1-1-2006

Congestion aware overlay network

Md. Moinuddin Bhuiyan
Ryerson University

Follow this and additional works at: <http://digitalcommons.ryerson.ca/dissertations>

Part of the [Electrical and Computer Engineering Commons](#)

Recommended Citation

Bhuiyan, Md. Moinuddin, "Congestion aware overlay network" (2006). *Theses and dissertations*. Paper 434.

This Thesis is brought to you for free and open access by Digital Commons @ Ryerson. It has been accepted for inclusion in Theses and dissertations by an authorized administrator of Digital Commons @ Ryerson. For more information, please contact bcameron@ryerson.ca.

CONGESTION AWARE OVERLAY NETWORK

By

Md. Moinuddin Bhuiyan

**B.A.Sc. in Electrical & Electronic Engineering
Chittagong University of Engineering & Technology (CUET)
Bangladesh, 1994**

**A thesis
presented to Ryerson University to the
in partial fulfillment of the
requirements for the degree of
Master of Applied Science
in the Program of
Electrical & computer Engineering**

Toronto, Ontario, Canada, 2006

©Md. Moinuddin Bhuiyan 2006

UMI Number: EC53806

INFORMATION TO USERS

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleed-through, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

UMI Microform EC53806
Copyright 2009 by ProQuest LLC
All rights reserved. This microform edition is protected against
unauthorized copying under Title 17, United States Code.

ProQuest LLC
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106-1346

Author's Declaration

I hereby declare that I am the sole author of this thesis.

I authorize Ryerson University to lend this thesis to other institutions or individuals for the purpose of scholarly research.

_____ January 27, 2006

I further authorize Ryerson University to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

_____ January 27, 2006

ABSTRACT

**Md. Moinuddin Bhuiyan, Congestion Aware Overlay Network.
M.A.Sc., Electrical & Computer Engineering, Ryerson University, 2006.**

Advances in wireless technology and mobile computing have provided a major impetus towards development of P2P network over MANET. Both peer-to-peer overlays and mobile ad-hoc network share the key characteristics of self-organizing and decentralization; both need to solve same fundamental problem, that is, how to route queries or packets efficiently. Congestion affecting the performance of P2P overlay on wireless ad-hoc network occurs not only due to buffer overflow but also due to error prone nature of the wireless medium used and the mobility of users. When paths of multiple downloading sessions intersect with each other, they mutually cause congestion for each other that result in packet drops. We identified that P2P MANET suffers query performance primarily due to significant congestion during download session. The unawareness of congestion in P2P MANET system may lead to the longer user response time, high overhead and many packet losses. In order to alleviate the congestion problem, we propose a modification to RAON, CAON, which includes congestion awareness for improving the RAON performance. We evaluated our system through simulation, which shows that CAON improves the performance of query success rate, query delay, packet losses significantly. We also evaluated the impact of mobility model on our system by simulating under both Random way point mobility model and Manhattan mobility model.

ACKNOWLEDGEMENT

I have had a great honour to be supervised by Dr. M. Jaseemuddin. I would like to take this opportunity to acknowledge his scientific guidance and support. I would like to thank Mr. Gabriel Lau, who completed the RAON part and helped me to understand his work in several times. I would like to thank my wife, who constantly supported me and encouraged me through out the course of this thesis. I also thank my son who constantly missed me during my thesis. I would like to thank my family, especially my Mother and father (late), for always being there for me when I need their prayers.

TABLE OF CONTENTS

AUTHOR’S DECLARATION	ii
ABSTRACT	iii
ACKNOWLEDGEMENT	iv
LIST OF FIGURES	viii
LIST OF TABLES...	x
LIST OF ABBREVIATIONS	xi
Chapter – 1:	
Introduction	1
1.1 Why P2P MANET?	2
1.2 Motivation	3
1.3 Thesis Contribution	5
Chapter – 2:	
BACKGROUND	7
2.1. Flow control, Congestion control and congestion avoidance	7
Mechanism	
2.1.1. Comparison between congestion control and congestion avoidance	9
2.1.2. MANET congestion control	9
2.1.3. Congestion in peer to peer network	11
2.2. Mobile Ad hoc network (MANET)	13
2.2.1. Characteristics of MANET	13

2.2.2.	MANET Routing protocol	15
2.2.2.1.	Proactive vs Reactive routing protocols	15
2.2.3.	Ad-hoc on-demand Distance vector (AODV)	16
2.3.	Peer-to-Peer network....	19
2.3.1.	Overlay network ...	19
2.4.	Unstructured peer-to-peer Network	21
2.4.1	Gia Design	21
2.5.	Similarities and dissimilarities between MANET and Peer to Peer network...	25
2.6.	RAON: A P2P Network for MANET....	26
2.6.1.	RAON Design features	27
2.7	Mobility Model for Mobile Ad hoc Network	28
2.8	Summary	31
Chapter 3:		
	CAON DESIGN	33
3.1.	Problem Identification	33
3.1.1.	Analysis of RAON's performance with download traffic?	36
3.1.2.	Our hypothesis	39
3.2.	CAON design	44
3.2.1	Local feedback mechanism	44
3.2.2.	Selection of download site	47
3.3.	Summary	50
CHAPTER 4:		
	PERFORMANCE EVALUATION	51

4.1.	Simulation set-up	51
4.2.	Results analysis:	53
4.2.1.	Query forwarding performance	54
4.2.2.	Network level analysis: AODV route failure performance	63
4.3.	Average number of messages generated in RAON and CAON	65
4.4.	Performance evaluation under Manhattan Mobility Model	65
4.4.1.	Simulation setup	65
4.4.2.	Results analysis	67
4.3.	Summary	72
Chapter-5:		
	Conclusion and Future work	73
	References	76
	Appendixes	81

LIST OF FIGURES

Figure1.1: Mapping the overlay network topology to the network infrastructure	4
Figure 2.1: shows Source and destination	7
Figure 2.2. Multiple Sources and Multiple destinations in a network	8
Figure 2.3. Network performance as a function of load	12
Figure 2.4. A simple Ad hoc network	13
Figure 2.5: AODV protocol Messaging	17
Figure 2.6: Peer to Peer network over MANET	25
Figure 2.7: Random way point mobility model	30
Figure 2.8: Map for Manhattan mobility Model	31
Figure 3.1: Query success rate of RAON	37
Figure: 3.2. Query delay Vs. number of peers	38
Figure: 3.3. Duplicate Acks Vs. number of peers	39
Figure 3.4: Packet losses Vs. simulation time	41
Figure3.5: RTT vs. simulation time	42
Figure 3.6: Average number of unstable links	43
Figure 3.7: An example of query search	45
Figure 4.1 RAON & CAON query forwarding performance	55
Figure 4.2: Query response delay	58
Figure 4.3: Average number of unstable links generates in RAON & CAON	59
Figure 4.4: Packet loss vs. simulation time	61
Figure 4.5: RTT vs. Simulation time	61

Figure 4.6: Total number of duplicate Acks for RAON and CAON	62
Figure 4.7: AODV route failure for RAON and CAON	64
Figure 4.8: RAON and CAON query success rate under Manhattan mobility	68
Figure 4.9: Query delay for RAON & CAON	68
Figure 4.10. Packet losses vs. simulation time	69
Figure 4:11: RTT vs. simulation time	70
Figure 4.12: duplicate Acks Vs. number of peers	70

LIST OF TABLES

Table –2.1: Comparison between peer to peer network and MANET	25
Table 2.2 Conditions for changing a neighbor's color (The symbol && and refers to the logical AND & OR operation respectively)	28
Table 3.1: Query color change & feedback option	47
Table 4.1: Average number of message generate	65
Table 4.2: performance comparison of CAON in both mobility models.	72

LIST OF ABBREVIATION

ACK	Acknowledgement
AODV	Ad-hoc On-demand Distance Vector routing
API	Application programming interface
AS	Autonomous system
CAN	Content Addressable network
CAON	Congestion Aware overlay
CPU	Central processing Unit
DHT	Distributed hash Table
DSDV	Destination-Sequence Distance_vector routing
DSR	Dynamic source Routing
IP	Internet protocol
MAC	Medium Access Control
MH	Manhattan
MANET	Mobile Ad hoc network
NCS	Neighboring Coloring scheme
OS	Operating System
P2P	Peer to Peer
PHY	Physical
RAON	Resource aware overlay
RREQ	route request
RREP	Route reply

RTT	Round trip time
RWP	Random Way point
TTL	Time to live
TCP	Transmission Control protocol
UDP	User datagram protocol
D1	With download
Do	Without download

Introduction

Wireless technology is rapidly evolving, and is playing an increasing role in the lives of people throughout the world. Peer-to-Peer (P2P) applications have become immensely popular in the Internet. Among P2P applications, file sharing is perhaps the most popular application. Compared to traditional client/sever file sharing (such as FTP, WWW), P2P file sharing has one big advantage, namely, scalability. The performance of traditional file sharing applications deteriorates rapidly as the number of clients increases, while in a well-designed P2P file sharing system, more peers means better performance [42]. Both mobile ad hoc network (MANET) and peer-to-peer network file sharing systems lack infrastructure and posses no prior knowledge of arriving and departing nodes. This common nature gives P2P file sharing attractive on MANET. We define the P2P overlay network as a virtual network constructed such that each virtual link in the overlay network connects two edge nodes in an underlying physical network, such as MANET. Hence, the virtual link is a multi-hop path in the underlying network, as shown in Figure 1.1. The characteristics of a virtual link such as latency, bandwidth, and losses are characteristics of the constituting the path in the underlying network. Generally, if the data rate sent on a link exceeds its capacity, the link shows large delay. This situation becomes worse in P2P network when the traffic of multiple download sessions may traverse the same link. Hence, the overtaxed links would prematurely exhaust their bandwidth that causes congestion and excessive delay of packets. Typically, wireless links have lower capacity than wired links, and

therefore, they may suffer from congestion more frequently. The growing popularity of peer-to-peer (P2P) systems has necessitated transporting significant volumes of data efficiently while ensuring acceptable user response time. An effective congestion control mechanism is necessary for maintaining the performance of P2P network. A proper congestion control P2P system not only provides considerable savings in network bandwidth, but also ensures the benefit of fast information access.

1.1 Why P2P MANET?

Rapid advancements in wireless technology along with greater demands for user mobility motivate intensive research on P2P network in mobile environment. Both Mobile Ad hoc networks and P2P networks are decentralized and self-organizing networks with dynamic topology. Due to their common features, P2P file sharing seems natural and attractive to be deployed on MANET. P2P MANET can be used in different areas, such as:

- P2P MANET can be used in interesting application scenarios including sharing traffic network and weather data by car-to-car communication in a wide-range of MANET [9].
- Nodes in mobile ad hoc network usually have low computing capacity and therefore are unable to play role of servers all the time. In the comparison to MANET, the peer-to-peer applications appear to be powerful tools to spread information on this scenario.

- Since P2P network does not possess a unique service at a certain period of time, the assignment of distributed network tasks among nodes prevents them from becoming overloaded [16]. Moreover, the nodes that work on application level don't have to do anything on routing.
- Some MANET applications such as rescue operation in disaster recovery and information exchange in battlefield could run into the risk of single point of failure if all the information is stored in a single server. Especially in battlefields where servers could be the main target of attacks. P2P MANET avoids single point of failure by widespread replication of especially important files.

