A NEW ALTERNATE ROUTING SCHEME WITH ENDPOINT ADMISSION CONTROL FOR LOW CALL LOSS PROBABILITY IN VOIP NETWORKS

A Thesis by

Sandipan Mandal

B.E., SGGS Institute of Engineering and Technology, Maharashtra, India, 2000

Submitted to the Department of Electrical and Computer Engineering
and the faculty of the Graduate School of
Wichita State University in partial fulfillment of
the requirements for the degree of
Master of Science

July 2006

A NEW ALTERNATE ROUTING SCHEME WITH ENDPOINT ADMISSION CONTROL FOR LOW CALL LOSS PROBABILITY IN VOIP NETWORKS

I have examined this final copy of this thesis for form and content, and recommend that it be accepted in partial fulfillment of the requirement for the degree of Master of Science with a major in Electrical Engineering.

____

Ravi Pendse, Committee Chair

We have read this thesis
and recommend its acceptance

____

John Watkins, Committee Member

____

Krishna K. Krishnan, Committee Member

DEDICATION

To my parents, my sister, and my dear friends

If we knew what it was we were doing, it would not be called research, would it?

– Albert Einstein

ACKNOWLEDGMENTS

I would like to thank my advisor, Dr. Ravi Pendse, for his personal attention, encouragement, patient guidance and support throughout the course of my study at Wichita State University. I would also like to thank my committee members, Dr. John Watkins and Dr. Krishna K. Krishnan, for their helpful comments and suggestions on all stages of this project.

I would like to thank Mr. Amarnath Jasti and Mr. Nagaraja Thanthry for their support, precious time, encouragement, and enthusiastic supervision during this work. Last but not least, I thank my parents, sister, and my friends for their constant support and encouragement.

ABSTRACT

Call admission control (CAC) extends the capabilities of Quality of service (QoS) tools which protect voice traffic from the negative effects of other voice traffic. It does not allow oversubscription of a Voice over Internet Protocol (VoIP) network. To achieve better performance for efficient call admission control, various dynamic routings are being proposed. In the dynamic routing mechanism, the condition of the network is learned by observing the network condition via the probe packets and according to the defined threshold, routes are chosen dynamically. In such schemes, various combination of route selection is used such as two routes are used where one is fixed and other is random or two random routes are chosen and after observation one is chosen if it passes the test. Few schemes use a route history table along with the two random routes. But all have some issues like it selects random routes (not considering the number of hops), does not process memorization before admission threshold test, it calculates all selected paths regardless of the fact that they are selected or not, thereby wasting central processing unit (CPU) time and since these uses two routes so obviously the call admission probability is less.

In this thesis work, a new dynamic routing scheme is proposed which considers a routing history table with endpoint admission control increasing the call admission probability, makes call establishment time faster and it saves valuable CPU resources. The proposed scheme considers a combination of three routes with routing history table—one is the direct route and the other two are selected randomly from all available routes and the routing history table is used to memorize the rejected calls. CAC tests like Admission Threshold were performed on the selected routes. Various parameters such as delay, packet loss, jitter, latency etc from the probe packets

are used to carry out the tests. Performance of the proposed scheme with respect to other dynamic routing schemes is studied using a mathematical / analytical model. Also, effect of arrival rate probe packets on utilization, busy period, waiting period, acceptance probability of calls, probe packets, and the number of successful calls was also studied.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS / NOMENCLATURE

AT	Admission Threshold
ATA	Analog Telephone Adapter
ATM	Asynchronous Transfer Mode
BB	Bandwidth Broker
CAC	Call Admission Control
CCM	Cisco Call Manager
CPU	Central Processing Unit
DPS	Dynamic Packet State
DR	Direct Route
FDM	Frequency Division Multiplexing
ICPIF	Impairment Calculated Planning Impairment Factor
IETF	Internet Engineering Task Force
IOS	Internetwork Operating System
IP	Internet Protocol
ITU	International Telecommunication Union
LAN	Local Area Network
MGCP	Media Gateway Control Protocol
PSTN	Public Switched Telephone Network
QoS	Quality of Service
RIRT	Route Information Refresh Time
RSVP	Resource Reservation Protocol

RTCP	RTP Control Protocol
RTP	Real-time Transport Protocol
SAA	Service Assurance Agent
SIP	Session Initiation Protocol
TDM	Time-Division Multiplexing
TR	Third Route
UDP	User Datagram Protocol
VoIP	Voice over Internet Protocol
WAN	Wide Area Network

CHAPTER 1

INTRODUCTION

1.1 Voice over Internet Protocol (VoIP)

VoIP is a method for taking audio signals (audio) and converting them into digital data signal which can be transmitted over the Internet. It is the routing of voice traffic over the Internet or through any Internet Protocol (IP) based network. The protocols which are used to carry these voice signals are known as VoIP protocols such as SIP (Session Initiation Protocol), H.323, MGCP (Media Gateway Control Protocol) etc. It is a revolutionary technology which has the capability to completely rework the world's telephonic systems. Various VoIP providers such as Vonage and AT&T have set up VoIP calling plans in markets throughout the United States.

In this technology, there are three main ways in which a call can be placed. The first is by using an analog telephone adapter (ATA). The ATA makes it possible to connect our standard telephone to our computer or our internet. The second is by using IP phones. These IP phones look same as our normal phones with a cradle, handset and buttons. But the basic difference is that it uses RJ-45 Ethernet connector instead of our standard RJ-11 connectors. The third type is computer-to-computer which is the easiest way to implement and use. Here we do not have to pay for even the long-distance calls. All that we need is the software, speakers, microphone, soundcard and an internet connection with decent speed.

Most companies which use the VoIP technology provide the features which normal phone companies charge extra such as call waiting, caller ID, call transfer, repeat dial, return call, three way calling, forwarding the call to a particular number, sending the call directly to the voicemail, giving the caller a busy signal etc. Our traditional Public Switched Telephone Network (PSTN) uses circuit switching technique and VoIP uses the packet switching technique. Circuit switching is the one which establishes a dedicated channel or circuit between the endpoints before the users can start communicating. In contrast, packet switching network do not need any kind of a circuit for the call to be established and it allows many pairs of endpoints to communicate simultaneously over the same channel. Figure 1.1 shows the general conversation between two persons via PSTN.

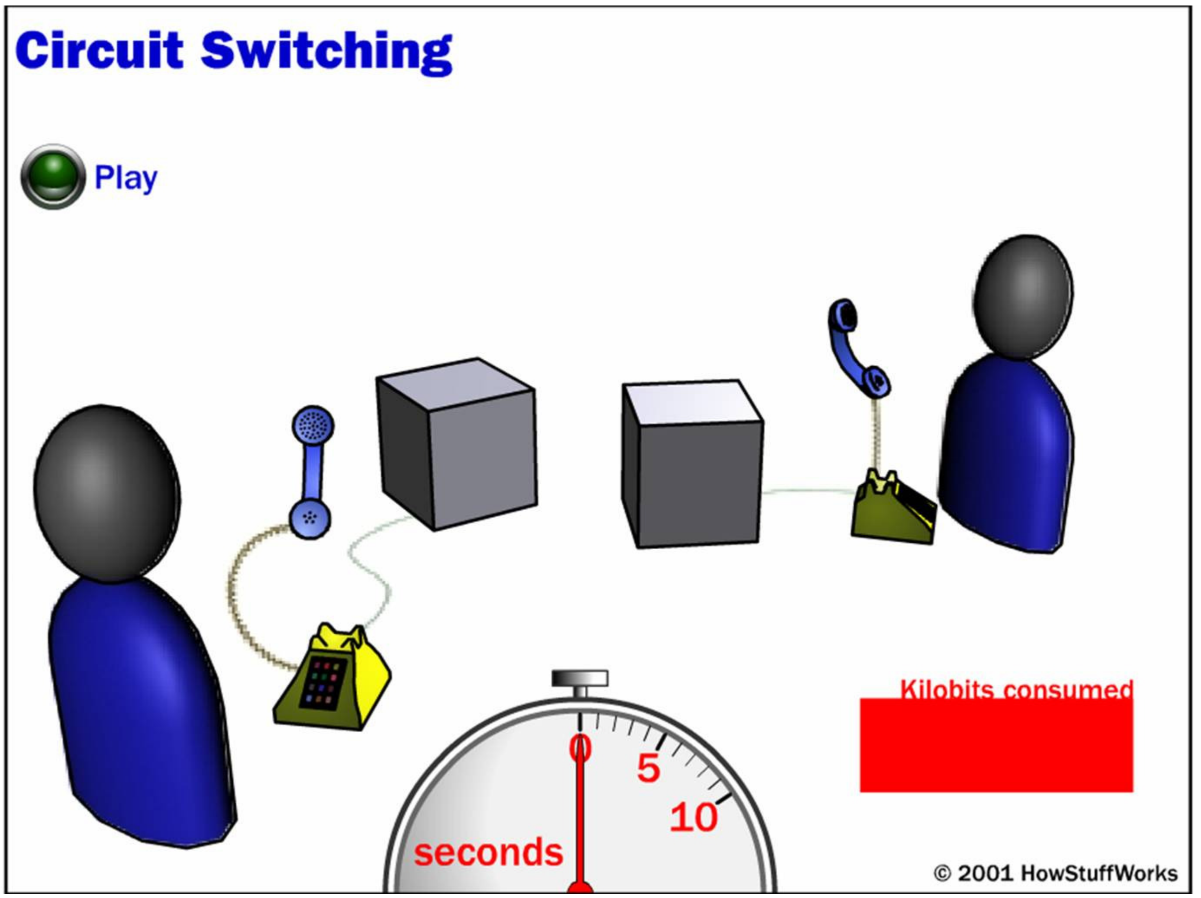

Figure 1.1. General voice conversations (PSTN) [42].

A typical telephone calls (circuit switched) works in the following way:

- Pick up the receiver and listen for a dial tone. This makes sure that we have a connection to the local office of our telephone carrier.
- Dial the number of the person with whom we wish to talk.

- The call is routed through the switch at the local carrier to the person whom we are calling.
- A connection is established between the telephone and the other person's line using several interconnected switches along the way.
- The phone at the other end rings, and somebody answers the call.
- The connection opens the circuit.
- Talk for a period of time and then hang up the receiver.
- When the caller / collie hang up, the circuit is closed and thereby freeing the line and all the lines in between.

Now we see a typical telephone call using VoIP over packet switched network.

Figure 1.2 shows the general conversation between two persons via PSTN.

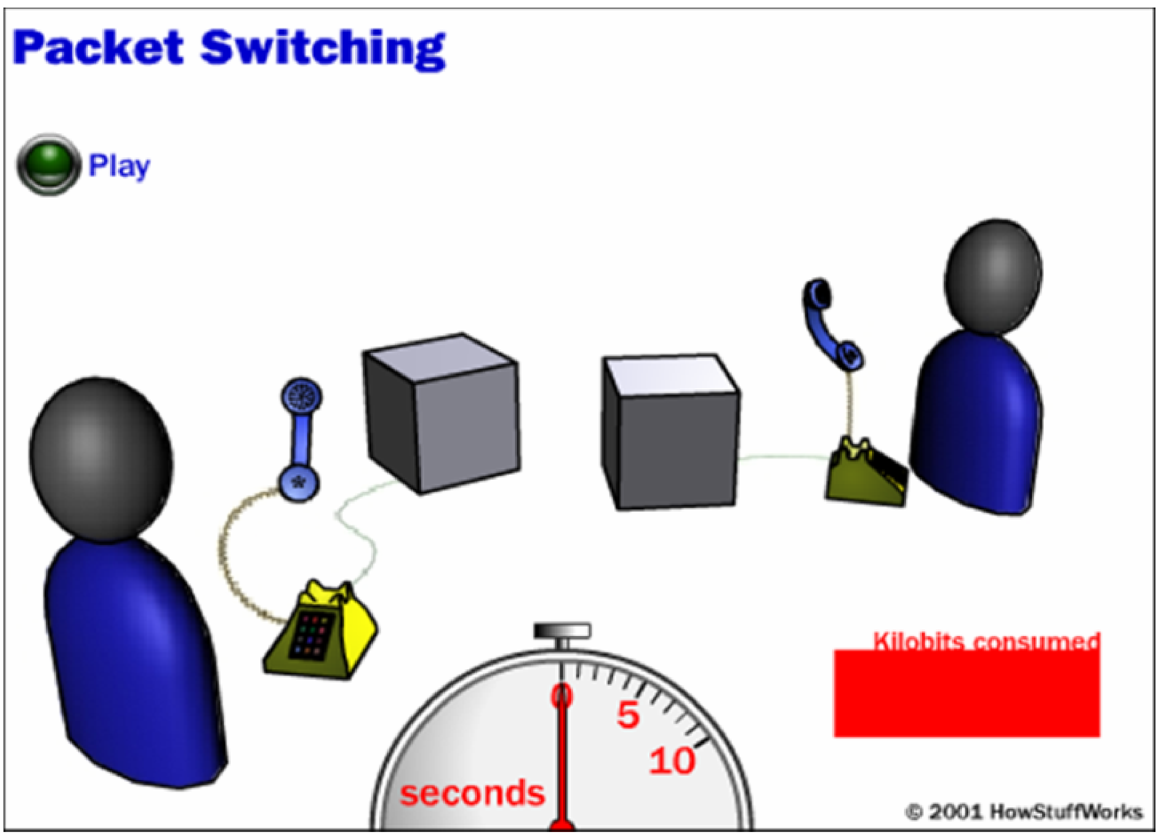

Figure 1.2. General voice conversations (VoIP) [42].

Various steps are as follows:

- ATA receives the signal and sends a dial tone. This makes sure that we have a connection to the Internet.
- Dial the phone number of the person with whom we wish to talk. These tones are first converted by the ATA into digital data and then temporarily stored.
- The phone number data is sent in the form of a request to the VoIP company's call processor. This makes sure that the phone number is in a valid format.
- The call processor determines to whom to map the phone number. In the process of mapping, translation of the phone number to an IP address takes place. Two devices on either end of the call are connected by network devices. On the other end, a signal is sent to the person's ATA, telling it to ask the connected phone to ring.
- Once the person picks up the phone, a session is established between our computer and the other computer. Each system knows to expect packets from the other system. In between the source and the destination, the normal Internet handles the call as if it were e-mail or any other web traffic. Each system must use the same protocol to communicate.
- During the conversation, packets are transmitted back and forth when there is data to be sent. The ATAs at each endpoint translates these packets as they are received and converts them to an analog audio signal that we finally hear. ATA keeps the circuit open between itself and the analog phone while it forwards packets to and from the IP host at the other endpoint.
- Finish talking and hang up the receiver.
- When we hang up, the circuit is closed between our phone and the ATA.

- The ATA sends a signal to the networking device which is connected to the caller / callie and terminates the session.

1.2 Advantages

VoIP technology has many advantages over our traditional phone systems. One of the most advantages of packet switching is that present network already understands the technology. By migrating to this technology, telephone networks automatically gain the ability to communicate the way computers do. Second, phones via VoIP are free or more specifically costs less than the PSTN. These cost savings are because of the fact that it uses a single network to carry voice and data. VoIP to VoIP phone calls are free but VoIP to PSTN calls involve costs for the VoIP user. Third, it can perform a few tasks that are more difficult to achieve with traditional PSTN networks. For example, incoming calls can be routed to any VoIP phones, regardless of the position of that phone in the network. so, we can make and receive phone calls irrespective of the location. Fourth, call center agents can use VoIP from any part of the world and it provides extra features such as call forwarding, call-waiting etc., for which telephone companies charge. Fifth, it has other services available over the Internet such as inclusion of video conversation, data file exchange in parallel with the conversation and audio conferencing.

1.3 Drawbacks

VoIP technology has some drawbacks that have led to the belief that it is not ready for widespread deployment such as reliability, Quality of Service (QoS), delay, jitter packet loss and echo. This technology is not as reliable as the traditional PSTN as it has battery backup at telephone exchanges whereas VoIP depends on the broadband modems and other equipments powered by household electricity which may be subject to outages. It suffers from packet loss due to route loss in the network. Since IP does not provide a technique to ensure that data packets are delivered in sequential and proper order, or that Quality of Service (QoS) is guaranteed, its implementation faces problems with latency and jitter. It has jitter problems too. Jitter is an abrupt and unwanted variation of one or more signal characteristics, such as the interval between successive pulses, the amplitude of successive cycles, or the frequency or phase of successive cycles. It has QoS issues such as some broadband connections have quality which may not be of the desired level. In those cases, if IP packets are lost or delayed at any point in the network between VoIP users,there will be a momentary drop-out of voice. Emergency calls also become a challenge for VoIP. It is geographically difficult to locate network users because of the nature of IP. It has security issue as the majority of VoIP does not yet support encryption. As a result, it can be an easy target for hackers and other miscreants.

1.4 Call Admission Control

Call Admission Control (CAC) is a concept that applies to voice traffic only—not data traffic. If an influx of data traffic oversubscribes a particular link in the network, queueing, buffering, and packet drop decisions resolve the congestion. The extra traffic is delayed until the link is again available to send / receive the traffic and if the traffic is dropped, the protocol or the end user initiates a timeout and requests a retransmission of the information.

Network congestion cannot be resolved in this manner when real-time traffic, sensitive to both latency and packet loss, is present, without jeopardizing the quality of service (QoS) expected by the users of that traffic. For real-time delay-sensitive traffic such as voice, it is better to deny network access under congestion conditions than to allow traffic onto the network to be dropped and delayed, causing intermittent impaired QoS and resulting in customer dissatisfaction.

