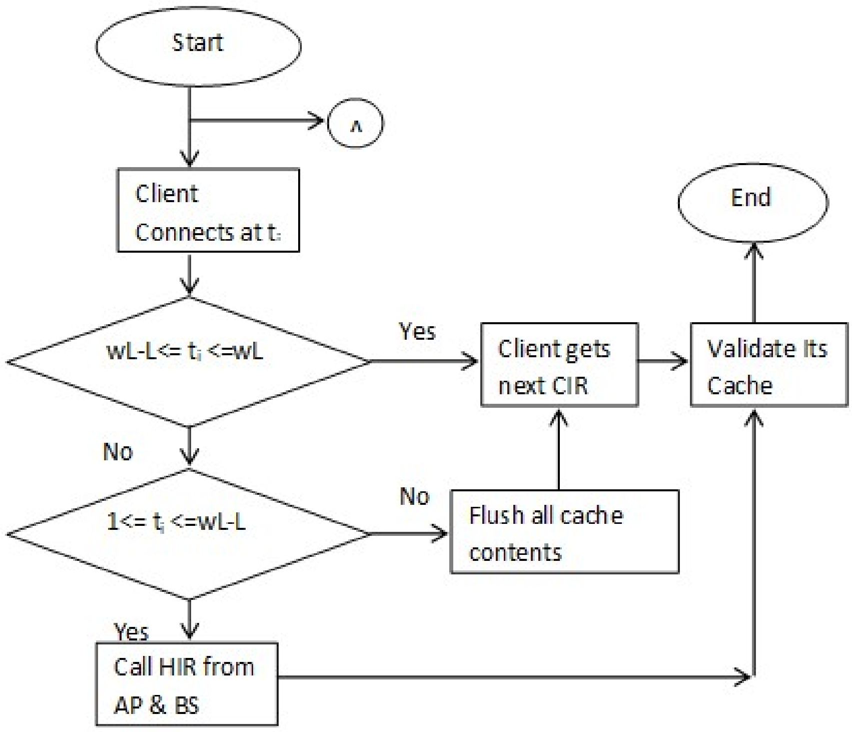

Figure 4.4: Cache Validation flowchart

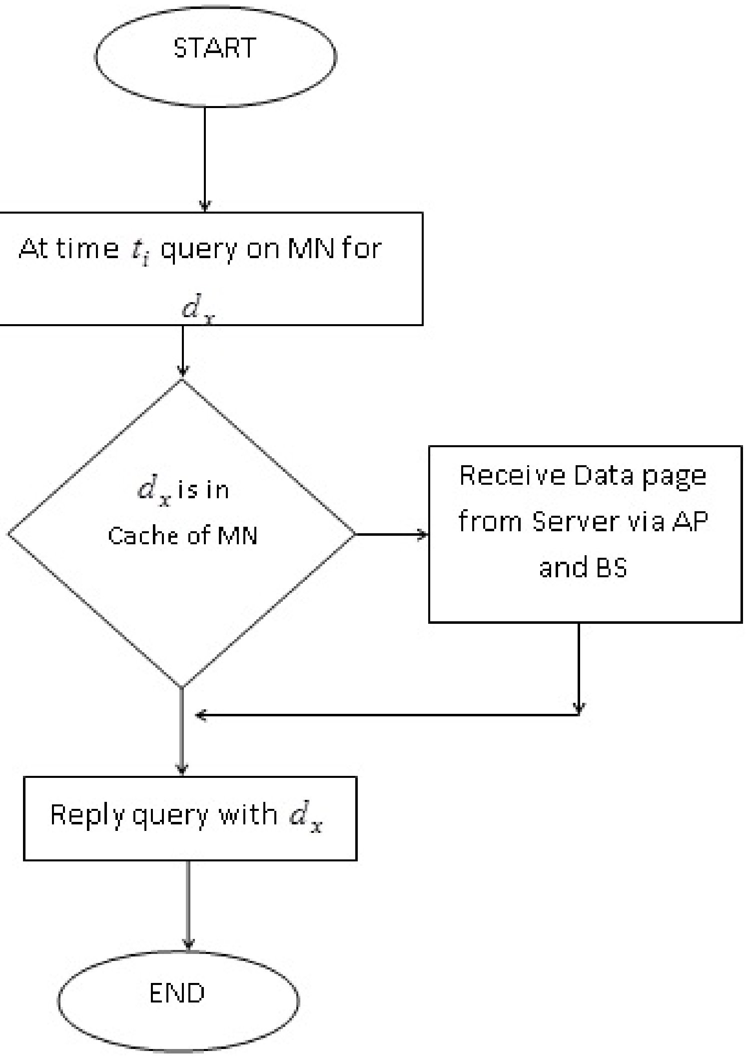

Figure 4.5: Query reply flowchart

4.5 Analytical Model

In this Section, we have analyzed the proposed ACIT with respect to various parameters such as- AQRT which consists of two types of time delays as time spent during hit occurrences (time spent 1) and time spent during miss events (time spent 2). The arrival of queries on MT and updates of data on server are considered as Poisson's processes while, their inter-arrival time are taken as Exponentially distributed. We have analyzed the following parameters using below mentioned analytical model.

4.5.1 Average Query Response Time

Whenever a query is generated on MT, it accesses IR for cache validation. The data may be in cache and may be validated by IR received via BS, AP from server. So, let the probability of data item to be cached is P_{cache} and probability of data to be updated during t_{bt} be P_{up}. Then, consider P_{hit} as the probability of cached data as valid and P_{miss} as probability of cached data as invalid. So, $P_{hit} = (1 - P_{miss})$. Hence, we define AQRT using Eq. (4.2) as follows.

$$\begin{aligned} AQRT &= P_{hit} \times (TimeSpent1) + P_{miss} \\ &\quad \times (TimeSpent2) \end{aligned} \tag{4.2}$$

We also assume that total data on the server is D, out of which H% is hot and C% is cold, i.e., $C = (100 - H)$. Let us assume that cache size is C_{size} % of D. For any data selected from hot or cold data set, we have probabilities as P_{hot} and P_{cold} where, $P_{hot} = (1 - P_{cold})$ holds. Then, the probability of data item cached in MT is represented as follows:

$$\begin{aligned} P_{cache} &= P_{hot} \times ((\frac{C_{size}}{H}) \times P_{hot}) + (1 - P_{hot}) \\ &\quad \times ((\frac{C_{size}}{C}) \times (1 - P_{hot})) \end{aligned} \tag{4.3}$$

When the cached data is queried after it is updated then, it is a invalid data item. So, probability of data item being updated during t_{bt} with respect to query arrival time where, $t_u < t_q$ is given as follows:

$$P_{up} = \int_{t_q=0}^{\infty} \int_{t_u=0}^{t_q} \mathrm{f(t_q)} \times \mathrm{g(t_u)}\ \mathrm{d}(t_u)\,\mathrm{d}(t_q) \tag{4.4}$$

$$P_{up} = \frac{U_{rate}}{(Q_{rate} + U_{rate})} \tag{4.5}$$

Hence P_{hit} is represented in terms of P_{cache} and P_{up} as given below.

$$P_{hit} = P_{cache} \times (1 - P_{up}) \tag{4.6}$$

We can write P_{hit} using Eq.(4.7) as follows.

$$\begin{aligned} P_{hit} &= (P_{hot} \times ((\frac{C_{size}}{H}) \times P_{hot}) + (1 - P_{hot}) \\ &\quad \times ((\frac{C_{size}}{C}) \times (1 - P_{hot}))) \times (\frac{Q_{rate}}{(Q_{rate} + U_{rate})}) \end{aligned} \tag{4.7}$$

Now, as per Eqs. (4.3,4.5, and 4.7), P_{hit}, and P_{miss} have been computed as above. In Eq. 4.2, we have two time components as- the time delay during query reply when hit has occurred on MT cache. So, it includes only query processing time of MT, i.e., QP_{mt}. Hence, following holds.

$$TimeSpent1 = QP_{mt} \tag{4.8}$$

The second component is the time delay occurred during query reply when a miss for data has occurred. So, request for data is made to server via AP and BS. Hence, requested data page is brought back to MT via same components and the time spent is represented as follows.

TimeSpent2 = (Request Sent on Server Via AP and BS) + (Response with requested Page via AP and BS).

$TimeSpent2 = (QP_{mt} + QP_{ap} + QP_{bs} + QP_s) + n \times (QP_{mt} + QP_{ap} + QP_{bs} + QP_s)$

Hence, we can write

$$TimeSpent2 = (n + 1) \times (QP_{mt} + QP_{ap} + QP_{bs} + QP_s) \tag{4.9}$$

Using Eqs.(4.2, 4.6, 4.8 and 4.9), we can write equation for AQRT as given below.

$$
\begin{aligned}
AQRT \;=\; & (P_{cache} \times (1 - P_{up}) \times QP_{mt}) + \; ((1 - P_{cache}) \\
& \times (1 - P_{up})) \times ((n+1) \times (QP_{mt} + QP_{ap} \\
& + QP_{bs} + QP_s))
\end{aligned} \tag{4.10}
$$

Figures 4.3,4.4 and 4.5 show flow of sequences of activities at the server side, for cache validation, and query reply.

4.6 Performance Evaluation

4.6.1 Simulation Settings

To evaluate the performance of the proposed scheme, we have used the discrete event simulator NS2 [101, 102]. We have used a scenario consists of wired and wireless connections with hierarchical addressing. We have taken 10 MTs, 2 APs, 2 BS and 1 server with all data items in network topology shown below for simulations. In our simulation model as shown in Figure 4.6, we have neither considered mobility nor location management of MTs. These are assumed to be taken care by MIPV6.

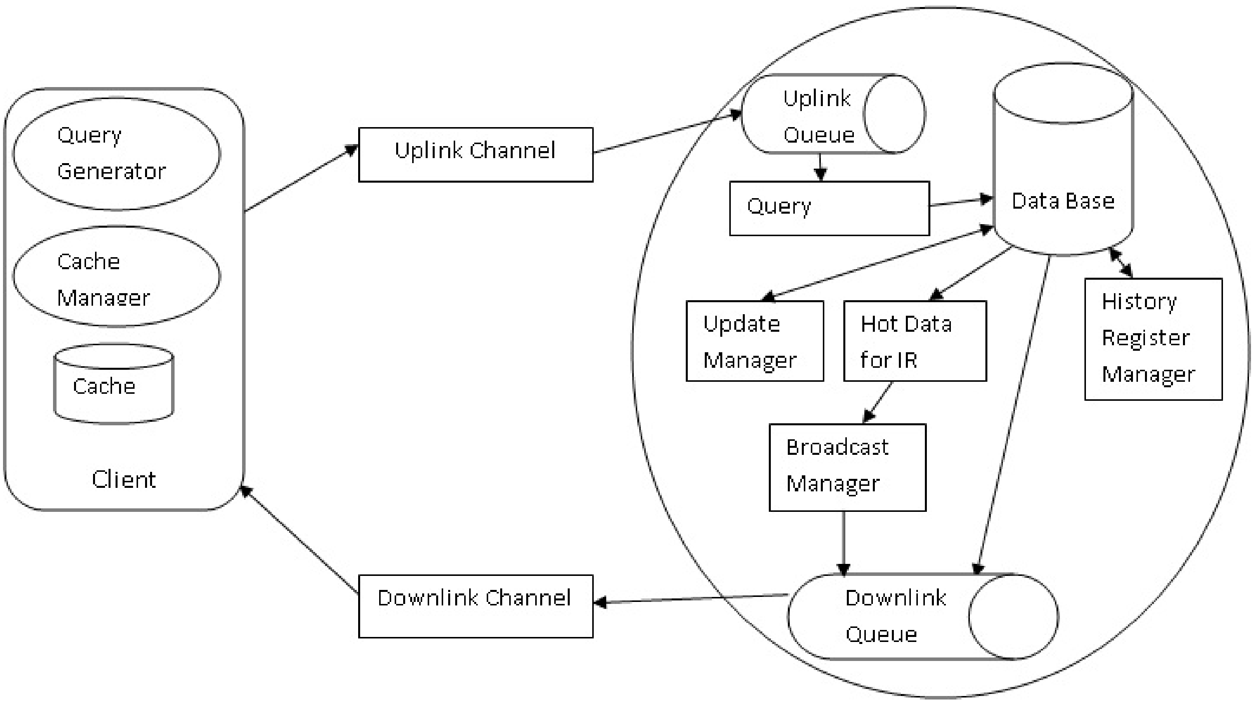

Figure 4.6: Simulation Model Used

We have kept the client with default configuration having cache manager. Query generator module is modified to generate thousands of queries. Broadcast Manager, Uplink Queue are used with their basic functionalities such that broadcasts the IR. Query module uses MT query to be answered from server and Uplink queue contains all upward traffic from lower components towards server. When a cache is full, Least Recently Used (LRU) cache replacement algorithm is used instead of any advanced replacement techniques [111].: We have used LRU replacement algorithm, because we have considered hotness of data. Most of the time, hot data reside in cache of MTs and is computed depending upon the counter and recently used data. So, to keep recently used data, for which LRU algorithm is used. We have also deployed additional modules such as- *update manager module, history register manager* and *hot data selector* for IR to enhance the functionality of server. Following are the functionalities of different components included in the simulation model.

- *Update Manager:* It keeps track of update rate of data pages and then matches this rate with a threshold value.
- *History Register Manager:* To keep track of history register for every data item and manages it for every t_{bt}.
- *Hot data Selector:* It selects only hot updated data to be included in the IR.

Depending upon MB value, t_{bt} is set for next broadcast time. Then, IR is generated and broadcasted to MT. Depending upon disconnection time of MT, a hit or a miss of data item occurs. A miss occurs if data item is not found in cache or it is marked as invalid in cache. Then, data item request is sent on an Uplink and brought from server to cache. The simulation parameters used in the proposed scheme are listed in Table 4.1.

Readings of simulations are recorded for each scenario by running simulator five times and then taken the average values to plot them in charts. Mostly values during each simulation were approximately close to each other for similar set of domain values. It shows the stability of simulator for depicted scenario. The detailed description of various parameters used in the proposed scheme are as follows. For query response time, we have considered time of query and query response time. IR report size is noted from parameter IR_{size}. From front and rear values of Uplink queue, the output

Table 4.1: Parameters used and their values in ACIT

Parameter	Default Value
Number of Mobile Terminals, n	10
Database Size, D	65536 data items
Cache Size, C_{size}	200
Broadcast window size, w	10
Broadcast time,L	20seconds.
UIR times,m	4
Data id, d_x	32 bits
Data Size	2048 bits
Time Stamp Size	64 bits
Distribution of query arrival times	exponential
Distribution of update arrival times	exponential
%age of hot data items, H	10 % of D
%age of cold data items, C	90 % of D(Remainder)
Topography	670*670m
Simulation time	1000s
Routing protocol	DSDV
No of BS	1
No of AP	2
No of Server	1

parameter, Uplink request is found. For BT time, depending upon mode bit of IR (MB=1/0) BT is taken as 5 or 20 seconds.

4.7 Results and Discussion

To check the effectiveness of the proposed scheme, it is compared with the other state-of-the-art schemes such as- SAS [28], and UIR [24]. The performance of the proposed technique is evaluated and compared using several metrics such as- AQRT, IR size, and Uplink requests. Each metric is measured by varying parameters such as object update rate, and query arrival rate. The results obtained are illustrated as follows.

4.7.1 Effect of Object Update Rate

When object update rate increases on server, then cache miss ratio also increases on MT side because most of the requested data items stored in the MTs cache are invalid. Also, the number of Uplink requests sent to server increases, which results in an increase in the AQRT and size of IR reports. Hence, there is an increase in bandwidth consumption. This occurs in both UIR and SAS techniques but, in the proposed scheme, we have update manager module on server which keeps track of update of data rate and set mode bit as MB=0 for normal operation and MB=1 for fast operation. In the proposed scheme, IR is adaptive as it has only hot data inside it so, size of IR is not increased considerably because all updated data are not included in IR. Depending upon MB status, broadcast time of proposed scheme is set. Hence, during higher update rate, IR is generated and broadcasted after shorter interval of time which results in small query response time. Only updated hot data pages are sent to MT so more hits occurs and fast query response time is achieved. Where HIR is stored on BS then it helps disconnected MTs to validate their cache beyond the BT boundaries by making a demand from BS for HIR. The Uplink requests are sent only for cold data query requests with reduced frequency in the proposed scheme which results a reduction in Uplink requests.

- *Impact on AQRT:* Figure 4.7 shows the variation of an AQRT with Object Update rate of UIR, SAS, and ACIT. All the schemes have lower AQRT for smaller values of object update rate. But, it increases with an increase in Object Update rate in all the schemes. AQRT increases sharply in case of UIR because with such larger update rates, the data in cache becomes invalid and more Uplink requests are sent. In SAS, with higher update rates, AQRT becomes larger due to longer IRs and historical IR data. But, in ACIT, the variation of AQRT is not sharp as it has fast travel of IRs to MTs so, queries can be replied instantly with updated page as data is sent along with IRs. For the worst case scenario, object update rate is increased up to 50, where UIR and SAS exhibits larger AQRT and they may become less effective, while ACIT, due to its adaptive nature managed to have small values of AQRT.