1.2. Motivation

The research in the area of Peer-to-peer connection in MANET is receiving much attention from emergency network setup, rescue team in disaster situation, military information share, conference within a company's different department, remote networking and battle field. With the increasing popularity of P2P network over MANET, it becomes necessary to think about the efficient information exchange and acceptable time to query search. When peer-to-peer network works on MANET, it introduces many complex issues. There are many open problems for research and significant contributions can be made.

Among all issues, efficient file discovery, acceptable user response time and active file downloading is the most common one. The performance of these issues is related to network congestion.

Figure 1.1: Mapping the overlay network topology to the network infrastructure

A peer node in overlay network, which is a neighbor of other peer at overlay network, might be far away from each other at network layer (see figure 1.1). The connection between these nodes is called virtual link. This virtual link is a combination of several underlying network links. As the underlying network is MANET, it suffers congestion problem due to node mobility and wireless medium used. If the underlying path becomes congested, the virtual path will be congested. So the total performance of the system would be worse. The tendency of a peer to send packets to its overlay neighbor without proper knowledge about the path would be a cause of congestion in the system. This problem will be more visible especially in random file downloading. Further more, most of the MANET P2P research efforts so far are still focused on the routing and topology adaptation, while congestion issues has not received sufficient attention. So it is very

important to do more research on efficient data handling in Peer to peer network. As far we know, Congestion issues over P2P MANET is our leading research attention.

1.3. Thesis Contribution

The objectives of this thesis can be stated as follows:

- Research on the congestion issues of peer-to-peer network in the MANET environment.
- Identification of problem that causes congestion in P2P MANET system.
- Design of a congestion awareness mechanism for a Peer to peer network that includes two feed back mechanisms. One is local and the other is external feedback.
- Investigation of mobility effect of mobile nodes on the overall performance with and without geographical restricted movements.

Our research concentrates on the issues of congestion in peer-to-peer network that works on MANET. Due to constrained resources of mobile host, the information-sharing links becomes overload over time. When a peer starts downloading a file from another peer in overlay network, the extra traffic flows through the link that may be the cause of congestion in overlay network. The situation gets sever if the download session starts randomly. We compare the performance of a P2P MANET system with download effect and without download effect. We observed that query success rate decreased, query delay increased and packet loss increased due to downloading activities in the system.

Our design considers the possible causes of congestion in P2P MANET system. The design also considers the neighbor situation when all links around it are unstable and the previous node gets a local feedback. The design also has the provision to select the appropriate destination for downloading a desired file. As a result, there is a significant improvement in the network performance. The objective of the design is to ensure better query success, with acceptable query response time and reduce packet losses in the network.

The thesis is organized as follows.

Chapter 2 provides background information about MANET and peer-to-peer network. It describes Gia and RAON, the two P2P networks that are the base systems for this thesis. It also describes Random Way-Point and Manhattan mobility models. Chapter 3 contains an evaluation of congestion situation in RAON, which leads to problem identification. It, then, describes the schemes proposed in this thesis of dealing with congestion caused by download traffic to avoid its harmful effect on query success rate and query delays. Chapter 4 presents simulation results and its analysis in evaluating the performance of the proposed schemes. Finally, in Chapter 5 we conclude the thesis and recommend some future works.

BACKGROUND

This chapter provides a discussion on the relevant background and related issues. The chapter is organized in four sections. Section-1 discusses the basic knowledge about MANET and related protocols. Section2 describes P2P network and the relevant protocols. Section3 describes the congestion that would happen in different types of MNAET and P2P networks. Finally, section4 introduces mobility Models that are used in our thesis.

2.1. Flow control and Congestion control in Wireless Networks

These flow control and congestion are related but topics distinct. They are related because all three solves problem [4]. They are distinct because they solve resource problems either in different parts of the networks or in different manner [10].

Flow control

If we consider a simple configuration where two nodes are directly connected via a link, they can exchange information at a pace too fast to the destination without any control. This may cause buffer overflow at the destination leading to packet losses, retransmissions and overall performance degradation [4].

Figure 2.1: Source and destination

A flow control scheme protects the destination from being flooded by the source.

Flow control is an agreement between a source and a destination to limits the flow

of packets without taking into account the load on the network [4][10]. The purpose of flow control is to ensure that a packet arriving at a destination will find a buffer there. Flow control solves the problem of the destination resources being the bottleneck. Different connections on a network can choose different flow control strategy.

Congestion Control

In P2P on MANET the nodes have limited memory, bandwidth, and processing speed. The querier nodes select download sites not only the basis of sites capabilities but also the link stability along the download path. Without this additional control the source may send packets at a pace too fast for the network, which results in long queues causing buffer overflow, retransmission and consequently query performance degradation [4]. Congestion control is primarily concerned with controlling the traffic to reduce overload in the network. A congestion control scheme protects the network from being flooded by its users.

In connection-oriented networks the congestion problem is generally solve by reserving the resources at all routers during connection setup.

Figure 2.2: Multiple Sources and Multiple destinations in a Network.

In the connectionless network, it can be done by sending explicit messages from the network to the source [4]. Congestion control solves the problem of the routers

and links being bottlenecks. All the MANET nodes in the network must follow the same congestion control scheme. Figure 2.2 shows the source and multiple destinations in a network that normally used in congestion control mechanism.

Congestion avoidance

Congestion control schemes help improve the performance of a network after congestion has occurred. It tries preventing the network from the congestion collapse. Congestion avoidance mechanism deal with congestion as it happens. The scheme that allows the network to operate at the knee of the congestion curve is called a congestion avoidance scheme.

2.1.1. Comparison between Congestion control and congestion avoidance

The distinction between congestion control and congestion avoidance is similar to that between deadlock recovery and dead lock avoidance [4]. Congestion control procedures are cures and the avoidance procedures are prevention in nature. A congestion control mechanism tries to bring the network back to the operating state, while the congestion avoidance schemes tries to keep the network at optimal state. The time when the congestion control scheme is called upon depends on the amount of memory available in the routers, whereas, the point at which the congestion avoidance schemes is invoked, is independent of the memory space [4]

2.1.2. MANET congestion control

Mobile Ad hoc network introduces several new congestion factors due to both the medium used and the mobility of the user [2]. Congestion control in Mobile Ad hoc

network occurs locally and must be dealt with in localized fashion [6] [8]. Mobility, Channel error and congestion are the main causes of packet losses in mobile ad hoc network [8]. Hotspots created in region of Mobile Ad hoc networks where flows converge and intersect with each other. This congestion may continue for longer period of time. Under such condition nodes consume more resources. Hotspot can be removed by Hotspot migration protocol (HMP)[33][6]. The simple goal of HMP is to redirect new routes away from the hot spots [8]. HMP directs new flows away from hotspots and congestion-prone areas, avoiding the further build up of traffic load in hotspot areas. HMP effectively changes hotspot conditions and reduces congestion-related problems [8].

Alternative multi-path routing protocols can also be used in conjunction with congestion avoidance scheme to generate alternative congestion free paths. A common feature of these protocols is that they utilize back up or alternate routes when the primary route fails. Some multi-path routing protocols are designed to distribute traffic among multiple paths [33].

The following issues make the wireless network more complicated

- Interference from other nodes
- Router failures
- Variable quality of radio signals
- Congestion
- Transmission power.

In [2], a dynamic load aware routing protocol is described. In this protocol the congestion level of a node along the path is considered. Each route is assigned a weight that is based on the length of time it takes to transmit a packets on that

link. The individual link weights are summed to get the weight for a route. The route of the lowest aggregate weight is selected. This method of route selection chooses links with short delay and high bandwidth, alleviating many of the transmission related issue [6][8].

2.1.3. Congestion in Peer to peer network

The growing popularity of P2P network has necessitated downloading substantial volume of data efficiently while ensuring the acceptable user response time [35]. Peers storing and handling high demand files may experience sever congestion and load imbalance. Once the node becomes congested user response time soars. When a node experiences packet loss in its incoming stream, it indicates other nodes for the incoming link as congested. The node should shed some incoming traffic [35].

If congestion is at or near the outgoing link of a node, the node shed out going traffic to alleviate the congestion. In contrast, If congestion is at or near the incoming link of the node, the node shed incoming traffic.

The situation is slightly different when a P2P network is deployed on MANET. The wireless link has several drawbacks, which aggravates the congestion in a network. If uncontrolled traffic traverses a link, the link becomes congested. If the link cannot accommodate the traffic then packet loss has happened. This shows the network is congested.

Congestion is a common cause of packet drops. We can analyse the congestion effect on packet loss by the network performance curve (Figure–2.3) [4][10]. According to the graph, if the load of the network is small, the throughput keeps up with the load. As the load increases, the throughput increases accordingly. RTT value remains

steady. After the load reaches the link capacity, throughput stops increasing and RTT value starts increasing. If the load increases any further, the queue starts building, potentially resulting in packets being dropped. The query response time also increases in this situation.

Figure 2.3: Network performance as a function of load [4][10]

Finally, as the queue starts overflowing, the RTT increase drastically. At the congestion collapse point, the throughput becomes zero and the RTT value goes to infinity. The bottleneck node has two choices when a packet comes in the non-congestion collapse situation. It may either discard the packet or buffer the packet. Buffering the packet also increases the RTT. The choice of optimum buffer length is not straightforward. Increasing congestion may show up as either increasing packet loss or increasing RTT. We got high value in both cases when the download session starts. So we can conclude that congestion is happening in our network.

In the next section, we describe our design that modifies query forwarding and proposes site selection by taking into consideration congestion in the underlying network.

2.2. Mobile Ad hoc Network

A Mobile Ad hoc Network (MANET) is an autonomous system of mobile hosts connected by wireless links. There is no static infrastructure in MANET such as base station [38] [12]. MANETs form a class of dynamic multi-hop network consisting of a set of mobile nodes.

Figure 2.4: A simple Ad hoc network

Each node in MANET can operate as a host as well as a router [37]. If two hosts are not within the radio range, all messages communicated between them must pass through one or more intermediate nodes [21]. The main advantages of MANET, such as no infrastructure and ease of deployment, make MANET an attractive choice for variety of applications. Example of such applications include communications in battlegrounds, in disaster recovery areas, among a group of islands or ships, conferencing without the support of any wired infrastructure and for interactive information sharing in remote areas [14].

2.2.1. Characteristics of MANET

In this section we describe the following characteristics of MANET:

1. Dynamic network topology
2. Bandwidth limitation
3. Low power and resource limited operations
4. Decentralized network control

Dynamic Network Topology

The nodes in Mobile Ad hoc Network can move freely within its designated area [37]. The movement may be random, freeway or fixed direction. This feature makes the network topology change unpredictably [37]. Also, the links of an ad hoc network can be bi-directional or unidirectional. Even though the network topology varies, connectivity in the network should be maintained to allow applications and services to operate without disruption [38][14][12]. The communications between the mobile nodes should be maintained properly. So an efficient routing mechanism needs for this purpose. Sometimes, this needs mobility management functions to support network access for devices located several radio hops away from a network access point [37].

Bandwidth limitation

Wireless links are expected to have substantially lower capacity compared to their hardwired counterpart [38]. Besides, the throughput of wireless communications in real environments is often much less than a radio's maximum transmission rate, because there may be the effects of multiple access, fading, noise, and interference conditions, etc [38] [18][19]. Further, multiple routes can share several common

links that could disrupt several sessions during periods of high bit-error transmission rates. [14][38].

