A MODEL FOR HEADER COMPRESSION CONTEXT TRANSFER IN CELLULAR IP

Dissertation submitted to Jawaharlal Nehru University
in partial fulfillment of the requirements
for the award of the degree of

MASTER OF TECHNOLOGY
In
COMPUTER SCIENCE & TECHNOLOGY

BY

MOHAMMAD ANBAR

Under the Supervision of

Dr. D.P.Vidyarthi

SCHOOL OF COMPUTER AND SYSTEMS SCIENCES
JAWAHARLAL NEHRU UNIVERSITY
NEW DELHI-110067 (INDIA)

JULY, 2007

जवाहरलाल नेहरू विश्वविद्यालय

SCHOOL OF COMPUTER AND SYSTEMS SCIENCES

JAWAHARLAL NEHRU UNIVERSITY

NEW DELHI-110067 (INDIA)

CERTIFICATE

This is to certify that the dissertation titled **"A Model for Header Compression Context Transfer in Cellular IP"**, which is being submitted by **Mr. Mohammad Anbar** to the **School of Computer & Systems Sciences, Jawaharlal Nehru University, New Delhi,** in partial fulfillment of the requirements for the award of **Master of Technology in Computer Science & Technology** is a bonafide work carried out by him under the supervision of **Dr. D.P. Vidyarthi**.

The matter embodied in the dissertation has not been submitted for the award of any other degree or diploma.

Mohammad Anbar
(Student)

24/9
Prof Parimala N:
Dean
School of Compu[illegible] & Systems Sciences
JAWAH[illegible] UNIVERSITY
N[illegible] 11 067

Dean, SC & SS
Jawaharlal Nehru University
New Delhi-110067

Vidyarthi 12/7/07
Dr. D.P. Vidyarthi
(Supervisor)
SC & SS
Jawaharlal Nehru University
New Delhi-110067

Dr. D.P. VIDYARTHI
Associate Professor
School of Computer and System Sciences
Jawaharlal Nehru University, New Delhi-67

Dedicated to

My Dear Parents...

ACKNOWLEDGEMENT

An endeavor over a long period can be successful only with the advice and support of many well-wishers. I take this opportunity to express my gratitude and appreciation to all of them.

I pour out my heartfelt thanks to **Dr. D.P. Vidyarthi** for his useful and perfect guidance by giving timely suggestions throughout the tenure of my project, and also for his continuous supervision and valuable guidance for the improvement and completion of my project.

I would also like to extend my thanks to the Dean, Faculty Members and Technical Staff of SC&SS, JNU. My special thanks to all my classmates, friends and family members for their continuous support, inspiration and encouragement without which this project would not have been a success.

Mohammad Anbar

ABSTRACT

Mobility management is one of the most important issues in cellular network as the users have to move frequently in their day to day work. This issue (mobility management) is suggested within Mobile Computing environment in which most of the solutions of the mobility management have been worked upon. As a result, an important protocol that works in a good manner in sense of providing the user all the services while roaming between two different networks is the Mobile IP protocol. It works very well for the mobile users. A problem with Mobile IP protocol is that it is not able to manage the mobility of users that move within the small geographical areas called "cells". Mobile IP protocol stops operating at the micro level of mobility. Eventually, another solution was provided that can manage the mobility of the users that move between the small geographical areas. The solution is a new protocol that works at the micro level of mobility resulting in a Cellular IP protocol.

This protocol as a solution of micro level of mobility could give a good contribution in mobility management. Though Cellular IP came into picture, still there are many problems that are to be addressed by the research community. One of these problems is how to provide a good Quality of Service (QoS) in sense of bandwidth and time delay in Cellular IP networks. We have proposed a solution to this problem by addressing the Header Compression technique to save the bandwidth of the network. In this the complete header will be compressed in such a way that the decompressor at the other side will be able to construct the compressed header. In this way we avoid sending redundant data repeatedly. The other problem is if the Mobile Host wishes to move to another sub-network (to another cell) within the same network. The classical method to deal with this is to repeat the compression and decompression procedure every time the Mobile Host handoffs to another cell. Our proposal, in this case, is to transfer the Context of Header Compression (Header Compression Context) from the Base Station of the Mobile Host to the Base Station in which the Mobile Host is approaching. The Header Compression

Context is established during compression and decompression procedure. The best scheme for the same is three states Header Compression Context.

By using the Context Transfer of the Header Compression we can save the time as the Mobile Host needs only to initiate a new session with the new Base Station to which it has moved without reestablishing the Header Compression Context. It is so because the Context has already been transferred to the new Base Station.

The chapter one of this dissertation contains an introduction about Mobile Computing, Mobile IP and Cellular IP in general. Solutions for mobility management and a brief idea about Header Compression, Context Transfer and there benefits in Mobile Computing also find place in this chapter.

In chapter two we elaborated Mobile IP and Cellular IP protocols and explained every operation that may takes place in these two protocols and also we elaborated Header Compression technique and Header Compression Context Transfer.

All of the operations mentioned above have been explained in the state diagram in chapter three.

Chapter four contains simulation experiments for Bandwidth and Context Transfer.

Finally a conclusion is drawn about the experimental result presented in chapter four.

TABLE OF CONTENTS

Page No

Certificate i

Dedication ii

Acknowledgment iii

Abstract iv

Table of contents vi

1. Introduction **1**

1.1 Mobile Computing Functionality 1

1.1.1 Mobile Computing Devices 4

1.1.2 Developing Mobile Computing Applications 4

1.2 Mobile IP 4

1.2.1Basic Mobile IP 4

1.2.2 Benefits of Mobile IP protocol 5

1.3 Cellular IP 5

1.4 IP Header Compression 7

1.5 Context Transfer 8

1.6 Organization of the Thesis 8

2. Mobile IP and Cellular IP **10**

2.1 Components of Mobile IP protocol 10

2.2 How does Mobile IP work 11

2.3 Operations in Mobile IP 12

2.3.1 Discovery 12

2.3.2 Registration 13

2.3.3 Tunneling 14

2.4 Optimization in Mobile IP protocol 16

2.5 Cellular IP 18

2.6 Cellular IP Model 19

2.7 Operations in Cellular IP networks 21

2.7.1 Routing 21

2.7.2 Handoff 22

2.7.2.1 Hard Handoff 22

2.7.2.2 Soft Handoff 24

2.7.3 Paging 24

2.8 IP Header Compression 26

2.8.1 The need 26

2.8.2 Header Compression and link efficiency 27

2.9 Compression/Decompression process 30

2.9.1 Compression process 30

2.9.2 Decompression process 31

2.10 Context Transfer 32

2.10.1 Context Definition 32

2.10.2 Context Transfer Definition 33

2.10.3 Need for Context Transfer 33

2.10.4 Context Transfer process 33

2.10.5 Three state Header Compression Context 34

3. The Proposed Model **36**

3.1 Analysis of Cellular IP Model 36

3.2 Explanation of the state diagram 39

4. Simulation Experiment **41**

4.1 Probability of error during Header Compression 41

4.2 Time Compression with context transfer and without context transfer with varying packet size 43

4.3 Simulation using NS-2 simulator 44

4.3.1 Design of NS-2 simulator 44

4.3.2 What is TCL language? 45

4.3.3 Nodes and packet forwarding 45

4.3.4 The classifier 47

4.3.5 Mobile Networking in NS 47

4.5.3.1 Creating wireless topology 48

4.3.5.2 Creating Node Movements 49

4.3.6 Creating output files for X-graph 49

4.3.7 Using NS-2 Simulator for Cellular IP Model 50

5. Conclusion 56

References **58**

Chapter 1

Introduction

The communication infrastructure around us has developed drastically, as a result of the human activities in the thrust of faster communication and data transfer. So besides the computing environments that are known to all of us another environment around us is the communication environment known as Mobile Computing. It is known as a computing environment over physical mobility for both physical and logical computing entities on move. Physical entities that we are talking about include computers, phone, etc. that change their locations while the user is moving. Logical entities can be considered as the programs and applications that the user is running or the mobile devices are supporting while moving with the physical entities. [1]

From system point of view Mobile Computing can be considered as a system which gives its user the ability to perform a task like contact another users, download data or perform any other task from any location using any computing device (laptop, mobile set, etc.) [1]. Mobile Computing can be generalized by making the communication bearer to spread over both wired and wireless media.

The access to information and data resources through Mobile Computing is necessary for optimal use of resource and increased productivity. [1]

1.1 Mobile Computing Functionality

Mobile Computing defined above, must provide many functionalities, as listed below [1].

(i) User mobility

This functionality enables the mobile user to move from one location to another and to be able to use the same service across the network whether it is home network or remote network.

(ii) Network Mobility

A good example here is Mobile IP protocol. In this case the mobile user is able to move from one network to another and yet is able to use mobile services when he gets into another network which is different from the home network. The other network in this case is called foreign network.

(iii) Bearer mobility

Depending on both the previous functionalities the mobile user can change its location within his home network as well as he can change his network and according to that if one service he was using in his home network is not available in the new network then the mobile computing environment should be able to provide the functionality known as "bearer mobility" in which the mobile user is able to change the bearer in the new network. For example, a user uses WAP bearer in his home network, and moves to another network where WAP is not supported. So he will switch over to another bearer like voice or SMS bearer to access the same application he was using.

(iv) Device mobility

This functionality is essential for mobile computing. Suppose a user is using his desktop to access an application and he is to move from his office to another place. While moving on the way he should access the application he was working with. So in this case he will definitely change the desktop to laptop to be able to move with, so changing the device must not affect his access to the application he was using.

(v) Session mobility

As the name indicates, the mobility should be applicable on the session that the user is doing. It means for the ongoing session, the session should move from one device to another. If the user, for example, changes his device which was laptop and uses desktop, then the unfinished session should move to the desktop for the user to be able to continue his session successfully.

(vi) Service mobility

When a user of a desktop is using Internet and trying to do a task he may need to use another service so he may simply shift to this service using the task bar without any difficulty. This functionality must be supported in Mobile Computing environment because the user also may need to move from one service to another while using a mobile device.

(vii) Host mobility

When a host is mobile, it results in the change of IP addresses when he moves from one network to another. We say that the IP mobility needs to be taken care of in case of host mobility.

After the discussion on the mobile computing functionalities, it is logically divided into the following segments:

a) User device

A fixed device like desktop computer in an office or a portable device like mobile phone.

b) Network

This is the main segment (component) of mobile computing because whenever a user is mobile he is going to use different networks at different places at different times.

c) Gateway

The gateway is required to connect the mobile network to Internet and to interface different transport bearers. As an example of the functionality of the gateway, it is to convert the analog signals generated by pressing the keys of a telephone into digital signals (digital data) by the IVR (Interactive Voice Response) of the gateway to interface with a computer application.

d) Middleware

This term is used in many fields. The main definition of it, form Mobile Computing point of view, is connectivity software consisting of a set of enabling services that allow multiple processes running on one or more machines to interact across a network.

e) Content

The content can be either a personal or corporate content and the origin server will have some means to access the database and storage services.

1.1.1 Mobile Computing Devices

We have two categories of devices that can be used in Mobile Computing. They are Computing devices and Communication devices. Computing device which is used in Mobile Computing can be a desktop computer, laptop computer or a palmtop computer. Communication device can be either a fixed telephone or a mobile telephone.

Nowadays both types of devices are integrated because they are used together. Sometimes fixed telephone is used and sometimes we use mobile telephone or both together. Any user of Mobile Computing device may face challenges like the inconsistent in the interaction from one device to another. [1]

1.1.2 Developing Mobile Computing Applications

The main thing which we must do in any postal system is to improve the user mobility. To improve any Mobile Computing environment is to adapt the context and behaviors of the applications in order to suit the current environment (Mobile Computing environment). The adaptation of context and behavior can be done through many ways; one way is to make the adaptation handled by the behavior of the middleware (software connectivity discussed previously). [19]

Developing a new mobile system will differ from making an existing application mobile. [1]

1.2 Mobile IP

Mobile IP protocol is a protocol that gives any mobile user in Mobile Computing environment the ability to roam beyond his home network. While movement, the IP datagrams will be routed to this user so that the session can be maintained in spite of the physical movement of the user between his network which is called home network and the other network which is called foreign network. [1]

1.2.1 Basic mobile IP

The basic architecture of mobile IP consists of three types of entities. These entities work together in integrated form and are as follows.

- **Mobile Host (MH)**
- **Home Agent (HA)**
- **Foreign Agent (FA)**

In this architecture, FA and HA are connected to the wired Internet and are responsible for providing all services to MHs as well as FA operates as a gateway between the wired and wireless side of the network.

Making comparison between cellular networks and Mobile IP networks we see that FA represents a base station that covers an area called cell. MH attaches itself to the nearest FA in Mobile IP networks while in cellular networks MH attaches himself to the base station of strongest signal. [5]

1.2.2 Benefits of Mobile IP protocol

The best and most useful solution, when the mobility of a user is required (desired), is Mobile IP protocol. It gives the user the ability to maintain a single address for the whole transitions between networks and network media, though Mobile IP maintains two addresses: home address and foreign address. However the user is considered to be having (maintaining) a single address. [2]

In general Mobile IP protocol is useful in cellular environments as well as wireless LAN situations that may allow roaming.

In Mobile IP networks each MH has his own home address and point of attachment with Internet. The only devices that need to be aware of the mobility of the user are mobile nodes and the router that serves the user and forwards the packets coming to the user depending on the topology of the network.

1.3 Cellular IP

Cellular IP is the protocol that manages the micro-level of the mobility. It consists of the following entities.

- **Base Station**
- **Gateway**
- **Mobile Node**

Cellular IP is the most useful mobility protocol as it can manage the mobility of users who frequently migrate. In this protocol, nodes maintain two kinds of caches; one is used for location management and the other is used for routing management.

Cellular IP protocol can distinguish between active nodes (nodes that send and receive data) and idle nodes (nodes that don't send or receive). It maintains the position of "idle" mobile in paging cache. This point is very important in Cellular IP environment where it can pinpoint "Idle" mobile quickly and efficiently by using this paging cache. On the other hand this approach is very beneficial because it can accommodate a large number of users attached to the network without overloading the location management system.

For active hosts, their positions are maintained in distributed routing cache. It dynamically refreshes the routing state in response to the handoff of active hosts. Thus using distributed location management and routing adds many benefits in Cellular IP. For example simple and low cost implementation of Internet host mobility without encapsulations or address space allocation. [6]

To consider Cellular IP networks and Cellular networks there are many features that Cellular networks offer to Cellular IP networks that can enhance the performance of Cellular IP networks. Maintaining the properties like flexibility, scalability and robustness that characterize all IP-based networks adds the to Cellular IP networks.

