INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

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

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

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

ProQuest Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600

DNS Based Routing Protocol in Ad hoc Networks

Linying Tong

A Thesis

in

The Department

of

Electrical and Computer Engineering

Presented in Partial Fulfillment of Requirements

for the Degree of Master of Applied Science at

Concordia University

Montreal, Quebec, Canada

August 2001

ABSTRACT

DNS Based Routing Protocol in Wireless Ad hoc Networks

Linying Tong

In this thesis, we consider a large population of mobile stations that are interconnected by a multihop wireless networks. DNS Based Routing Protocol is introduced for use in the multi-hop wireless ad hoc networks. In this protocol, each node knows the node connectivity within its reachable set; some nodes have DNS capability, which are responsible for route discovery. The routing is performed on two levels: Local node and terrestrial Global IP. By using DNS technique, the performance of the networks is highly improved. Simulation results give the characteristics and functionalities of the protocol, including forwarding latency, overhead cost, total transmission delay and network throughput. This routing protocol supports real-time streams.

ACKNOWLEDGEMENTS

At the outset I would like to express my most sincere gratitude to my thesis supervisor, Dr. A. K. Elhakeem, for his guidance, support, encouragement, especially his patience during the entire course of this thesis. This work was born form his ideas and his intellectual properties, also he provided constructive suggestions to enable me to demonstrate this technique by theoretical analysis and computer simulation. I am deeply impressed by his solid knowledge and rich experiences in the area of communication. Here I would like to express my greatest admiration to my supervisor again.

I give my special thanks to Mr. Yan Feng and Mrs. Xiao Jing, who give me very useful advice and suggestion to this thesis. Also I would like to thank my great group of friends in Montreal who help me during the course of this work.

Finally this is dedicated to my wonderful family, my husband, my parents, my sister and brother. They are always standing besides me and helping me with love and encouragement during the entire period of the study of my Masters.

Index

List of Symbols and Abbreviations

DNS	Domain Name Server
WLAN	Wireless Local Area Networks
MAC	Medium Access Control
PHY	Physical Layer
BSS	Basic Service Set
AP	Access Point
ESS	Extended Service Set
DS	Distribution System
PDU	Protocol Data Unit
CSMA/CS	Carrier Sense Multiple Access with Collision Avoidance
CSMA/CD	Carrier Sense Multiple Access with Collision Detection
DIFS	Distributed Interframe Space
ACK	Acknowledgment
SIFS	Short Interframe Space
RTS	Request To Send
CTS	Clear To Send
DCF	Distributed Coordination Function
PCF	Point Coordination Function
DSDV	Destination Sequenced Distance Vector Routing Protocol
AODV	Ad-hoc On-demand Distance Vector Routing Protocol

ZRP	Zone Routing Protocol
RREQ	Route Request
RREP	Route Reply
HSR	Hierarchical Routing Protocol
DBRP	DNS Based Routing Protocol
IP	Internet Protocol
TCP	Transmission Control Protocol
QoS	Quality of Service
TTL	Time To Live
ATTD	Average Total Transfer Delay
VTTD	Variance of Total Transfer Delay
AQD	Average Queuing Delay
VQD	Variance of Queuing Delay
ABO	Average Buffer Overflow
VBO	Variance of Buffer Overflow
OH	Overhead
AT	Average Throughput
AL	Average Latency
VL	Variance Latency
FDB	Forward Database
RLT	Route List Table

Table of Figures

CHAPTER 1

1 INTRODUCTION

1.1 Introduction to IEEE 802.11 Wireless Local Area Network

Wireless local area networks (WLANs) are the same as the traditional LAN but they have a wireless interface. With the introduction of small portable devices such as PDAs (personal digital assistants), the WLAN technology is becoming very popular. WLANs offer the following advantages over traditional wired network:

Mobility: Wireless LAN systems allow users to move their wireless device (i.e. laptop, organizer) throughout the WLAN coverage area without any disturbance to network access. This enables real-time access to network resources.

Simplicity of installation: With the use of wireless LAN's, network managers can set up or augment network without installing or moving wires. All they have to do is equip all LAN devices with a WLAN card and they are ready to go. This significantly decreases installation time.

Reduced Cost: Even though the initial cost of wireless LAN hardware can be higher than the cost of wired LAN hardware, overall installation expenses and the life-cycle costs can be significantly lower.

Scalability: WLANs are easily expanded by the addition of more access points, they can be easily reconfigured (no need to change Ethernet cable configurations) into different network topologies, from peer-to-peer networks suitable for a small number of users to full infrastructure networks of thousands of users that enable roaming over a broad area.

Wireless Local Area Networks (WLANs) are being developed to provide high bandwidth to users in a limited geographical area. WLAN can be used in many places, a common one is wireless office. Other examples include conference registrations, campus classrooms, emergency relief centers, tactical military communications, and so on [2] [3].

Currently, there are two emerging WLAN standards: the European Telecommunications Standards Institute (ETSI) High Performance European Radio LAN (HIPERLAN) and IEEE 802.11 WLAN. Both stardards cover the physical layer and medium access control(MAC) sublayer of the open systems interconnection(OSI) seven-layer reference model.

The scope of IEEE 802.11 is " to develop a Medium Access Control (MAC) and Physical Layer (PHY) specification for wireless connectivity for fixed, portable and moving stations within a local area." The purpose of standard is twofold:

- " To provide wireless connectivity to automatic machinery, equipment, or stations that require rapid deployment, which may be portable, or hand-held or which may be mounted on moving vehicles within a local area"
- " To offer a standard for use by regulatory bodies to standardize access to one or more frequency bands for the purpose of local area communication"[4].

1.1.1 802.11 Topology

The IEEE 802.11 standard considers two network topologies: ad hoc and infrastructure based. In an ad hoc configuration (see Fig. 1), the mobile terminals communicate with each other in an independent basic service set (BSS) without

connectivity to the wired backbone network. In an infrastructure network (see Fig. 2), mobile terminals communicate with the backbone network through an access point (AP)[5]. The AP is a bridge supporting range extension by providing the integration points necessary for network connectivity between multiple BSSs, thus forming an extended service set (ESS). In other words, the ESS consists of multiple BSSs that are integrated together using a common distribution system (DS). A mobile terminal can roam among different BSS in one ESS without losing connectivity to the backbone.

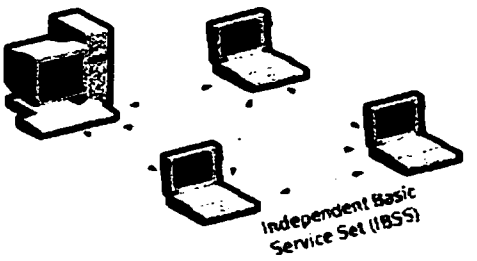


Figure 1- 1: Ad Hoc Mode

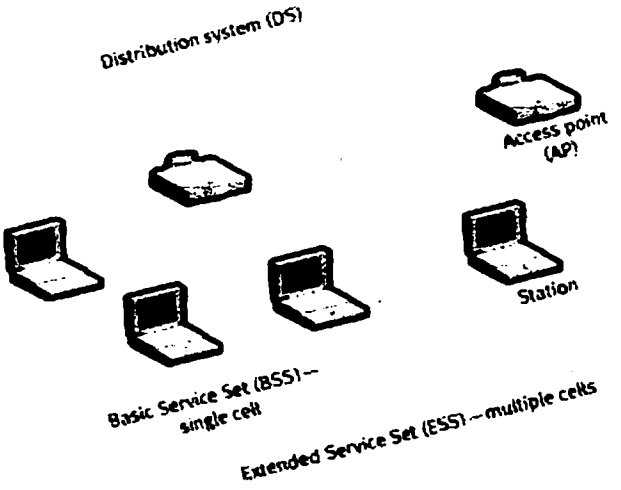


Figure 1- 2: Infrastructure Mode

1.1.2 802.11 Physical Layer

The 802.11 standard provides for three different types of physical layers to be used:

- 2.4 GHz ISM band ***frequency hopping*** (FH) spread-spectrum radio
- 2.4 GHz ISM band ***direct sequence*** (DS) spread-spectrum radio
- Infrared (IR) light [6]

The FHSS utilizes the 2.4 GHz frequency band. In FH systems, the frequency at which data is transmitted is varied among a set of frequencies, in the United States, a maximum of 79 channels are specified in the hopping set. The transmitter sends data on a given frequency for a fixed length of time (the dwell time in 802.11) and then switches to the next frequency for another fixed length of time. The FH pattern is known to the receiver so that the receiver's frequency synthesizer can hop in synchronism and recover the original data signal. The FH systems defined in the 802.11 PHY are slow FH systems. In FH systems, adjacent or overlapping BSSs use different hopping patterns. In the United State, three different hopping sequence sets are established with 26 hopping sequences per set. Different hopping sequences enable multiple BSSs to coexist in the same geographical area, which may become important to alleviate congestion and maximize the total throughput in a single BSS.

The DSSS also uses the 2.4 GHz frequency band. In DS systems, the original data signal is modulated by a wideband spreading signal. This spreading signal is known to the receiver, which can recover the original data signal. The factor by which the bandwidth of the signal is expanded is known as the processing gain of the DS

system; in 802.11, it is 11(10.4DB). In the United States, 11 DS center frequencies are defined.

The IR specification identifies a wavelength range from 850 to 950nm. The IR band is designed for indoor use only and operates with nondirected transmissions. The IR specification was designed to enable stations to receive lone-of-site and reflected transmissions.

In summary, since an FH system can offer a larger number of channels (e.g., frequency-hopping patterns) than a DS system, an FH system may be more useful for dense environments in which BSSs have overlap with many adjacent BSSs. Furthermore, FH and DS systems have somewhat different types of resilience to narrowband interference. FH systems experience the interference only for a fraction of time, whereas DS systems experience a fraction of the interference power all of the time. Thus, FH systems have the performance advantage if the interference is high, DS systems if the interference is low.

1.1.3 802.11 MAC Layer

The MAC sublayer is responsible for the channel allocation procedures, protocol data unit (PDU) addressing, frame formatting, error checking, and fragmentation and reassembly [1]. The 802.11 MAC layer protocol provides asynchronous, time-bounded, and contention free access control on a variety of physical layers. These functions are provided independent of the characteristics of the underlying physical layers and/or data rates. The basic access method in the 802.11 protocol is the distributed coordination function (DCF), which is best described as the

carrier sense multiple access with collision avoidance (CSMA/CA) protocol. In addition to the DCF the 802.11 also incorporates an alternative optional access method known as the point coordination function (PCF), an access method similar to polling that uses a point coordinator (the AP) to determine which station has right to transmit [7].

Distributed Coordination Function

The DCF is basically a Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA). CSMA protocols are well known in the industry, where the most popular is the Ethernet, which is CSMA/CD (collision detection). While collision detection (CD) is useful for wired LANs, it cannot be used in a wireless LAN environment because of the following reasons:

- The ability to detect collisions requires the ability to both send (one's own signal) and receive (to determine if another station's transmissions is interfering with one's own transmission) at the same time. This can be costly.
- In a wireless environment, we can not assume all stations hear each other (which is the basic assumption of collision detection scheme), and the fact that a station is willing to transmit and senses the medium free, does not necessarily means that the medium is free around the receiver area. In fact, even if one station had collision detection and sensed no collision when sending, a collision could still occur at the receiver. This situation results from the particular characteristics of the wireless channel. With the so-called hidden terminal problem, physical obstructions in the environment (i.e., a

mountain) may prevent station hearing each other's transmissions. Fading also results in undetectable collision at the receiver.

When using DCF, a station with a packet to transmit, monitors the channel activity until an idle period equal to a distributed interframe space (DIFS) is detected. After sensing an idle DIFS, the station generates a random backoff interval before transmitting. The backoff time counter is decremented as long as the channel is sensed idle, stopped when a transmission is detected on the channel, and reactivated when the cannel is sensed idle again for more than a DIFS. And then, a positive acknowledgment (ACK) is transmitted by the destination station to signal the successful packet transmission. To allow an immediate response, the ACK is transmitted following the received packet, after a short interframe space (SIFS). Figure 3 is a timing diagram illustrating the successful transmission of a data frame. When the data frame is transmitted, the duration field of the frame is used to let all stations in the BSS know how long the medium will be busy [8].

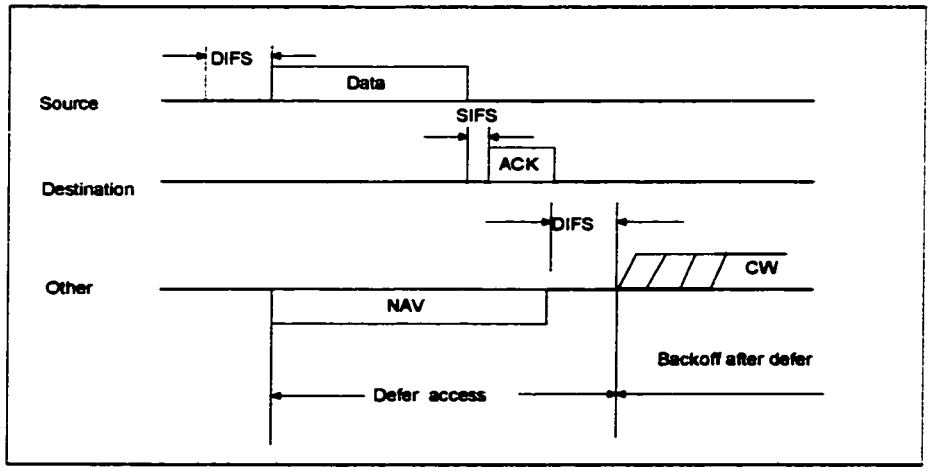


Figure 1- 3: Transmission of a data frame without RTS/CTS

If the PDU is large, a lot of channel bandwidth is wasted due to a corrupt PDU. Request to send (RTS) and clear to send (CTS) control frames can be used by a station to reserve channel bandwidth prior to the transmission of a data frame, this mechanism can improve the system throughput. As shown in Figure 4, an RTS frame is transmitted by a station which needs to transmit a packet. When the receiving station detects an RTS frame, it responds, after a SIFS, with a CTS frame. The transmitting station is thus allowed to transmit its packet only if it correctly receives the CTS frame. Moreover, the frames RTS and CTS carry the information of the length of the packet to be transmitted. This information can be read by each station, which is then able to update its NAV. If a collision occurs with an RTS or CTS, far less bandwidth is wasted when compared to a large data frame. However, for a lightly loaded media, additional delay is imposed by the overhead of the RTS/CTS frames.

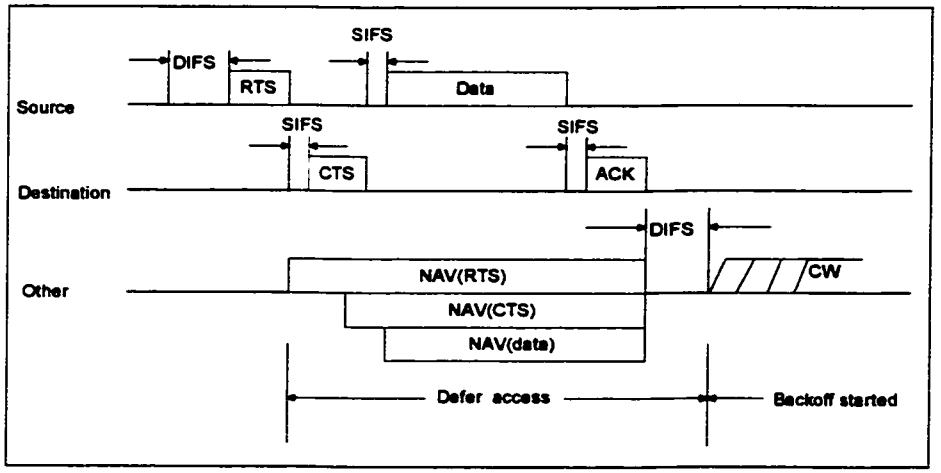


Figure 1- 4: Transmission of a data frame using RTS/CTS

Point Coordination Function

In order to support time-bounded services, the 802.11 standard specifies the optional use of the aforementioned PCF. The PCF is an optional capability, which is connection-oriented, and provides contention-free (CF) frame transfer. The PCF relies on the point coordinator (PC) to perform poling, enabling polled stations to transmit without contending for the channel. That is, when the PCF is active, the PCF station allows only a single station in each BSS to have priority access to the medium at any one time. This is implemented through the use of the point coordination function IFS (PIFS) and a beacon frame (Fig. 5) that notifies all of the other stations in the cell not to initiate transmissions for the length of the contention-free period (CFP). The PCF is required to coexist with the DCF and logically sits on top of the DCF (Fig. 6). The CFP repetition interval is used to determine the frequency with which the PCF occurs. Within a repetition interval, apportion of the time is allotted to contention-free traffic, and the remainder is provided for contention-based traffic.