Low-Power and Resource-Limited Operation

In most cases, the nodes in a MANET may depend on batteries or other exhaustible means for their energy [38]. This feature makes the power budget tight for all the power-consuming components in a mobile device. For example, this will affect CPU processing, memory size and usage, signal processing, and transceiver output/input power [37]. For these nodes, energy conservation should be considered for the optimization as a key system design criteria [38].

Decentralized Network Control

As an advantage, the decentralized nature of network control in MANET is robust against the single points of failure of more centralized approach [38].

2.2.2. MANET routing protocols

Among the existing routing protocols for MANET, we can differentiate them in two categories.

- Pro-active or table driven routing protocols and
- Reactive or on-demand routing protocols.

2.2.2.1 Proactive Vs Reactive routing protocols

In proactive routing protocol, the whole network view is available at every time [21][38]. Most routing protocols in MANET derive from distance vector or link state algorithm [21]. In *distance-vector routing*, each router maintains a table containing the distance from itself to all possible destinations. Each node

periodically transmits this table information to its neighbor node, and updates its own table by using the values received from its neighbors. A neighbor can decide the shortest path among all possible paths for communicating with each other [21][34]. When each node has a packet to send to some destination, it simply forwards the packet to the next hop router. DSDV is a proactive routing protocol that is based on the classical *Bellman-Ford* routing algorithm to apply to MANET. Reactive routing protocol is an on demand routing protocol. Routing information is neither maintained all the time nor periodically advertised. That is, only when two nodes need to communicate with each other, will they forward routing packets to maintain connectivity between the two nodes [21] [37]. Usually, when there is a need for communication between two nodes, each mobile node transmits a local broadcast packet known as a *hello* message [21]. By exchanging routing information, they communicate data packets with each other. AODV and DSR are the reactive routing protocols. We chose AODV for our underlying network. The, main benefit of the on-demand protocols is that the “routing overhead” (typically measured in terms of the number of routing packets transmitted, as opposed to data packets) is typically lower than proactive routing protocols.

2.2.3. Ad-hoc On-demand Distance vector (AODV)

Our underlying protocol for MANET is AODV. The AODV routing protocols is the combination of DSDV and DSR routing protocol.

Figure 2.5: AODV protocol messaging

It offers quick adaptation to dynamic link conditions, low processing and memory overhead, low network utilization, and determines unicast routes to destinations within the Ad hoc network [39][38]. When a source node desires to establish a communication session, it initiates a route discovery process to locate the destination node, by generating a route request message, which might be replied by the intermediate nodes in the path to destination or the destination node itself. Route Requests (RREQs), Route Replies (RREPs), and Route Errors (RERRs) are the message types defined by AODV (figure-2.5). When a node wants to find a route to a destination, it broadcasts RREQ message [39]. The RREQ contains the source address, the destination address, and a unique sequence number. The range of dissemination of such RREQs is indicated by the TTL in the IP header [RFC 3561] [39]. As long as the endpoints of a communication connection have valid routes to each other, AODV does not play any role. When a route to a new destination is needed, the node broadcasts a RREQ to find a route to the destination. A route can be determined when the RREQ reaches either the

destination itself or an intermediate node with a 'fresh enough' route to the destination [39] [RFC3561].

A 'fresh enough' route is a valid route entry for the destination whose associated sequence number is at least as great as that contained in the RREQ [RFC3561]. A node receiving the RREQ may send a route reply (RREP) if it is either the destination or if it has a route to the destination with corresponding sequence number greater than or equal to that contained in the RREQ. If this is the case, it unicasts a RREP back to the source. Otherwise, it rebroadcasts the RREQ. Nodes keep track of the RREQ's source IP address and broadcast ID. If they receive a RREQ, which they have already processed, they discard the RREQ and do not forward it [21] [39]. As the RREP propagates back to the source, nodes set up forward pointers to the destination. Once the source node receives the RREP, it may begin to forward data packets to the destination. If the source later receives a RREP containing a greater sequence number or contains the same sequence number with a smaller hop count, it may update its routing information for that destination and begin using the better route. As long as the route remains active, it will continue to be maintained. A route is considered active as long as there are data packets periodically travelling from the source to the destination along that path. Once the source stops sending data packets, the links will time out and eventually be deleted from the intermediate node routing tables [39][RFC3561]. If a link break occurs while the route is active, the node upstream of the break propagates a route error (RERR) message to the source node to inform it of the now unreachable destination(s). After receiving the RERR, if the source node still desires the route, it can reinitiate route discovery.

2.3. Peer-to-Peer (P2P) network

Peer to peer network is an overlay network for distributed objects store, search and sharing. P2P network gains popularity as low-cost individuals computing and storage resources become widely uses [6][8]. Peer to peer system aggregate large numbers of computers that join and leave the network frequently and that might not have permanent network addresses [23][24]. This network breed of systems creates application level virtual networks with their own routing mechanism that allows individual computers to share information and resources directly without dedicated servers [25]. In P2P network nodes have equal functionalities and provision capabilities and, as a consequence, are called “peer” entities. Every peer is able to send and reply to request message originated from each other. This shows the dual interface of these peers, since they might play the role of servers and clients simultaneously. That’s why they are called as *servents* (SERVers/cliENTS) [1]. In P2P system nodes are equal and can act as a client, as a server and as a router [20]. This approach provides important benefits, such as improved scalability and fault tolerance, and cost maintenance reduction due to self-organizing nature of the network. Some other popular applications can be deployed on peer to Peer network, namely multicast and VoIP [1].

2.3.1. Overlay Network

In overlay networks, all participating hosts logically connected to small subset of other participating hosts at the application level [1]. One host is called the neighbor of another host if they are connected directly via logical link [24] [30]. Data transfer is happened through underlying network. However, data transfer is

not as efficient as in the underlying physical network because a nearby node in an overlay network may actually be far away from each other in the underlying network [1] [25]. This in turn might lead the waste of network resources and degraded performance significantly. Peer to peer network is an overlay network like Gia, Gnutella, kazza, RAON, Napster. The underlying network can be wireless or wired network.

Peer-to-peer network can be divided into two types:

1. Unstructured and
2. Structured

Unstructured Vs structured P2Pnetwork

Unstructured P2P network organize nodes into a random graph and use flooding or random walks to discover data stored by overlay nodes [19]. Each node visited during a flood or random walk evaluates the query locally on the data items that it stores [23]. Some examples of P2P network are Gnutell, GIA, Napster etc. Structured overlays assigns keys to data items and build a graph that maps each key to the node that stores the corresponding data. Structured overlays were primarily developed to improve the performance of data discovery [35]. The data or a pointer to the data is stored at the node responsible for its key. Structured P2P system uses distributed hash tables to directly locate the peers that contain the desired content [23]. It is commonly believed that structured graphs are more expensive to maintain than unstructured P2P network [24]. Constrains imposed by the structure make it harder to exploit heterogeneity effectively. Some examples of structured peer-to-peer system are CAN [24], Chord [5,24], pastry [24], and Tapestry [24].

2.4. Unstructured Peer-to-Peer Network

Though Gnutella is an unstructured P2P network, but it suffers by scalability problem [15]. Gia combines numerous schemes that were proposed for improving the scalability and robustness of such system. The detail of Gia design is described below.

2.4.1. Gia Design

The basic problem of Gnutella like system is scalability because Gnutella uses flooding mechanism in the search protocol [1]. To locate a file, a node send its queries to each of its neighbors, which in turn propagate the query to their neighbors, and so on until the query reaches all of the clients within a certain radius [24] [23]. This flooding mechanism can quickly overload the nodes and it becomes worse when the network size increases. To overcome the flooding draw back, random walk technique is proposed. Random walk is a well-known technique in which a query messages is forwarded to a randomly chosen neighbor at step until sufficient response to the query is found.

Although it makes better utilization of the P2P network than flooding, random walk is essentially a blind search. At each steps a query is forwarded to a randomly chosen node without taking into account any indication of how likely it is that the node will have response for the query [1]. If a random walk query arrives at a node that is already overloaded with traffic, it may get queued for a long time before it is handled. Gia modified Gnutella to incorporate the ideas of super nodes and random walk [1]. The capacity of a node depends upon a number of factors including its processing power, disk latencies and access bandwidth. The Gia

design consists of four main components (a) dynamic topology adaptation (b) active flow control (c) One hop replication (d) search protocol [15].

Dynamic Topology adaptation

Topology adaptation ensures that high capacity nodes are well-connected nodes and the low capacity nodes are within short reach of higher capacity nodes. Topology adaptation is used for creating overlay topology [15]. The capacity is a quantity that represents the number of queries that the node can handle per second. The capacity is determined by the node's access bandwidth, processing power, disk speed etc. The number of connections a node makes to other nodes depends upon its capacity level; higher the degree more will be the connections the node is able to make. Each Gia node maintains a host cache consisting of information about other Gia nodes (their IP address, port number, and capacity). Neighbors inform each other about their capacity levels and current degree when they initially connected [15]. Each node independently computes a level of satisfaction (S). This is a Quantity between 0 and 1 that represents how satisfied the node is with its current set of neighbors. A value of $S=0$ means that the node is quite dissatisfied, where $S=1$ suggest that the node is fully satisfied [15]. As long as a node is not fully satisfied, the topology adaptation continues to search for appropriate neighbors to improve the satisfaction level. Satisfaction level increases as it gather more neighbors up to a certain predefined limit.

To establish a new connection, a node say node A, randomly selects a small number of candidates from its host cache that are not marked dead and are not already neighbors [15]. It selects the one that has maximum capacity greater than its own capacity. If the new node doesn't exceed the X's predefined maximum

neighbor value then the connection would accept. Otherwise the node must see if it can find an appropriate existing neighbor to drop and replace with the new connection. X always favors Y and drops an existing neighbor if Y has higher capacity than all of X's current neighbors [15].

Flow control

Gia uses an active flow control scheme in which sender is allowed to direct queries to a neighbor only if that neighbor has notify the sender that it is willing to accept queries from the sender [15]. Gia uses random walks to forward a single copy of each query. To provide better flow control, each Gia client periodically assigns flow-control tokens to its neighbors. Each token represents a single query that the node willing to accept [15]. Thus a node can send a query to a neighbor only if it has received a token from that neighbor. In the long run a node allocates tokens at the rate at which it can process queries. If it receives queries faster than it can forward them (either because it is overloaded or because it has not received enough token from its neighbor), then it starts to queue up the excess queries. If this queue gets too long, it tries to reduce the inflow of queries by lowering its token allocation rate [15]. Gia uses start-time fair queuing (SFQ). Each neighbor assigned a fair queuing weight equal to its capacity. Neighbors that are not using any of their assigned tokens are marked as inactive and the leftover capacity is automatically redistributed proportionally between the remaining neighbors.

One hop replication

Every Gia node maintains an index of the content of each of its neighbors. These indices are exchanged when neighbors establish connections with each other and

periodically update any incremental changes. Thus when a node receives a query, it can response not only with matches from its own contents but also provide matches from the contents offered by all of its neighbors [15]. When a neighbor is lost either because it leaves the system, or due to topology adaptation, the index information for that neighbor gets flushed. This ensures that all index information gets up to date.

