Adaptive Anycasting in Delay Tolerant Networks

Developed by

Nadir Shah
(264 – FAS/MSCS/F05)

Fazl-e-Hadi
(23 – FAS/PHDCS/S05)

Supervised by
Dr. Syed Afaq Hussain
Dr. Mehboob Yasin

Department of Computer Science
Faculty of Basic & Applied Sciences
International Islamic University, Islamabad.

(2007)

بسم الله الرحمن الرحيم

International Islamic University, Islamabad

Date: 02-06-2007

Final Approval

It is certified that we have read the thesis titled **"Adaptive Anycasting in Delay Tolerant Networks"** submitted by Mr. Nadir Shah Reg. No. 264-FAS/MSCS/F05 and Mr. Fazl-e-Hadi Reg. No. 23-FAS/PHDCS/S05 which, in our judgment this thesis is of sufficient standard to warrant its acceptance by the International Islamic University, Islamabad for the award of MS in Computer Sciences.

Committee

External Examiner

Dr. Abdus Sattar
H.# 143, St # 60, I-8/3,
Islamabad, Pakistan.

Internal Examiner

Prof. Dr. Sikander Hayat Khayal
Head,
Department of Computer Science,
International Islamic University, Islamabad, Pakistan.

Supervisors:

Dr. Syed Afaq Hussain
Head,
Dept. of Electronics Engineering,
International Islamic University, Islamabad, Pakistan.

Dr. Mehboob Yasin
Professor,
Dept. of Computer Science,
COMSATS Institute of Information Technology, Islamabad, Pakistan.

A dissertation submitted to the
Department of Computer Science,
International Islamic University, Islamabad,
As a partial fulfillment of the requirements
For the award of the degree of
MS in Computer Science

Dedicated to our parents whose love, affection, and prayers enabled us to achieve this status

DECLARATION

We, hereby declare that this research, neither as a whole nor as a part, has been copied from any source. It is further declared that we have developed this research entirely based on our personel efforts under the able guidence of our supervisors, Dr. Syed Afaq Hussain and Dr Mehboob Yasin. If any part of this thesis is proved to be copied or reported at any stage, we accept the responsibility to face the subsequant consequences. No part of this work inscribed in this thesis has either been submitted to any other university for the award of degree/qualification.

Nadir Shah
264-FAS/MSCS/F05

Fazl-e-Hadi
23-FAS/PHDCS/S05

Project in Brief

Project Title: **Adaptive Anycasting in Delay Tolerant Networks**

Objective: The primary goal of the research was to study anycasting for a specific type of DTNs, where the mobile nodes are sparsely distributed; communicating via low radio range, experience frequent and long duration partition and end-to-end connectivity may not be present between source and destination at the time of message origination. We also proposed RBF scheme for anycasting in DTNs, which consider the path length as well as the number of receivers in forwarding the anycast bundle to the next node. We study the performance of the new developed algorithm.

Organization: Department of Computer Science,
International Islamic University, Islamabad.

Undertaken By: Nadir Shah (264-FAS/MSCS/F05)
Fazl-e-Hadi (23-FAS/PHDCS/S05)

Supervised By: Dr. Syed Afaq Hussain
Head,
Department of Electronics Engineering,
International Islamic University, Islamabad.
Dr. Mehboob Yasin,
Professor,
Department of Computer Science,
COMSATS Institute of Information Technology, Islamabad.

Language: C++, OTCL

Tool Used: Network Simulator NS-2

Operating System: Red Hat Enterprise Edition

System Used: Pentium III

Date Started: September 2006

Date Completed: April 2007

Table of Contents

CHAPTER 1

INTRODUCTION

1 Introduction

This section provides the background of different areas that are related to this research.

1.1 Internet

TCP/IP protocol suit is used for communication among the devices on the Internet. These protocols are used for routing data and ensuring the reliability of message exchange.

Communication on the Internet is based on *packet switching*. A message that is to be transmitted is divided into blocks called Packets. Packets travel independently from source toward destination through network of links connected by routers. The source host, routers and destination host are collectively called *nodes*. Each packet of same message can take a different path through the network depending upon link availability. If one link is not available, packets take another link. Each packet has two parts

- Payload part- contain application-program user data
- Header part – contains addresses and control information that determine how the packet is switched from one node to another node on the path.

The packets of same message may arrive out of order. It is the responsibility of destination transport mechanism to arrange the packets in correct order. Thus communication on the Internet is based on some of the following assumption [1]:

- ***Continuous, Bidirectional End-to-End Path***: A continuously available bidirectional connection between source and destination to support end-to-end interaction.
- ***Short Round-Trips***: Small and relatively consistent network delay in sending data packets and receiving the corresponding acknowledgement packets.
- ***Symmetric Data Rates***: Relatively consistent data rates in both directions between source and destination.
- ***Low Error Rates***: Relatively little loss or corruption of data on each link.

Messages are moved through the Internet by *protocol layers*, a set of functions performed by network nodes on data communicated between nodes. *Hosts* (computers or other communicating devices that are the sources or destinations of messages) usually implement at least five protocol layers, which perform the following functions: *Application Layer:* Generates or consumes user data (messages).

Transport Layer: Source-to-destination (end-to-end) segmentation of messages into message pieces and reassembly into complete messages, with error control and flow control. TCP is transport layer protocol.

Network Layer: Source-to-destination routing of addressed message pieces through intermediate nodes, with fragmentation and reassembly if required. On the Internet, the Internet Protocol *(IP)* is used.

Link Layer: Link-to-link transmission and reception of addressed message pieces. with error control. Common link-layer protocols include Ethernet for Local-Area Networks (LANs) and Point-to-Point Protocol (PPP) for dial-up modems or very high-speed links.

Physical Layer: Link-to-link transmission and reception of bit streams.

1.2 Mobile Ad hoc Networks

Ad hoc networks are wireless networks where fixed infrastructure is not available. Each node can act both as host and router. If nodes are in communication range then they communicate with each other directly. If the nodes are not in communication range then other nodes that lie between source and destination, act as router and forward the packet to destination. The nodes in ad hoc network may be mobile, called Mobil Ad Hoc Networks (MANETs). These networks have the following characteristics

- High error rates compare to wired networks
- Frequent topology change

Keeping in view these unique characteristics, different protocols have been developed. These protocols are classified into following two types

1.2.1 Proactive Routing Protocols

These are routing protocols which try to maintain always up to date entries in routing table for every possible source and destination. The advantage of these protocols is that when data packets are generated, they are transmitted according to routing tables entries. That is transmission occur without delay, due to maintainability of upto date routing table entries. These protocols are suitable for wired networks and ad hoc networks where mobility is low.

But it has a disadvantage that in case of high mobility a lot of traffic will be generated to maintain up to date routing table entries due to frequent topology changes. So it is not suitable to networks where mobility is high.

Some of proactive routing protocols are given

- Destination Sequence Distance Vector (DSDV) routing protocol [2],
- Optimized Link State Routing (OLSR) [3, 4]
- Topology Broadcast based on Reverse-Path Forwarding (TBRPF) routing protocol [5, 2]

1.2.2 Reactive Routing Protocol

In these routing protocols, routes are determined on-demand. When a node wants to transmit the data packets, it initiates the route discovery process to the destination. In this way it reduces control traffic. Therefore it is best suited for network with high mobility. However its data transmission is more than that of proactive routing protocol due to route discovery for data packet on-demand. Some of reactive routing protocols are:

- Ad-hoc On-demand Distance Vector (AODV) [127],
- Dynamic Source Routing (DSR) [7]
- Temporally Ordered Routing Algorithm (TORA) [8]

1.2.3 Hybrid Routing Protocol

There are also hybrid protocols, like Zone Routing Protocol (ZRP)[9] by combining both reactive and proactive protocol.

The objective of these protocols is

- To minimize power consumption in order to increase life time of network.
- Selecting routes which remain stable for longer time in order to minimize transmit ion delay, increase throughput and increase efficiency.
- Fair forwarding responsibilities among the nodes.

1.3 Dynamic Source Routing protocol:

Dynamic Source Routing (DSR) [7] is reactive routing protocol. It means the routes are discovered on-demand. When a source node S wants to send packet to destination node D, if the source node does not have route to the destination then source node S broadcast RREQ to the destination D. Each node receiving RREQ appends its identifier to RREQ and broadcast RREQ. Node receiving same RREQ, which has been forwarded by it already, again would discard it. Thus broadcasting is controlled.

If two or more than two nodes transmit to a particular node at the same time then there is a chance of collision. This process is repeated at every node. But when intended destination node D receive the RREQ, it does not forward RREQ i.e. broadcasting is stopped. The intended destination D sends the route reply (RREP) to the source node S for first RREQ on the path obtained by reversing the path in received RREQ. The RREP also contain the route from the source node S to destination node D, the path on which RREQ reached from source node S to destination node D. This is done if the links are guaranteed to be bi-directional. However, if the links are unidirectional then the destination node D also have to initiate the route discover similarly for source node S. In this case RREP is piggybacked in the RREQ from D for S.

Upon receiving the RREP, source node S stores the path included in RREP in caches. When node S sends data packet to node D, it includes that path from caches in packet header. Due to this, it is called source routing. Therefore the packet header size increases as path length increase. Every node receiving data packet examines this source route in the packet. According to source route in data packet header, the intermediate node forwards to the next node.

Several optimizations have been proposed by researchers

- Every node store in cache the routes learnt by any mean. This can be done by forwarding the RREQ or overhearing the data packet.
- When a node come to know that the route to destination node D is broken, it uses another path to node D if exist in cache. Otherwise that node initiates similarly the route discovery for destination node D.
- When a node A receive RREQ for some node D, if node A has route to D , can send RREP to source nod of RREQ.

These above mentioned optimization results in

- Speed up route discovery.
- Reduce propagations of route request: which results in reducing control traffic.

However, there are several route errors can result from these route caching in DSR.

1.4 Delay/Disruption Tolerant Networks

In Internet and ad hoc network, routing is based on the assumption of end-to-end connectivity. There are however many more extreme scenario where end-to-end delays can be several minutes, or even hours, which is much longer than protocols such as TCP can handle, or where a fully connected end-to-end path through the network rarely, or never, exists between two entities wishing to communicate. Such communication networks are commonly grouped as *delay/disruption tolerant networks (DTNs)*. These networks are characterized by:

- ***Intermittent Connectivity:*** If there does not exist end-to-end path between the source and destination at the time of message generation, then packet is forwarded to next hope (according to routing algorithm). The intermediate nodes store the packets and as opportunity arises, the packets are forwarded to next node, according to routing algorithm. In this situation for end-to-end communication using the TCP/IP suite protocols do not work.
- ***Long or Variable Delay:*** Long propagation delays between nodes, variable queuing delays at nodes and long duration partition causes large end-to-end path

non-randomness to help deliver data. Scheduled contacts require time-synchronization throughout the DTN.

1.4.2 Applications of DTNs

Some applications of DTNs are

Deep Space Communication: The round-trip duration is several minuets between planets due to the long distance and the propagation speed of radio waves. Also, in communication with a satellite orbiting Mars, some time the satellite position comes behind Mars (due to continues motion) which causes disconnection. This contact schedule can be obtained from the speed of planets.

Remote/Indigenous Communities For example communication between villages of the reindeer herding population in the north of Sweden suffer from lack of infrastructure such as wired network access, and other means of networking (GSM, satellite, etc) are either unavailable, intermittent, or too expensive to be viable. Through the use of community gateways, and mobile relays, it is possible to use the existing infrastructure of vehicles and human mobility to provide connectivity between remote and more populated areas. Similar problems exist among other indigenous populations

and populations in developing countries [14].

Sensor Networking In sensor networks, a large number of sensors with little storage, power and wireless coverage is deployed in a region to collect data of concern. Therefore sensor network may be mobile and may not be fully connected.. Examples of such scenarios include collection of oceanographic data from tags attached to seals [15] .

Crisis environments In crisis environments, like disaster recovery and military deployment, the people concerning sensors are deployed over an area with limited storage, power and wireless coverage. The field agents need communication with other field agents as well to base station to share the required information [16].

As TCP/IP protocols suit works only when fully connected path from the source to destination exists. As TCP is connection-oriented protocol, so if round-trip time is long then its performance will be poor.

Therefore keeping in view these characteristics, there is need for designing architecture and protocols that can operate efficiently in these networks.

Recently the research community has focused the Delay Tolerant Network. Various forms of DTNs have been exposed. Delay Tolerant Network Research Group (DTNRG) [17] is working on developing and specifying protocols related to DTN. Different routing protocols for different scenarios and different services have been proposed. Next we are going to discuss the DTN architecture and Bundle protocol for DTNs.

1.4.3 The Delay Tolerant Networking (DTN) Architecture

A DTN as an overlay built upon underlying networks, such as wireless ad hoc networks. Its network architecture is based on the asynchronous message (called bundle in DTN) forwarding paradigm [18]. Only those nodes that implement the DTN functionalities to send and receive bundles are DTN nodes, while the others are denoted as normal nodes. A DTN link may contain several underlying links in multiple hops. Fig. 1 depicts a simple example. In the DTN layer, bundles are transmitted in a hop-by-hop store-and-forward manner. Each DTN node has finite-size buffers for bundle acceptance and bundle custody[8].

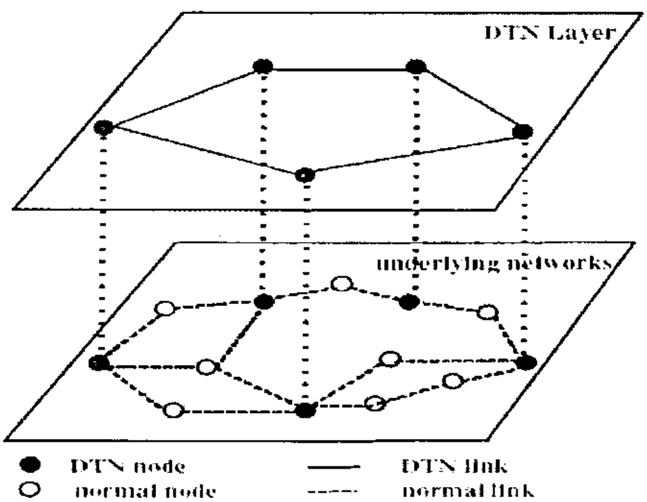

Fig 1. A simple DTN network model.

1.4.4 Bundle Layer (DTN Layer):

The bundle layer specification is included in Internet drafts issued by Internet Engineering Task Force [1]. Bundle Layer operates above transport layer and below the application layer of the network. Only those nodes that implement the DTN functionalities to send and receive bundles are DTN nodes. Below transport layer, protocol of own choice can be used depending on the network environment while a single DTN layer is used across all the networks(regions) of DTNs as shown in figure 2 [1].

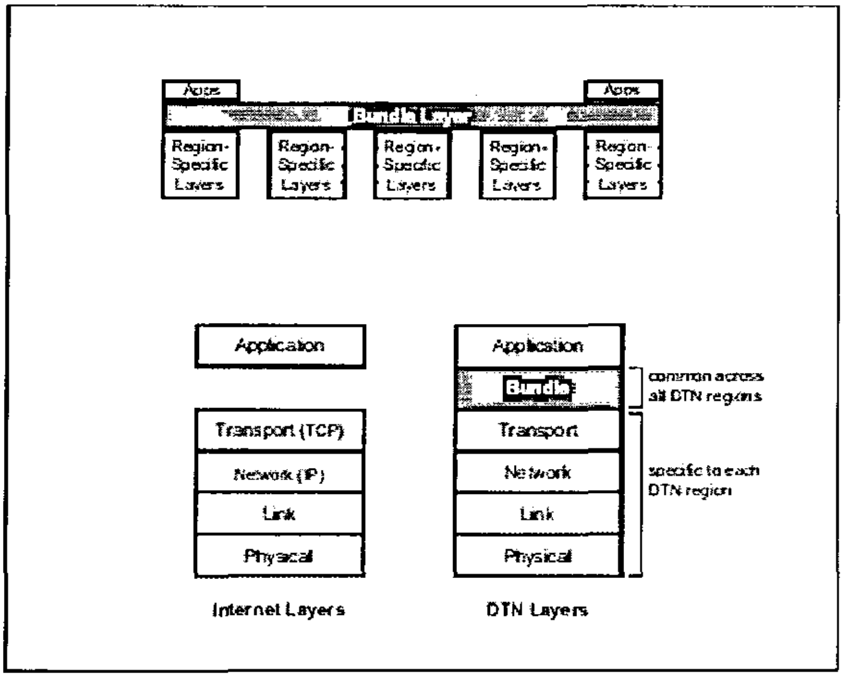

Fig.2 Bundle Layer

Each Bundle has

- User data- source application's data.
- Control information- provided by the source application for the destination application.
- Bundle header-inserted by the bundle layer.

A DTN node (source) sends bundle of arbitrary size. A bundle layer may break a message into fragments. The reassembles of fragmented message is done at the destination DTN node. The encapsulation of bundle is shown in fig.3 [1].