1.2 Thesis Outline

The information and work presented in this thesis is organized as follows:

In chapter 1, a brief introduction of IEEE 802.11 wireless Local Area Network Protocol is presented, IEEE 802.11 topology, physical layer and MAC layer is presented.

In chapter 2, recount some of the current routing protocols used for Wireless Ad hoc network. We catalogued two types of current ad hoc routing techniques. DSDV, AODV, ZRP and HSR routing protocols are emphasized.

In chapter 3, we proposed a new routing protocol named DNS Based Routing Protocol in wireless ad hoc networks. We give a detailed description of the scheme and its various parameters are presented. We also give the analysis of our new protocol; the goal is to obtain the performance of this protocol. We begin with analysis of the Average Total Transfer Delay, Average Queuing Delay, followed by evaluation of the Average Buffer Overflow, Overhead, then verification of the Throughput and Average Latency of the First Route Reply of the networks. Results are shown in figures with respect to various parameters. Right through these discussions, we could see the property of our DBRP.

In chapter 4, summary, conclusion and suggestions for future work is presented.

CHAPTER 2

2 OVERVIEW OF ROUTING PROTOCOLS IN AD HOC NETWORK

2.1 Introduction of Ad-hoc routing techniques for wireless LANs

Ad hoc networks are dynamically self-organizing and self-configuring, with nodes establishing the necessary routing between each other without requirement for any existing infrastructure or administration. Ad hoc networks are easier to deploy compared to cellular counterparts which necessities expensive planning, support ground networks and expensive base-stations. Failure of few access points (Wireless LAN base-stations) many not lead to overall network failure. On the other hand, the limited communication range Wireless LANs may have is more than offset by the high rate and cheaper operations in the unlicensed 900MHz, 1900 MHz, 2.4GHz and 6GHz bands.

Ad hoc networks have no fixed routers; all nodes are capable of movement and can be connected dynamically in an arbitrary manner. Nodes of these networks function as routers, which discover and maintain routes to other nodes in the network. Ad-hoc routing techniques for wireless LANs can be categorized as follows [Iwata 1999]:

a) Global Pre-computed routing (sometimes called flat, Table-Driven or Proactive):

Where all routes to all nodes are computed a priori and updated periodically at each node. Distance Vector Routing DSDV [Perkins 1994] and links state LS [Jaquet 2000] routing fit into this category.

b) Hierarchical Routing:

Here two kinds of nodes exist, end points and switches. End points select the switch (similar to the BS of the cellular system) and form a cell around it. In their turn, the switches form clusters among themselves. Cluster heads are appointed and form a higher level cluster and so on.

c) Flooding and limited flooding techniques:

Here a node hearing a data racket rebroadcasts to all neighbors and so on. To limit the amount of overhead traffic, limited flooding techniques are proposed [Elhakeem 2000]

d) On-Demand Routing techniques (sometimes called Reactive or Source-initiated):

Here nodes issue route requests and receive route replies, as the need arises with no build-up of routing tables nor Forwarding data basis.

2.2 Destination-Sequenced Distance-Vector Routing (DSDV)

The Destination-Sequenced Distance-Vector Routing protocol (DSDV) is a table-driven algorithm based on the classical Bellman-Ford routing mechanism. The improvements mad to the Bellman-Ford algorithm include freedom from loops in routing tables.

Every mobile mode in the network maintains a routing table in which all of the possible destinations within the network and the number of hops to each destination are recorded. Each entry is marked with a sequence number assigned by the destination node. The sequence number is assigned by the destination node. The sequence numbers enable the mobile nodes to distinguish stale routes from new ones, thereby avoiding the formation of routing loops. Routing table updates are periodically transmitted throughout the network in order to maintain table consistency. To help alleviate the potentially large amount of network traffic that such updates can generate, route updates can employ two possible types of packets. The first is known as a full dump. This type of packet carries all available routing information and requires multiple network protocol data units (NPDUs). During periods of occasional movement, these packets are transmitted infrequently. Smaller incremental packets are used to relay only that information which has changed since the last full dump. Each of these broadcasts should fit into a standard-size NPDU, thereby decreasing the amount of traffic generated. The mobile modes maintain an additional table where they store the data sent in the incremental routing information packets.

New route broadcasts contains the address of the destination, the number of hops to reach the destination, the sequence number of the information received

regarding the destination, as well as new sequence number unique to the broadcast. The route labelled with the most recent sequence number is always used. In the event that two updates have the same sequence number, the route with the smaller metric is used in order to optimize (shorten) the path. Mobiles also keep track of the settling time of routes, or the weighted average time that routs to a destination will fluctuate before the route with the best metric is received. By delaying the broadcast of a routing update by the length of the settling time, mobiles can reduce network traffic and optimize routs by eliminating those broadcasts that would occur if a better route was discovered in the very near future.

DSDV is a flat routing technique that propagates 100% all the routing table contents of all nodes, and with the same frequency for all table entries. The frequency of transmitting the routing tables becomes higher as nodes speeds increase. DSDV is inefficient because of the requirement of periodic update transmissions, regardless of the number of changes in the network topology. This effectively limits the number of nodes that can connect to the network since the overhead grows rapidly.

2.3 Ad Hoc On-Demand Distance Vector Routing (AODV)

The Ad Hoc On-Demand Distance Vector (AODV) routing protocol builds on the DSDV; moreover, it is an improvement on DSDV because it typically minimizes the number of required broadcasts by creating routes on a demand basis, as opposed to maintaining a complete list of routes as in the DSDV algorithm. The authors of AODV classify it as a pure on-demand route acquisition system, since nodes that are

not on a selected path do not maintain routing information or participate in routing table exchanges.

When a source node desires to send a message to some destination node and does not already have a valid route to that destination, it initiates a path discovery process to locate the other node. It broadcasts a route request (RREQ) packet to its neighbors, and so on, until either the destination or an intermediate node with a "fresh enough" route to the destination is located. Figure 2- 1 illustrates the propagation of the broadcast RREQs across the network. AODV utilizes destination sequence numbers to ensure all routes are loop-free and contain the most recent route information. Each node maintains its won sequence number, as well as broadcast ID. The broadcast ID is incremented for every RREQ the node initiates, and together with the node's IP address, uniquely identifies and RREQ. Along with its own sequence number and the broadcast ID, the source node includes in the RREQ the most recent sequence number it has for the destination. Intermediate nodes can reply to the RREQ only if they have a route to the destination whose corresponding destination sequence number is greater than or equal to that contained in the RREQ.

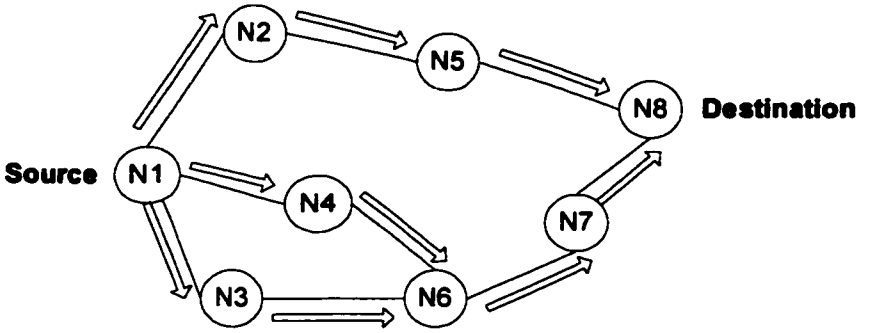
During the process of forwarding the RRWQ, intermediate nodes record in their route tables the address of the neighbor from which the first copy of the broadcast packet is received, thereby establishing a reverse path. If additional copies of the same RREQ are later received, these packets are discarded. Once the RREQ reaches the destination or an intermediate node with a fresh enough route, the destination/intermediate node responds by unicasting a route reply (RREP) packet back to the neighbor from which it first received the RREQ (Fig. 2-3-1b). As the

RREP is routed back along the reverse path, nodes along this path set up forward route entries in their route tables that point to the node from which the RREP came. These forward route entries indicate the active forward route. Associated with each route entry is a route timer which will cause the deletion of the entry if it is not used within the specified lifetime. Because the RREP is forwarded along the path established by the RREQ, AODV only supports the use of symmetric links.

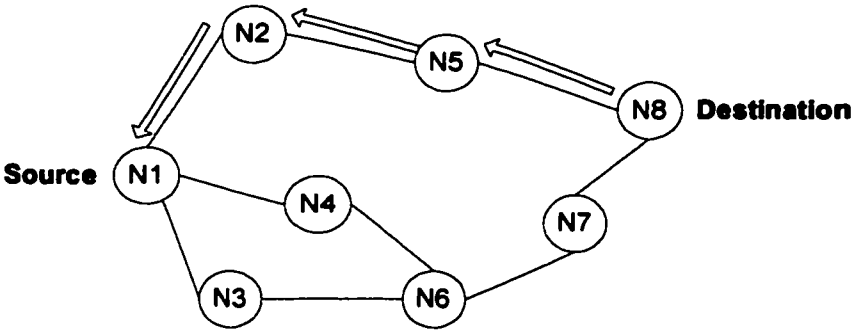
Routes are maintained as follow. If a source nodes moves, it is able to reinitiate the route discovery protocol to find a new route to the destination. If a node along the route moves, its upstream neighbor notices the move and propagates a link failure notification message (an RREP with infinite metric) to each of its active upstream neighbors to inform them of the erasure of that part of the route. These nodes in turn propagate the link failure notification to their upstream neighbors, and so on until the source node is reached. The source node may then choose to reinitiate route discovery for that destination if a route is still desired.

An additional aspect of the protocol is the use of Hello message, periodic local broadcasts by a node to inform each mobile node of other nodes in its neighborhood. Hello messages can be used to maintain the local connectivity of a node. However, the use of Hello messages is not required. Nodes listen for retransmission of data packets to ensure that the next hop is still within reach. If such a retransmission is not heard, the node may use any one of a number of techniques, including the reception of Hello messages, to determine whether the next hop is within communication range. The Hello messages may list the other nodes from which a mobile has heard, thereby yielding greater knowledge of network connectivity.

AODV offers features such as local repair of a route (due to intermediate or end node mobility. Gratuitous mode where an intermediate node who know the route to a certain destination would unicast this back immediately to the requesting source without further broadcasting the "R-request" to destination also exists in AODV. The advantage of AODV is its support for multicast, but on the downside, AODV requires symmetric links between nodes, and hence cannot utilize routes with asymmetric links.



(a) Propagation of the RREQ



(B) Path of the RREP to the source

Figure 2- 1:AODV route discovery

2.4 Hierarchical State Routing (HSR)

HSR combines dynamic, distributed multilevel hierarchical grouping (clustering) and efficient location management.

Clustering (dividing nodes into groups and different kinds of nodes) at the MAC and network layers help the routing of data as the number of nodes growth and subsequently the cost of processing (nodes memory and processing required).

HSR keeps a hierarchical topology, where selected cluster heads at the lowest level become member of the next higher level and so on... While this clustering is based on network topology (i.e. physical), further logical partitioning of nodes eases the location management problem in HSR. Figure 2-2 shows four physical level clusters namely CO-1, CO-2, CO-3 and CO-4. Cluster heads are selected either manually (by the network administrator) or via the appropriate real time distributed voting mechanism. In Figure 2-2, nodes 1,2,3,4 are assumed to be cluster heads, nodes 5, 9, 10 are internal nodes and nodes 6, 7, 8, 11 are gateway nodes.

Gateway nodes are those belonging to more than one physical cluster. At the physical cluster level all cluster heads, gateways, and internal nodes use the MAC address, but each also has a hierarchical address as will follow.

Cluster heads exchange the link state LS information with other cluster heads via the gateway nodes. For example in Fig. 14 cluster heads 2,3 exchange LS information via gateway 8 . Cluster heads 4,3 via gateway 11 and so on. In the sequel logical level C1-1 is formed of cluster heads 1, 2, 3, 4. However, in level C1-1, only 1 and 3 are cluster heads. Similarly at level C2-1, only 1 is a cluster head while node 3 is an ordinary node at C1-2 level.

Routing table storage at each node is reduced by the aforementioned hierarchical topology. For example, for node 5 to deliver a packet to node 10, it will forward it to the cluster head 1 which has a tunnel (route) to the cluster head 3 which finally deliver the packet to node 10. But how would the cluster head 1 know that node 10 is a member of a cluster headed by 3? The answer is: Nodes in each cluster exchange virtual LS information about the cluster (who is the head, who are the members) and lower cluster (with less details) and the process repeats at lower clusters. Each virtual node floods this LS information down to nodes within its lower level cluster. Consequently, each physical node would have hierarchical topology information (actually, summary of topology including cluster heads and member nodes) rather than the full topology of flat routing where all individual routes to each node in the networks are stored at all nodes and are exchanged at the same rate to all nodes.

The hierarchical address (a sequence of MAC addresses) is sufficient for packet delivery to any node in the network. For example, node HID (5)= (1, 1, 5) going from the top to the lowest cluster, 1 is the cluster head of clusters C1-1, CO-1 and node 5 is an interior node of CO-1. Similarly HID (6)= (3, 2, 6) and HID (10)= (3, 3, 10). Returning back to the example above where node 5 seeks a route to node 10. It will ask 1 (its cluster head). Node 1 has a virtual link or tunnel, i.e. the succession of nodes (1, 6, 2, 8, 3) to node 3 which is the cluster head of the final destination. This tunnel is computed from the LS information flooded down from higher clusters heads as above). Finally node 3, delivers the packet to node 10 along the downward

hierarchical path which is a mere single hop. Gateways have multiple HID since they belong to more then on physical cluster.

Utilization of long hierarchical addresses, and the cost of continuously updating the cluster hierarchical and hierarchical addresses as nodes move (memory and processing) of nodes are the shortcomings of HSR. As we described before, Clustering is dividing nodes into groups and different kinds of nodes. Also clustering and voting for cluster heads may consume more overhead never mentioning creating processing, security and reliability hazards at cluster heads at different hierarchical levels.