Search Protocol

High capacity nodes have more neighbors (highly connected) and one hop replication ensures that high capacity nodes can typically provide useful response for a large number of queries. Biased random walk is used in Gia that queries will be forwarded to high capacity node first. [15]. If the query doesn't find its response, the query is forwarded to another neighbor. Gia uses TTL to bound the duration of the biased random walk and assigns each query an unique GUID by its neighbor nodes. If a node finds a same query twice with same GUID it just forwards the query to different neighbor.

2.5. Similarities and dissimilarities between MANET and P2P network

Both Mobile ad hoc network (MANET) and P2P networks are decentralized and self-organizing with dynamic topologies and responsible for routing query [9].

Figure 2.6: Peer- to-peer network over MANET.

MANET is composed of resource-constrained devices that are susceptible to faults; where as peer-to-peer networks are fault tolerant [2]. MANET and peer-to-peer network are not only similar, but also they are complementary too. A comparison between P2P and MANET has given below [21].

Items	P2p network	MANET
Motivation for creating the network	Create a logical infrastructure to provide a service	Create a physical infrastructure to provide connectivity
Connection between two nodes	Wired and direct connection (at the p2p level)	Wireless and indirect
Connection confidence	High (physical connection many paths in internet)	Low (wireless connection)

Items	P2p network	MANET
Structure	Physical apart from logical structure	Physical structure corresponds to logical structure
Routing	Only reactive algorithm is possible, reliable algorithm not implemented yet	Reactive, proactive and reliable algorithm is exists.
Broadcast	Virtual, multiple unicasts	Physical to all nodes in transmission range area

Table –2.1: Comparison between P2Pnetwork and MANET [21].

2.6. RAON: A P2P Network for MANET

The Resource Aware Overlay Network (RAON) is designed based on Gia as an overlay network and the underlying network is MANET. RAON uses Gia as a foundation and have the features to address problems specific to MANET. P2P network is mainly designed to work on wired network. RAON is designed to work over wireless network [1]. The main features of RAON are: modified random walk query forwarding to take into consideration link instability and neighbors power, and topology adaptation in case of the neighbor is connected through highly unstable links.

MANET and P2P share the common characteristics of dynamic topology and lack of infrastructure. MANET nodes have the characteristics of unpredictable mobility, link instability, and limited power. RAON considered these factors to design a P2P overlay network. RAON implementation considers all the four design features of Gia like Topology adaptation, flow control, one-hop replication, and biased random walk. But RAON implements only topology adaptation and search protocol. Query forwarding algorithm used in RAON is biased random walk. This forwarding method greatly reduces the number of queries that is caused by

flooding algorithm and at the same time offers high probability of getting query hit than a pure random walk. This approach is favorable to MANET since it greatly reduces the number of messages generated per query. Since the biased random walk always tends to forward queries to high capacity node and avoid the links those experience more traffic than other links. RAON considers link instability and power constraints and modifies Gia's design to adapt on MNAET.

2.6.1. RAON Design features

RAON identifies that Gia is a perfect match to MANET as an overlay network on MANET though Gia was designed for wired network [1]. The performance of query search is defined, as the time required for the query originator to receive the query hit repose for the particular query. As the link instability is not uncommon in MANET, the performance of query search can suffer badly if it is not aware of the instability of the links to high capacity nodes. Taking into consideration of link instability and power constraint, RAON modifies Gia's design to adapt to those constraints in MANET [1]. RAON employs a link-coloring scheme described below.

Link Coloring Scheme (LCS)

As RAON consider link instability and power consumption, RAON nodes keep track of delay that is experiencing with its neighbor and neighbor energy level. The LCS at a node monitors the RTT and residual power of each of its neighbors, and based on their values color the neighbors to GREEN, YELLOW or RED. In RAON, biased random walk algorithm also takes neighbor's color into consideration when making a forward decision in addition to the neighbor capacity levels. Among the available neighbors, it groups them according to their color and selects the one

with the greatest capacity from the highest color group. Thus, if neighbor A has a higher capacity but lower color than neighbor B, the query would forward to B instead of A.

Neighbor's color	Condition
GREEN	Energy = High && LATENCY = LOW
YELLOW	ENERGY = MEDIUM && LATENCY !=HIGH (LATENCY = MEDIUM && ENERGY != LOW)
RED	ENERGY = LOW LATENCY = HIGH

Table 2.2: Condition for changing a neighbor's color. The symbol && and || refers to the logical AND and OR operations receptively.

Thus the rank of a neighbor is lowered if either the delay is high or the energy level is low. RTT and power level is measured for calculating the rank of a neighbor. Each node determines the delay by probing the neighbors periodically and measures the RTT. The table 2.2 refers to RAON link color scheme. The node send a *prob* message every 30 sec. If the node does not receive the Acks, the RTT has the worse case value. LCS provides a scheme for the forwarding algorithm to avoid using high delay routes or consuming power of low energy neighbor.

2.7. Mobility Model for Mobile Ad hoc Network

The performance of MANET depends on node mobility pattern [41]. Nodes of a MANET may move arbitrarily, causing frequent disruptions and changes in routes. Highly dynamic, random and sometimes rapidly changing topologies is, therefore,

an inherent feature of MANET [16]. The performance of MANET in terms of throughput, latency, and scalability is related to the efficiency of the routing protocol [41]. Nodes need to adapt to the network topology caused by mobility. Signaling overhead for route maintenance of routes in MANET is proportional to the rate of link changes, which in turn is a function of the mobility of the nodes [41]. In order to conduct meaningful performance analysis of routing algorithms for MANET it is essential that the mobility model on which the simulation is based reflect realistic mobility behavior. In [41], The author provides a comprehensive survey of mobility models used in simulating ad hoc networks. Mobility models are divided into two categories:

- **Entity mobility model:** Entity mobility model specifies individual node movement.
- **Group Mobility model:** Group mobility model describes group movement as well as individual node movement inside a group.

Random Waypoint and Manhattan Grid (similar to City section mobility model) are characterized to be entity mobility model. RPGM is a group mobility model. Ref [1] used only random waypoint mobility model in evaluating RAON's performance, which is not geographically restricted; but in evaluating CAON's performance in Chapter 4 we use both random waypoint mobility model and Manhattan mobility model. The main objective is to use another mobility model to see the effect of mobility on performance. Manhattan mobility is mainly designed to simulate node movement in urban areas.

Random Waypoint Mobility Model

In random way-point model, at every instance, a node randomly chooses a destination and moves towards it with a velocity chosen based on uniform random distribution in the range of $[0, V_{\max}]$, where $V_{\max}$ is the maximum allowable velocity for every mobile node [1].

Figure 2.7: Random way point mobility model

When a node reaches its destination it pauses for an interval before taking another destination to move. When pause time expires, another destination and moving speed are randomly picked. The speed is uniformly chosen between specified maximum and minimum speed. Then the mobile node will move towards the destination with the chosen speed. Figure -2.7 shows the random way point mobility model.

Manhattan Mobility model

Manhattan Mobility model emulates the movement pattern of mobile nodes on streets in the metropolitan area, which is defined by Manhattan Grid map [41].

Manhattan Grid is proposed to model a city section with streets crossing each other perpendicularly [41]. Unlike the Random Way point model where the nodes can freely move anywhere in the simulation field, the mobile node in Manhattan model is restricted to travel along the grid of horizontal and vertical streets on the

Figure 2.8: Map for Manhattan mobility Model

map with the randomly chosen velocity [41]. So, in the Manhattan Mobility model, the geographic restriction is imposed for node movement. The node then chooses a random destination and moves towards this destination within a predefined speed range [41]. Upon reaching the destination, the node pauses for certain time and then repeat the process again. Figure- 2.8 shows a Map that nodes can traverse.

2.8. Summary

In this chapter we reviewed the background that is necessary to understand our thesis. Our thesis is based on P2P network and Mobile Ad hoc network. P2P network works at application level and MANET works at network level. We described MANET, its characteristics and the protocols that we used in our design. We used on demand routing, AODV in the simulation of the underlying MANET in Chapter 4. We provided an overview of AODV protocol. We describe Gia, an unstructured P2P network, in detail, as this is the basis of RAON and CAON that is

proposed in this thesis. Then we described RAON, which we modified to develop CAON. Finally we described two mobility models for MANET, one is Manhattan which is geographical restricted, and other is random way point mobility model, which is not restricted to any particular geography.

The aim of this thesis is to design a congestion aware P2P network that works on MANET. Both MANET and P2P file sharing systems lack fixed infrastructure and possess no priori knowledge of arriving and departing nodes. Due to this common nature, P2P file sharing seems natural and attractive to be deployed on MANET. Node capacity and Bandwidth are the limiting constraints in most Mobile Ad hoc networks. The overlay virtual link is the combination of several underlying links. Whenever the incoming traffic load exceeds the capacity of a node, the node buffer becomes full. So the node becomes congested and packets start dropping. Especially when multiple hosts try to access a shared link simultaneously, the link becomes overloaded. We took RAON (resource aware overlay network) as our base whereas RAON is the combination of Gia (overlay) and MANET (underlay). That's why we are going to analysis Gia's congestion control first and then RAON situation. Finally we will go for our design analysis.

3.1. Problem Identification

The flow control of Gia is based on token distribution to its neighbors according to its capacity. The capacity assigned to a node is based on bandwidth and number of queries it can process per second. In Gia, a node sends query to its neighbor that is highly connected. So the traffic around the highly connected node makes aggregated traffic. The aggregated traffic in Gia would be varying characteristic due to topology adaptation that may overload the highly connected node. In Gia, it

is assumed that all nodes generate queries at the same rate and the queue length is infinite. But practically queue is not infinite. If a high-connected node becomes busy due to a high aggregated traffic load coming to the node, the packet is stored in the queue at that node. Eventually, as the node serves its queue, the packet is forwarded towards its destination. Usually congested queue means longer delivery times. The delivery time for a packet from its source to its destination got varying characteristics. As the load increases, the delivery time will typically increase accordingly. There is a critical load where the delivery time diverges, or at least increases dramatically. At this point the network is congested [7].

Gia's flow control mechanism is basically designed for the network where end-to-end connection is wired. However, MANET inherent characteristics of frequent topology changes, channel error, transmission limitation, and link breakage add complexity to routing and make the network unstable. So Gia's flow control needs to be modified in wireless medium. In overlay network, the nodes are connected via a virtual link. Each virtual link may consist of multiple physical hops. So, token distribution may not work appropriately if the underlying link condition is not stable. If a virtual link is unstable due to underlying link congestion or breakage, then a node may consume tokens in forwarding queries that are eventually lost. As a result, the overlay network starves sometimes for query or overloaded. Congestion control in overlay network in mobile environment should be dynamic, fast and adaptive.

Resource Aware Overlay Network (RAON) has a little control for forwarding queries to its neighbor but doesn't have any control of network traffic. RAON is implemented to improve the query performance of Gia under different MANET

situations. RAON mainly implements the topology adaptation of Gia in MANET environment. In [1] RAON's performance was not evaluated by including the download traffic.. Download traffic can be huge to significantly impact query performance. RAON query forwarding performance is based on link condition and node energy. Situation of a link with download traffic and without download traffic may be quite different. The link that was not unstable before download session begun might become unstable as a result of download traffic. This will affect the link colors of RAON, which eventually would change the query forwarding decision. Hence, we do not expect the query success rate to be the same in both cases of with download traffic and without download traffic.