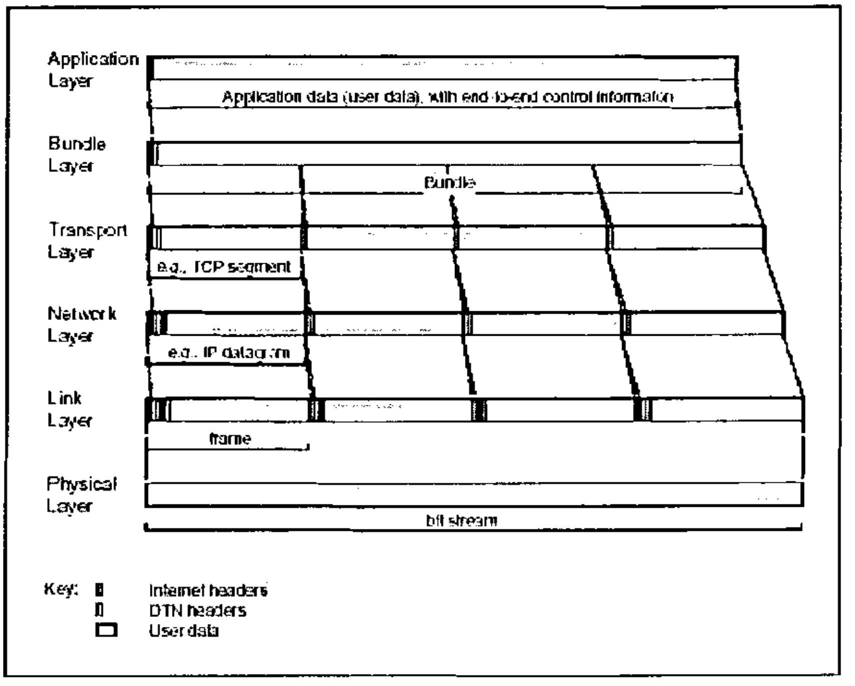

Fig.3 Encapsulation of Bundle

A DTNs node can act as a source, destination, *router*, or *gateway* (or some combination)[11]

Both router and gateway act as a forwarder, the difference is that router forwards the bundle within a single region and gateway forwards the bundle between two or more DTN region. Each DTNs node can support custody transfer. The source, destination and router may need persistent storage if they operate on link that has high delay. But the gateway must be persistent storage so that less packets are dropped when there is more traffic and link availability is low.

There are following six classes of services in bundle layer[11] shown in fig4.

- *Custody Transfer:* Delegation of retransmission responsibility to an accepting node, so that the sending node can recover its retransmission resources. The accepting node returns a custodial-acceptance acknowledgement to the previous custodian.
- *Return Receipt:* Confirmation to the source, or its reply-to entity, that the bundle has been received by the destination application.

- *Custody-Transfer Notification:* Notification to the source, or its reply-to entity, when a node accepts a custody transfer of the bundle.
- *Bundle-Forwarding Notification:* Notification to the source, or its reply-to entity, whenever the bundle is forwarded to another node
- *Priority of Delivery:* Bulk, Normal, or Expedited.
- *Authentication:* The method (e.g., digital signature), if any, used to verify the sender's identity and the integrity of the message.

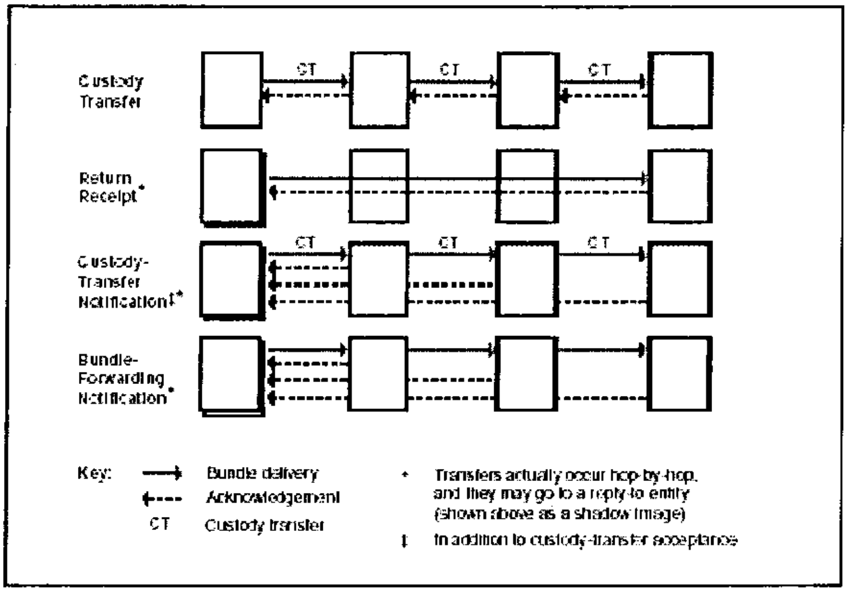

Fig.4 Service classes in Bundle Layer

Another important terminology used in DTN is region. A DTN region is defined as "the networks which have homogenous communication characteristics". Some examples of region are:

- Sensor networks
- Vehicle-to-vehicle networks
- Wireless personal digital assistant (PDA) networks.

DTNs may compose of many regions. Each region has a unique identifier. Therefore region identifier is also a part of node identifier. Since a gateway has the responsibility of forwarding between two or more region. Therefore, a gateway is member of two or more than two regions.

Fig.5 [11] shows example some of the possible regions of the IPN Special Interest Group's Interplanetary (IPN) Internet concept, along with the region namespace hierarchy. The ipn.sol.int region forms the IPN backbone of gateways on long-haul links. As mentioned earlier that each DTN node is identified by region ID and entity (node) ID. Region ID is used for routing among the regions while node ID is used for routing within the region. The region ID is unique and known throughout DTN. Each region has its own addresses for node ID i.e. node ID is bound only within the region. Region IDs have syntax similar to that of Internet's Domain Name System. Therefore, a gateway has two or more than region IDs while each host (source, destination) and router has only one region ID.

The security consideration for DTNs is out of scope of this thesis.

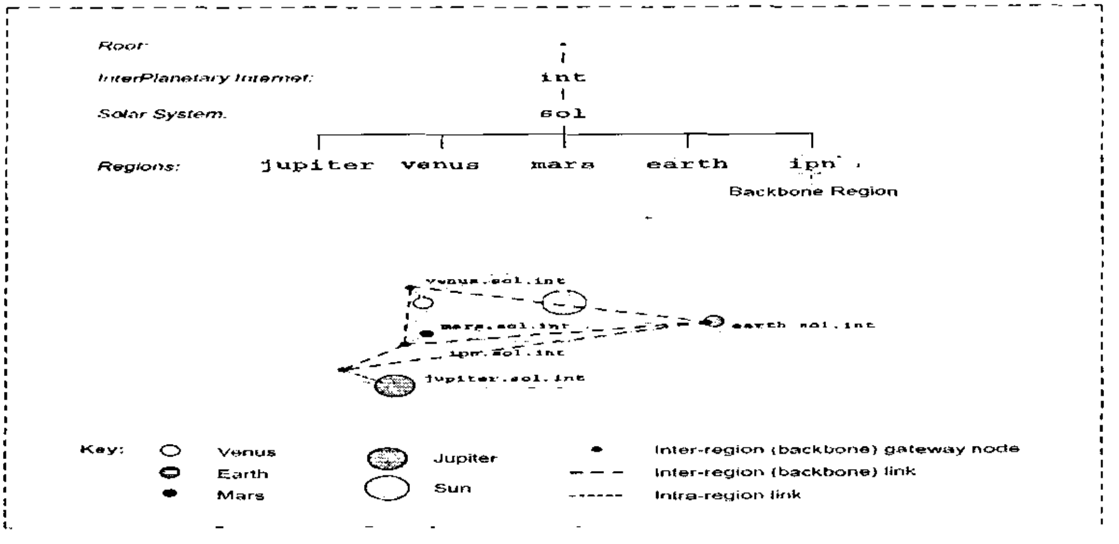

Fig5. showing examples of regions.

1.5 Anycasting in DTNs

Anycast is a service that allows a node to send a message to at least one, and preferably only one, of the members in a group. The idea behind anycast is that a client wants to send packets to any one of several possible servers offering a particular service or application but does not really care about any specific one.

Anycast can be used to implement resource discovery mechanisms which are powerful building blocks for many distributed systems, including file sharing etc. Anycast in DTNs means that a node wants to send a message to any one of a destination group and intermediate nodes help to deliver the message by leveraging their mobility when no contemporaneous path exists between the sender node and any node of the destination group. Moreover, DTN anycast can be used in a disaster rescue field, in which people may want to find a doctor or a fireman without knowing their IDs or accurate locations. Thus efficient anycast service is important for supporting these applications in DTNs. Anycasting essentially provides a means to locate and communicate with *any one* of a set of distributed servers or service access points within a network. This is analogous to providing an individual that needs to make a phone call with directions to a public payphone. While there are potentially many points of service, the end user only needs to find one. In a networking context, anycasting facilitates more robust distributed system design and eases network configuration and management.

Though anycast in the Internet and mobile ad hoc networks has been studied extensively in the past, due to the unpredictability of network connectivity and delay, and limited storage capacity, anycast in DTNs is a quite unique and challenging problem. It requires both re-definition of anycast semantics and new routing algorithms. Moreover, in unicast in DTNs, the destination of a message is determined when it is generated, while in anycast, the destination can be any one of a group of nodes and during routing, both the path to a group member and the destination can change dynamically according to current mobile device movement situation. The anycasting routing objectives in DTNs can be

- Maximizing message delivery ratio, Messages will be dropped if buffer of intermediate node is overflowed.
- Minimize message latency; the average message latency should be minimized as much as possible.
- Routing efficiency, the overall routing should be efficient.

1.5.1 Anycasting Applications in DTNs:

There are various applications in DTN which require anycasting services. Some of them are:

- In Disaster recovery the rescue workers need to give information about the victims, hazards and other related to one of the server out of a set of servers.
- In military deployment there is need to deliver fast and reliably information from a mobile field commander to a nearest mobile command center or another mobile field commander out of many mobile command centers or commanders.
- In DTN Geocasting (Delivering of Message to all hosts in a geographical location) the efficient approach will be that first use the anycast and then broadcast within that particular geographical region.

CHAPTER 2

LITERATURE SURVEY

2 Literature Survey

This section provides the related works.

2.1 Anycasting in Internet and MANETs

There is a rich literature on anycast routing in the Internet and mobile ad hoc networks. Now-a-days the use World-Wide-Web has been increased. Business, industrial, education and other communities are taking keen interest in using World-Wide-Web for variety of applications. Most visited websites like British Broad-Casting (BBC), Yahoo websites, can deliver poorly for clients in overloaded states if even powerful single server is used. Also if someone hacks a site then no information will be available. That's why the mirrors of a website are created. Each mirror provides same services. The mirrors are kept consistent by using different approaches.

Shah et al. [19] proposed a selective anycast service to choose best mirror server from among many available depending upon some application-specific metrics. They describe metrics can be

- The network delay between the client and each server.
- The available bandwidth between the client and each server.
- Whether or not the request can be served out of the cache at each server.
- The load at each server.

They assumed that amount of data to be transmitted is small. They are taking following two metrics

- The network delay between the client and each server.
- The available bandwidth between the client and each server.

Their selective anycast protocol is designed to select best one server out of set on the basis of above two metrics. They also proposed mechanism for computation of these application-specific metrics.

This is an application-layer anycasting, here the connectivity is guaranteed and the end-to-end delay is within the range of TCP/IP protocol suits. But this approach cannot be directly applied to anycasting service for Delay Tolerant Network, where the end-to-end delay is more than the range that TCP/IP protocol suit support.

Katabi et al. [20] paper proposed a scalable architecture for Global IP-Anycast (GIA). GIA scales due to two types of routes generation in its inter-domain routing protocol:

- Default inexpensive routes that consume no bandwidth and storage space.
- Enhanced shortest path routes that are customized according to the beneficiary domain's interest.

Park et al [21] discussed the anycast service in mobile ad hoc networks(MANETs). They considered how to locate and forward the networking traffic to one out of set of distributed services in MANETs. The anycasting can provide significant improvement to the mobile network architecture e.g. in military mobile network which experience frequent topological changes. Anycast routing also simplifies the required configuration and management. Anycast routing also reduces latency setup and overall message delay.

Xie et al. [22] solve the problem of ***SARP*** (an anycast protocol in mobile ad hoc network) to transmit real time data and overhead of continuously rebuilding transmission route by proposing new anycast routing protocol in mobile ad hoc network, called efficient anycast routing protocol (EARP). Comparing to SARP their protocol outperform by

- Increasing packet delivery rate.
- Reducing the recovery time of broken links.
- Reducing route discovery time.
- Reducing controlling over heading.

They assumed the environment for their new anycast protocol characterized by

- Wireless antenna is used for communication by every node.
- Link is bi-direction i.e. if one node can transmit data to its neighbours, the node can also receive data from that node.
- Reliability of their ad hoc network is ignored. The upper layer protocols are responsible for the lost and retransmission of packets.

Wang et al [23] proposed anycast protocol, called Anycast Ad hoc On-demand Distance Vector (A-AODV) for IP flow on the basis of ad hoc on-demand distance vector (AODV) unicast routing protocol in MANETs. They concluded that their anycast routing protocol in mobile ad hoc network can:

- Balance the network load efficiently.
- Reduce the delay of packet
- Improve the network throughput.

They presented anycasting in mobile ad hoc networks at network layer. However it has mobility but still it is assumed that end-to-end connectivity is guaranteed and end-to-end delay is within the range that TCP/IP protocols support due to handling mobility.Therefore, this approach also will perform poor in delay tolerant networks. Because in delay tolerant network the end-to-end delay may exceed the range that TCP/IP support. Although IP layer also support storage support but its duration is limited while in delay tolerant networks the outgoing links may not be available for longer duration. So IP layer storage will not work. Therefore in such scenario routing data using store-and-forward fashion in mobile ad hoc network can work efficiently. Each intermediate store the in-transit data packet in its local buffer. The intermediate node forwards the packet to the next hop as opportunity arises, according to routing algorithm. The characteristics of delay tolerant network are different than mobile ad hoc networks (MANETs) and Internet.

Therefore there is a need to design an anycast protocol according to challenges of delay tolerant network.

2.2 DTNs Routing Protocols

Next we are going to describe the related work about DTNs' architecture and unicast algorithms.

Fall in [18] proposed architecture for delay tolerant networks. It is based on asynchronous message forward and operates as an overlay above transport layer. It provides key services such as in-network data storage and retransmission, interoperable naming, authenticated forwarding and a coarse-grained class of service.

They argued that link-repair approaches and network-specific proxies approach can not suit to achieve interoperability among the networks that experience frequent network partitions (challenged networks) and Internet. In link-repair approach, problem links are engineered to become like the links for which TCP/IP suit. This approach maintains the end-to-end reliability and fate sharing model of Internet, and generally use IP in all participating routers and end hosts. In network-specific proxies approach, the challenged networks are attached to the Internet by means of a special proxy agent.

Vahdat et al [16] proposed a routing algorithm, called Epidemic Routing, in ad hoc networks to deliver messages in the case where there is never connected path from source to destination or when a network partition exists at the time a message is originated. They considered the scenario where the mobile nodes are communication with low radio range. This can cause to the scenario where is no end-to-end path between the source node and destination node at the time of message origination. In this technique, there is a pair-wise exchange of message between mobile nodes which results ultimately to destination node. Their minimal assumptions about the connectivity of the underlying ad hoc network

- The sender is never in range of any base stations
- The sender does not know where the receiver is currently located or the best "route" to follow.
- The receiver may also be a roaming wireless host.
- Any pairs of hosts periodically and randomly come into communication range of one another through node mobility.

In their Epidemic routing algorithm each node transmits to all its neighbours the packet. This flooding causes early delivery of messages and high message delivery ratio. To minimize the aggregate system resources consumed in message delivery, they assign an upper bound to message hop count and per-node buffer space i.e. if the message is forwarded by the number of hopes exceeding hop-count in message, the message is dropped. Also the message is dropped, if a message arrives at the node and the node does not have buffer for it. Different dropping policies can be used.
Epidemic routing algorithm is good in a situation where high delivery ratio and low delay is required provided that buffer space and power resources are large enough.

Thrasyvoulos et al [24] proposed routing scheme, called Spray and Wait, for intermittent mobile ad hoc networks that are sparse wireless networks. They also assumed that most of time there does not exist end-to-end path from the source to destination at the time of message generation. Their routing scheme "sprays" a number of copies into the network and then "waits" till one of these nodes meets the destination. Their routing scheme bounds the total number of copies and transmissions per message without compromising performance. They results that Spray-and-Wait routing algorithm:

- Outperforms flooding-based routing algorithm under low load due to fewer transmission and comparable or smaller delay.
- Yields significantly better delays and fewer transmissions than flooding-based routing schemes under high load.
- Scalable, having good and predictable performance for large size networks.
- Tuned online to achieve given Quality-of-Service requirements, even in unknown networks.

Their algorithm outperforms in term of average end-to-end delay and network traffic in scenario where resources like power and storage is limited.

Jain et al [25] developed and compared several unicast routing algorithms in DTNs. They considered the class of delay tolerant networks where connectivity among the nodes is scheduled. They have studied the impact of knowledge availability about network topology on different routing algorithms. They have presented the following classes of routing algorithms for delay tolerant networks depending upon knowledge availability related to network topology.