CAC is therefore a deterministic and informed decision that is made before a voice call is established and is based on whether the required network resources are available to provide suitable QoS for the new call.

Congestion control in voice traffic is done by CAC. It is also known as preventive congestion control procedure. It is not only used in the call setup phase but also in the prevention of congestion in connection-oriented protocols such as ATM. Various CAC mechanisms are available today and are discussed in the next section.

Call admission control is basically used in VoIP to ensure quality of service and to prevent loss of packets. The difference is that VoIP uses RTP/UDP/IP which are connectionless protocols. In such cases, integrated services with reserve resources for packet flow through the network, using controlled-load service is used so as to ensure that the call is dropped if the flow of the call cannot be supported. As we know that, in external link , CAC comes into play to reject calls when either there is not enough CPU processing power, or the network traffic becomes more than the prescribed thresholds, or the number of calls in the network at a given period of time exceeds the prescribed limit.

A number of QoS are present in the Cisco internetworking operating system (IOS) other than CAC for ensuring necessary low latency and proper delivery for voice traffic. These mechanisms are policing, queuing, traffic shaping, packet marking, and fragmentation. Cisco IOS differ from CAC in the following ways:

- Protection of voice traffic from data traffic when both types of traffics are contending for the same available resources.
- Only deals with traffics that is already present in the current network

1.5 CAC Mechanisms

Several different solutions have come into prominence for deploying CAC on packet networks. But unfortunately, none of them solves the entire problem, but they all are helpful to address a particular aspect of CAC.

Local Call Admission Control mechanism [41]: This mechanism is used on the outgoing gateway. The decision is primarily based on nodal information like Local Area Network (LAN) or Wide Area Network link (WAN). It is based on the fact that if the local network link is down, then it is useless to apply any kind of algorithm based on the condition of the network. It is mainly applied in the configuration where more calls than the prescribed limit are not allowed and these limits are fixed. For example if the network cannot bear more than five calls then the local node can be configured so that it dose not allow more than five calls

Resource-based Call Admission Control mechanism [41]: This mechanism includes calculation of resources needed or available, and reserving them for the call. Here, resources mean the available bandwidth, DS0 time slots on the Time Division Multiplexing (TDM) links, and CPU power / memory. Restrictions can be placed on any of these resources on any one or more nodes where the calls will travel to the destination.

Measurement-based Call Admission Control Mechanism [41]: This mechanism is arguably the best one. It basically looks ahead to determine the state or health of the network in order to determine whether a new call can be allowed or not. Here, gauging the state of the network means that probes are sent to the destination that is normally the terminating gateway or the gatekeeper. When these probes return to the outgoing gateway, they bring in some valuable information about the network while traversing the network towards the destination. Some of the information can be delay or loss characteristics. Service Assurance Agent (SAA) probes are used to find the condition of the network. They travel the given network to a destination IP address and calculate the delay and packet loss characteristics. The values which are obtained by the probes are sent to the outgoing gateways, which in turn use these values to make a decision about the status of the network and finally to decide whether the network can withstand a voice call or not .The attributes of measurement-based CAC mechanisms that are obtained from their use of a SAA probe include the following:

- Since a SAA probe is an IP packet that travels to an IP destination address, all the measurement based CAC mechanisms apply to the VoIP traffic only.
- As mentioned earlier, that these probes are sent in the network, so there will be a certain amount of overhead traffic generated.
- For a CAC decision to take place, it is required to wait for the probes to be dispatched as well as for their arrival. Therefore a small post-dial delay for the call may be experienced. But this can be avoided by properly designing the network.

1.6 SAA Probe packets vs. Ping packets

One can argue that ping packets can be easily used instead of overloading the network by sending probe packets. Basically SAA are almost similar to ping IP mechanisms. A ping packet does not look like a voice packet either in size or in the sense of protocol. Even the qualities of service mechanisms which are deployed in the network classify these types of packets as voice packets. The delay and other characteristics which are experience by ping packets may not be experienced by the voice packets. Therefore, if ping packets are used to gauge the condition of the network, then the result will be crude and the admission decision will not be accurate.

1.7 SAA Protocol

The SAA protocol [36] is a client/server protocol which is defined on a User Datagram Protocol (UDP). The sender or client builds and sends the probe, and the target device or the server returns the probe packets to the sender or the client. These probe packets which are used for CAC, go out randomly on ports that are selected from a UDP-defined port range (16,384 to 2,767 0. They basically use a packet size based on the codec that the call will use. They can even use a set the desired IP precedence and a full RTP/UDP/IP header, which in turn simulates a real header carried by a real voice packet. The SAA probe uses the Real Time Control Protocol (RTCP) port (odd RTP port number) by default.

1.8 Impairment Calculated Planning Impairment Factor (ICPIF)

The Impairment Calculated Planning Impairment Factor (ICPIF) is an International Telecommunication Union (ITU) standard. The calculation based on various parameters such as network delay and packet loss, gives a value that can then be used to gauge the condition of the network. It can be used as a threshold for the call admission control decision based on the network.

ICPIF values given by ITU g.113 are as follows:

5	Very good
10	Good
20	Adequate
30	Limiting case
45	Exceptional limiting case
55	Customers likely to react strongly

1.10 Endpoint Call Admission Control

The end-point call admission control is measurement-based call admission control. Here, measurements are taken by the end nodes on the flow of packets, which are then injected into the network to guage the source to the destination path. It has two features. First, it does not rely on any additional procedure in internal network routers other than the capability to apply different service priorities to probing and data packets. Second, the connection admission decision is based on the analysis of the probing flow delay variation statistics.

IP telephony (real-time services) has very stringent delay and loss requirements which are required to be met throughout the call time. The typical requirement is 150 ms mouth-to-ear delay for a quality voice. If the delay components are analyzed over the source and destination paths then we see that up to 100 to 150 ms can be utilized for the compression, jitter compensation, packetization, propagation delay, etc. which effectively leaves no more than 10ms for queuing delay through all the routers in the path.

Call admission control is one of the few elements which are required for providing QoS in the network. If there is no control over number of calls which are currently active at a given point of time, then the demand of overall traffic might be higher than the traffic supported by the network, which will lead to the degradation of calls. For example, delay and packet losses may be higher than the prescribed limit. With the network having the support of call admission control, the algorithm calculates the available bandwidth in each link and then makes a decision on whether there are sufficient resources and whether the new call will be allowed or dropped.

The next section discusses about the different CAC schemes available today. The problems faced by these schemes are also described.

CHAPTER 2

LITERATURE REVIEW

2.1 Different schemes

Many different types of routing schemes have been proposed. One of the schemes concentrated on only one element, the admission control functions [3]. Although these functions are pushed to only one element it is difficult for that element, to manage all reservations in a network and store the information about all paths, elements and flows. Trying to profit from the best of two Internet Engineering Task Force (IETF) architectures and reduce the scalability problems, several novel architectures and algorithms have been proposed like similar to the Dynamic Packet State (DPS) [4] whereby state information is inserted into the packet header; aggregation [5] that performs only admission control for a group of flows, decreasing the number of signaling messages, and egress admission control [6] which is based on passive monitoring of the network.

Although these mechanisms provide an efficient service model and scalability, they require specific functionalities in the core and edge routers, such as insertion of packet state in the headers, a special scheduler to be implemented in each router and rate monitoring. To prevent the use of a signaling protocol and special packet processing in core nodes, a call admission control mechanism based on probing was proposed. Here a test flow is inserted into the network to measure its congestion level.

2.2 Resource Reservation Protocol (RSVP)

RSVP establishes and maintains the resource reservation in an IntServ network. Its main aim is to communicate with the resource demands and allocate reservations to each router along the flow's path. But this scheme has few problems. Although it seems that this signaling protocol is very strong in providing better QoS support, but the problem is that it is not scalable. This is due to the fact that it is required to maintain a flow state in each router throughout the path flow, and all routers from the source to destination participate in the signaling protocol. Another fact is that the number of RSVP messages processed is directly proportional to the number of flows in the network and that is why the bandwidth must be reserved in each router on a per-flow basis. The above mentioned two facts can lead to poor router performance.

2.3 Bandwidth Broker (BB) Based Admission Control

Bandwidth Brokers (BB) removes the need for any kind of QoS reservation states in the core routers. This is done by storing and managing the information in a central location. A BB [3] can be a software package installed in a router/switch or a simple router in the network. With this mechanism, the available bandwidth is calculated for each connection through information about active flows, their paths from source and destination and their traffic characteristics throughout the path traversed—all stored in BB. Flows with the same characteristics can be grouped together in various service classes in such a way that BB operations become faster and as a result increase the number of flows it can support. But this scheme has few problems. Although, the core routers are not required to perform admission control decisions, their technique needs to not only manage the overall network but also store information about all elements, flows, and paths in the network. This is difficult when only one element is involved. Therefore, if the network is very large, a distributed mechanism is preferable.

2.4 Dynamic Packet State (DPS)

In the technique DPS [4], flow state information such as reserved rate and variables used in the scheduling process is inserted into the packet headers, thus overcoming the need for per-flow signaling and state management. The ingress router does this by initializing the state information. The core routers, before forwarding it to the next hop, process each incoming packet based on the state carried on it and eventually update its internal state and the state in the packet's header But this scheme has few problems. Although the core routers are freed from maintaining per flow-state, a deterministic service is still provided since admission control is based only on the flow's rate inserted in the packet header. Because of this, the utilization is reduced significantly. Moreover, all routers in the flow's path are required to implement the same scheduling discipline.

2.5 Aggregation in Int-Serv

Aggregation [5] is used to reduce the number of signaling messages in IntServ architecture. In this technique, since the core routers only needs to maintain the reservation state of each aggregate so the admission control is only performed on an aggregated set of flows. For the aggregate, the RSVP protocol is used. Therefore, the core routers do not have to store the reservation state of the individual flows. But this scheme has few problems. This aggregation mechanism implies a tradeoff: If there is more aggregation, then more flows are not admitted and, as a result, the utilization decreases. If there is small aggregation, then the decrease in utilization is neglected but the number of signaling messages always remain on the higher side. Nodes are rarely needed to be signaled if the loads are relatively constant. Otherwise, signaling will be close to IntServ's one.

2.6 Measurement-Based Admission Control in Egress Routers

In this mechanism [6], admission control decisions are only performed by egress routers, without maintaining per-flow state in either egress routers or in core routers. Here the admission decisions are based only on aggregate measurements which are collected in the egress router.

The main technique applied here is to passively measure the available service in the end-to-end path. These measurements can incorporate the cross traffic effects without explicitly measuring it or controlling it by using a black box system model. But this scheme has few problems. Although the only router that needs to perform admission control decisions is the egress router, only a large-scale prediction of the congestion level is made. Because network conditions may change rapidly, QoS requirements may be degraded.

2.7 End-Point Admission Control Through Probing

In this mechanism [8], the admission of a new flow is performed by the end-hosts or egress/ingress routers. This is done via the inference of the network congestion state in the flow's path from the source to the destination. The source sends a packet stream called a probe packet to the flow's path with the same traffic characteristics of the flow that is requesting admission before a new flow is accepted. By observing the packet loss ratio, the delay, or delay variation, which normally verifies the network congestion level, a decision is made whether to accept or reject the flow. This is called probing. If the measured parameters are within the acceptable threshold (according to the required QoS), then the flow is admitted, otherwise, the flow is rejected. Here QoS functionalities are given to the end-points, thereby not requiring any kind of signaling protocol or special functions in the core or edge routers. Problem with this scheme: As probe packets are injected into the network, so a overhead is introduced also and as a result, the set-up time required to initiate a call becomes an issue, which is definitely a disadvantage of this technique.

2.8 Dynamic Routing Scheme

Dynamic routing technique [10, 34, 31] is one of the new mechanisms which are proposed to achieve better performance for deploying call admission control. In this type of method, condition of the network is learned by observing the condition of the probe packets and according to the pre defined threshold, we choose routes dynamically from the available routes to a particular destination.

In the next chapter, dynamic routing scheme is discussed in detail. It also discusses about the various problems it faces and my proposed schemes to counter those problems.

CHAPTER 3

METHODOLOGY

3.1 Dynamic Routing Scheme

The dynamic routing technique [10, 31, 34] is one of the new mechanisms proposed to achieve better performance for deploying call admission control. With this type of method, the condition of the network is learned by observing the condition of the probe packets, and according to the pre -defined threshold, routes are chosen dynamically from the available routes to a particular destination. In one of the proposed schemes, probe packets are sent simultaneously along with two candidate routes to the desired destination. The shortest route is chosen as the default route and an alternate route is chosen for the backup. This second route is chosen randomly from all available routes to the desired destination. Using a route table where the last rejected routes are memorized. Any of these rejected routes are not used as the second randomly chosen route, because it is likely that this same route will again be congested when the next call arrives. One constraint is that it will be valid only when the average call duration is short enough, so that the rejected route is very likely to be congested for all future calls. In today's network this is very common. It is common knowledge that VoIP traffic duration will be very much similar to that of traditional telephony. So, after some time, these rejected routes could provide enough performance to accept a new call since it can be safely assumed that the condition of the network changes with time. The other method proposes a new route selection scheme with a table of rejected routes, which is reset after a fixed period of time. This is valid because the network cannot be expected to be in the same state for a long time since its condition changes very rapidly in today's busy world. The next call admission control will examine other possible routes that are not present in the table of rejected routes. This results in more efficient use of the network resources, which ultimately leads to lower call-loss probability than the other existing mechanisms.

The basic and advanced dynamic routing CAC schemes [10, 31, 34] measure traffic congestion of the shortest route and also an alternate route. Even when the shortest route is likely to be congested, this route is used as a default candidate route. Therefore, both of these schemes have a problem of efficient network resource utilization when the shortest route is congested, because it is likely that a new call will not be accepted on the shortest route.

Proposed Methodology/Algorithm:

It is well known that the traffic volume of IP networks changes very frequently. Therefore, if the interval between successive calls in a network becomes longer, then traffic conditions can easily change. As a result, the best QoS route will also change from time to time. Also when the interval between successive calls is short, then the traffic volume of the network will also change very slowly. As a result, the busy route will be free of congestion as the traffic condition is changed during a certain period of time. The advanced dynamic routing CAC scheme does not use a route as an alternative, even after it is free of congestion.

Here, probe packets are sent along two routes that are selected randomly [10, 31]. If both routes satisfy the admission conditions, the lower packet loss route is adopted. If either or both of these routes does or does not meet the requirements, the routing history table at the originating node temporarily memorizes the rejected route/routes for the last call request. After a fixed time interval, the memorized route information is cleared. This time interval is called “route information refresh time” (RIRT) . A flow chart of the proposed scheme is shown in Figure 3.1

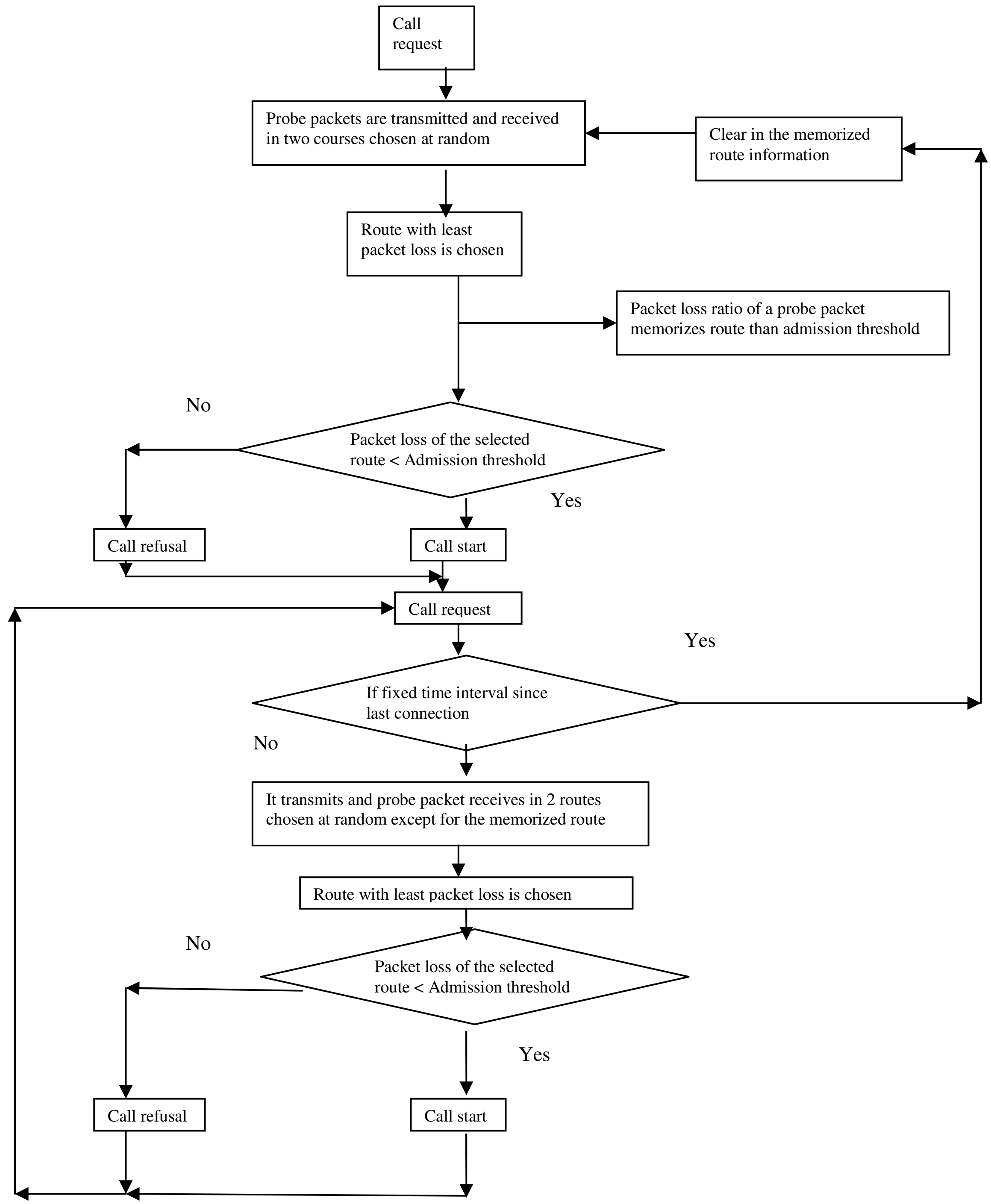

Figure 3.1. Algorithm in the referred paper.