Since RAON works in MANET environment, it faces many challenges posed by link instability, node congestion, node transient, intermittent disconnection, low battery power, and resource constraints. Nodes in RAON get overloaded over time due to lack of congestion control. RAON only consider query-forwarding control in different MANET scenarios. It doesn't consider the effect of downloaded traffic that might flow over the same link as the query packet flows. Gia's flow control is only for preventing overloaded condition of a node where the end-to-end connection is wired. However that is implemented at the application level is not sufficient for a MANET node. In the wired network all P2P nodes are assumed to be end hosts, where they do not do any routing. The underlying network condition changes all the time. Hot spots are often created in regions of mobile ad hoc network where flows converge and intersect with each other. So if the underlying

network is congested due to heavy traffic, node mobility, wireless medium error or bandwidth fluctuation, it will affect the performance of P2P network.

Even though a P2P node controls token assignment to its neighbors, it may still experience congestion due to high underlying network traffic caused by download and control packets. Many download sessions may take paths that may traverse the same unstable links causing performance degradation due to excessive traffic contributing to link instability. Hence, a proper congestion control mechanism is needed in RAON.

The goal of RAON is to improve the performance of Gia under different MANET situations. Link color is the additional factor other than capacity of a node in RAON for forwarding query to that node. This additional factor is expected to improve the quality of forwarding decision and as consequence improve the query success rates and query delays. From our study, it is observed RAON suffer 10-15 % query losses in without download condition [1]. It is observed that it consumes more energy than Gia and creates more dead nodes [1]. We talked about possible downloading effect on network performance. The link might change its color from GREEN to RED over time due to heavy download traffic flow over the link. So obviously query forwarding decision will be changed. Let us analyze the simulation results of RAON with download traffic.

3.1.1. Analysis of RAON's performance with download traffic

We simulated the RAON for download and without download condition. We observed that the system performance goes down due to downloading effect. We observed RAON suffers significant query losses, query delay as well as more packet

losses. We tested the system first in random waypoint mobility model and final we tested the system for Manhattan Mobility model as we like to see the effect of geographical restricted movement on the performance.

Case-1: Query success rate

Figure-3.1 shows the query performance under random waypoint mobility model for download and without download conditions. It shows that RAON performance degrades 20% even more in some cases due to downloading effect in different

Figure: 3.1: Query success rate of RAON (Here D1 for download and D0 for without download)

MANET scenarios. If we pick one scenario with and without download, like speed 5 m/s, the yellow one, we can conclude that RAON performance degrades almost 20% even more in case of peer 30.

The query performance also decreases with the increase of number of peers. This indicates, the network becomes congested when number of peers increase. This is the indication of more network activities due to more number of peers.

Case –2: Query delay performance

We measured query delay for RAON in different speeds. We measured the query delay and plotted query delay with increasing number of peers. Figure 3.2 indicates the delay behavior of RAON with and without download option. The speeds we used are 20 m/s, 5 m/s and 2 m/s. Figure 3.2 clearly shows that the delay time increases in case of downloading function.

Figure: 3.2: Query delay vs. number of peers

If we consider speed 20 m/s from the figure 3.2, we can observe that delay time increases almost 4 sec for downloading activities. The figure also shows that query delay increase with the increases of peer numbers. The cumulative effect of this delay makes the network inefficient and unresponsive.

Case –3: Duplicate Acknowledgement

As we know, a packet loss is an indication the network is congested; a packet loss can be detected via duplicate Acks. Out of sequence packet delivery generates duplicate Acks. But, out of order is not a real issue for P2P systems, since messages can be processed despite of their order of arrival. From the figure-3.3, we can observe that

duplicate

(a) Speed 5 m/s

(b) speed 20 m/s

Figure 3.3: Duplicate Acks Vs number of peers

Acks also increases in the network for downloading effect and the increase of the number of peers. Figure 3.3 shows that speed also has an impact on duplicate Acks. The above two figures are for speed 20 m/s and 5 m/s. Due to downloading effect the duplicate Acks increase as well.

3.1.2. Our Hypothesis

Why this significant amount of performance degradation happens in RAON system after including a download function? Following is our hypotheses to identify possible causes.

- When query hits happen, the source node makes a separate TCP connection with the destination node. Downloaded traffic is heavy compare to any other traffic in the network. The path, which is followed by the downloading packets, was not pre-defined. So it may be happened that downloading packets follows the unstable link many times. If the download session starts with YELLOW link, the link get unstable very soon and converts to RED and the RED link becomes more congested. So it was uncertain in RAON whether the query response will travel the same GREEN link or not. Because there is no tracking mechanism in RAON.
- RAON doesn't have any idea to deal with no neighbor situation (all links are RED). If all neighbors around a node are unstable, it is wasteful to send any more queries to that node again. The unawareness of this situation is forced to send a query through the unstable link. This increase the query travel time.
- We observed that RAON generates more dead nodes over times [1]. This may happen due to flow of downloaded heavy traffic in the link and the nodes kept busy for long time for giving service to its neighbors. So battery power exhausted rapidly.
- The aggregated traffic may be the cause of congestion in the network.
- By increasing the number of peers in the network, the network activities also increase.
- Multiple physical hops in the underlying network usually form a virtual link between two nodes in overlay network. If a mobile node along the path moves from its current position, the virtual link may break. Causes AODV restore the link. Due to route failure the intermediate node may develop large queue. In that situation if the sender is not aware of the link condition and sends queries along

that path again, eventually query loss happen and performance degradation n occurs. Route failure may lead to congestion a node. From the above discussion we assume that the network faces congestion due to downloaded traffic. The next section we will analysis our assumptions.

In order to justify our assumptions, we measured the packet losses for our P2P MANET system for the entire simulation. We plotted the graphs in every 25-sec to see the exact picture of packet losses.

Figure 3.4: Packet loss vs. simulation time

We can observe from the above figure- 3.4 that packet loss results with download option. The total number of packet losses with higher download is higher than the packet losses without download. We can also conclude from the figure – 3.4 that Packet loss is less at the beginning of the simulation for the both systems and it starts increasing after 175 sec. At the time of simulation starts, a less number of peers involve in the network activities. So less number of packets flows in the

network at the starting time of simulation. When the simulation time increase, the network activities also increase as well.

Figure 3.5: RTT Vs Simulation time

So, the more number of controlled packets and downloaded packets flows in the network. Due to this heavy traffic, network becomes congested. Furthermore, if a link breakage happened in the underlying network, the packet loss will be more. At that time if we measure the RTT of the link, the value will be higher. We can also observe from the above graph, Packet losses are not significant up to the simulation time 400 sec and after that the losses starts rising due to congestion of the system. To clarify the cause more clearly, we plotted the RTT graph with simulation time for the same network in the same time interval (see figure-3.5). We observed that the RTT graph follows the loss graph trends in the same manner. Where the higher packet loss happens, the RTT value also higher. We use RTT measurements to identify the link condition as well as congestion. This is a clear indication of congestion in the network that accelerates packet losses. We can make a general

conclusion that the network performs well at the half time of the simulation time, after that the network gets congested.

We measured the average number of unstable link that generates in the network for the whole simulation time. We mentioned in background chapter that a link is to be considered unstable when it got RED mark. RED mark to a link means when the RTT value of a link exceeds 3 sec. We observed from our calculation that unstable link is increased due to download option in the system. The higher number of unstable link means, the higher chance of congestion. We plotted figure- 3.6 for average number of peers and average number of unstable links. We measured this unstable links for the speed of 20 m/s and 5 m/s (see the figure 3.6).

Figure 3.6: average number of unstable link for the speed 20 m/s and 5 m/s

From Figure 3.6, we can calculate the average number of unstable link differences between download times and no download times. The system generates 925 more unstable links due to the download effect (see figure 3.6 (b)).

We can also observe from the above graph that the number of unstable links also increase with the increase of speed. By increasing the speed, more number of link breakage happens. Another possible cause of this increasing number of unstable link is the downloaded traffic. We observe from figure-3.5, RTT also increase with the downloaded effect. By considering all of our problem identifications and analysis, we can conclude that download traffic cause congestion in our P2P MANET system. It is clear from the above discussion that RAON is suffering congestion problem due to downloaded traffic.

3.2. CAON Design

The ideas that we consider in designing our Congestion Aware Overlay Network (CAON) are described below:

- Link must be *stable and highly connected* for forwarding a Query
- Select a stable link for Downloading.
- Local feedback mechanism is needed in case of no neighbor situation.
- Monitor RTT

CAON is a P2P network on MANET that consists of the following components: (1) Congestion monitoring and local feedback mechanism (2) Alternative route selection for sending query (3) search of a stable peer for downloading purpose.

3.2.1. Local feed back mechanism

Aggregate traffic, link failure and channel errors are the causes of high RTT in a wireless network. Each node can determine the delay by probing the neighbors periodically and measure the RTT. A node sends a *prob-message* to its neighbor and it sleeps for *prob- interval* seconds and waits for an Acks.

When the timer expires, it would consider the previous probe to be unsuccessful and assign a worst-case rtt to that neighbor. Delay is measured by measuring the average RTT for a node with its neighbor. We used RAON's coloring scheme that is described in chapter-2 table- 2.2. RED means more unstable or congested, YELLOW means moderate congestion and GREEN means the link is not congested. If a link carries more traffic, the link will be congested and RTT will be high.

Figure 3.7: An example of query search

So the link color will be RED, if the RTT value exceeds 3 sec. The threshold that is used for determining the link color is 1 sec and 3 sec. If the average RTT value is less than 1 sec then the Link will be GREEN. The average RTT between 1 sec and 3-

sec means the link is YELLOW. If the RTT value is more than 3 sec, the link is Red and its condition is unstable and congested.

As shown in the above figure-3.7, node A send query to B as B is highly connected node and link B condition is GREEN. Node B sends query to node E for the same conditions. At node E when all links are RED, which is a no neighbor situation, then it will give a local feed back to B to inform that “All links around me is RED, don’t send any more query to me”. So the previous node (B) will be informed not to forward any more query to E although the B’s immediate link is GREEN. If the situation changes after one probe interval, then respective node will inform its current status to its previous node again. Every time the query packet changes its color by traveling a link. If the link color is RED, the query gets RED color. Once the query packet gets RED, it will keep RED up to its destination. The query packet color changes all the time until it reaches the desire destination. CAON used biased random walk for query forwarding. A CAON node sends a query to its high-connected node as the high-connected node has more possibilities to get queries hit. According to one hop replication, a high-connected node gets the file index of all of its neighbors. So CAON sends query only high-connected node if the link color is GREEN. If the high connected node link color is RED, CAON will select another node, which is less connected, but link color is GREEN. In addition, if a CAON node gets all links as RED, the node will inform to its previous neighbor about the situation. For simplicity our design uses only RED and GREEN. If we get YELLOW we will assume that is also RED. Once the query color is RED, the query color will not be changed up to destination whatever the link it travels further. Because, once a link in a path is unstable, it is assumed that the whole path is

unstable. So it gets the permanent RED color. Table 3.1 gives the condition for query to change its color.

Link color	Query color (at the time of arrival)	Ultimate Query color	Local feedback	Query hit (Feedback)
GREEN(any one)	GREEN (if never RED)	GREEN	No	GREEN
RED (all links)	GREEN RED	RED	Yes (RED)	RED
GREEN	RED (if already RED)	RED	No	RED

Table 3.1: Query color changes & feed back options

3.2.2. Selections of download site

When node B gets a RED feed back from node E, node B will assume that node E gets no neighbor situation. So node B will not send any more queries to E until next update. Node B, then search another neighbor for the same query that have GREEN link. As soon the node gets the GREEN link and the node is highly connected, then it will send the query to that node. If node B enters the no-neighbor situation, it will send RED feed back message to its originator and wait up to probe interval time.