But also applying Cellular techniques to Cellular IP networks may impose some difficulties. Cellular telephony systems rely on circuit switching that requires connection before the establishment of the connection, but in Cellular IP networks there is packet switching where routing on per packet basis is performed. In spite of difficulties mobility management and handoff techniques found in Cellular networks are applied.

1.4 IP Header Compression

The need of number of users in any network requires more bandwidth and as a result services and consumers of these applications compete for the bandwidth available in the network. This becomes very important for operators to offer a high quality of services in order to attract more customers and encourage them to use their networks providing higher Average Revenue Per User (ARPU). This is the problem faced in most of the networks.

A brief look on wireless networks shows that there is a high bit error rate where data are highly prone to wireless communications. It leads to the difficulty in providing bandwidth required by the users, thus the available resources must be used as efficiently as possible.

The above conditions and requirements need a solution through which the bandwidth be available to all the users in any network maintaining a good quality of service. The best solution for this is using Header Compression technique which is an effective method to reduce the large overhead when transmitting the voice packets over wireless links. The studies show that Header Compression can reduce the load on the wireless link about 50-70 %. [4]

Another benefit with Header Compression is as follows. In TCP/IP protocol, a packet data is sent and in every layer a header is added to this packet. Thus a final packet, to be sent, becomes a large packet consuming bandwidth and also imposing time delay on the network. This problem can be solved or at least reduced with Header Compression technique. By using Header Compression technique abbreviated as (HC) we can compress the header of TCP/IP packet with any scheme to avoid sending redundant data. [4]

The principle of header compression is to send a packet with full header, and subsequent compressed headers refer to the context established by full header and may contain incremental changes to the context. [4]

1.5 Context Transfer

Context transfer means that we transfer the services that have already been established in one subnet to another subnet. This approach aims to reduce the time required to establish the services again.

A Context Transfer protocol aims to minimize the impact of transport/routing/security related services on the handover performance. This protocol will result in a quick re-establishment of Context Transfer candidate services at the new domain. It would also contribute to the seamless operation of application streams. [7]

Finally, the Context Transfer Protocol (CTP) is an end-to-end data transport protocol that supports data processing and improves the quality of service in the network in which it is applied. [7]

1.6 Organization of the Thesis

The dissertation consists of the five chapters arranged as follows.

Chapter one discusses about Mobile Computing in general, Applications of Mobile Computing, a brief idea about Mobile IP, Cellular IP, Header Compression and Context Transfer are briefed in this chapter.

Chapter two elaborates in detail about Mobile IP and Cellular IP protocols and also explains the operations that are included in these two protocols. It mentions importance of header compression in wireless communication along with the benefits of header compression. This chapter points out context transfer, its importance and benefits in Mobile Computing.
Chapter three discusses the proposed model. It takes the automation of the Model and discusses the outcome of this Model.

Chapter four talks about the simulation experiment carried out using network simulator NS-2. It also briefs about this simulator and its abilities and then discusses the results received from the simulation experiment.

Chapter five contains the concluding remarks about our work and results. It contains the benefits of our model in mobile computing. Future work is also pointed out in this chapter.

Chapter 2

Mobile IP and Cellular IP

In chapter one, we discussed that Mobile Computing Environment provides the user continuous access to data and services in a state of mobility from one network to another network without losing the connection. This principle differs from the other principle which is the portable computing environment in which the user moves from one location to another in the same network.

Suppose a user of TCP/IP network connects to one sub-network. This connection requires source IP address, source TCP port, target IP address and target TCP port. The user in current state connects to Internet and uses the services offered from this sub-network. Now there is a need that the user must change his point of attachment to another sub network while still using the same services and connected to the Internet. The way to do so is by using the "Mobile Computing" technology. The term mobile refers that while the user is connected to applications across the Internet and the user's point of attachment changes dynamically; all connections are maintained despite the change in underlying network properties. This principle is similar to the principle of known cellular networks with the difference that in cellular networks the user handoffs from one cell to another cell and here the point of attachment is the base station that controls the cell in which the user currently exists.[1]

2.1 Components of Mobile IP Protocol

As mentioned in chapter one, there are three entities in Mobile IP protocol shown in figure2.1 [5, 17].

Mobile Host which is a user with computing device that changes its point of attachment from one sub-network to another.

Home Agent this entity is a router found in user's home network and it is responsible of providing all services to the users in this sub-network where it tunnels datagrams (explained in Sec.2.3.3) and delivers them to the mobile node when the later is away from his home network. Also home Agent is responsible for maintaining current location information for the Mobile Node.

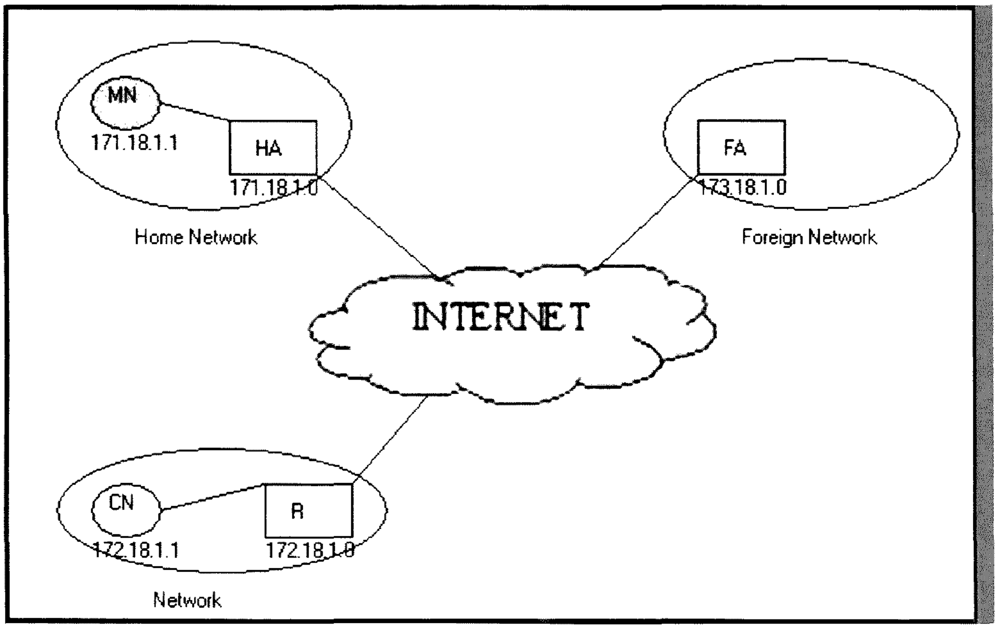


Figure 2.1 Mobile IP protocol components

Foreign Agent is found in the visited network and is responsible for providing routing services to the user who registered himself in this sub- network, where the foreign agent de-tunnels the datagrams that they had been tunneled previously and delivers them to the mobile node (user).In case the user sends datagrams, Foreign Agent may serve as a default router for registered Mobile Nodes.

2.2 How does Mobile IP work?

Mobile IP depends on two addresses during its movement. These two addresses are the home address and the care of address. The home address is the identity of the host and is static address. Also the host is known in its home network by the home address. The care of address is changed when the host changes its point of attachment. When the Mobile Node moves it will register its care-of-address with its Home Agent. Now if there are data to be sent to this host, the Home Agent will forward these data to the Foreign Agent in the foreign network in which the host exists, and this forwarding will be done based on the care of address. When the packet is to be delivered to this host there should be modifications at the fields of packet header in which the field that contains the destination address must be changed and put the destination address as a care-of-address. [1, 15].

2.3 Operations in Mobile IP

The operations of Mobile IP are elaborated as follows.

2.3.1 Discovery

This operation in Mobile IP networks uses ICMP protocol (Internet Control Message Protocol) which is integrated with IP protocol. ICMP messages are used for the purposes listed below.

- Announce network errors.
- Announce network congestion.

Discovery function (or Agent Discovery) is a double side operation. Through the Agent which is responsible of one network, it is discovered that whether any new Mobile Host (MH) entered into its network. On the other hand the Mobile Host can know whether it is in a foreign network.

The discovery procedure takes place as follows.

The Agent of the network periodically issues (sends) advertisement. This advertisement is an ICMP message, in that the Agent tells the hosts from another network that it can have them in its network and serve them. On the other side the Mobile Host receives the message and then the important discovery procedure takes place. The Mobile Host starts comparing between the IP addresses of the Agent and its IP address. It compares the network portion of the two IP addresses. The decision is taken by the Mobile Host depending on this comparison. If the two previous parts are same then it decides that it is in its home network otherwise it discovers that it is in a foreign network under the responsibility of another Agent.

In every advertisement message, an Agent sends, there may be information about the default routers (Agents) that Mobile Host can register with and also information about one or more care-of-addresses. [1]

2.3.2 Registration

After completion of discovery procedure the Mobile Host will be granted a care-of-address from the foreign network. This care-of-address needs to be registered with the home agent. The registration procedure is as follows [5], as depicted in figure2.2.

1- The Mobile Node sends a registration request to the foreign agent.

2- When the foreign agent receives this request it relays this request to the home agent of the mobile Node.

3- Now it is the turn of Home Agent either to accept or to reject the request sent to it by the foreign agent where it sends a registration reply to the foreign Agent.

4- Final step of registration is done by relaying the reply the foreign agent gets from home agent to the Mobile Node.

Consider the case where the Mobile Node wants to move to a foreign network where there is no foreign agent or all foreign agents are busy. In this case how this Mobile Node will be granted the care-of-address and be registered?

This can be done by an alternative method called co-allocated care-of-address in which the Mobile Node acts as a foreign agent. The Mobile Node makes the registration directly with its home address as it is clear from figure 2.2(b). [5]

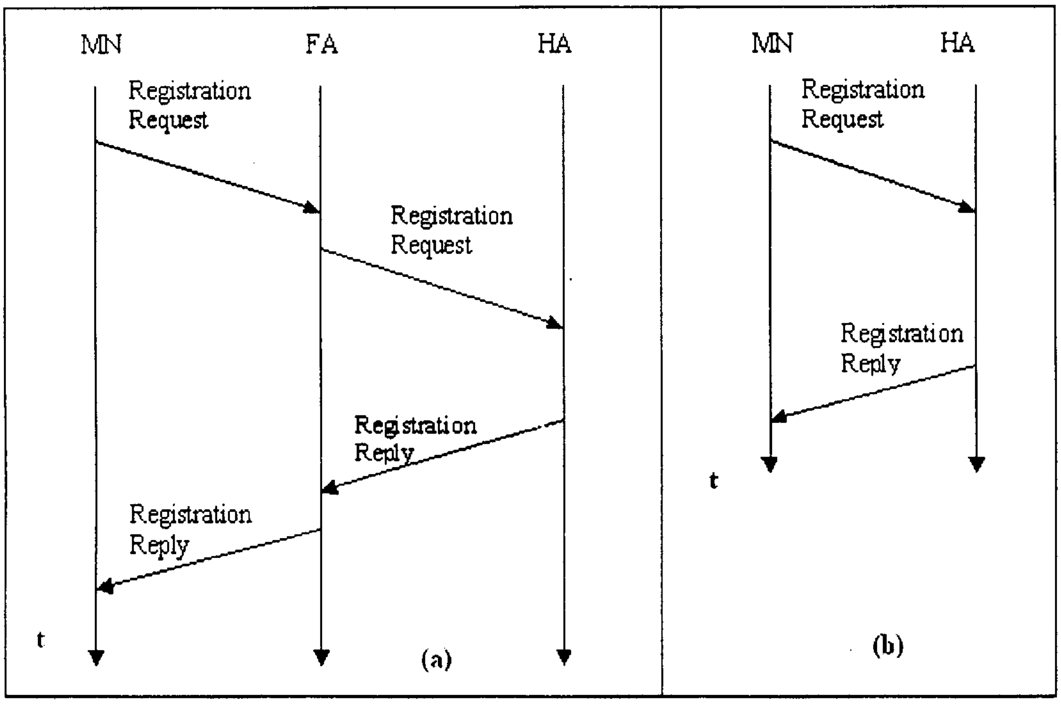
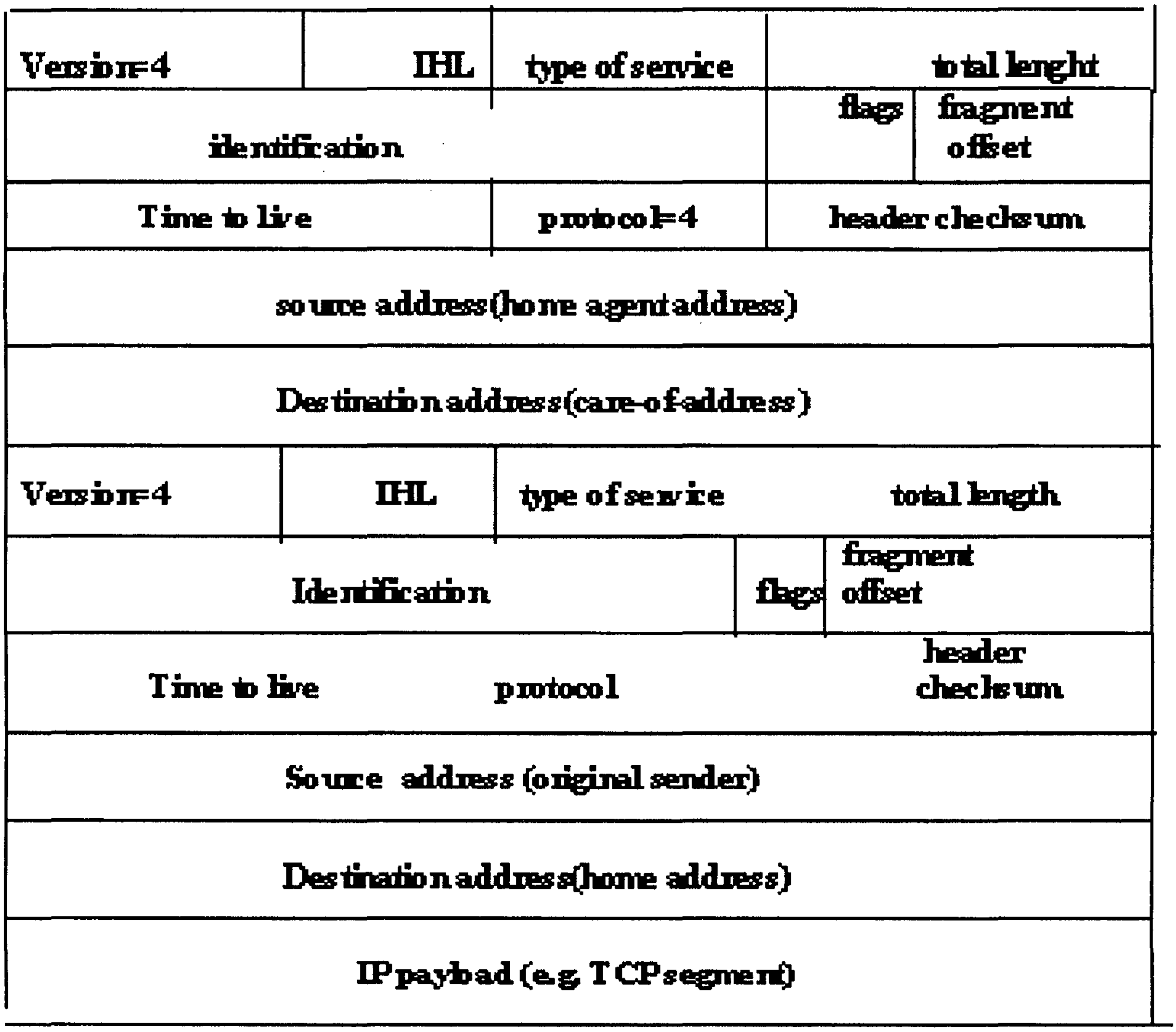


Figure 2.2 Registration procedure in Mobile IP protocol

2.3.3 Tunneling

This operation is done between the home agent and the foreign agent. In this operation the home agent adds a new IP address to the packet sent to the Mobile Host. This packet originally consists of IP header and the data (payload). In tunneling technique the total packet consists of two IP headers besides the payload. This operation is called encapsulation where the original has been encapsulated into another IP header. In the outer IP header the field source address is the same as home agent address, but destination address is equal to the care-of-address that the Mobile Host gets from the foreign network. The encapsulated packet is depicted in figure2.3 [1].