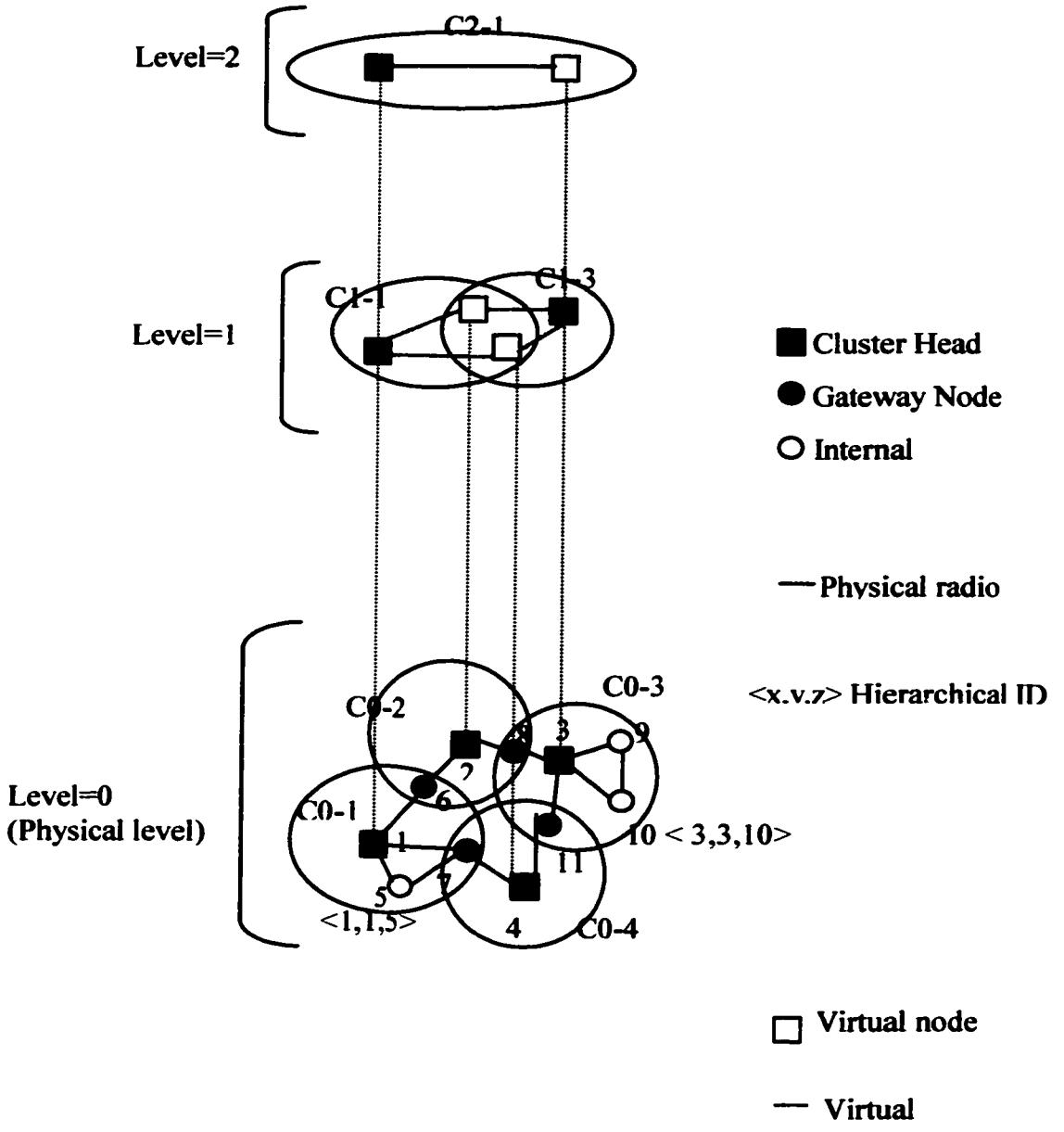


Figure 2- 2:An example of physical/virtual clustering

2.5 Zone Routing Protocol (ZRP)

ZRP is a hybrid reactive/proactive routing technique. Routing is flat and not hierarchical thus leading to reducing the processing and overhead. Nodes keep routing tables of nodes neighborhood and not of the whole network.

"Query/Reply" messaging is needed, if the destination is not found in the routing table. The "Query/Reply" process is handled only by certain selected nodes (called border nodes of the zone). On the other hand, all interior nodes repeat (Physical Broadcast) the "Query/Reply" packet but do not process them. The border nodes are neither gateways nor cluster heads such as the case in HSR. They are plain nodes located at the borders of the routing zone of the applicable node. The routing zone of a certain node of radius r>1 is defined as the collection of these nodes which can be reached via 1, 2, or 4... r hops from the applicable node. Nodes within one hop from the center node (e.g. node S in Figure 2- 3) are those who can directly hear the Radio transmission of node S. These are nodes A, B, C, D, E (one-hop nodes). The higher the power of a typical node S the larger is the number of one hop nodes. This may lead to increasing the control traffic necessary for these nodes to exchange their routing tables. Also this increases the level of contention of the IEEE 802.11 CSMA based access. The routing zone in Figure 2- 3 corresponds to a routing zone of two hops (r=2). Each node in this routing zone exchanges routing packets only with members in its routing zone (nodes A-K in Figure 2- 3). This takes place according to the Intrazone protocol (IARP), which forms the proactive part of the ZRP routing protocol.

Nodes G-K in Figure 2- 3 are called the peripheral nodes of node S. These again are ordinary nodes but just happened to be at 2 hops, i.e., at the border of the routing zone of node S (r==2, zone radius of routing zone of Figure 2- 3). For a different node, the peripheral nodes will change.

The IARP protocol provides nodes with routing tables with limited number of entries corresponding to nodes in the same routing zone .If the destination lies outside the routing zone such as node L. The calling node (say node S) will issue a "Query" to search for a route to node L, This query packet will be retransmitted by nodes A-E to the peripheral nodes. These are the only nodes that will process and responds to the route query packet.

This process is called Border-casting and the underlying protocol is called "Interzone Routing Protocol" IERP. The identities, distances, and # of hops of the peripheral nodes of applicable node as well as all other nodes (interior nodes) in its routing zone is assumed known to each node in the wireless Internet. Query packets are sent unicast or broadcast to all peripheral nodes if the applicable node does not find the destination in its routing tables. Further if these peripheral nodes do not find the required destination in their zone, they would forward the query to their own peripheral nodes and so on. If one of the peripheral nodes knows the route to destination, it will return a "Route -reply" packet to the node that sent the "Query". In Figure 2- 4, node S does not find the destination node D in its routing zone.

S broadcasts a "Query" packet to nodes C, G, H. Each of these retransmits this "Query" packet to its peripheral nodes, after finding that D is not in their routing zones. The process repeats till node B finds the destination node D in its routing zone.

Node B returns a "Reply" containing the ID sequence S-H-B-D. In this route accumulation routine process, each node adds its ID to the query packets then transmits it to its peripheral nodes.

The destination node uses the reversed accumulated sequence of nodes IDs to send a route reply back to the source node. The accumulated ID sequence is likened to source routing techniques in terrestrial networks.

To alleviate, longer route reply packets due to this accumulation, intermediate peripheral nodes may store temporary short routing tables. These contain the IDs of the previous peripheral nodes who have sent the "Query". The intermediate peripheral nodes will also overwrite the ID of the previous peripheral node from which they have received the "Query" before retransmitting the "Query" packet rather than appending its ID to the "Query" packet. Once this intermediate node receives a route "Reply" it will send it to the previous peripheral node whose ID is stored in the short temporary table. Needles to say, even with border casting, there is still some flooding of route "Query" (but less than pure flooding) and the source node of the "Query" may select the route with minimum number of hops.

ZRP is a good example of an Ad-hoc routing technique that tries to strike a balance between proactivity and reactivity. The MAC layer neighbor discovery control overhead (which may not be belittled!) is called association traffic in IEEE 802.11. This overhead, being related to the MAC layer was not accounted for in all of the results above. Broadcasting is seen to decrease the volumes of IARP traffic. However, it was noted that accuracy of the optimal zone radius is a very complex

function of underlying wireless Internet and its too many parameters (mobilities, user density, traffic volumes and types... etc.)

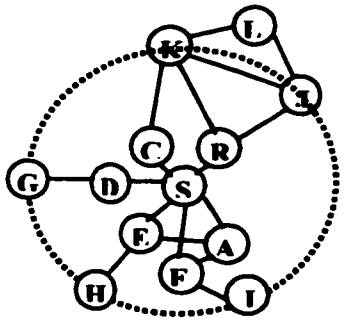


Figure 2- 3: A routing zone of two hops radius

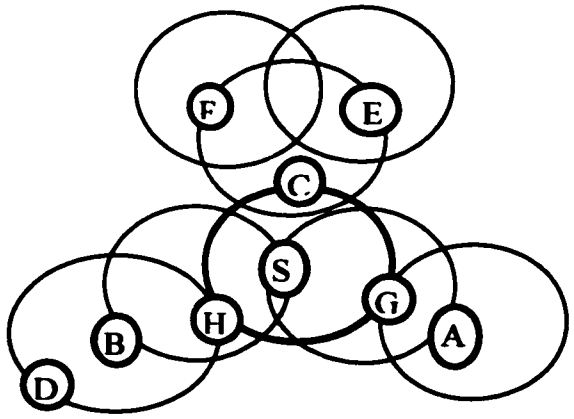


Figure 2- 4: An example of IERP operation

CHAPTER 3

3 DNS BASED ROUTING PROTOCOL

3.1 Objective and Motivation

A variety of ad-hoc routing techniques were presented at last chapter. The ideal or standard ad-hoc routing technique does not exist yet. Pure proactive or pure reactive routing did not provide the efficient nor the dynamic requirement in interconnected Wireless LANs. So, we consider concentrating around hybrid techniques (partly Proactive/partly Reactive). Establishing the best mix of proactive and reactive techniques is a complex function of the traffic amounts and types, and the channel, required QoS, ...etc conditions.

At this circumstance, we introduce a new routing protocol, which is also a hybrid routing technique based on some nodes within network having DNS capability, so that DNS servers anticipate route discovery and also message handling.

3.2 Description of DBRP

As we know, DNS provides a static mapping from names to address. In fact, DNS organizes names and name servers into hierarchies. The hierarchy of DNS servers ensures their uniqueness, and the hierarchy of servers keeps every server from having to know every name.

In DBRP, some nodes within ad hoc networks have DNS capability that means these nodes are fixed and connected with Global IP. Since they have DNS capability, they know each other's IP address within network and also through DNS server to communication with terrestrial Global IP.

3.3 Simulation Procedure of DBRP

3.3.1 DBRP Simulation Assumption

DNS Based Wireless Routing Protocol introduced here is designed for TCP/IP suite; the high level layers could be any protocols amenable to TCP/IP. We assume the data link layer uses IEEE802.11 that utilizes CSMA/CA. The physical layer is presented by the global parameter Pc, which is the probability of successfully receiving a message.

We assume that the channel capacity is 1 MHz. Each packet contains 20000 bits. The transmitting time for one packet is 0.02 second (20ms), which is the programming interval, we define it as iteration. A maximum of 25 nodes can send messages in each programming interval (iteration) in the whole of the networks simulated.

We also assume 200 nodes in an area of 20km*20km parameters. 30 of 200 nodes have fixed positions. Within those fixed nodes, some have DNS capability. Others move randomly with a speed that is selected from 0 a to maximum moving speed, but keep the same speed for the whole simulation period. In our simulation, the maximum moving speed is 108 Km/hour, meaning, is 0.6cm/0.02sec or 0 .6cm/program iteration.

According to each node's destination, we classify two classes of nodes. The destination of Class 1 is within the local network; the destination of Class 2 is terrestrial Global IP, which means the data message of those nodes will not hop

wireless too much, it will go to nearest DNS node and then get handed off to the global web.

Each node changes its direction every-400-iteration (8 sec) within the whole simulation. There are 4 possibilities of moving direction, which are North, South, East and West, the moving direction of each node depends on the outcome of a random variant uniformly distributed between (0,1). As described by Figure 3- 1, we call a uniform distributed function, if the output is within 0-0.25, the node will move east; if the output is within 0.25-0.5, the node will move west; if the output is within 0.5-0.75, the node will move south; if the output is within 0.75-1, it will move North.

If the node reaches the simulation boundary, it keeps its current position till it changes direction.

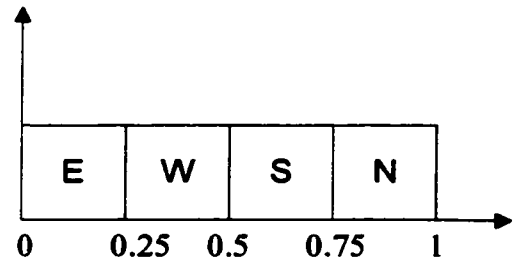


Figure 3- 1: Direction-selected method

Reachable set of each node is defined as the area within a radius of 2km. We also assume that whenever a node moves into the other node's reachable site, they will hear each other (within direct hearing distance).

3.3.2 Input and Output Parameters

3.3.2.1 Input Parameters

There are two kinds of input parameters, one is fixed constants, and the other is input variables. For the fixed constants, they are unchangeable within the whole simulation; for the variables, by assigning them different values, we can achieve different performance.

The fixed constants are:

- Bound_Size: the whole area in which nodes can move. This is equal to 20000m*20000m.
- Reachable_Radius: when the node sends one message, only the neighboring nodes within this radius can receive it. In other words, the node only reaches a certain reachable area. In this project, the reachable radius is defined as 2000m.
- Max_speed: nodes randomly select a certain speed. Here we define the Maximum speed as 0.6m/program iteration, i.e. equivalent to 108Km/hour.
- Population: the number of nodes in the network, set to 200, among these, node number 30 to 69, whose destination are terrestrial Global IP, the others' destination are within network.

The input variables are:

- Max_BufferSize: each node maintains a buffer that can store Max_BufferSize .
- Max_TemporaryBuffersize: Data messages will be stored in a temporary buffer till the node finds the route. The temporary buffer is a part of the total buffer. The temporary buffer may store Max_TemporaryBufferSize Data messages. The reason we defined the temporary buffer is simulation convenience. In the buffer of the node, the forwarded messages (of same node)

and generated messages (of other node) are queued waiting to be serviced according to the FIFO policy. To avoid queuing of unknown-route Data messages in the main buffer, we set the temporary buffer.

- Max_RouteListTableSize: each node establishes a route list table to record the discovery route information that passes through. The route list table can record MAX_RouteListTalbeSize entities.
- Max_FDBTableSize: each node establishes an FDB table to record the neighbouring nodes, to which it can forward messages.
- Max_ReceivedMessageSequenceListSize: when the node receives one message, it records its identifier. This is the maximum receiving message identifiers the node can record.
- Lambda: This is the input traffic parameter; changing Lambda corresponds to different loads. Lambda ranges from 0.0125 to 0.125.
- Max_QueryHop: Query messages are flooded; this is the maximum number of hops that a query message can be flooded.
- QueryWaitingTime: when the node sends the Query message, it will wait a certain time to receive the response. When it doesn't receive response, it may send another query message with a different hop count.
- Max_RouteListTableTimeOut: when the node receives one message, it will record or update the corresponding route information in its route list table. After a certain time, this information will expire.
- Max_FDBTableTimeOut: when the node receives a message, it will also update its FDB Table. After a certain time, this information will expire.

- tTL (time to live): when a node generates a data message, it will be stored into the temporary buffer and awaits transmission. After a certain time, if the message hasn't reached its destination before its tTL, then a certain indication is set. This is an important input parameter.
- ControlMessage_tTL: like the data message, the control message also has a limited value of TTL.
- P_c: This is defined as the probability of message successful reception. If the message succeeds, all neighbour nodes will hear it. It is set to be 0.5, 0.6,0.7, 0.8,0.9 and 0.99 in simulation.
- DNSNumber: Within the wireless, there are some nodes, which have DNS capability. This is an important input parameter which indicates the number of DNS nodes.
- GlobalDNSTransmissionTime: Average Time for DNS nodes to successfully transmit a data message to a destination within the terrestrial Global IP.
- WirelessDNSTransmissionTime: Average Time for DNS nodes to successfully transmit a data message to a destination within Wireless LAN.

3.3.2.2 Output Parameters (Performance Criteria)

There are many parameters to be considered. Changing any of them will affect the final performance. The following performance criteria are defined in order to evaluate the performance. Each performance criteria are averaged over all program iterations and over the number of nodes.

1. Average and Variance of Total Transfer Delay

Total Transfer Delay is defined as the total time that one message takes from its generation to its successful delivery to the final destination. This includes the time it waits in the various buffers (Queuing Delay) from the source to the destination as well as the various Propagation Delays. The Mean of Total Transfer delay is calculated by dividing the sum of all total transfer delays of all successful messages of all nodes by the number of these successfully transmitted messages. The Variance of Total Transfer Delay is calculated by dividing the sum of the squares of the differences between the Average Transfer Delay and individual Transfer Delay by the total number of successfully transmitted messages. The formula of Average total Transfer Delay (ATTD) and Variance of Total Transfer Delay (VTTD) are listed as follows:

$$ATTD = \frac{\sum_{i=1}^{n} TTD_i}{n-1}$$

$$VTTD = \frac{\sum_{i=1}^{n} (ATTD - TTD_i)^2}{n-1}$$

Where n is total number of terrestrial successfully transmitted messages during the whole simulation program, TTD_i represents the Transfer Delay of each successfully transmitted message. In our simulation, this equals the difference between final delivery iteration number and generation iteration number. These two numbers represent the generation time and final arrival time of each message respectively.

2. Average and Variance of Queuing Delay

Queuing Delay is defined as the time between one message generation and its actual transmission from a certain buffer. Average queuing delay can also be represented by the sum of buffer sizes in all program iterations of all nodes divided by the number of nodes and number of iterations. The variance equals the sum of squares of the differences between average queuing delay and individual queuing delay over all program iterations and all nodes divided by the total number of iterations and nodes. The formulas for the Average Queuing Delay (AQD) and Variance of Queuing Delay (VQD) are listed below:

$$AQD = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} QD_{ij}}{(m-1)*(n-1)}$$

$$VQD = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} (AQD - QD_{ij})^2}{(m-1)*(n-1)}$$

Where m is the number of program iterations and n is the total numbers of nodes, QD_{ij} represents the buffer size for the j^{th} node in i^{th} iteration. In our project, n equals 200 and m equals 6000.