When the query-hits happened, the destination node sends a query response to the respective query originator according to its query color. The source may get query response in RED or GREEN. AS soon as the GREEN query response received by the source, the source sends a TCP request to its destination for downloading a file. If the node is not busy then it will accept the request and starts downloading packets.

If the node is busy, the node can hold the request for the next service. If the request makes queue, drop the latest request. The operation of downloading function causes more stress on link. As a result, the link may become congested and consumes more resources, since downloaded packet is generally larger than query messages. We believe that if the download session can avoid the RED link during download session the link condition will be improved. If the source node doesn't get GREEN response, the node should not go for any download connection because this will hamper the other query messages those are traveling through the link. Figure 3.7 shows that J node get GREEN Query hits. So the download connection has been set up between A and J not A and L. if the node runs low in bandwidth, due to congestion or serving multiple download requests, the node would serve the GREEN request that arrived first.

Let Cap_i represent the capacity of neighbor i
 Let Q_Pkt_j represent the Query Packets color
 Let $link_i$ represent the link color of neighbor
 Let Q_hit_j represent the Query_response color

$Max_capacity \leftarrow 0$
 $Max_node \leftarrow x$

Struct color {
 $Max_color \leftarrow GREEN$
 $Low_Color \leftarrow RED$
 } $link_i, Q_Pkt_j$

Q_Pkt_j color will be change according to link color

//Query forwarding decision

If ($link_i == Max_color$) and ($Cap_i > max_capacity$)
 Then {
 $Max_color \leftarrow link_i$
 $Max_node \leftarrow i$

```

}
return i
// For no neighbor situation

For Q-Pktj{

if (All Linki == Low_color ) {
Q-Pktsj = Low_color
feed back message to previous node.
Sends update message after next probe interval.
find another node which is Max_color & next high connected
}
Else {
(Q_Pktj == linki, if Q_Pktj already not low color )
no feed back message // to previous node
Q_Pktj already low color
Q_Pktj == Low color up to destination
}
}
//For Destination node

If (Q_Pktsj){
Send Q_hitj == Q_Pktj // same color response
}
//For source

if (Q_hitj == Max-color){
Send a Down load connection request to destination node
If destination node is not busy
Set up a download connection
If busy, hold request upto probe interval // because the link color may change over time
Source will monitor is there any new Q_hit
}
else {
wait for Q_hit == Max_color // up to prob_ interval time
No Q_hit == Max_color
Discard Q_hit & Send new query again
}
}

```

Algorithm-1: protocol for query packet, query response and download session

3.3. Summary

In this chapter, we have identified the problems that cause congestion in the network. We measured query success and query delay for the system. We observed that the system performance degrades more than 20% because of the downloading effect. We discuss the possible causes of this performance degradation. We observed that the performance is degraded due to more number of unstable link generated in the network after download session starts. We noticed that more number of packet loss happens during the downloading session. We also observed that RTT graph have similar trend to loss graph. That means losses due to congestion. We proposed CAON, a Congestion Aware system for P2P MANET system. We introduce a possible design for improving congestion in the network. In the design, we mainly avoid the RED link and we denied downloading when the link color is RED. We have local feed back mechanism for the no-neighbor situation. We hope our design will improve the system performance. In the next chapter we will describe our evaluation and results.

We used ns-2 for our simulation that was developed by the VINT project at the University of Californian at Berkley and its extension for mobile wireless contributed by the MONARCH project [44]. We designed CAON for improving RAON performance in a variety of network conditions. In the previous chapter, we observed that RAON suffers a significant congestion problem due to downloading effect. In our simulation we evaluated the performance of our scheme including file download traffic. We provide the results that we obtained, along with a detailed analysis and comparison.

4.1. Simulation set-up

We simulated both systems, RAON and CAON, using random waypoint mobility model. In random waypoint mobility model nodes choose a random destination with a uniform random speed between 0 to *max_speed* and pauses for a fixed amount of *pause_time* after reaching the destination. The pause time we used in our simulation is 30 sec. After the pause time, the node again starts to move another location. We described about the mobility model in our chapter-2 in details. We designed CAON and simulated it in NS2 simulator. We used RAON implementation that was developed by the MOWIN group at Ryerson University. We used the IEEE 802.11 standard MAC layer. The energy model used by CAON is the same as RAON because our research is not energy concern. RAON uses all nodes at the same power level. The Radio propagation model in CAON is the two-

ray-ground model and the transmission power is set to 281.8 mW, which result in a transmission range of approximately 250m. Each node is equipped with an antenna and our antenna is 1.5 m above the ground for every node.

The main goal of our simulation is to investigate the performance of both systems (RAON & CAON) by including file download traffic. We also consider the network configuration in case of performance evaluation. The developed the algorithm is described in Chapter-3, for improving RAON performance under downloaded situation in different MANET scenarios. The system was designed to avoid the possible congestion in the network. Our simulation objective is to study the network performance in P2P MANET system when the system downloads its desired files. We use different MANET scenarios for the fixed network size of 500 x 500 sq.-m. The total number of mobile node that we used in our simulation is 100. The maximum speeds of 2 m/s, 5 m/s and 20 m/s are used to model the movement of pedestrians, non-motorized vehicles and motor vehicles respectively. The total simulation time is 1000 sec and the node pause time is 30 sec. We set up query time out to be 120 sec. The file size that we assigned is 100 bytes. Each node can store maximum 100 files and the maximum number of concurrent files downloaded from a node is 1. The nodes are randomly placed in simulation area that creates topology in MANET environment. We use AODV routing protocol in our underlying network, which is an on-demand routing protocol.

Our system implementation is independent of the underlying protocol and we expect it to work with other routing protocol too. The interface queue (IFQ) is set to 30 packets. Each node at P2P level gets uniform random number between $[0, \text{sim_time}/2]$ to determine when to join the network. No P2P topology is defined in

the system at the beginning of the simulation and a virtual boot server that has the all peer list is attached with the system. If a new node wants to join, it gets all the information about existing peers in the system. The desired node sends request to an arbitrary node for connection. After a node is connected to the overlay network, it periodically updates its information to its neighbor. We chose the update interval to be 30 sec. The number of peers that we used is 10, 20, 30, 40, and 50. The query generation interval is (0, 10) sec. Each query contains a keyword, which is represented by an integer between [0,99], and every keyword is mapped to a set of files. The files located on each node are generated randomly and the number of files and the file sizes depend on the nodes capacity level. Our main goal is to see the congestion level under the download condition for both systems. Hence, we use the same capacity level of all the nodes. The number of files stored at each peer is [0, 100]. The latency thresholds for link coloring scheme that we used are 1 sec and 3 sec and TTL was 31. Energy thresholds are 50 % and 20%.

4.2. Results analysis

CAON builds on the structure of RAON. The main difference between these two systems is RAON didn't consider the download effect for measuring its performance. It just evaluated the impact of different MANET scenarios on overlay network performance like query success & query delay. We have shown in Chapter 3 that there may be a significant impact of downloading activities on peer-to-peer network performance. Moreover downloaded packets have a great impact on network congestion. In this section we present analysis of the simulation results that evaluate the performance of RAON and CAON under file download traffic.

4.2.1. Query forwarding performance

Query success rate

A query is considered to be successful if the query originator receives at least one query hit response before the query time out. CAON and RAON both used the biased random walk for query forwarding to its neighbor. Both RAON and CAON send query to a

(a) Query success rate for speed 20 m/s (dimension 500x500)

(b) Query success rate for speed 5 m/s (dimension 500x500)

(c) Query success rate for speed 20 m/s (dimension 500x500)

Figure 4.1 RAON & CAON query forwarding performance (a. Speed 20 m/s, c. 2 m/s and b. 5 m/s and Dimension 500X500)

high degree neighbor that is connected through the GREEN link. In addition, CAON query forwarding also considers the no-neighbor situation (when no GREEN link with any neighbor) and downloading effect. In the case of RAON, if it finds a no-neighbor condition, it forwards the queries any way and RAON doesn't consider the downloading effect, as it doesn't include any downloading option during its performance measurements. We measured the query performance with the presence of downloading traffic. RAON uses color of the immediate links in the forwarding decision without knowing the status of links farther in the path. When no neighbor is accessible through GREEN or YELLOW, the RAON node forced its query to travel over the RED link.

In CAON, if a node gets no neighbor situation, it sends local feedbacks to its previous neighbor to advise the situation. We expect the local feedback can improve the query performance. In CAON, if a sender gets local feedback about the no neighbor situation, it avoids sending query to its immediate neighbor even though the neighbor is accessible through the GREEN link. CAON also considers the condition of the download path. In CAON, the color of a query packet undergoes changes according to the link color over which it travels to the destination. So the query originator has the option to make a decision from where it will download. It decides to download from the destination that returns GREEN packet, indicating that all the links along the query path is GREEN.

We measured the query success rate for RAON and CAON, under downloading situation. Figures 4.1(a), (b) & (c) show the query success rate of RAON and

CAON with the presence of download traffic. From the above graphs, we can see that CAON shows higher query success rate than RAON with download traffic. If we compare RAON and CAON from the above graphs, we can see that CAON performs on average 27 % better than RAON for 20 m/s (figure 4.1 (a)), 21% better for 5 m/s (see figure 4.1 (b)) and 11 % better for 2 m/s (figure 4.1 (b)). The performances of both systems degrade with increasing number of peers. By increasing the number of peers the network becomes more unstable, which increase the chance of query loss and consequently drop the query success rate. CAON always takes stable link for it's downloading purpose. RAON system doesn't have any idea about downloading. So the performance went down due to congestion.

Query response delay

A query-hit delay is the time taken by the originator to receive the query-hit response. We measured the query-hit delay for RAON and CAON under downloaded option. We plotted three figures for three speeds with increasing number of peers. We can observe from the figure -4.2 (a), (b) and (c), Query delay improve in CAON case. In figure (a) for speed 20 m/s, CAON improves is 2 sec. This performance improvement happened due to two feed back mechanisms. CAON have local and external feed back mechanisms whereas RAON doesn't have. No neighbor situation that is happened in RAON is very important to take care of it. We included an algorithm for this situation in our design.

(a) Speed 20 m/s

(b) speed 2 m/s

(C) Speed 5 m/s

Figure 4.2 (a.b.c): Query hit delay Vs. Number of peers

From the above figure 4.2, we can see that delay is less for less number of peers.

The delay increases as the number of peers increase. By increasing the number of

peers, the network becomes more congested, which increase the chance of query loss. CAON performance is better than RAON because of CAON's awareness about congestion. The primary reason of low delay in case of CAON is attributed to the more informative download site selection, where download sites are selected that are reachable through paths composed of GREEN links.

Average number of unstable Link used

We measured the average number of unstable link generated in both systems in the whole simulation time (see figure-4.3). A link is *unstable* if its color is RED.

Figure 4.3: Average number of unstable links generated in RAON & CAON, (a) for speed 20 m/s and (b) for speed 5 m/s.