Version=4	IHL	type of service	total lenght	
identification			flags	fragment offset
Time to live		protocol=4	header checksum	
source address(home agent address)				
Destination address(care-of-address)				
Version=4	IHL	type of service	total length	
Identification			flags	fragment offset
Time to live		protocol	header checksum	
Source address (original sender)				
Destination address(home address)				
IP payload (e.g. TCP segment)				

Figure 2.3 Encapsulated packet during tunneling operation

We can observe from the figure 2.3 that the original IP header is preserved and became as a payload of the outer header which is called the tunnel header. [1]

In tunneling operation Home Agent intercepts all packets forwarded to the Mobile Node which is now in a foreign network, but now the home agent has the identity of the Mobile Node, this is for the home agent to be able to capture all packets destined for the Mobile Node that are transmitted across the home network.[1]

2.4 Optimization in Mobile IP protocol

In Mobile IP networks the packets sent between a Corresponding Node (CN) and Home Agent (HA) from one side and between (HA) and Foreign Agent (FA) from the other side creates overhead.

The overhead involved in the network as a result from the non-optimized Mobile IP is called triangular routing. [5]. It is because the packets are routed from CN to HA and from HA to MH which has the care of address in the foreign network. The third segment is from Mobile Host (MH) back to CN. That's why this routing is referred as triangular routing. This is depicted in figure 2.4. [5, 20]

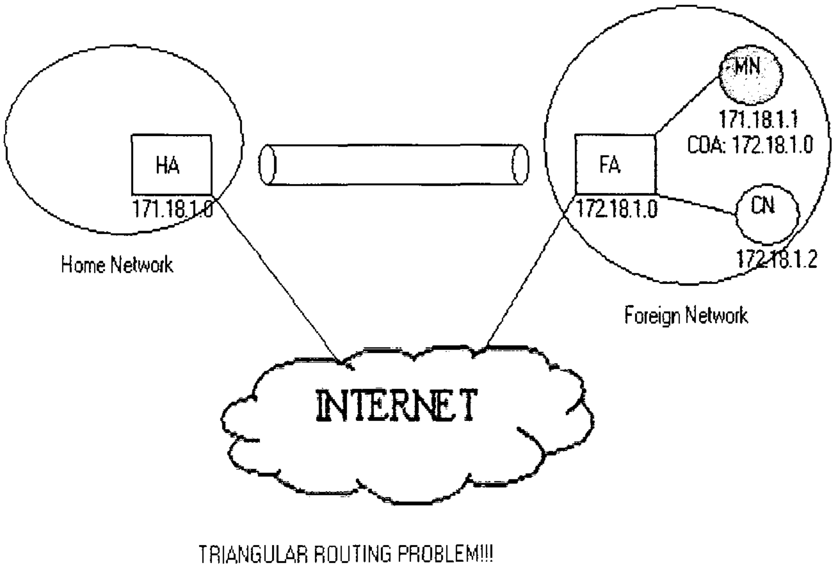


Figure 2.4 Triangular routing in Mobile IP

Optimization of the route in Mobile IP protocol can be done by informing the CN of the current location of MN. But how can the CN be informed of the current location of MN? [5]

CN can be informed of the current location of MN by caching this location in a binding cache and this binding cache is a part of the routing table that exists in the CN. This operation is done through (HA). [5]

The Route Optimization in Mobile IP networks requires the following messages.

- Binding Request: This message is needed and sent by any Mobile Node in the network that wants to know the current location of any other Mobile Node. The procedure is implemented by sending a binding request from (MN) to (HA) which can tell this MN about the location of another MN by sending another message called binding update [5].

- Binding Update: This is the message that is sent by HA to any node in the network asking about a current location of any MH. This message includes the Home Address of the MH and the care-of-address that MH got from the foreign network in which it is residing now. After that the HA may request for acknowledgement from the Node to which the Binding Update message is sent. This Acknowledgement is called Binding Acknowledgement.[5]

- Binding Acknowledgment: It is the signal sent by any Mobile Node having received Binding Update from HA. This message may not be needed always and is sent only when the HA asks for it.[5]

- Binding Warning: It may happen that a MH receives a packet from FA and this FA is not the current FA of this MH. In this case the MH will send Binding Warning telling this FA that this packet is intended to be sent to another Mobile Node. This Binding Warning contains MN's address and the address of the MN from which this packet came.[5]

The above discussion is in case when the MN migrated to a foreign network. The other case is when the MH makes handoff to another foreign network and changes its current FA to register with another FA. In this, there should be some changes like HA must update its location database because the registration is forwarded to it. Also the new FA informs the old FA about the new registration of MN. [5]

An update message is used to pass this information to old FA and in turn FA acknowledges this information. On the other hand CN, willing to send a packet to this MN, will not be informed about these changes. So it does not know any thing about the new location of the MN. As a result it will send the packets that it is willing to send to old FA. Upon receiving these packets, old foreign agent is now not responsible of the MN and it will forward these packets to the new FA under which the MN is now. The new FA sends a binding warning to CN to tell it that its binding cache became old. After that the CN will request a binding update from the HA to inform it about the new location of the MN. HA in turn will send a binding update to the CN informing it about the new location of the MN. CN will send Binding Acknowledgment back to the HA and at this point the CN is able of sending its packets directly to the new FA. [5]

2.5 Cellular IP

Mobile IP is a good protocol for mobility management but this mobility is the macro mobility in which the Mobile users move from one network to another. On the other hand if the Mobile user is willing to move within one network from one cell to another the "Mobile IP" protocol will not be efficient. In this case the best solution for management of this mobility (micro mobility) is cellular IP protocol, a new approach to Internet host mobility. This protocol gives an efficient location management and inherits Cellular principles for mobility management. It takes into consideration the passive connectivity, paging aspect, and fast handoff control. The difference here is that cellular IP protocol implements the mentioned principles from IP point of view (IP paradigm). [1]

Some of the good features of this protocol are minimal need of resources, simplicity in design and minimal use of signaling which is very good in sense of not overloading the network. [1]

2.6 Cellular IP Model

As Cellular IP protocol inherits the features of Cellular networks it is evident that Cellular IP Model will be similar, in basic elements, to the Cellular networks. Thus Cellular IP Model consists of the following components (elements). [21]

- Base station: In Cellular IP networks Base station works as a wireless access point and router of IP packets where if a Mobile host is willing to move from one location (cell) to another it has to attach himself with a new access point (Base station) and it will be responsible of providing all services to this Mobile Host. It will make routing of all packets coming to this Mobile Host until they reach their destination. As a solution, micro mobility Cellular IP protocol can be applied to indoor systems. Here the Base station will be needed in each office or office floor because as we have said that Base station in Cellular IP protocol is the access point needed to accommodate all hosts in the area under its control.[6]

- Gateway: This component is very important in Cellular IP protocol because the gateway is the element through which Mobile Hosts connect to Internet and through the gateway all Mobile hosts are able to communicate with correspondent nodes in another network and can also access Internet. Any Mobile Host moves to a new network it will be given the IP address of the gateway that will be the care-of-address for the Mobile Host.[6]

- Mobile Host: This entity in Cellular IP networks moves from one cell under control of one Base station to another cell where it has to register with the new Base station under which it is willing to move through an operation called handoff.

Figure 2.4 shows the structure of Cellular IP network. [6]

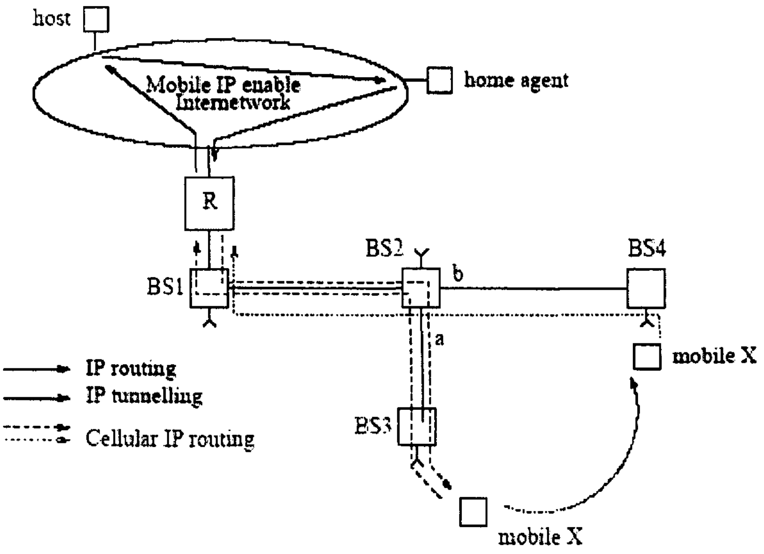


Figure 2.5 Cellular IP Model.

If we have a look inside Cellular IP network we'll see that all Mobile Hosts are identified by their home address and if there are data packets to be sent and routed then no need of tunneling (like in Mobile IP protocol).

One feature of Cellular IP protocol is that it ensures that the packets are delivered to the hosts at any location. On the other hand if there is any Mobile Host which is willing to send some data packets, the way that it will be followed will be from Mobile Host to the gateway then from gateway to Internet till they reach their final destination. [6]

In Cellular IP Model control messaging is tried to be minimized. In order to achieve that, packets sent by Mobile Hosts are themselves used to refresh host location information and in that case there is no need for more control messages to be sent. The Mobile Host which doesn't have any data to be sent from his side he has to send special IP packets towards the gateway. In this case the gateway will maintain the downlink route through which it communicates with the Mobile Host. [6]

Cellular IP network classifies Mobile Hosts into two types: Passive Hosts and Active Hosts.

- Passive Hosts: This can be defined as the hosts that have not received packets for period of time. Those hosts allow their downlink routes to be cleared from the cache, where every Mobile Host maintains a cache.[6]

- Active Hosts: They are the hosts that are in active mode. It means that they send and receive packets from other nodes in the network.[6]

2.7 Operations in Cellular IP networks

The following operations are identified for Cellular IP networks.

2.7.1 Routing

Routing mechanism starts with beacons that gateway sends and flood in the network. Base stations receive these beacons and record the neighbor they last received this beacon from and use it later to make routing of the packets towards the gateway. The routes that are recorded will be used to route the packets coming from all Mobile Hosts to the gateway without taking care of destination addresses of senders. Route information are recorded by each Base station where upon the packet reaches the base station. The last (Base station) will store the IP address of the source Mobile Host and the neighbor from which the packet reached to the node. Finally this packet will reach to the gateway. In this case the route now became known, when a packet comes to the Mobile Host that initiated (originated) the packets. The gate way will send back the coming through the same route. This route remains valid (available) for time called route-timeout. We can define this time as the time during which the route is valid and data packets can be forwarded through this route. We have to notice here that the packets that the Mobile Host sends through this route are used to refresh the mapping (route information stored in every Base station cache). [6]

As the route mapping becomes available for the Mobile Host, this Mobile Host may some times wish to preserve this route mapping. Even it doesn't send data packets regularly through this route, but how can the Mobile Host preserve this route mapping?

This can be done by sending packets called route-update packets on the uplink to the gateway at regular intervals called “called-update time”. Route update packets refresh the routing cache information as the normal packets do.