3. Average and Variance of Buffer Overflow

Each node has a buffer, once a message is generated or received from his neighbouring node; the message is inserted into the buffer and queued for transmission. If the number of queued messages exceeds the maximum buffer size, either the generated or the received messages will be lost. This is buffer overflow. We set a counter to record the number of instances of buffer overflow. The mean of the buffer overflow is calculated by dividing the sum of these overflows by the total

simulation iterations and by the number of nodes. Variance of the buffer overflow is calculated by dividing the sum of all squares of the differences between the average buffer overflow and each buffer overflow by the total simulation time (number of iterations) and by the number of nodes. The following are the formulas for Average Buffer Overflow (ABO) and Variance of Buffer Overflow (VBO).

$$ABO = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} BO_{ij}}{(m-1)*(n-1)}$$

$$VBO = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} (ABO - BO_{ij})^2}{(m-1)*(n-1)}$$

Where BO_{ij} is the buffer overflow number in i^{th} iteration and j^{th} node, (equals 1 if buffer overflow takes place and "0" if there is no buffer overflow) m is the number of iteration and n is the number of nodes.

4. Overhead ratio

Once the node generates a data message, it needs to first find the route before sending the data message. In order to find the route, many control messages (Hello, Query, Response message as will follow) are generated. Overhead is defined as the number of control messages generated by all nodes divided by the all messages including data messages and control messages generated by all nodes in the whole simulation time, i.e.

$$Overhead\ ratio = \frac{\sum_{i=1}^{n}(hellomessage + querymessage + responsemessage)}{\sum_{i=1}^{n}(hellomessage + querymessage + responsemessage + datamessage)}$$

Where n is the number of iterations.

5. Average Throughput

The average throughput is defined as the ratio of the number of packets that are successfully transmitted in a very long interval to the maximum number of packets that could have been transmitted with continued transmission on the channel [9]. By simulation model, we write the average throughput as:

$$AT = \frac{\sum_{i=1}^{n} successfuldatamessage}{\sum_{i=1}^{n} totaldatamessage}$$

Where n is the number of iterations.

6. Average and Variance of Latency for First Route Reply

This protocol attempts to discover a route to a destination when it is presented with a packet for forwarding to that destination. This discovery must be completed before the information data message can be sent, which adds to the latency of delivering the packet. The Latency of first route reply can be considered as the time between generating the first data message and receiving the first route reply message. We calculate average and variance latency for first route reply as:

$$AL = \frac{\sum_{i=1}^{n} L_i}{n-1}$$

$$VL = \frac{\sum_{i=1}^{n} (AL - L_i)^2}{n-1}$$

Where L_i is the latency in i^{th} node, and n is the number of nodes.

3.3.3 DBRP simulation data structure

Message data structure:

In the simulation, each message contains 11 fields (all decimal). If the message does not use specified field, the field is set to –1.

- Message Type (1~4): there are 4 kinds of messages, which are Hello (1), Query (2), Response (3) and Data (4).
- Identifier: each message has a unique identifier (0~149991994). All identifiers have the same format: iteration*10000+SourceIP*10+Message Type. The right bit is used to indicate the message type, the second right bit to the fourth right bit are used to indicate the source IP, the other bit is used to indicate the iteration.
- Source IP (0~199): In hello, query, response and data Messages, the source name is the user who generates the message originally.
- Destination IP (0~200): Query, Response and Data messages contain this field to indicate destination. The value 200 means the destination is Terrestrial Global IP.
- Intermediate IP (0~199): Query, Response and Data messages contain this field to indicate who is sending the message in this iteration.

- Next IP (0~199): Response and Data messages contain this field to indicate who receives the message.
- Hop Counter (0~ 20): each message contains this field to indicate how many hops the message has traveled so far.
- Begin Iteration (0~14999): each message contains this field to indicate when the message is generated.
- End Iteration (0~14999): Query, Response and Data messages contain this field to indicate when the message reached the destination.
- Query Hop Counter: This field relates to query message. It will decrement by 1 after each hop, when the value reaches 0, the query message will not be transmitted.
- DNSID: Response messages contain this field to indicate which DNS node has given this DNS response and Data messages use it to indicate which DNS node will handle the DNS request.

Node data structure:

Each node contains the following parameters (all in decimal):

- Node IP (0~199): each node has a unique node IP.
- Position (X, Y): We use coordinate point (float Point_X, float Point_Y).
- Call Destination (0~199): Node 0-29 and 70-200 whose destination lies within (0-199), user 30-69, whose destination is terrestrial Global IP (200).
- Moving speed (0~0.6cm/iteration): speed is randomly select from 0 to maximum speed. If the node is stationary, then the speed is zero. In the beginning of the

program, each node calls a uniform random function z = random (0,1). The speed equals to z * maximum speed. Each node keeps its speed throughout the whole simulation.

- Moving Direction (0,1,2,3,4): each node calls a uniform random function z = random (0,1) once every 400 iterations. If z<0.25, moving direction equals East (1). If 0.25<a<0.50, moving direction equals West (2). IF 0.5<a<0.75, moving direction equals North (3). If 0.75<a<1.0, moving direction equals South (4). For fixed node, the moving direction equals None (0). Each programming interval, the program will calculate the node position.
- Sending Periodic Hello message iteration (0~99): in the beginning of the program, each node calls a uniform random function z = random (0,1). Sending Hello message iteration equal to z * 100 for this node. This parameter is kept the same for the whole simulation.
- Sending Query iteration: (-1~14999): the field is initialled to –1. Whenever the source sends one Query message, this parameter is set to current iteration.
- Send Query Hop Counter: this field indicates how many hops that Query message is to be flooded.
- Reachable ID List: this is a linked structure that contains node ID in its reachable range. The list is calculated by the program genie every interval.
- FDB table: is updated by receiving every message. This table has the following format:

FDB table 1	timer
FDB table 2	timer
.....	.....
.....	.....
FDB table size	timer

- FDB table size: is current FDB table size.
- Received Message Identifier List: contains message identifiers that the node (node) has received. Whenever one node receives a message, it puts this message identifier at the end of the List. If the List is full, the first message identifier is eliminated. Whenever one node receives one message, it checks the List from the end to the beginning. If the node receives the same message identifier, the node will ignore the new message. This table has the following format:

ReceivedPacketSequenceList 1
ReceivedPacketSequenceList 2
.....
ReceivedPacketSequenceList receivedPacketSequenceListSize

- Received Message Identifier List size: contains current Received Message Identifiers List size.
- Route List Table: this table contains current route discovering information what has passed through the node (See controlling route list table policy). This table has the following format:

Q or R	Source IP	Destination IP	Intermediate IP	DNSID	Timer
...	...	...	...	...	...
...	...	...	...	...	...

- Route List Table size: contains current route list table size.

- Temporary Buffer: is used for unknown route Data massages queued.
- Temporary Buffer size: indicates current number of message in the temporary buffer.
- Buffer: is used for messages queued to be sent.
- Buffer size: indicates current number of message in the buffer.

3.3.4 DBRP Simulation Model

In DBRP simulation, each node maintains a forward database table (FDB) table and a route list table. Whenever a node generates a Data message, it first checks its local tables. If a route exists, it sends the data message to destination. If a route does not exist, it has to discover the route first. The Data messages have to be stored in a temporary buffer until the source knows the route.

Hello messages are used by a node to declare itself to its neighbouring nodes. When a node is busy, Hellos message are unnecessary because the neighbour can use the intermediate IP field in the other messages to update its FDB table. In the first iteration, each node generates one Hello message. In the first fifty iterations, each nodes tries to transmit successfully this generated Hello message. According to the IEEE 802.11 protocol, each node has a random chance to transmit the Hello message, such that no two nodes from same zone will transmit in the same iteration.

As was mentioned before, we have two classes of nodes, Class 1's destination is within the network, and Class 2's is terrestrial Global IP. These two classes of nodes all must discover the route according to their own route discover policy before sending the Data message, whenever they generate a Data message.

The destination or DNS nodes will receive the Query message, and reply by sending a corresponding Response message. When the Response reaches the source, the source will set up the route according to Response Message and begins to put Data messages into buffer.

In the interconnected wireless LANs, the route may be broken because of mobility. When route lost occurs, the program will delete the relevant route, and the source will rediscover the route.

The main flow chart of DBRP simulation program is shown on Figure 3- 2. We first initialize nodes, giving its initial position, its speed and its ID and its destination ID. Every iteration, we calculate new position for each node, and also create new neighbouring nodes for it, every 400 iterations, we change node moving direction; and then, each node will generate message according to generating message policy and put it into the Initial Queue and wait for transmit. We check each node to see if it can send or not and put can-send node into "check". In our simulation, only maximum 25 nodes can transmit simultaneously, so if "check" is full, we put the node into the Next Queue. And then, we get neighbouring nodes of each node in the "check" and let them to receive message according receiving message policy. In the end, we put set the Initial Queue equal to the Next Queue, empty the Next Queue and increment iteration. We keep doing this loop until the iteration reaches Simulation Time.

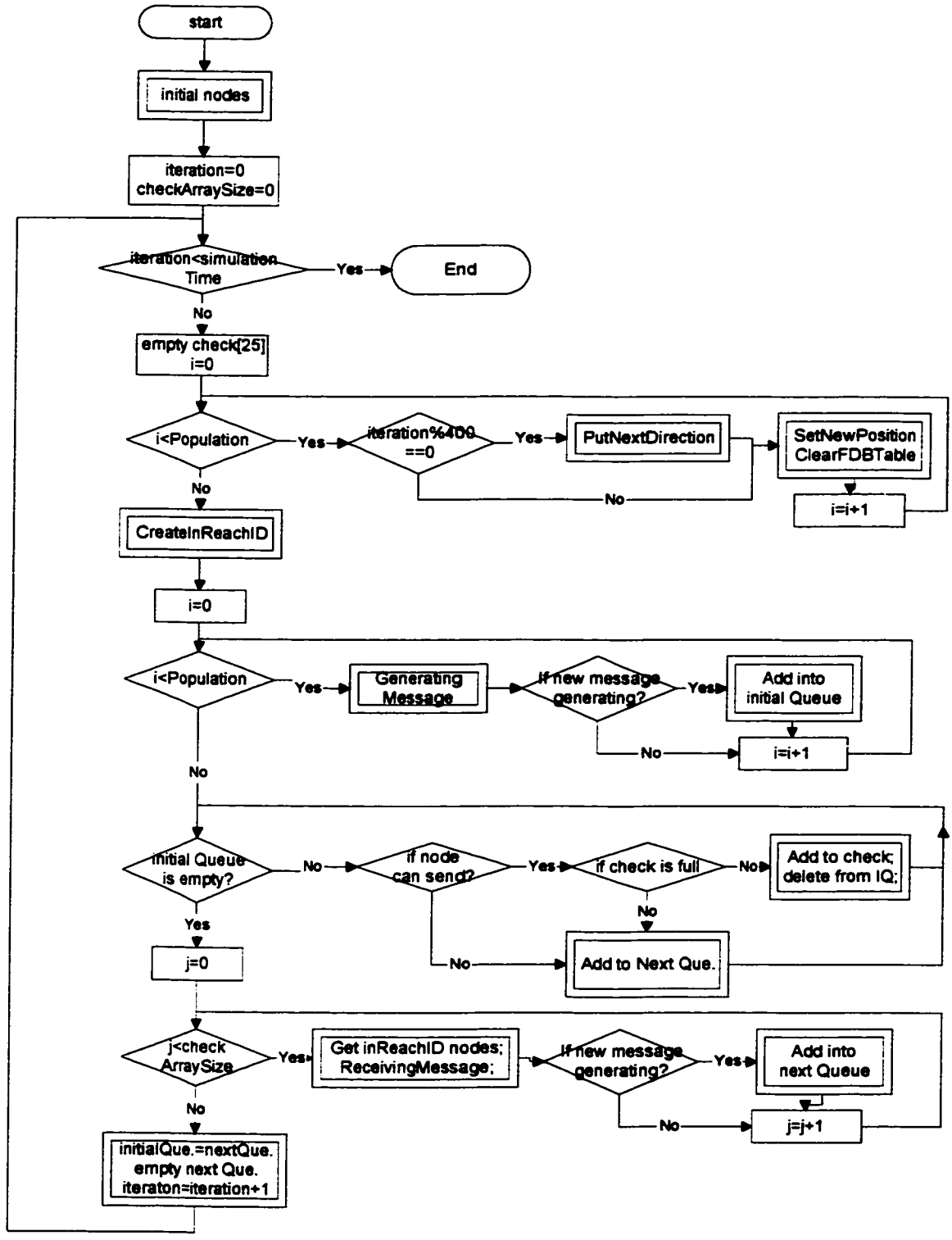


Figure 3- 2:DBRP Simulation Main Flowchart

3.3.4.1 Generating & Sending Message Policy

Each node generates one Hello message periodically. Each node contains a parameter called sending Hello message iteration. This parameter is randomly selected from 0 to 99. Whenever the current iteration number modulus 100 equals the node parameter above, the node generates one Hello message and puts it into buffer.

Each node generates one Data message in one interval according to input traffic λ_μ.

$$\rho = \frac{\lambda_\mu * U}{\mu}$$

Where, ρ is the load of input traffic, U is the number of the total subscribers and μ is the number of messages serviced in the whole network within one iteration. In our simulation, we change ρ from 0.1 to 1 to get different traffic load; U is equal to 200, since there are totally 200 nodes; μ is set to 25, because the total simulation area is 20km*20km, all transmitting nodes should be at least 2km from each other in a certain iteration, therefore, the maximum number of messages serviced in the whole network is 25. So, in our simulation, the range of $\lambda_\mu$is changed from 0.0125 to 0.125.

To generate data message, each node call a uniform variant z, if $z \leq \lambda_\mu$, the node will generate *one* Data message, otherwise, the node will not generate Data message. Whenever the source generates one Data message, it will put it into his temporary buffer.

Generating Hello & Data Message policy flow chart is shown on Figure 3- 3; Put message into buffer flow chart is shown in Figure 3-3,and Put message into temporary buffer flow chart is shown in Figure 3-4.