3.2 Problems

Several problems are associated with this scheme:

1. It selects two routes randomly, not considering the number of hops. Therefore, in larger networks, it may happen that it chooses two paths with large number of hops even though there may be some shorter routes which are not congested.
2. It does the process of memorization before the admission threshold test, thus causing problems if it fails the test?
3. It calculates of all the selected paths regardless of whether they will be selected or rejected, thereby wasting time by calculating the rejected path and also using precious CPU time.
4. Since it uses two routes, the call admission probability is less.

3.3 Solution:

Three schemes are being proposed in this thesis:

In the first scheme (figure 3.2), the routes are partly chosen randomly. Here we choose first route (DR)—the shortest route is the default route and the other two are chosen randomly. To decrease the call loss probability, three routes instead of two routes (TR1 and TR2) are used. The Admission threshold (AT) is set for DR and TRs . The AT for the DR is set a bit lower than for the TRs since DR is very likely to be congested most of the time. Therefore, this route is made intentionally difficult to attain. The AT for the other two routes is fixed higher than the DR so that the possibility of a successful call increases. First, it checks for DR, then TR1, and then TR2. Routes that fail the AT test, are memorized for a certain period of time because of the changing nature of the network. After this fixed period of time, the memory is cleared so that the discarded routes can be considered again. Here different times are used for DR and TR. Less time is kept for the DR since it is supposed to change very rapidly (being the busiest route), than the TRs. The flowchart is as follows:

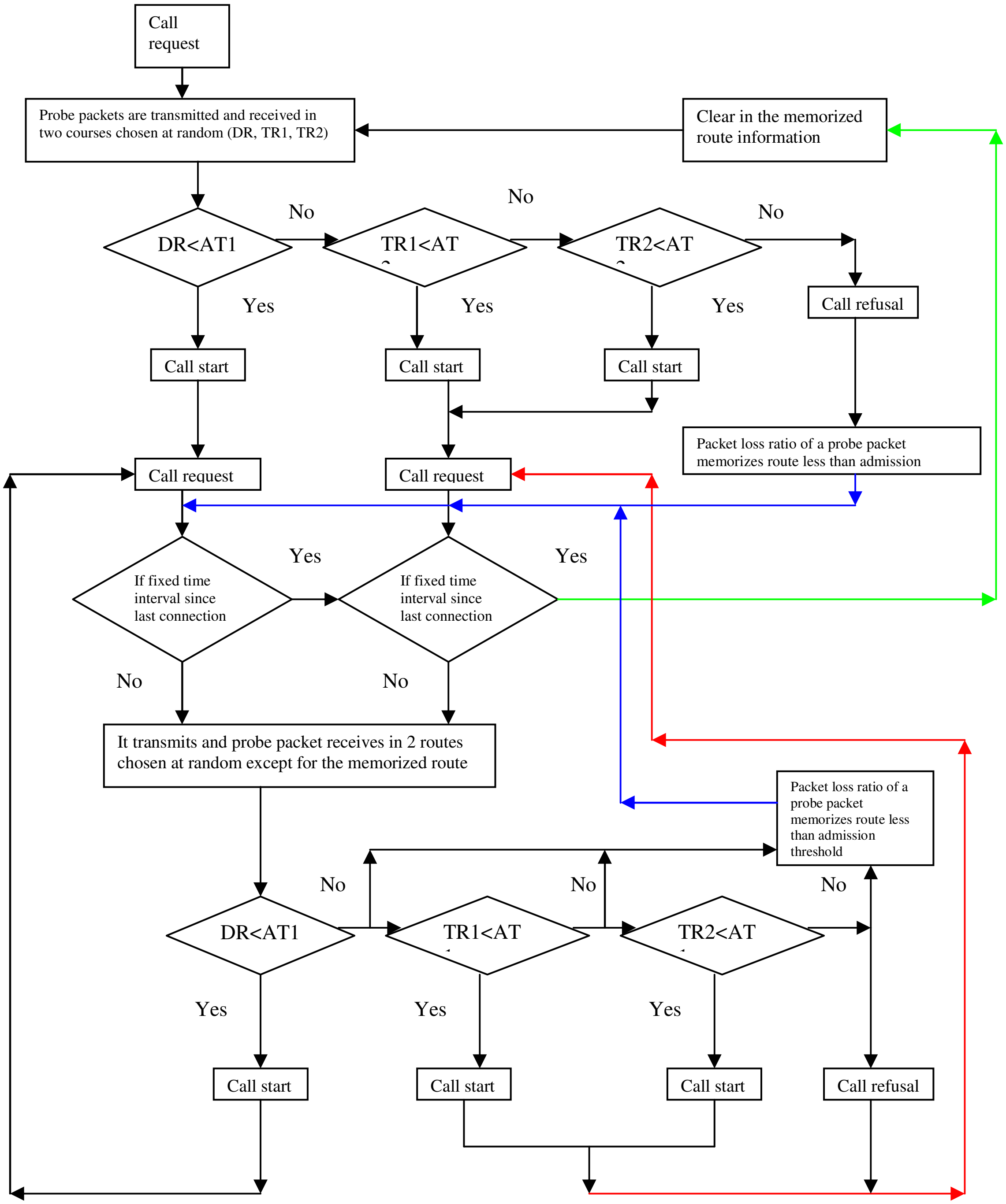

Figure 3.2. My proposed algorithm number 1.

The entire process is as follows: First the three routes are chosen. Then the AT test is done for the DR; if it fails, then DR is placed in the memory table for a fixed amount of time. Then the AT test is done for TR1; if it also fails, then it is sent to memory and then the test is carried out for the TR2 route. If the DR test (any test) is a success then the call is established and another call is requested. Then it checks to determine whether the fixed time has expired. If not, then again it chooses three routes except those in the memory and the same process continues. If the fixed time have expired, passed then the memory is cleared, and when the routes are selected, then these failed routes can again be considered for the test or call establishment. Since there are three routes to choose from, obviously the call drop probability becomes less compared to the two-route policy.

In the second scheme (figure 3..3), one DR is chosen as in the first scheme. But here, AT1 is chosen as the next shortest path and AT2 is randomly chosen. If the call is through the DR or the AT1 route, then the efficiency increases. If not, then AT2 is always available for the call. Since we know that the DR is supposed to be busier than the TR1, the AT for the DR is chosen to be less than the TR1. Therefore, the values of the AT for the paths are in the order DR<TR1<TR2. In this same way, time to keep in the memory table is also in the order DR<TR1<TR2, whereby less time is kept for the busier routes because of their tendency to change frequently. Routes are tested in that same order DR, TR1 and TR2. This scheme should give the best results because each route is set with proper values, keeping in mind the volatile and unpredictable network. It may take a bit longer time but more is hoped to be achieved than lost. The flow chart is as follows:

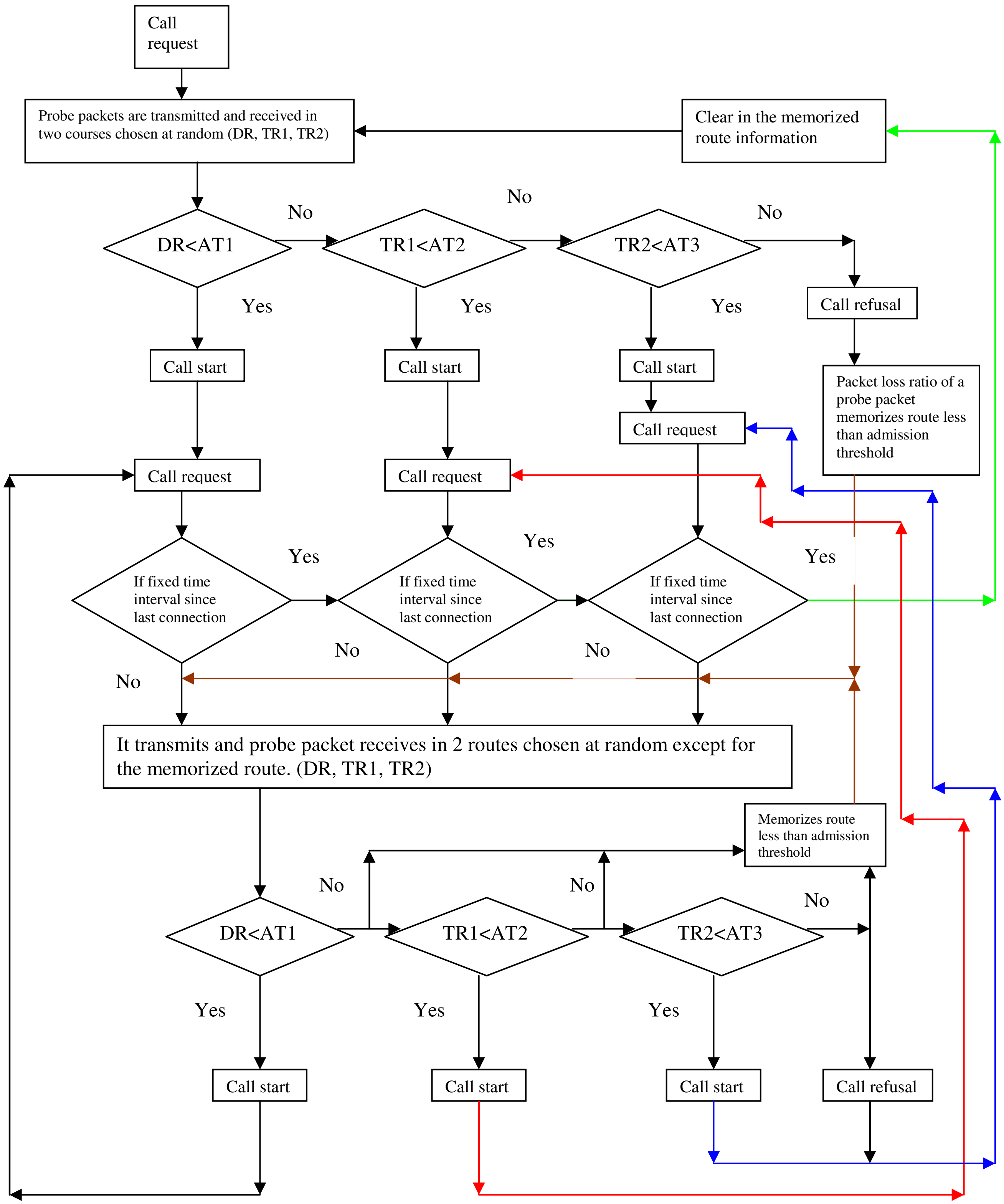

Figure 3.3. My proposed algorithm number 2.

In the third scheme (figure 3.4), everything is same as the first scheme except for the fact that the AT is same for all the routes. The flow chart is as follows:

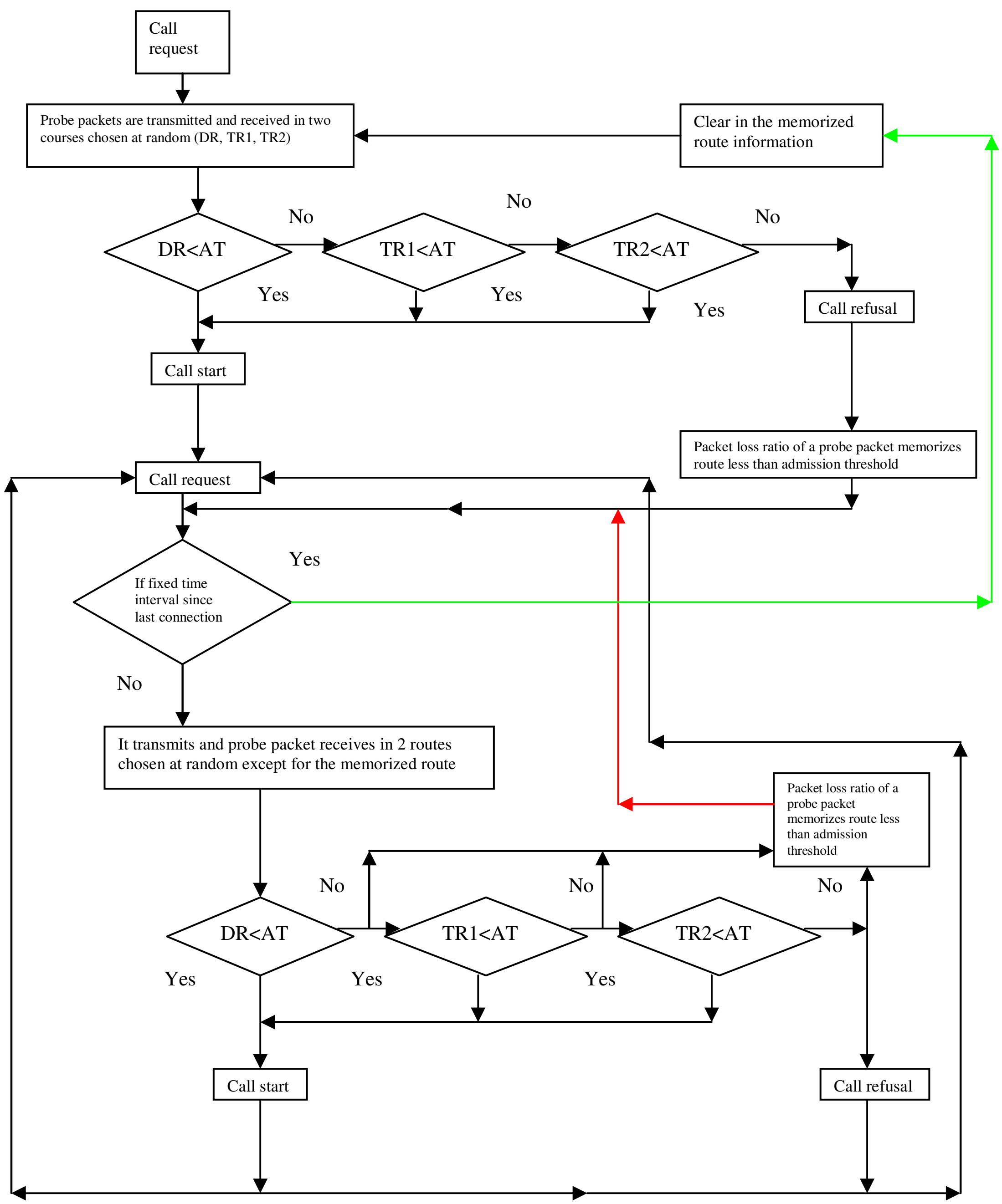

Figure 3.4 My proposed algorithm number 3

In the same way, the time to keep in the memory is also same for all the routes. This is supposed to be the faster one as not many complicated calculations are involved as same value is used for all three routes.

These proposed algorithms have some differences. The first scheme has the least call drop probability. The second scheme is the most complicated algorithm to implement as it uses different threshold values and these values have to be fixed by observing different network parameters. It may take a bit longer to implement than the other two, but it is the most efficient scheme out of these three. The third scheme is the fastest one because of the use of same threshold values.

The next chapter discusses the mathematical / analytical model to see how the algorithm fares with respect to the other already proposed algorithm. It also discusses about the probe rate on different network parameters for the proposed algorithm.

CHAPTER 4

MATHEMATICAL ANALYSIS & RESULTS

4.1 Mathematical Analysis I and Results (Algorithm)

Here, we see how the proposed algorithm fares better than the other algorithms that were proposed earlier.

Assumptions:

There are ten paths to particular destinations.

Values inside brackets [] are the congestion level in the path which is our Pme.

1->5: increasing order of congestion.

1: least congestion and

5: most congested path

Congestion level >= 4: bad quality (not acceptable)

Congestion level < 4: good quality (acceptable)

Values inside brackets [] also represent number of hops.

1->5: increasing number of hops

1: least number of hops to the destination

5: most number of hops to the destination.

Time for the path in the history table: 5 seconds

Color convention:

Green: accepted calls

Red: rejected calls

Pink: path in history table

The figure 4.1 shows the ten paths with congestion levels are as follows

1	[1 | 2 | 3| 4 | 5]
2	[1 | 2 | 3| 4 | 5]
3	[1 | 2 | 3| 4 | 5]
4	[1 | 2 | 3| 4 | 5]
5	[1 | 2 | 3| 4 | 5]
6	[1 | 2 | 3| 4 | 5]
7	[1 | 2 | 3| 4 | 5]
8	[1 | 2 | 3| 4 | 5]
9	[1 | 2 | 3| 4 | 5]
10	[1 | 2 | 3| 4 | 5]

Figure 4.1. Ten paths with congestion levels.

The flow of algorithm is as follows:

There are four (A1, A2, A3, A4) algorithms A1, A2, A3 and A4.

Figure 4.2 shows the flow of the above four different algorithms including the one proposed in this thesis.