- *Impact on IR report Size:* Figure 4.8 shows an impact of IR size on server. As size of IR becomes large with higher update rates then, there is large consumption of network resources such as bandwidth and memory. In SAS, carrying history of updates of data in IR to facilitate disconnected MTs makes size of IR larger which results a more resource consumption. Same is the case with UIR, with an increase in number of UIR reports are generated along with IR

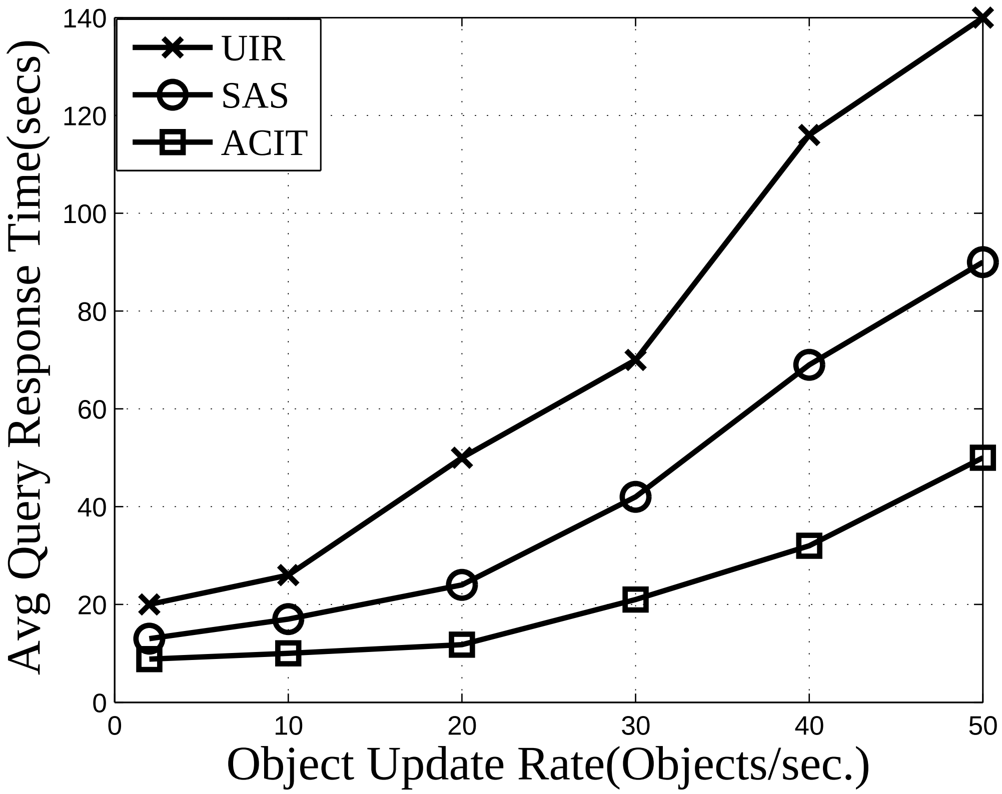

Figure 4.7: Variation of average query response time with Object Update Rate

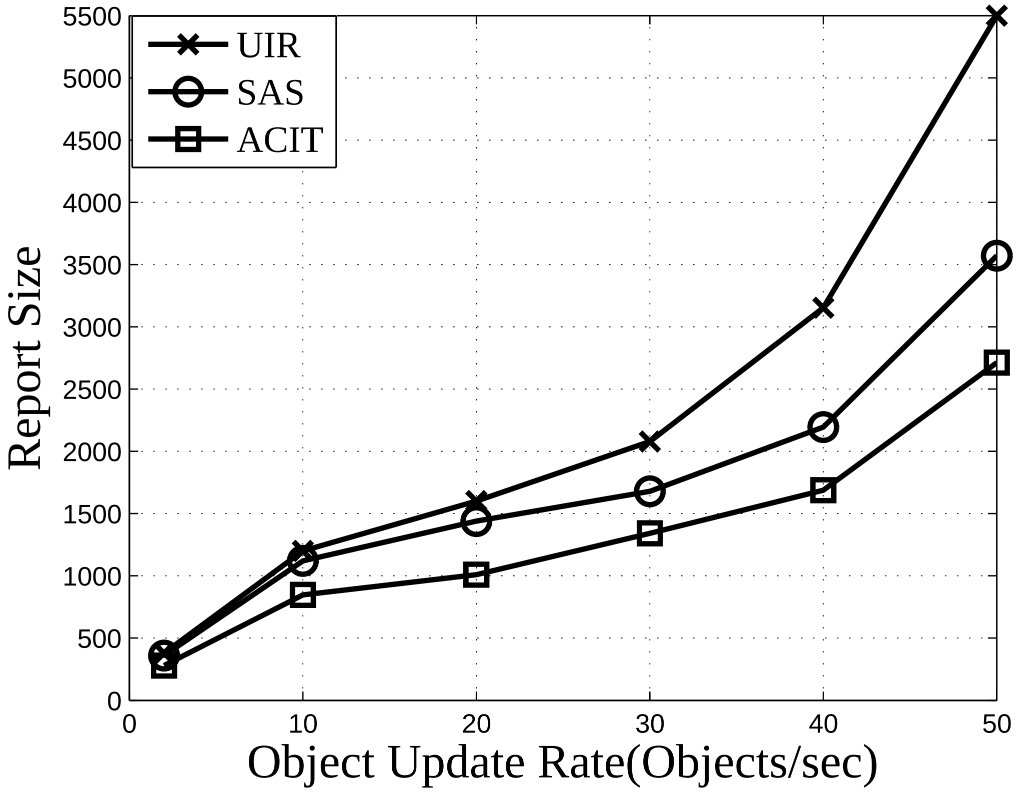

Figure 4.8: Variation of Size of IR using Object Update rate

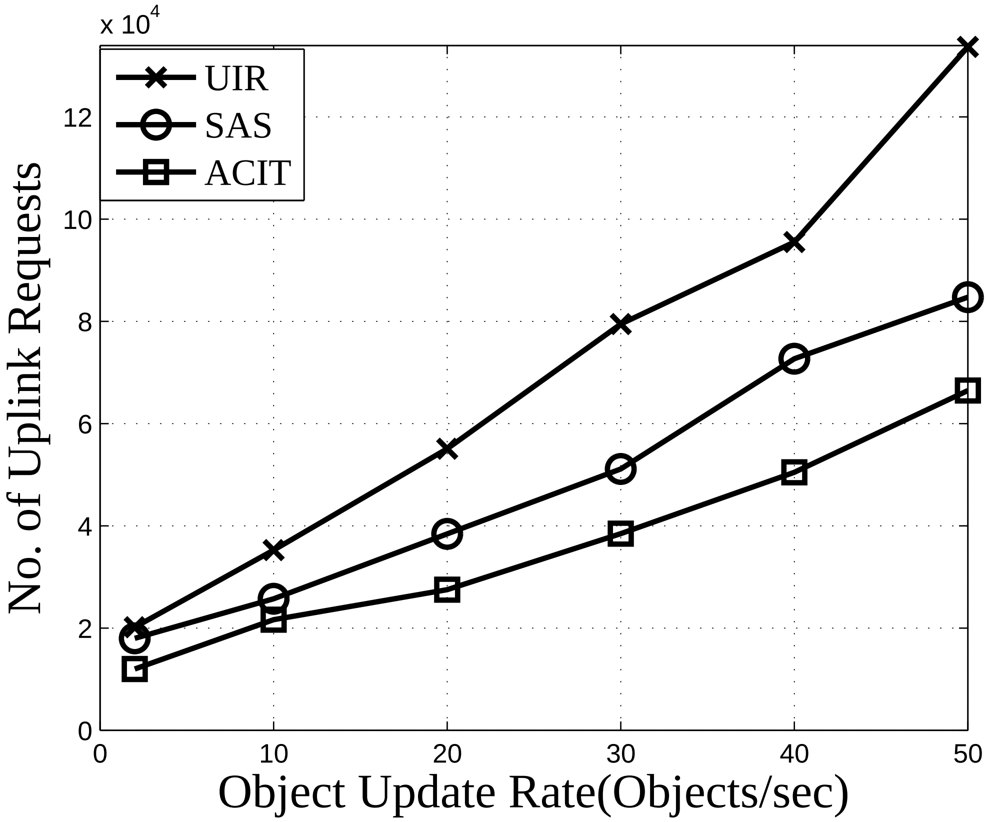

Figure 4.9: Number of Uplink request using Object Update Rate

report which significantly increases the size of IR report. But, in ACIT, only updates of hot data items are kept in IRs along with small historical updates information which results a smaller size of IRs. Moreover, when update rate goes higher from threshold update rate then, due to fast mode working, IRs are broadcasted faster so, query access delay is reduced in the proposed scheme.

- *Impact on No of Uplink Requests:* In Figure 4.9, with higher update rates, data in the cache becomes invalid in case of UIR and SAS. So, more Uplink requests are sent on server to validate a cache data. But, in case of ACIT, as update rate becomes higher, its BT is adapted and IRs reached at MT faster so, cache is invalidated and mostly hot data is pre-fetched by MT so, lesser Uplink requests are sent to the nodes in the hierarchy.

- *Impact on Broadcast Times:* As shown in Figure 4.10, BTs are fixed in UIR and SAS schemes but, it is adaptive in ACIT. BT is more when update rate is small and becomes smaller when update rate becomes more. This adaptive nature of of BT helps the proposed scheme to have smaller IRs with higher update rates.

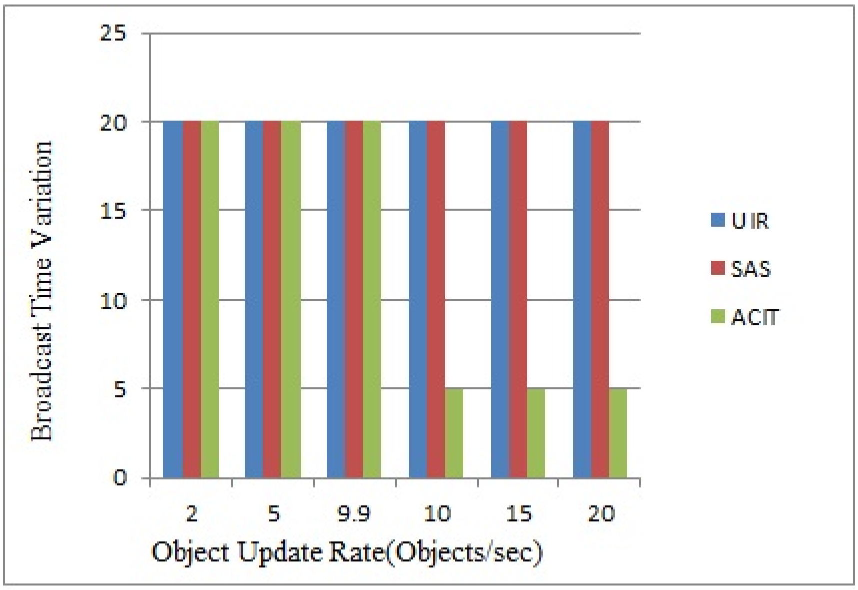

Figure 4.10: Variation of BT using Object Update Rate

4.7.2 Effect of Query Arrival Rate

To evaluate the performance of the proposed scheme, we have varied the query arrival rate or query arrival time using metrics such as AQRT and Uplink requests. Results obtained are discussed as follows.

- *Impact on Average Query Response Time:* Figure 4.11 shows the impact of query arrival rate on AQRT of UIR, SAS and ACIT strategies. Query arrival rate is varied from 125 queries/second to 1250 queries/second for each MT. Then, their AQRT is recorded and analyzed. For lower query arrival rates upto 500 query per second, AQRT of UIR and SAS increases, but due to hotness of the data item, queries are replied by cache copy of data on MT so, smaller query response time occurs in the proposed scheme. As there are less misses in ACIT so, less AQRT is observed with more query arrival rates in the proposed scheme. With lower query arrival rates, the probability of requesting a cached data which is updated on server becomes very less. When query arrival rate reaches 800 to 1250 queries/second then, performance of UIR degraded due to more misses so, queries are replied from server by an Uplink request. For SAS, queried data is looked with large IRs and histories so, more time is consumed in getting the reply for a query. Hence, ACIT has outperformed other two schemes as most of queries are answered from local cache which results a less query response time

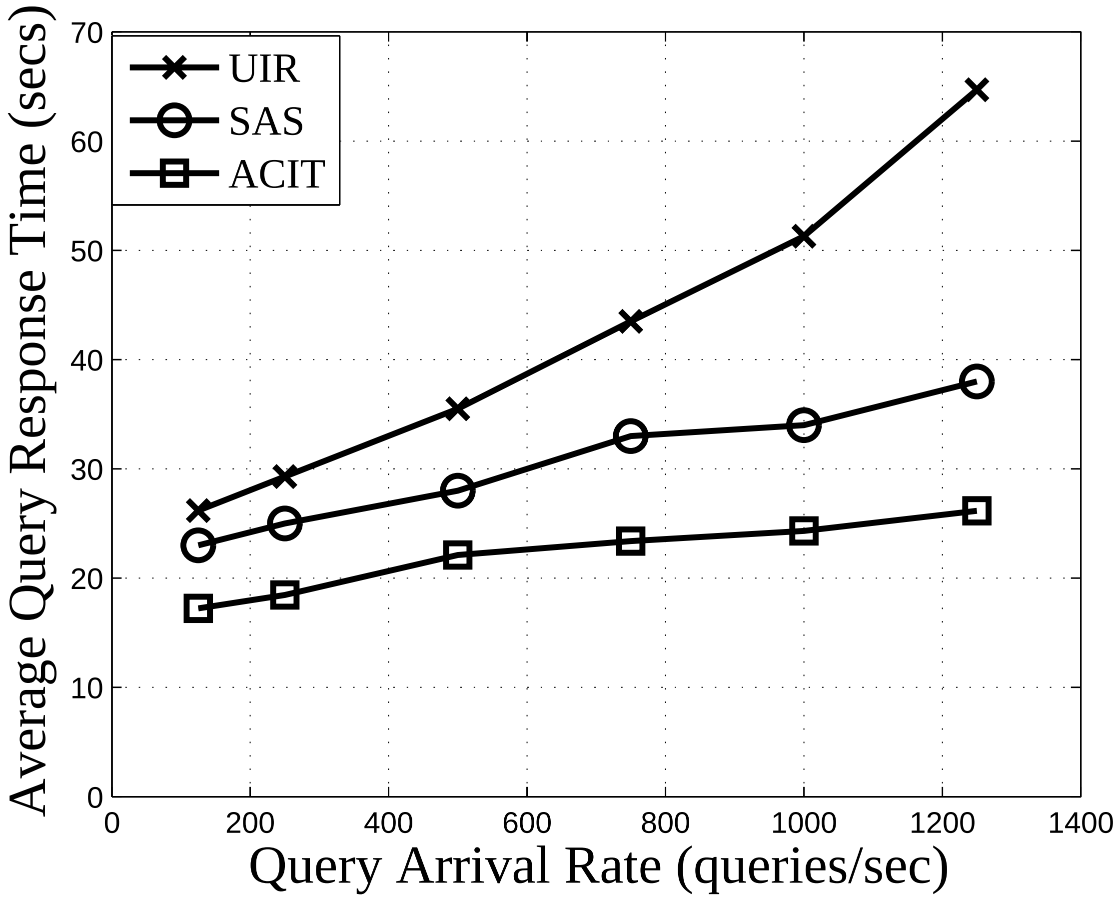

Figure 4.11: Average Query Response Time with respect to Query Arrival Rate

in the proposed scheme.

- *Impact on Uplink Requests:* Figure 4.12 shows that the Uplink requests increases as the number of queries increases because there is very less probability of queried data found and validated from cache in UIR and SAS schemes. But, in ACIT, there are higher chances of locating the queried data in cache because only hot data is kept in IRs and in cache. So, for UIR and SAS, as the need of queried data is raised, the number of misses occurs more frequently. So, more Uplink requests from MTs are sent for getting the requested data from the server. Therefore, more number of Uplink requests in UIR and SAS is generated but in ACIT the number of Uplink requests are very less which results an enhancement in the overall system performance.