- Zero Knowledge: These algorithms do not use any knowledge about network topology.
- Complete Knowledge: These algorithms utilizes complete knowledge, like Traffic Demand Oracle, Queuing Oracle etc, about network condition.
- Partial Knowledge: These algorithms do not use traffic demand oracle and use one or more knowledge about network condition.

They considered issues like routing objectives, the amount of knowledge available, when routes are computed, the use of multiple paths and the use of source routing in context of delay tolerant networks for routing algorithm.

They concluded that:

- The algorithms utilizing least knowledge of network condition for route computation tends to perform poorly.
- With limited additional knowledge of network condition, far less than complete knowledge, efficient routing algorithms can be constructed routing in such environments.

The above discussion is related to unicast routing algorithms in delay tolerant network.

2.3 Group Communication in DTNs

Now we are going to discuss group communication, like anycasting and multicasting, in delay tolerant networks. There is a lot of literature available on group communication in Internet and mobile ad hoc networks. We are going to present multicasting and anycasting in delay tolerant networks. These are to show importance of group communication in delay tolerant networks.

Zhao et al. [26] presented new semantic models and several multicast routing algorithms with different routing strategies in delay tolerant networks. They have presented following semantic models for multicasting in DTN to explicitly to specify the members of multicast group.

- Temporal Membership Model: A message includes a membership interval that specifies the period during which the group members are defined.
- Temporal Delivery Model: A message specifies both a membership interval and a delivery interval. The delivery interval indicates the time period during which the message should be delivered to the intended receivers.
- Current-Member Delivery Model: the receivers of the message should be group members at the time of message delivery.

They have studied various oracles of memberships and contact effects on performance of routing approaches. They have evaluated the following routing approaches to address multicasting in delay tolerant networks.

- Unicast Based Routing.
- Broadcast Based Routing.
- Dynamic Tree Based (DTBR).

For multicasting addresses, they use explicit addresses of each group member in each packets, unlike that of Internet multicasting and MANETs. There is group number for each multicast group in IP multicasting while in Overlays multicasting, application layer multicasting, there is structured maintained among group members.

The DTBR causes less traffic as compared to other two.

They concluded that

- Efficient multicast routing for DTNs can be constructed using only partial knowledge.
- Accurate topology information is generally more important in routing than up-to-date membership information.
- Routing algorithms that forward data along multiple paths achieve better delivery ratios, especially when available knowledge is limited.

They also assumed that prior knowledge about connectivity among nodes and memberships of users are available.

Ye et al [10] presented OS-multicast, multicast routing algorithm, for the class of delay tolerant networks where connectivity is opportunistic. Their mechanism use on-demand path discovery and overall situation awareness of link availability to handle the opportunistic links connectivity in delay tolerant networks, which have many applications in wireless networks. They use Dynamic Source Routing [6], reactive routing protocol, as an underlying network routing protocol. The situation awareness is achieved by making collaboration between DTNs agent and underlying routing agent. It also uses explicit addresses of group members in each packet. In DTBR [26] a node does not take advantage of path availability to the node which was multicast receiver but not included in its downstream list. The OS-multicast also handles this problem.

They concluded that

- OS-multicast achieve better message delivery ratios as compared to DTBR [26].
- OS-multicast achieves better efficiency than DTBR [26] when the probability of link availability is high and the duration of link downtime is large.

Gong et al. [27] presented Anycast Routing in Delay Tolerant Networks.The work is discussed in detail below

a) Network Model:

They assume that the DTN graph is a directed graph $G = (V,E)$, where V is the set of nodes and E is the set of edges. An edge between node u and v means that there exist some mobile devices moving from the initial node u (source) to the terminal node v (destination) (see Fig. 2). The storage capacity on all the nodes and the mobile devices is limited. They assume that every mobile device that moves between the same initial node, u, and terminal node, v, has the same moving speed, thus having the same moving delay, $d(u, v)$, from the source to the destination. The departure time of the mobile devices is a random variable and follows some given probability distribution. $w(u, v)$ is the probability distribution function of the waiting time until a mobile device leaves from the source to the destination. $c(u, v)$ is the storage capacity of the mobile device. They assume nodes in the network are stationary and generate messages, while mobile devices' mobility can't be controlled by the nodes and do not generate messages themselves. Nodes might be disconnected themselves, thus mobile devices can act as carriers to deliver messages for the nodes. Before it departs, a mobile device's initial node uploads selected messages. Upon arriving at its terminal node, it offloads the messages. Fig. 3 illustrates a simple example of their DTN model, in which a single node represents a cluster of users: the cluster is relatively stationary and the users only move within it. There are some mobile devices, such as shuttle buses, cars or people, moving among the clusters. The exact schedule of the moving devices is unknown, but their moving patterns can be obtained.

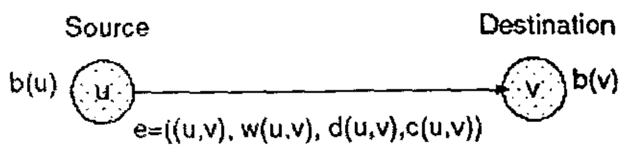

Fig. 6. An edge in a DTN graph, which is characterized by its source and destination nodes, plus a PDF of mobile device leaving (*we*), a moving delay (*de*) and a Carrier storage capacity (*ce*). $b(u)$ is the storage capacity of node u.[21]

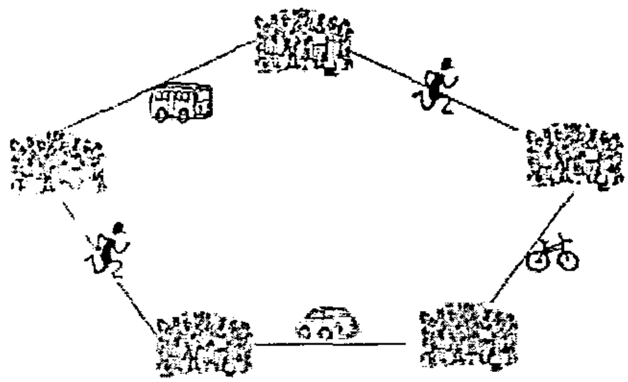

Fig.7 A typical scenario of DTN [21]

b) Semantic Model for Anycast in DTN:

Next they describe following three anycast semantic models that allow message senders to explicitly specify the intended receivers of a message.

- Current Membership Model (CM)

  A message should be delivered to a node which is a destination group member when a message arrives at it.

- Temporal Interval Membership Model (TIM)

  In TIM a message includes an interval that specifies the period during which the group members are defined.

- Temporal Point Membership Model (TPM)

  In TPM a message also includes a membership interval. For a message for an anycast group G with temporal interval $[t1, t2]$, its intended receiver at least should be a member of group G at some time during the interval.

The authors use the CM model to define their anycast routing metric and algorithm.

c) EMDDA (Expected Multi-Destination Delay for Anycast):

This metric accurately indicates the delay from a node to the nearest member of the destination anycast group. They use Practical Expected Delay (PED) to denote the delay between any two nodes which takes into account the probability of choosing each neighbor as the next node.

Given the network graph $G = (V,E)$, and suppose a message m which should be sent to any member of a group $Dm = \{d0, d1, \ldots, dL-1\}$, $di \in V$, $0 \le i \le L-1$, and L is the size of the group. They assume that before a mobile device starts moving, all messages that it should carry can be uploaded as long as the storage of the mobile device is not full.

One simple anycast routing algorithm is to use the sum of the expected waiting time ($E(w(u, v))$) and moving delay ($d(u, v)$) as the weight of each edge, then to calculate the shortest path to each group member and choose the one with the smallest path weight as the anycast destination. They called this Minimum Expected Delay (MED) approach.

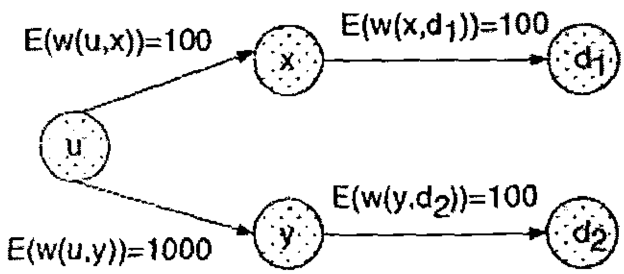

Fig. 8. A simple example of changing route. In this graph, they assume the moving delay of each edge is identical. According to the MED approach, a message should take the path $u \to x \to d1$. But when a mobile device from u to y is leaving, the better choice is the route $u \to y \to d2$ [21].

Let us use $MED(s, d)$ to represent the minimum expected delay between nodes s and d, ($s, d \in V$), which can be calculated through a shortest path algorithm, such as the Dijkstra algorithm. While a message is waiting for a mobile device to the next node on the path of the minimum expected delay, another mobile device to another destination comes, should the message change its next-step node or destination and use this mobile device? Yes, if some conditions hold. Fig. 4 shows a simple example of changing route. According to the MED approach, an anycast message heading for a group $\{d1, d2\}$ should follow the path $u \to x \to d1$. While the message is waiting a mobile device to node x, a mobile device to node y comes, the message should change its route, take the mobile device to y and arrive at $d2$ finally, because the leaving time of mobile devices is random and the expected waiting time can't accurately reflect the specific situation that each message meets on its way to its destination. They introduce a new routing metric called Practical Expected Delay (PED) to characterize the expected delay of taking different paths with corresponding probability between a node pair. For a node c, $Nb(c)$ represents the set of all its neighbors. Let $P(c, d, a)$ be the probability that a message at node c to destination node d takes one of node c's neighboring nodes, a, as the next step node, i.e. $a \in Nb(c)$. Then

$$\text{Then } PED(c, d) = E(w(c,d)) + \sum_{a \in Nb(c)} P(c,d,a) \cdot (d(c,a) + PED(a,d))$$

Next, they calculate the value of $P(c, d, a)$ ($a \in Nb(c)$). If $d(c, a) + MED(a, d) > MED(c, d)$, $P(c, d, a) = 0$, i.e. while waiting, even if a mobile device to node a is ready to move, the message won't take it. Let set $A(c, d) = \{a \in Nb(c) | (d(c, a)+MED(a, d)) \leq MED(c, d)\}$. $P(c, d, a)$, $a \in A(c, d)$, is equal to the probability that a mobile device to node a leaves earlier than any mobile device to any other node in the set $A(c, d)$.

They extend the metric PED to anycast routing and get a metric called Expected Multi-Destination Delay for Anycast (EMDDA). $EMDDA(s,D) = min(PED(s, d0), PED(s, d1), \ldots, PED(s, dL-1))$, $s \in V$, and D is the destination node set and $D = \{d0, d1, \ldots, dL-1\}$. The key advantage of EMDDA is that it can reflect the expected delay between a pair of nodes by taking all possible paths into account instead of only the shortest path.

d) Routing algorithm:

Their anycast routing algorithm is based on the metric EMDDA. It can be seen that if mobile devices move infrequently between two nodes, the edge between them will have high expected delay. Thus the shortest path algorithm won't use these edges. However, if the edge is available immediately, its waiting time becomes zero at this moment and it may be a very good choice for routing. One advantage of our routing algorithm is that it can make use of this phenomenon.

e) Performance Evaluation

They evaluate the performance improvement EMDDA-based algorithms can achieve over MED-based algorithms when applied to anycast in DTN environments.

Performance Metric:

- Anycast Delivery Delay

  Anycast Delivery Delay(ADD) of a message, is defined as the time to route it from its sender to any node in its anycast destination group. *Average Anycast Delivering Delay* (AADD) is the average ADD of all possible anycast sessions.

- Average max queue length(MQL)

  Average max queue length is the average of the max queue lengths on all nodes. Messages will be queued on intermediate nodes if the mobile devices to the next hop are not available immediately or they do not have sufficient capacity to carry.

Through simulation results they show the following results

- EMDDA can effectively reduce the delay in anycast routing.
- The improvement of EMDDA over MED will increase as the hop number increases since EMDDA can route messages to a closer destination intelligently along the route.
- For both EMDDA and MED, increasing storage capacity will reduce the AADD because mobile devices can carry more messages in one delivery and consequently reduce the queuing delay in intermediate nodes.

The average MQL of EMDDA is always smaller than that of MED.

CHAPTER 3
PROBLEM DOMAIN

3 Problem Domain

Delay tolerant networks are characterized by frequent network partition, large end-to-end delay and end-to-end path may not exist between source and destination at the time of message generation. Due to these unique characteristics of DTNs, MANETs and Internet protocol perform poor. So designing different routing protocols for DTNs is challenging task. Demand for efficient anycasting service has been grown in various applications. Anycasting has been studied extensively in mobile ad hoc networks and Internet. However, due to unique characteristics of DTNs, efficient anycasting in DTNs is challenging task.

3.1 Problem Identification:

Gong et al. [27] assume that nodes in the network are stationary. The connectivity among the nodes is the mobile devices which act as carrier to deliver messages for the nodes. The mobile devices do not generate messages themselves, i.e. message carriers, known as message ferries [13], are fixed. While in most of DTN applications like in [28] any node may act as message ferry. Also in [27] the moving patterns of the mobile carriers can be obtained while in most of DTN applications one can't predict the mobility pattern.

We are interested in a specific type of DTN where the mobile nodes are sparsely distributed and communicating via limited wireless radio range such that network experience frequent and long duration partitions. All DTN nodes are moving having limited storage capacity, and connectivity among the nodes is opportunistic. A node may be a source, receiver or the carrier which carry the message and forward the message as the opportunity arises. These scenarios are required by many wireless and sensor network application like military deployment and disaster recovery etc.

MANET and Internet anycast will fail due to the challenge of frequent and long duration partitions and end-to-end connectivity may not exist.

Furthermore in anycasting the packet is transferred to any one out of a set of nodes. Therefore the destination may changes on every intermediate node. On each intermediate node the packet is forwarded to the node that have shortest path. In tie situation, i.e more than one shortest paths are available, the packet is forwarded to first found shortest path or depending on node ID. We refer this forwarding scheme as Normal Forwarding (NF).

This scheme is valid in MANETs and Internet due to guaranteed end-to-end path. However DTNs experience frequent partition and there is probabilistic connectivity among the nodes.

Consider there is tie situation, one node leads to one receiver and the other node leads to more than one receiver. NF forwarded the packet to the node from which one receiver is reachable. If the network partition occurs, which is more frequent in DTNs, and path became unavailable to the receiver then the packets will be buffered for longer time and may be dropped due to buffer overflow (because there may be long duration partition). Generalizing the problem if the packet is forwarded to the next hop from which fewer receivers are reachable then the probability of receivers' availability will be less and this will leads to longer end-to-end delay, lesser packet delivery ratio and the overall efficiency will decrease.

3.2 Proposed Solution:

We proposed packet forwarding mechanism for anycasting in delay tolerant networks named Receiver Base Forwarding (RBF). RBF consider not only path length but also take the number of reachable receivers into account. In tie situation, RBF forwards the packet to the next hop from which more receivers are reachable so that the probability of receivers' availability increases. As the propagation delay is less than the partition delay so if the path to one receiver is not available then it can be forwarded to another available receiver. Therefore RBF decrease the average end-to-end delay, increase the packet delivery ratio and enhance the overall efficiency. Thus RBF improve the performance of anycasting in DTN.

We studied a specific type of DTN where the mobile nodes are sparsely distributed and communicating via limited wireless radio range such that network experience frequent and long duration partitions. All DTN nodes are moving having limited storage capacity, and connectivity among the nodes is opportunistic. A node may be a source, receiver or the carrier which carry the message and forward the message as the opportunity arises. Ye et al. [10] used situation discovery mechanism provided by the underlying network layer for multicasting in such scenario. It is a good effort and we use the situation awareness mechanism provided by underlying network for the anycast service in DTN, with new packet forwarding mechanism (RBF).

For anycasting in DTNs we implemented collaboration between DTNs agent and routing agent used in underlying network, in this way situation awareness is achieved. We used the DTN agent that sends periodically request message to underlying routing agent. The underlying routing agent sends back to DTNs agent the network condition in response. In this way DTNs agent updates its knowledge about underlying network condition which will help in making routing decision.

Next we show our proposed scheme through an example. Fig. 9 shows a simple anycast session Node 0 is source and node 5, 7 and 9 are members of anycast. When node 0 wants to send anycast bundle to one of anycast members then node 0 will send request message to underlying agent to trigger the network condition to anycast members. Node 0 will come to know about the paths to anycast members. Now node 0 will make mesh by combining these discovered paths to anycast members as shown in Fig. 9(a).

The dashed lines show available links so the node 0 will delete all those outgoing links that are currently not available, shown in Fig. 9(b).