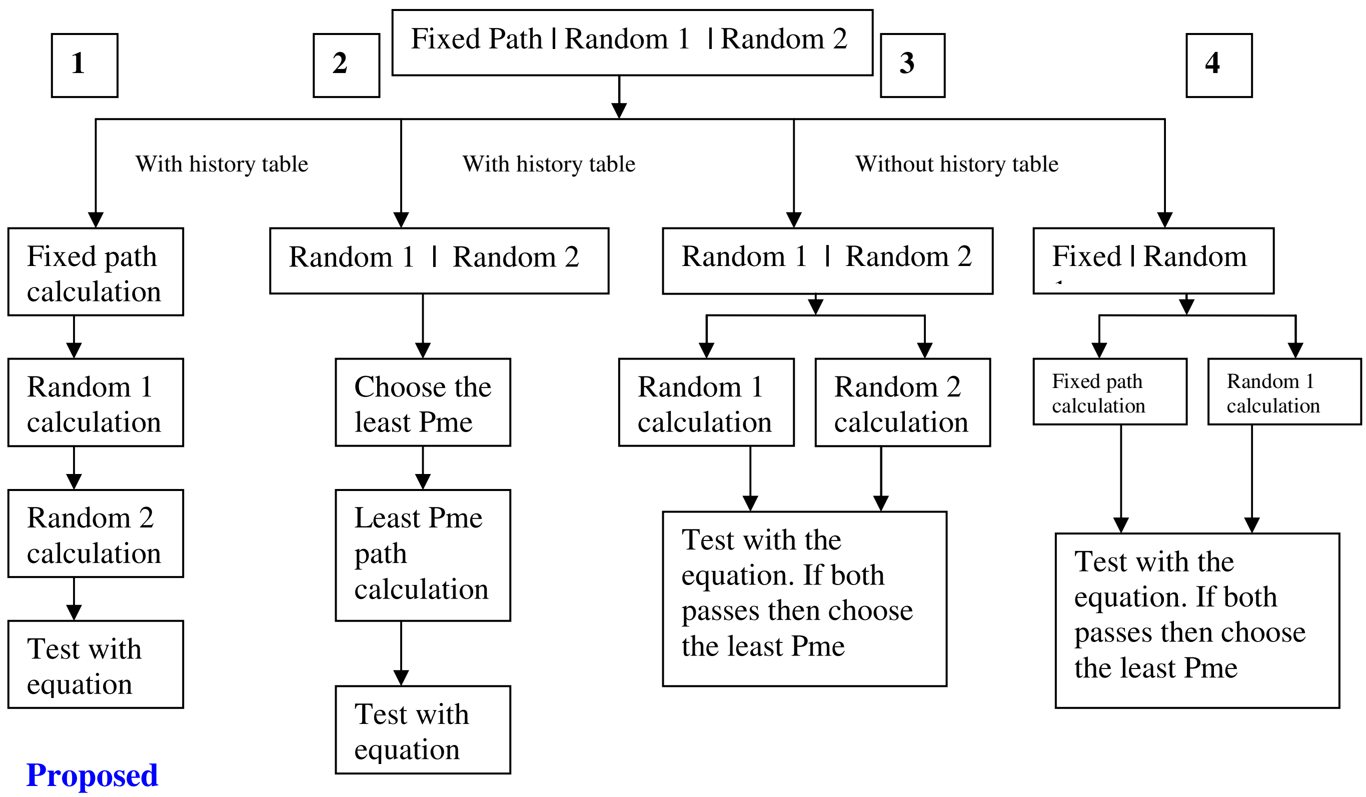

Figure 4.2. Flow of different algorithms.

The following abbreviations are used in the formulae

Th: Acceptance packet loss threshold.

Tpr: Length of probe

Rpr: Probe rate (bps)

Lpr: Probe packet size (b)

Npr: Number of probe packets per probing period

P_{me} : Measured value of congestion level

Probability that a call is accepted is given by [26, 31, 34],

$$P_{me} + Z_R \sqrt{\frac{P_{me}(1-P_{me})}{N_{pr}}} \quad (4.1)$$

where Zr=1-(1-R/2)

N_{pr} = probe packets / probing period [26, 31]

$$N_{pr} = \frac{R_{pr}.S_{pr}}{L_{pr}} \quad (4.2)$$

where, S_{pr} = probe size

L_{pr}= probe length

R_{pr}= probe rate

$N_{min} = N_{pr} - [Pme.Npr]$

where P_{me} = measured probe loss probability

To pass our admission threshold test,

$$P_{me} + Z_R \sqrt{\frac{P_{me}(1-P_{me})}{N_{pr}}} < \text{Th}$$

where Th= Defined threshold

Experimental setup:

Pme: one of the random numbers from each path.

Zr = 1-1(1-R/2)= 0.475

Where, R = required confidence level= 95%

Lpr= 64 bytes

Rpr= 100 kbps

Tpr= 2 s

Npr= Rpr.Tpr/Lpr =100* 1000 * 2 / 64 * 8= 390.625

Path in history table: path 8

To simulate the network, a random number generator is used to find the paths.

Obtain three paths:

One fixed path : path 1

Random path 1: path 5

Random path 2: path 8

Another random number generator is used to obtain the associated congestion level of the selected paths.

Path 1: level 2

Path 5: level 4

Path 7: level 3

CASE I: When the chosen path is not in the history table.

Call Number: 1

Pme + Zr sqrt(Pme((1-Pme))/Npr) <= Th

Th : 4.00

Path 1

2 + 0.475 sqrt(3(|1-2|))/390= 2.04

Path 5

4 + 0.475 sqrt(3(|1-4|))/390= 4.041

Path 7

3 + 0.475 sqrt(3(|1-3|))/390= 3.058

Performance of four algorithms:

Algorithm1 (proposed): Call will be accepted because of fixed path and random path 2; however, it will choose the least congested path.

Algorithm 2: Call will be accepted because of random path 2; however, Algorithm 1 fares better since the congestion level of the fixed path is less than random path 2.

Algorithm 3: Call will be accepted because of random path 2; however, Algorithm 1 fares better since the congestion level of fixed path is less than random path 2.

Algorithm 4: Call will be accepted because of fixed path.

Figure 4.3 shows that all calls are being accepted for all algorithms.

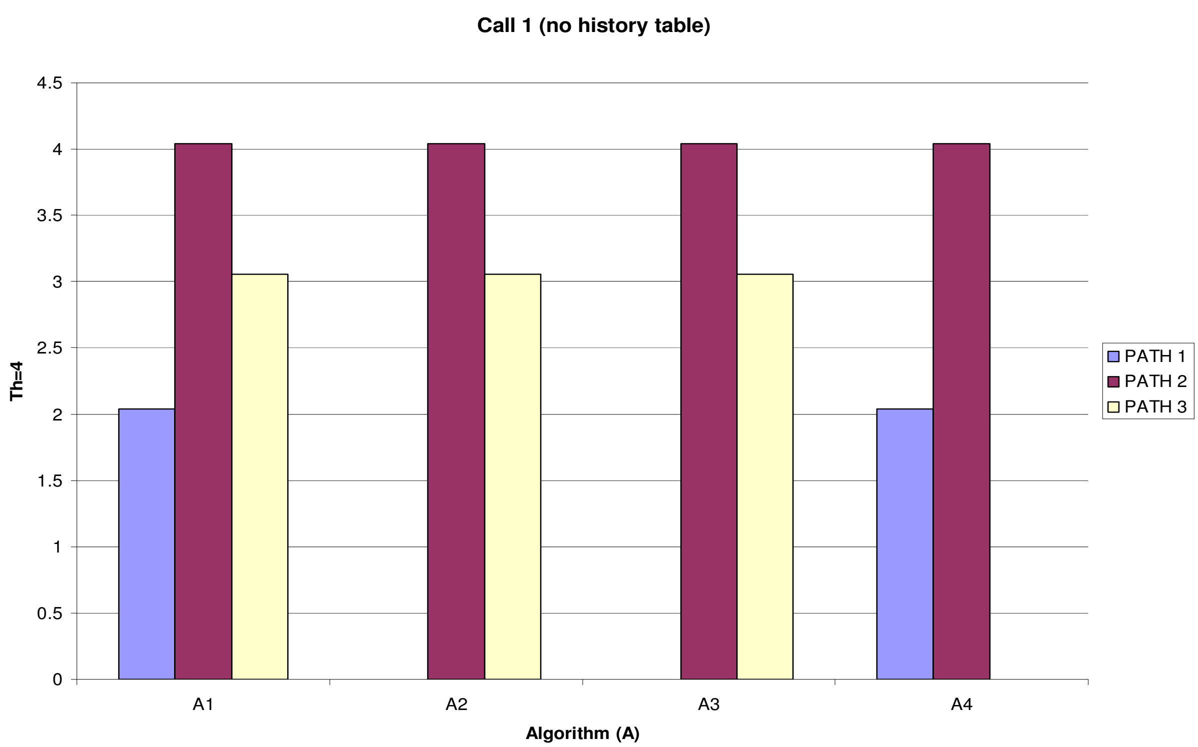

Figure 4.3 All calls accepted.

CASE II: When the chosen path is in the history table.

Path in history table: path 8 (level 5)

To simulate the network, a random number generator is used to find the paths.

Obtain three paths.

One fixed path: path 1

Random path 1: path 5

Random path 2: path 8 (will not be selected for A1/2)

Random path 2: path 7

Another random number generator is used to obtain the associated congestion level of the selected paths.

Path 1: level 2

Path 5: level 4

Path 8: level 3 (will NOT be selected for A1/2)

Path 7: level 3 (for only A1/A2)

<u>Call Number: 2</u>

Pme + Zr sqroot(Pme((1-Pme))/Npr) <= Th

Th : 4.00

Path 1

2 + 0.475 sqrt(3(|1-2|))/390= 2.04

Path 5

4 + 0.475 sqrt(3(|1-4|))/390= 4.041

Path 8 (for A3/A4)

5 + 0.475 sqrt(3(|1-3|))/390= 5.058

Path 7 (for A1/A2)

3 + 0.475 sqrt(3(|1-3|))/390= 3.058

Performance of four algorithms:

Algorithm1 (proposed): Call will be accepted because of fixed path and random path 2; however, it will choose the least congested path.

Algorithm 2: Call will be accepted because of the random path 2; however, Algorithm 1 fares better as the congestion level of fixed path is less than random path 2.

Algorithm 3: Call will be dropped.

Algorithm 4: Call will be accepted because of fixed path.

Figure 4.4 shows that call in algorithm A3 is dropped whereas call is accepted in the other algorithms.

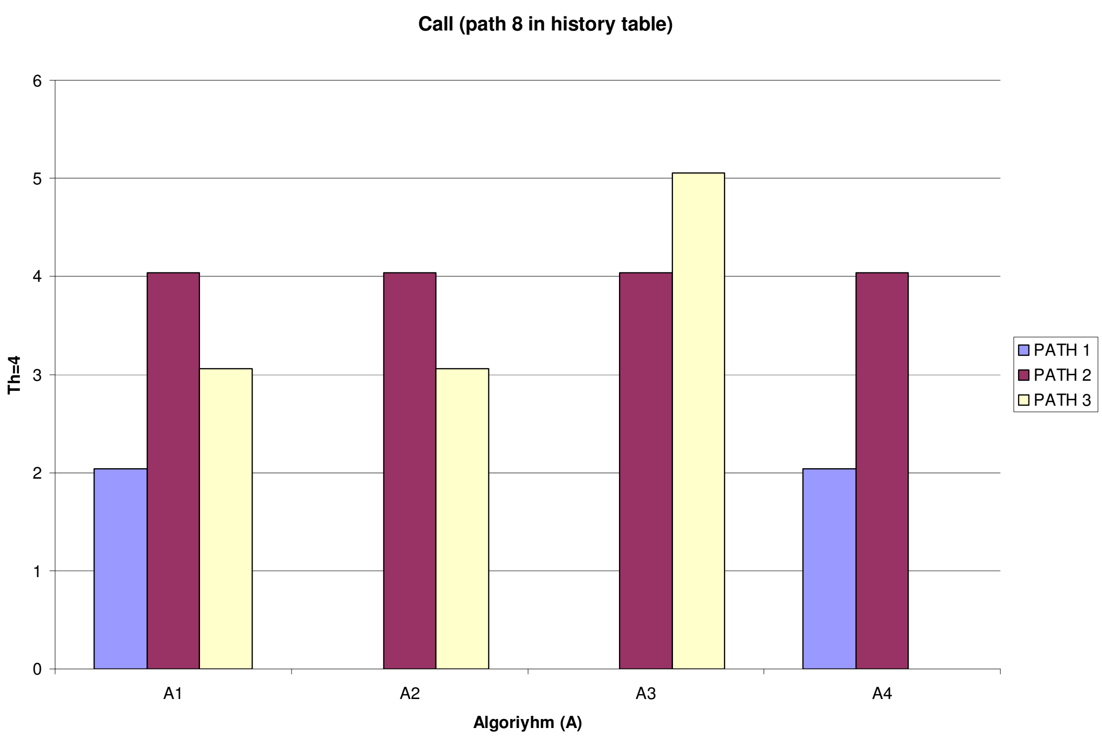

Figure 4.4 Call in A3 dropped.

CASE III: When the chosen path is in the history table.

Path in history table: path 8

To simulate the network, a random number generator is used to find the paths.

Obtain three paths.

One fixed path: path 1

Random path 1: path 5

Random path 2: path 8 (will not be selected for A1/2)

Random path 2: path 7

Another random number generator is used to obtain the associated congestion level of the.

Path 1: level 2

Path 5: level 4

Path 8: level 3 (will not be selected for A1/2)

Path 7: level 5 (for only A1/A2)

Call Number: 3

Pme + Zr sqroot(Pme((1-Pme))/Npr) <= Th

Th : 4.00

Path 1

2 + 0.475 sqrt(3(|1-2|))/390= 2.04

Path 5

4 + 0.475 sqrt(3(|1-4|))/390= 4.041

Path 8 (for A2/A3)

3 + 0.475 sqrt(3(|1-3|))/390= 3.058

Path 7 (for A1/A2)

5+ 0.475 sqrt(3(|1-3|))/390= 5.058

Performance of four algorithms:

Algorithm1 (proposed): Call will be accepted because of fixed path .

Algorithm 2: Call will be dropped.

Algorithm 3: Call will be dropped.

Algorithm 4: Call will be accepted because of fixed path .

Figure 4.5 shows that call in algorithms A2 and A3 is dropped and rest are accepted.

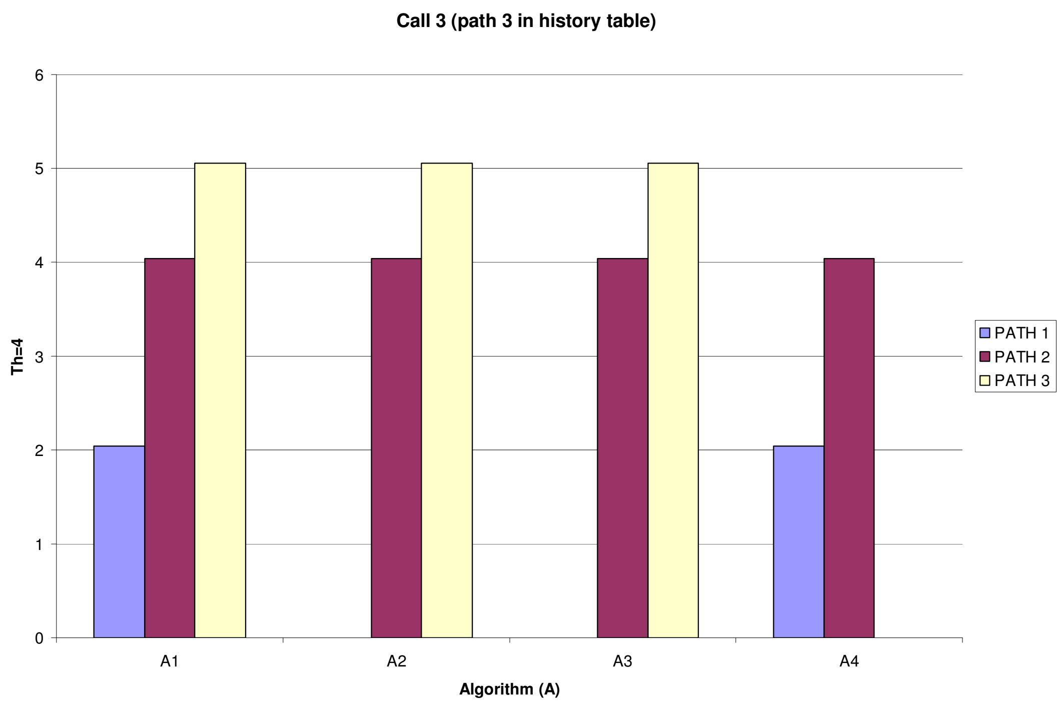

Figure 4.5 Call in A2 and A3 is dropped

CASE IV: When the chosen path is in the history table.

Path in history table: Fixed path

To simulate the network, a random number generator is used to find the paths.

Obtain three paths.

One fixed path: path 1 (for A1/A4)

Random path 1: path 5

Random path 2: path 8 (NOT for A4)

Another random number generator is used to obtain the associated congestion level of the.

Path 1: level 2 (for A1/A4)

Path 5: level 4

Path 8: level 3 (not for A4)

Call Number: 4

Pme + Zr sqroot(Pme((1-Pme))/Npr) <= Th

Th : 4.00

Path 1

4 + 0.475 sqrt(3(|1-2|))/390= 4.04

Path 5

4 + 0.475 sqrt(3(|1-4|))/390= 4.041

Path 8 (for A2/A3)

3 + 0.475 sqrt(3(|1-3|))/390= 3.058

Performance of four algorithms:

Algorithm1 (proposed): Call will be accepted because of the random path 2.