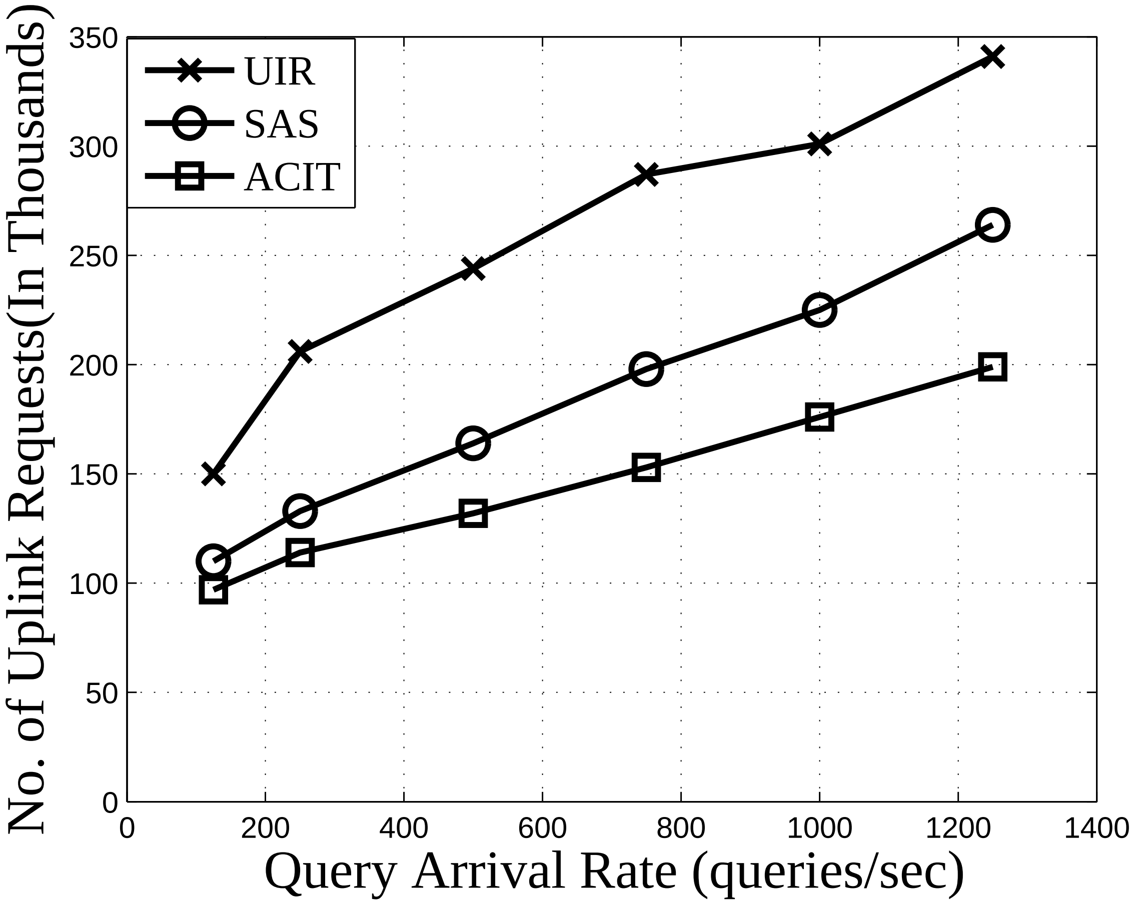

Figure 4.12: No of Uplink request with respect to Query Arrival Rate Rate

4.8 Conclusion

In this chapter, we proposed a new CIT called *ACIT for Wireless Environment"* which overcomes the higher query response time and more number of Uplink requests problem found in the existing state-of-the-art schemes in literature. The proposed technique includes only hot data updates in IR to reduce the size of IR which helps in better utilization of bandwidth and number of hits. The proposed ACIT has historical data with timestamps of w BT intervals for disconnected MT. So, MTs can sleep or disconnect to save power consumption in the proposed scheme. The proposed scheme works in fast mode with adaptive IR and broadcast time when update rate is higher than a predefined threshold value which results to have faster query replies. Due to inclusion of hot data in IR, cache contains only hot data content so, queries are replied from cache which reduces the number of Uplink requests. To evaluate the performance of ACIT, exhaustive simulations are performed by varying the parameters such as-update rate and query arrival rate and observed their impact on AQRT, IR report size, Uplink requests and broadcast time. Results obtained show the effectiveness of proposed ACIT scheme over other state-of-the-art schemes such as UIR and SAS.

Chapter 5

Cooperative Gateway Cache Invalidation- CGCI

In this chapter, a dynamic cache invalidation technique is proposed in Internet based Vehicular Ad Hoc Networks (IVANETs). In IVANETs, vehicles reside at lowest hierarchy. Then an access point, gateway foreign agent, home agent, and server resides at different levels of hierarchy. In this technique cooperation among the neighboring GFAs is used to increase the hit ratio during query response. So, a cooperative gateway cache invalidation technique [1]is proposed with supported algorithms. The results obtained are compared with existing techniques.

In recent years, IVANETs have emerged as a new class of efficient data dissemination. They establish communication among high speed mobile vehicles, and various other devices connected to the Internet for data sharing and communication [112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128]. In IVANETs, vehicles act as an intelligent machine for data collection and processing from various other devices connected to the Internet using IEEE 802.11 a/b/g standards along with short, medium and long range transmissions such as WiFi, Bluetooth, WiMAX, and Long Term Evolution (LTE). For connecting any device like laptop, PDA, and MT using these standards with Internet, the device must be uniquely identified on global positions. All these devices are considered as objects on the Internet and their inter-

[1]The contents of this chapter have been peer reviewed and publications in
Rajeev Tiwari, Neeraj Kumar, "Cooperative Gateway Cache Invalidation Scheme for Internet-Based Vehicular Ad Hoc Networks", Wireless Personal Communications, 85(4): 1789-1814 (2015). **(SCI-Thomson Reuters Rated Journal with Impact Factor 0.653)**

connection with each other for communication and data sharing is called as Internet of Things (IoT) [112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 124, 125, 126, 127]. Connecting vehicles to Internet requires connecting infrastructure with wireless MTs. The mutual exchange of safety information between vehicles for safe driving is one of the applications of IoT in IVANETs[113, 114].

To maintain the wireless communication among vehicles is one of the most challenging tasks. The main objective of communication is to provide safety to the on board passengers [124, 127, 128, 129] in wireless environment. Vehicles act as intelligent machines in modern era because of advances in communication technologies. Now Global Positioning System (GPS), digital maps and wireless interfaces [125, 129, 130], large cache memory are embedded in vehicles from all leading companies, so that user can access various resources from the Internet easily. There are many state-of-the-art protocols available for caching in IVANETs, but these solutions are not sufficient to provide quality of service (QoS) to various applications in these networks. This is because of cache inconsistency, more communication costs, and longer delays of query reply. Moreover, vehicles have high mobility as nodes in IVANETs lead to constant topological changes[112, 131, 132, 133]. So, maintaining a cache consistency with high mobility and constant topological changes are the most challenging tasks which requires a special attention in this environment.

One of the key objectives for improving the performance of IVANETs is to cache the frequently accessed data items at local sites. In this environment, the critical design issue is the cache invalidation, since all applications require consistency with the server to avoid any inconsistency whenever data items are updated at the server. When a data item at the server is updated, it must be invalidated in cache of vehicles by broadcasting an IR. However, due to fast roaming of vehicles, this task is most challenging as vehicles may change their positions quickly. During the data access by the vehicles from the server, query delay, and Uplink requests need to be minimized whenever frequent data updates occur.

From the above discussion, it is clear that delay is a major issue in accessing services when MTs have high mobility in IVANET environment. One of the feasible solutions to reduce it is to make use of caching of frequently accesses data items to minimize the network traffic and enhance the performance. So, caching of popular data in MTs or at nearest APs may generate query response quickly, as query request can be replied from local nodes instead of remote servers. However, data consistency must be maintained between MTs and servers to prevent clients from answering to those queries which are invalidated from the server. A data item is invalidated if a query to access the data item has been generated from the clients and it has been updated by the server and clients are still using the older data for query replies.

In literature, there exists many CITs for IVANETs which are described as follow. TS [23] collects all updates pages for IR and broadcasts it to MTs after a fixed BT interval of L seconds. Due to the mobility, if MT missed IR, then queried data cached in MT is validated from remote server which incurs longer delay. In UIR[24] technique, a smaller fixed size m UIRs are sent between BT times and at a BT time, fixed IR is sent. In SAS, a GIR [28] is generated which has a longer delay without cooperation with the gateway. Many of these proposals are based on IR to handle frequent disconnections [23]. CCI and ECCI [41] are schemes which are cooperative and stateful, where a server and network agents coordinate together for invalidation of cache data, but these techniques missed cooperation of GFA which may reduce the longer delay for many queries generated from the clients in this environment.

5.1 Motivation

Most of the above discussed techniques[23, 24, 28, 29] suffer from longer query delay and large bandwidth consumptions due to the longer time spent in validating the cached data items from server. So, in such a case, MTs can't reply for query till its cached data is validated by IR sent from server which may increase the query response time for MTs. These schemes have not exploited the advantage of cooperation of proxy servers or other network components. Authors in [24] have proposed an UIR-based cache technique in which a server sends UIRs after every L/m^{th} seconds, where $m < L$. The sending of UIRs may decrease query response time to the extent of hit occurrences, due to UIRs on MT. But, the replication of UIR instead of IR consumes a lot of bandwidth and also increases the congestion in the network. Due to this, query response may take a longer time with a large number of queries missed with respect to the UIR report. Due to high mobility in vehicles, they may reside in multiple coverage area of different GFAs. So, they frequently hands off the IR sent by the server which may results a longer query delay. This type of delay can be reduced by cooperation among gateways.

Moreover, when one GFA is not able to validate the queried data of vehicles, then other neighboring GFAs may have valid copy of queried data. So, there is a need of cooperation among the multiple GFA to increase the efficiency of any cache management scheme in this environment.

We have also motivated from our earlier solutions [60, 112], in which we have designed cache management scheme for peer-to-peer (P2P) mobile clients in vehicular environments where the cooperation of network components like GFAs has not been considered. But, in the current solution, we have considered this factor in different environment and tested the performance of the designed scheme using cooperation of GFAs.

5.2 Methodology and Contribution

Based upon the above discussion and challenges in IVANET environment, in this work, following contributions are presented.

i) A hierarchical network model for IVANETs is presented which consists of APs, GFAs, HAs, and data server with cooperation from neighboring GFAs. The proposed scheme consists of a stateful, location management with cooperation from the neighboring GFAs.

ii) A cooperative scheme is proposed in which both the server and HA coordinate for cache invalidation and server asynchronously sends an IR to a HA rather than blindly broadcasts to vehicles. Then, the HA judiciously filters and distributes the IR to appropriate GFAs and cooperation among neighboring GFAs is maintained which answer the validity of the queried data item to reduce the unavoidable query delay which was one of the major issues in most of the existing invalidation techniques.

iii) An analytical model is developed for predicting the query delay of the proposed Cache Invalidation algorithm with cooperation from the neighboring GFAs.

iv) We have modified the previous cache invalidation schemes: TS[23], CCI and ECCI [41, 71, 72] to make them suitable for IVANETs environment. The designed scheme is compared with the above discussed schemes with respect to various performance evaluation metrics.

5.3 System Model

This section illustrates the network and, mobility models along with the problem definition for more clarity of the proposed technique.

5.3.1 Network Model

The hierarchical network model for IVANET consists of infrastructure support such as-APs, GFAs, HAs, and servers for data residency. In this model, GFAs have cooperation with one another. Vehicles have IEEE 802.11-based Dedicated Short Range Communication (DSRC) transceiver. So, vehicles can either communicate with one another or connect to the Internet. For providing a location tracking capability in vehicles, they have a built-in navigation system using GPS with a digital map to know their locations. We have considered a hybrid architecture in which APs and GFAs are deployed in such a manner so as to connect MTs to clients in different network domains. APs and GFAs are statically connected to S as shown in Figure 5.1.

5.3.2 Mobility Model

In the mobility model, vehicles are bounded to the underlying fixed roads, mostly linearly with speed limits and traffic lights in IVANETs. Vehicles communicate with other vehicles or to the fixed infrastructure components such as APs. When a vehicle accesses the Internet, it looks for its geographical region AP, which is a gateway to the Internet for accessing information and services. Also, APs play an additional role to perform a location update/register operation when a vehicle moves from one coverage area to another that involves an AP, a GFA, and HA. So, whenever hand off occurs, location of vehicle is maintained in GFA, and HA. So, when HA has to re-distribute IRs to vehicles, it sends it to new GFA's location in which a vehicle resides.

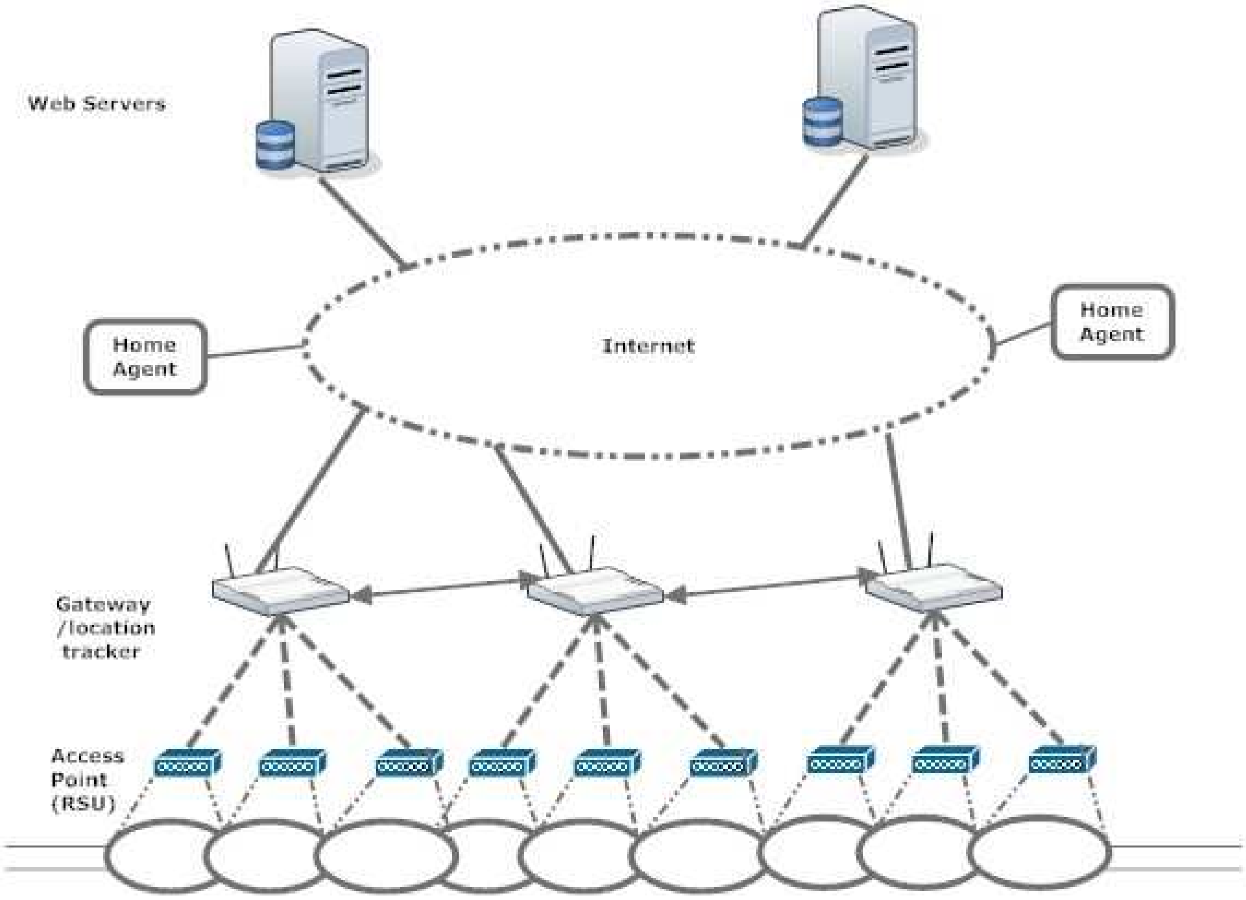

Figure 5.1: Network Model Used

5.3.3 Problem Formulation

There are a number of challenges for validating the cache consistency of vehicles in an IVANET. In a vehicle initiated cache invalidation approach, MTs send Uplink requests to server to invalidate the cache. So, as the number of MTs increases, they generates more Uplink requests which results in generation of congestion to the Uplink channel. Due to the mobility, vehicle may hand off from its current GFA location to neighboring GFA. By considering the hot data access of a geographical region, its popularity would be same in neighboring GFAs. So, there are fair chances to get the requested data from the neighboring GFAs. Hence, we have exploited GFA cooperation in IVANET environment, which was missed from earlier CIT techniques. As in ECCI, GFAs cache data items in their local storage for future query requests from vehicles. So, GFAs serve the requests by checking validity at GFA and reply to vehicles query quickly to minimize the query response time and reduces the Uplink request considerably. So, hot data pages are cached on GFAs. Hence, by making cooperation of neighboring

GFAs, queries are replied faster as round trip time of MT to GFA is lesser than round trip time of MTs to server.

Following are the objectives of the proposed technique.

$Minimize\{Query\ Response\ Time(QRT)\ and\ Uplink\ Requests\ (UR)\}$

Subject to the following constraints,

$$\begin{cases} P_{hit} + P_{miss} = 1 \\ P_{hit} \geq P_{miss} \geq 0 \\ P_{GFA}^{hit} \geq 0 \\ \tau_{GFA} \leq \tau_{server} \end{cases} \tag{5.1}$$

Eq.(5.1) is to minimize the query response time for each query reply. Because we have not considered any other state in cache except hit or miss, so probability of hit and miss is considered as one. If we consider the hot data, then probability of hit is greater than probability of miss. So by considering LRU page replacement algorithm, hot data is kept in cache. So, we have finite value of P_{hit}, i.e., $P_{hit} \geq P_{miss} \geq 0$ which results $P_{GFA}^{hit} \geq 0$. Moreover, as GFA is located near to roads as compared to servers deployed remotely, so RTT from vehicle to GFA is lesser than from vehicle to server. Hence, queries replies are generated between vehicle cache, GFA, and neighboring cache by using the cooperation among these entities. Hence, lesser Uplink requests are generated towards server which generates a less overhead to maintain the cache at various levels in IVANETs environment.