Now the node 0 has two choices to forward the bundle, either to node 3 or to node 1. The normal forwarding solve the tie situation by forwarding the bundle to the node having smaller node ID or first available node, but we will select that node through which more anycast receivers can be visited, so the probability of receiving the bundle by any one receiver will increase, so node 0 will forward the bundle to node 1 as shown in Fig. 9(c). Along the message list of anycast relievers will be included. Similar operation will be done at each DTN node for each bundle. Node 1 get paths to node 5 and node 7 then forward to 4 as in Fig. 9 (d). Node 4 finds the path to node 5 so it forward the bundle to node 5 as in Fig. 9 (e)

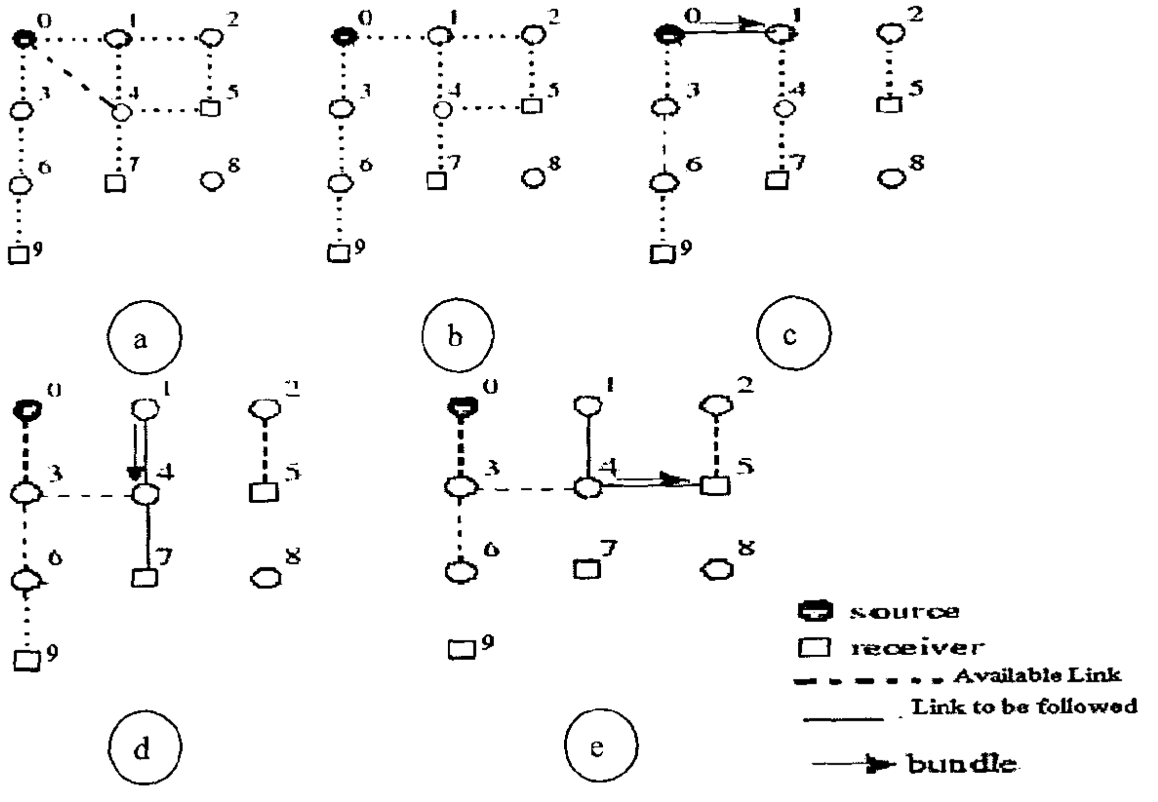

Fig9. Shows an Example of Adaptive Anycast Protocol

4 System Design

The architecture for the involving implementations is created in design phase. The output of design phase is a model of the system that can be used latter in implementation phase to build the system. Its goal is to select the optimize design according to the limitation imposed by the requirements and the operational environment among a set of alternatives. Object oriented design is the method of designed encompassing the process of objects oriented decomposition and notation for depicting logical and physical as well as static and dynamic models of system under design.

4.1 High Level System Model

High level design of the system is shown below.

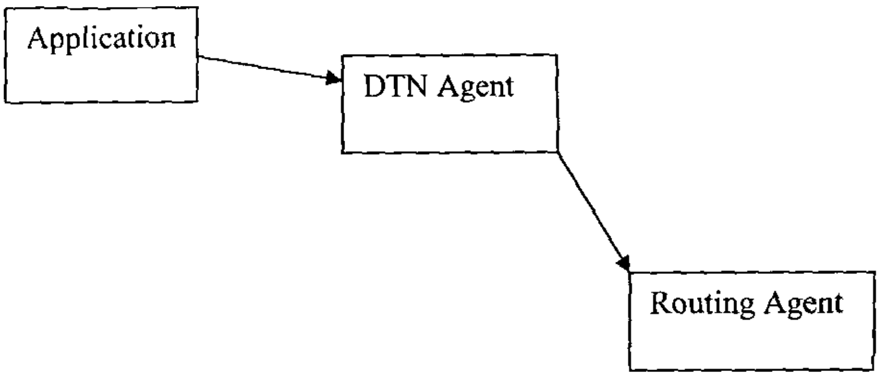

Fig.10 High level System Model

Following is the explanation for the above-mentioned components.

- **Application**; creates data to be sent to another application across network.
- **DTN Agent**; its responsibilities include to buffer the message, routing at bundle layer and other functionality specified in bundle protocol.
- **Routing Agent;** its responsibility is to do appropriate routing in underlying networks. We have achieved situation awareness by implementing collaboration between DTN agent and underlying network Routing agent.

4.2 DTN Agent

Our major concerned is with DTN agent and underlying Routing agent. Therefore next we are going to describe our modification to these two in detail.

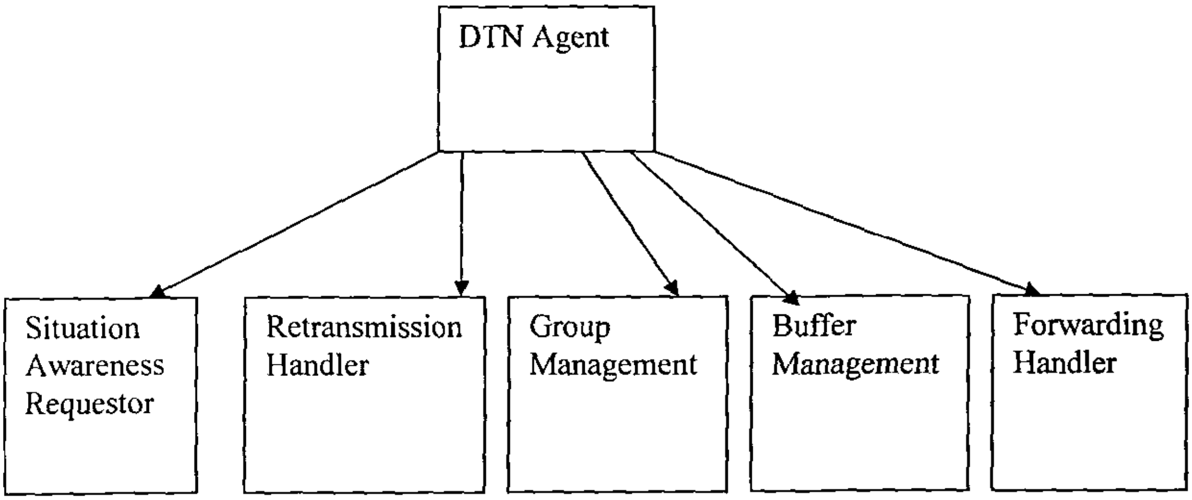

Fig.11 Showing Our components for DTN agent.

Following is the description for the above components.

Situation Awareness Requestor

As DTN experiences frequent and long duration partitions. We assume that underlying network is operational for long time so that caches are built up and sufficient knowledge about the network topological condition is available, to achieve situation awareness each This component has the responsibility to periodically send request (RoutRqt) to underlying routing agent for the current network condition, for this we use Dynamic

Source Routing (DSR) as an underlying routing protocol. The RoutRqt contains the list of anycast receivers' addresses; we use explicit addresses instead of group number.

Group Management

This is designed to handle group join and leave request. Any DTN node that wants to join anycast group sends a join request (JRqt), containing time interval during which it will be available. The join request is flooded so that it can reach the anycast source as early as possible. Similarly a member can explicitly drop membership by sending the drop information packet (DinfPkt), so that the anycast source will not consider it for next more traffic. Each DTN anycast source updates its members list on receiving the joining or drop requests.

Buffer Management

Each DTN node has limited buffer to store the in-transit packets. The packet is forwarded to the next hop and is removed from the buffer on receiving the acknowledgment from the next hop that is the custodian transmission is enabled to ensure the reliable delivery between two hops. Packets are dropped when the buffer overflows, depending upon the packet drop policy, for which we use first-in-first-out (FIFO) policy.

Forwarding Handler

When a node wants to forward the packet it gets the current topological information from the underlying routing agent. In response of RoutRqt, the underlying routing agent sends the discovered paths information from the current node to the receivers in the list, to the forwarding handler. It then removes the current unavailable outgoing links. Forwarding handler finds the path to the closest anycast member. In a situation when there are more than one shortest path available to the anycast receivers, the normal forwarding scheme takes the first one shortest path or on the base of node ID(if NF is enable) or a receiver base forwarding scheme for anycast in DTN that select the shortest path on which more receivers are reachable (if RBF is enabled), so that the probability of the receiver availability increases.

Retransmission Handler

It checks periodically every one-second its local buffer for unacknowledged packets, provided its existence they are retransmitted according to anycast receiver list. In order to reduce the overhead there are controlled retransmissions for a threshold. Once the acknowledgment is received, the packet will be deleted from the buffer.

4.3 Routing Agent

We have implemented collaboration between DTN Agent and underlying routing agent for situation awareness. Therefore we have made several changes in underlying DSR routing protocol. Following is the model of Routing Agent

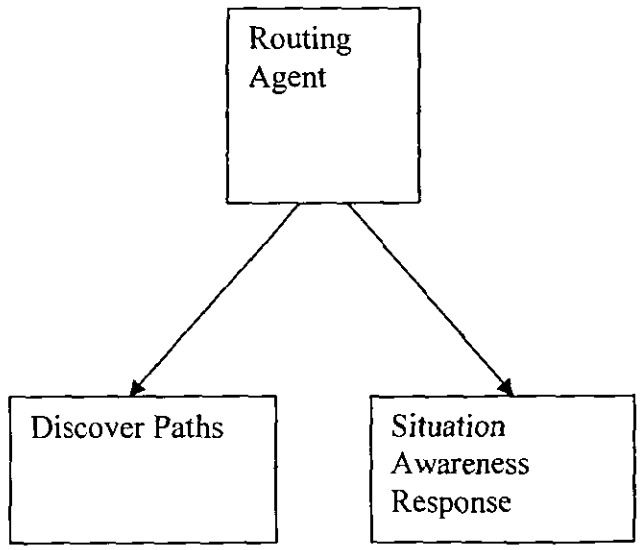

Fig. 12 shows the model for Routing Agent

Following is the description about the above mentioned components.

Discover Paths

This component receives the RoutRqt from the DTN agent. RoutRqt contains the list of receivers to which paths are to find. It finds all routes to the destination DTN nodes specified in RoutRqt.

Situation Awareness Response

It finds the length of each path to every DTN node (anycast receivers specified in message), format the message and sends RoutResp to the DTN agent. RoutResp contains all possible paths with length of each to anycast receivers, specified in RoutRqt.

CHAPTER 5

IMPLEMENTATION

5 Implementation

We have simulated the system, Adaptive anycast protocol, in ns-2[29] simulator. There are various advantages of using simulation as compared to real networks for testing. Some of them are

- It is cheaper to setup simulation environment than real networks. Because in real network one have to use a lot of machines, cables and other network components while in simulation environment these are not required.
- It is faster to setup simulation environment than real networks. Because in real network setup took more time to configure different machines and other network components while in simulation one only needs network topology description.
- Simulation environment take less space as compared to real networks. Because in real network one have to put a lot of machines, power cables and other network components while in simulation one have to only installed simulator on a machine.

Also there are some difficulties; one can say limitation, in simulation as compared to real networks e.g.

- Each simulator has its own code-base. That is the protocol to be implemented should be interoperable to the simulator.
- Validation is another problem in simulations due to assumption made regarding the network environment.

Learning ns-2 simulator was very difficult for us because there was no good tutorial available which could provide a step-by-step approach. However reading ns-document [30] extensively, practicing in ns-2 and using ns-group on Internet enable us to implement our protocol in ns-2.

5.1 NS-2 Simulator:

Ns began as a variant of the REAL network simulator in 1989 and has evolved substantially over the past few years. In 1995 ns development was supported by DARPA through the VINT project at LBL, Xerox PARC, UCB, and USC/ISI. Currently ns development is supported through DARPA with SAMAN and through NSF with CONSER, both in collaboration with other researchers including ACIRI. NS is open source and there is a great contribution from other researchers' e.g. wireless code from the UCB Daedelus and CMU Monarch projects and Sun Microsystems. NS is discrete event network simulator use for networking simulation. Ns provides substantial support for simulation of various routing protocols (including unicast, multicast etc) over wired and wireless (local and satellite) networks, different queuing schemes and other network related schemes.

NS is object-oriented simulator, written in C++, with OTCL interpreter as front-end. NS uses two languages C++ and OTCL. C++ is used for detailed simulations of protocols which require efficiently manipulating bytes, packet headers, and implementing algorithms that run over large data sets. Because for these tasks run-time speed is important and turn-around time (run simulation, find bug, fix bug, recompile, re-run) is less important that is provided by C++. OTCL is used for slightly varying parameters or configurations, or quickly exploring a number of scenarios. Because such tasks iteration time (change the model and re-run) is more important and run-time of this part of the task is less important that is provided by OTCL. There is basic advise in ns-document[30] regarding use of OTCL for

• Configuration, setup, and "one-time" stuff

• If you can do what you want by manipulating existing C++ objects

And their advice [30] regarding use of C++

• If you are doing *anything* that requires processing each packet of a flow

• If you have to change the behavior of an existing C++ class in ways that weren't anticipated

5.2 Network Animator (Nam)

Nam [31], network animator, is used for visualisation of network scenario. It provides visualization of

- Packet flows, different packets can be coloured.
- Nodes' native packets queue.
- Packets which are dropped.

For wireless network simulation, nam plays an important role because it can help that whether a node is within range of another node. Nam is very important to analysis the mobile nodes' movements during simulation.

5.3 Ns-allinone Components

Ns-allinone is a package that contains required components and some optional components used in running ns. The package contains an "install" script to automatically configure, compile and install these components. After downloading, run the install script. We have found ns-allinone easier than getting all the pieces manually.

The ns-allinone-2.29 package that we have used contains following components:

- Tcl release 8.4.11 (required component).
- Tk release 8.4.11 (required component).
- Otcl release 1.11 (required component).
- TclCL release 1.17 (required component).
- Ns release 2.29 (required component).
- Nam release 1.11 (optional component).
- Xgraph version 12 (optional component).
- CWeb version 3.4g (optional component).
- SGB version 1.0 (optional component, builds sgblib for all UNIX type platforms).
- Gt-itm gt-itm and sgb2ns 1.1 (optional component).
- Zlib version 1.2.3 (optional, but required should Nam be used).

5.4 Plateforms for ns:

Ns can be used on following plateforms

- Unix (Free BSD, SunOS, Solaris).
- Linux (Red Hat, Enterprise Edition)
- Windows (95,98,2000,XP).

However for windows, Cygwing emulator is required for ns. The favorable operating system for ns is Linux/Unix operating system. We have used RedHat Enterprise Edition. We have used mobile wireless node with limited wireless range.

5.2 Mobile Networking in ns-2

We have used the wireless model that was originally ported as CMU's Monarch group's mobility extension to ns. Following API is used to configure mobile nodes

```
$ns_ node-config -adhocRouting $opt(adhocRouting)
                 -llType $opt(ll)
                 -macType $opt(mac)
                 -ifqType $opt(ifq)
                 -ifqLen $opt(ifqlen)
                 -antType $opt(ant)
                 -propInstance [new $opt(prop)]
```

-phyType $opt(netif)

-channel [new $opt(chan)]

-topoInstance $topo

-wiredRouting OFF

-agentTrace ON

-routerTrace OFF

-macTrace OFF

The mobile node is designed to move in three dimensions (X, Y, Z) however Z dimension is always zero. Mobile nodes movements are two types

- Random movements.
- Non-random movements (user defined movements for mobile nodes).

It is necessary to define topography for mobile nodes before nodes creation by using command

set topo [new Topography]

$topo load_flatgrid $opt(x) $opt(y)

Where opt(x) and opt(y) specify the boundary of simulation.
Currently in ns-2 following ad hoc routing protocols are supported for wireless mobile nodes

- Destination Sequence Distance Vector (DSDV)
- Dynamic Source Routing (DSR)
- Temporally ordered Routing Algorithm (TORA)
- Adhoc On-demand Distance Vector (AODV).

An improved trace format has been introduced for merging merging wireless trace with ns tracing, using cmu-trace objects. This new trace support is backwards compatible with the old trace formatting and can be enabled by the following
command:
$ns use-newtrace

The above command should be called before the universal trace command $ns trace-all <trace-fd>. Primitive *use-newtrace* sets up new format for wireless tracing by setting a simulator variable called *newTraceFormat*. Currently this new trace support is available for wireless simulations only and shall be extended to rest of ns in the near future.
An example of the new trace format is shown below:

```
s -t 0.267662078 -Hs 0 -Hd -1 -Ni 0 -Nx 5.00 -Ny 2.00 -Nz 0.00 -Ne
-1.000000 -Nl RTR -Nw --- -Ma 0 -Md 0 -Ms 0 -Mt 0 -Is 0.255 -Id -1.255 -
It
message -Il 32 -If 0 -Ii 0 -Iv 32
s -t 1.511681090 -Hs 1 -Hd -1 -Ni 1 -Nx 390.00 -Ny 385.00 -Nz 0.00 -Ne
```

```
-1.000000 -Nl RTR -Nw --- -Ma 0 -Md 0 -Ms 0 -Mt 0 -Is 1.255 -Id -1.255 -
It
message -Il 32 -If 0 -Ii 1 -Iv 32
```

Some mobile nodes' movement files can be found in ~ns/tcl/mobility/scene/. But the generator for creating node movement files are to be found under ~*ns*/indep-utils/cmu-scen-gen/setdest/ directory. Compile the files under *setdest* to create an executable and then use following command for creating mobile nodes' movement file.