Algorithm 2: Call will be accepted because of the random path 2.

Algorithm 3: Call will be accepted because of the random path 2.

Algorithm 4: Call will be dropped.

Figure 4.6 shows that the call in A4 is dropped.

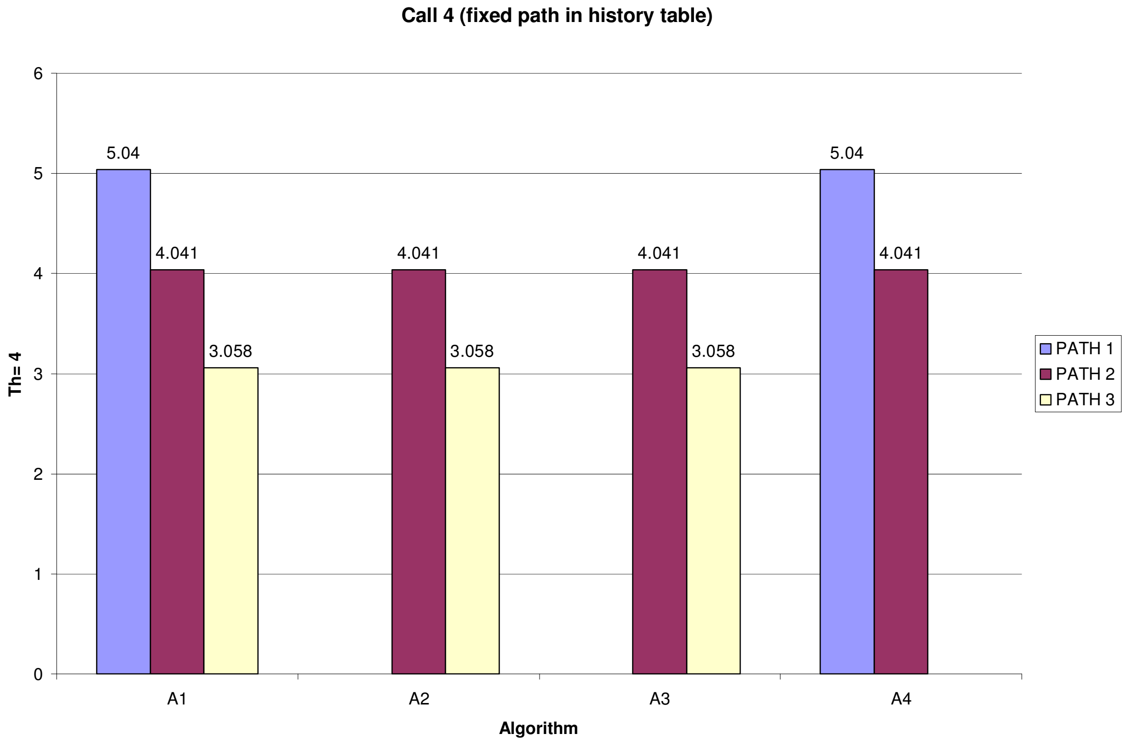

Figure 4.6 Call in A4 is dropped.

4.1 Mathematical analysis II and results (probe packets)

Here, we will observe the effect of the probe packets on the network using the proposed algorithm.

This system resembles a M/D/1 process [11,12,13,14,15,16,17,20] requiring constant service time and random arrival of customers.

With constant service time, the point of completion is determined precisely immediately when the service has started. Any service begun at the first will have finished by the second point. Any service which starts during the interval will be unfinished at the second point. The arrival of jobs to the system in the interval can be obtained by dividing the intervals into the times between arrivals:

S = service time

n = number of calls arriving during interval

r_1= start of the interval, the remainder of the current inter-arrival interval

t_2= arrival interval between first and second call

t_3= arrival interval between second and third calls

t_n = arrival interval between $(n-1)^{th}$ and n^{th} customer.

v = portion of arrival interval (current) at the end of the interval which has elapsed at that time.

$$\therefore s = r_1 + t_2 + t_3 + \ldots\ldots\ldots\ldots + t_n + v \tag{4.3}$$

The probability distribution of "n" intervals taking a time 'u' is [17, 19,20]:

$$f(u) = \lambda^n \frac{u^{n-1} e^{-\lambda\mu}}{(n-1)!} \tag{4.4}$$

The probability that an arrival interval will exceed "v" is,

$$\int_{v}^{\infty} \lambda e^{-\lambda t}\, dt \;=\; e^{-\lambda v} \tag{4.5}$$

Probability that "n" arrivals occur in s= u + v is [17,19,20,21]

$$\text{a(n)} = \int_{u=0}^{u=s} f(u) e^{-\lambda v}\, du$$

$$\because \text{v = s-u}$$

$$\text{a (n)} = \int_{0}^{s} f(u) e^{-\lambda(s-u)}\, du$$

$$= \int_{0}^{s} \lambda^{n} \frac{u^{n-1}}{(n-1)!} e^{-\lambda u} e^{-\lambda(s-u)} du$$

$$= \frac{\lambda^{n} e^{-\lambda s}}{(n-1)!} \int_{\partial}^{s} u^{\,n-1} du \;=\; \frac{(\lambda s)^{n}}{n!} e^{-\lambda s} \tag{4.6}$$

Where "n" is the number of arrivals, in the service time "s", it has a Poisson distribution with a means λs.

$\lambda s =$ Average number of arrivals in a service time "s".

Let,

P_t(n) = "n" customers probability in the sys at time "t"

a (j) = "j" customers probability arriving in a service time "s"

$$\text{a (j)} = \frac{(\lambda s)^{n}}{n!} e^{-\lambda s} \;\ldots\ldots\ldots\ldots\ldots\ldots\ldots\text{from equation (4.2)}$$

Therefore,

$$P_{t+s}(0) = P_t(0)\, a(0) + P_t(1)\, a(0)$$

which states that at the start of the interval there would have been no jobs in the systems, none arrive during the interval , thus leaving none at the end.

Similarly, other probabilities can be written as

$$P_{(t+s)}(1) = P_t(0)\, a(1) + P_t(1)\, a(1) + P_t(2)\, a(0)$$

$$P_{(t+s)}(j) = P_t(0)\, a(j) + P_t(1)\, a(j) + P_t(2)\, a(j-1) + P_t(3)\, a(j-2) + \ldots\ldots\ldots + P_t(n)\, a(j-r+1) + \ldots\ldots\ldots + P(j+1)\, a(0) \qquad (4.7)$$

where $j = 2, 3, \ldots, \infty$

The above equation 4.3 will give the probabilities at any time given the initial state of the system. The steady state solution can be found dropping the suffix "t" in the equation (4.3) and solving the resulting equations

$$P(0) = \{P(0) + P(1)\}\, a(0)$$

$$P(j) = (P(0) + P(1))\, a(j) + \sum_{r=2}^{r=j+1} P(r)\, a(j-r+1) \qquad (4.8)$$

The probability in terms of P (0) is

$$P(1) = [P(1)(1-a(1)) - P(0)\, a(1)] / a(0) \ldots\ldots\ldots \text{etc.}$$

Consider the function

$$F(x) = \sum_{j=0}^{\infty} x^j P(j) \qquad (4.9)$$

By using equation 4.4 & multiplying the equation with P(j) on the left hand side by x^j for all values of j from 0 to ∞ and summing all the equations, then

$$\sum_{j=0}^{\infty} x^j P(j) = P(0) + P(1)\, a(0) + \sum_{j=1}^{\infty} x^j [P(0) + P(1)\, a(j) + \sum_{j=2}^{r=j+1} P(r) a(j-r+1)]$$

$$(P(0) + P(1) \sum_{J=0}^{\infty} x^{j} a(j) + \sum_{j=2}^{\infty} p(j) \sum_{k=0}^{\infty} x^{k+j-1} a(k)$$

$$\because \sum_{j=0}^{\infty} x^{j} a(j) = \sum x_{j} \frac{\rho}{j!} e^{-\rho} = e^{\rho x - \rho}$$

where $\rho = \lambda s$

The required equation reduces to

$$F(x) = [P(0) + P(1)] e^{p(x-1)} + \sum_{j=2}^{\infty} p(j) x^{j-1} e^{\rho x - \rho}$$

$$F(x) = e^{\rho (1-x)} = P(0) + P(1) + \sum_{J=2}^{\infty} x^{j} P(J)/x$$

$$= P(0) + P(1) + [F(x) – x P(1) – P(0)]/x$$

$$F(x) [e^{\rho (1-x)} – 1/x] = P(0) (1-1/x)$$

$$F(x) = \frac{P(0)(x-1)}{xe^{\rho(1-x)} - 1} \tag{4.10}$$

$$\text{when } x = 1,\ F(1) = \sum p(j) = 1$$

$$R.H.S = P(0) \lim_{x->1} [\frac{(x-1)}{xe^{\rho(1-x)} - 1}]$$

When x =1, both the numerator and the denominator are zero.

$$\lim_{x->1} \frac{(x-1)}{xe^{\rho(1-x)} - 1} = \lim_{x->1} \frac{1}{(1-\rho x)e^{\rho(1-x)}} = \frac{1}{1-\rho}$$

$$\text{hence, } P(0) \frac{1}{1-\rho} = 1$$

$$P(0) = 1 - \rho \tag{4.11}$$

Probabilities can be found by differentiating F(x) and evaluating at x=0

$$\frac{d^n}{dx^n}F(x) = \frac{1}{n!} + \frac{x}{(n+1)!}P(n+1) + \text{ higher powers of x}$$

$$\frac{d^n}{dx^n}F(x)\,|_{x=0} = \frac{1}{n!P(n)} \tag{4.12}$$

The average number of calls in the system = $\sum_{j=0}^{\infty} jP(1)$ [14,16,19,20]

$$\frac{d}{dx}F(x)\,|_{x=1} = \frac{d}{dx}[\sum_{j=0}^{\infty} x^j P(j)]\,|_{x=1}$$

$$= \sum_{j=1}^{\infty} jx^{j-1}P(j)\,|_{x=1}$$

Therefore, average number of calls in the system = $\frac{d}{dx}F(x)$ [19,20,21]

$$= \frac{(1-\rho)[xe^{\rho(1-x)} - 1 - (x-1)(1-\rho x)e^{\rho(1-x)}]}{(xe^{\rho(1-x)} - 1)^2}$$

at x=1

Since the numerator and denominator are both zero, so the ratio of the first non-zero differential must be used to obtain the value, giving

$$= (1-\rho)\frac{\rho(2-\rho)}{2(1-\rho)^2} = \frac{\rho(2-\rho)}{2(1-\rho)} \tag{4.13}$$

Figure 4.7 shows the relation of average number of calls in the system with respect to the traffic intensity. The traffic intensity is varied from 0.1 to 0.9 and corresponding number of calls are found out in the system by the equation 4.9. Values can be seen in Appendix A. Here, as the traffic intensity increases, the number of calls also increases. This also depends on the arrival rate of the probe packets as the service time remains constant. Number of calls increases sharply as the traffic intensity reaches 70 percent.

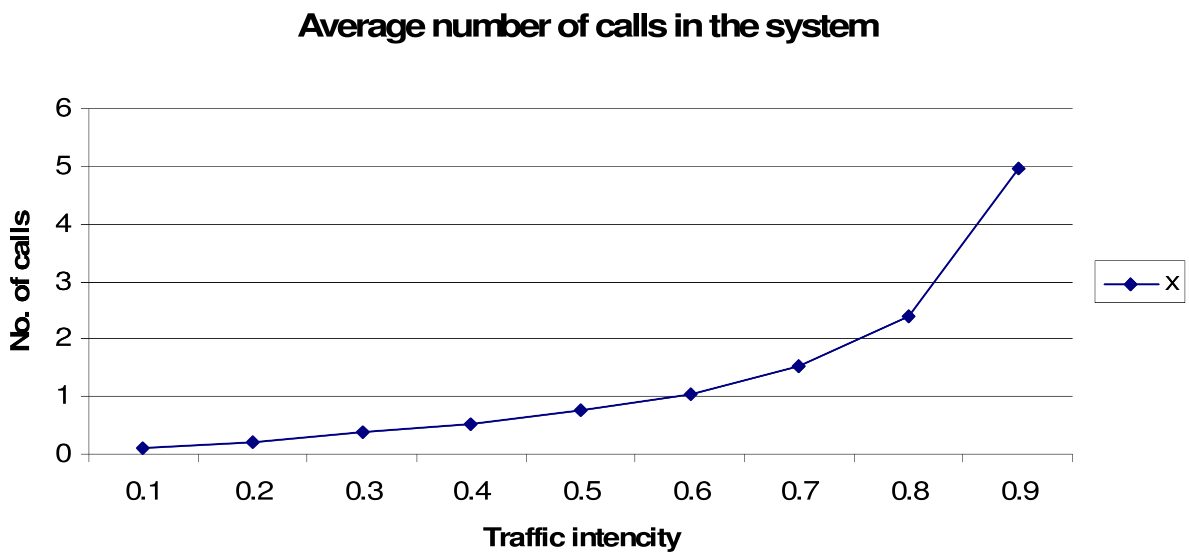

Figure 4.7 Average number of calls in the system with respect to the traffic intensity.

The average number of calls in the queue = $\sum_{j=2}^{\infty}(j-1)(p(j))$ [15,16,19,20,21,22]

$$=\sum_{j=1}^{\infty} j(P(j)) - \sum_{j=1}^{\infty} P(j)$$

$$=\frac{\rho(2-\rho)}{2(1-\rho)} - (1-P(0))$$

$$=\frac{\rho(2-\rho)}{2(1-\rho)} - \rho$$

$$=\frac{\rho^2}{2(1-\rho)} \tag{4.14}$$

Figure 4.8 shows the relationship of the average number of calls in the queue with respect to traffic intensity. The traffic intensity is varied from 0.1 to 0.9 and corresponding number of calls are found out in the system by the equation 4.10. Values can be seen in Appendix A.

Here, as the traffic intensity increases, the number of calls also increases. This also depends on the arrival rate of the probe packets since the service time is constant. Number of calls increases rapidly as the traffic intensity reaches 70%.

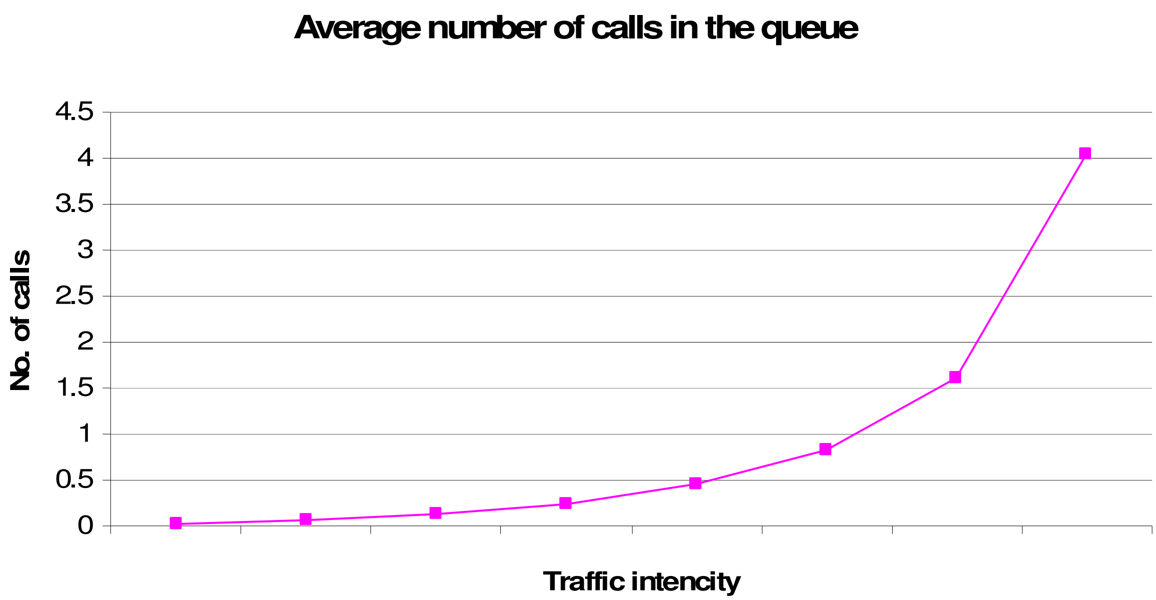

Figure 4.8 Average number of calls in the queue with respect to the traffic intensity.

Since the service time is constant and the packets could arrive at any time during a service with equal probability, the average of the remaining of the service in progress is $\bar{r}$,where

$$\bar{r} = \int_0^s r f(r) dr$$

Where f(r) is the distribution function of the remainder and f(r)= $\frac{1}{s}$

$$\bar{r} = \int_0^s \frac{r}{s} dr = \frac{s}{2} \qquad (4.15)$$

$\therefore$ Average waiting time = $\bar{r} + (j-1)s$

= (j-½) s

$\therefore$ Average waiting time for all the packets

$$= \sum_{j=1}^{\infty} P(j)(j - ½)\ \mathrm{s} \quad [15,16,19,20,21,22]$$

$$= \mathrm{s}\ [\ \sum_{j=1}^{\infty} jP(j) - \frac{1}{2}[1 - P(0)]$$

$$= \mathrm{s}\ [\frac{\rho(2-\rho)}{2(1-\rho)} - \frac{1}{2}\rho]$$

$$= \mathrm{s}\ [\frac{\rho}{2(1-\rho)}] \qquad (4.16)$$

Figure 4.9 shows the average waiting time for all packets with respect to the traffic intensity. The traffic intensity is varied from 0.1 to 0.9 and the service time "s" is varied from 1 seconds to 9 seconds and corresponding waiting time is calculated from the equation number 4.12. Values can be seen in Appendix A. Here, as the traffic intensity increases, the waiting time of the packets also increases. Also, the waiting time depends on the arrival rate of the probe packets. Waiting time increases very sharply after the traffic intensity reaches 80 percent.