It is observed that there is a profit by maintaining route mapping in sense of time. The time taken to establish (to get information) this route will be avoided when the Mobile Host maintains its route mapping and it can directly use this route if it is to send data packets though this route again.[6]

2.7.2 Handoff

Handoff operation in Cellular IP network refers to the movement of Mobile Host from one geographical area called cell under the control of one Base station to another cell (Base station). This operation depends on the signal measurements taking place by the Mobile Host and according to the signal strength Mobile Host will decide the cell to which it has to handoff. Cellular IP protocol supports two types of handoff: Hard Handoff and Soft Handoff. [6]

2.7.2.1 Hard Handoff

Mobile Host measures signal strength sent by those stations by listening to beacons transmitted by those Base stations where, as we have mentioned earlier, that each base station sends beacons flood into the network. Now Mobile Host shifts to the new cell and becomes under the control of the new Base station. After that Mobile Host sends route-update packets to the new Base station under its control it is now. This route update packet changes route mapping existing in routing caches in all Base stations and establishes a new path. This is because this Mobile Host wants all Base stations to know its new location where routing in Cellular IP is done on hop-by-hop basis and all data packets sent to this Mobile host should be routed to this new location according to the new route mapping. [6]

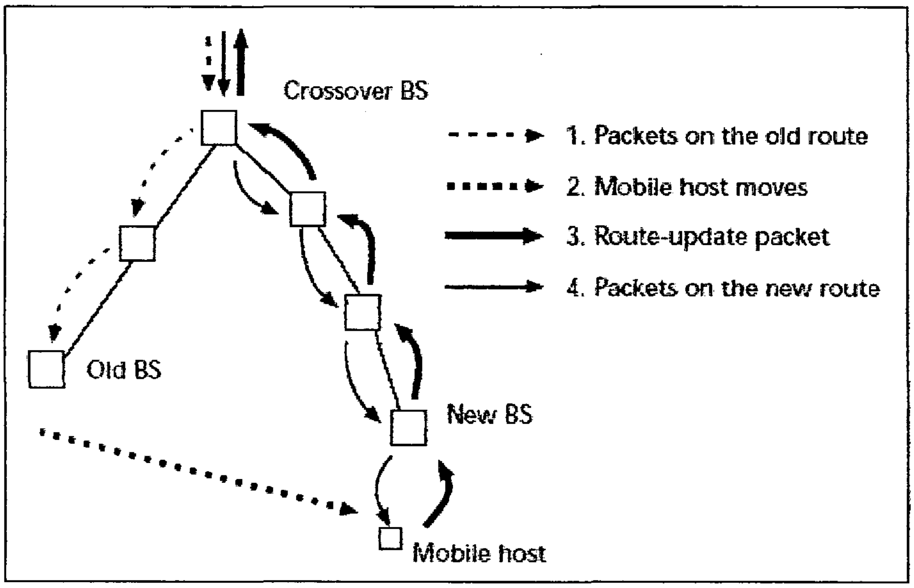


Figure 2.6 Handoff operation in Cellular IP networks

As it is clear from figure 2.6 that during hard handoff operation and before the Mobile Host moves to the other Base station there will be an intermediate Base station called crossover Base station which is a common branch between old and new Base station. [6, 22]

Advantage of hard handoff is its simplicity. Only the Mobile Host needs to measure signal's strength to move to the other cell sending a route-update packet. As a result hard handoff can minimize network traffic during handoff operation. But also there are many disadvantages of hard handoff like the time delay and packet loss during the period in which the Mobile Host switches to the new Base station. A time, called handoff latency, is the time that starts by initiation of hard handoff and the arrival of the first packet along the new route. This time is equal to the round trip time between Mobile Host and the crossover Base station. [6]

Regarding packet loss that can happen during hard handoff, we can say that in comparison with Mobile IP protocol the time taken to redirect packets to the new location of Mobile Host is shorter than that in Mobile IP protocol. [6]

2.7.2.2 Soft Handoff

In soft handoff scenario in cellular IP networks Mobile Host sends a soft packet which is a request to the new Base station, and then returns back at the same time to the old Base station to continue the session that was taking place between it and the old Base station. Using this semi-soft packet, routing cache mapping will be changed in order to establish a new route related to this Mobile Host. At the time of new route establishment, the Mobile Host will be connected to the old Base station and continue the old session with the old Base station. This procedure continues for a semi-soft delay. After this time delay the Mobile Host will be moved to the new cell and begins a new session with the new Base station. [6, 22]

Advantage of soft handoff is that it is suitable for large numbers of Mobile Hosts (large Cellular IP networks) that make frequent handoffs between cells. In soft handoff, time delay is less than time delay in hard handoff because creation (establishment) of the new route is done by semi-soft packet that the Mobile Host sends before it hands-off to the new cell.[6]

2.7.3 Paging

In Cellular IP networks, Mobile Hosts need to update their location information in case they are in active mode. Because of that they will be connected to the internet all the time, the thing which makes them reachable all the time and consume bandwidth and battery power in a big amount. This is different in wired networks where hosts are connected all the time even they don’t consume that much of bandwidth. [6]

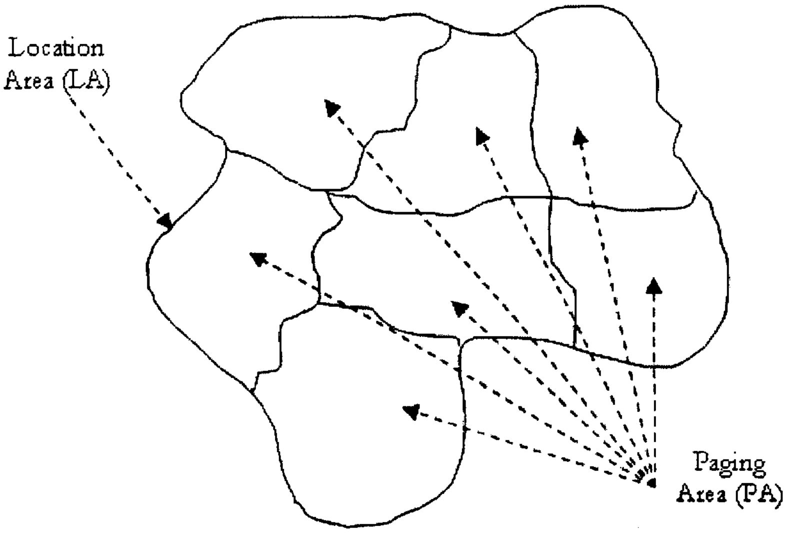


Figure 2.7 paging areas and active areas in Cellular IP

In Cellular IP system, if the Mobile Host is not sending or receiving any packets, it will be idle and is in a passive connectivity. This passive connectivity state is defined in Cellular IP networks for a period of time called active state-time-out. [8]

Base stations in Cellular IP networks are grouped into paging areas as it is clear from figure 2.7. In the case in which there is a Mobile Host reachable in the network, it has to send paging-update packets at regular time intervals called paging update-time. These packets should be sent to the gateway while routing in Cellular IP network is done in hop-by-hop. Then page mapping should be kept in Base stations at paging caches and this mapping is changed by paging-update packets sent by Mobile Hosts. [6]

Suppose there was a packet addressed to an idle Mobile Host, it will reach to the gateway coming from its source; the gateway then will forward this packet to a base station in order to make hop-by-hop routing. The base station will see that there is no valid routing cache mapping for this mobile host, so the base station will check for the validity of paging cache mapping. If the Mobile host has a valid paging cache mapping, the base station forwards this packet to the Mobile Host which will be at that time in the active mode. [6]

2.8 IP Header Compression

The Header Compression and its possibility in Cellular IP is elaborated in this section.

2.8.1 The need

In any network there is a competition for the availability of bandwidth to customers in this network. It is clear from figure 2.8

The chain is as strong as its weakest link

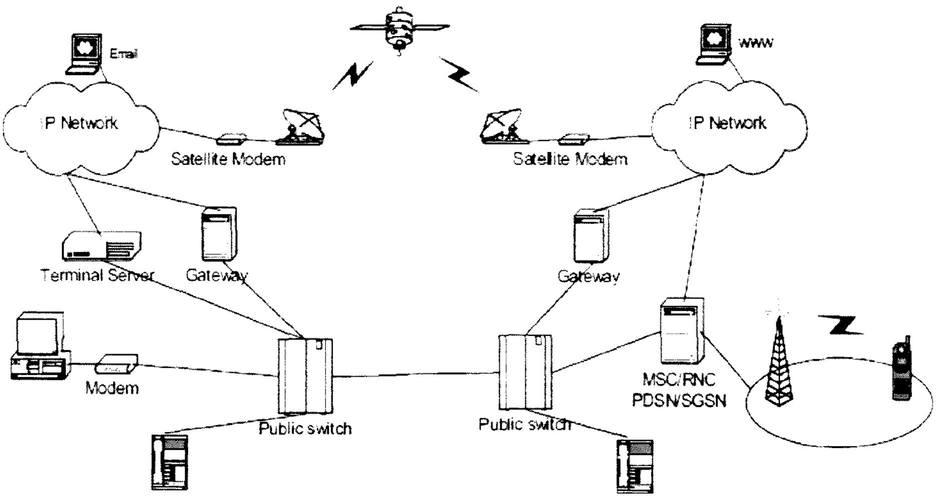


Figure 2.8 Networks developing so there is need for bandwidth.

As a result, the network operators should offer a good quality of service including bandwidth in order to make many customers subscribe their networks. [4]

For wireless networks there is a high probability of error, high latency, and high bit error rate. Under these circumstances there is the need to use the available resources in the network in an efficient way for this network to be able to provide the services for all users in a good manner. [4]

As mentioned previously, with the problems of wireless networks, IP packets are big in size because of the headers added from every layer to the basic payload. That's why these packets will consume a big amount of bandwidth. The only solution and efficient solution for this problem is to compress these packets in order to make saving of 90% in bandwidth. According to the studies this is one benefit of Header Compression. There are many other benefits like: reduction in packet loss and improvement in the response time of the network. [4]

So we can define Header Compression as the process of decreasing the size of IP packets before transmitting them in the link and then decompressing them when receiving them again. This process is done by avoiding sending redundant fields of the IP header as we will see later. [4]

2.8.2 Header Compression and link efficiency

Efficiency of Header Compression is observed by studying IPv4 is total 40 bytes in size of which:
IPv4 size=20 bytes.
User Datagram Protocol (UDP) size=8 bytes.
Real Time Transport Protocol (RTP) size =12 bytes.
It has been observed that by using Header Compression we can decrease the size from 40 bytes to 2-4 bytes as evident from figure 2.9

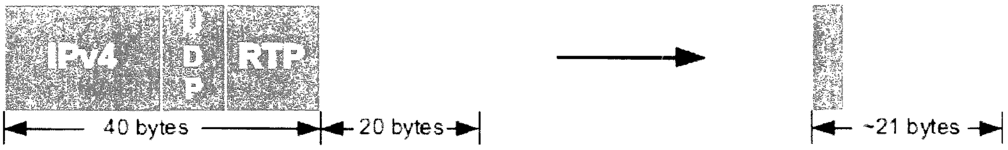


Figure 2.9 Header Compression technique

This decrement in packet size is good and results in major bandwidth saving. [4]

Header Compression can improve the efficiency of transmission link by improving response time due to smaller packet size that we get from Header Compression and also reduction in the probability of packet loss. Efficiency improvement of link can be observed by using Header Compression where Header Compression will decrease packet header overhead (bandwidth saving), reduces packet loss and improve response time of the network. Through the benefits we can say that Header Compression results in efficient link but we still have to explain these benefits in detail. [18]

Improve interactive response time: In low-speed links if we want to send data this may take a long time like (100-200) ms which is not good for users who require high transmission speed. On the other hand by using Header Compression the time required to transmit these data is far less than before because here (by using Header Compression) we send only some fields and avoid sending the redundant fields and that results is a less transmission time. [9]

Sending bulk data using small packets: Where we have bulk traffic (e.g. FTP) and interactive traffic (e.g. Telnet), these are mixed together. Then sending bulk in small packets (using Header Compression) will decrease the waiting time which is an important parameter for interactive data .[9]

Decrease Header overhead: Overhead in general is an extra amount added to the basic amount due to some operation taking place. For example if we took IPv6 for mobile IP Protocol which has a size of 512 octets, when a tunneling operation (discussed earlier) takes place there will be 100 octets added as overhead to the main packet. It means that there will be 19.53% overhead is added. By using Header Compression this amount can be reduced from 19.53% to 1% at least. [9]

Reduce packet loss: It is obvious that by using Header Compression we are going to send fewer amounts of bits. It will reduce the packet loss and in turn increase the throughput for TCP protocol. [9]

Now let's have a close look at the structure of IPv4 header in figure 2.10 [11].

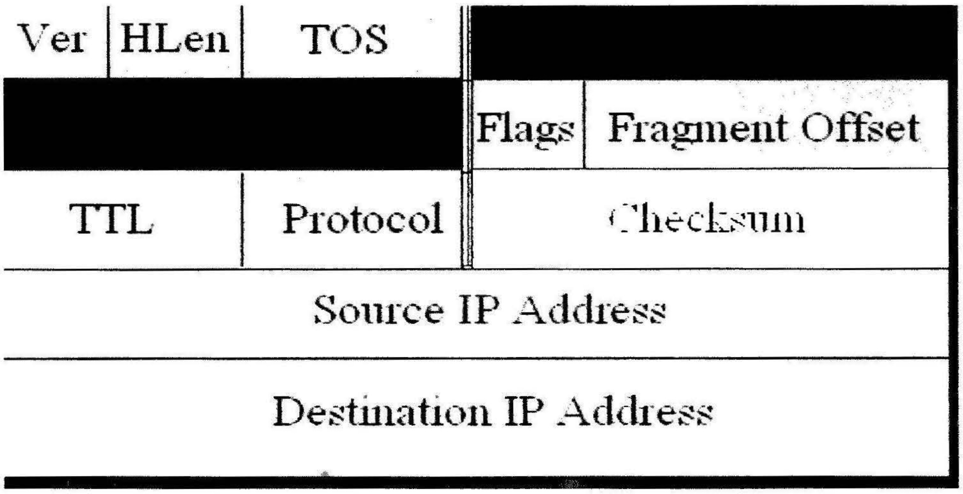


Figure 2.10 IPv4 Header format

Various fields are as follows.

Ver: This field includes the Version of IP header used.

HLen: Indicates IP Header Length in bytes.

TOS: Type of Service in which there is the level of importance assigned to the datagram.

Length: Includes the total length.

ID (Identification): which contains an integer that identifies the current datagram. in case there were segments it will help in grouping them.

Flags: It is three bit field used for the purpose of fragmentation.

Fragment Offset: In case of fragmentation this field is used to indicate the position of the fragment's data relative to the beginning of the data in the original datagram.

TTL (Time To Live): maintains a counter which decrements down to zero at the point the datagram discarded. This field is useful because it does not allow packets to loop endlessly.

Protocol: This field indicates which upper-layer protocol receives incoming.

Checksum: Used for error detecting.

Source IP Address: Indicates the address of the receiver.

Destination IP Address: Indicates the final address of the sent packet.

We can classify these fields into four categories [23]

- Static fields (STATIC): these fields don't change during transmitting or receiving operation.
- Delta fields: these fields can be changed by small values (delta values). These delta values can be sent instead of the whole fields because sending of the whole fields consume more bandwidth and more time, so we can avoid that by sending these delta values.
- Random fields: these fields can be changed randomly so when using Header Compression we need to send them.
- Inferred values: when we receive the compressed packet, then values of these fields can be inferred from the other values.

2.9 Compression/Decompression process

The Header Compression/Decompression process is explained here in this section.

2.9.1 Compression Process

In compression process, context is established on both sides Compressor and Decompressor.

Figure 2.11 shows the Model used for Compression/Decompression in Van Jacobson compression algorithm. [9].

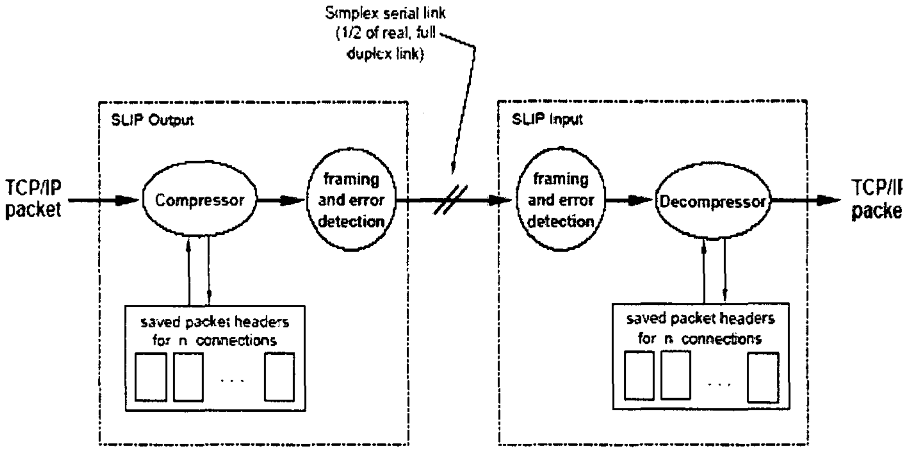

Simplex serial link
(1/2 of real, full duplex link)
SLIP Output
TCP/IP packet
Compressor
framing and error detection
saved packet headers for n connections
SLIP Input
framing and error detection
Decompressor
saved packet headers for n connections
TCP/II packe

Figure 2.11 Compression/Decompression Model

First of all we will explain the Compression side of this Model and later decompression side.