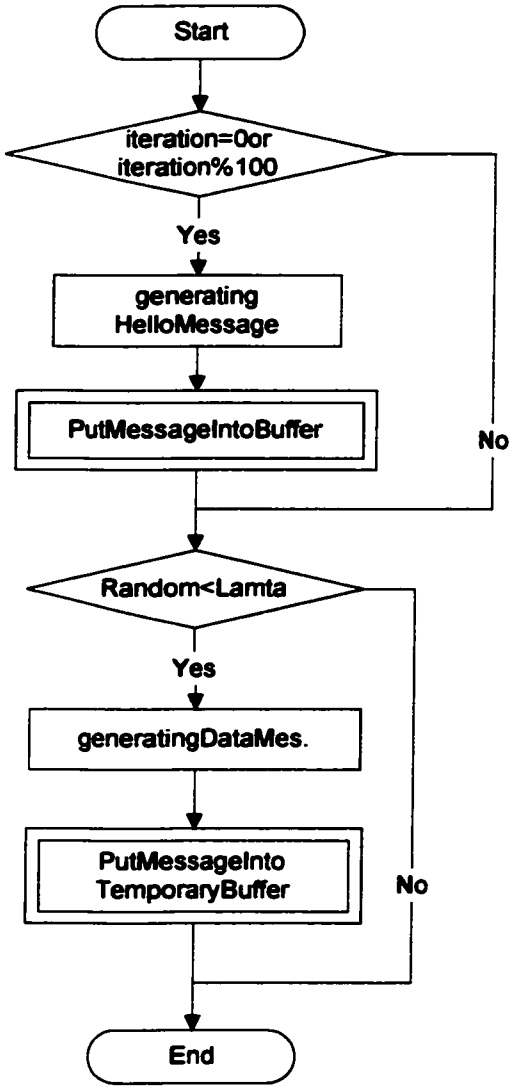


Figure 3- 3: Generating Hello & Data Message Policy

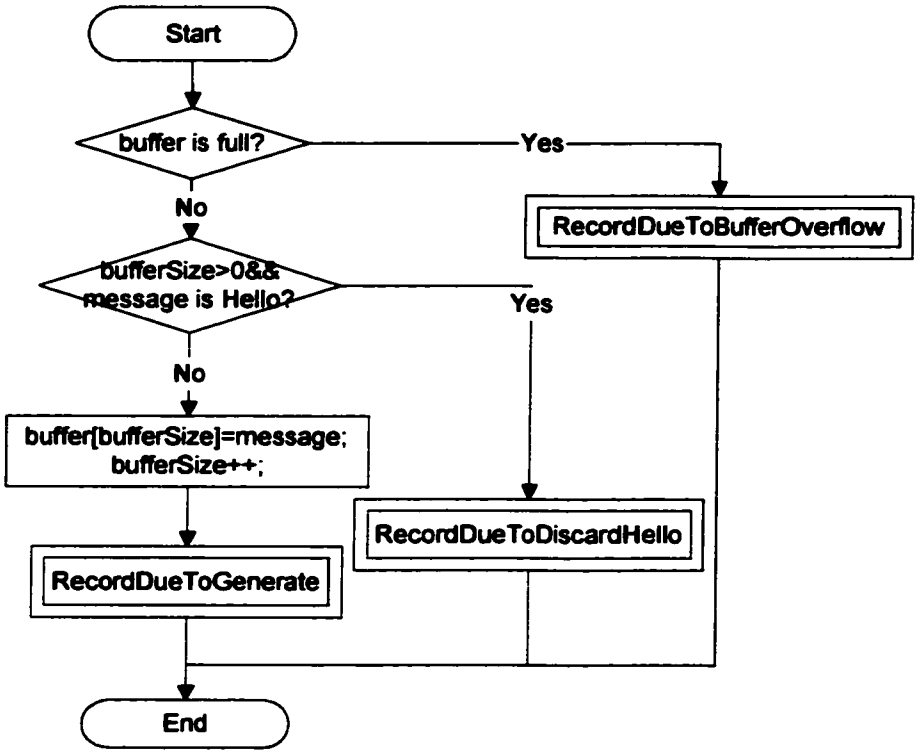


Figure 3- 4: Put Message Into Buffer

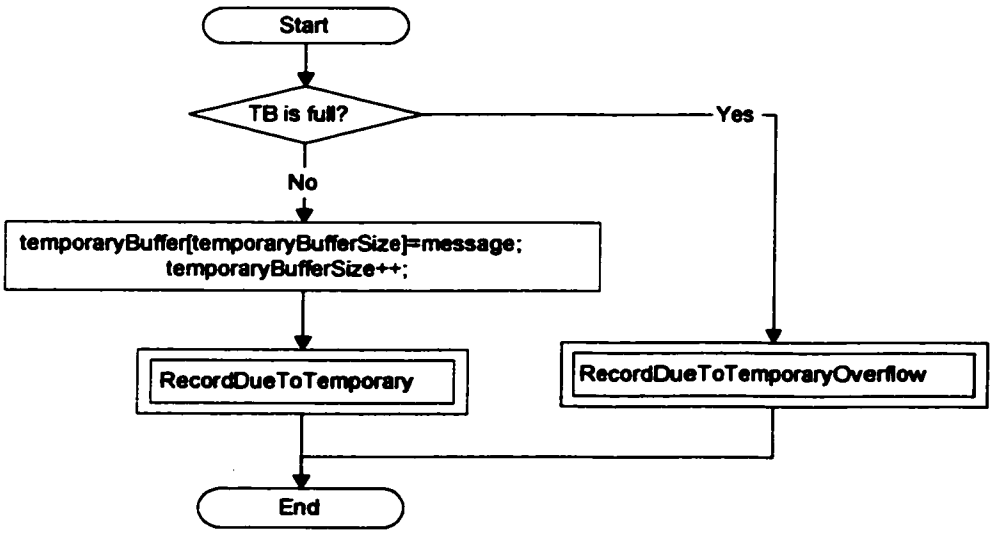


Figure 3- 5: Put Message Into Temporary Buffer

Every iteration, each node checks its temporary buffer, if there are data messages need to send, it will look at its local FDB table and route list table. If the corresponded destination or DNS is not found in the tables, the source generates a Query message. As we mentioned before, there are two classes of nodes. Class 1's destination is within the network and Class 2's destination is terrestrial Global IP. As we see the Figure 3- 6 and Figure 3- 7, the policy of generating Query message is different.

For Class 1, each node checks out its local FDB and route list tables, if no route for destination, it will generate Query message, which hops once, and then waits for a certain time; if there is no destination response, it will generate another Query message which hops twice, and then waits double the time of the first Query message; if there is still no destination response, the source node will try to use DNS node to serve its data message; If there is no destination and DNS route for the source node, it will generate another Query message which hops three and waits triple the time of the first Query message; if there is still no destination or DNS response, the source node will keep generating new Query messages which hop one more than the previous Query message and waits for corresponded time, until it gets response or the Query message hops 10.

Compared with Class 1, the Class 2 nodes will try to find the nearest DNS node. Source node will check its local FDB and route list tables, if there is no DNS route, it will generate a Query message which hops once, and then waits for a certain time; if there is no DNS response, it will generate another Query message which hops

twice, and then waits for double the time of the first Query message; the source node will keep generating Query messages which hop one more than the previous Query message until it finds the DNS node and hands its data message to the DNS node immediately or the Query message hops 10.

Whenever nodes within the whole network receive any Query message, it successfully records this information in its route list table. The destination and DNS node will generate relevant Response message.

Each node sends the messages from its buffer according to a First In First Out policy.

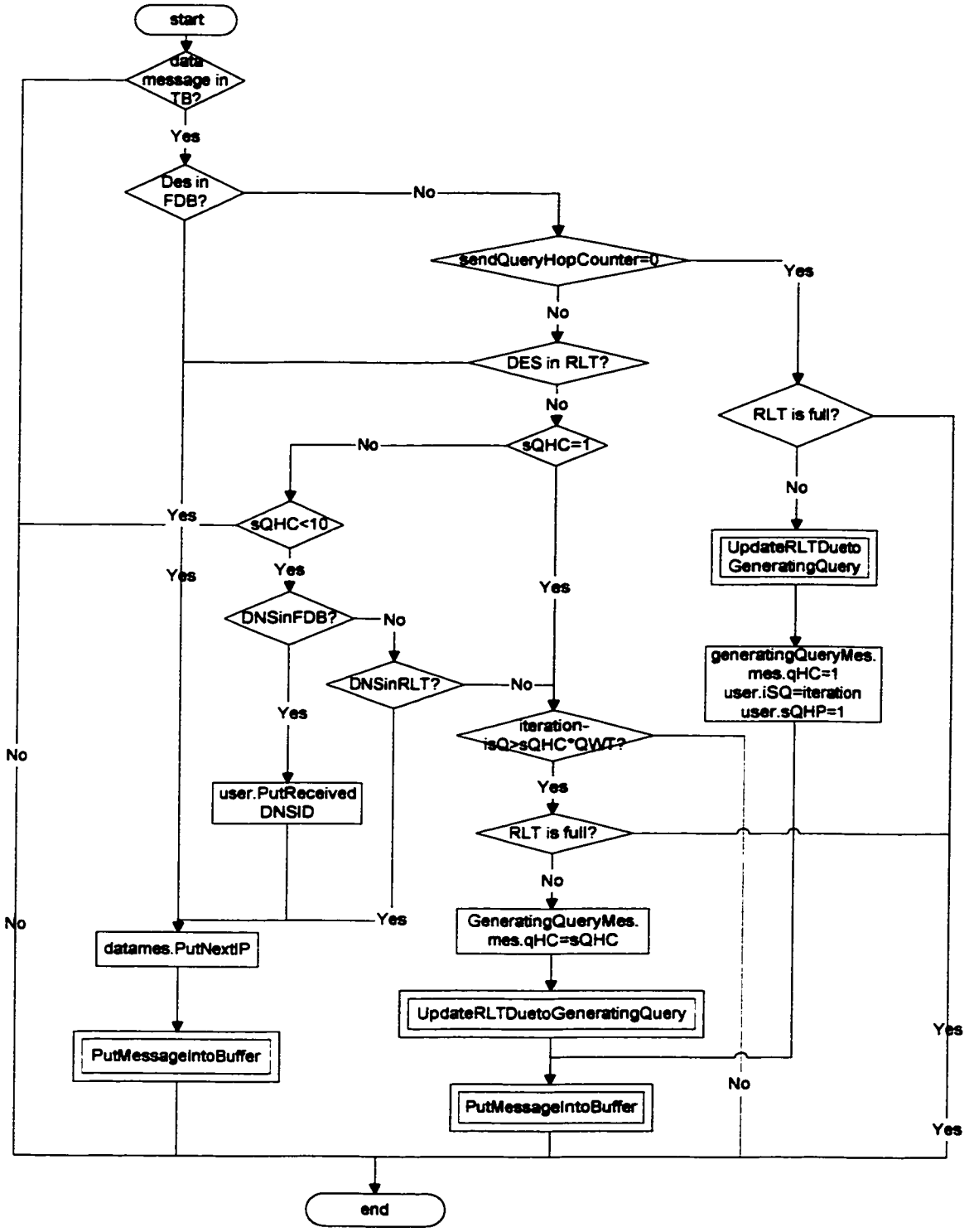


Figure 3- 6: Generating Query Message Policy for Class 1 Nodes

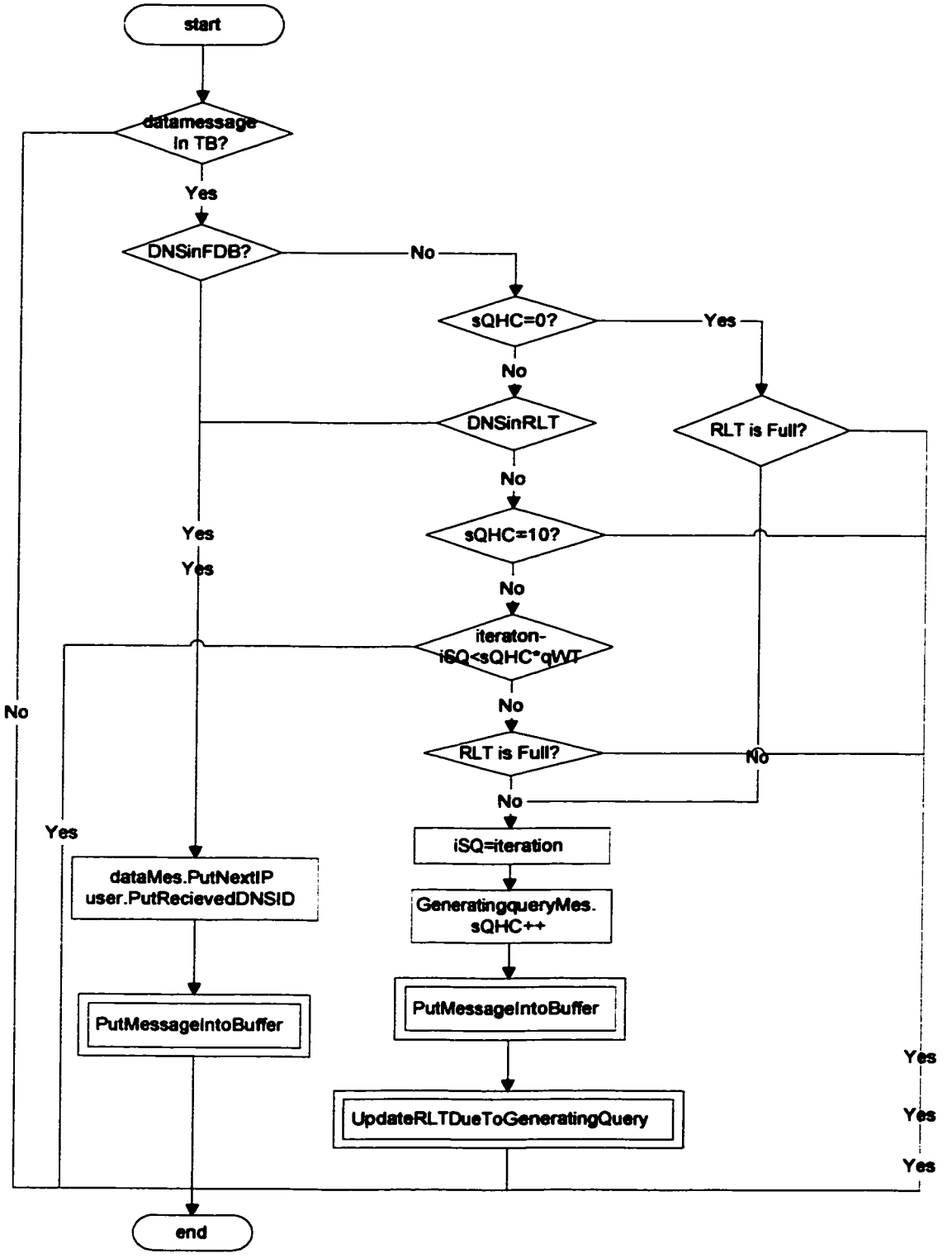


Figure 3- 7: Generating Query Message Policy For Class 2 Nodes

3.3.4.2 Receiving Message Policy

Whenever a node receives one message, it first checks if it has received this message before, if not, it will update its Received Message Sequence List and its FDB table according to the intermediate IP field in the message, As shown in Figure 3- 8.

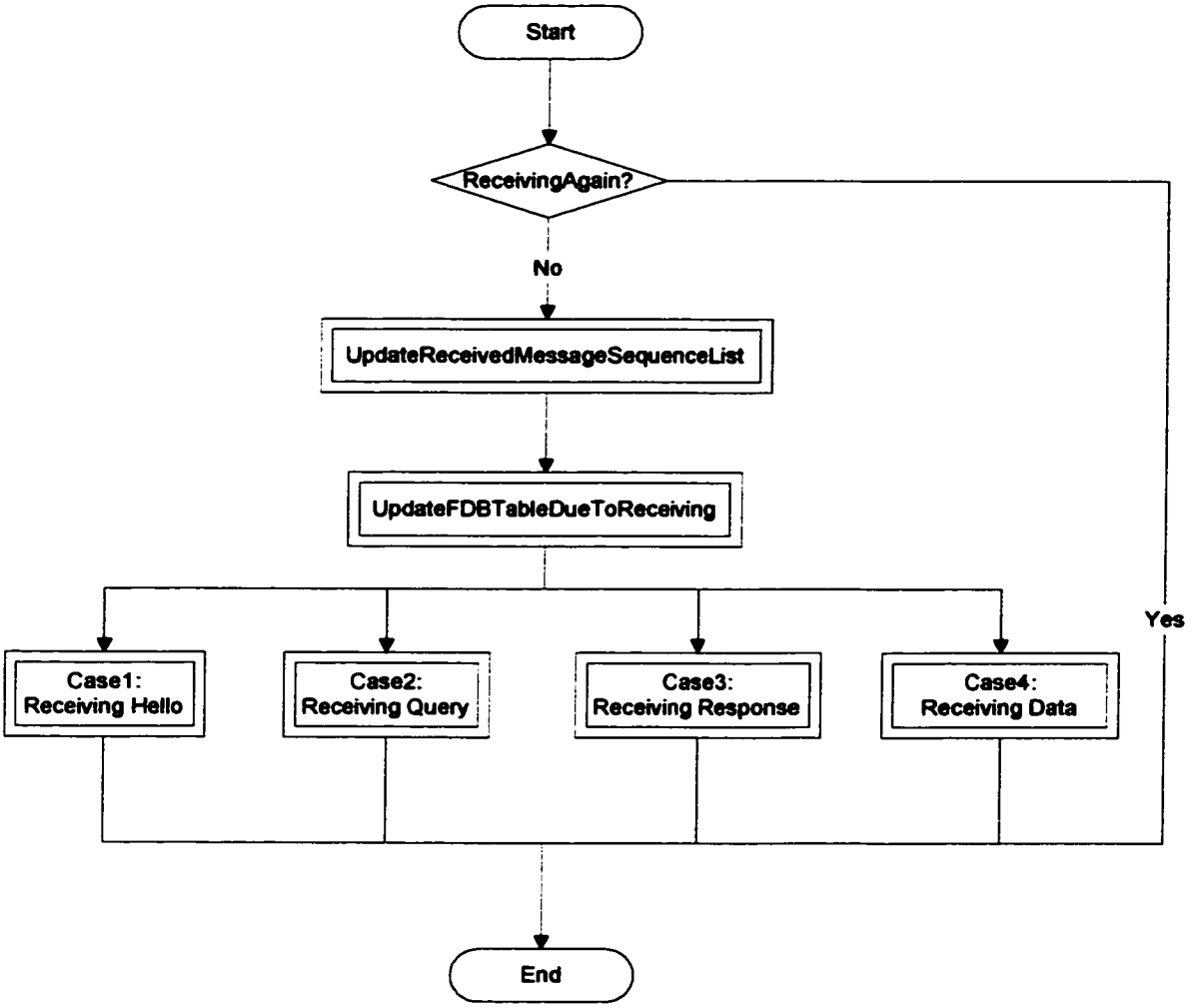


Figure 3- 8: Receiving Message Policy

In case 1: when a node receives a Hello message, it will not do anything, since it has already updated its FDB table.