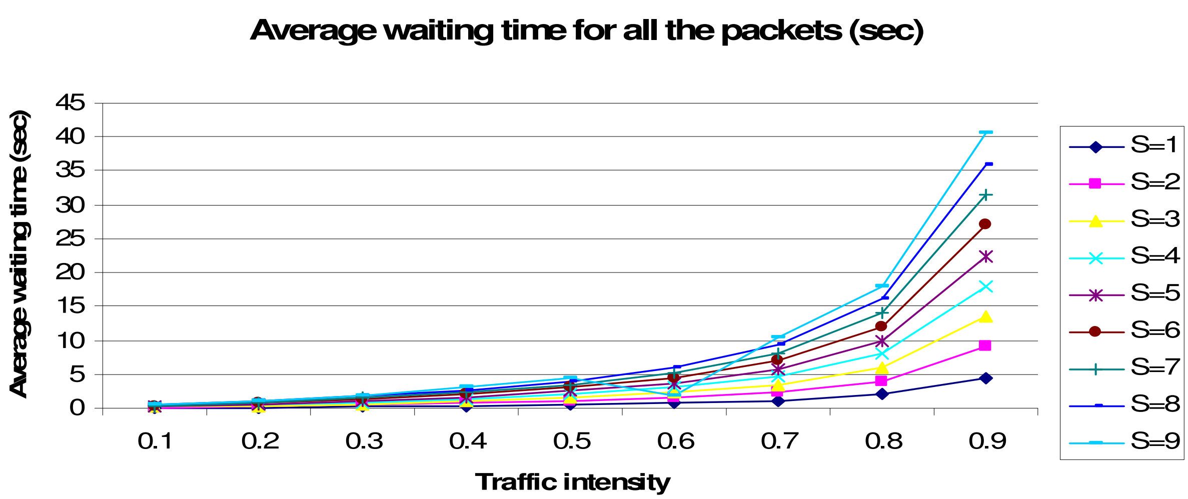

Figure 4.9 Average waiting time for all the packets with respect to the traffic intensity.

The chance a packet does not have to wait is

$$P\ (0) = 1 - \rho$$

The utilization of the server = 1- P (0)

$$=1/\ (-\lambda - \rho)\ = \rho$$

Since the arrivals are random, the busy period = $\bar{B}$ = [1- P (0)]/ [$\lambda P(0)$]

$$= \frac{S}{1-\rho} \quad [15,16,19,20,21,22]$$

Figure 4.10 shows that as the traffic intensity increases, the average busy time of the server also increases which states that the busy period depends on the arrival rate of the probe packets because of constant service time. Busy period increases sharply when the traffic intensity reaches 80%. The traffic intensity is varied from 0.1 to 0.9 and the service time "s" is varied from 1 second to 9 seconds and corresponding waiting time is calculated from the equation number 4.12. Values can be seen in Appendix A.

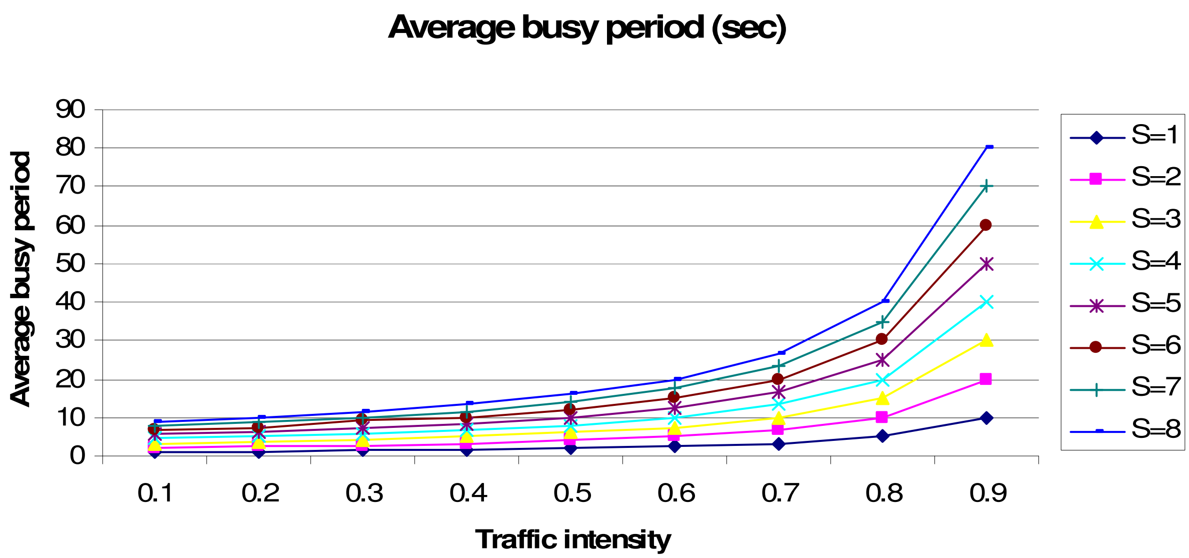

Figure 4.10 Average busy period with respect to traffic intensity

4.3 Mathematical analysis III and results (probe packets)

Here, we will observe the effect of the probe packets on the acceptance probability of the call in the algorithm, utilization and other parameters of the network [11,12,13,14,15,16,17,20].

Let,

P_a = Acceptance probability

R_{pr} = Rate of probe in bits/secs

S_{pr} = Size of probe pec/cet in bits.

N_{min} = Minimum no. of successful probe packets for admission control decision.

P_{loss} = packet (data) loss

U = final link utilization

L_{pr} = length (time) of the probe (seconds)

Th = threshold value of packet loss

P_{succ}= successful transmission probability of one probe packet.

This analysis will try to derive the probability that a new flow is accepted which in turn can be expressed as a function of the probability of successful probe transmissions. This can be obtained by Bernaulli's trial formula,

$$X_1(0, 1), X2(0, 1), X3(0, 1) \ldots X_n$$

Assumption

(1) X_1, X_2, X_3 are independent

(2) P = probability of success

(3) n = number

$$P[X=x] = p^x. (1-p)^{1-x}, \text{ where,}$$

$$x = 0, 1$$

Now,

$$\# \text{ Of success} = X = X_1 + X_2 + X_3 + \ldots\ldots\ldots\ldots..$$

$$=0, 1, 2\ldots n$$

$$P[X = x] = {}^{n}C_{x} (p)^{x} (1-x)^{n-x}$$

which can also be written as [13,31]

$$P_a = \sum_{l=N_{\min}}^{N_{pr}} \binom{N_{pr}}{i} P_{succ}{}^{i} (1 - P_{succ})^{N_{pr} - I} \qquad (4.17)$$

Figure 4.11 shows the relationship between the acceptance probabilities of one call with respect to the probability of success of one probe packet in the network. , minimum number of probe packets is taken as 10 and maximum is taken as 15 and incremented by 1, probability of success is varied from 0.1 to 0.9 and then the corresponding acceptance probability is found out from the equation 4.13. Values can be seen in Appendix A. Here, as the probability of success of one probe packet increases with the acceptance probability also increases.

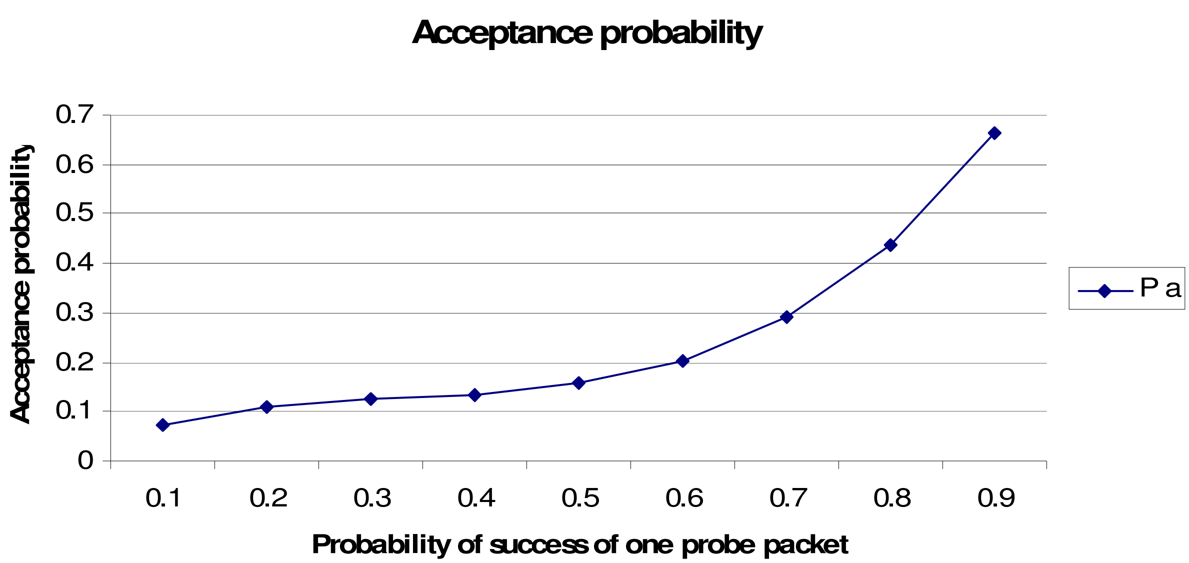

Figure 4.11 Acceptance probability v/s probability of success of one probe packet

The probability that a call is accepted is given by, [21,26,31]

$$P_{me} + Z_R \sqrt{\frac{P_{me}(1-P_{me})}{N_{pr}}} \qquad (4.18)$$

where, Zr=1-(1-R/2)

Figure 4.12 shows the relationship between the probability that a call is accepted with the number of probe packets sent in the network. “R” is taken as 0.95, measured congestion in the network “Pme” is varied from 0.01 to 0.09 , number of probe packets are also varied from 100 to 900 and the corresponding acceptance probability is found out from the equation 4.14. Values can be seen in Appendix A.

Here, for smaller values of Pme, the probability is not effected much but as the Pme increases, the probability seems to decrease with the increase in the Pme.

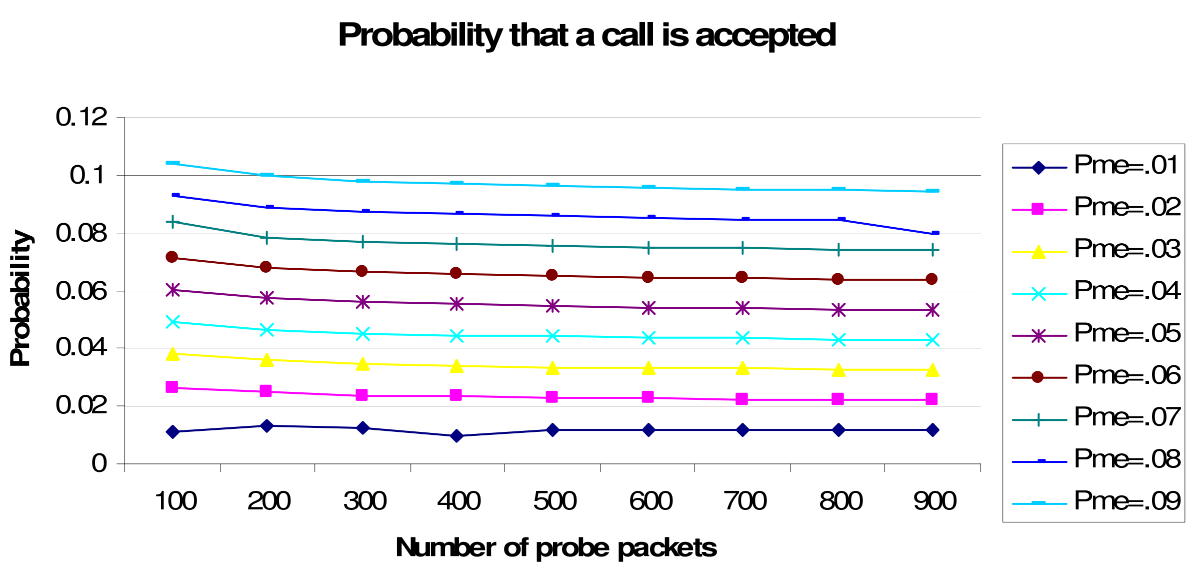

Figure 4.12 Probability that a call is accepted v/s number of probe packets

N_{pr} = probe packets / probing period [21,26,31]

$$N_{pr} = \frac{R_{pr} . S_{pr}}{L_{pr}} \tag{4.19}$$

where, S_{pr} = probe size

L_{pr}= probe length

R_{pr}= probe rate

$$N_{min} = N_{pr} - [Pme.Npr]$$

where P_{me} = measured probe loss probability

Figure 4.13 shows the relationship between the number of probe packets to the length of the probe packets. Lpr is varied from 50 to 500 and the corresponding number of probe packets are found out from the equation 4.15. Values can be seen in Appendix A.

Here, as the number of probe packet increases, the length of the probe increases. This states that for having more number of probe packets we need to decrease the probe length.

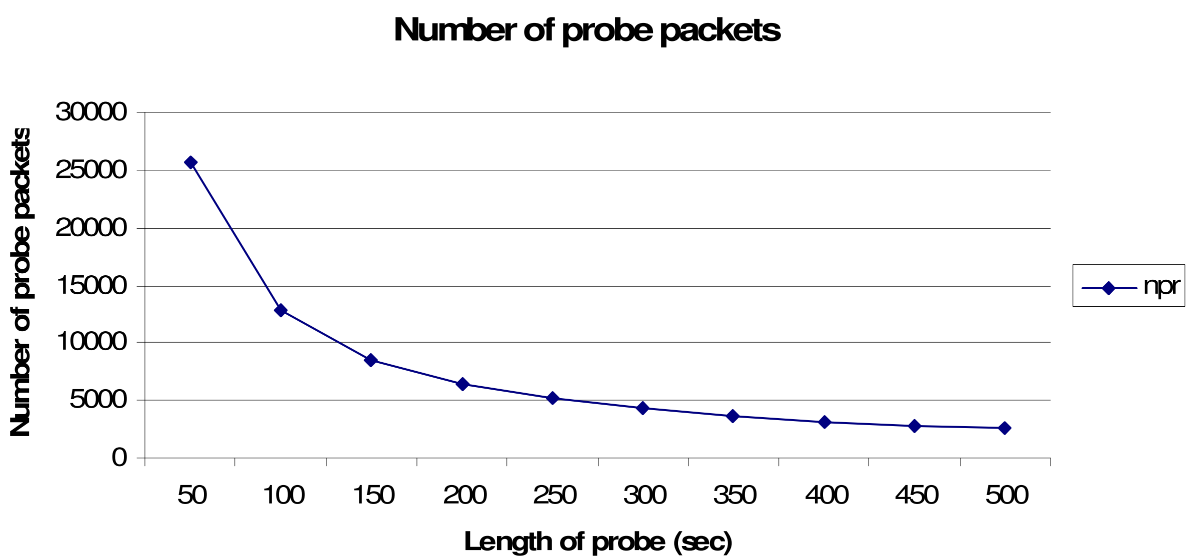

Figure 4.13 Number of probe packets with respect to length of the probe

Now let us see the probability of success of one probe packet. [15,16,19,20,21,22,31]

$$\text{Let T = inter –arrival time} = \frac{1}{R_{pr}}$$

ρ = utilization of the queue

$$\therefore P_{succ} = (1- \rho) + \rho \ F_{rb}(T)$$

$$= (1- \rho) + \rho \int_0^T f_{rb}(t)dt \qquad (4.20)$$

F_{bp} (t) can be found out from the Borel distribution which is given by [13,21,22,23],

$$F_{bp}(t) = \sum_{j=1}^{n} \frac{(jp)^{j-1}}{j!} e^{-j\rho}$$

$$\text{Where, } n = \frac{t}{E(s)}$$

$\therefore$ Remaining busy period can be obtained by

$$F_{rb}(t) = \frac{1-F_{bp}(t)}{E_{bp}}$$

$$\text{Where } E_{bp} = \frac{s}{1-\rho}$$

Derivation of $E_{bp} = \dfrac{s}{1-\rho}$ is shown in a separate piece of paper.

where E(s) = average packet service time

$E_{(bp)}$ = average length of busy period

$$\therefore P_{succ} = (1-\rho) + \rho\int_0^T f_{rb}(t)dt$$

$$= (1-\ \rho) + \rho\int_0^T [\frac{1-F_{bp}(t)}{E_{(bp)}}]dt$$

$$= (1-\ \rho) + \rho\int_0^T \frac{[1-F_{bp}(t)]}{s/_{1-\rho}}dt$$

$$P_{succ} = (1-\rho) + \rho\int_0^T \frac{(1-\rho)(1-F_{bp}(t))}{s}dt \quad [28,31]$$

$$\int_0^T f_{rb}(t)dt = \sum_{i=0}^{T-1} f_{rb}(i)\ [28,31] \tag{4.21}$$

$\therefore$ Putting (4.17) in (4.16),

$$P_{succ} = (1-\ \rho) + \rho\int_0^T f_{rb}(t)dt$$

$$= (1-\rho) + \rho\sum_{t=0}^{T-1} f_{rb}(i)$$

$$P_{succ} = (1-\rho) + \rho\sum_{t=0}^{T-1}\sum_{j=1}^{n}\frac{(j\rho)^{j-1}}{j!}e^{-j\rho}$$

where n = $\dfrac{t}{E(s)}$

Link utilization:-

We use the birth death Morkov chain for modeling the no. of accepted connections.