5.4 Detailed Description of Scheme

Based upon above issues and challenges, this section illustrates the detailed description of the problems discussed above. In the architecture shown in Section 5.3.1, every vehicle has cache memory with it. A server has all the data D so that all updates of data items occurs only on the server. Here, server is a stateful and synchronous in nature. So, a hybrid cooperative scheme is proposed with cooperation of infrastructure components and neighboring GFAs. As in ECCI, whenever a miss occurs in cache of vehicle, queried data is requested from server and then the GFAs cache data items in their local storage. So, GFAs serve the requests by checking the validity at GFA and reply to vehicles query quickly. We have enhanced ECCI by considering cooperation of neighboring GFAs.

When neighboring GFAs start cooperation among themselves, i.e., they send request and reply to each other. As in ECCI, GFAs maintain a list of updated valid pages so that they can fulfill future needs of vehicles.

When a vehicle requests for a page and is not cached in AP, then it looks into GFAs for updated valid page. If data item is presented, then it replied back. Otherwise, it looks into server (remote or far away) for the requested page. GFAs cooperate with one another. So, requested page can be looked into neighboring GFAs cache. Neighboring GFA cache has a probability of P^{hit}_{GFA} of finding the requested page in their cache. So, if updated and validated page exists at neighboring GFA, then page is replied back to home GFA to its AP to the vehicle.

5.4.1 Cooperative Approach

We have designed a cooperative stateful approach, where a server and HAs of location management coordinate together for cache invalidation operations along with cooperation of neighboring GFAs. The server and HAs have a list of data item accessed by vehicle and are aware of vehicle's *id*. Thus, whenever a data item is updated, it sends an IR to an HA which further unicasts the intended data to vehicles only. In CGCI, major cooperation is of neighboring GFAs for validating the data requests and query responses. After a fixed time interval t, GFAs exchange their list of pages with neighboring GFAs to make a $Valid\_Unified\_list$ which is refereed when any miss occurs from a vehicle cache.

5.4.2 Design and Structure

Each GFA has a updated registry (list of updated pages). They exchange this update registry among neighboring GFAs after a fixed time. Then, every GFA has a consolidated list named as $Valid\_Unified\_List$ of its own updated pages plus list of neighboring GFA updated pages. So, when a request for a non cached page comes to GFAs, then GFAs check whether it is presented in its own list or in its neighboring GFA. If it is presented in neighbors GFA, only then request is sent to neighboring

GFAs. Otherwise, it acts like ECCI [41]. So, if there is any chance of negotiating query delay only then CGCI comes into action: Otherwise, ECCI works well. Modifying ECCI i.e., CGCI technique in this manner provides the following advantages.

- Faster Query Response Time: Because data requests after missing from cache of vehicles are resolved from cooperating GFAs instead of remote server, so, queries are replied faster as follows:

  $\tau_{GFA} \leq \tau_{server}$.
- Lesser Congestion on server occurs, as most of the requests are fulfilled at lower level GFAs
- Higher hit rates: There is an increase in hit rates as it is the contribtion of two terms as- hit rate of GFA (similar to ECCI) + hit rate of neighboring GFAs
- More efficient utilization of GFAs.

5.4.3 Detailed Design

In this subsection, we have described the detailed operation of the CGCI scheme and cooperation of neighboring GFAs. The infrastructure components are same as proposed in [41]. Stateful server is used to keep track of used list, i.e., which data d_x is used by which vehicle node, where $d_x \epsilon D$, and D is set of all data items on server. On updates of any data on the server after a fixed time interval of L seconds, an IR is generated and sent to HA. It contains records of all the vehicles who roam under their GFAs. Then, IR is regenerated depending upon which data is cached under which vehicle and GFA. Depending upon number of matched GFAs, registry entries are created, Then, this newly generated registry entries are unicasted to corresponding GFAs and to the vehicles. When a vehicle connects to a server, then all its data cached and time stamp $< dx, ts, COA_V >$ are recorded on the server. Whenever a server gets a request for data validation, then it also updates data residency in vehicle and its time stamp ts as t_{curr} in its records. When a vehicle handoffs to the coverage area which is under the different GFA, then the HA updates its registry and forwards the updated pages to the new GFA when a cached data item is updated in the vehicle. The data item is also forwarded to the appropriate APs to reduce the query latency, but it may generate an extra overhead for the GFA, if a vehicle frequently moves among the coverage areas. So, handoffs are managed using location agents and GFAs.

According to proposed CGCI technique, let us describe the query response and cache invalidation mechanisms. When a vehicle generates a query for data item d_x, following procedure is executed. Firstly, when data is cached in vehicle and is validated after getting registry entry from HA or sending a DRP to AP, then acknowledgment packet is sent to vehicle directly to validate the data cached. Then, vehicle replies to queried data fastly with cached data. Secondly, when data is not cached in cache or not invalidated from AP or GFA then DRP is appended with COA_{AP} and forwarded to GFA. DRP is then checked with $Valid\_Unified\_list$. If requested data request is matched with this unified list of neighboring GFAs, then request is forwarded to neighboring GFA instead of sending it on server. Then DRP is validated and requested data is sent to corresponding GFA which sends it further to desired vehicle using AP. Algorithm for GFAs with cooperation is given below.

To describe the functioning of cooperation of GFAs, a sequence diagram is given in Figure 5.2 describes following set of sequence of steps.

Step 1.1: As vehicle request for data using AP, then the DRP is forwarded to GFA using AP.

Step 1.2: AT GFA, validity of requested data is checked. If found valid then.

Step 1.3: Reply with an acknowledgment packet and vehicle replies with cached data.

*Step 1.4:***If** (not validated on GFA) **then**

Step 1.5: Append COA_{GFA} to DRP as DRP, COA_{GFA}.

Step 1.6: Then, GFA's $Valid\_Unified\_List$ is refereed to see if DRP is present on neighboring GFA. If presented on neighboring GFA, then

Step 1.7: Sends DRP, COA_{GFA} to neighboring GFA.

Step 1.8: Neighboring GFA replies with data.

*Step 1.9:*Then, at parent GFA, registry is updated and page is saved to fulfill future requests.

Step 1.10: GFA replies with requested page and vehicle replies with it.

Algorithm 5 GFA Cooperative Algorithm

```
1: Input w, & w = n * t_bt. Server Keeps Track of Update of Data
2: Output: DRP to neighboring GFA or Requested Data Page to Vehicle.
3: When GFA receives request from AP
4: if DRQ matches Cached Data then
5:    Reply an Ack. to via AP
6: else
7:    Append to request packet
8:    if requested data is present in neighbor GFA then
9:       Send this requested page to neighbor GFA
10:   else
11:      Send request packet to server
12:   end if
13: end if
14: When GFA Receives request packet from neighbor GFA then
15: Reply requested page to GFA
16: When GFA receives a reply packet from GFA then
17: Go to step 24.
18: When GFA receives a reply packet from server then
19: Update its registry with updated page and store it in its cache
20: Update list of valid pages.
21: When GFA receives an IR from HA then
22: Maintains list of updated pages for validation
23: After a fixed interval: Call Valid_Unified_List();
24: Valid_Unified_List() {
25: Send list of stored pages to its neighbors
26: Receive list of stored pages from neighbors GFA
27: Create the unified list of stored pages of its own and neighbors.
28: }
```

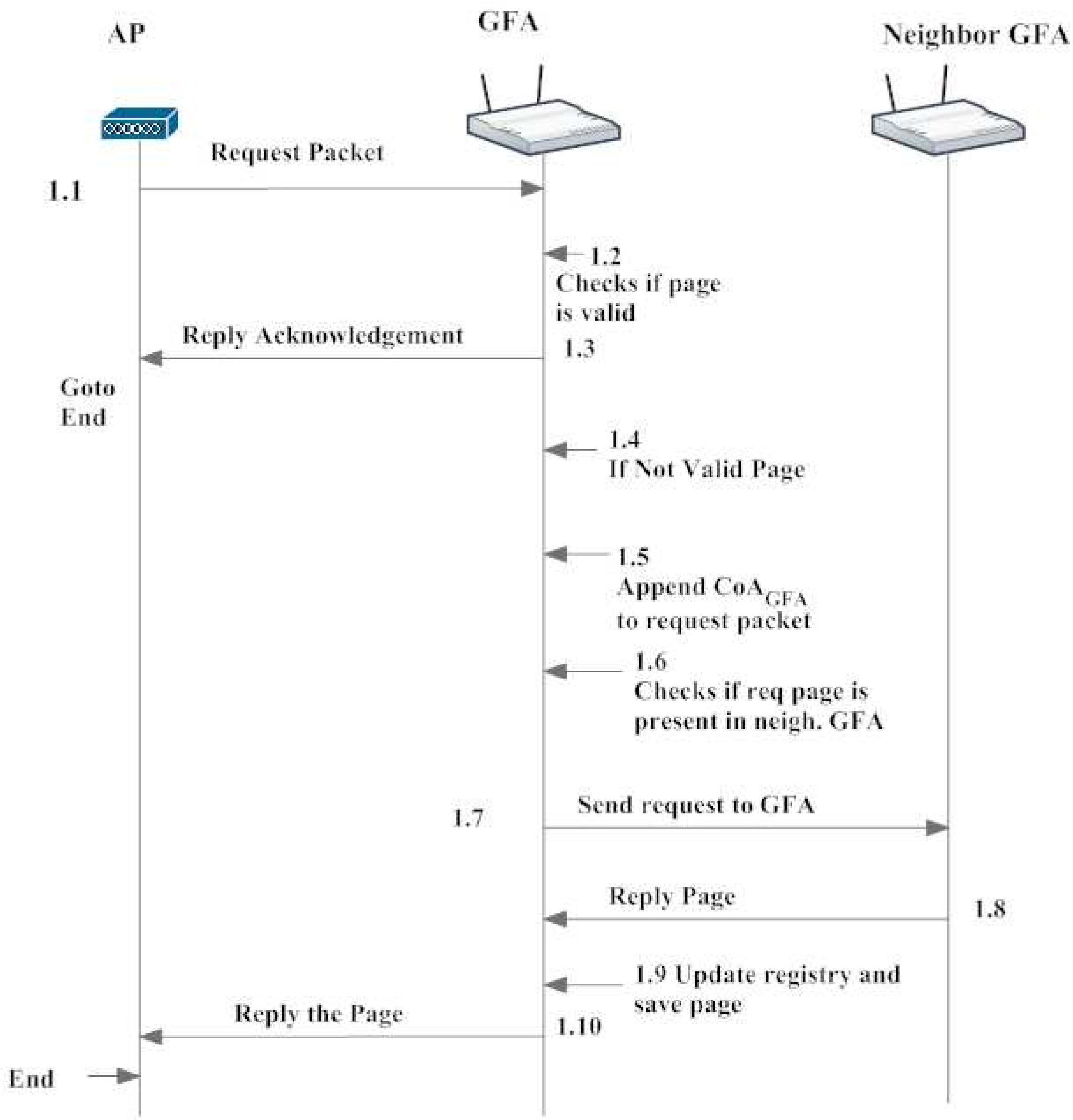

Figure 5.2: Sequence Diagram Used for CGCI

5.5 Analytical Model

In this section, we have presented an analytical model for computing the cost of the proposed CGCI. In this model, we have computed the cost for query reply using cooperation among neighboring GFAs thus providing the mathematical formulation for Query Response Time, where arrival of queries and updates are taken as a Poisson distribution.

5.5.1 Query Response Time and Uplink requests

$$\begin{aligned} QRT &= P_{hit} \times (TimeSpent1) + P_{miss} \\ &\quad \times (TimeSpent2) \end{aligned} \tag{5.2}$$

where P_{hit} is the probability of getting a hit of queried data on GFA, and P_{miss} is the probability of getting a miss from GFA. $TimeSpent1$ is the time spent when hit occurs in local cache of vehicle for queried data and $TimeSpent2$ is the time spent when miss occurs in cache of vehicle when the data is looked from GFA, neighboring GFA, and server.

The $TimeSpent1$ is considered as a hit case, when queried data is in cache and is validated from GFA using AP. So, query processing occurs twice on AP, i.e., to-and-fro of data packets. Once query processing occurs on GFA, then twice query processing delay is incurred for Vehicle and AP to GFA and has a factor $2 \times k \times (N_{gfa}^{ap} + w_c)$ so the combined value can be written as follows.

$$\begin{aligned} TimeSpent1 &= 2 \times QP_{ap} + QP_{gfa} + 2 \times k \times \\ &\quad (N_{gfa}^{ap} + w_c) \end{aligned} \tag{5.3}$$

Moreover, $TimeSpent2$ is the time spent during miss occurrence. It consists of cost of looking data from neighboring GFA due to cooperation and cost of looking data from server. So, in CGCI technique P_{miss}, can be written as follows.

$$P_{miss} = P_{GFA} + P_S \tag{5.4}$$

where P_{GFA} and P_S are the probabilities of finding data from neighbor GFA and finding such missed data on server. So, $TimeSpent2$ can be written as below.

$$TimeSpent2 = C_{query}^{miss}(GFA) + C_{query}^{miss}(S) \tag{5.5}$$

$$\begin{aligned} (P_{miss} \times TimeSpent2) &= \{P_{GFA} \times C_{query}^{miss}(GFA) \\ &\quad + P_S \times C_{query}^{miss}(S)\} \end{aligned} \tag{5.6}$$

Here, $C_{query}^{miss}(GFA)$ is the total time spent for query data validation on miss occurrence from neighboring GFAs, and $C_{query}^{miss}(S)$ is the total time spent for query data validation on miss occurrence from server which is similar to ECCI [41] scheme.

Moreover, $C_{query}^{miss}(S)$ and $C_{query}^{miss}(GFA)$ are given in terms of delays of infrastructure components as given below in Eqs.(5.7, and 5.8).

$$\begin{aligned} C_{query}^{miss}(S) &= \{2 \times QP_{ap} + 2 \times QP_{gfa} + QP_s\} + \\ &\{N_{gfa}^{ap} \times k + N_{gfa}^{s} + w_c \times k\} + \\ &\{N_{gfa}^{ap} \times k \times n + N_{gfa}^{s} \times k \times n + \\ &k \times n \times w_c\} + \{w_c \times k + N_{gfa}^{ap} \times k\} \\ &+ \{k \times n \times w_c + N_{gfa}^{ap} \times k \times n\} \\ &+ 2 \times QP_{gfa} \end{aligned} \quad (5.7)$$

$$\begin{aligned} C_{query}^{miss}(GFA) &= \{2 \times QP_{ap} + 2 \times QP_{gfa} + QP_s\} + \\ &\{w_c \times k + N_{gfa}^{ap} \times k\} + \\ &\{k \times n \times w_c + N_{gfa}^{ap} \times k \times n\} \\ &+2 \times QP_{gfa} \end{aligned} \quad (5.8)$$

So, Eqs.(5.2) can be rewritten using Eqs. (5.3,5.6,5.7, 5.8) in terms of delay constants of infrastructure components and delay of neighboring GFA as follows.

$$\begin{aligned} QRT &= P_{hit} \times (2 \times QP_{ap} + QP_{gfa} + \\ &2 \times k \times (N_{gfa}^{ap} + w_c)) + \{P_{gfa} \times C_{query}^{miss}(GFA) \\ &+ P_S \times C_{query}^{miss}(S)\} \end{aligned} \quad (5.9)$$

Using Eqs.(5.8), P_{GFA} times the query miss is validated from neighboring GFAs instead of getting valid data from server. So, it reduces the Uplink requests of validation of data leaving upper side network less congested.

5.6 Performance Evaluation of CGCI

5.6.1 Simulation Settings

We have customized discrete event network simulator to evaluate the performance of the proposed scheme. We have used simulator NS2 [101, 102] for simulation of the designed scheme. We have considered a scenario consisting of wired and wireless connection with hierarchical addressing, comprising of domains and clusters in network topology. We have taken 10 Vehicles, 3 GFAs, 2 APs, 1 HA, and 1 server with all data items. We have deployed 2 APs to cover area under a GFA and considered AP situated centrally. Then, 10 vehicles are deployed on a straight track with velocity of 30 km/hr which is suitable for local city driving. We have considered 100 Mbps bandwidth of infrastructure network considered in the proposed scheme, i.e., between server and HA. Both update and query arrival times are exponential distributed with query arrival and update as Poisson's distributed. We have taken 10% of total data D as hot data. The simulation parameters are summarized in Table 5.1 given below.