$./setdest -n <num_of_nodes> -p <pausetime> -s <maxspeed> -t <simtime> -x <maxx> -y <maxy> > <outdir>/<scenario-file>

CHAPTER 6

TESTING

6. Testing

System testing, an essential step for the development of a reliable and error-free system, is the process of executing a program with the explicit intention of finding errors.

This chapter explains our test framework that enabled us to test early and often. After completing design phase carefully, we implemented the system by adding one module (function) at a time with extensive testing. In this way we continuously controlled our work in order to be able to provide good quality of coding. Our test framework consists of three main parts. They are all described in the next paragraphs.

The basic strategies that were used for testing were following

6.1 *Structural (White Box) Testing*

It involves testing of a single unit of software. In our case these units are our modules (functions). The tests checked all the important elements of the module. We added all structural tests to one unit so that they can be performed in a batch process (all tests at the same time). This enabled us to run the batch often and regularly. The tests were run after each small change in the code, so in this way our system was always in a consistent state. Structural testing is also called white box testing because the exact kind of functions and code tested is known. There are two approaches for structural testing

- Top-Down testing
- Bottom-Up testing

In our Adaptive Anycast in DTNs implementation there are two direction of data flows. One direction is from application toward underlying routing agent. The other one is from underlying routing agent toward application or DTNs agent. Therefore we implemented both.

We applied the extreme Programming paradigm "test driven development". We first wrote down all the test functions needed to test the whole Adaptive Anycasting in DTNs implementation. This helped us thinking about the whole programming structure of our end product. We added the test functions to main test routine. At first, all these test functions were just stubs, without any implemented code. These functions/modules were implemented with a predefined interface. The whole structure of the certain module was tested by the test routine of that module.

Following features were implemented in module test function

- Implementation with a predefined interface and naming convention.
  - Interface: int function(void)
  - Name: modulename_test()
- When the test function was returned without an error, we were sure that the module is implemented properly and without bugs.
- We did many tests of the same function, if required. Input data may be random (i.e. for value range testing) or statically programmed.
- Called many different functions of the module.
- The tests were reproducible.
- As long the test function was not implemented, it was printed out that the module test function has not yet been implemented.

When we ran the test function first time, a list of messages was printed. Each line corresponded to one module test function and it showed that the module had not yet been implemented. We performed testing and correcting until we got message "All tests were successful" for each module test function.

6.2 Functional (Black Box) Testing

The second step in our testing concept was functional testing. In this phase we tested the functionality of our Adaptive Anycast in DTNs implementation. It is also called black box testing because it is not cared how the tested functionality was implemented and what functions are needed to implement this functionality. In this we assured that certain functionality did its job properly. For example that whether Forward Handler forwards the packet properly according to NF/RBF scheme.

We started functional testing with small tests. As more modules were implemented, the functional tests also increased and became complex. In this way we assured that the module work properly with each other. The last functional test was the test that checked the whole functionality of our Adaptive Anycast in DTN implementation.

Similar rules were used for functional testing that were used for structural testing. If function failed then it printed out the message "Failed" with input, output and expected output in order for the error to be reproducible.

6.3 Interoperability Testing

Interoperability testing is very important. We tested our implementation that it did not effect the operation of other communication protocol, used by some other application, in the network. That is if other services are using DSR protocol then it worked properly as expected. We have modified the DSR protocol according to our own requirements but these modifications do not affect it for other application use. We have tested it by running other application that used also DSR as underlying protocol.

6.4 Specification Testing

If the code testing is performed exclusively but still there is chances of program failure. Because it does not assure that whether the code meets the agreed/proposed specification and whether all the aspects of the design are implemented.

Therefore we performed the specification testing. In this we examined specification stating what and how program should do under various conditions. Test cases were developed to test the range of values expected including both valid and invalid data. Through specification testing we checked for discrepancies between the system implemented and its original objectives. During this phase bugs and minor errors were removed.

6.5 Regression Testing

In regression testing the system was tested abnormal values and it was tested that the system did not behave abnormally at any time. This phase was also succeeded.

6.6 Assertion Testing

In assertion testing the system is tested against the possible assertions. In this phase we checked the programs and various locations and were found that the state of the program at a particular point was the same as expected.

6.7 Unit Testing

Our system consists of multiple units/components and each has been designed for specific task. Before integration of components unit testing was performed. In this phase we checked that all components individually were working fine and ready to be integrated with other ones.

6.8 System Testing

When all the units were working properly and unit testing was performed then we checked all the integrated components as a whole and looked for possible discrepancies, which could have arisen after the integration. The system was operating rightly.

6.9Acceptance Testing

In acceptance testing the system is tested for its completeness that it is ready. Normally the quality assurance department performs the acceptance testing that the system is ready and can be used.

6.10 System Evaluation

The objectives of the system evaluation are to determine whether the desired objectives have been accomplished or not. Determining the merits and demerits of the proposed system over the existing system is also covered in the system evaluation. In this phase we studied in detail the developed system, from implementation point of view. At the end, some suggestions for the improvements of the system were coded.

CHAPTER 7

PERFORMANCE

7 Performance

For performance evaluation we implemented the proposed approach for anycasting in DTN with both NF and RBF in ns-2[29] simulator.

7.1 Metrics

We used following metric for the performance evaluation:

1) Message Delivery Ratio: it is defined as the ratio of total number of unique anycast bundles received by any anycast group member to the total number of bundles transmitted by the anycast source.

2) Data Efficiency: it is defined as the ratio of total number of unique anycast bundles received by any group member to the total traffic generated, both data bundles and control packets.

3) Average Message Delay: It is defined as the ratio of total delay for received anycast bundles to the total received anycast bundles.

7.2 Simulation scenario

We simulate a specific type of DTNs; in simulation we use

- 20 nodes.
- 800 x 800 simulation area.
- Mobile nodes with 110 m transmission range.
- DSR [7] is used as routing approach for underlying routing in network.
- MAC layer IEEE 802.11.
- Each DTN node with buffer capacity of 30 bundles.

We study one anycast session for 60 seconds. Node 1 is fixed as the anycast source while the other DTN nodes randomly join the anycast group by sending join request.The anycast source sends the bundle at the rate of 1 bundle per second. At every 2 second each DTN agent checks its buffer to see bundle waiting for retransmission.

7.3 Simulation Results

We have studied the performance of Adaptive Anycast protocol for DTNs with both NF and RBF by varying the link availability of each link uniformly for 10% to 90%.

7.3.1 Delivery Ratios

Fig. 3 shows the average delivery ratio of adaptive anycast algorithm with both NF and RBF schemes. Varying the link availability from 10% to 90%, the delivery ratio increases. When the availability is low, RBF achieves better delivery ratio because the probability of receiver's availability increases. As partition delay is for long duration. Therefore in RBF if one receiver is not available then it can be forwarded to other available, the packet is not buffered, and hence there is less chances of packets loss due to buffer overflows. While in NF if there is no path exist for further forwarding then it will be buffered, because the alternative receivers may be limited, so there is probability of buffer overflow which may results to packets loss. The delivery ratio of both schemes NF and RBF are almost equal at 90% or more link availability.

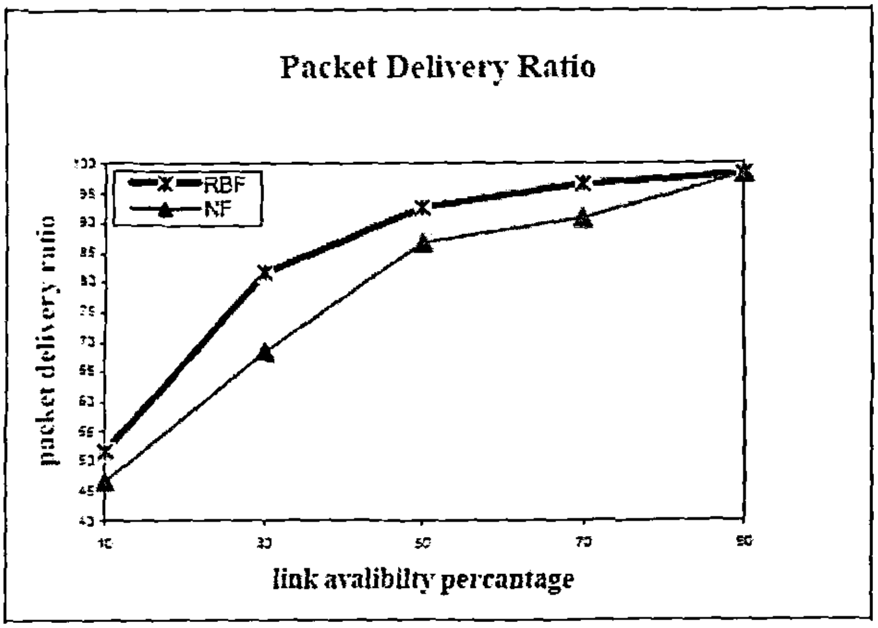

Fig. 3 Packet delivery ratio of Normal Forwarding (NF) and Receiver Base forwarding (RBF).

7.3.2 Overall Efficiency

Fig. 4 shows that when link availability is low the overall data efficiency of RBF is high because the probability of number of receivers is high and if one receiver is not available, the bundle may be forwarded to another receiver, and no extra traffic is generated for retransmission and checking the underlying network condition. At high link availability the difference between NF and RBF data efficiency decreases.

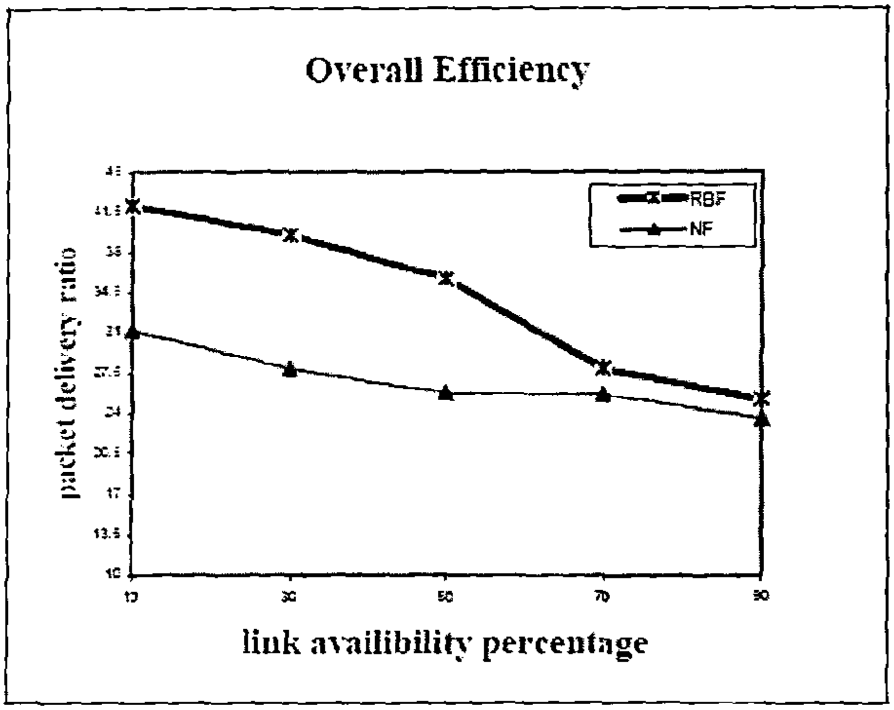

Fig. 4 Overall efficiency of Normal Forwarding (NF) and Receiver Base forwarding (RBF).

7.3.2 Average End-to-End Delay

Fig. 5 shows the average end-to-end delay performance using both NF and RBF. At low link availability the average end-to-end delay of both schemes abruptly increases due to the long duration partitions, which causes packets to be buffered for longer time. Similarly we noted that RBF experiences comparatively low delay due to the alternate receivers' availability and that the propagation delay is far less than the long duration partition delay. At high link availability the average end-to-end delay of both schemes is minimal. Above the 90% link availability the average end-to- end delay of both schemes is almost equal.

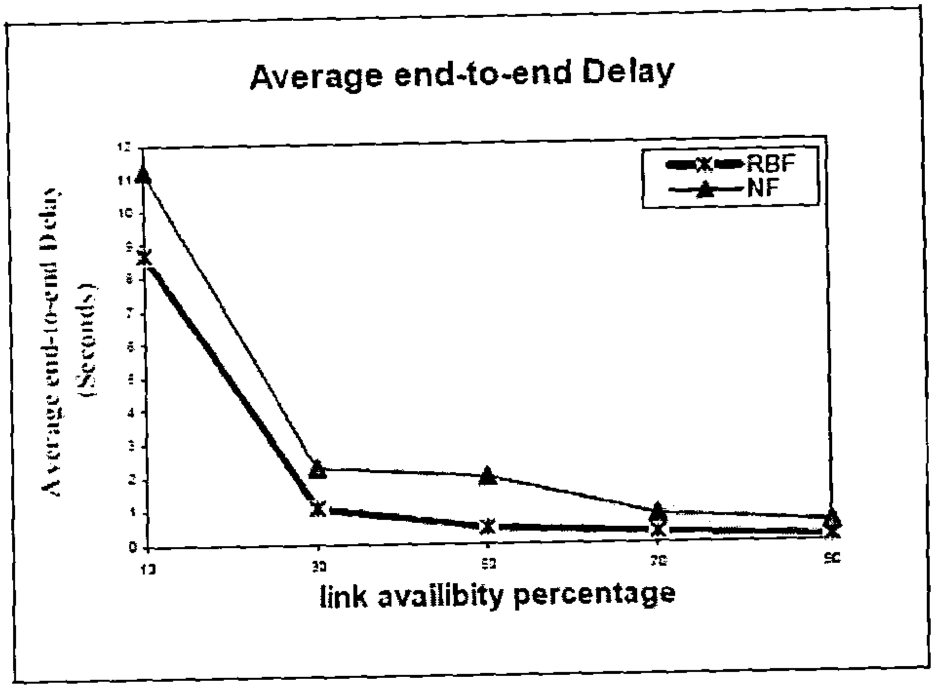

Fig. 5 Average end-to-end delay of Normal Forwarding (NF) and Receiver Base forwarding (RBF).

CHAPTER 8

CONCLUSION

8.1 Delay/ Disruption Tolerant Network

In Internet and mobile ad hoc networks the end-to-end connectivity is guaranteed. These networks have less end-to-end delay supported by TCP. However end-to-end delay is not guaranteed due to frequent and long duration partitions in delay tolerant networks. Also such networks have large end-to-end delay.

Delay tolerant networks are found in different forms like Inter-Planetary-Networks, intermittent connected mobile ad hoc networks and networks used for delivery of internet services to rural areas. Each of these types has unique features. We studied a specific type of delay tolerant networks where mobile nodes having limited storage and communicating through short range transmission such that they experience frequent partition, long duration partition and end-to-end connectivity may not be present between source and destination at the time of message generation. Delay tolerant networks operate above the transport layer. The packets are forwarded in store-and-forward fashion in DTNs.

Keeping in view these unique characteristics of DTN, the routing protocol of MANETs and Internet will perform poor in DTN. Several routing algorithms have been proposed by research community for DTNs.

8.2 Group Communication

The applications of group communication, like multicasting and anycasting, have grown. Several schemes have been proposed by research community for group communication in MANETs and Internet. But these approaches cannot be applied directly to DTNs due to its unique characteristics.

Anycasting service has a large number of applications in DTNs. We studied anycasting for a specific type of DTNs, where the mobile nodes are sparsely distributed, communicating via low radio range, experience frequent and long duration partition and end-to-end path may not be present at the time of message generation. We also proposed a RBF scheme for anycasting in DTNs, which consider the path length as well as the number of receivers in forwarding the anycast bundle to the next hop. NS simulation results show that the RBF performs better than the NF in terms of data delivery ratios, average end-to-end delay and overall data efficiency because the probability of available receivers increases.

When the availability is low, RBF achieves better delivery ratio because the probability of receiver's availability increases. As partition delay is for long duration. Therefore in RBF if one receiver is not available then it can be forwarded to other available, the packet is not buffered, and hence there is less chances of packets loss due to buffer overflows. While in NF if there is no path exist for further forwarding then it will be buffered, because the alternative receivers may be limited, so there is probability of buffer overflow which may results to packets loss. The delivery ratio of both schemes NF and RBF are almost equal at 90% or more link availability

When link availability is low the overall data efficiency of RBF is high because the probability of number of receivers is high and if one receiver is not available, the bundle may be forwarded to another receiver, and no extra traffic is generated for retransmission and checking the underlying network condition. At high link availability the difference between NF and RBF data efficiency decreases

At low link availability the average end-to-end delay of both schemes abruptly increases due to the long duration partitions, which causes packets to be buffered for longer time. Similarly we noted that RBF experiences comparatively low delay due to the alternate receivers' availability and that the propagation delay is far less than the long duration partition delay. At high link availability the average end-to-end delay of both schemes is minimal. Above the 90% link availability the average end-to- end delay of both schemes is almost equal.