Assume the following

(1) Poisson arrival process

(2) $\lambda =$ average flow /sec

(3) $1/_{\mu} = average$ holding time

(4) Acceptance decisions are independent

$\therefore$ Birth - death coefficients are as follows:-

$$\lambda_j = \lambda P_a(j)$$

$$\mu_j = J\mu$$

where j =number of connections

P_a (j) = acceptance probability of ‘J ‘accepted connections

Let S_j = steady state probability of birth death model.

For our model in M/D/1 system [11,12,13,14,15,16,17,20]

S_J for ‘j ‘jobs in system is given by

For j=0

$$S_j = 1-\rho$$

For j=1

$$S_j = (1-\rho)(e^{\rho} - 1) \tag{4.22}$$

Figure 4.14 shows the relationship between the steady state probabilities for one call with respect to the traffic intensity. Traffic intensity is varied from 0.1 to 0.9 and the corresponding steady state probability of one call is found out from the equation 4.18. Values can be seen in Appendix A.

Here, as that the probability increases for lower traffic intensity but starts decreasing for the higher traffic intensity.

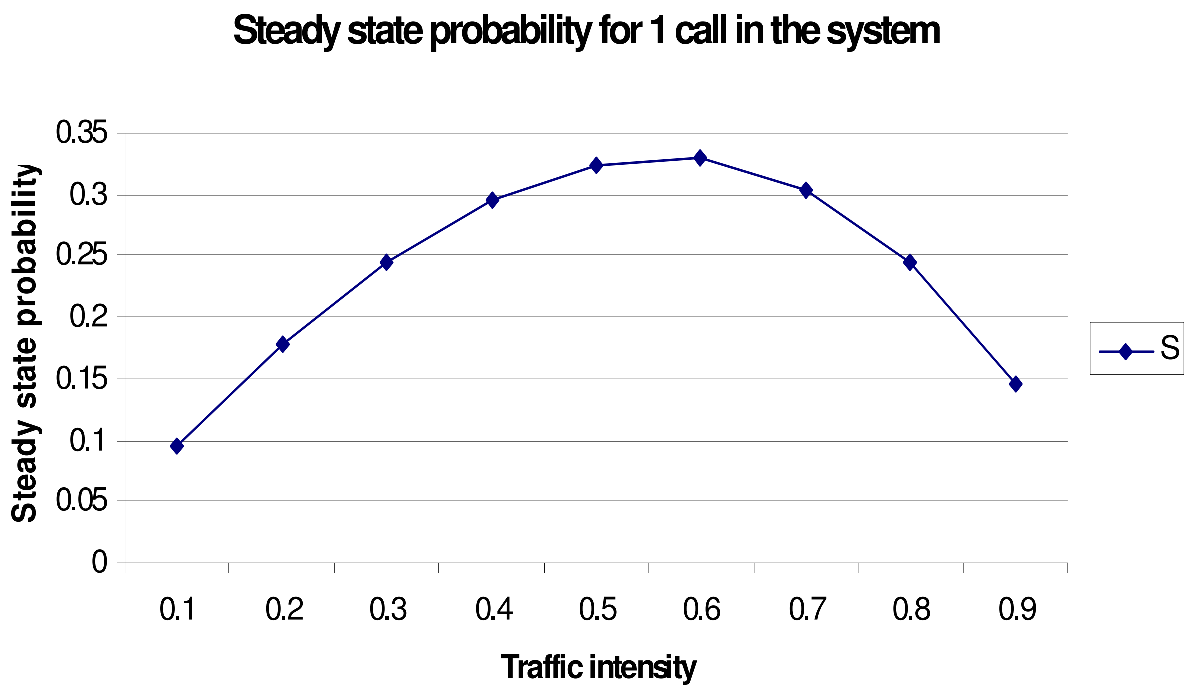

Figure 4.14 Steady state probability for one call vs. traffic intensity [21,22]

For j>= 2

$$S_j = (1-\rho)\sum_{n=0}^{j} \frac{(-1)^{j-n}(n\rho)^{j-n-1}(n\rho + j - n)e^{n\rho}}{(j-n)!} \tag{4.23}$$

Figure 4.15 shows the relation between the steady state probability for more than or equal to 2 calls with respect to the traffic intensity. Traffic intensity is varied from 0.1 to 0.9 and the corresponding steady state probability of one call is found out from the equation 4.18. Values can be seen in Appendix A.

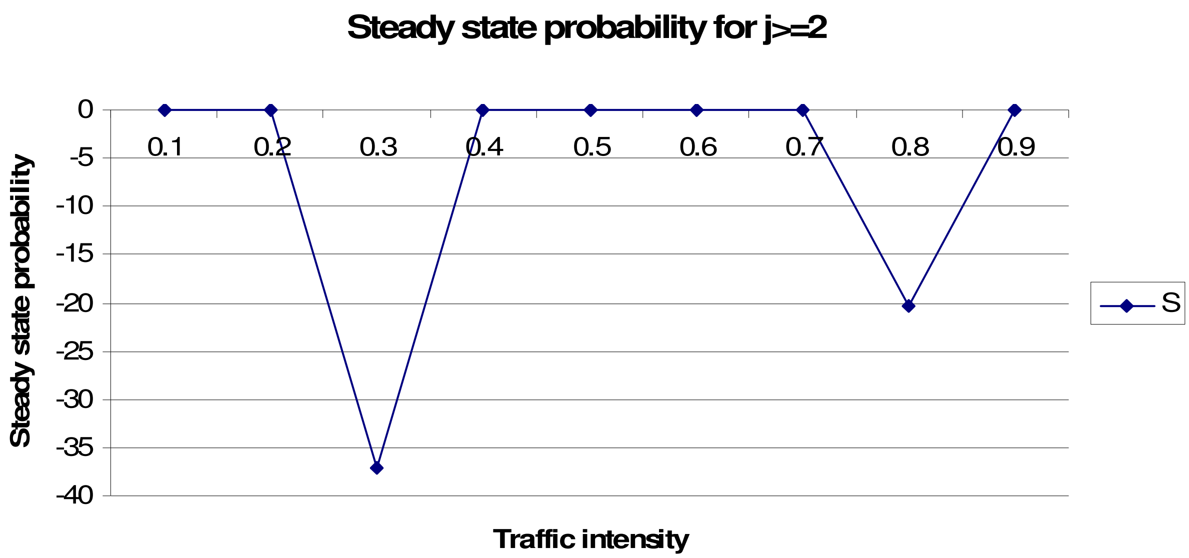

Figure 4.15 Steady-state probability for more than or equal to two calls vs. traffic intensity

Let C = link capacity

α = Activity factor of all the sources available i.e. % of time in 'on' state

$$\therefore utilization = \frac{s_j(j)(\alpha)}{c} R_{pr} \quad [28,31]$$

For 'j' number of connections [21,22]

$$U = \sum_j \frac{(1-\rho)\sum_{n=0}^{j}(-1)^{j-n}..(n\rho)^{j-n-1}(n\rho + j - n)e^{n\rho}.J.\alpha.R_{pr}}{c} \tag{4.24}$$

Figure 4.16 shows the relationship of Utilization with respect to the traffic intensity. Traffic intensity is varied from 0.1 to 0.9, number of accepted calls is taken as 10 activity factor is taken as 0.95 and probe rate at 64Kbps and the corresponding utilization is found out from the equation 4.20. Values can be seen in Appendix A.

Here, utilization increases as the traffic intensity increases depending on the arrival rate of the probe packets. Also, utilization increases very sharply when the traffic intensity reaches 70 percent.

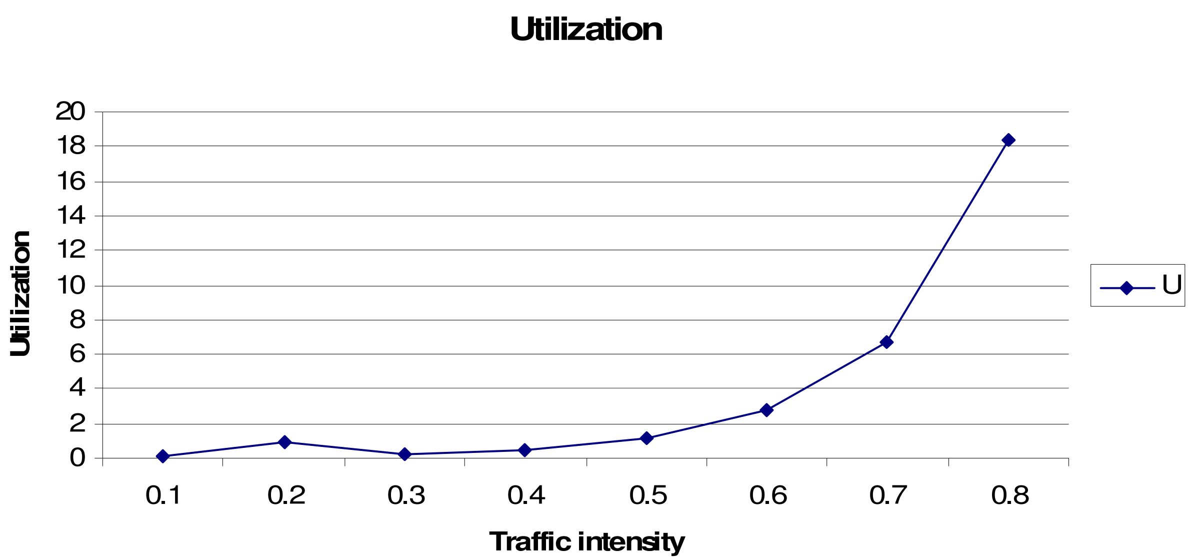

Figure 4.16 Utilization vs. traffic intensity

Borel distribution

In a queuing process with random arrivals at a rate 'q' per unit time and constant service time β, the no. of units served during a busy period follows a Borel distribution [22,23]

$$P_n = \frac{e^{-n\beta q}(n\beta q)^{n-1}}{n!} \quad (n = 1, 2, 3\ldots) \qquad (4.25)$$

For the table 4.12, service time is taken as 20 seconds, arrival rate "q" as 50, "r"=5, "n"=15 and the corresponding number of calls are found out by the equation 4.21

TABLE 4.12

NUMBER OF CALLS WITH BOREL'S DISTRIBUTION

q	S
.1	$5.07*10^{57}$
.2	$4.86*10^{47}$
.3	$2.62*10^{36}$
.4	$4.36*10^{28}$
.5	$3.8*10^{12}$
.6	$2.2*10^{3}$
.7	$1.03*10^{1}$
.8	0

The above table shows relationship between the numbers of calls served with the Borel's distribution with respect to the traffic intensity. Service time is taken as 20 seconds, arrival rate "q" as 50, "r"=5, "n"=15 and the corresponding number of calls are found out by the equation 4.21. Values can be seen in Appendix A.

Here, as the traffic intensity increases the number of calls served decreases. Therefore, the number of calls is inversely proportional to the arrival rate of the probe packets since the service time remains constant

This was further augmented /extended by Tanner who showed that the distribution of the number of units served in a busy period starting with an accumulation of 'r' units is [22,23]

$$P_{n,r} = \frac{e^{-n\beta q}(\beta q)^{n-r}\, r.n^{n-r-1}}{n-r!} \text{ (n=r,r+1,...)}$$

$$= \frac{e^{-n\beta q}(n\beta q)^{n-r}}{n-r!} * \frac{r}{n} \text{ (n=r,r+1.....)} \qquad (4.26)$$

The necessary and sufficient conditions for a busy period to contain just 'n 'units include the following:

(1) Precisely (n-r) units shall arrive within a time n β

(2) The pattern of these arrivals shall be admissible which means that at least one unit must arrive during a time r β ,at least two within (r+1) β ……..so that the server will remain continuously occupied for the whole of the time n β

For more server power we can model our sys in m/d/n [20,21,22]system. The methods which we used for the single-server queue with constant service can be used to derive the equations for multi-server queues with constant time.

Let,

P_t (j) = 'j' no. of packets probability in the system in time't'

a (i) = probability of 'I' packets arrive in an interval in the service time's'

From previous derivations, the chance that 'k 'no. of packets arrive in an interval equal to the service time's' is given by

$$a\ (k) = \frac{\rho^{-k}}{k!} e^{\ -\rho}$$

$$\text{where, } \rho = \frac{servicetime}{arrival \operatorname{int} erval} = \lambda s$$

Since we know that any packet being served at the start of the interval will have finished being served and left the system by the end of the interval. Any other packets will still be in the system at the end of the interval.

$\therefore$ The probabilities are as follows

$$P_{t+s}\ (0) = P_t\ (0)\ a\ (0) + P_t\ (1)\ a\ (0) + \ldots\ldots\ldots\ldots + P_t\ (n)\ a\ (0)$$

$$P_{t+s}(1) = [P_t\ (0) + P_t(1) + \ldots\ldots\ldots\ldots\ldots + Pt(n)]a(1) + P_t(n)\ a(0)$$

$$P_{t+s}(j) = P_t(0) + P_t(1) + \ldots . + P_t(n+j)a(0) \tag{4.27}$$

where j=1, 2, 3… ∞

Now he steady-state probabilities of dropping the time dependence,

$$P(0) = [\ P(0) + P(1) + \dots\dots\dots + P(n)]a(0)$$

$$P\ (j) = P\ (0) + P\ (1) + \dots\dots\dots P\ (n)]\ a\ (j) + \sum_{k=1}^{j} P(n+k)a(j-k) \tag{4.28}$$

$$\text{where } j=1,\ 2,\ 3\dots\dots\infty$$

If the probabilities P (0), P (1)…P (n-1) are known then the required probabilities can be calculated directly by the probability generating function

$$F(x) = \sum_{j=0}^{\infty} x^{j} P(j)$$

By multiplying eq. (2) by $x^{0}, x, x^{2}\dots\dots x^{j}$ and then summing all the resulting equations, given,

$$F(x) = [P\ (0) + P\ (1) + P\ (2) + \dots P\ (n)] + \sum_{k=1}^{\infty} P(n+k)s(k)$$

$$\text{Where } s\ (k) = \sum_{j=0}^{\infty} x^{j+k} a(j)$$

$$= x^{k} e^{-\rho(1-x)}$$

$$\therefore F(x) = P(0) + P(1) + \dots\dots\dots P(n)]e^{-\rho(1-x)} + \sum_{k=1}^{\infty} x^{k} e^{-\rho(1-x)} P(n+k)$$

$$= [P\ (0) + P\ (1) + \dots\dots + P\ (n)]\ e^{-\rho} + x^{-n} e^{-\rho(1-x)} \sum_{k=1}^{\infty} x^{n+k} P(n+k)$$

$$=[P(0)+P(1)+\dots\dots\dots\dots+P(n)]\, e^{-\rho(1-x)} + x^{-n} e^{-\rho(1-x)} \{F(x) - P(0) - xP(1) - \dots\dots\dots x^{n} P(n)\}$$

$$F(x)[1-x^{-n} e^{-\rho(1-x)}] = P(0)(1-x^{-n}) + P(1)(1-x^{-n+1}) + P(2)(1-x^{-n+2}) + \dots\dots + P(n-1)$$

$$X\ (1-x^{-1})\ [R.H.S\ (say)]$$

$$\therefore F(x) = R.H.S \big/ [1 - x^{-n} e^{-\rho(1-x)}]$$

$$= {(R.H.S)x^n} \Big/ {[x^n - e^{-\rho(1-x)}]} \tag{4.29}$$

Since,

F(x) is a finite number for any value of "x" that is less than 1 in absolute value. Any such value which is a root of the equation.

$$x^n - e^{\rho(1-x)} = 0 \tag{4.30}$$

which states that there are n such roots to the equation above which in turn gives n equations for the probabilities P(0),P(1),……………P(n-1)

Suppose, those roots are $x_1, x_2, \ldots\ldots x_n$, the equations will be as follows:-

$$P(0)(x_i^n-1) + P(1)(x_i^n-x) + \ldots\ldots\ldots\ldots + P(n-1)(x_i^n-x^{n-1}) = 0 \tag{4.31}$$

where i=1, 2, 3, .n

The above equations we can easily find the utilization of the server.

Server Utilization = ρ / n

$$= \frac{1}{n}\{0P(0) + 1(P(1)) + 2P(2) + 3P(3)........ + (n-1)P(n-1) + n[P(n) + P(n+1) + ...]\}$$

$$\therefore \rho = n[1-P(0) - P(1)\ldots\ldots\ldots -P(n-1)] + P(1) + 2P(2) + \ldots\ldots\ldots\ldots + (n-1)[P(n-1)]$$

$$\therefore n-\rho = nP(0) + (n-1)P(1) + \ldots\ldots\ldots\ldots + 2P(n-2) + P(n-1) \tag{4.32}$$

The chance that a job does not have to wait is

$$= \sum_{j=0}^{n-1} P(j)$$

$$= P(0) \tag{4.33}$$

The average waiting time in the steady state can be shown to be related to the average no. of jobs in the queue and the arrival rate of jobs by [20,21]

$$L = \lambda w \tag{4.34}$$

Where L = average number of jobs in the queue

λ =arrival rate

w = average waiting time

L = steady state probabilities

Figure 4.18 shows the relationship between the average waiting time for the greater server power with respect to traffic intensity. Traffic intensity is varied from 0.1 to 0.9 and server power from 2 to 8 and the corresponding waiting time is found out. Values can be seen in Appendix A.