Table 5.1: Parameters used and their values in CGCI

Parameter	Default Value
Number of Vehicles	10
Database Size, D	65536 data items
Cache Size, C_{size}	500 to 2500
Broadcast window size, w	10
Broadcast time,L	20 seconds
UIR times,m	4
Data id, d_x	32 bits
Data Size	2048 bits
Time Stamp Size	64 bits
Distribution of query arrival times	exponential
Distribution of update arrival times	exponential
%age of hot data items, H	10 % of D
%age of cold data items, C	90 % of D(Remainder)
topography	670*670m
Simulation time	1000s
Routing protocol	DSDV
No of GFA	3
No of AP	2
No of Server	1

5.7 Results and Discussion

To evaluate the performance of the CGCI technique, we have modified earlier existing techniques to make them suitable for IVANET environment. We have considered the existing benchmarked techniques for comparison as-No cache (NC), UIR, SAS, CCI, and ECCI [24, 28, 41, 71] as described in chapter 2. The results obtained are discussed as follows.

5.7.1 Effect of Object Update Rate

When object update rate increases on server, then the cache miss ratio also increases on vehicle side because most of the requested data items are stored in the cache of vehicle and are invalid. Also, as the number of Uplink requests sent to server increases, then there is an increase in query response time as queries are answered from server. This occurs in NC, UIR, SAS, CCI, and ECCI techniques but, in our proposed scheme, we have cooperation among neighbor GFAs which contains list of valid hot data. So, queried data is validated from GFAs instead of server. Moreover, round trip time of vehicle to GFAs is lesser than vehicle to server for getting data from server. So, there are more chances of getting valid data either from GFA or from neighboring GFAs in the proposed scheme, i.e.,

$\tau_{GFA} \leq \tau_{server}$.

So, majority of queried data requests are satisfied or validated from GFAs or through neighboring GFAs. The Uplink requests are sent only for cold data query requests with reduced frequency in the proposed scheme which results a reduction in Uplink requests. For reference, we have taken the base case of NC scheme where, query delay is very large and higher number of Uplink requests occurs.

- *Impact on Query Response Time-* Figure 5.3 shows the variation of QRT with Object Update rate of NC, UIR, SAS, CCI, ECCI, and CGCI. All the schemes have lower QRT for smaller values of object update rate. But, it increases with an increase in Object Update rate in all the schemes. QRT is very large in NC case as it has no lacal cache to respond even to frequently accessed pages. It increases sharply in case of UIR because, with such larger update rates, the data in the cache becomes invalid and more Uplink requests are sent. In SAS, with higher update rates, the QRT becomes larger due to longer IRs and historical IR data. In CCI and ECCI, it has been reduced and not varied sharply as it has considerations of hot data in cache and cooperation of location agent and HA. But, in CGCI, the variation of QRT is not much as it has cooperation of GFAs. So, queries can be replied instantly with updated page as data is sent along with IRs. Otherwise can be resolved from cooperative GFAs.

- *Impact on Uplink requests:* In Figure 5.4, with higher update rates, data in the cache becomes invalid in other existing techniques. Once any miss occurs for queried data then, in these techniques data is brought only from the server and there is no intermediate component or any cooperation to invalidate the data at lower ends to reduce the Uplink requests. Only in ECCI technique, it has some lesser Uplink request as it's GFA has an enhancement of storing the popular hot data in it and can respond directly to queried data in future. So, in other

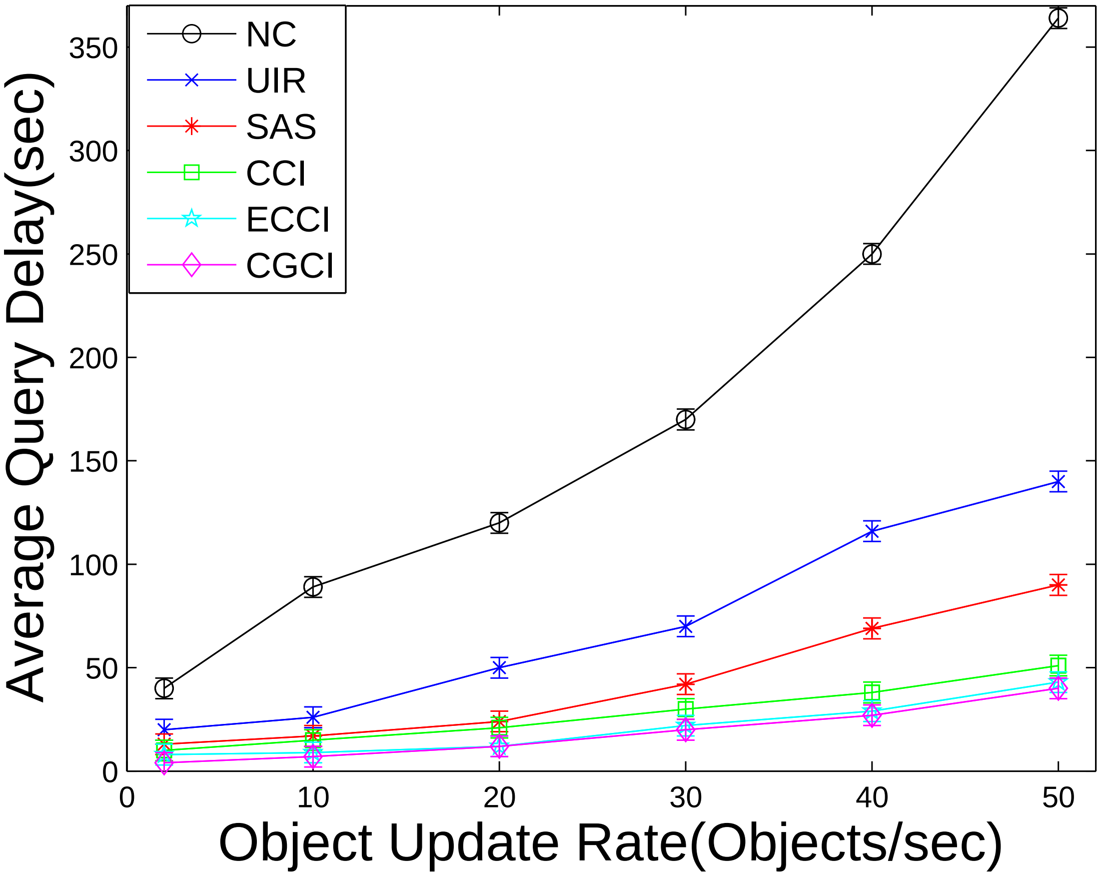

Figure 5.3: Variation of Query Delay with Object Update Rate

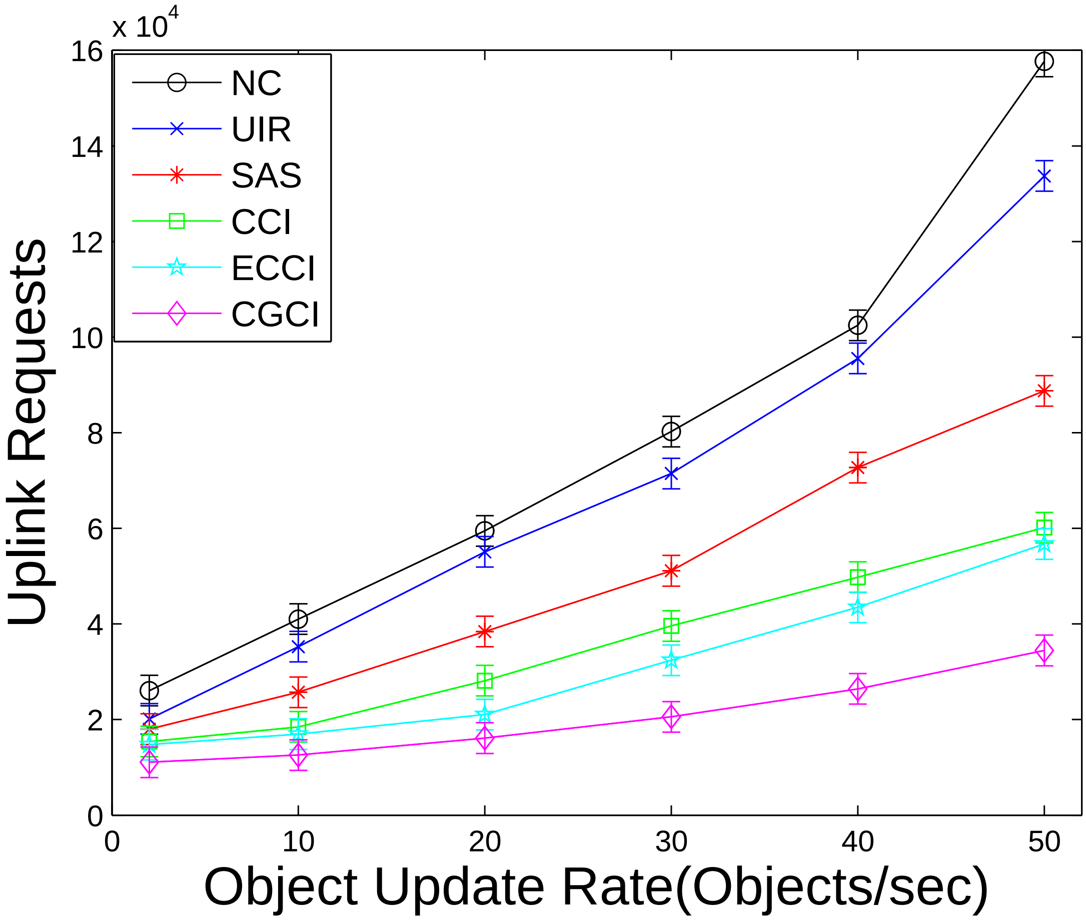

Figure 5.4: Variation of no of Uplink Requests with Object Update Rate

cases, more Uplink requests are sent on the server to validate a cache data. But, in case of CGCI, as update rate becomes higher, hot data is considered which has more chances of being cached and due to cooperation among infrastructure components, updated data reaches to GFAs and cooperating GFAs validate the data of vehicles. So, lesser Uplink requests are sent to the server in hierarchy.

- *Impact on Hit Ratio:* As shown in Figure 5.5 with higher update rates the hit ratio of all the schemes is decreased. In UIR, it falls sharply with an increase in update rate. In SAS, there is a sharp decrease of hit ratio. But in CCI, the hit ratio is not decreased sharply as it has considered the hot data cached in the cache of vehicles, whose updates are stored at corresponding GFAs by HA. In ECCI, the decrease in hit ratio is gradual as it has GFA, which stores hot data pages in it and responds on time. But in CGCI technique, hit ratio is good and is not decreased sharply as neighboring GFAs have cooperation among themselves

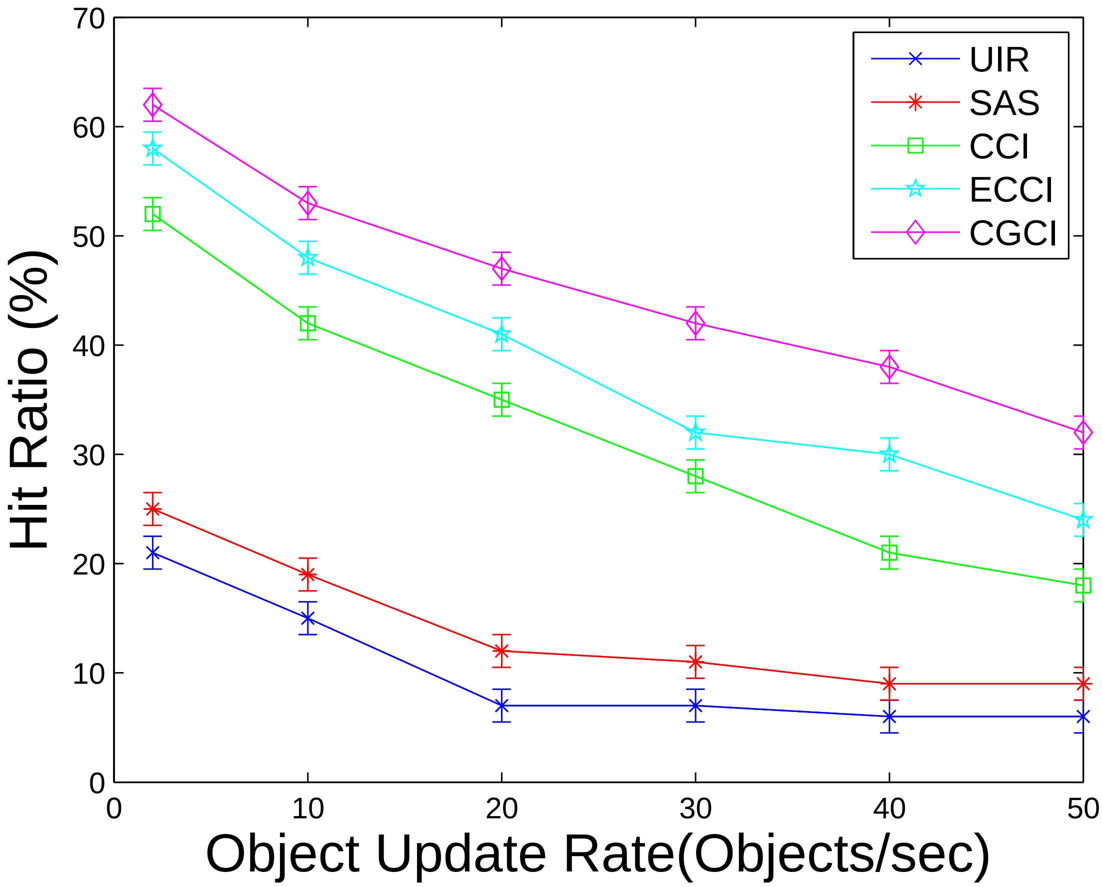

Figure 5.5: Hit Ratio (%) with Object Update Rate

and data is validated from the corresponding GFAs and its neighboring GFAs after checking the meta data of $Valid\_Unified\_List$ of GFAs.

5.7.2 Impact of cooperative approach

Techniques such as NC, UIR, SAS, CCI have no cooperation among nodes so they have larger query delays, more Uplink requests, more congestion and lesser hit ratio. But, we have exploited the cooperation of neighboring GFAs in CGCI. So, cooperation has influenced effectively for hit ratio of data, i.e., data is validated quickly and queries are replied from local components. As P_{GFA} is the probability of finding data from neighboring GFA so, there is an increase in the hit ratio. As queries are replied from lower side of hierarchy, so upper side of the network is very less congested which increases the network performance. So, cooperation effect can be seen directly on query response time, hit ratio, and Uplink requests.

5.7.3 Effect of Mean Query Arrival Time (MQAT)

When MQAT is smaller on server, i.e., more number of queries are generated per unit time, then cache miss ratio increases on vehicle side because most of the requested data items stored in the cache of vehicle are invalid due to limited size of cache. Also, the number of Uplink requests sent to server increases, which increases the query response time as queries are replied from server. This occurs in NC, UIR, SAS,CCI and ECCI techniques but, in the proposed scheme, due to neighbor GFAs and their cooperation, queried data is validated from GFAs instead of server. So, majority of queried data requests are satisfied or validated from GFAs or through neighboring GFAs. The Uplink requests are sent only for non cached data query requests with reduced frequency in the proposed scheme which results a reduction in Uplink requests.

- *Impact on Query Response Time-* Figure 5.6 shows the variation of Query Response Time with MQAT of NC, UIR, SAS, CCI, ECCI, and CGCI. All the schemes have higher QRT for smaller values of MQAT. But, it decreases with an increase in MQAT in all the schemes. QRT is very large in NC case as it has no local cache to respond even to frequently accessed pages. It decreases sharply in case of UIR because, with larger query rates, the data in cache becomes invalid and more Uplink requests are sent. In SAS, with such higher query rates, the QRT becomes larger due to longer and historical IRs. In CCI and ECCI, it has reduced and not varied sharply as it has considerations of hot data in cache and cooperation of location agent and HA. But, in CGCI, the variation of QRT is not much as it has cooperation of GFAs, and queries are replied instantly with updated page as data is sent along with IRs.