We can observe from the above graph that RAON uses more unstable links than CAON in download condition. CAON tends to avoid RED links more aggressively than RAON. It further avoids using RED links for download traffic. From the above figure 4.3 (a) & (b), we can observe that CAON system shows less number of

unstable links compared to RAON in downloaded situation. Less number of unstable link means, the better performance of the network. We did not plot the graph for speed 2 m/s, because it differs very little with 5 m/s. CAON performance is better due to awareness of congestion in the network.

Packet loss & RTT relation

We measured the packet loss tendency of both RAON and CAON for the whole simulation time. The figure 4.4 shows that RAON & CAON packet loss tendency. We can observe from the figure 4.4, the packet drop is less at the beginning of simulation in the both systems. As soon as simulation increase, the network gets overloaded that increase the chances of packet losses. Therefore, if a node is congested, any packet that originates or destines at that node is at risk of being dropped. CAON performance is better in this case too. CAON takes care of congestion due to heavy download traffic and take better decision for sending a query to its neighbor by considering the link condition in advance. CAON nodes deny download if it gets the query hit through an unstable link. Both systems packet drops gradually increase with time and the maximum packet drop occurs in the region of 550 sec to 925 sec. CAON performance is better due to awareness of congestion.

Figure 4.4: Packet loss vs. simulation time for RAON and CAON (Peers 10, Speed 20 m/s, Dimension: 500x500)

Figure 4.5: RTT vs. simulation time

We also measured the RTT for the system to understand the packet loss behavior. We observed that RTT graph follows the same trends of packet loss graph. That means, the time when the RTT value is high the packet loss is also high. This indicates that download activities was higher at that time. It is also observed that RTT value is higher in RAON compare to CAON.

Duplicate Acknowledgement

When packet being lost or delivered out-of-order, duplicate Acks sent by the receiver. Packet lost or out-of-order is the product of congestion in the network. An out-of-order packet is not a real issue for p2p system, since message can be processed despite of their order of arrival. We measured the number of duplicate Acks with the increasing number of peers.

(a) speed 5 m/s

(b) speed 20 m/s

Figure 4.6: Total number of duplicate Acks for RAON and CAON in speed 20 m/s and 5 m/s.

The above graphs show that there are duplicate Acks in RAON as RAON has higher packet loss tendency that we have already seen in our previous analysis. Duplicate Acks have rising trends with the increase of number of peers. By increasing the number of peers, the network is more congested, which increase the chances of more packet losses. This effect causes more duplicate Acks in RAON than CAON.

4.2.2. Network-level analysis: AODV route failure notification

In the previous section, we have evaluated and analyzed RAON and CAON congestion level using data collection at the p2p level. In this section we would like evaluate the RAON and CAON performance by comparing the network level details. A path consisting of multiple physical hops in the underlying network usually forms a virtual link between two overlay nodes. If a single mobile node along the path moves from its current position, the virtual link may break and causes AODV to start a recovery process by invoking local repair mechanism. Figure 5.2 shows the situation of AODV route failure with varying speed. As speed increase the route failure also increase in the both systems. Route failure may lead to congesting of a node. If a node detects a link break in an active route, it sends out a RERR message to up stream neighbors and starts local repair. The intermediate node queue the packet up to the route repair, when a new route generate, the intermediate node flush all the packets through the new route. So duplicate packet received by the destination. When the queue is full packet may be dropped. This increases the retransmission of packets. Figure -4.7 shows that the two systems suffer approximately the same number of route failures in all MANET scenarios for a specific speed. With the increase of speed the route failure also increase due to more number of link breakages. A link failure is remains latent in case of low speeds and the link is not along the query-forwarding path.

(a) Speed 20 m/s

(b) AODV route failure for speed 5 m/s

Figure 4.7: AODV route failure for RAON and CAON

As soon as the speed increase, the route failure increase, in this case query loss increase. Still CAON route failure is less compared to RAON because of better protocol that CAPON

4.3. Average number of messages generated in RAON and CAON.

Type of message	RAON	CAON
All forwarding messages (Query, query hit, ping and pong)	8734.42	7391.34
Token update +ACK	2834	2134
Connection	179.4	173.2
Probe +ACK	87.3	87.3
AODV route failure (average & speed 20 m/s)	2579.6	2686.6

Table 4.1: Average number of message generate

4.4: Performance evaluation under Manhattan Mobility Model

We evaluated our system under Random way point Mobility model where nodes can move anywhere any time any direction. There is no restriction of node movement. In Manhattan model, nodes are restricted to move within a map. Manhattan mobility model is designed for moving a node in urban area where the road maps consists of horizontal and vertical lines. Detail about Manhattan mobility model is described in chapter-2.

4.4.1. Simulation setup

We introduce the Manhattan model to emulate the movement pattern of mobile nodes on streets defined by map.

The map we used in our simulation can be shown as follows:

HOR_STREET_NUM<num_of_horizontal_street>

VER_STREET_NUM<num_of_vertical_street>

LANE_NUM<overall_num_of_lanes>

LANE<street_id><lane_id><direction><start_xo><start_yo><end_xo><end_yo>, total_number_of_crosspoints_in_this_lane><vmin><vmax>

CROSSPOINT

<crosspoint_id><street_id><lane_id><direction><position_x><position_y>

We used the mobility generator that is developed by ProTest Lab, university of Southern California. We used version important – 1.0 – beta and released on Feb 2004. In MH model, several horizontal and vertical streets co-exist in the simulation field and mobile nodes are moving on the lanes of the streets. For each street it has several lanes in both directions. However, the vertical and horizontal street is crossed with each other at the cross points. At the cross point the mobile node can move ahead, turn left or turn right. The probability of moving on the same street is 0.5, the probability of turning left is 0.25 and the probability of turning right is 0.25. The maximum velocity of a node is 20 m/s and minimum velocity of the node is 0 m/s. We didn't use 5 m/s for Manhattan mobility model because it is too slow in urban street area. The mobile node is allowed to move along the grid of horizontal and vertical streets. Each street has two lanes for each direction (NS for Vertical Street & EW for Horizontal street). We set the acceleration speed to 10 % of max velocity. In Manhattan mobility model, nodes

were placed on the streets randomly in both directions initially. If the node moves beyond the boundary of the area it is re-inserted at the beginning position in a randomly chosen lane in the area.

4.4.2. Results analysis

Before analyzing the performance of our system under Manhattan mobility model, we define few terms.

- **Link Duration (LD):** The link duration is the time interval between the time when the link is created and the time when the link breaks.
- **Path Duration (PD):** The path duration is the interval between the time when the path is set up and the time when the path is broken. Since each path is composed of several links, it is likely that the behavior of path duration will be determined by the behavior of link duration.
- **Warm-up time:** The time after which the protocol performance remains relatively stable as the ‘warm-up time’.
- **Spatial dependence:** The spatial component describes *where* the mobile entity is moving
- **Temporal dependence:** The temporal component describes *when* an entity is moving and at which speed.

Query success rate

Figure- 4.8 shows the performance graph of both systems under MH mobility model. We observed that the system performance degrades under Manhattan mobility Model. The performance degradation hits almost 50% due to downloading effect. We

chose speed only 20 m/s as Manhattan mobility works in urban areas. So other speed will be too slow. The node movement is the basis of street Map.

If you analyze the above figure 4.8, we observe that the query performance curve for RAON with download and CAON is pretty much close to each other

Figure 4.8: Manhattan mobility model query success rate vs. number of peers

This indicates that CAON performance is still better than RAON under downloaded condition.

Query Delay

Figure: 4.9: Query delay for RAON & CAON under MH

We can observe from the above figure 4.9 that CAON query delay is less with compare to RAON in download option.

Packet loss & RTT relation

We also plotted the packet losses vs. simulation time graph by adding Manhattan mobility model instead of random waypoint mobility model. We observed that packet loss is more in case Manhattan Mobility Model. It is clear that packet losses happen due to congestion of the network and some additional factors. We will discuss the additional causes later on in details that create more packet losses. We can clarify the causes of these packet losses by plotting the RTT vs. simulation time (see figure 4.11). We can observe that RTT is high at the time of 250 sec and the packet losses also start high at this time (see figure 4.10). So we are getting a co-relation between packet losses and RTT. It is clear that congestion is happened in the network.

Figure 4.10: Packet losses vs. simulation time (Manhattan Mobility model)

Figure 4.11: RTT vs. simulation for RAO & CAON

CAON improves the network performance by reducing the packet loss.

For a given spatial dependence, if a mobility pattern has a high relative speed, the nodes might move out of range more quickly. Thus an already existing link and the paths keeps remain stable for a relative shorter duration. This may lead to more packets being dropped due to path breakage. So we can conclude that the Manhattan mobility pattern with high relative speed and low degree of spatial dependence lead to a worse performance.

Duplicate Acknowledgement

Figure 4.12: Duplicate Acks. Vs. number of Peers under MH mobility model, Speed 20 m/s.

We plotted the duplicate Acks vs. number of peers for our system by adding Manhattan mobility Model. We can observe from the figure-4.12, the duplicate Acks increases in RAON due to the download effect. CAON provides some improvement over RAON. If we consider peers 30, CAON have almost 27 % improvement compared to RAON. It is clear that CAON performance is better than RAON in the case of download activities.

Additional causes for packet loss in Manhattan Mobility model

We observed that Manhattan mobility model gets more packet loss than Random way point Mobility Model. We can define the causes according to the research that conducted by authors in [41]. The authors proved that the average link duration (LD) and path duration (PD) is less in Manhattan Mobility model Compare to Random way points mobility model. LD and PD depend on relative speeds of the nodes. These values are lower for Manhattan because of high relative speeds. The mobility pattern influences the connectivity graph that in turn influences the protocol performances. It was written by the author that the AODV had a higher through put and lower overhead for random waypoint mobility model than Manhattan mobility model. With the same degree of spatial dependency, between Manhattan and random way point mobility, high relative speed (Manhattan) means lower link duration and corresponding lower path duration, which is result in lower throughput and higher

Performance comparison of CAON in Random way points mobility and Manhattan Mobility.

Criteria	Random way point Mobility Model	Manhattan Mobility model
Spatial dependence	No	No
Number of unstable link	Less	More
Number of stable link	More	Less
Total number of packet loss	Less	More
Total number of query loss due to lack of neighbor.	Less	More
Application	General	Urban

Table 4.2: Performance comparison of CAON in both mobility models.

The performance of CAON under Manhattan mobility model is not expected level as random waypoint mobility model. Manhattan mobility model can be used in urban area where a pervasive computing service between portable devices is provided. Map is used in this model. The map is composed of horizontal and vertical streets where the streets have lanes, and the nodes can turn right or left.

4.3. Summary

In this chapter we presented the simulation result of RAON and CAON under random way point mobility model and Manhattan mobility model. Both systems simulated under down load session. We discovered that CAON is able to achieve the performance gain in terms of query success, query delay and packet loss under both mobility models. We observed that Manhattan mobility model causes more packet drop than Random waypoint mobility model.

Chapter-5

Conclusion and Future work

Conclusion

In our design of a P2P MANET system, we considered the effect of wireless medium used as well as congestion in the system. Due to congestion, the system does not achieve its desired performance. We used RAON as our base for P2P MANET system that is developed by the MoWin group of Ryerson University. We introduced download option in RAON and measured its performance. We observed significant performance degradation due to download traffic in the network.