Here, for lower utilization of the queue, waiting time of the calls reduces significantly, and in the case of higher utilization, there is not the desired efficiency. For example, for lower utilization, if the server power is doubled, the waiting time is reduced by more than 15 times. But as utilization increases, the waiting time is reduced by only two times. Therefore, we can conclude that merely increasing the server capacity is not an intelligent solution. It is also necessary to have a very efficient algorithm to run these servers for increased productivity.

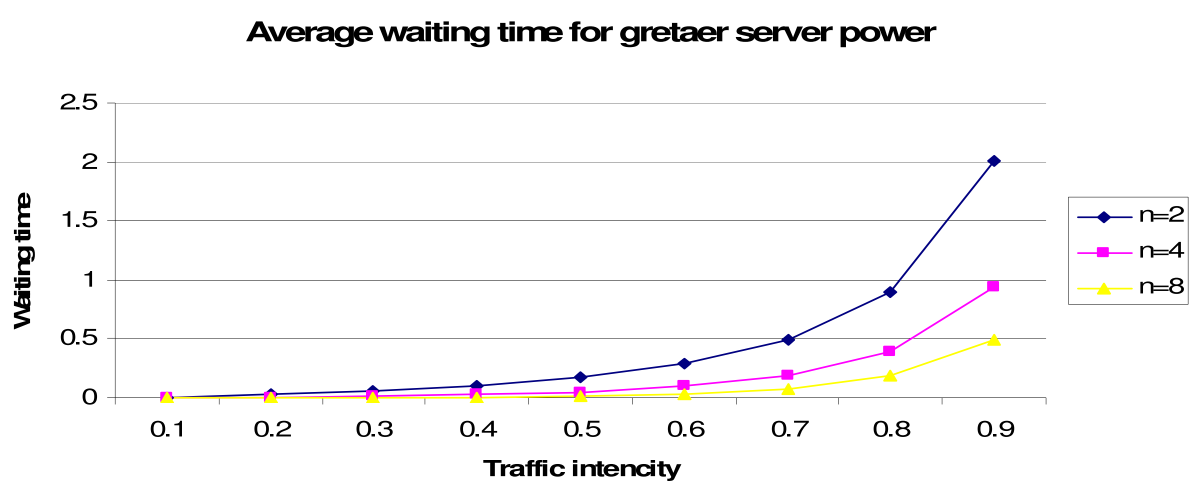

Figure 4.18 Average waiting time for greater server power v/s traffic intensity

CHAPTER 5

CONCLUSION

The proposed algorithm fared better than all the other schemes. We can observe that no calls were dropped in all the four cases, whereas at least one call was dropped for the other algorithms. So, it helps in minimizing call blocking probability. Since it uses the fixed path, so, there is always a chance that it chooses the shortest distance to the path. It does one calculation at a time, so, it will save valuable CPU recourses. It was observed that the probability that a new call is accepted is dependent on the probe arrival rate, probe packet size should be small so that the number of probe packets can be increased per probe period for more accurate results and other network characteristics such as busy period, utilization, waiting time and number of successful calls heavily depends on the arrival rate of the probe packets as the service time remains constant. It was found that the probability of call acceptance does depend on the number of probe packets in the network for low values of Pme and for larger values of Pme, the probability does not change much. The probability that a call is accepted is directly proportional to the probability of success of one probe packet. I tried to see the effect of increase in server power for the call admission and found that for lower utilization of the queue, waiting time of the calls reduces significantly and in case of higher utilization we do not get the desired efficiency. For example, for lower utilization, if the server power is doubled the waiting time reduces by more than fifteen times. But as the utilization increases the waiting time reduces by only two times. So, we can conclude that merely increasing the server capacity is not an intelligent solution. We also need to have a very efficient algorithm to run these servers for increased productivity.

CHAPTER 6

FUTURE WORK

Though the proposed algorithms are proved mathematically but due to lack of resources it could not be simulated in real time environment. So, these algorithms can be tested in Cisco call manager and observe how it matches the analytical results. One can also observe the probe thrashing effect on the network when a number of calls are trying to establish a session simultaneously. Also, one can improve these algorithms to increase its efficiency further as there is always a chance for improvement.

REFERENCES

REFERENCES:

[1] L. Zhang, S Deering, D Estrin, S Shenker and D Zappala "RSVP: A New Resource ReSerVation Protocol," IEEE Network, vol. 7, pp. 8-18, September 1993.

[2] D Black, S Blake, M Carlson, E Davies, W. Weiss and Z Wang, "An Architecture for Differentiated Services," Internet RFC 2475, 1998.

[3] Z. L Zhang, Z Duan, L Gao and YT Hou, "Decoupling QoS Control from Core Routers: a Novel Bandwidth Broker Architecture for Scalable Support of Guaranteed Services," In Proceedings of ACM SIGCOMM'00, Stockholm, Sweden, August 2000.

[4] I. Stoica and H. Zhang, "Providing Guaranteed Services without per Flow Management", In Proceedings of ACM SIGCOMM'99, Cambridge, MA, August 1999.

[5] F. Baker, C. Iturralde, F. le Faucher and B. Davie, "Aggregation of RSVP for IPv4 and IPv6 Reservations," Internet Draft, draft-ietf-issll-rsvp-aggr-02.txt, Mach 2000.

[6] C. Centinkaya and E. Knightly "Scalable Services via Egress Admission Control," In Proceedings of IEEE INFOCOM'00, Tel Aviv, Israel, March 2000.

[7] J. Qiu and E. Knightly "Inter-class Resource Sharing using Statistical Service Envelopes," In Proceedings of IEEE INFOCOM'99, New York, NY, March 1999.

[8] V. Elek et al. "Admission Control Based on End-to-End Measurements," In Proceedings of IEEE INFOCOM'00, Tel Aviv, Israel, March 2000.

[9] L. Breslau et al. "End-point Admission Control: Architectural Issues and Performance," In Proceedings of ACM SIGCOMM'00, Stockholm, Sweden, August 2000.

[10] Xiang Kong and Kenichi Mase, Dynamic routing with endpoint admission control for VOIP networks, Department of Information Engineering, Niigata University, Japan, 2003.

[11] Takashi Ezaki, Hirotada Kawakami and Koichi Asatani, A new VOIP Call Admission Control Scheme with use of Alternate Routing for Low Call Loss Probability, Graduate School of Electrical and Electronic Engineering, Kogakuin University, Tokyo. Japan, 2005.

[12] J. Medhi, Stochastic models in queuing theory, 2nd ed., Amsterdam ; Boston : Academic Press, c2003.

[13] Oscar Kempthorne and Leroy Folks , Probability statistics and data analysis, 1st ed., Ames, Iowa State University Press ,1971.

[14] Hideaki Takagi, Queuing analysis: Vocation and priority systems, v.1, Elsevier science publishers B.V, 1991.

[15] Sanjay K Bose, Introduction to Queuing theory, springer publisher, Mathematics - 2002

[16] Hideaki Takagi, Queuing analysis: Finite systems, v.4, Elsevier science publishers B.V, 1991.

[17] L. Gorney, Queuing Theory: A Problem Solving Approach. New York: Petrocell Books, 1982.

[19] Donald Gross, Carl M. Harris, Fundamentals of Queuing Theory, Wiley Publication, 2001.

[20] E. Page, Queuing Theory in OR, Crane Russak & Company, Inc., N.Y., 1971.

[21] R. K. Jain, The Art of Computer Systems Performance Analysis : Techniques for Experimental Design, Measurement, Simulation, and Modeling, Wiley Publishers, 1991.

[22] J. W. Roberts (Editor), COST 224: Performance evaluation and design of multiservice networks, Commision of the European Communities, 1991.

[23] JSTOR, Biometrica, Biometrika Trust Vols. 1 - 87, 1901-2000.

[24] Lee Breslau, Edward W. Knightly, Scott Shenker, Ion Stoica, Hui Zhang, Endpoint admission control: Architectural issues and performance, Proceedings of the conference on Applications, Technologies, Architectures, and Protocols for Computer Communication, Stockholm, Sweden, 2000.

[25] Stylianos Georgoulas, Panos Trimintzios, George Pavlou and Kin-Hon Ho , Measurement-based admission control for real-time traffic in IP differentiated services networks, 2003.

[26] Bianchi, G., Capone, A. and Petrioli, C., Dipt. di Ingegneria Elettrica, Palermo Univ Throughput analysis of end-to-end measurement based admission control in IP, INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE, March 2000.

[27] Viktoria Elek, Gunnar Karlsson and Robert Ronngren, Admission control based on end-end measurement, Royal Instititute of Technology, Sweden, IEEE Infocom, 2000.

[28] Han S. Kim and Ness B. Shroff, An approximation of the end-to-end delay distribution, School of Electrical and Computer Engineering, Purdue, IN, 2004.

[29] G. Bianchi, F. Borgonovo, A. Capone, L. Fratta, and C. Petrioli, End-point admission control with delay variation measurement for Qos in IP networks, ACM SIGCOMM Computer Communications Review, 2002.

[30] Lee Breslau and Sugih Jamin, Comments on the performance of measurement-based admission control algorithms, Proceedings of IEEE INFOCOM 2000, Tel Aviv, Israel, March 2000.

[31] Ignacio M´as, Viktoria Fodor, and Gunnar Karlsson,The performance of Endpoint admission control based on packet loss, Department of Microelectronics and Information Technology, KTH, Royal Institute of Technology, Kista, Sweden, 2003.

[32] K Mase, Y Toyama, AA Bilhaj, Y Suda, Qos management for VoIP networks using edge-to-edge admission control, Global Telecommunications Conference, 2001. GLOBECOM '01. IEEE, TX, USA, 2001.

[33] K Mase and H Kobayashi, An efficient end-to-end measurement based admission control for VoIP networks, Communications, 2004 IEEE International Conference on, 2004.

[34] M Ivars and G Karlsson, PBAC: Probe based admission control, Department of Microelectronics and Information Technology, Royal Institute of Technology (KTH), 16440 Kista, Sweden, 2002.

[35] Susana Sargento, Rui Valadas and Edward Knightly, Call admission control in IP networks with qos support, Instituto de Telecomunicações, Universidade de Aveiro, P-3810 Aveiro, Portugal, ECE Department, Rice University, Houston TX 77005, USA, 2001.

[36] Cisco Systems, Endpoint admission control: Cisco document, 2003,World Wide Web: < http://www.cisco.com/univercd/cc/td/doc/cisintwk/intsolns/voipsol/cac.htm >.

[37] I Mas, V Fodor, G Karlsson - Probe based admission control for multicast, Quality of Service, Tenth IEEE International Workshop, 2002.

[38] Frank P. Kelly, Peter B. Key and Stan Zachary, Distributed admission control, IEEE Journal on Selected Areas in Communications, 2000.

[39] G. Karlsson, "Providing quality for Internet video services,"' in Proc. CNIT/IEEE 10th Int'l Tyrrhenian Workshop on Dig. Comm., pp. 133-146, Ischia, Italy, Sept. 15-18, 1998

[40] Kenichi Mase, Yuichiro Toyama, Abdulkhalig A. Bilhaj, and Yosuke Suda, Qos management for voip networks with edge-to-edge admission control, Department of Information Engineering, Niigata University, Global Telecommunication Conference, Niigata 950-2181, Japan, 2002.

[41] Cisco Systems, Endpoint admission control: Cisco document, 2003,World Wide Web: < http://www.cisco.com/univercd/cc/td/doc/cisintwk/intsolns/voipsol/cac.htm

[42] Howstuffswork.com World Wide Web: <http://electronics.howstuffworks.com/ip-telephony1.htm>

APPENDIX

APPENDIX A

TABLE A.1

PERFORMANCE OF ALL THE ALGORITHMS

	PATH 1	PATH 2	PATH 3
A1	2.04	4.041	3.058
A2		4.041	3.058
A3		4.041	3.058
A4	2.04	4.041	
	PATH 1	PATH 2	PATH 3
A1	2.04	4.041	3.058
A2		4.041	3.058
A3		4.041	5.058
A4	2.04	4.041	
	PATH 1	PATH 2	PATH 3
A1	2.04	4.041	5.058
A2		4.041	5.058
A3		4.041	5.058
A4	2.04	4.041	
	PATH 1	PATH 2	PATH 3
A1	5.04	4.041	3.058
A2		4.041	3.058
A3		4.041	3.058
A4	5.04	4.041	

Green: accepted calls

Red: rejected calls.

TABLE A.2

AVERAGE NUMBER OF CALLS IN THE SYSTEM

ρ	X
0.1	0.1055
0.2	0.225
0.3	0.3643
0.4	0.5333
0.5	0.75
0.6	1.05
.7	1.5167
0.8	2.4
0.9	4.95

TABLE A.3

AVERAGE NUMBER OF CALLS IN THE QUEUE

ρ	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
x	0.0056	0.025	0.064	0.1333	0.25	0.45	0.8167	1.6	4.05

TABLE A.4

AVERAGE WAITING TIME FOR ALL THE PACKETS IN SECONDS

ρ	S=1	S=2	S=3	S=4	S=5	S=6	S=7	S=8	S=9
.1	.056	.111	.167	.222	.278	.333	.389	.444	.5
.2	.125	.25	.375	.5	.625	.75	.875	1	1.125
.3	.214	.43	.643	.857	1.071	1.286	1.5	1.714	1.93
.4	.33	.67	1	1.33	1.67	2	2.33	2.67	3
.5	.5	1	1.5	2	2.5	3	3.5	4	4.5
.6	.75	1.5	2.25	3	3.75	4.5	5.25	6	1.75
.7	1.167	2.33	3.5	4.67	5.83	7	8.167	9.33	10.5
.8	2	4	6	8	10	12	14	16	18
.9	4.5	9	13.5	18	22.5	27	31.5	36	40.5

TABLE A.5

AVERAGE BUSY PERIOD IN SECONDS

ρ	S=1	S=2	S=3	S=4	S=5	S=6	S=7	S=8	S=9
.1	1.11	2.22	3.33	4.44	5.55	6.66	7.77	8.88	9.99
.2	1.25	2.5	3.75	5	6.25	7.5	8.75	10	11.25
.3	1.43	2.86	4.29	5.72	7.15	9.58	10.01	11.44	12.87
.4	1.67	3.33	5	6.67	8.33	10	11.67	13.33	15
.5	2	4	6	8	10	12	14	16	18
.6	2.5	5	7.5	10	12.5	15	17.5	20	22.5
.7	3.33	6.67	10	13.33	16.67	20	23.33	26.67	30
.8	5	10	15	20	25	30	35	40	45
.9	10	20	30	40	50	60	70	80	90

TABLE A.6

PROBABILITY OF ACCEPTANCE

P_{succ}	P_a
.1	.07371
.2	.1088
.3	.12423
.4	.13504
.5	.15625
.6	.20256
.7	.28987
.8	.43648
.9	.66429

TABLE A.7

PROBABILITY THAT A CALL IS ACCEPTED

n_{pr}	P_{me}=.01	P_{me}=.02	P_{me}=.03	P_{me}=.04	P_{me}=.05	P_{me}=.06	P_{me}=.07	P_{me}=.08	P_{me}=.09
100	.011425	.02665	.03810	.049308	.0604	.0713	.084	.093	.104
200	.0133421	.02470	.03573	.04658	.05732	.06797	.0786	.0891	.0996
300	.01272	.02384	.03468	.04537	.05597	.06651	.0769	.0874	.0978
400	.010041	.02333	.03405	.04465	.05517	.0656	.07605	.0864	.0968
500	.01211	.02297	.03362	.04416	.05463	.0650	.0754	.0857	.0961
600	.01193	.02271	.03330	.0438	.0542	.0646	.0749	.0853	.0955
700	.011739	.02251	.03306	.0435	.054	.0643	.0746	.0849	.0951
800	.01167	.02235	.03286	.04329	.05366	.0639	.0743	.0845	.0948
900	.01157	.02221	.03270	.04310	.05345	.0637	.0740	.08	.0945

TABLE A.8

NUMBER OF PROBE PACKETS

l_{pr}	n_{pr}
50	25600
100	12800
150	8533.33
200	6400
250	5120
300	4266.67
350	3657.1429
400	3200
450	2844.44
500	2560

TABLE A.9

STEADY STATE PROBABILITY FOR 1 CALL

ρ	S
.1	.0946
.2	.1771
.3	.2449
.4	.2950
.5	.3244
.6	.3288
.7	.3041
.8	.2451
.9	.1459

TABLE A.10

STEADY STATE PROBABILITY FOR TWO OR MORE THAN TWO CALLS

ρ	S
.1	$4.5*10^{-3}$
.2	$32*10^{-3}$
.3	$-3.4*10^{-3}$
.4	-.0204
.5	-.015
.6	-.024
.7	-.015
.8	-20.375
.9	-.083

TABLE A.12

NUMBER OF CALLS WITH BOREL'S DISTRIBUTION

q	S
.1	$5.07*10^{57}$
.2	$4.86*10^{47}$
.3	$2.62*10^{36}$
.4	$4.36*10^{28}$
.5	$3.8*10^{12}$
.6	$2.2*10^{3}$
.7	$1.03*10^{1}$
.8	0

TABLE A.14

AVERAGE WAITING TIME FOR GREATER SERVER POWER [20]

	n=2	n=4	n=8
0.1	0.0062	0.0002	0
0.2	0.0242	0.0021	0.0001
0.3	0.0553	0.0085	0.0009
0.4	0.1033	0.0227	0.0043
0.5	0.1767	0.0497	0.0135
0.6	0.293	0.0984	0.0342
0.7	0.4936	0.1897	0.0788
0.8	0.903	0.386	0.1833
0.9	2.0138	0.934	0.4894