- *Impact on Uplink requests:* In Figure 5.7, with higher MQAT, data in cache becomes invalid in peer test case techniques. Once any miss occurs for queried data then, in these techniques data is brought only from the server. There is no intermediate component or any cooperation to invalidate the data at lower ends to reduce the Uplink requests. Only ECCI technique has lesser Uplink request as it has GFA with an enhancement of storing the popular hot data in it. So,

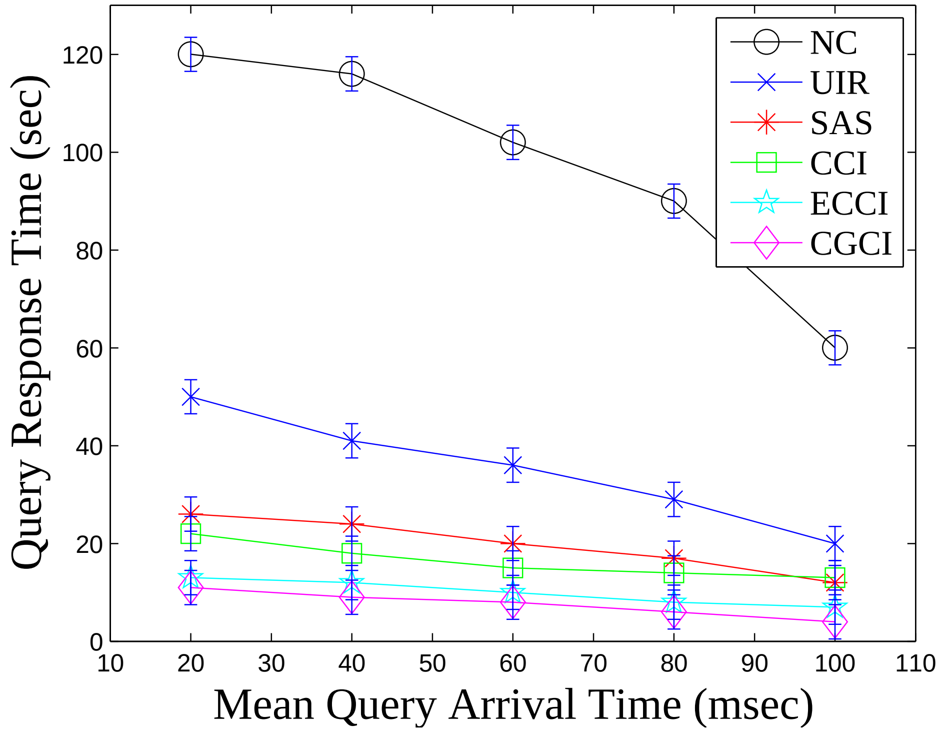

Figure 5.6: Variation of Query Delay with Mean Query Arrival Time

no more Uplink requests are sent on the server to validate a cache data. But, in case of CGCI, as query rate becomes higher, the hot data is considered for caching which has more chances of being cached and due to cooperation among infrastructure components, data is validated from local devices so, lesser Uplink requests are sent to the server in hierarchy.

- *Impact on Hit Ratio:* As shown in Figure 5.8 with smaller MQAT, the hit ratio of all the schemes is smaller. As in UIR, it increases gradually with an increase in MQAT from 20 to 100 milliseconds. In SAS, there is also a gradual increase in hit ratio. But in CCI and ECCI, the hit ratio has raised sharply as it has considered hot data cached in the cache of vehicles, whose updates are stored at corresponding GFAs by HA. But in CGCI technique, hit ratio is good and is does not fall very sharply as neighboring GFAs have cooperation among themselves and data is validated from corresponding GFA after checking the meta data of $Valid\_Unified\_List$ of GFAs.

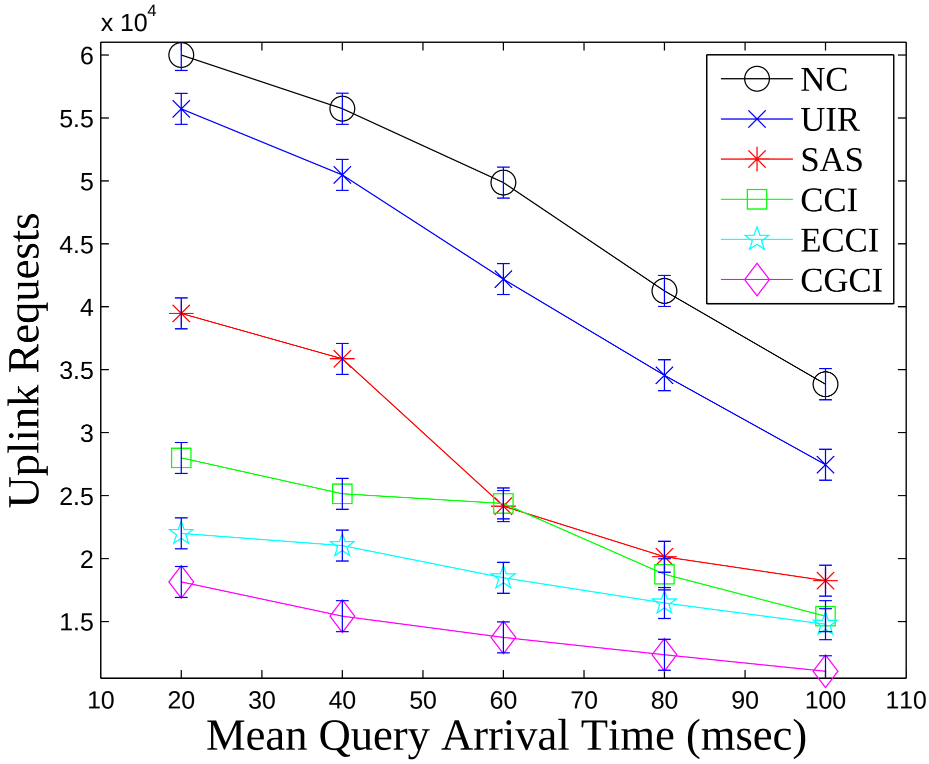

Figure 5.7: Variation of no of Uplink Requests with Mean Query Arrival Time

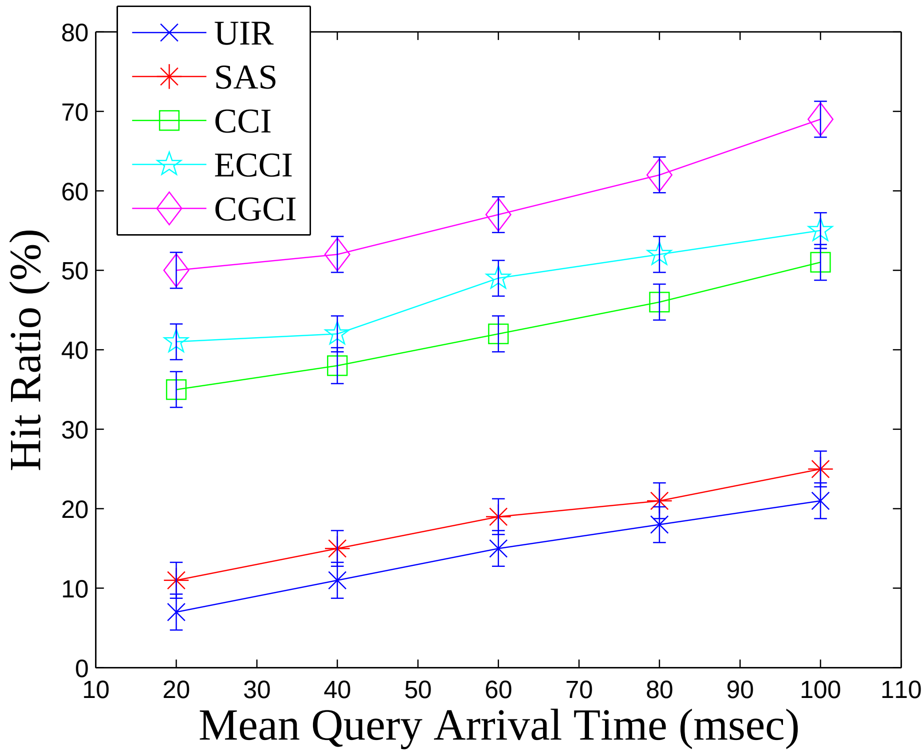

Figure 5.8: Hit Ratio (%) with Mean Query Arrival Time

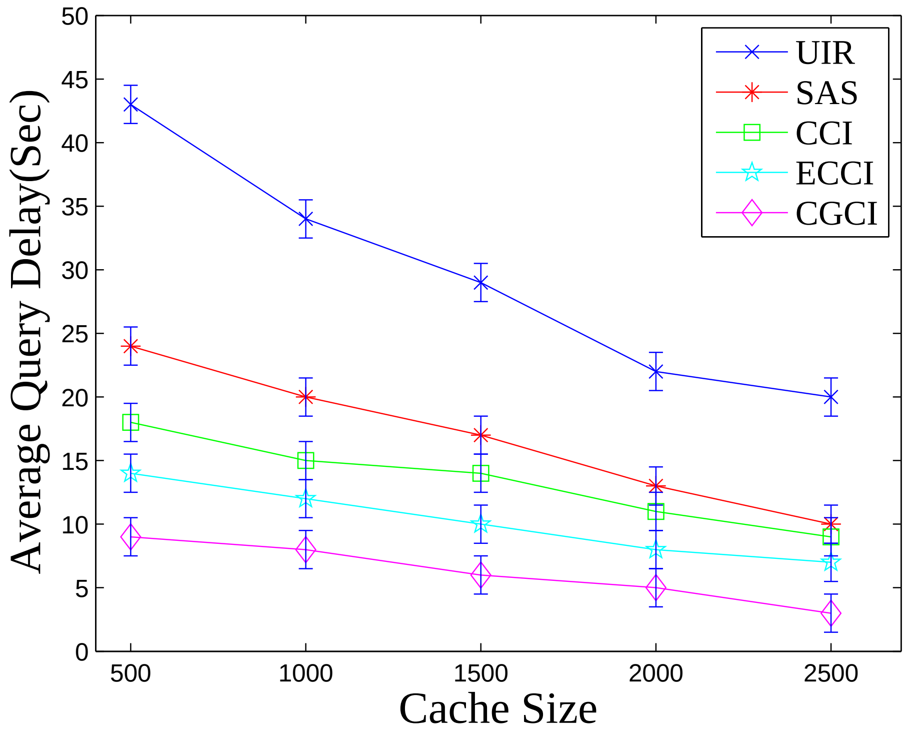

Figure 5.9: Average query delay with variation in Cache Size

5.7.4 Impact of cache size on average query delay, Uplink requests, and Hit ratio

Figure 5.9 shows the average query delay, Uplink requests, and hit ratio by varying the cache size. As shown in the Figure 5.9, with an increase in the cache size, there is a decrease in the average query delay in all the schemes. But, the decrease in the average query delay is more in the proposed scheme as compared to the other schemes. This is due to the fact that proposed scheme used a cooperative approach from the nearest locations. So, in case of emergency situations, contents can be retrieved from the nearest locations as per the demand. This reduces the average query delay in the proposed scheme. None of the other existing schemes have used the cooperation from the nearest locations which causes a longer delay in accessing the contents from a particular location.

Figure 5.10 shows the variation of Uplink request with a variation in the cache size. With an increase in the cache size, the Uplink requests are decreased but the proposed gateway cooperative scheme has the least Uplink requests as compared to the other schemes of its category. This is mainly due to the reason that there is a

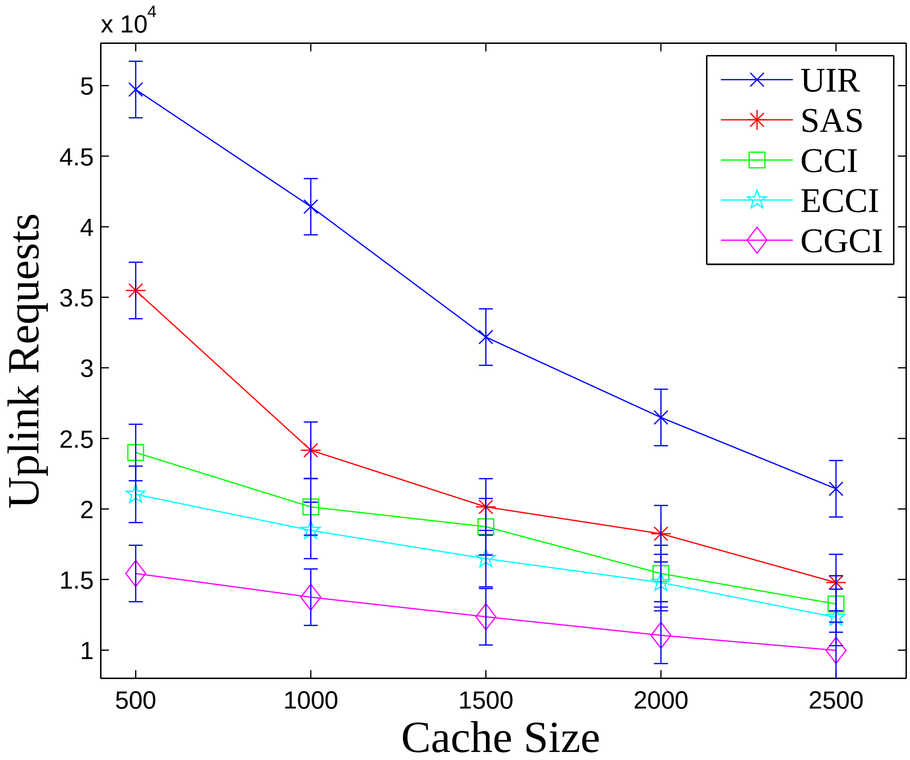

Figure 5.10: Uplink requests with variation in cache size

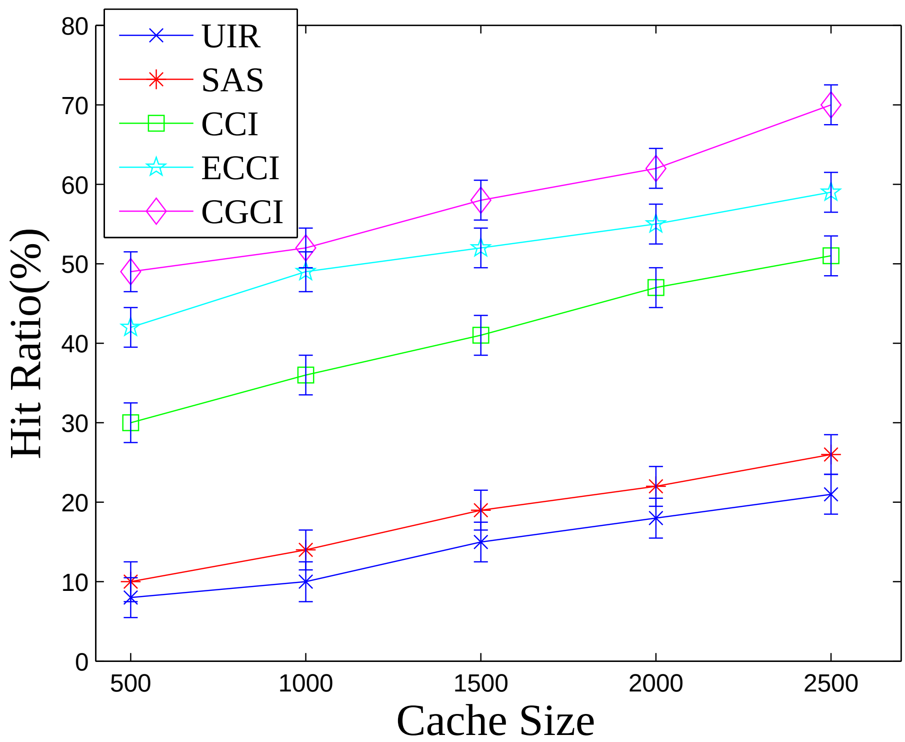

Figure 5.11: Hit ratio with variation in cache size

Table 5.2: Features comparison Table of various schemes with CGCI

Features	CGCI	UIR [24]	SAS [28]	CCI [41]	ECCI [41]
Coopartion	✔	✗	✗	✔	✔
GFA Cooperation	✔	✗	✗	✗	✗
Faster Query Response	✔	✔	✗	✗	✔
Higher Hit Ratio	✔	✗	✗	✗	✗

✔Represents Scheme has incorporated this feature.
✗Represents Scheme has not incorporated this feature.

cooperation among the nodes while sharing the content for caching in the proposed scheme which is missed from most of the existing schemes in the literature. This results an improved performance of the proposed scheme as compared to the other existing schemes.

Figure 5.11 shows the variation in hit ratio with varying cache size. As shown by the results obtained, with an increase in the cache size, cache hit ratio is highest in the proposed scheme as compared to the other schemes of its category. This is mainly due to the reason that the proposed scheme is cooperative with support from the nearest gateways which results an increase in the cache hit ratio with a decrease in broadcast time and Uplink requests. None of the other existing schemes in literature have used the cooperation from the gateway for sharing the contents among themselves. Moreover, in the proposed scheme using the cooperation from the gateway cache invalidation is done periodically and the notification of the same is brodcasted which makes the proposed scheme effective as compared to the other existing schemes. Hence, there is an improvement in the cache hit ratio in the proposed scheme as compared to the the other existing schemes in literature. We have considered approaches for comparison which are much closer to the proposed CGCI scheme, i.e., the behavior of the selected schemes is almost similar to the proposed scheme.

We have considered all test cases for caching in the proposed CGCI scheme. Then, we have analyzed an improvement with respect to various parameters due to which the proposed technique has an upper edge from existing techniques. Table 5.2 clearly shows that the proposed scheme has incorporated more features in it, and is better than the other schemes such as UIR, SAS, CCI, and ECCI.

5.8 Conclusion

In this chapter, we have proposed a new cooperative GFA based CIT called *CGCI for IVANET Environment* that overcomes the higher query response time and more number of Uplink requests problems found in the existing state-of-the-art schemes in literature. The proposed strategy included stateful server thats keeps track of updates of data and generates IR which is broadcasted to HA which in turn, unicast the updated data to GFA and hence to vehicles. The proposed scheme has taken benefit of cooperation of neighboring GFAs which store the hot data fetched from server on any query reply. The proposed scheme works faster with cooperation of GFA, so queries are replied faster from cache or GFA which reduces the number of Uplink requests. To evaluate the performance of CGCI, exhaustive simulations are performed by varying the parameters such as-update rate query response time, Uplink requests hit ratio, and broadcast time. Results obtained show the effectiveness of proposed CGCI scheme over other state-of-the-art schemes such as-UIR, SAS,CCI and ECCI.