8.3 Future Work

Delay tolerant networks offer a great number of challenges to research community. Security in DTNs is also a challenging task. We are interested in extending our proposed anycast scheme to Geocasting in DTNs. We are also taking keen interest in security mechanism, especially key management, in DTNs.

REFERENCES

Reference & Bibliography:

[1] Richard G. Ogier, Fred L. Templin, and Mark G. Lewis. "Topology dissemination based on reverse-path forwarding", IETF RFC3684 - Experimental Standard, February 2004.

[2] Charles Perkins and Pravin Bhagwat. "Highly dynamic destinationsequenced distance-vector routing (DSDV) for mobile computers". ACM SIGCOMM'9 Proceeding on Communications Architectures, Protocols and Applications, pages 234–244, 1994.

[3] Thomas H. Clausen and Philippe Jacquet, "Optimized link state routing protocol". IETF RFC3626 - Experimental Standard, October 2003.

[4] Thomas H. Clausen, Philippe Jacquet, Anis Laouiti, Paul Muhlethaler, Amir Qayyum, and Laurent Viennot. "Optimized link state routing protocol". IEEE INMIC Pakistan, 2001.

[5] Bhargav Bellur and Richard G. Ogier. "A reliable, efficient topology broadcast protocol for dynamic networks". Proccedings of the INFOCOM'99, volume 1, pages 178–186, March 1999.

[6] Charles Perkins and Elizabeth Royer."Ad hoc on demand distance vector routing." Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications, pages 90–100, February 1999.

[7] David B. Johnson and David A. Maltz. "Dynamic source routing in ad hoc wireless networks", Mobile Computing, volume 353, chapter 5, pages 153–181. Kluwer Academic Publishers, 1996.

[8] Vincent D. Park and M. Scott Corson. "A highly adaptive distributed routing algorithm for mobile wireless networks". Proceedings of IEEE INFOCOM '97, Kobe, Japan, April 1997.

[9] Zygmunt J. Haas and Marc R. Pearlman. "The performance of query control schemes for the zone routing protocol". SIGCOMM, pages 167– 177, 1998.

[10] Q. Ye, L. Cheng, M. Chuah, B. D. Davison, "On-Demand Situation Aware Multicasting in DTNs", Proceedings of IEEE VTC, Spring, 2006.

[11http://www.ipnsig.org/reports/DTN_Tutorial11.pdf

[12] H. Wu, R. Fujimoto, and G. Riley. "Analytical models for data dissemination in vehicle-to-vehicle networks". IEEE VTC 2004.

[13] Wenrui Zhao, Mostafa Ammar and Ellen Zegura, "A Message Ferrying Approach for Data Delivery in Sparse Mobile Ad Hoc Networks ", MobiHoc ACM, 2004.

[14] " Routing and Quality of Service in Wireless and Disruption Tolerant Networks" by Anders Lindgren, PhD Thesis, Lulea University of Technology, 2006.

[15] George W. Boehlert, Daniel P. Costa, Daniel E. Crocker, Phaedra Green, Todd O'Brien, Syd Levitus, and Burney J. Le Boeuf. "Autonomous pinniped environmental samplers; using instrumented animals as oceanographic data collectors". Journal of Atmospheric and Oceanic Technology, 18(11):1882–1893, 2001. 18 (11).

[16] A. Vahdat and D. Becker. "Epidemic routing for partially-connected ad hoc networks". Technical report, Duke University, 2000.

[17] www.dtnrg.org

[18] K. Fall, "A delay-tolerant network architecture for challenged Internets", Proceedings of SIGCOMM'03, August 2003.

[19] S. Bhattachargee, M. Ammar, E. Zegura, N. Shah, and Z. Fei. Application Layer Anycasting. In Proc. of the IEEE INFOCOM'97, 1997.

[20] D. Katabi and J. Wroclawski. "A Framework for Scalable Global IPAnycast (GIA)". Proceeding Applications, Technologies, Architectures, and Protocols for Computer Communication, Stockholm, Sweden, 2000.

[21] V. D. Park and J. P. Macher. "Anycast Routing for Mobile Networking". Proc. of the IEEE Military Communications Conference, volume 1, pages 1–5, 1999

[22] Zheng Xie, Jianxin Wang, Yuan Zheng and Songqiao hen, "A Novel Anycast Routing Algorithm in MANET", PACRIM IEEE, 2003.

[23] Jianxin Wang, Yuan zheiig, Weijia Jia, "An AODV Based Anycast Protocol in Mobile Ad Hoc Network", PACRIM IEEE, 2003.

[24] Thrasyvoulos Spyropoulos,Konstantinos Psounis, Cauligi S. Raghavendra "Spray and wait: an efficient routing scheme for intermittently connected mobile networks", Proceeding in ACM SIGCOMM 2005.

[25] S. Jain, K. Fall, and R. Patra. Routing in a delay tolerant network. ACM SIGCOMM 2004.

[26] W. Zhao et al. Multicasting in Delay Tolerant Networks: Semantic Models and Routing Algorithm. ACM SIGCOMM 2005.

[27] Yili Gong, Yongqiang Xiong, Qian Zhang, Zhensheng Zhang, Wenjie Wang, Zhiwei Xu. Anycast Routing in Delay Tolerant Networks. In IEEE Globecom'06, San Francisco, CA, Nov. 27 - Dec. 1, 2006

[28] Q. Ye, L. Cheng, M. Chuah, B. D. Davison, "Perfomance Comparison of Multicasting Approaches in DTN", Technical Report, 2006.

[29] Network Simulator 2, ns2. http://www.isi.edu/nsnam/ns/

[30] http://www.isi.edu/nsnam/ns/ns-documentation.

[31] http://www.isi.edu./nsnam/nam

APPENDIX

USER MANUAL

A User Manual

User manual provides the steps which enable a user to use the system. In this section we provides details required to run, simulation of, our adaptive anycast protocol in ns-2. All these are easy and simple.

A-1 Check out hardware

As far as hardware goes the minimum system requirements are recommended at the end of the performance testing and are as follows...
Minimum recommended system requirements:

- 500 MHz processor.
- 128 MB RAM.
- Monitor.
- Keyboard and Mouse.

A-2 Download and install NS-2

This part describes how to install NS2 on Red Hat Enterprise Edition(4 version). The NS2 version for this document is ns-2.29.

NS2 setup

1. Download NS-2 fromhttp://prdownloads.sourceforge.net/nsnam/ns-allinone-2.29.2.tar.gz?download.

2. Decompress the ns-allinone-2.29.rar.

3. Copy this folder under ~/root/

4. Change the path to ns-allinone-2.29/ns-2.29

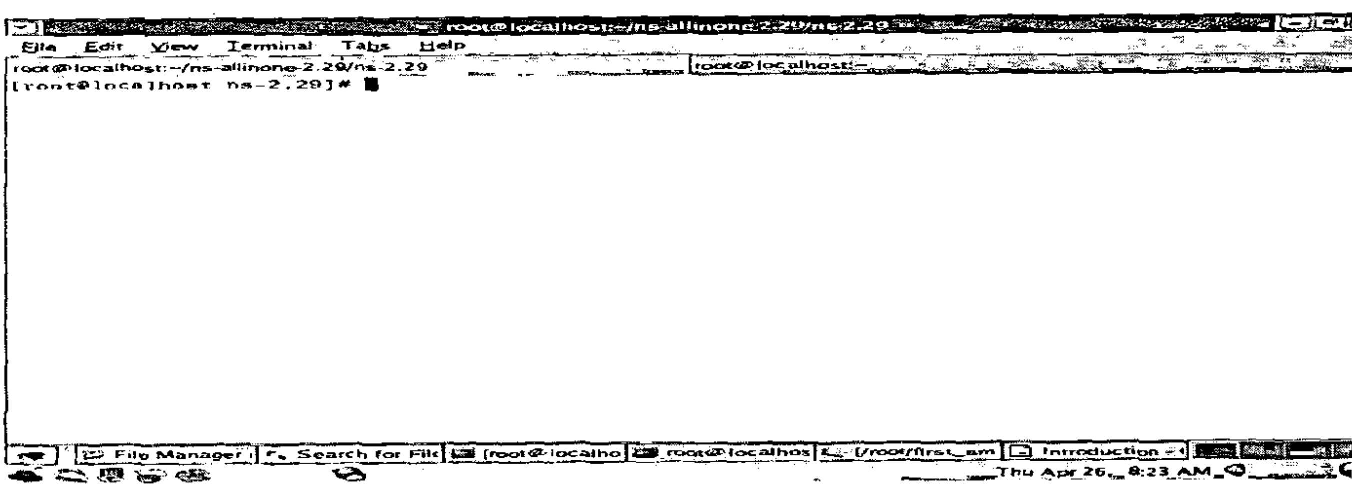

Figure A.1 Changing Path

5. Run the command "./configure"

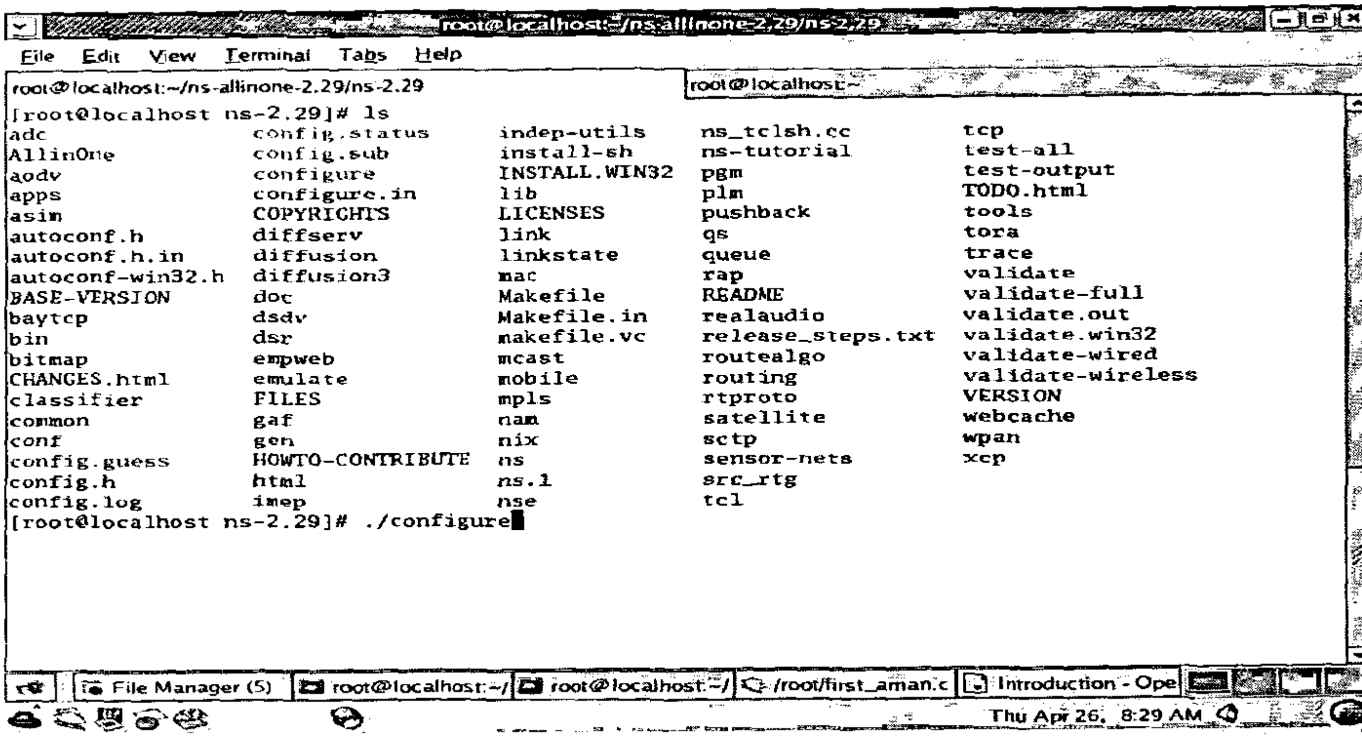

Figure A.2 Configure NS2

6. Now to compile ns-2 run make file

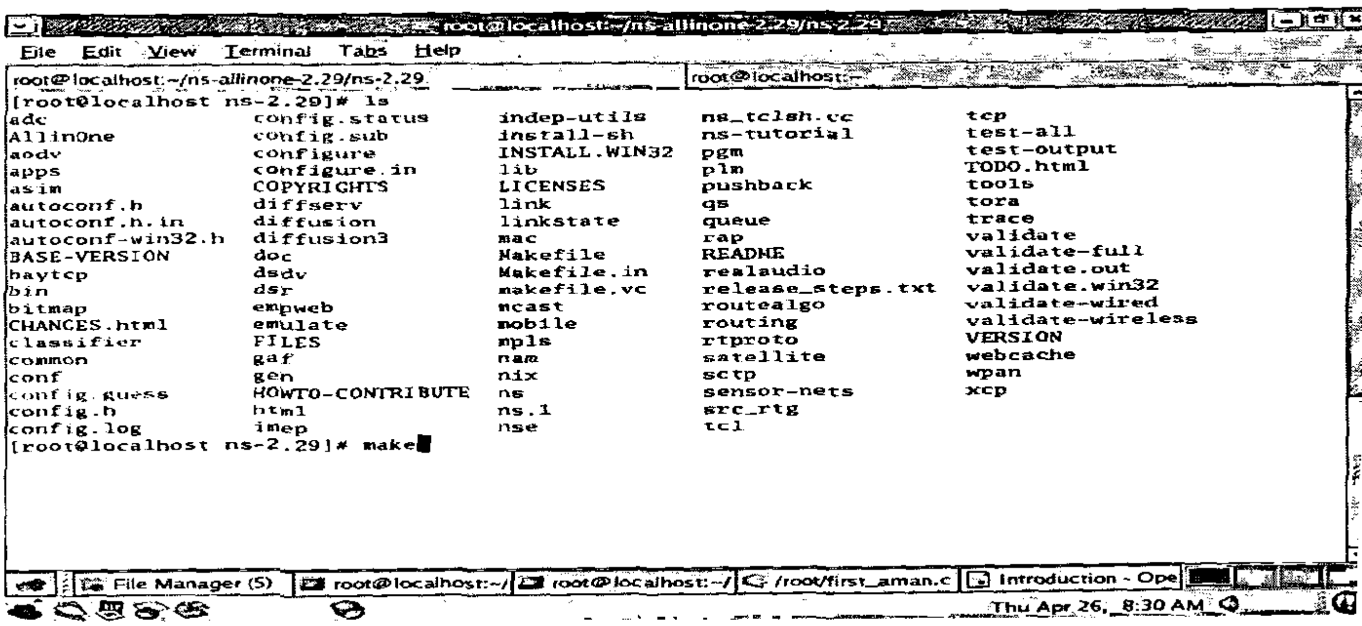

Fig3. Compile ns-2

7. Please be patient. It will take some time to finish the compilation.

8. When it is done, it should look like as follows.

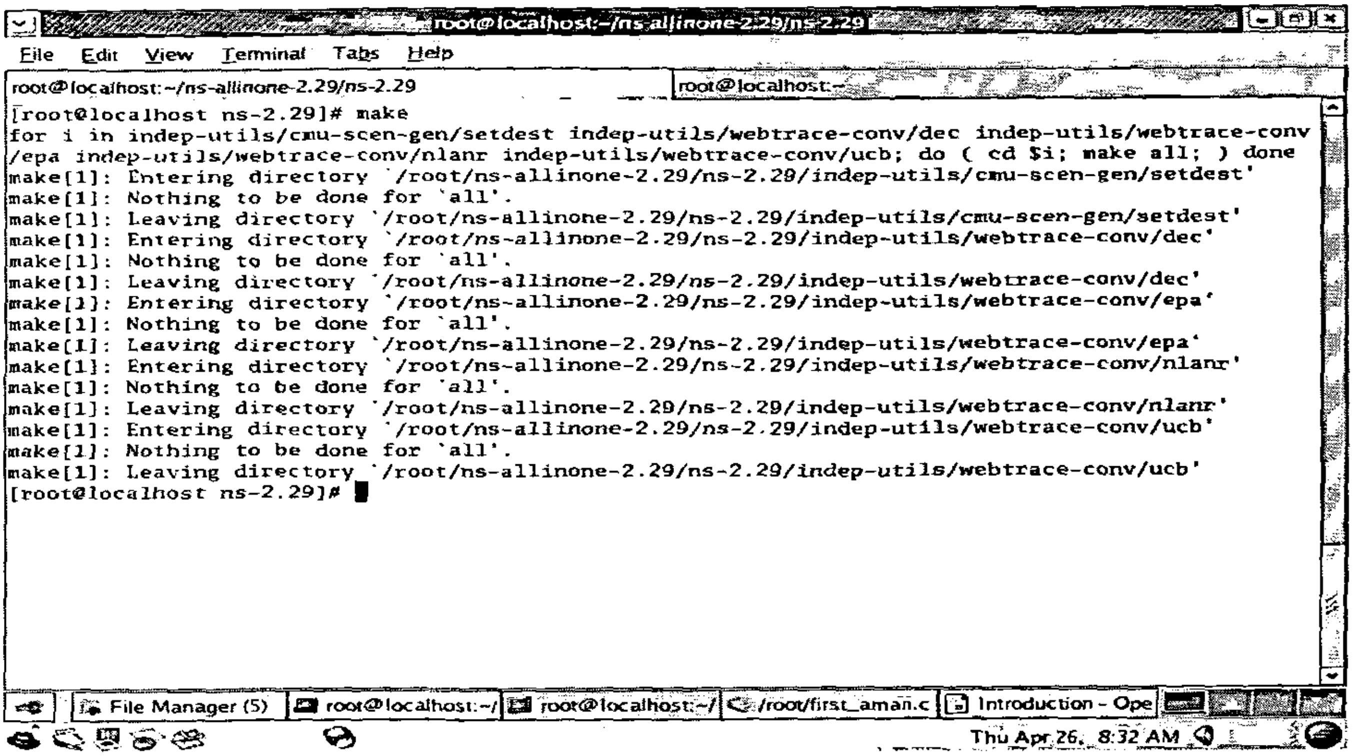

Figure A.4 Compilation Complete

9. Then set the variables path in ~/etc/profile of the variables PATH , LD_LIBRARY_PATH and TCL_LIBRARY

10. To make sure that you have successfully installed ns-2 and have set the paths correctly, run an example present ~/ns-2.29/ns-tutorial/examples/ by following command