There are many factors that may cause congestion in the network. We select RTT for identifying the congestion level of the system as RTT measures the stability of a link. We found that when congestion occurs, then it becomes the dominant factor in increasing the user response time and packet losses. We observed significant packet losses in RAON and identified the possible cause of the packet loss. We analyzed different performance measures under variety of situations and concluded that RAON faces congestion problem due to heavy download traffic in the network. We observed similar pattern of packet loss and RTT graphs for the same system. We also plotted the total number of unstable link generated in the whole simulation time. We observed that the number of unstable links increase during download times. We modified the RAON design to propose CAON for improving the network performance. We modified the query forwarding algorithm of RAON to avoid forwarding queries to the neighbors that are expected to

forwarding along RED links. This is achieved by incorporating a local feedback mechanism from a node to its neighbors. A node x receives a local feedback from its neighbor y if node y is connected to all other neighbors with RED links. The node x then uses this feedback from y to avoid sending query to y . We also proposed a heuristic for selecting download sites. We select a download site that is connected through a stable path. We proposed an algorithm of coloring query according to the link it traverses. A query becomes RED when it traverses at least a RED link. The query response acquires the color of the corresponding query. Thus, the querier node avoids download sites generating RED query responses.

We simulated the performance of CAON under variety of network and mobility situations. Our simulation results show that CAON performance is better as compared to RAON. It achieves on average 20% more query successes after including download traffic. It also shows less amount of packet drops as compared to RAON. Both RAON and CAON generate the same amount of AODV route requests, which shows that CAON does not introduce an overhead in the network. We also evaluated our system for geographically restricted movement by changing mobility model from random waypoint to Manhattan. We noticed that the overall network performance in case of Manhattan is not as good as in case of random waypoint mobility model. But CAON shows low query delay and low packet losses as compared to RAON even in case of Manhattan. The above performance gain happened due to congestion awareness of CAON. In summary, we made following contributions in this thesis. First we identified the factors that cause RAON performance degradation. Second, we analyzed the cause of the problem. We came to the conclusion that this performance degradation is primarily caused by

congestion. Third, we developed a system called CAON that improves the query performance by avoiding RED links in query forwarding algorithm and also by selecting download sites that are connected through stable paths.

Future work

In future, further investigation can be made on Gia's flow control mechanism. We identified following factors that can contribute to further performance improvements: (a) RED links get less tokens than Green links, and (b) token distribution is more dynamic. By assigning less tokens to a neighbor connected through unstable links reduce the chances for the neighbor to make the system unstable.

More investigation can be made on underlying network congestion control by bringing the information from application level to network level [see paper 12]. More research is needed to better understand the complex interactions between TCP and lower layer protocols when used over MANET and find solutions for the problems caused by these interactions. We observed from Manhattan mobility model that Performance decrease due to restricted movement of mobile nodes. The results suggest that more aggressive cache management protocols are needed to counter the effects of mobility, such as adaptive route cache timeout, negative information or signal strength information.

References

- [1] *G Lau, M. Jaseemuddin, and G. Ravindran*, RAON: A P2P Network for MANET, Proceedings of the Second IEEE/IFIP International Conference on Wireless and Optical Communications Networks (WOCN 2005), Dubai, UAE, March 6-8 2005.
- [2]. *Bryan J. Hogan, Michael Barry, sean McGrath* “ Congestion Avoidance Routed Ad Hoc Networks” wireless Access research group, university of Limerick, Ireland
- [3]. *M.M.El-Sayed, Obazan, M. Jasemuddin* “ Performance evaluation of TCP in Mobile Ad hoc networks” The second international Conference on Innovation in Information technology (IIT’05)
- [4]. *Raj Jain and K.K.Ramakrishnan*, “ Congestion avoidance in Computer Networks with a Connectionless Network layer: concepts, goals and Methodology” IEEE , Proceedings Computer Networking Symposium, Washington, D.C. April 1998
- [5]. *Matei Ripeanu and Ian Foster* “ Mapping the Gnutella Networks” IEEE the Internet computing January- February 2002
- [6]. *Duc A. Tran and Harish Raghavendra* “ Routing with Congestion Awareness and Adaptivity in Mobile Ad hoc networks. Multimedia and collaborative Networking Group, Department of Computer science, university of Dayton.
- [7]. *Qin Lv, Sylvia Ratnasamy, Scott Shenker* “ Can heterogeneity make Gnutella Scalable?” <http://www.cs.rice.edu/Conferences/IPTPS02/165.pdf>

- [8]. *Seoung-Bum Lee and Andrew T. Campbell* “ HMP: Hotspot Migration Protocol for Mobile Ad hoc network” COMET Group, Department of Electrical Engineering , Columbia University, New York
- [9]. *Leonardo B. Oliveira, Antonio A.F. Loureiro, Hao Chi Wong* “ Evaluation of Peer to peer Network Content Discovery Techniques over Mobile Ad hoc networks” federal University of Minas Gerais, Computer Science department Brazil
- [10]. Kai Chen and Klara Nahrstedt “Exact : An explicit Rate- based flow control frame work in MANET” department of computer science, university of Illinois
- [11]. *Alexander Klemm, Christoph Lindemann, and Oliver P. Waldhorst* “ A special Purpose peer to peer File Sharing system for Mobile Ad hoc networks” department of computer Science, university of Dortmund, Germany, IEEE 2003
- [12]. *Kalppesh patel, sridhar Iyer, Krishna Paul* “ RINGS: Lookup service for Peer to peer Systems in Mobile Adhoc networks” Kanwal Rekhi School of Information Technology, IIT Mumbai, India
- [13]. *Selma Boumerdassi, Eric Renault and Anne Wei* “Impact of Mobility Environment on the QoS in Ad hoc networks” CNAM- Laboratory CEDRIC France
- [14]. *Byung-Gon chun, Mary Baker* “ Evaluation of packet Scheduling Algorithms in Mobile ad hoc networks” Computer science Department, stanford University, California.
- [15]. *Yatin Chawathe, Sylvia Ratnasamy, Lee Breslau,* “ Making Gnutella Like P2P systems Scalable” AT &T Labs-Research, Intel Research, SIGCOMM’03
- [16]. *Alexander Klemm, Chrishtoph Lindemann and Oliver P. waldhorst* “A special-purpose peer to peer file sharing system for Mobile Ad hoc networks” university of Dortmund, department of computer science

- [17]. *Qi He, Mostafa Ammar, George Riley, Himanshu raj, Richard Fujimoto* "Mapping peer Behavior to packet level details: A framework for peer- level simulation of peer to peer Systems" Networking and Telecommunication Group, College of computing, Georgia Institute of Technology, Atlanta
- [18]. *Gavin Holland and Nitin Vaidya*, "Analysis of TCP performance over Mobile ad hoc networks" department of computer science, Texas A 7 M university Feb 15, 1999
- [19]. *Feng wang, Yongguang Zhang* " improving TCP performance over mobile Adhoc networks with out of order detection and response" MOBIHOC 2002
- [20]. *Chonggang wang, Bo Li* " Peer to peer Networks: A survey" Department of Computer science" the Hong Kong University of Science and Technology, Hong kong April 2003
- [21]. *Rudiger Schollmeier, Ingo gruber and Micheal Finkenzeller* " Routing in Mobile ad hoc and Peer to peer Networks: A comparison" Siemens corporation technology, Siemens AG.
- [22]. *Gerd Kortuem* " Proem: A peer to peer computing platform for Mobile Ad hoc networks" Wearable Computing Laboratory, Department of computer Science, University of Oregon.
- [23]. *Zhang Zhenjie, Yu Feng, Yang Xianoyan* " Report on Unstructured Network in peer to peer system" computer science department, national university of Singapore.
- [24]. *Miguel Castro, Manuel Costa and Antony Rowstron*, "Peer to peer overlays: Structure, unstructured or both?" Technical report, Microsoft Corporation.

- [25]. *Qin Lu, Pei cao, Edith Cohen, kai Li, Scott Shenker* “ search and Replication in Unstructured Peer to peer Networks” department of computer science, Princeton University
- [26]. *Vicent Cholvi, Pascal Felber, Ernst Biersack* “ Efficient search in Unstructure peer to peer networks” Institut EuRECOM
- [27]. *Dimitrios Tsoumakos, Nick Roussopoulos* “ Analysis and comparison of P2P search Methods” department of computer science, university of maryland, USA
- [28]. *Gang Ding and Bharat Bhargava* “ Peer to peer file sharing over Mobile Ad hoc network” department of computer Science, Purdue University
- [29]. *Y. Charlie Hu, saumittra, M.das, and Himabindu Pucha* “ Exploiting the synergy between Peer to peer and Mobile ad hoc networks” USENIX Association, 9th workshop on Hot topics in Operating systems.
- [30]. *Lionel M. Ni and Yunhao Liu* “ Efficient peer to peer overlay Construction” IEEE international Conference on E-Commerce Technology for Dynamic E-Business
- [31]. *Anwitaman Datta* “ MobiGrid: Peer to peer overlay and Mobile Ad-Hoc Networks rendezvous – a data management Perspective. Distributed information system laboratory, Switzerland
- [32]. *Xiao Shao* “ knowledge resource distribution by peer to peer networks” department of computer science, Tokyo institute of technology, Tokyo, Japan
- [33]. *Saad Biaz, Nitin H. Vaidya* “ Distinguishing Congestion Losses from wireless Transmission Losses: A negative Result” IEEE computer society 1998

- [34]. *Asis Nasipuri and castaneda* “ Performance of Multi-path Routing for On demand Protocols in mobile Ad hoc Networks” Kluwer networks Academic publishers 2001
- [35]. *Anirban Mondal, Kazuo Goda, Masaru Kitsuregawa* “ Effective load balancing of peer to peer systems” DEWS 2003
- [36]. *Elias Leontiadis, vassilios V. dimakopoulos, Evaggelia pitoura* “ Cache updates in a peer to peer Networks of mobile Agents” Departments of computer science, university of Ioanniana.
- [37]. *R. Hekmat*, “Fundamental Properties of Wireless Mobile Ad-hoc Networks,” *KiVI Telecommunicatieprijs*, Netherlands, March 2004.
- [38]. *S. Corson and J. Macker*, “Mobile Ad Hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations,” IETF RFC 2501.
- [39]. *Ian D. chakeres, Elizebeth M. Beldings Royer* “ AODV Routing protocol implementation” University of California
- [40]. *Fan Bai, Narayanana sadagopan, Ahmed Helmy* “ user manual for MANET mobility tools generators in ns-2 simulator” USC, Department of electrical & computer engineering.
- [41] *Fan Bai, Narayanana sadagopan, Ahmed Helmy* “the IMPORTANT framework for analyzing the impact of mobility on performance of routing protocols for Ad hoc networks” Department of Electrical & computer engineering, USC, USA
- [42]. *J.A Dongyu qui & R. srikant* “ Modeling and performance analysis of bit-tolerant like peer to peer network” The bit tollerrent P2P file sharing system:” Coordinated science laboratory, university of illions at urbana, urbana, IL.
- [43] <http://www.isi.edu/nsnam/ns/>

RAON & CAON simulation result for dimension 1000 m x 1000 m:

Appendix A: Query success rate for RAON & CAON

Packet losses in RAON & CAON

RAON & CAON AODV Route failure

RAON & CAON Query performance

(Manhattan Mobility Model)

Total number of unstable link In MH

Packet losses under Manhattan Mobility Model

② SL-10-43