Chapter 6

Conclusion & Future Work

In Wireless network, MTs are connected via unreliable low-bandwidth communication channels with the server having frequent disconnections due to power saving strategies and mobility. These MTs have limited battery power. Due to mobility, dynamic topology, and unreliable low-bandwidth communication channels, MTs have frequent disconnections. These factors affect the overall performnace of the networks. Because of above discussed problems, conventional techniques of data access may not be suitable for wireless environments. So, caching of frequently accessed data items on the client side is an effective way to reduce network traffic and query latency. But, data consistency maintenance between clients and servers is a major issue so that clients need not have to answer the queries by out-of-date cached data items updated by servers. Maintaining cache consistency implies that when data items get changed on the server then the changes must also be reflected in the cache of the MTs using IR reports sent by server after a fixed BT. With higher update rate, size of IR becomes large and can congest the network. But, larger size IRs are not suitable for broadcasting. In literature, existing techniques have fixed BTs, due to which at higher update rate, IR size becomes unmanageable. So there is a need of variable BT for CITs. Also in CITs, MTs have frequent disconnections with server due to limited battery life. So, CITs should provide higher disconnection time for MTs. Such dynamic changes are required for an efficient CIT technique. In this thesis, we have designed dynamic cache invalidation techniques in wireless environments. We have considered two wireless environments.

First is infrastructure based wireless environment in which new caching technique is designed for variable size of IRs and BT. Only updates of data is not considered for including it in IR, but hotness is computed for updated data items and hot updated data is recorded in IR. Then, smaller IR is broadcasted. The proposed technique performs well in wireless environment. Also, proposed technique provides larger disconnection time to MTs, which allows MTs for power saving strategies. Proposed technique for cache management is named as *"Adaptive Cache Invalidation Technique(ACIT) for wireless Environment"* which overcomes the higher query response time and more number of Uplink requests problems which are found in the existing state-of-the-art schemes in literature. The proposed technique includes only hot data updates in IR to reduce the size of IR which helps in better utilization of bandwidth and number of hits. The proposed ACIT has historical data with timestamps of w BT intervals for disconnected MT. So, MTs can sleep or disconnect to save power consumption in the proposed scheme. The proposed scheme works in fast mode with updated IR and broadcast time when update rate is higher than a predefined threshold value which results to have fast query replies. Due to inclusion of hot data in IR, cache contains only hot data content so, queries are replied from cache which reduces the number of Uplink requests. To evaluate the performance of ACIT, exhaustive simulations are performed by varying the parameters such as- update rate and query arrival rate and observed their impact on AQRT, IR report size, Uplink requests and broadcast time. Results obtained show the effectiveness of proposed ACIT scheme over other state-of-the-art schemes such as UIR and SAS.

For second wireless environment, we have tested the proposed technique in IVANET environment. In this technique, GFAs work in cooperation with neighboring GFAs, which provides higher hit ratio for query replies. So, better hit ratio is recorded. In this work, a new cooperative GFA Cache Invalidation scheme called as *"Cooperative Gateway Cache Invalidation Technique (CGCI)" for IVANET Environment* is proposed, which overcomes the higher query response time and more number of Uplink requests problems found in the existing state-of-the-art schemes in literature. The proposed strategy includes stateful server, to keep track of updates of data. It generates IR which is broadcasted to HA which in turn, unicast the updated data to GFA to vehicles. This scheme has taken benefit of cooperation of neighboring GFAs which store the hot data fetched from server on any query reply. The proposed scheme works faster with cooperation of GFA so, queries are replied faster from cache or GFA which reduces the number of Uplink requests. To evaluate the performance of CGCI, exhaustive simulations are performed by varying the parameters such as-update rate query response time, Uplink requests hit ratio, and broadcast time. Results obtained show the effectiveness of proposed CGCI scheme over other state-of-the-art schemes such as-UIR, SAS,CCI and ECCI.

In the future, we will test the performance of the proposed scheme in environments where mobility rate is higher. Also, security aspects of the proposed scheme would be evaluated with respect to different network threats presented in the network. In the future, we will also test the performance of the proposed scheme in environments where mobility rate is higher than 30km/hr for urban areas. Moreover, P2P cooperation of the proposed scheme would also be evaluated with respect to different network parameters. Proposed schemes would be evaluated and tested for bursty traffics and would be modeled as hyper-exponential distribution.

Bibliography

[1] D. Bertsekas, R. Galager, “Data Networks”, Prentice Hall, New Jersey, Edition 2nd, Prentice Hall, ISBN: 0132009161, 1992

[2] C. K. Toh, “Ad Hoc Mobile Wireless Networks”, Protocols and Systems, Prentice Hall, PTR, ISBN: 0130078174, 2002.

[3] P. Krishnamurthy, K. Pahlavan, “Principles of Wireless Networks: A Unified Approach”, Pearson with Prentice Hall Publication, edition 3rd, ISBN 0-13- 093003-2, 2001.

[4] Y. Li, I. Chen, ”Adaptive Per-User Per-Object Cache Consistency Management For Mobile Data Access In Wireless Mesh Networks”, Journal of Parallel and Distributed Computing, vol. 71, no. 2, pp. 1034-1046, 2011.

[5] M. Newman , ”The Structure and Function of Complex Networks”, SIAM Review, vol. 45, no. 1,pp. 167-256, 2003.

[6] C. Sutton, ”Internet Began 35 Years Ago at UCLA with First Message Ever Sent Between Two Computers”. UCLA, 2008.

[7] N. Schacham and J. Westcott, ”Future directions in packet radio architectures and protocols”, In proceedings of the IEEE, vol. 75, no. 1, pp. 83-99, 1987.

[8] C. Diacui ,M. Berkenbroc, ”Supporting Cache Coherence In Mobile Cooperative System”, In proceeding of Seventh IEEE International Symposium on Network Computing and Applications, pp. 240-243, 2008.

[9] J. P. Chuang and Y. Chiu, ”Constructing Efficient Cache Invalidation Schemes for Mobile Environments”, In Proceeding of Third International IEEE Conference on Signal-Image Technology and Internet Based System”, pp.281-288, 2008.

[10] M. Artimy, W. Robertson and W. Phillips, “Connectivity In Inter-Vehicle Ad Hoc Networks”, Engineering Canadian Conference on Electrical and Computer, vol. 1, pp. 293-298, 2004.

[11] J. Blum, A. J. Eskandarian, and L. J. Hoffman, “Challenges Of Inter-Vehicle Ad Hoc Networks”, IEEE Transactions Intelligent Transportation Systems, vol. 5, no. 4, pp. 347–351, 2004.

[12] S. Lim, S. Chae, C. Yu, and C.R. Das, “Cooperative Cache Invalidation Strategies For Internet-Based Vehicular Ad Hoc Networks,” In proceeding of IEEE International Conference On Mobile Ad Hoc and Sensor Systems,pp. 712–717, 2008.

[13] V. Kumar and N. Chand, “Data Scheduling in VANETs: A Review”, International Journal of Computer Science and Communication, vol. 1, no. 2, pp. 399-403, 2010.

[14] B.Ram, N. Chauhan, N. Chand, L.K. Awasthi, “A New Mechanism To Query Latency Minimization For Cache Invalidation In Vehicular Adhoc Networks”, In proceeding of 3rd International Conference on Electronics Computer Technology (ICECT), vol.4, pp. 250-253, 2011.

[15] Z. Wang, S. Das, H. Che, & M. Kumar,” A Scalable Asynchronous Cache Consistency Scheme(SACCS) For Mobile Environments”, In Proceeding of IEEE Transactions Parallel & Distributed Systems, vol. 15, no. 11, pp. 983-995, 2004.

[16] S. Khurana, A. Kahol, S. Gupta and P. Srimani, ”A Strategy To Manage Cache Consistency In A Distributed Mobile Wireless Environment”,In Proceeding of IEEE International Conference On Distributed Computing Systems (ICDCS),pp. 530-537, 2000.

[17] W.He, I. Chen and B. Gu, ”A Proxy-Based Integrated Cache Consistency And Mobility Management Scheme For Mobile IP Systems”, Advanced Networking and Applications, pp. 354–361, 2007.

[18] J. Jing, A. Elmagarmid, A. Helal, and A. Alonso,” Bit Sequences: An Adaptive Cache Invalidation Method in Mobile/Server Environments”, Mobile Networks and Applications, vol. 2, no. 2,pp. 115-127, 1997.

[19] J. Cai, K. L. Tan, "Energy Efficient Selective Cache Invalidation", Wireless Networks, vol.5, no. 6, pp. 489-502, 1999.

[20] K. Tan and A. Jik, "Hybrid Cache Invalidation Scheme For Wireless Environments", In Proceedings of IEEE, pp. 287-294, 1999

[21] S. Lim, S. Chae, C. Yu and C.R. Das, "Cooperative Cache Invalidation Strategies For Internet-Based Vehicular Ad Hoc Networks",In Proceeding of 12th International Conference On Computer Communications and Networks (ICCCN),pp. 1-6, 2009.

[22] Q. Hu, D. Lee, "Cache Algorithms Based On Adaptive Invalidation Reports For Mobile Environments", Cluster Computing, vol. 1, pp. 39–50, 1998.

[23] D. Barbara ,T. Imielinksi, "Sleepers And Workaholics: Caching Strategies For Mobile Environments",In proceeding of ACM SIGMOD, pp. 1-12, 1994.

[24] G. Cao, "A Scalable Low-Latency Cache Invalidation Strategy For Mobile Environments", In Proceeding of ACM MOBICOM pp. 200-209, 2000.

[25] R. Frank, E. Giordano,P. Cataldi and M. Gerla, "TrafRoute: A Different Approach to Invalidation in Vehicular Networks", In Proceeding of VECON, pp. 171-185, 2010.

[26] C.H. Lee and K.J. Lee, "An Efficient Cache Invalidation Scheme to Support Strong Data Consistency in Mobile Ad-Hoc Network", In Proceeding of 8th IEEE International Symposium on Communication Systems, Networks and Digital Signal Processing, pp. 1-6, 2012.

[27] T. Hara, and S. Madria, "Data Replication For Improving Data Accessibility In Ad Hoc Networks", In Proceedings of IEEE Transactions on Mobile Computing, vol. 5, no. 11, pp. 1515-1532, 2006.

[28] H. Safa, H. Aartail and M. Nahhas, "A Cache Invalidation Strategy for Mobile Networks", Journal of Network and Computer Applications.vol. 33, no. 2,pp. 168-182, 2010.

[29] H. Safa and H. Aartail, ”COACS: A Cooperative and Adaptive Caching System for MANETs”, IEEE Transactions on Mobile Computing, vol. 7, no. 8, pp. 951-977, 2008.

[30] G. Cao and C. Dar, ”On the Effectiveness of a Counter-Based Cache Invalidation Scheme and its Resiliency to Failures in Mobile Environments”, In Proceeding of 20th IEEE Symposium on Reliable Distributed System (SRDS), pp. 247-256, 2001.

[31] G. Cao, ”On Improving the Performance of Cache Invalidation in Mobile Environments”, Mobile Networks and Applications (MONET), vol. 7, no.4, pp. 291-303, 2002.

[32] C.W. Chao , F.J. Fang and H. P. Chun, ”A Counter-Based Cache Invalidation Scheme for Mobile Environment with Stateless Servers”,In Proceeding of IEEE Pacific Rim Conference on Communications, Computers and signal Processing PACRIM , 2003.

[33] G. Cao, “A Scalable Low Latency Cache Invalidation Strategy For Mobile Environments”, In Proceeding of IEEE Transactions On Knowledge And Data Engineering, vol 15, no. 5, pp. 1251-1265, 2003.

[34] N. Chand ,R.C. Joshi,and M. Misra, ”Energy Efficient Invalidation in Mobile Environment”, Journal of Digital Information Management, vol 3, no. 2, pp. 119-125, 2005.

[35] M.S. Joseph , M.Kumar, H. Shen, and S.K. Das, ”Energy Efficient Data Retrieval & Caching in Mobile P2P Network”, In Proceeding of International Conference on Pervasive Computing & Communications workshop, pp. 50-54, 2005.

[36] C. Chin, J. Fie and P.Chun , ”A counter Based Cache Invalidation Scheme for Mobile Environments with Stateless Servers” Communications, Computer & Signal Processing, vol. 2, pp. 623-626, 2009.

[37] C.C. Wu, J. Fang and P.C.Hung, ”A Counter-Based Cache Invalidation Scheme for Mobile Environments with Stateless Servers”, In Proceeding of IEEE Pacific Rim Conference on Communications, Computers and Signal Processing, pp 623-626, 2003.

[38] P.Chuang and Y. Chiu, "Efficient Cache Invalidation Schemes for Mobile Data Accesses", Information Science, pp. 5084-5101, 2011.

[39] N. Fatima,P. Khader, "Enhanced Adaptive Cache Invalidation Approach for Mobile Environments", In Proceeding of 3rd IEEE international conference on Electronic Computer Technology (ICECT)", pp. 76-80, 2011.

[40] S. Lim, S. Lee, M. Soha, and B. Lee, "Energy-aware optimal cache consistency level for mobile devices", Information Sciences. , vol. 230, no. 5, pp. 94-105, 2013.

[41] L. Sunho, Y. Chansu, C.R. Das, "Cache Invalidation Strategies For Internet-Based Vehicular Ad Hoc Networks", Computer Communications. vol. 35, pp. 380-391, 2012.

[42] Y. Bao and R. Alhajj "Hybrid Cache Invalidation Schemes in Mobile Environments", In Proceeding of International Conference on Pervasive Services, pp. 209-218, 2004.

[43] T. Cai and H. Ooi, "An Evaluation Of Cache Invalidation Strategies In Wireless Environments", In Proceeding of IEEE Transactions On Parallel & Distributed Systems,vol. 12, no.8, pp. 789-807, 2001.

[44] Y. Xu, Y. Li, T. Lin, Z. Wang, W. Niu,H. Tang and S. Ci, "A Novel Cache Size Optimization Scheme Based On Manifold Learning In Content Centric Networking" , Journal of Network and Computer Applications, vol. 37, pp. 273-281, 2014.

[45] Y. Chang, I.Tin and T. Lin, "Dynamic Cache Invalidation Scheme in IR-based Wireless Environments", In proceeding of 22nd International Conference on Advance Information Networking and Applications IEEE Computer Society, pp. 669-704, 2008.

[46] R. Tiwari, N. Kumar, "Dynamic Web Caching: For Robustness, Low Latency & Disconnection Handling", In Proceeding of 2nd IEEE International Conference on Parallel, Distributed and Grid Computing,pp. 909-914, 2012.

[47] R. Tiwari, N. Kumar,”A Novel Hybrid Approach for Web Caching”, In Proceeding of Sixth International conference on Innovative Mobile and Internet Services in Ubiquitous Computing, pp. 512-517, 2012.

[48] Y.H. Kharbade, R.N. Chopde, “Review Study on Distributed Cache Invalidation in Wireless Mobile Networks”, International Journal of Emerging Research in Management & Technology, vol. 2, no.4, pp. 137-154, 2013.

[49] R. Tiwari,G. Khan, ”Load Balancing Through Distributed Web Caching With Clusters”, In Proceeding of Communication systems and Network Applications 2010, pp. 46-54, 2010.

[50] T. Nguyen, T. Dong, “An Adaptive Cache Consistency Strategy in a Disconnected Mobile Wireless Networks” , In Proceeding of International Conference on Computer Science and Automation Engineering (CSAE), pp. 1-3 , 2011.

[51] J. Chun, E. Chan and K. Lam, “An Adaptive AVI-Based Cache Invalidation Scheme for Mobile Computing Systems”, International Workshop on Mobility in Databases and Distributed Systems, pp. 55-159, 2000.

[52] J. Ravi “A Survey On Dynamic Web Content Generation And Delivery Techniques”, Journal of Network and Computer Application, vol. 32, no. 5, pp. 943-960, 2009.

[53] J. Famaey, “Towards A Predictive Cache Replacement Strategy For Multimedia Content”, Journal of Network and Computer Application, vol. 36, no.1, 219-227, 2013.

[54] K. L. Wu, P.S. Yu and M.S. Chen, ”Energy-efficient caching for wireless mobile computing”, In Proceeding of 20th International Conference on Data Engineering, pp. 336–345, 1996.

[55] Q. Hu, D.L. Lee ”Cache algorithms based on adaptive invalidation reports for mobile environments”, Cluster Computing , vol. 1, no. 1, pp.39–50, 1998.

[56] M.K. Denko and J. Tian ”Layer Design for Cooperative Caching in Mobile ad hoc network”, In Proceedings of IEEE Conference on Consumer Communications and Networking, pp. 375-380, 2008.

[57] N. Kumar and H. J. Lee, "Peer-to-Peer Cooperative Caching for Data Dissemination in Urban Vehicular Communications", In Proceeding of IEEE Systems Journal. 2014.