In case 2: when a node receives a Query Message, it first checks whether its ID equals message's destination IP, if yes, the node records the query information and generates a Response Message regarded to destination; If not, if the node is has DNS capability, it will record the query information and generate a Response Message in regard to DNS, and then, the node whatever it is a DNS node or not will decide whether to flood Query Message or not, according to query hop counter which has been decremented by 1. The flowchart is shown on Figure 3- 9.

In case 3, If the node receives a Response message whose source IP equals this node IP, it will check what kind of Response message it has received, if the message is a destination response, the node records the message and begins to deal with its data message; if the message is a DNS response, the node will check if it has already received DNS Response before, if not, it will record the message. If the node receives one Response message whose next IP equals to this node IP, the node will check its route list table. If the relevant Query information exists, the node will update the Query information as per this Response information and unicast the Response information. The flowchart is shown on Figure 3- 13.If the relevant information does not exist, in other words, the route has been broken, the node will report route lost. The reason Response message cannot reach source is that Query messages took too long time to reach destination; and the Query information in the forwarding node' route list table has expired.

In Case 4: If the node receives one Data message whose next IP equals to this node IP and destination IP does not equal to the node IP, the node will check its route list table. If the relevant Response information exists, the node will update the

Response information, get next IP address for the Data message and forward it to this specific node. If the relevant information does not exist, it will report the rout lost.

If a node receives one Data message whose destination IP equals this node IP, the node will record this Data message.

If Data message's DNSID equals to the node's IP, that means DNS node will deal with this Data message. Since it is very hard to guess how long for DNS node to deal with the route request and forwarding a data message, we just call a Gaussian distribution function to get a certain time (in units of one iteration time) and add it to current number of iteration which the message has hopped so far, so as to calculate the total transfer delay.

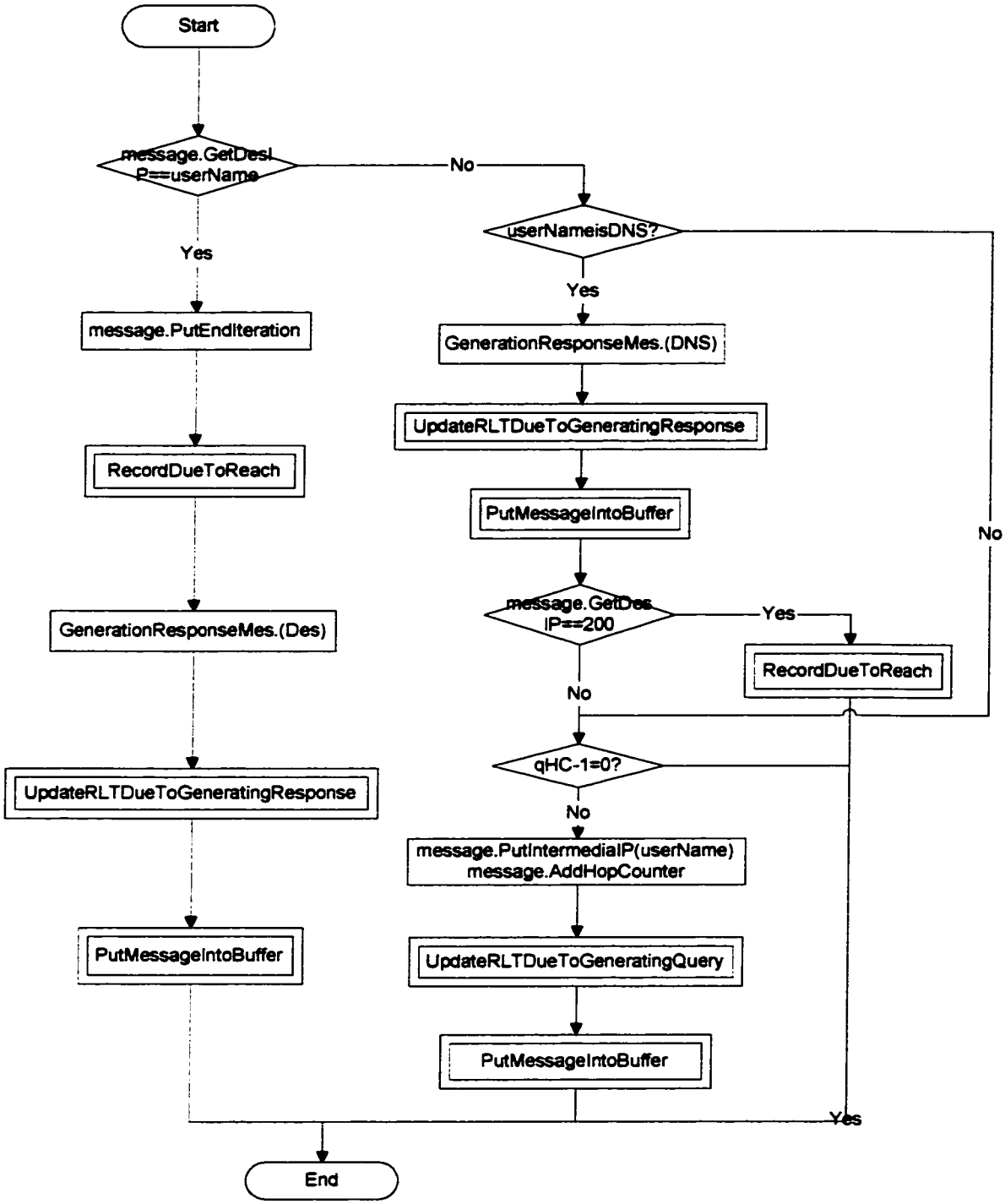


Figure 3- 9: Receiving Query Message Policy

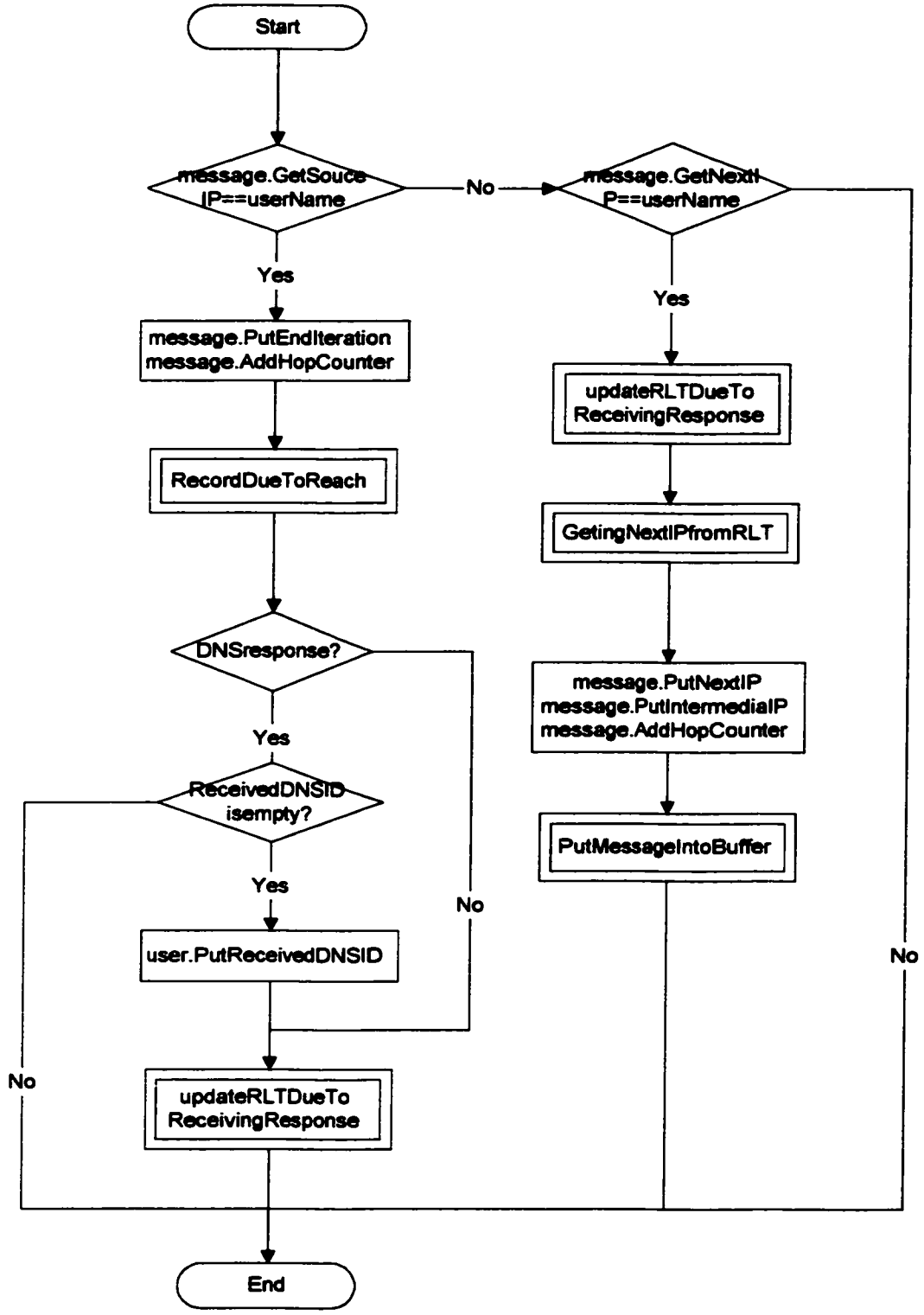


Figure 3- 10: Receiving Response Message Policy

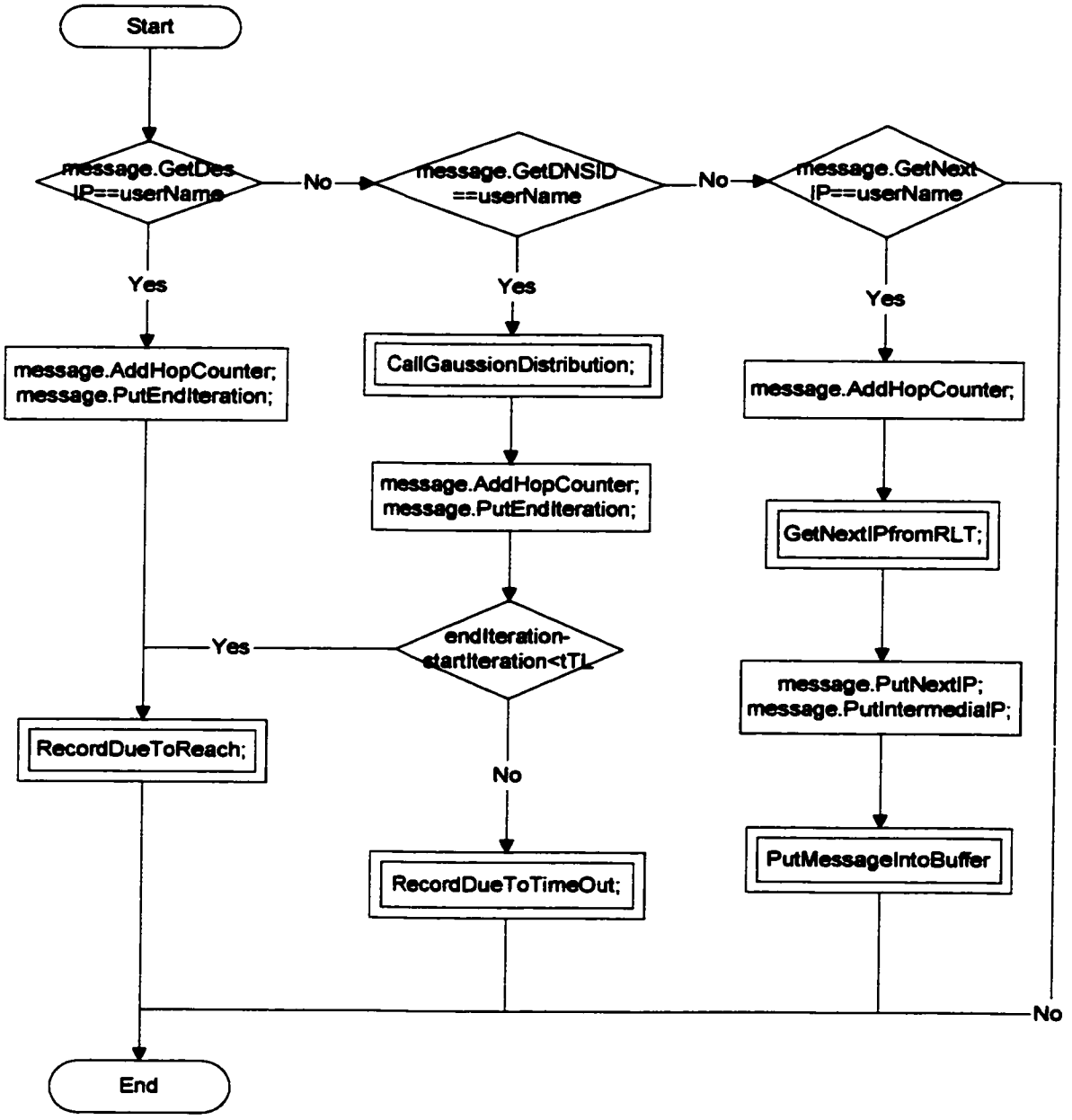


Figure 3- 11: Receiving Data Message Policy

3.3.4.3 Accessing Channel Policy

Each node must apply one sending token if it decides to send one message. In our simulation, whenever, a node has successfully put one message into buffer, it will get one sending token.

According to IEEE802.11, each node sends message with fair distributed queue. We must check who can send and who have to wait every programming interval (iteration).

Whenever one node gets a sending token, we must put the node at the end of a virtual queue (say Big Queue). In each programming interval (iteration), the first node in the Big Queue is the one to send, which is then eliminated from the Big Queue and put into Determined Sending Array. The program also checks other nodes in the Big Queue to avoid a node receiving two messages from two sources in one interval. If the node does not conflict with nodes in the Determined Sending Array, it is eliminated form Big Queue and put it into Determined Sending Array. Otherwise, the program puts the node into temporary virtual queue.

This scheduling continues until the Determined Sending Array is full or all nodes in the Big Queue have been checked. Then the program copies the temporary queue to the end of the Big Queue. Before copying, the Big Queue could be nonempty. When the system achieves maximum capacity, calculating the remaining nodes of the Big Queue is stopped.

Whenever the message is eliminated from the node buffer due to time-to-live, the node must take back its sending token once from the Big Queue.

Accessing channel policy flow chart is shown in Figure 3- 152.