Compression side

The Compressor uses SLIP (Serial Line Internet Protocol) which is useful for allowing hosts and routers to communicate with one another. Using it, host-host, host-router and router-router communication are possible. Compressor also uses a Framer which is the last point in the Compressor side and first point in Decompressor side. Framer is responsible for communicating packet data between Compressor and Decompressor also putting the type and boundary of the packets. This operation is useful for the Decompressor to know how many bytes came out of the Compressor. After highlighting the structure of the Compressor used, the mechanism used for Compression in the Compressor is as follows.

When the packets come to the Compressor it starts to check the type of these packets. If the coming packet is either Non-TCP Protocol packet or Uncompressible TCP packet, the Compressor will put this packet into the category TYPE_IP packet and passes it directly to the framer. If the coming packet is of the type Compressible TCP packet, the Compressor at that time will be having an array in which it stores packet headers. When this TCP packet comes, the Compressor will look up this packet into the stored array, if matching is found then the packet is compressed and passed to the framer [10]. If no matching is found, then it will be a new entry into the array to be compressed with and directly a packet of the type UNCOMPRESSED_TCP is sent to the framer.

2.9.2 Decompression Process

Upon reaching the compressed packet to the Decompressor, if that packet was corrupted (error has been detected in it) then this packet obviously can not be decompressed and has to be dropped making the decompressor out of synchronization. [9] [29]

This corrupted packet which the decompressor couldn't decompress, when it arrives to the decompressor it will assume that the reason behind not decompressing it correctly is that one or more previous packets are lost. Because of that the following packets received by the decompressor may also be dropped even though they may be transmitted correctly. This is called error propagation. [9] [29]

The decompressor recognizes the types of incoming packets and does a "switch" on the type of incoming packets. Non-TCP packets are simply passed through the decompressor. For UNCOMPRESSED_TCP packets the decompressor will get the connection number and use it as index into the receiver's array of saved TCP/IP headers. For the third type which is compressed TCP packets, after the decompressor gets the connection number it uses it as an array index to get the TCP/IP header of the last packet from that connection.[9] [29]

Also the establishment of context is done on the decompressor side and to increase the possibility that the decompressor correctly establishes the context n full header packets are initially sent. Establishment of context is discussed in more detail in the following section. [29].

2.10 Context Transfer

Context, an important parameter, to establish the connection, is transferred. The process is discussed here.

2.10.1 Context Definition

Whenever a Mobile Host exists in a network it must be provided some services like Authentication, Authorization, Accounting (AAA), QoS, etc.
These services form the important thing which is called Context.

These information are needed to re-establish these services on a new subnet if this Mobile Host moves to another subnet. We can take an example of Context in Cellular IP protocol when we use a Header Compression technique which is an important service to be provided to the Mobile Host. The Context of this Header Compression is established and later on transferred when this Host handoffs to a new subnet. The Context of Header

Compression usually consists of STATIC fields and DELTA fields which are needed to reconstruct the compressed header on the Decompressor side. [7].

2.10.2 Context Transfer Definition

If the Mobile Host moves to another sub network, it needs to re-establish those mentioned services in the new sub-network. But reestablishing these services again and again seems to be costly in sense of time and in sense of bandwidth. We can say that it is at any way not good for network resources utilization.

So the substitute solution that can be suggested is Context Transfer which means the movement of the established Context in one sub-network to the new sub-network to which the Mobile Host is willing to move. It is the movement of the established Context from one router or any other network entity to another as a means of re-establishing specific services. [7].

2.10.3 Need for Context Transfer

The most important need for Context Transfer observed is the quick re-establishment of services that exist in the previous subnet without wasting a lot of time again.

An important example, in this regard, is transferring Header Compression Context which has been established during Compression/Decompression process. Transferring the Context should be done when the Mobile Host is no longer receiving packets from the old base station and connected to the new base station before receiving any packet through this new base station. [7]

2.10.4 Context Transfer Process

Establishment of context is done during the operations of Compression /Decompression. The compressor which has the packet examines the packet header and classifies the fields of header that will be compressed. Because some of these fields will form the context later on, after the establishment of the context, this context should be unique and this can be done by giving it a number called Context Identifier number

(CID). This is the process of Context establishment on the Compressor side. After that the compressor transmits the full packet (uncompressed) to the decompressor which will start building the Context on its side. Usually the established Context consists of STATIC and DELTA fields described previously of the header. We need to notice here that while the Context contains STATIC fields, that mean that the compression can be done by not including STATIC fields in the packet. The compressor usually sends n full packets in order to increase the probability that the decompressor establishes the Context correctly. After establishment of the context the decompressor sends an Acknowledgement to confirm the establishment of the Context. [11]

2.10.5 Three state Header Compression Context

This scheme is suggested to further enhance the compression reliability where it has three phases of compression and can be discussed as follows.

From figure 2.12 it is clear that in partly compressed packet stage the packet header will consist of CID, Random fields, Delta fields and possibly Inferred fields. Therefore here the context will consist of STATIC fields.

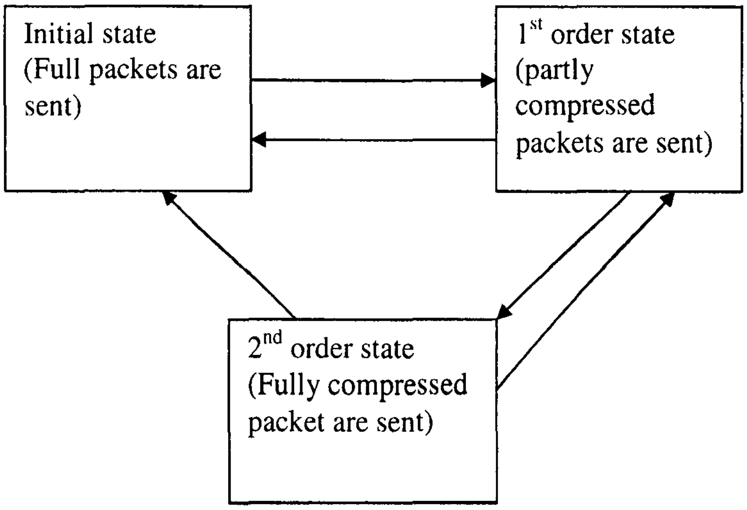

Initial state
(Full packets are sent)
1st order state
(partly compressed packets are sent)
2nd order state
(Fully compressed packet are sent)

Figure 2.12 Three states Header Compression Context

In the other state (fully compressed packets) the packet header consists of CID, Random fields and Delta values so that the context consists of STATIC and Delta fields. The importance of this scheme comes from the fact that it limits the impact of lost context of the first state. [11]

Chapter 3

The Proposed Model

This chapter contains the model proposed for the context transfer with Header Compression in Cellular IP

3.1 Analysis of Cellular IP Model

During this analysis, the transition diagram has been explained that describes the whole operations that take place in cellular IP network as well as the context transfer operation.

In this transition diagram we used four bits to recognize all possible inputs. Four bits are sufficient enough to represent all possible states. In what follows we will show all possible inputs and the operations of every input. Table 3.1 briefs the operation corresponding to each string.

Input string	Operation that it represents
0000	Information about the network sent to the MH from the BSh
0001	Signal sent from BSh to MH and it is strong
0010	Signal sent from BSf to MH and it is weak
0011	Registration request from MH to BSh
0100	Registration reply from BSh to MH
0101	Data sent from MH to CN
0110	Data received from CN by BSh
0111	Full packets are sent from BSh to MH
1000	Partially compressed packets are sent to MH
1001	Fully compressed packets are sent to MH
1010	MH sends rout-update packet to the BSf
1011	Registration request to the BSf appended with it the context which was established before
1100	Registration reply from BSf
1101	ACK sent from BSf that it got the context.

Table 3.1

In this table:

BSh refers to the Base station in which the Mobile Host is now.

BSf refers to the Base station to which the Mobile Host is willing to move (handoff).

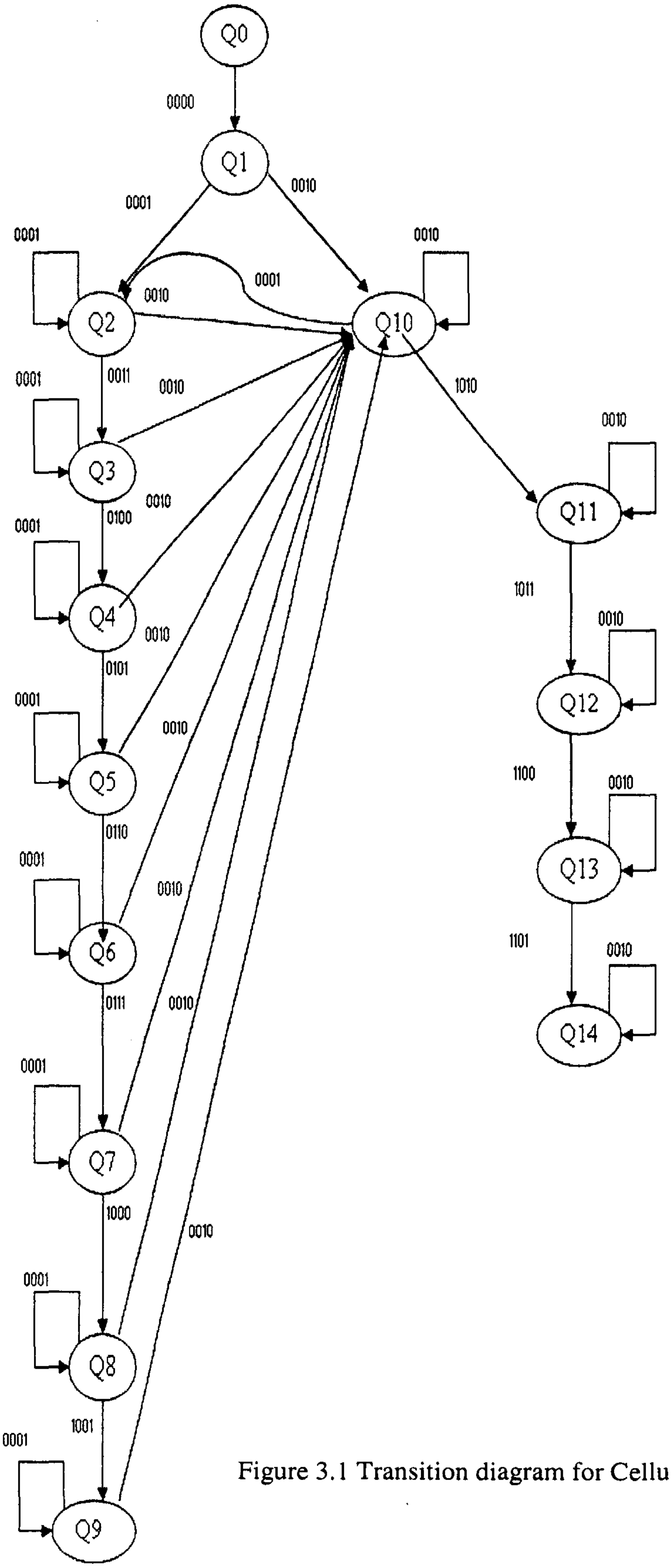


Figure 3.1 Transition diagram for Cellular IP

List of states

Various states, used in the Transition diagram of Cellular IP(figure 3.1), is explained as below.

Q1

The initial state which tells us that the Mobile Host has just switched on and received information from the base station which controls the cell in which it is now.

Q2

This state tells us the strength of the signal coming from the base station is controlling the cell in which the Mobile Host is now stronger than other signal strengths. This base station is called as BSh.

Q3

It means that the Mobile Host has sent a registration request to BSh because it measured the strongest signal from it.

Q4

BSh replies the registration request sent from the Mobile Host.

Q5

In this state the Mobile Host sends data to another mobile node in the network. This mobile node is called as CN (Corresponding Node).

Q6

This state is reached when the CN sends data to the Mobile Host. Here we need to consider that BSh will receive these data before they arrive to MH where data compression procedure will start in the next state.

Q7

It is the first state of the three state HC context in which BSh sends n full packet to the MH.

Q8

Second state of the three state HC context and in this state BSh sends partially compressed packets to MH.

Q9

Third state of the three state HC context. Here BSh sends full compressed packets to MH.

Q10

The strength of signal coming from another base station is stronger than the strength of the signal coming from BSh. We call the other base station BSf.

Q11

Here MH is willing to make handoff to the other cell with the stronger signal so first it will send rout-update packet to BSf before it performs the handoff.

Q12

Mobile Host sends registration request to BSf and appends with it the context which had been established in the three state HC context.

Q13

BSf sends reply to the MH for the previous request.

Q14

BSf sends ACK to MH that it got the context and now ready to make session with it without re-establishing the context again.

3.2 Explanation of the state diagram

First of all, when the Mobile Host (MH) is switched on it receives cell information from the base station that controls the cell in which it is now (BSh). At that time MH will measure the strength of the signals coming to it from all base stations in the network. If the strength of the signal coming from BSh is strong, it will send a registration request to this base station in order to register itself in this cell until it moves to another cell. After that BSh will reply to MH's request and allow it to register itself with it. Now the Mobile Host (MH) is able to make session with the base station BSh, and MH is able to send data to a mobile node in another cell of the same network. We call the other mobile node CN (Corresponding Node). Data will also be sent from the CN to the Mobile Host (MH). When data will arrive first to the BSh the later will start header compression procedure during which the establishment of the context will also be done. The establishment of the context will be done through three state header compression context.

This scheme is suggested to enhance compression reliability. In first order state of this scheme BSh sends 'n' full packets to the Mobile Host (MH) in order to increase the probability that the decompressor (MH) correctly establishes the HC context. Then in second order state BSh sends partially compressed packets to MH. Final state of the HC context starts when BSh sends fully compressed packets to MH. Now from this state (Q9) if the MH detects that the strength of the signal coming from BSf is stronger than the strength of the signal coming form BSh it will send request to the BSf in order to make handoff to the cell under control of BSf. This request is called soft handoff request. With the soft packet the MH will append the context as had been established before. BSf in turn will reply to the request and get the context. In this way the context is transferred to the other base station and this base station will send an ACK to MH to inform it that base station got the context it sent and ready to initiate the session with the MH.