[58] N. Kumar, S.Zeadally and J.P.C. Rodrigues,"QoS-Aware Hierarchical Web Caching Scheme for Online Video Streaming Applications in Internet-Based Vehicular Ad Hoc Networks",IEEE Transactions on Industrial Electronics,vol.99, 2015.

[59] P. Parvathy and A. Kumar, "2TierCoCS: A two-tier cooperative caching scheme for Internet-based vehicular ad hoc networks" ,In Procedia Computer Science- International Conference on Information and Communication Technologies (ICICT), pp. 1079 – 1086, 2015.

[60] R. Tiwari, N. Kumar "A Novel Hybrid Approach for Web Caching", In Proceeding of IMIS, pp. 512-517, 2012.

[61] Safa H, Artail H,Nahhas, M., "Enhancing Cache Invalidation In Mobile Environments",ACM International Conference On Mobile Technology, Applications And Systems (Mobility'08), 1–8, 2008

[62] P. Paul and N. Saravanan, "Efficient Service Cache Management In Mobile P2P Networks", Future Generation Computer Systems, vol.29, no. 6, pp. 1505-1521, 2013.

[63] T. Nguyen, T. Dong, "An Efficient Cache Invalidation System in Mobile Information Systems", In Proceeding of International Conference On Computing And Communication Technology, Research, Innovation, And Vision For Future, pp.1-4, 2010.

[64] Xia F., Ahmed A.M., Yang L.T., Ma J., Rodrigues J., "Exploiting Social Relationship to Enable Efficient Replica Allocation in Ad-hoc Social Networks", IEEE Transactions on Parallel and Distributed Systems, vol. 25, no. 12,pp. 3167-3176, 2014.

[65] E. Chan, W. Li and D. Chen, "Energy Saving Strategies For Cooperative Cache Replacement In Mobile Ad Hoc Networks", Pervasive and Mobile Computing, vol. 5, no. 1 , pp. 77-92, 2009.

[66] P. Paul, N. Saravanan, ”Efficient Service Cache Management In Mobile P2P Networks”, Future Generation Computer Systems, vol 23, no.6, pp. 1505-1521, 2013.

[67] P. Paul, N. Saravanan, R. Baskaran and P. Dhavachelvan, ”Efficient Service Cache Management in Mobile P2P Networks”, Future Generation Computer Systems, vol. 29, no. 6, pp. 1505-1521, 2013.

[68] E. Chan, W. Li and D. Chen, ”Energy Saving Strategies For Cooperative Cache Replacement In Mobile Ad Hoc Networks”, Pervasive and Mobile Computing, vol. 5, no. 1, pp. 77-92, 2009.

[69] Sharma H., Bansal J.C. and Arya K.V., ” Power law-based local search in artificial bee colony”, International Journal of Artificial Intelligence and Soft Computing,vol. 4, pp. 164-194, 2014.

[70] C. Naveen, L.K. Avasthi ”Pre-fetching based cooperative caching in mobile adhoc and emerging trends in networks”, In Proceeding of International conference on emerging trend in computer and electronics engineering, pp. 60-64.

[71] G. Cao and L.C.Das, ”Cooperative Cache Based Data Access in Ad-Hoc Networks”, Computer, vol. 37, no. 2, pp. 32-39, 2004.

[72] H. Shen H, M.S. Joseph,M. Kumar and S. Das, ”A Scheme for Cooperative Caching in Mobile Peer to Peer Network”, In Proceeding of International Parllel & Distributed Processing Symposium, pp. 57-64, 2005.

[73] E. Chan, L. Liw, ”Movement Prediction Based Cooperative Caching For Location Dependent Information Service In Mobile Adhoc Network”, Journal of Supercomputing”, vol. 59, no. 1, pp. 297-322, 2012.

[74] L. Yin and G. Cao, ”Supporting Cooperative Caching In Ad Hoc Networks”, In Proceeding of IEEE Transaction on Mobile Computing, vol. 5, no. 1, pp. 77-89, 2006.

[75] P. Kumar, N. Chauhan, L. Awasthi and N. Chand, ”Proactive Approach for Cooperative Caching in Mobile Adhoc Networks”, International Journal of Computer Science, vol 7, no. 8, pp 21-27, 2010.

[76] S. Lim, Y. Lee and M. Min, "ConSens: Consistency-Sensitive Opportunistic Data Access in Wireless Networks", Wireless Network, vol. 2, pp. 804-809, 2011.

[77] S. Lim, C. Yu and C. Das, "Cooperative Cache Invalidation Strategies for Internet-based Vehicular Ad Hoc Networks", In Proceeding of 12th International Conference on Computer Communications and Networks (ICCCN), pp. 1-6, 2009.

[78] F. Marmol, G. Perez, "TRIP: A Trust And Reputation Infrastructure-Based Proposal For Vehicular Ad Hoc Networks", Journal of Network and Computer Application, vol. 35, no. 3, pp. 934–941, 2012.

[79] A. Dubey, S. Sharma, "A Cache Invalidation Scheme Through Data Classification in IVANET", International Journal of Computer Application, vol. 25, no. 9, 2011.

[80] P. T. Joy and K. P. Jacob, "Cooperative Caching Techniques for Mobile Ad hoc Networks". In Proceeding of IEEE, pp. 175-180, 2012.

[81] J. Xu, X. Tang and D.L.Lee, "Performance analysis of location-dependent cache invalidation schemes for mobile environments", IEEE Transaction of Knowledge Data Engineering, vol 15, no. 2, pp. 474–488, 2003.

[82] B. Zheng , J. Xu and D.L. Lee, "Cache invalidation and replacement strategies for location-dependent data in mobile environments", IEEE Transaction on Computing, vol 51, no. 10, 2002.

[83] X. Wang and P. Fan, "A strongly consistent cached data access algorithm for wireless data networks", Wireless Network, vol. 15, no. 8, 2009.

[84] Y. Huang,J. Cao and B. Jin, "A predictive approach to achieving consistency in cooperative caching in Manet", In Proceedings of the 1st International Conference On Scalable Information Systems, 2006.

[85] Y. Huang, J. Cao, Z. Wang, B. Jin and Feng Y, "Achieving flexible cache consistency for pervasive Internet access",In proceeding of fifth annual IEEE international conference on pervasive computing and communications, pp 239–250,2007.

[86] Mittal R., Das M.L., "Secure Node Localization in Mobile Sensor Networks", International Journal of Wireless Networks and Broadband Technologies, Vol. 3, no. 1, pp. 18-33.

[87] W. Li, E. Chan, D. Chen and S. Lu, "Maintaining probabilistic consistency for frequently off-line devices in mobile ad hoc networks", In Proceeding of 29th international conference on distributed computing systems, 2009.

[88] S. Lim and Y.Lee, "User-defined consistency sensitive cache invalidation strategies for wireless data access", Computer Communications, vol. 41, pp. 55-66,2014.

[89] K. Marshad and H. Artail, "SSUM: Smart Server Update Mechanism for Maintaining Cache Consistency in Mobile Environments", In Proceeding of IEEE Transaction on Mobile Computing, vol. 9, no. 6, pp. 778-795, 2010.

[90] Deng D.J., Cheng R.S., Chang H.J., Lin H.T. and Chang R.S.,"A cross-layer congestion and contention window control scheme for TCP performance improvement in wireless LANs", Telecommunication Systems, vol. 42, no. 1-2, pp 17-27, 2009.

[91] Gadiraju S., Kumar V. and Dunham M., "Recovery in the Mobile Wireless Environment Using Mobile Agents", IEEE Transactions on Mobile Computing, Vol. 3, No. 2, April-June 2004, pp: 180-191.

[92] R Singh, M Dave, " Antecedence graph approach to check pointing for fault tolerance in mobile agent systems", IEEE Transactions on Computers, vol. 62, no. 2, 247-258, 2013.

[93] Li. Wenzhong and E. Chan, "Performance analysis of cache consistency strategies for multi-hop wireless networks" Journal of Supercomputing,Vol. 62, pp. 1065–1090, 2012.

[94] K. Sravya1 and K. Bhawna,"Performance Analysis of Distributed Cache Invalidation Method in Mobile Ad hoc Networks",International Journal on Recent and Innovation Trends in Computing and Communication, vol. 2, no. 10, pp. 3045-3049, 2014.

[95] S. Lim,W.C. Lee, G. Cao and C. Das, "Performance Comparison Of Cache Invalidation Strategies For Internet Based Mobile Adhoc Netwroks",In Proceeding of Internet Conference On Mobile Adhoc & Sensor Systems, pp. 104-113, 2004.

[96] K. Fawaz, H. Artail, "DCIM: Distributed Cache Invalidation Method for Maintaining Cache Consistency in Wireless Mobile Network", In Proceeding of IEEE Transactions on Mobile Computing, vol. 12, no. 4, pp. 680-693, 2013.

[97] A. Madhukar, T. Ozyer and R. Alhajj, "Dynamic Cache Invalidation Scheme For Wireless Mobile Environments, Wireless Networks , vol. 15, no. 6, pp. 727-740, 2009.

[98] A. Madhukar and R. Alhaji, "An Adaptive Energy Efficient Cache Invalidation Scheme For Mobile DataBases", In Proceeding of Symposium on Applied Computing, pp. 1122-1126, 2006.

[99] N. Chand, R.C. Joshi and M. Mishra, "Broadcast Based Cache Invalidation & Prefetching In Mobile Environment", In Proceeding of International Conference On High Performance Computing (HiPC), pp. 410-419, 2004.

[100] M.K. Yeung, Y.K. Kwok, " Wireless Cache Invalidation Schemes With Link Adaptation And Down Link Traffic", IEEE Transaction on Mobile Computing, vol. 4, no 1,pp. 68-83, 2005.

[101] NS2 simulator,http://www.insi.edu/nsnam/ns (2008).

[102] Issariyakul T., Hossain E. Introduction to Network Simulator NS2,Springer.

[103] Kevin Fall, Kannan Varadhan, "The NS Manual", http://www.isi.edu/nsnam/ns/ns-documentation.html, March 9, 2006

[104] T. Ernst and H-Y. Lach, "Network Mobility Support Terminology", draftietf-nemo-terminology-04.txt, 2005.

[105] T. Ernst, "MobiWan: A NS-2.1b6 simulation platform for Mobile IPv6 in Wide Area Networks", http://www.inrialpes.fr/planete/mobiwan/, 2001.

[106] R. Kong, H. Zhou, "Extensions to Mobiwan according to RFC 3775 based on NS2",In Proceeding of International Symposium on Computer Science and Technology, pp. 998-1001, 2007.

[107] D.L. Oliva, C. J. Bernardos and M. Calderon, ”Practical evaluation of a network mobility solution”, In Proceeding of 11th Open Eur. Summer School: Networked Applications, pp. 60-66, 2005.

[108] M. Calderon, J. C. Bernardos, M. Bagnulo, I. Soto, and A.D.L. Oliva, ”Design and Experimental Evaluation of a Route Optimization Solution for NEMO”, IEEE Journal on Selected Areas in Communications, vol. 24, no. 9, pp.1702-1716 , 2006.

[109] Y. Wang, F. Li, ” Vehicular ad hoc networks”, Chapter 20. 2005; 503-525.

[110] N. Ahmadifard, H. Nabizadeh,and M. Abbaspour, ”ISEFF: An ID-Based Scalable and Efficient Distributed File Sharing Technique in Vehicular Ad Hoc Networks” Wireless Personal Communications,vol 75, no. 2,pp. 821-841, 2014.

[111] S. Lim, W. Lee, G. Cao, C.R. Das, ”A novel caching scheme for improving Internet based mobile ad hoc networks performance” Ad Hoc Networks, vol. 4, no. 2, pp. 225-239, 2006.

[112] N. Kumar, J.H. Lee, ”Peer-to-Peer Cooperative Caching for Data Dissemination in Urban Vehicular Communications”, IEEE Systems Journal, vol. 8, no. 4, pp. 1136-1144, 2014.

[113] N. Kumar, S. Misra, M.S. Obaidat, ”Collaborative Learning Automata-Based Routing for Rescue Operations in Dense Urban Regions Using Vehicular Sensor Networks” IEEE Systems Journal, vol. 9, no. 3, pp. 1081 - 1090, 2015.

[114] N. Kumar, J.H. Lee, J.P.C. Rodrigues, ”Intelligent Mobile Video Surveillance System as a Bayesian Coalition Game in Vehicular Sensor Networks: Learning Automata Approach”, IEEE Transactions on Intelligent Transportation Systems, vol. 16, no. 2, pp. 1148 - 1161,2015

[115] N. Kumar, N. Chilamkurti, S.C.Misra, ”Bayesian Coalition Game for Internet of Things: An ambient intelligence approach”, IEEE Communication Magazine, vol. 53, no. 1, pp. 48-55, 2015.

[116] N. Kumar, N. Chilamkurti, J.P.C. Rodrigues, ”Learning automata-based opportunistic data aggregation and forwarding scheme for alert generation in vehicular ad hoc networks”, Computer Communications, vol. 39, no. 2, pp. 22-32, 2014.

[117] N. Kumar, S. Misra , J.P.C.Rodrigues and M.S. Obaidat, "Networks of Learning Automata: A Performance Analysis Study", IEEE Wireless Communication Magazine, vol. 21, no. 6, pp. 41-47, 2014.

[118] N. Kumar and C.C. Lin, "Reliable Multicast as a Bayesian Coalition Game for a Non-stationary Environment in Vehicular Ad Hoc Networks: A Learning Automata Based Approach", International Journal of Adhoc and Ubiquitous Computing, vol.19, no.3, pp. 168 - 182, 2015.

[119] N. Kumar, R. Iqbal, S. Misra, J.P.C. Rodrigues, "Bayesian Coalition Game for Contention-Aware Reliable Data Forwarding in Vehicular Mobile Cloud", Future Generation Computer Systems, vol. 48, pp. 60–72, 2015.

[120] N. Kumar, S.Tyagi and D.J. Deng, "LA-EEHSC: Learning automata-based energy efficient heterogeneous selective clustering for wireless sensor networks", Journal of Networks and Computer Applications, vol. 46 , no. 11, pp. 264-279, 2014.

[121] N. Kumar, S. Misra, J.P.C. Rodrigues, "An intelligent approach for building a secure decentralized public key infrastructure in VANET", Journal of Computer and System Sciences, vol. 81, no. 6, pp. 1042–1058, 2015.

[122] N. Kumar, S. Misra, J.P.C. Rodrigues,M.S. Obaidat, "Coalition Games for Spatio-Temporal Big Data in Internet of Vehicles Environment: A Comparative Analysis, IEEE Internet of Things Journal, vol. 2, no. 4, pp. 310-320, 2015.

[123] Iqbal R., Nazaraf S. N., James A.E., Jacob Duursma,"From work practices to redesign for usability", Expert System Appl. , vol. 38, no.2, pp 1182-1192, 2011.

[124] N. Kumar, J.P.C. Rodrigues, N. Chilamkurti, "Bayesian Coalition Game as-a-Service for Content Distribution in Internet of Vehicles", IEEE Internet of Things Journal, vol. 1, no. 6, pp. 544-555, 2014.

[125] R. Bali, N. Kumar and J.P.C.Rodrigues, "Clustering in vehicular ad hoc networks: Taxonomy, challenges and solutions", Vehicular Communications, vol. 1, no. 3, pp. 134-152, 2014.

[126] A. Dua, N. Kumar, S. Bawa, "QoS-Aware Data Dissemination for Dense Urban Regions in Vehicular Ad Hoc Networks", Mobile Networks and Applications, 2014.

[127] N. Kumar,J. Kim, "Probabilistic trust aware data replica placement strategy for online video streaming applications in vehicular delay tolerant networks", Mathematial and Computer Modelling, vol. 58, no. 1/2, pp.3-14, 2014.

[128] A. Dua, N. Kumar, S. Bawa, "A Systematic Review on Routing Protocols for Vehicular Ad Hoc Networks", Vehicular Communications, vol. 1, no. 1, pp. 33-52, 2014.

[129] N. Kumar , N. Chilamkurti, J.P.C. Rodrigues, "Learning automata-based opportunistic data aggregation and forwarding scheme for alert generation in vehicular ad hoc networks", Computer Communications, vol. 59, no. 1, pp. 22-32, 2014.

[130] Q. Yang, A. Lim , S. Li, J. Fang and P. Agrawal, "ACAR Adaptive Connectivity Aware Routing for Vehicular Ad Hoc Networks in City Scenarios", Mobile Networks and Applications, vol 15, no. 1, pp. 36-60, 2010.

[131] C. Diacui, M. Berkenbroc, "Supporting Cache Coherence In Mobile Cooperative System", In Proceeding of Seventh IEEE International Symposium on Network Computing and Applications, pp. 240-243, 2008.

[132] J.P. Chuang, Y. Chiu, "Constructing Efficient Cache Invalidation Schemes for Mobile Environments", In Proceeding of Third International IEEE Conference on Signal-Image Technology and Internet Based System, pp. 281-288, 2008.

[133] G.S. Li , W.L.Wang and X.W. Yao, "An Adaptive and Opportunistic Broadcast Protocol for Vehicular Ad Hoc Networks", International Journal of Automation and Computing, vol. 9, no. 4, pp.378-387, 2012.