$ ns example1a.tcl

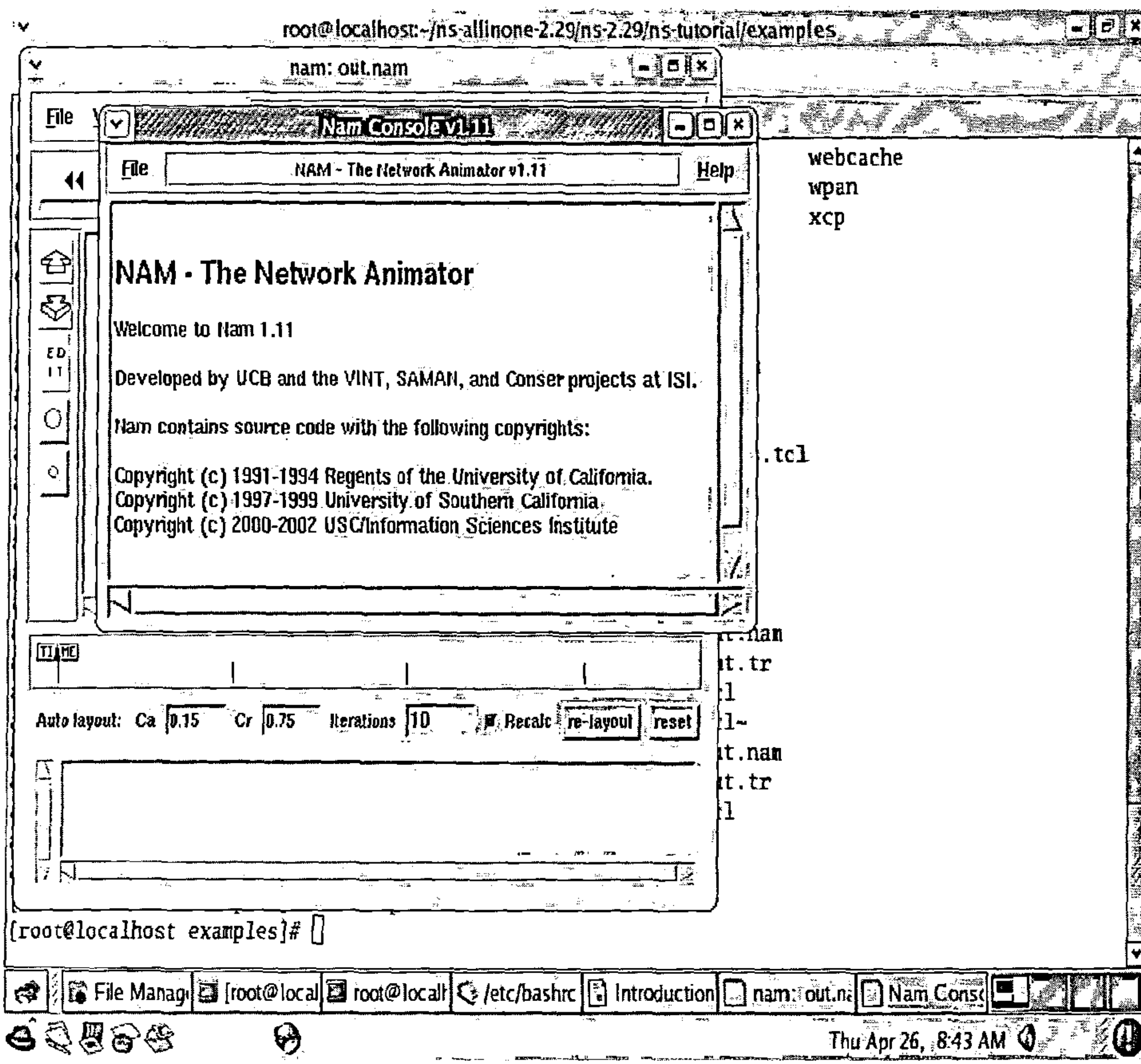

Figure A.5 Checking ns

11. If you see the error message like '§ns command not found'', no worry about this. Sometimes even you have setup the path, but it does not work. You can copy the ns.exe (nam.exe) to the same place as the simulation script.

To run our adaptive anycast protocol Simulation

1. Copy our anycast and dsr folder under ~/ns-2.29/
2. Copy our Makefile.in under ~/ns-2.29/
3. Repeat step 5 and 6.
4. Set the transmission rang
   I. Compile the threshold file in ~/ns-2.29/indep-utils/propagation.
   II. Use the command
   $./threshold –m TwoRayGround 110
   where 110 is distance range.

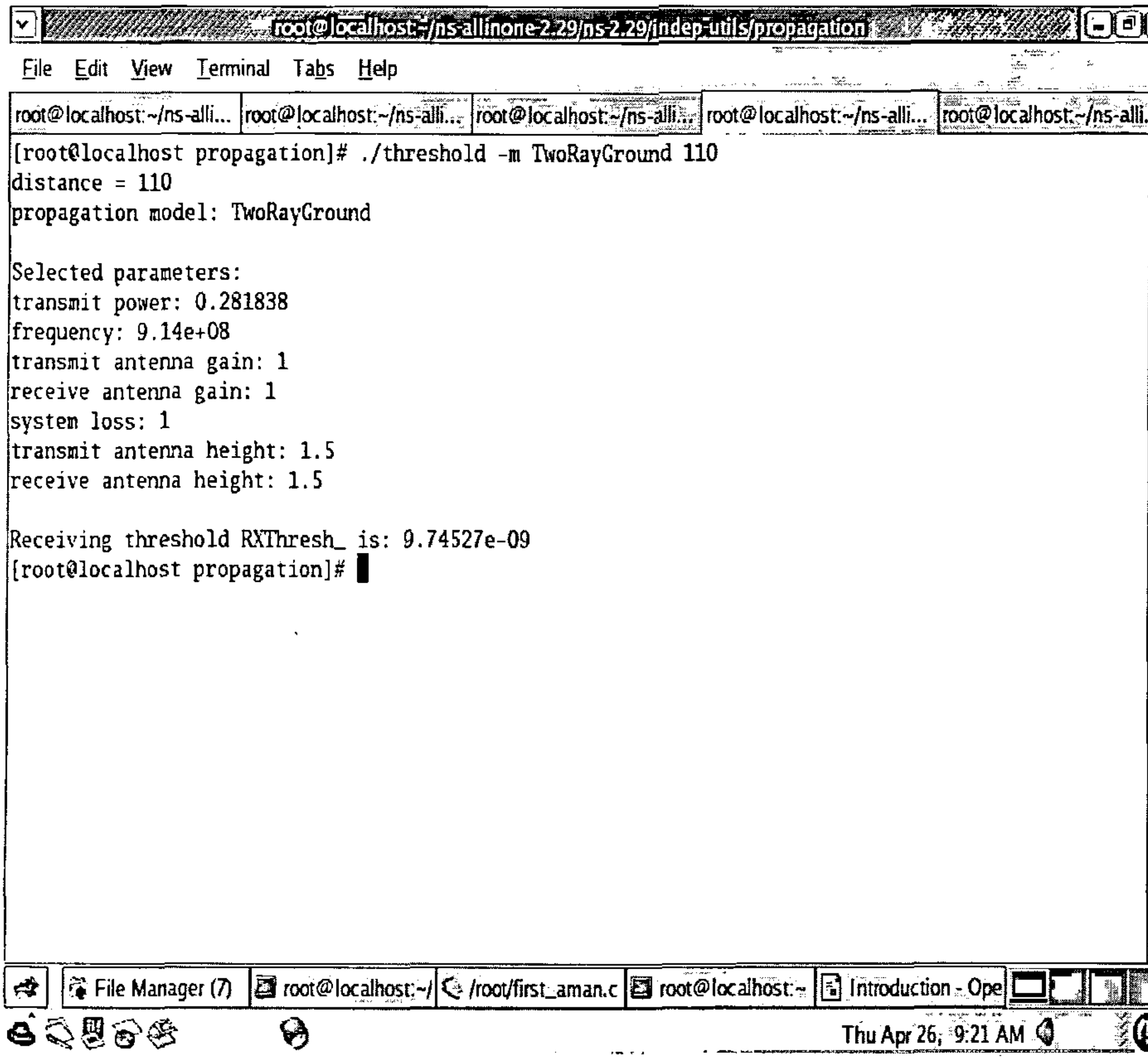

Fig.A.6 Setting the transmission range

2. Run the example tcl script in our anycast folder or make your own. (Change to ~/ns-allinone-2.29/ns-2.29/anycast/examples. Then run the command "ns ac_ldt_new1.tcl")

Our examples are of two types

- Files having names ac_ldt_new/RBF are RBF scheme enabled.
- Files having names ac_ldt_old/NF are NF scheme enabled.

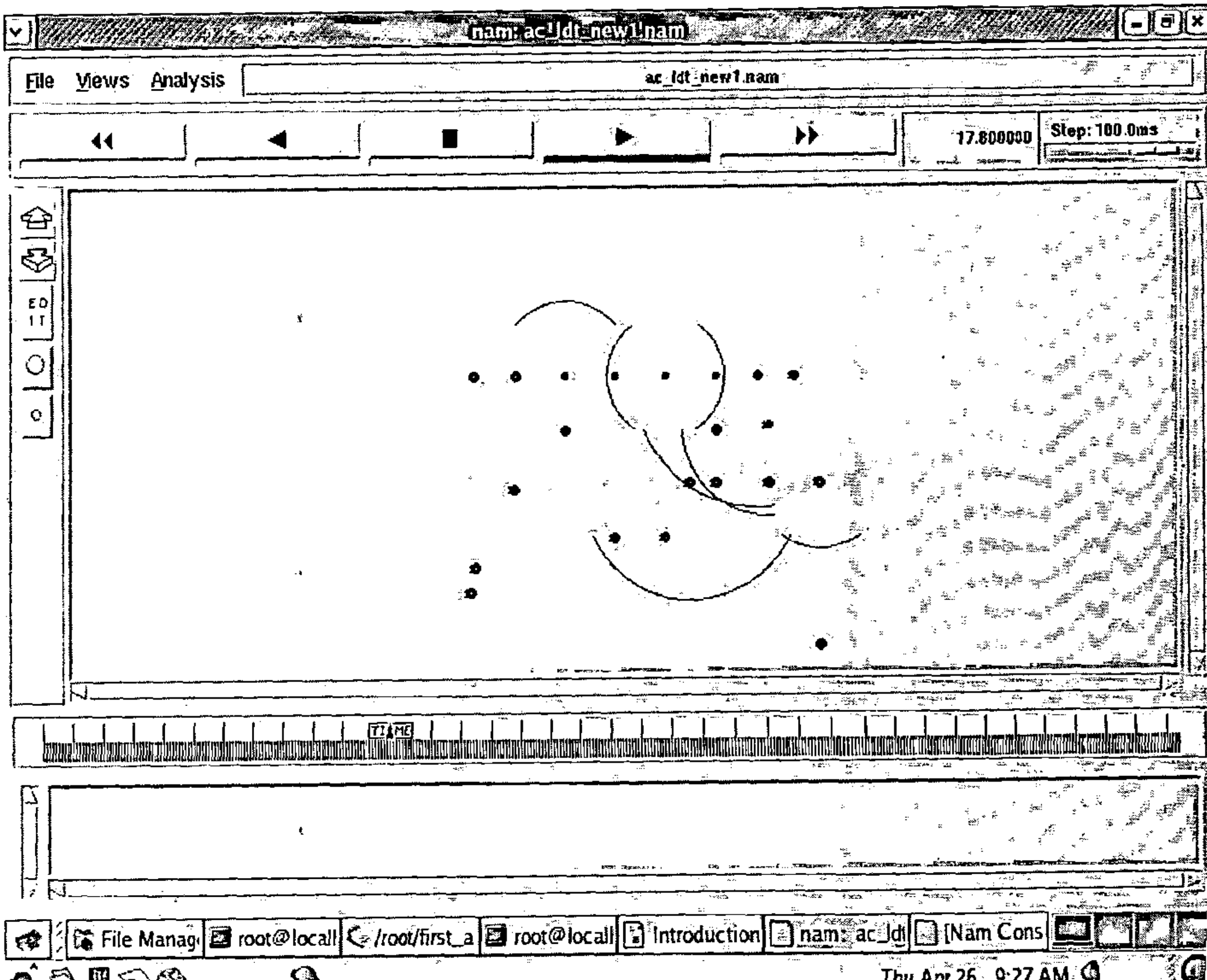

Figure A.7 Transmission of Data

PUBLICATION

Proceedings of the 2007 IEEE
International Conference on Integration Technology
March 20 - 24, 2007, Shenzhen, China

Adaptive Anycast: A New Anycast Protocol for Performance Improvement in Delay Tolerant Networks

Fazl-e-Hadi, Nadir Shah, Dr. Afaq Hussain Syed
Department of Computer Science
International Islamic University
Islamabad, Pakistan.
{fhadi76 & nadirshah82}@yahoo.com , drafaq@iiu.edu.pk

Dr. Mehboob Yasin
Department of Computer Science
COMSATS Institute of Information Technology
Islamabad, Pakistan
mmyasin@comsats.edu.pk

Abstract **- Delay/Disruption Tolerant Networks (DTNs) are characterized by frequent and long duration partitions and end-to-end connectivity may never be present between the source and the destination at the message origination time. Anycast is an important service used for many applications in DTN such as information exchange in hazards/crisis situation, recourse discovery etc. In this paper we study anycasting for a specific type of DTN, where the mobile nodes are sparsely distributed, communicating via low radio range, experience frequent and long duration partition and end-to-end path may not be present at the time of message generation. We also propose a Receiver Based Forwarding (RBF) scheme for anycasting in DTN, which considers the path length as well as the number of receivers in forwarding the anycast bundle to the next hop. NS simulation results show that the RBF performs better than the normal forwarding (NF) in term of data delivery ratio, average end-to-end delay and overall data efficiency because the probability of available receivers increases.**

Keywords - Delay/Disruption tolerant network, Anycast

I. INTRODUCTION

Delay tolerant networks (DTNs) are characterized by frequent and long duration partition. Moreover end-to-end connectivity between the source and destination may not be present at time of message generation. In these networks, packets (Bundles) are transferred in store-forward fashion. The DTN nodes connectivity may be opportunistic or periodical/predictable. These networks have various applications such as vehicular communication [1], mobile sensor network, smart dust, disaster recovery/military deployment, etc. [2].

In this paper we study the problem of anycasting in DTN. Anycast means to deliver the packet to any of the members in a group. There are various applications of anycast in DTN like:
(a) In disaster rescue field, in which people may want to find a doctor or fireman without knowing their locations and specific IDs [3],
(b) In battle fields a soldier may want to convey the surrounding information to any one of the command centers or a command center may want to deliver a particular message to any soldier among a group (squad)
(c) Offline content distribution like a student may want to get an article from any one of the libraries, and many other applications.

As in DTN, resources such as available bandwidth, storage and connectivity among the nodes are limited; therefore an efficient anycast service is necessary for supporting these applications.

Anycasting in the Internet and mobile ad hoc network (MANET) has been studied [4, 5, 6] but anycasting in DTN is different and more challenging due to the frequent partitions and long end-to-end delays. In this paper we propose a new adaptive anycasting approach for mobile nodes where the link availability is opportunistic and the approach is adaptive in nature by communicating with the underlying network for getting the current topological information. We also propose a new forwarding scheme called as Receiver Base Forwarding (RBF). In this scheme the source forward the packet to the next hop based on the path length as well as number of receivers reachable from that next hop. Through simulation we show that our proposed approach is dynamic in nature and also that the RBF outperforms the normal forwarding (NF) in tie case in term of average end-to-end delays, packet delivery ratio and the overall efficiency.