Chapter 4

Simulation Experiment

We carried out the experiment to find the probability of error during Header Compression (explained in Sec. 4.1). We also have noted the variation in time with context transfer and without context transfer with varying packet sizes. It is explained in Sec. 4.2. Finally we carried the experiment to study effect of Header Compression on Bandwidth using NS-2 simulator. In Sec. 4.3 the brief discussion on NS-2 follows the experiment and results. Concluding remarks are suffixed at the end of each experiment.

4.1 Probability of error during Header Compression

Despite the fact that we are studying context transfer, we have to take care of the error that can be found because context is a result form header compression procedure.

We considered that the IP header, we deal with, is IPv4. The size of this header is 40 bytes before we apply header compression technique on this header, but the size that we can get after header compression is 4 bytes.

So the probability that no errors is in a packet is

$$p_n = (1 - \alpha) \text{ where } \alpha \text{ is (Bit Error Rate) BER}$$

Probability that there is one byte in the packet in error is

$$p_b = 1 - (1 - \alpha)^8$$

Now we have applied these two probabilities for compressed packet and uncompressed packet. The following results are obtained (table 4.1)

Bit Error Rate (BER)	Probability of error in case of compressed header	Probability of error in case of uncompressed header
0.001	0.0315	0.27397
0.002	0.06205	0.47304
0.01	0.27519	0.95988

Table 4.1

The same is explained by a bar diagram (figure 4.1)

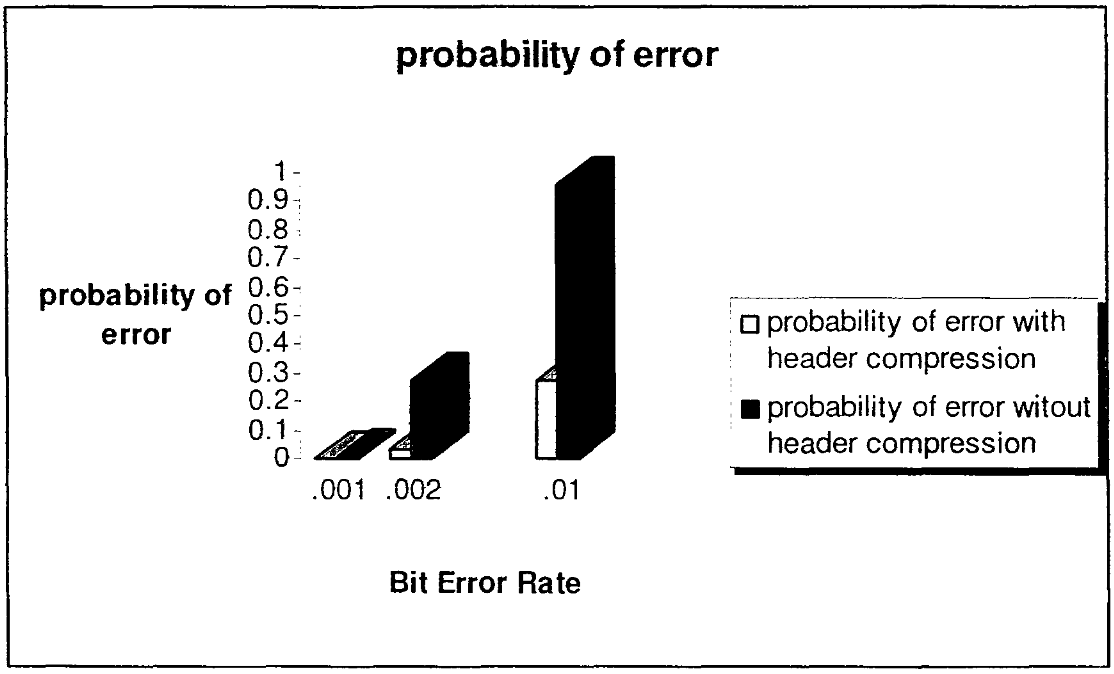


Figure 4.1 Probability of error for various BER

Observations

1- Probability of error is increased in case of a big error rate.

2- Probability of error is bigger in case of uncompressed headers, which is one benefit of the Header Compression. As a result bit error rate will be decreased in case of context establishment and transfer because it is only a part of header compression.

4.2 Time Compression with context transfer and without context transfer with varying packet size

From state diagram we can see that context establishment takes three states Q7, Q8, Q9 to finish this operation, while transferring the established context consumes only two states Q10, Q11.

So the time taken to transfer the context is much less than the time taken to re-establish the context again and start a session again. This time changes every time due to the nature of the packet every time. It means that since the data sent by the packet changes every time so the size of the packet changes every time and as a result the time taken in each case (Context Establishment and Context Transfer) will be different. Table 4.2 shows different packets sizes and the time taken followed by figure 4.2.

Size of packet	Context transfer time	Context re-establishment time	Average overhead
500	6.2188	20.0436	4.072
1000	1.5031	11.0094	4.036
1500	1.068	6.2708	4.024
2000	0.8004	4.7027	4.018
2500	0.6002	4.4017	4.014
3000	1.0335	3.1343	4.012
3500	0.4572	2.6580	4.010
4000	0.3751	2.3506	4.009
4500	0.7112	2.0671	4.008
5000	0.3201	2.1803	4.007
5500	0.2728	1.7094	4.006
6000	0.2667	1.8169	4.005
6500	0.2462	1.4309	4.004
7000	0.4429	1.3430	4.004
7500	0.2001	1.4668	4.004
8000	0.4000	1.1626	4.004
8500	0.1765	1.2942	4.003
9000	0.3263	0.9895	4.003
9500	0.1600	0.9401	4.003

Table 4.2

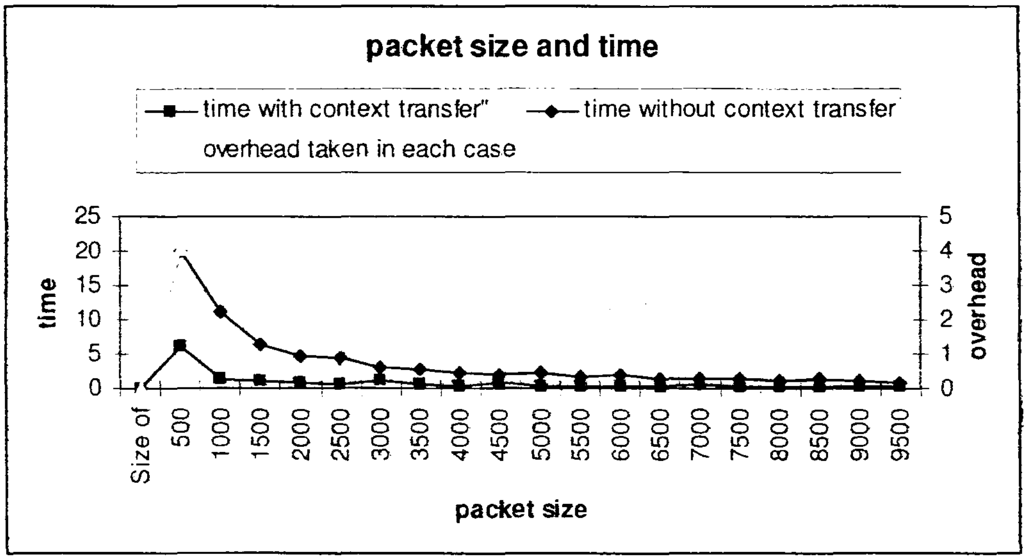


Figure 4.2 Time taken for different packet sizes

Observations

1- The time taken for the context transfer is less than the time taken for context re-establishment for all sizes of sent packets even that's proved for bigger sizes of packets.

2- When the time taken to send a number of packets reached to a specific value the value of overhead will become constant.

4.3 Simulation using NS-2 Simulator

NS or the Network Simulator (popularly called NS-2 in reference to its current generation). It is a popular simulator for its extensibility (due to its open source model). NS is mostly used in the simulation of routing and multicast protocols, among others, and is heavily used in AD-HOC research. NS supports an array of popular network protocols, offering simulation results for wired and wireless networks alike. [13] [28]

4.3.1 Design of NS-2 Simulator

NS-2 was built in C++ and provides a simulation interface through OTCL, an object-oriented dialect of TCL. The user describes a network topology by writing OTCL scripts, and then the main NS program simulates that topology with specified parameters. [12]

4.3.2 What is TCL language?

TCL stands for Tool Command Language. It is designed to be glue that assembles software building blocks into applications. In addition, TCL is interpreted when the application runs. The interpreter makes it easy to build and refine the application in an interactive manner. [15]

4.3.3 Nodes and packet forwarding

Node Basics

The basic primitive for creating a node is

```
$ ns [new Simulator]
$ ns node
```

The instance procedure node constructs a node out of simpler classifier objects. The node itself is a standalone class in TCL. However, most of the components of the node are themselves TCL objects. [14]

The typical structure of a (uni-cast) node is as shown in figure 4.3.

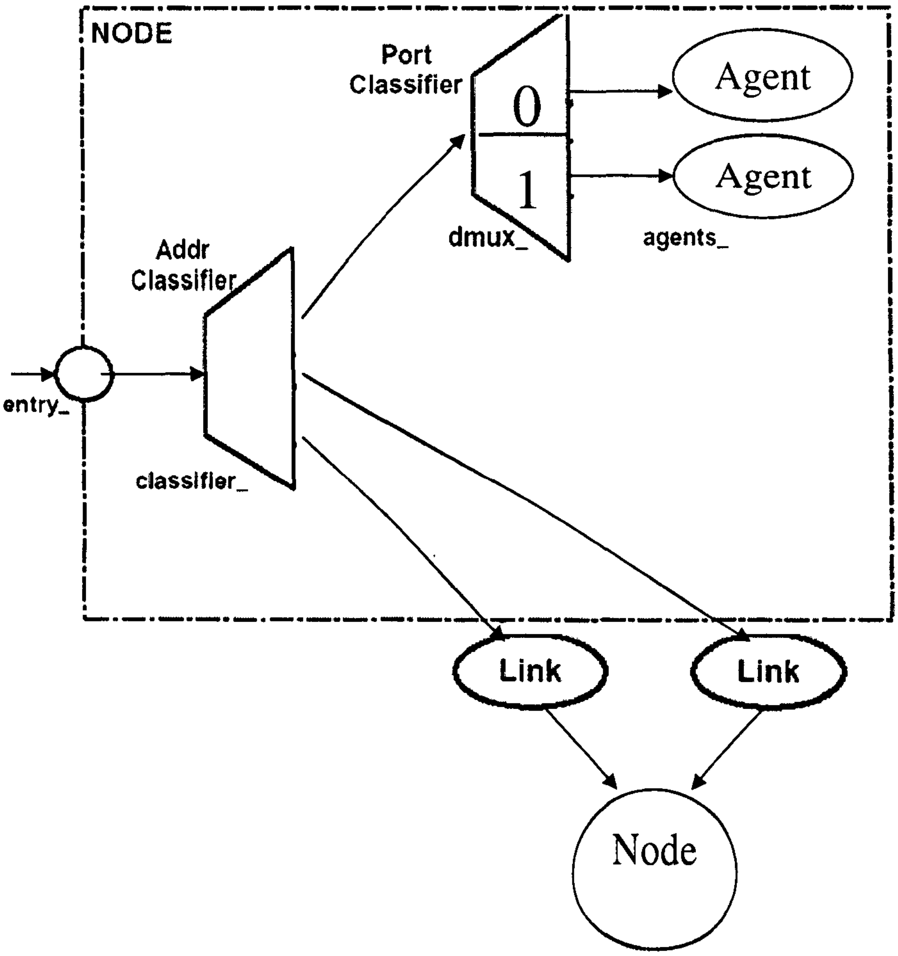


Figure 4.3 Structure of uni-cast node used in NS-2 Simulator

This simple node consists of two TCL objects

Address classifier and port classifier. The function of this classifier is to distribute incoming packets to the agent or outgoing link.

All nodes contain at least the following components

- An address or ID_
- A list of neighbors
- A list of agents
- A list of agents
- A node type identifier
- A routing module

In order to enable multicast simulation, the simulation should be created with an option "multicast on" e.g.
$Set ns [new Simulator-multicast on] [14]

Node Methods: configuring the Node

Procedures to configure an individual node can be classified into [14]

- Control functions
- Address and port number management, uni-cast routing functions.
- Agent management
- Adding neighbors

4.3.4 The classifier

The function of a node when it receives a packet is to examine the packets fields, usually its destination address, and on occasion, its source address. It should then map to the values to an outgoing interface object that is the next downstream recipient of this packet.

A classifier provides a way to match a packet against some logical criteria and retrieve a reference to another simulation object based on the match results. Each classifier contains a table of simulation objects indexed by slot number. The job of a classifier is to determine the slot number associated with a received packet and forward that packet to the object referenced by that particular slot. [14, 25]

4.3.5 Mobile Networking in NS

The basic wireless model in NS essentially consists of the Mobile Node at the core with additional supporting features that allows simulations of multi-hop Ad-hoc networks, wireless LAN etc. A Mobile Node is the basic node object with added functionalities of a wireless and mobile node like ability to move within a given topology, ability to receive and transmit signals to and from a wireless channel etc. A major difference between them, though, is that a mobile node is not connected by means of links to other nodes or mobile nodes. [14]

4.3.5.1 Creating wireless topology

Mobile Node is the basic NS node object with added functionalities like movement, ability to transmit and receive on a channel that allows it to be used to create mobile, wireless simulation environments. The class Mobile Node is derived from the base class node. The mobility features including node movement, periodic position updates, maintaining topology boundary etc, are implemented in C++ while plumbing of network components with mobile node itself have been implemented in OTCL.

Creation of a mobile node can be performed as in this code:

```
$ ns_node_config        -adhoc Routing $opt (adhoc Routing)
                        -ll Type $opt (LL)
                        -mac Type $opt (mac)
                        -if qType $opt (ifq)
                        -if qLen $opt (if qLen)
                        -ant Type $opt (ant)
                        -propInstance [new $opt (prop)]
                        -phyType $opt (netif)
                        -channel [new $opt (chan)]
                        -topInstance $topo
                        -wire Routing OFF
                        -agent Trace ON
                        -router Trace OFF
                        -mac Trace OFF
```

Next actually create the mobile nodes as follows

```
For {set j 0} {$ j < $ opt (nn)} {incr j} {
 Set   node_ ($j) [$ ns_node]
$ node_ ($i) random-motion 0; # disable random motion}
```

The above procedure creates a Mobile Node, creates an adhoc- routing agents a specified , creates the network stack consisting of a link layer, interface queue, mac layer, and a network interface with an antenna, uses the defined propagation model, interconnects these components and connects the stack to the channel. [14]

4.3.5.2 Creating Node Movements

The Mobile Node is designed to move in a three dimensional topology. However the third dimension (Z) is not used, that is the Mobile Node is assumed to move always on a flat terrain with Z always equal to 0, thus the Mobile Node has X, Y, (Z=0) co-ordinates that is continually adjusted as the node moves. There are two mechanisms to induce movement in Mobile Node. In the first method, starting position of the node and its future destinations may be set explicitly these directives are normally included in a separate movement scenario file.