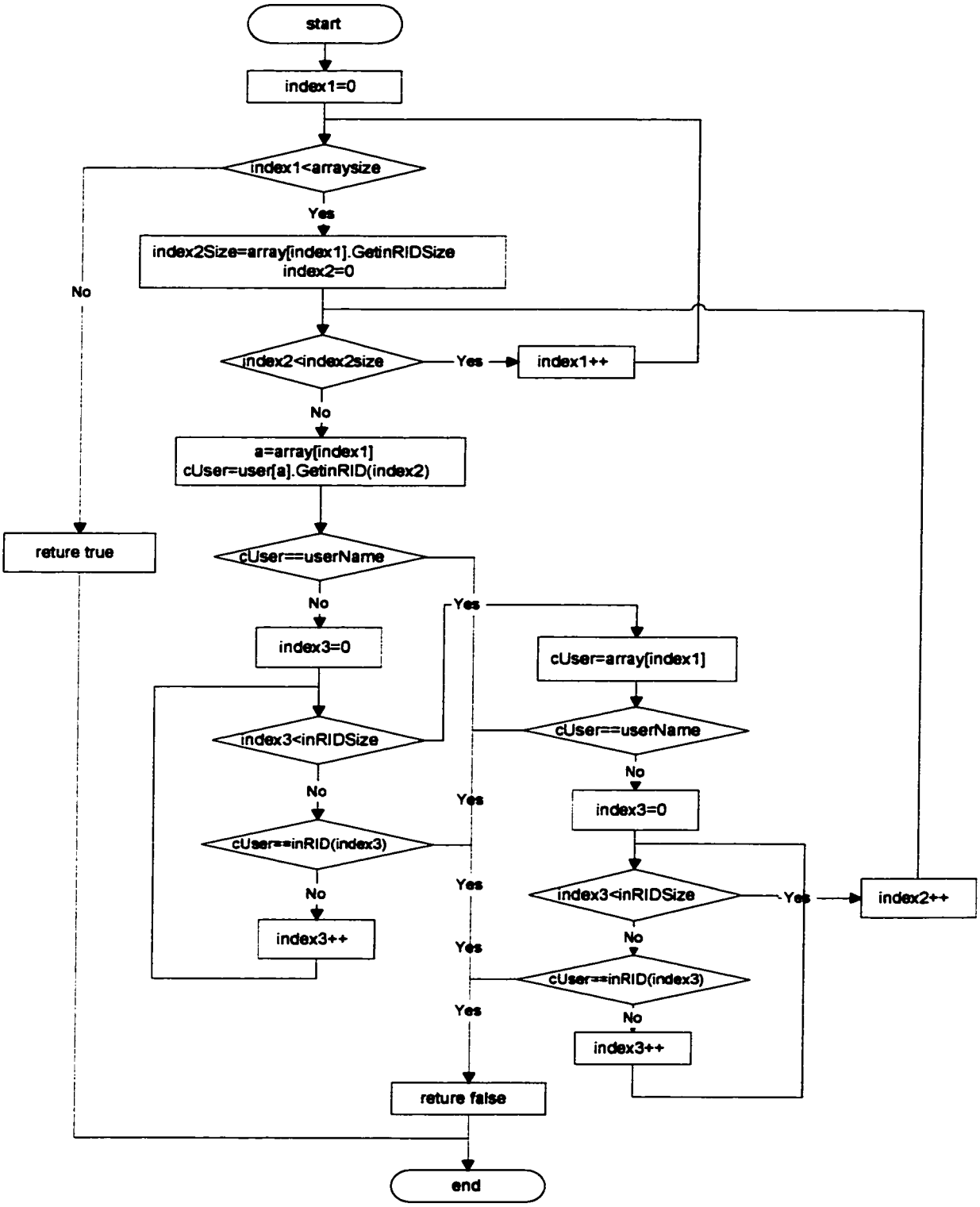


Figure 3- 12: Access Channel Policy

3.3.4.4 Controlling Buffer Policy

In simulation, each node maintains one buffer. The node buffer is divided into two parts, one is called temporary buffer, and the other is called buffer. All messages have the same format, the buffer service them as First In First Out policy.

After the data message is generated and before the route is determined, the Data message is stored in the temporary buffer. When the route is found, the data message is put into buffer and gets sent.

Whenever the message timeouts, it is deleted from the buffer or temporary buffer. Whenever the message is sent, it is eliminated from the buffer.

3.3.4.5 Controlling FDB Table Policy

Each line in the FDB zone table contains two fields. The first is reachable node IP, and the second is timer. Whenever the node receives one message, it updates its FDB table according to the intermediate IP field of the message. The node sets the timer to current iteration, if in a certain time, the node does not receive a message from same intermediate IP before timer expiry, it will delete the relevant information from the table.

Please note again, the Reachable ID list is used for the program to determine which node can send, which can receive and storing the information about the reachable range and reachable nodes. On the other hand, the FDB table is used for routing.

3.4 Simulation Results

There are a lot of variables, by proper setting of which we can get better performance in our routing policy. The following are important variables:

- Buffer Size
- TimetoLive (how long a message can survive in the whole of simulation time)
- DNS nodes number (how many DNS nodes within the whole network)
- Max_Iteration (total time units in the whole simulation time)
- Input Traffic Intensity (Rhou)
- Successful receiving probability of one message (Pc) due to channel impairments.
- Maximum route list table size (the maximum size of routing information of the local table)

We ran the simulation model programs with different values of the input parameters. In the following we compare the performances of the DNS Based Routing Protocol with different input parameters.

3.4.1 Average Total Transfer Delay (ATTD)

As we mentioned above, we define ATTD as the total time that one message takes from its generation to its successful delivery to the final destination. This includes the time it waits in the various buffers (Queuing Delay) from the source to the destination as well as the various Propagation Delays. Class 1 nodes' destination are within the network and Class 2 nodes' are terrestrial Global IP.

As shown in Figure 3- 13 and Figure 3- 15, we fix the buffer size and successful receiving probability (Pc) and vary the input traffic load; we can see the change of the

average total transfer delay (ATTD) parameter. Simulations are run by varying the input traffic from 0.1 to 1.0 and fixed buffer size =20 packets, Timeout =600 iteration, Pc =0.99 and maximum number of iteration =6000. It is seen that when the traffic is low, the average total transfer delay is low, when the input traffic becomes heavier; the average total transfer delay also increases, but the variance doesn't change much (Figure 3- 14 and Figure 3- 16).

Similarly, Keeping the other input parameters at the same values, we can see that the ATTD increases while increasing buffer size (Figure 3- 17and Figure 3- 19).

When changing the value of DNS numbers within the whole network, we see that when the ATTD increases a little bit when DNS numbers<5, and then as the numbers increasing, the ATTD decreases gradually (Figure 3- 21and Figure 3- 23).

When Pc increases, ATTD also decreases, since more messages succeed. ATTD achieves its highest value when Pc=0.6. And then when Pc increases, ATTD begins to decrease, because more messages succeed causes to intermediate node overloading, therefore, the message will take longer time to reach its destination. (Figure 3- 25 and Figure 3- 27).

From 3-29 and 3-31, we know the Class 1 nodes' ATTD decreases a little bit when we increase the simulation time and Class 2 nodes' ATTD haven't been affected.

We can also draw a conclusion that Class 2 nodes' ATTD are much larger than Class 1 nodes', since Class 2 's destination is out of the wireless local LAN.