The rest of the paper is organized as follows: Section 2 describes the related work while Section 3 describes the proposed system models. We present the new adaptive anycast algorithm in section 4. Section 5 describes the simulation results and finally section 6 concludes the study with future work.

II. RELATED WORK

Anycast routing has been studied extensively in Internet and mobile ad hoc networks (MANETs). Zheng Xie et al. in [7] propose an anycast routing protocol in MANET which can balance the network load and improve network performance. Jinxin Wang et al. in [8] propose an anycast for IP flow in MANET based on an AODV protocol. But these approaches are not suitable for DTNs due to the frequent network partitions and long end-to-end delays. The architecture of DTN proposed by K.Fall in [9] is based on asynchronous message forward and operates as an overlay above the transport layer. Vahdat et al. in [2] developed a technique to deliver messages in the case where there is never connected path from source to destination or when a network partition exists at the time a message is originated. Jain et al. in [10] developed and compared several unicast routing algorithms in DTN.

Recently Yili Gong et al. analyzed Anycast semantic for TN and presented a metric named EMDDA (Expected ulti-Destination Delay for Anycast) [3]. They assumed that odes in the network were stationary. The connectivity among e nodes was the mobile devices that act as carrier to deliver essages for the nodes. The mobile devices do not generate essages themselves, i.e. message carriers are fixed. Also the oving patterns of these mobile carriers can be obtained.

We are interested in a specific type of DTN where the obile nodes are sparsely distributed and communicating via nited wireless radio range such that network experience equent and long duration partitions. All DTN nodes are oving with limited storage capacity, and connectivity among e nodes is opportunistic. A node may be a source, receiver or e carrier which carry the message and forward the message the opportunity arises. We also propose packet forwarding echanism named RBF, which improves the performance of nycasting in DTN. Qing Ye et al. in [11] use situation scovery mechanism provided by the underlying network yer for multicasting in DTN. It is a good effort and we use e situation awareness mechanism provided by underlying twork for the anycast service in DTN, with new packet rwarding mechanism.

III. SYSTEM MODELS

Network Model

As presented in [9], the DTN is an overlay network built on the underlying network working above the transport yer, which is based on asynchronous message forward. All des on which the DTN agent is implemented are the DTN des among the other nodes, the two neighbor DTN nodes ay be multi-hops away, the packet known as bundle in DTN transmitted at DTN layer (Bundle Layer) in store-forward shion. Fig.1 shows a simple DTN example presented in [11]. e storage capacity of each DTN node is limited and the cket may drop on the intermediate nodes due to buffer erflow. We ensured the custodian hop by hop transfer, oposed in [12] in which reliable packet delivery is ensured tween two hops.

Anycasting Model

Anycast means to deliver the packet to any of the members a group. Anycast in DTN is defined as bundle transmission any of the members in a group of DTN nodes. Every DTN de has a name and the translation between DTN name and derlying is done by DTN routing agent. Anycast source use explicit names of the anycast receivers as the destination dress.

IV. ADAPTIVE ANYCAST ALGORITHM

Situation Awareness

As DTN experiences frequent and long duration partitions. assume that underlying network is operational for long e so that caches are built up and sufficient knowledge about the network topological condition is available. To achieve situation awareness each DTN agent

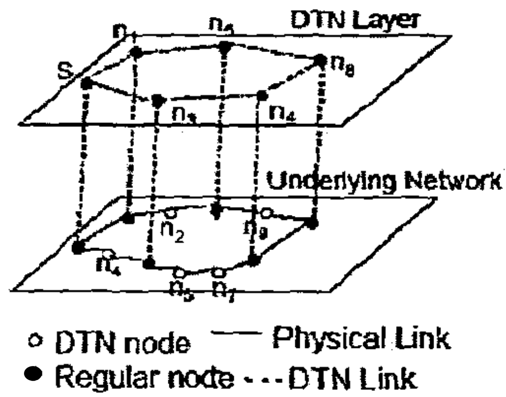

Fig. 1 A Simple example of DTN

periodically sends request to underlying routing agent for the current network condition, for which we use Dynamic Source Routing (DSR) [13] as an underlying routing protocol. The underlying routing agent responds with the current condition of the underlying network like discovered path from current node to the specified destinations.

B. Group joining and leaving

Any DTN node that wants to join anycast group sends a join request (JRqt), containing time interval during which it will be available. The join request is flooded so that it can reach the anycast source as early as possible. Similarly a member can explicitly drop membership by sending the drop information packet (DinfPkt), so that the anycast source will not consider it for more traffic. Each DTN anycast source updates its members list on receiving the joining or drop requests. The group joining model also confirms the TIM model proposed in [3].

C. Message Buffering

Each DTN node has limited buffer to store the in-transit packets. The packet is forwarded to the next hop and is removed from the buffer on receiving the acknowledgment from the next hop i.e., the custodian transmission is enabled to ensure the reliable delivery between two hops. Packets are dropped when the buffer overflows, depending upon the packet drop policy, for which we use FIFO policy.

D. Forwarding state

Each anycast bundle has the explicit list of receivers, so each DTN node forwards the packet according to its local knowledge about the underlying network.

E. Receiver Base Forwarding (RBF)

When a node wants to forward the packet it gets the current topological information from the underlying routing agent. The DTN agent sends a rout request to the underlying routing agent (RoutRqt); the rout request contains the anycast receiver list. The underlying routing agent sends the discovered paths information (from the current node to the receivers in the list) to the DTN agent. The DTN agent removes the currently unavailable outgoing links. The DTN agent finds the path to the closest anycast member. In a situation when there is more than one shortest path available to the anycast receivers, the normal forwarding scheme takes either the first shortest path or on the basis of node ID. In such situations we have proposed a receiver based forwarding scheme for anycast in DTN. This scheme selects the shortest path through which more receivers are reachable so that the probability of the receiver's availability increases. This results in an increase in the delivery ratio and decrease in the end-to-end delay. This improves its overall efficiency.

For example, a simple anycast Session is shown Fig. 2. Node 0 is a source and node 5, 7 and 9 are members of anycast. When node 0 wants to send anycast bundle to any of the anycast members then node 0 will send request message to underlying agent to trigger the network condition to anycast members. Node 0 will come to know about the paths to anycast members. Now node 0 will make mesh by combining these discovered paths to anycast members as shown in Fig. 2 (a). The dashed lines show available links. The node 0 will delete all those outgoing links that are currently not available, as shown in Fig. 2 (b).

The node 0 has two choices to forward the bundle, either to node 3 or to node 1. The normal forwarding algorithm solves the tie situation by forwarding the bundle to the first one or on the basis of node ID, but RBF will select the node through which more anycast receivers can be reached, so that the probability of available receivers will increase, therefore, node 0 will forward the bundle to node 1 as shown in Fig. 2 (c).

Along the message, the list of anycast relievers is also included. Similar operation will be done at each DTN node for each bundle. Node 1 gets paths to node 5 and node 7 and then forwards the bundle to node 4 as shown in Fig. 2(d). Node 4 finds the path to node 5 so it will forward the bundle to node 5 as shown in Fig. 2(e).

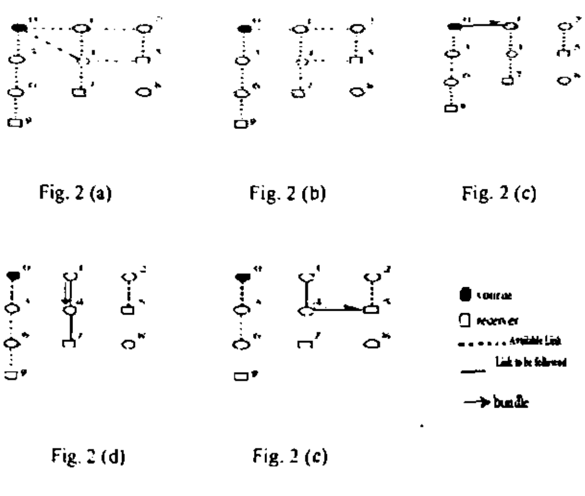

Fig. 2 A simple Example of Receiver Base Forwarding (RBF).

F. Bundle retransmission

DTN node checks periodically its local buffer for unacknowledged packets, provided its existence, they are retransmitted according to anycast receiver list. In order to reduce the overhead there are controlled retransmissions for a threshold. Once the acknowledgment is received, the packet will be deleted from the buffer or will be dropped in case of buffer overflow.

V. PERFORMANCE EVALUATION

For performance evaluation we implemented the proposed approach for anycasting in DTN with both NF and RBF in ns-2[14] simulator. We used following metric for the performance evaluation:

1) Message Delivery Ratio: It is defined as the ratio of total number of unique anycast bundles received by any anycast group member to the total number of bundles transmitted by the anycast source.

2) Data Efficiency: It is defined as the ratio of total number of unique anycast bundles received by any group member to the total traffic generated, both data bundles and control packets.

3) Average Message Delay: It is defined as the ratio of total delay for received anycast bundles to the total received anycast bundles.

We simulate a specific type of DTNs; in simulation we used 20 nodes over 800 x 800 area with 110 m transmission range. DSR [13] is used as routing approach for underlying routing in network. For situation awareness, we collaborate between DTN routing agent and underlying routing agent. MAC layer is IEEE 802.11.

We study one anycast session for 60 seconds. Node 1 is fixed as the anycast source. While the other DTN nodes randomly join the anycast group by sending join request. The anycast source sends the bundle at the rate of 1 bundle per second. The buffer capacity of each DTN node is 30 bundles. At every 2 second each DTN agent checks its buffer to see bundle waiting for retransmission. We study the performance by varying the link availability of each link uniformly for 10% to 90%.

Fig. 3 shows the average delivery ratio of adaptive anycast algorithm with both NF and RBF schemes. Varying the link availability from 10% to 90%, the delivery ratio increases. When the availability is low, RBF achieves better delivery ratio because the probability of receiver's availability increases. The delivery ratio of both schemes NF and RBF are almost equal at 90% or more link availability.

Fig. 4 shows that when link availability is low the overall data efficiency of RBF is high because the probability of number of receivers is high and if one receiver is not available, the bundle may be forwarded to another receiver, and no extra traffic is generated for retransmission and checking the underlying network condition. At high link availability the difference between NF and RBF data efficiency decreases.

Fig. 5 shows the average end-to-end delay performance using both NF and RBF. At low link availability the average end-to-end delay of both schemes abruptly increases due to the long duration partitions, which causes packets to be buffered for longer time. Similarly we noted that RBF experiences comparatively low delay due to the alternate receivers' availability and that the propagation delay is less than the long duration partition delay. At high link availability the average end-to-end delay of both schemes is minimal. Above the 90% link availability the average end-to- end delay of both schemes is almost equal.

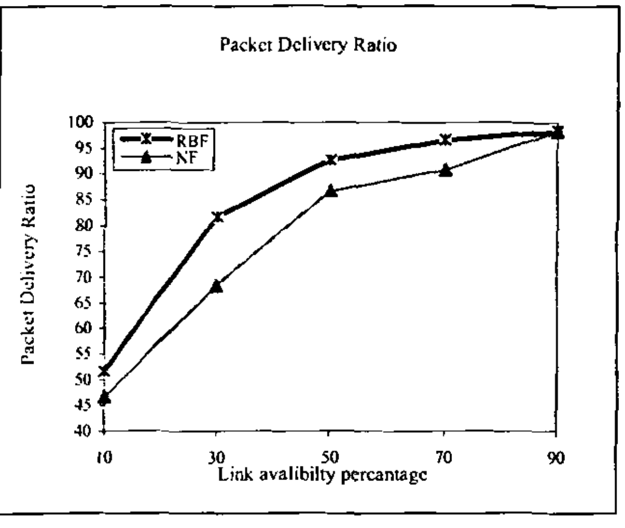

Fig. 3 Packet delivery ratio of Normal Forwarding (NF) and Receiver Base Forwarding (RBF).

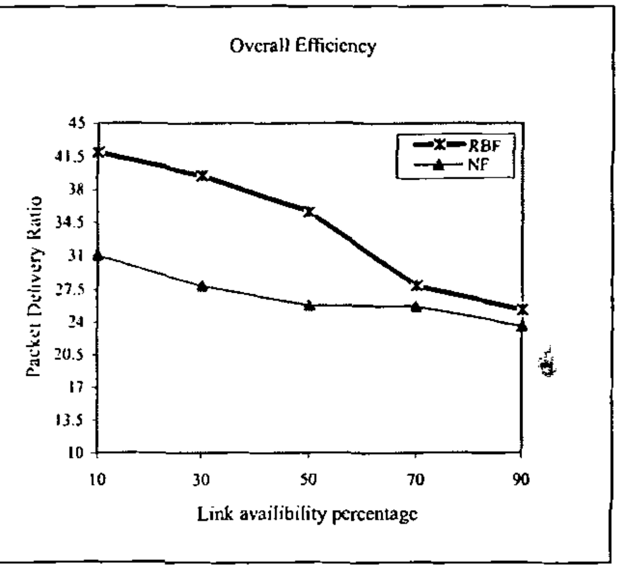

Fig. 4 Overall efficiency of Normal Forwarding (NF) and Receiver Base Forwarding (RBF).

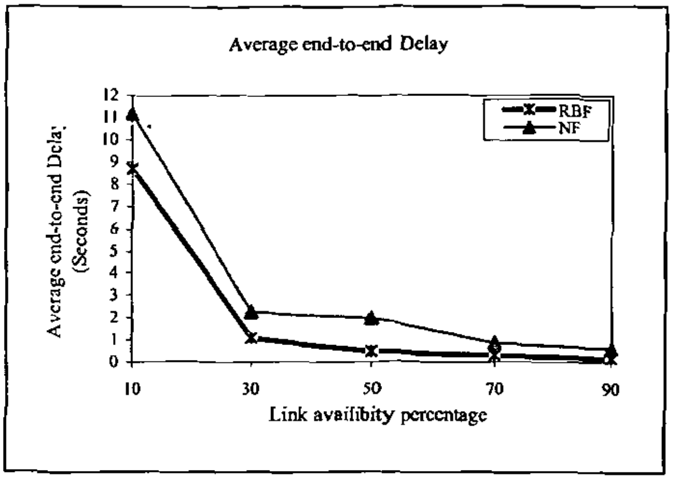

Fig. 5 Average end-to-end delay of Normal Forwarding (NF) and Receiver Base Forwarding (RBF).

V. CONCLUSION AND FUTURE WORK

In this paper we have studied anycasting for a specific type of DTN, where the mobile nodes are sparsely distributed, communicating via low radio range, experience frequent and long duration partition and end-to-end path may not be present at the time of message generation. We have also proposed an RBF scheme for anycasting in DTN, which consider the path length as well as the number of receivers in forwarding the anycast bundle to the next hop. NS simulation results show that the RBF performs better than the NF in terms of data delivery ratio, average end-to-end delay and overall data efficiency because the probability of available receivers increases. We are interested in extending our proposed anycast scheme to incorporate Geocasting in future work.

REFERENCES

[1] H. Wu, R. Fujimoto, and G. Riley. "Analytical models for data dissemination in vehicle-to-vehicle networks", In *IEEE, VTC 2004 Fall.*

[2] A.Vahdat and D. Becker. "Epidemic routing for partially- connected ad hoc networks", *Technical report, Duke University*, 2000.

[3] Yili Gong, Yongqiang Xiong, Qian Zhang, Zhensheng Zhang, Wenjie Wang, Zhiwei Xu. "Anycast Routing in Delay Tolerant Networks", In *IEEE Globecom'06, San Francisco, CA, Nov. 27 - Dec. 1, 2006*

[4] D. Katabi and J. Wroclawski. "A Framework for Scalable lobal IPAnycast (GIA)", In *Proc. of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, Stockholm, Sweden, 2000.

[5] S. Bhattachargee, M. Ammar, E. Zegura, N. Shah, and Z. Fei. "Application Layer Anycasting", In *Proc. of the IEEE INFOCOM'97*, 1997.

[6] V. D. Park and J. P. Macher. "Anycast Routing for Mobile Networking", In *Proc. of the IEEE Military Communications Conference*, volume 1, pages 1–5, 1999.

[7] Zheng Xie, Jianxin Wang, Yuan Zheng and Songqiao hen, "A Novel Anycast Routing Algorithm in MANET", IEEE, 2003.

[8] Jianxin Wang, Yuan zheiig, Weijia Jia, "An AODV Based Anycast Protocol in Mobile Ad Hoc Network", IEEE, 2003.

[9] K.Fall, "delay-tolerant network architecture for challenged Internets", *Proceedings of SIGCOMM'03*, August 2003.

] S. Jain, K. Fall, and R. Patra. "Routing in a delay tolerant network" in ACM *Sigcomm 2004*, Portland, OR, 2004.
] Q. Ye, L. Cheng, M. Chuah, B. D. Davison, "On-Demand Situation aware Multicasting in DTNs", Proceedings of IEEE VTC, spring 2006.
[12] K. Fall, "Messaging in difficult environments," Intel Research Berkeley, IRB-*TR*-04-019, Dec. 27, 2004.
[13] D.B. Johnson and D.A. Maltz, "Dynamic source routing in ad hoc wireless networks," in *Mobile Computing*, edited by T. Imielinski and H.Korth, chapter 5, pp.153-181, Kluwer Academic Publishers, 1996.
[14] Network Simulator 2, ns2. http://www.isi.edu/nsnam/ns