The start –position and future destinations for a Mobile Node may be set using the following APIs:

```
$ node  Set X_  <X1>
$ node  Set Y_  <Y1>
$ node  Set Z_  <Z1>
$ ns at  $ time $ node set dest <X2><Y2>
<speed>
```

At ($ time) second, the node would start moving from its initial position of (X1,Y1) towards a destination (X2,Y2) at the defined speed. In this method the node-movement-updates are triggered whenever the position of the node at a given time is required to be known. This may be triggered by a query from a neighboring node seeking to know the distance between them or the set dest directive described above that changes the direction and speed of the node. [14]

4.3.6 Creating output files for X-graph

One part of the NS-2 simulator is 'X-graph' a plotting program which can be used to create graphic representations of simulation results. We can create output files in our

TCL scripts which can be used as data sets for X-graph but for that we have to use traffic generators to be able to make such output files. [14]

4.3.7 Using NS-2 Simulator for Cellular IP Model

The Model we are going to simulate is described in figure 4.4

In this model (topology) three base stations are connected together, each one control a cell and we have two Mobile Nodes move from one cell to another. Cell limits are not depicted in the figure due to simulator abilities.

The simulation scenario is as follows: the Mobile Node (0)

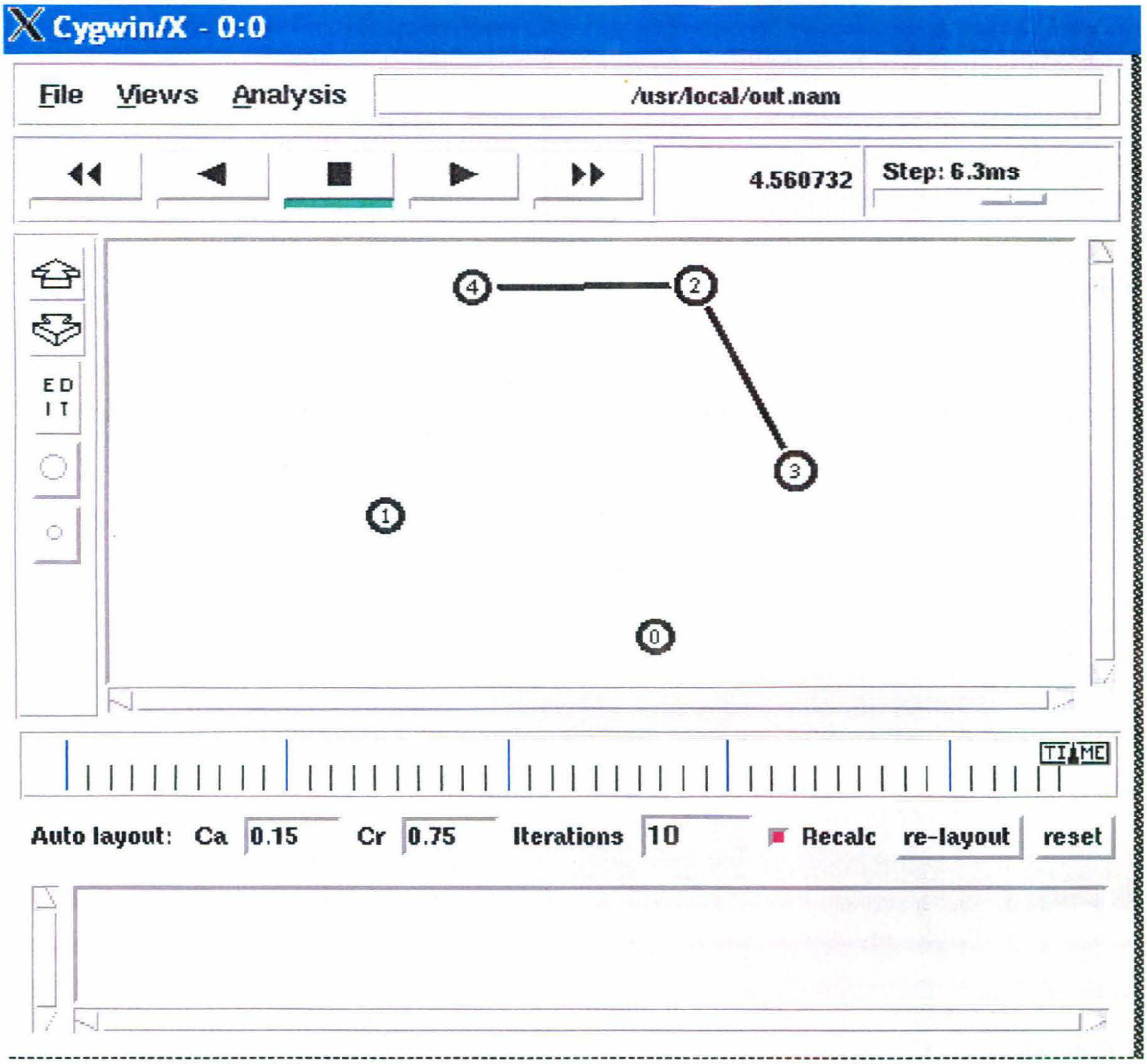


Figure 4.4 Cellular IP Model

As we can see, this model consists of three Base Stations (4), (2), (3) connected together as well as two Mobile Hosts (0), (1).

We can show that Mobile Host (1) sends data packets to Mobile Host (0) where he sends these packets through the Base Station (4) then from (4) to Base Station (2) and (3) till the packets reach the Mobile Host (0).

Where figure 4.5 shows how node (1) is sending data packets to node (0)

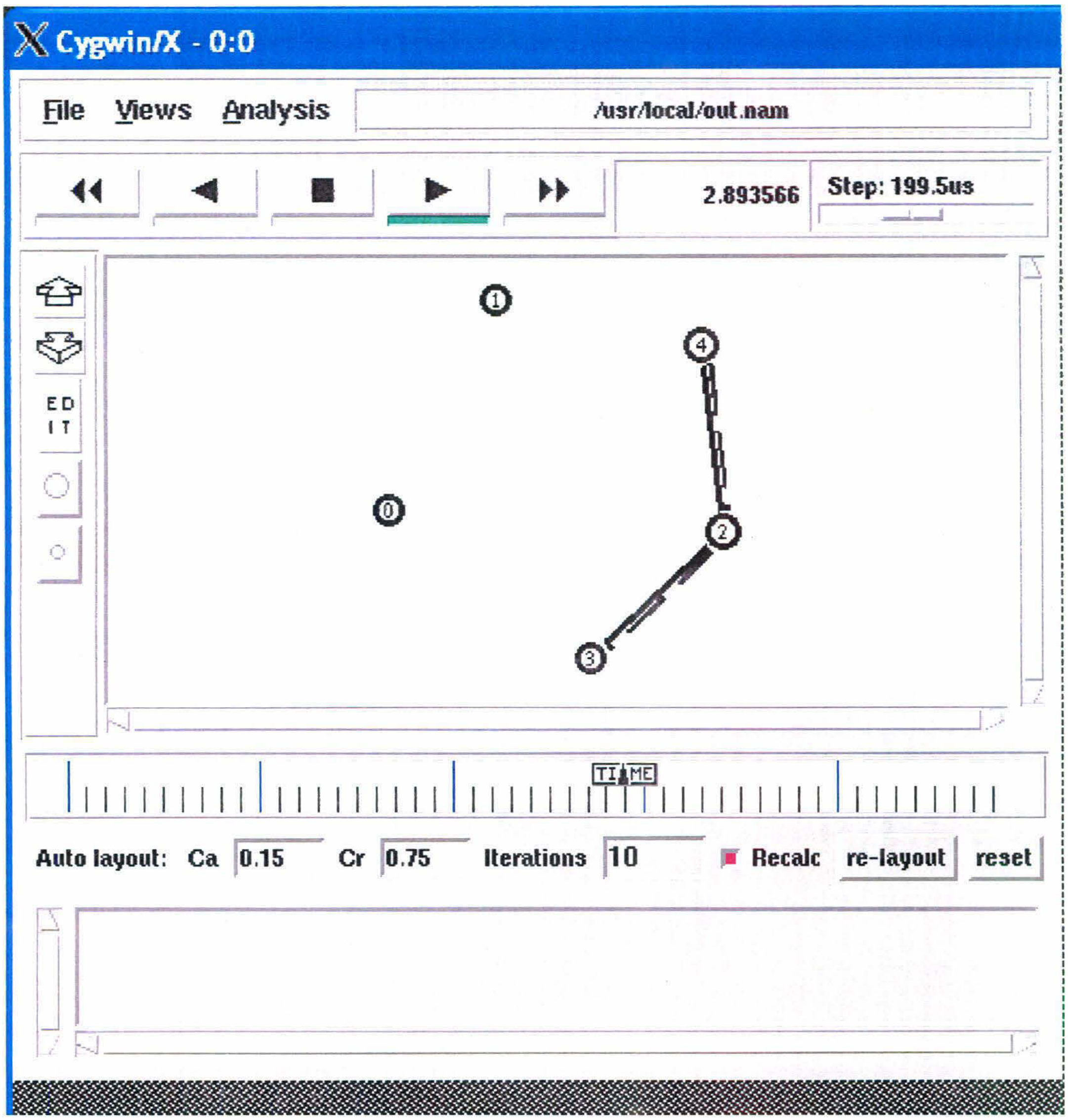


Figure 4.5 Node 1 is sending data packets to node 0

Here we can see hop by hop routing where the sent packets are forwarded from one Base Station to another till they arrive the last destination, and we can show here that when node (0) wants to send data packets to node (1) the same route will be followed till data packets arrive final destination as we can see from figure 4.6.

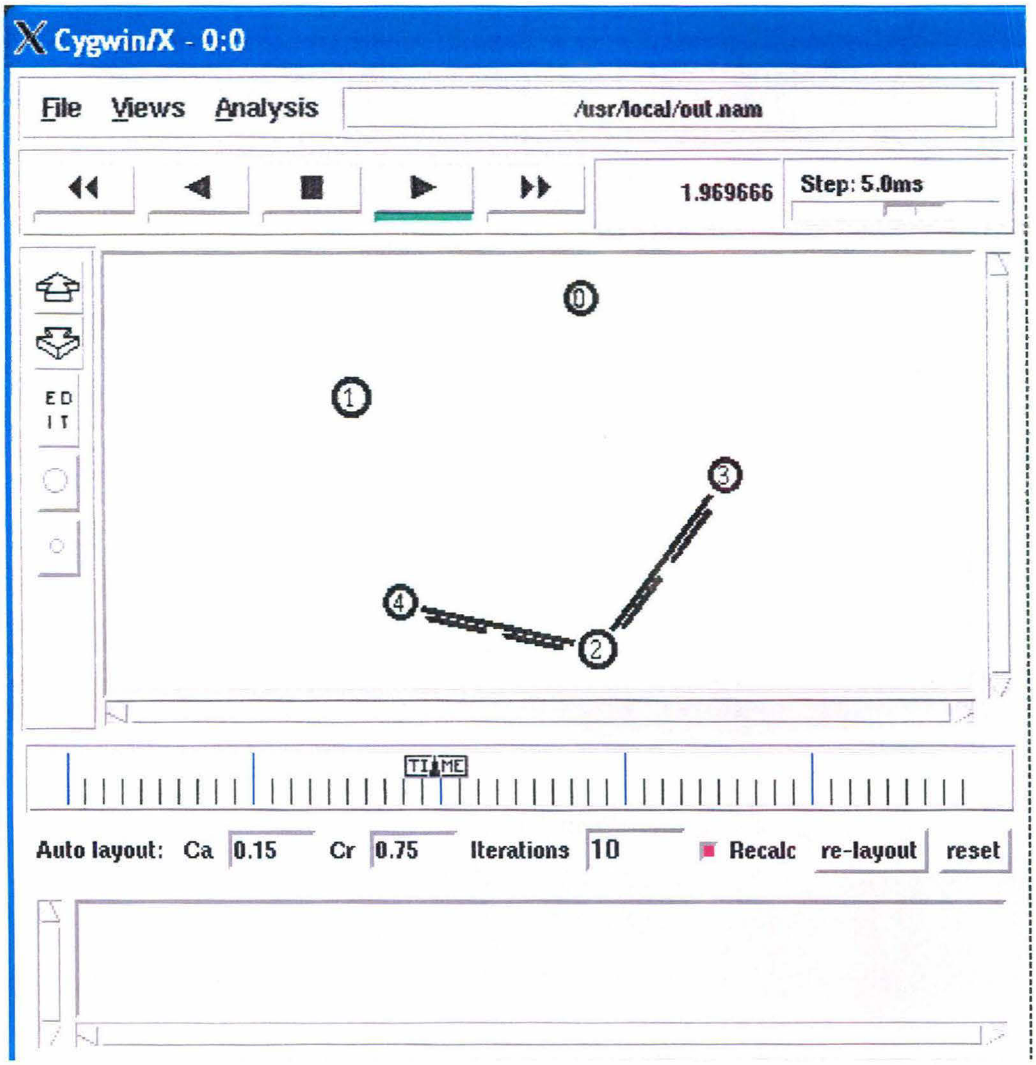


Figure 4.6 Node 0 sending data packets to node 1

Sending those packets can be done with Header Compression and Context transfer or without Header Compression and Context Transfer. Here we can see the difference in consumed Bandwidth through X-graph we got.

In what follows we will show the effect of packet size (uncompressed) packet on the consumed bandwidth. In our simulation experiment we have two wired channels connecting two Base Stations. TCL script, which we have written, calculates the consumed bandwidth and writes it to the output files named out0, out1. During our observation of packets sizes we have found out the following cases.

4.3.8 Effects of Header Compression on Bandwidth

In the Model, when the Mobile Host sends packets without Header Compression it will consume more bandwidth, where the consumed bandwidth by the Mobile Host is calculated and then the graphs are drawn as follows.

Without Header Compression

Sizes of sent packets are 10,000 bytes. The graph of consumed band width will be as figure 4.7

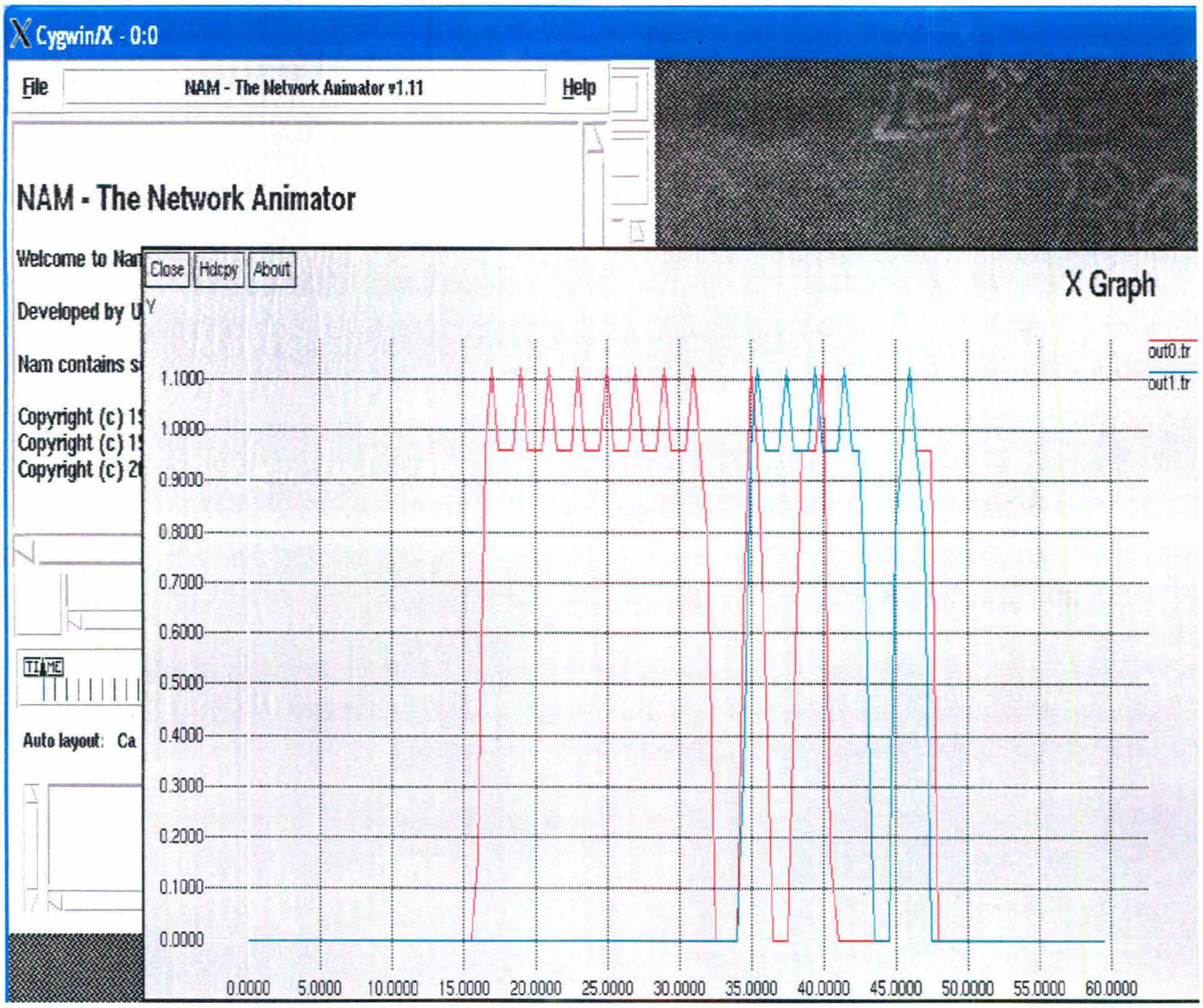


Figure 4.7 Consumed bandwidth in the case of no Header Compression

With Header Compression

We can see the difference in consumed bandwidth from figure 4.8. Noticeable as the maximum value in the first case is 11000, but the maximum value in the second case is 10,000.

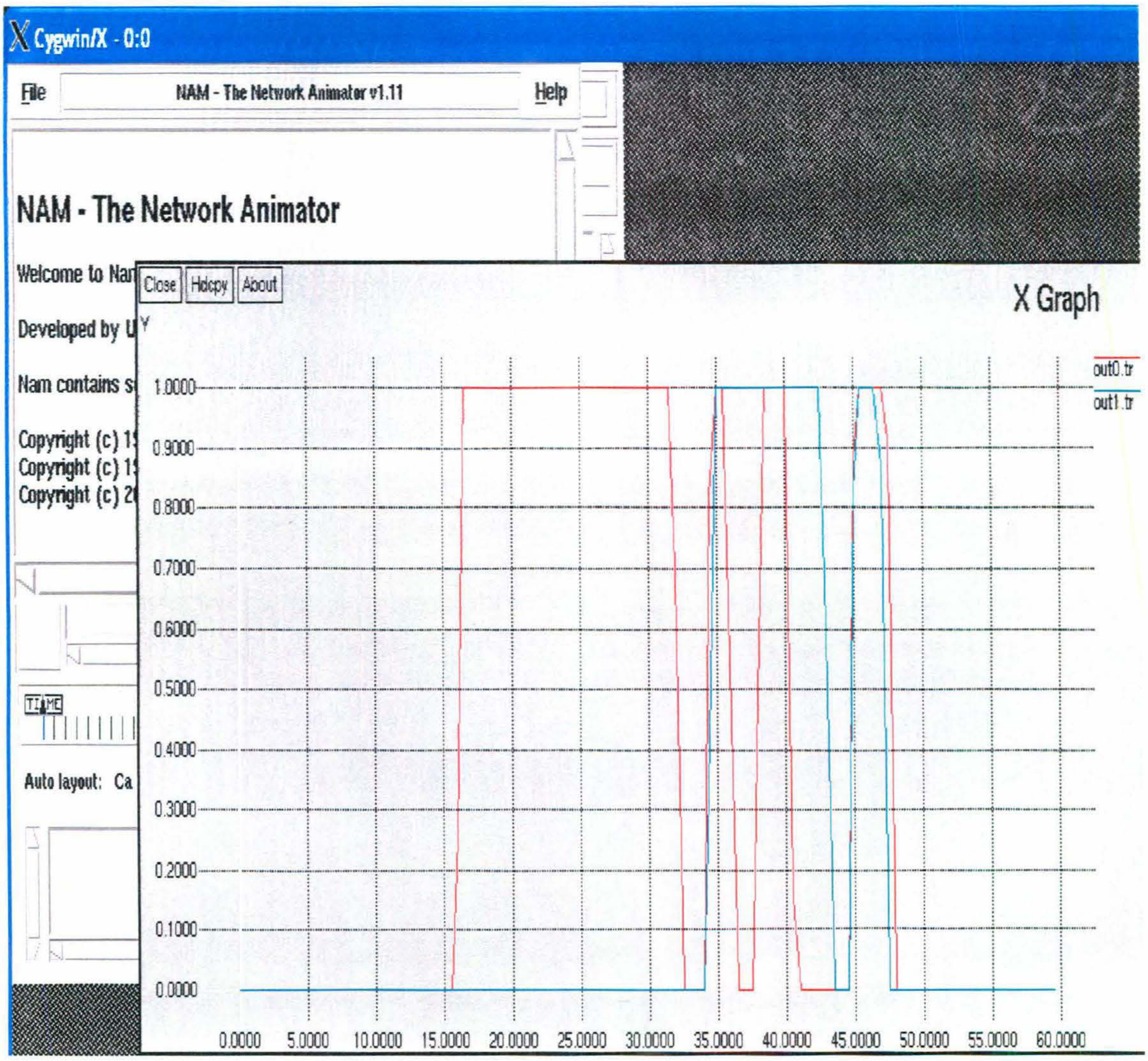


Figure 4.8 Consumed bandwidth in case of Header Compression

Chapter 5

Conclusion

Over the past few years, developments in Mobile Computing are enormous and all pointers around us refer to the fact that the future is governed by Mobile Computing techniques.

This dissertation work deals with Mobile Computing in general and Cellular IP in particular as application of Mobile Computing Environment. The work in chapter four discusses the QoS issue in this dissertation depending on the time delay that is required in Cellular IP network for two cases:

Case I with Context Transfer.

Case II without Context Transfer.

In the two cases, the size of the packet varies according to the payload in each packet while the size of the header is constant. We observe from results that the Header Compression Algorithm used is efficient and as a result Context transfer is efficient where small size packets are sent instead of re-establishing the context again. The efficiency of both Header Compression Algorithm and the Context transfer is useful in some places like Cellular IP networks where using Context transfer technique makes handoff operation more efficient with respect to time. On the other hand, performing handoff operation in micro mobility environment without Context transfer will force the Mobile Host as well as the Base Station to re-establish Context at any handoff operation, the thing which will consume time without profit.

While we are talking about Context transfer operation after Header Compression, we considered the probability of error during Compression operation. It is observed that changing Bit Error Rate affects the probability of error in two cases; in case of uncompressed header, and in case of compressed header.

Finite Automata is presented and discussed for our model (Cellular IP model). It shows all operations and all the states through which the Mobile Host goes till the end and data transfer takes place between the particular Mobile Host and Correspondent Node (CN) in another cell.

Cellular IP Model, as we have noticed, inherits the features of Cellular networks and achieves a good performance and good mobility management for its users. So it is to conclude that the proposed method has good achievement with respect to time as obvious from the previous comparison.

In section 4.3 in chapter four, we used NS-2 Simulator which we operate under Windows operating system using CYGWIN interface.

Cellular IP Model built here consists of three Base Stations and two Mobile Hosts. Every Base Station controls one cell which we can't simulate (using this simulator the limits of the cell can't be shown). In this model, one Mobile Host is sending data packets to another Mobile Host in another cell. We have simulated the compression operation by decreasing the size of the sent packet and every time calculates the bandwidth consumed (in case of data compression and in the case of uncompressed data). This means in case of context transfer and without context transfer by the X-graph (figure 4.7, figure 4.8) we observe that the bandwidth consumption decreases using Header Compression technique.

As a conclusion we can say that the proposed model could achieve improvement of time taken to make a session by suing context transfer before the Mobile Host handoffs to another cell. Also improvement is in the bandwidth consumed. As a result this proposal gives a good achievement in sense of Quality of Service that takes care of time taken and bandwidth.

The future work is to concentrate more on QoS features of Cellular IP.

References

[1] Asoke K Talukder *"Mobile Computing: Technology, Applications and Services creation"*. Tata McGraw-Hill Publishing Company Limited.

[2] http://www.cisco.com

[3] http://www.portal.acm.org

[4] http://EFFNET.com

[5] http://www.it.iitb.ac.in/xnet/mobile_ip/html/homepage.html

[6] Andrew T.Campbell, Javier Gomez, Sanghyo Kim, Andras G. Valko, Zoltan R. Turanyi, Technical University of Budapest *"Design, Implementation, and Evaluation of Cellular IP"*. www.csie.ncnu.edu.tw/~ccyang/WirelessNetwork/Papers.

[7] J. Kempf, Ed RFC 3374 *"Reasons For Performing Context Transfers Between Nodes in an IP Access Network"* September 2002.

[8] http://www.cse.wustl.edu.

[9] V. Jacobson/1/ LBL RFC 1144 *"Compressing TCP/IP Headers for Low-Speed Serial Links"* February 1990.

[10] http://www.it.iitb.ac.in/~jaju/tutorials/net/tcpip/node6.html.

[11] Ha Duong, Arek Dadej and Steven Gordon. Institute for Telecommunication Research, University of South Australia *"Transferring Header Compression Context in Mobile IP Networks"*. www.itr.unisa.edu.au/~sgordon/doc.

[12] http://www.beedub.com/book 1999.

[13] http://www.wekipdia.com.

[14] The VINT Project *"The ns Manual (formerly ns Notes and Documentation)* " March 1, 2007.www.isi.edu/nsnam/ns/doc.

[15] Yuh-Rong Leu and Chun-Lai Cheng Network and Communication Lab, Institute of Information Industry, Taipei, Taiwan, R.O.C *"Implementation Consideration for Mobile IP"*.

[16] Son Vuong and Laurent Mathy, Department of Computer Science, the University of British Columbia *"Simulating The Mobile-IP Protocol Using Wave"*. ieeexplore.ieee.org/xpls/abs_all.jsp.

[17] Network Working Group, C. Perkins, Editor, Request for Comments: RFC 2002 *"IP Mobility Support"*.

[18] Network Working Group, S. Casner, Cisco Systems, Request for Comments: RFC 2508 *"Compressing IP/UDP/RTP Headers for Low-Speed Serial Links"*.

[19] R.K.Ghosh *"Mobile Computing"* CSE 100, April, 2005._www.cse.iitk.ac.in/users.

[20] Marco Carli, Alessandro Nrli, Alessandro Neri, Andrea Rem picci, University of Roma Tre *"Mobile IP and Cellular IP Integration for Integration for Inter Access Network Handoff"*. ieeexplore.ieee.org/iel5/7452/20276/00936594.pdf.

[21] http://www.comet.columbia.edu/cellularip.

[22] Eriko Nurvitadhi, Ben Lee, Chansu Yu, and Myungchul Kim "Adaptive Semi-Soft Handoff for Cellular IP Networks". www.academic.csuohio.edu/yuc/papers.

[23] Jeremy Lilley, Jason Yang, Hari Balakrishnan, Srinivasan Seshan, MIT laboratory for computer science "*A unified Header Compression Framework for Low-Bandwidth Links*". www.nms.lcs.mit.edu/papers/headerpaper.pdf

[24] http://www.comet.columbia.edu/cellularip.

[25] Aga Zhang, Dependable Lab "*NS2 Tutorial*". www.dcl.ee.ncku.edu.tw/~aga/files.

[26] Cedric Westphal, Nokia Research Center, Mountain, CA "*Improvements on IP Header Compression*". people.nokia.net/cedric/Papers/globecom2003.pdf.

[27] M.Georgiades, H. Wang, R. Tafazolli, "*Security of Context Transfer in Future Wireless Communications*". www.ambient-networks.org/docs/Sceurity_of_Context

[28] Univ-Doz Dr. Karl Entacher and Dipl-Ing. (FH) Bernhard Hechenlietner, University of Applied Sciences and Technologies "*On Shortcomings of the ns-2 Random Number Generator*". www.st.inf.tu-dresden.de/aquila/files/pub/cnds2002-spu-ns2_rng_shortcomings.pdf.

[29] Zoran Kostic Xiaoxin Qiu and Li Fung Chang, Wireless Systems Research, AT&Labs-Research "*Impact of TCP/IP Header Compression on the Performance of a Cellular System*". ieeexplore.ieee.org/iel5/7252/19563/00904643.pdf.

[30] Changli Jiao, Loren Schwiebert, Golden Richard "*Adaptive Header Compression for Wireless Networks*". www.doi.ieeecomputersociety.org/10.1109/LCN.2001.990812.