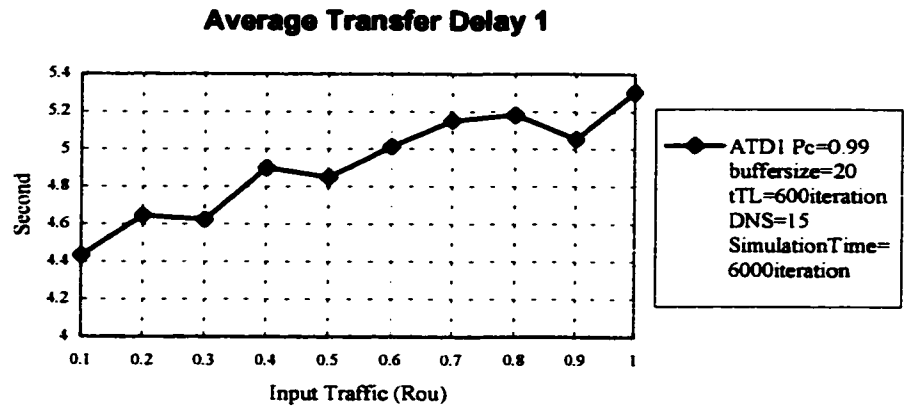


Figure 3- 13: Average Total Transfer Delay 1 vs. Input Traffic

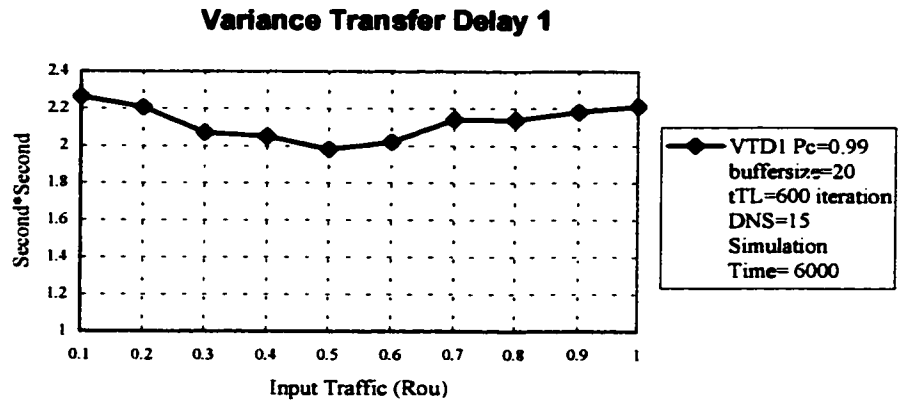


Figure 3- 14: Variance Total Transfer Delay 1 vs. Input Traffic

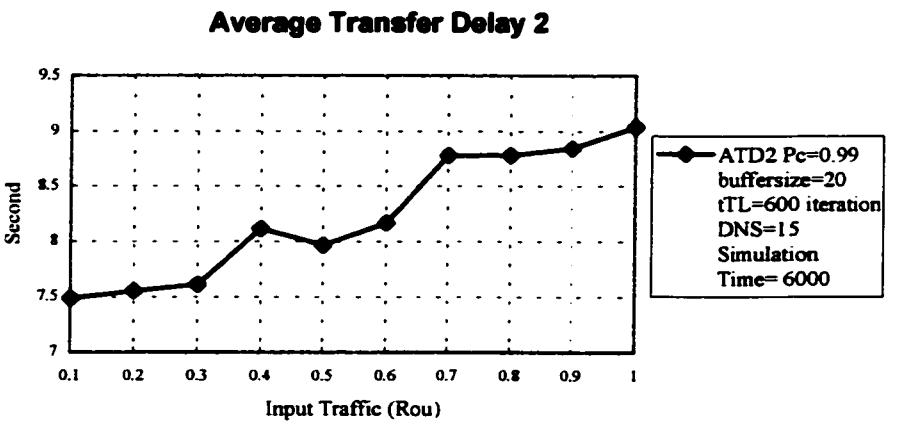


Figure 3- 15: Average Total Transfer Delay 2 vs. Input Traffic

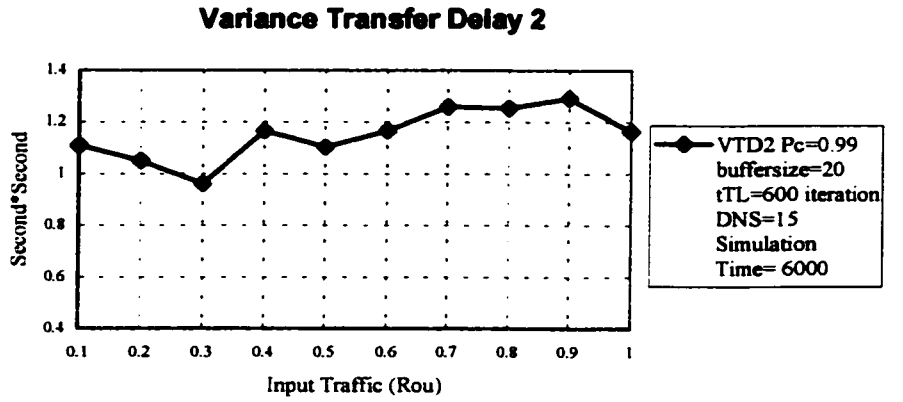


Figure 3- 16: Variance Total Transfer Delay 2 vs. Input Traffic

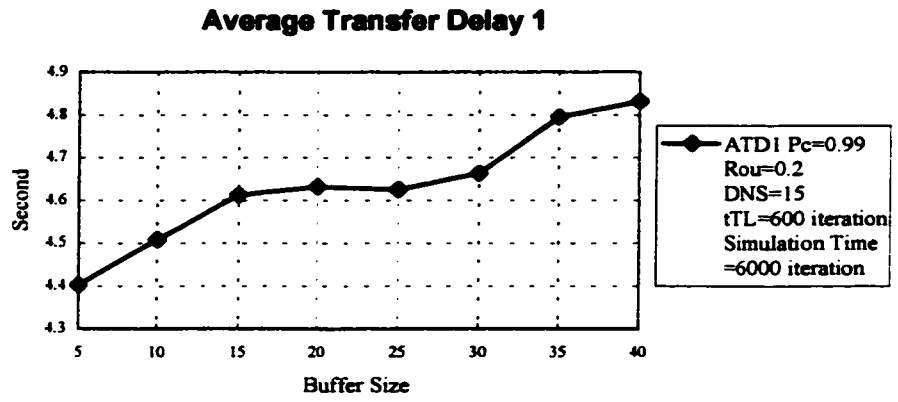


Figure 3- 17: Average Total Transfer Delay1 vs. Buffer Size

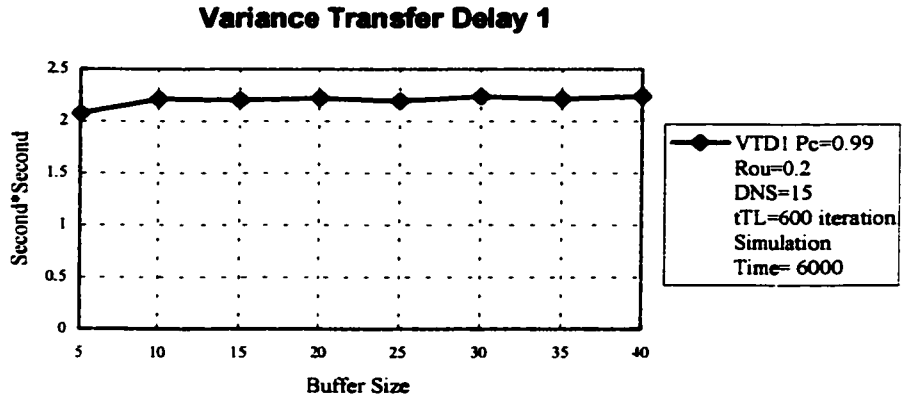


Figure 3- 18: Variance Total Transfer Delay 1 vs. Buffer Size

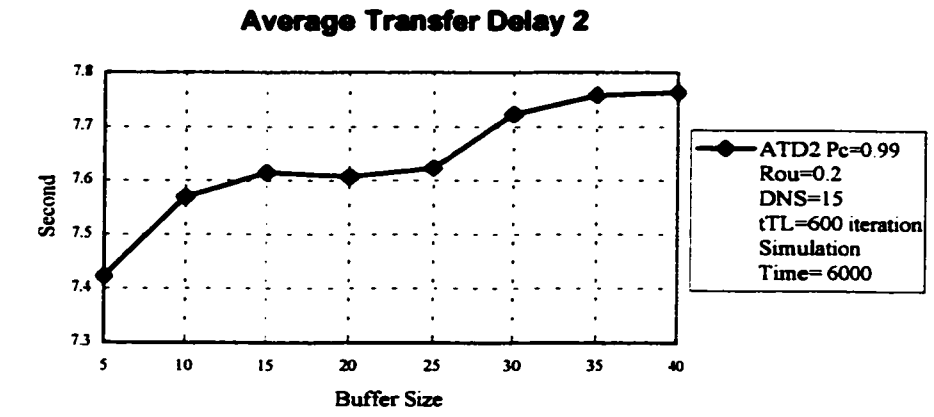


Figure 3- 19: Average Total Transfer Delay 2 vs. Buffer Size

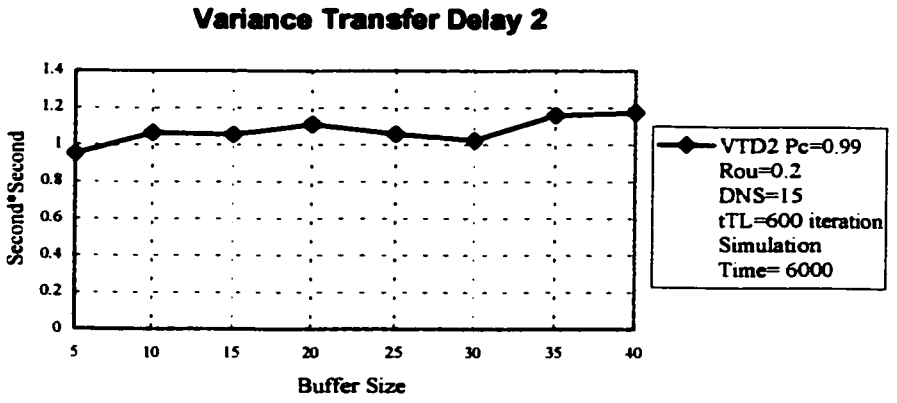


Figure 3- 20: Variance Total Transfer Delay 2 vs. Buffer Size

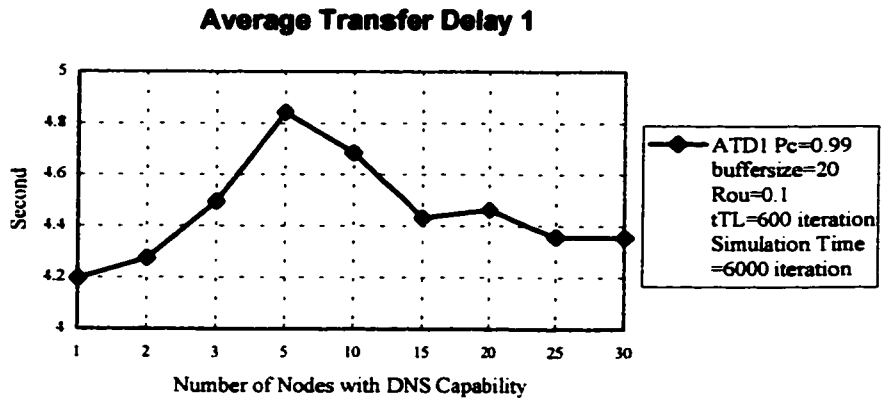


Figure 3- 21: Average Total Transfer Delay 1 vs. DNS numbers

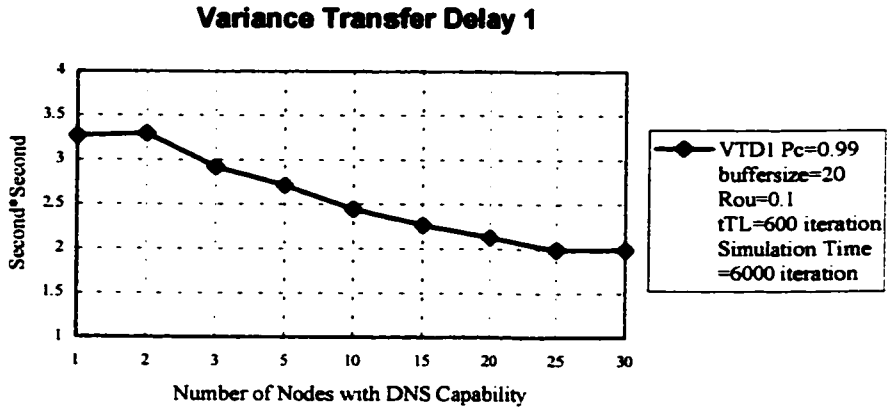


Figure 3- 22: Variance Total Transfer Delay1 vs. DNS numbers

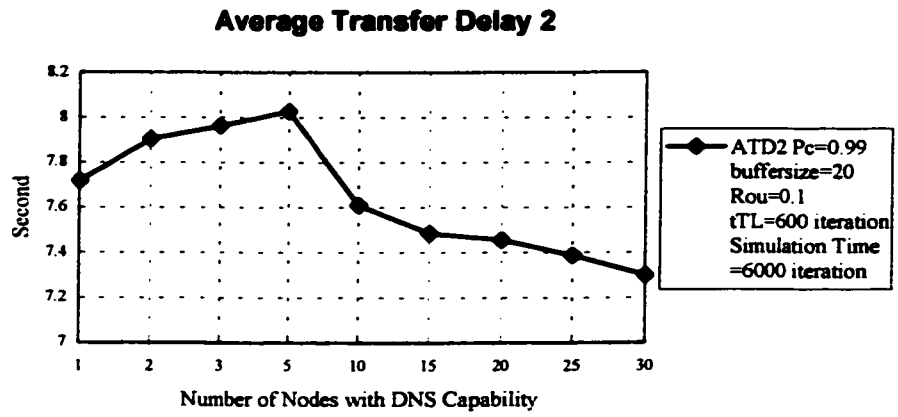


Figure 3- 23: Average Total Transfer Delay 2 vs. DNS numbers

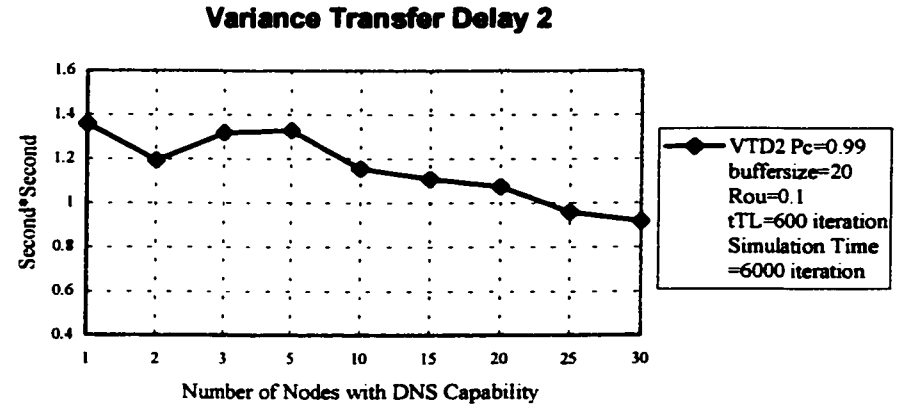


Figure 3- 24: Variance Total Transfer Delay2 vs. DNS numbers

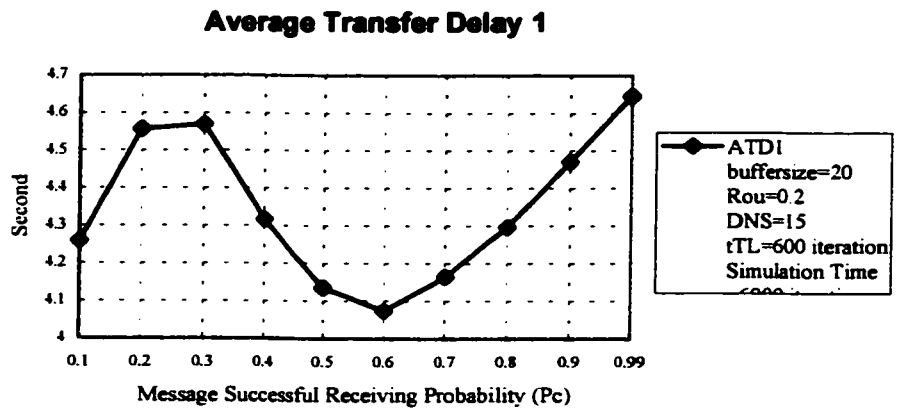


Figure 3- 25: Average Total Transfer Delay 1 vs. Pc

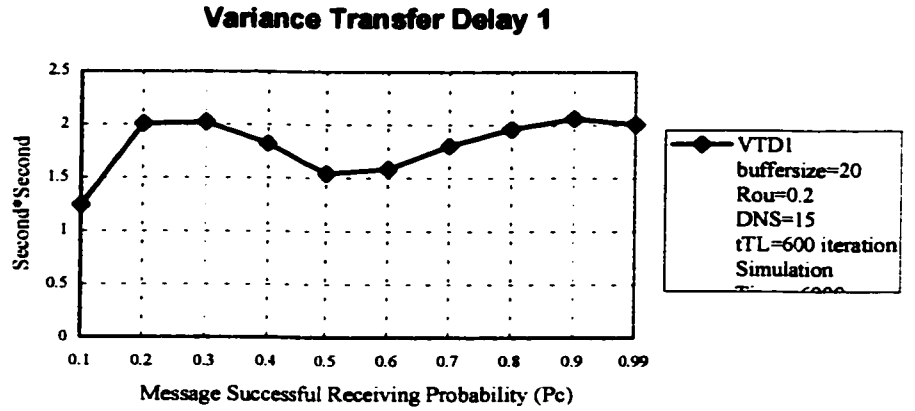


Figure 3- 26: Variance Total Transfer Delay1 vs. Pc

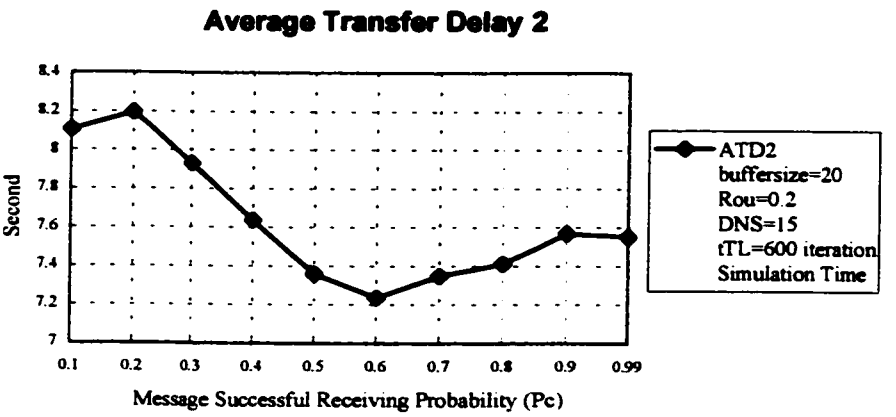


Figure 3- 27: Average Total Transfer Delay 2 vs. Pc

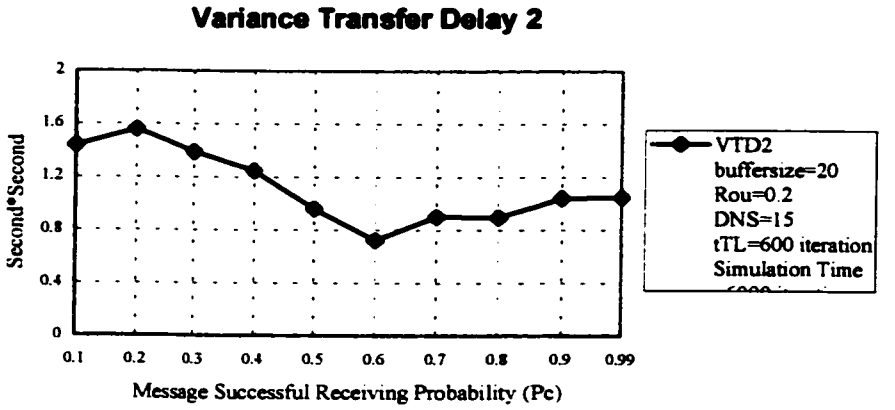


Figure 3- 28: Variance Total Transfer Delay2 vs. Pc

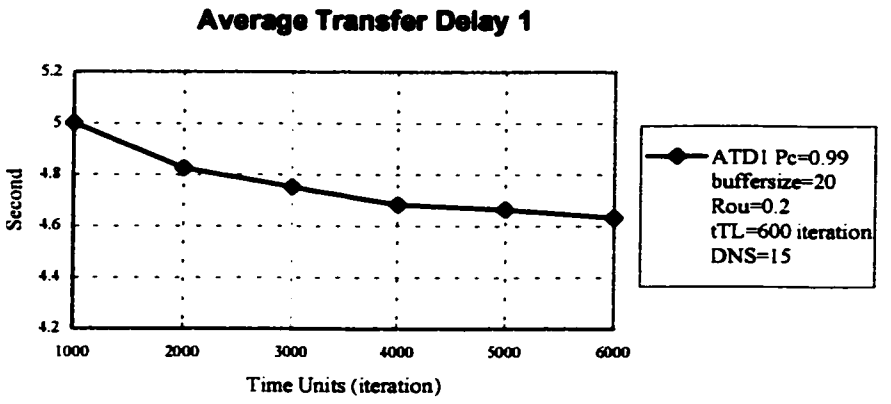


Figure 3- 29: Average Total Transfer Delay1 vs. Time Units

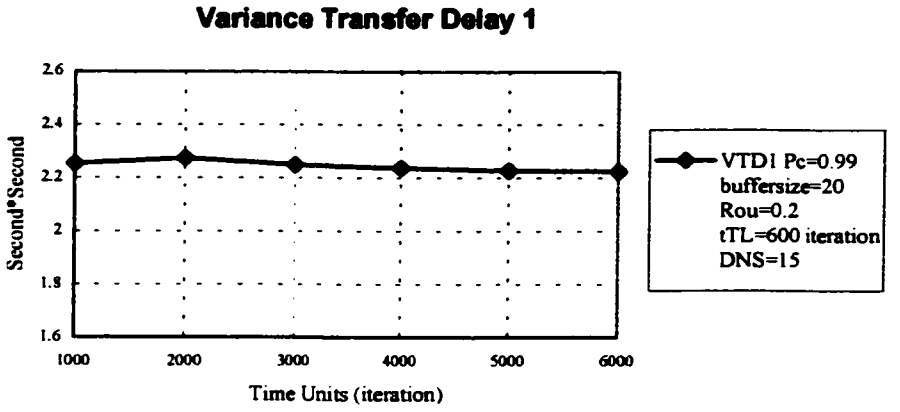


Figure 3- 30: Variance Total Transfer Delay1 vs. Time Units

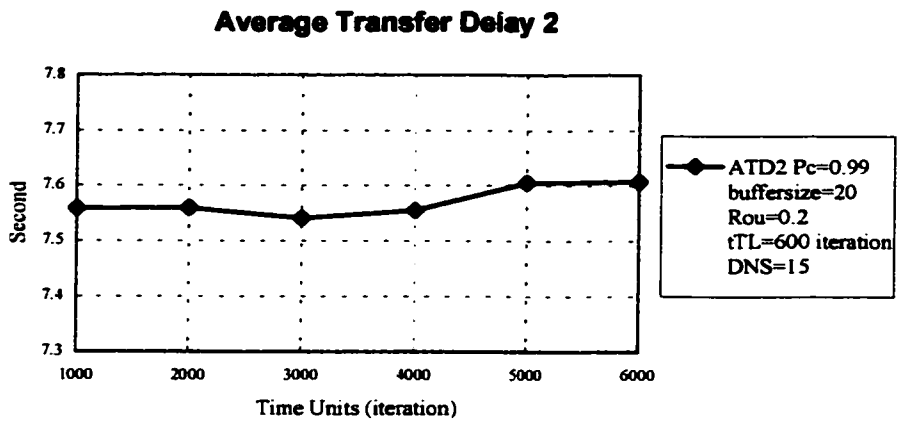


Figure 3- 31: Average Total Transfer Delay2 vs. Time Units

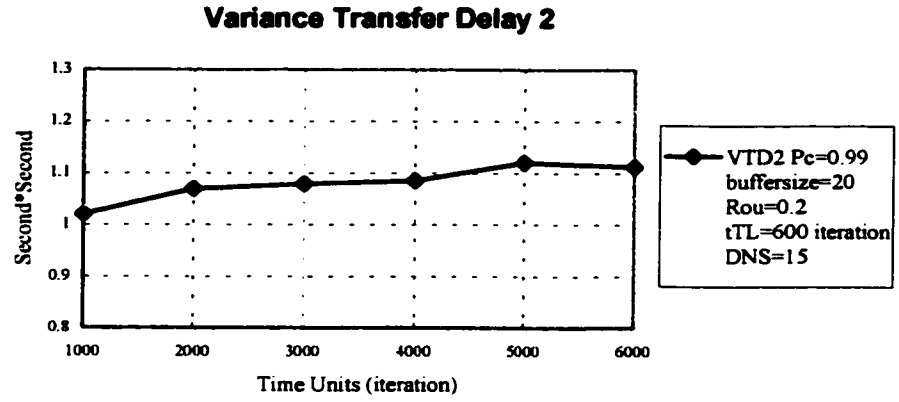


Figure 3- 32: Variance Total Transfer Delay2 vs. Time Units

3.4.2 Average Queuing Delay (AQD)

With lower traffic, a message will take less time to queue in the buffer, and as the traffic rises the queuing delay rises gradually (see Figure 3- 33), but the variance doesn't change.

When the buffer size increases, more messages will be able to queue in the buffer, the average delay and the variance of the queuing delay also increases. This can be verified from the Figure 3- 35 and Figure 3- 36.

Similarly, from Figure 3- 39, one finds that AQD increases with increasing the successful receiving probability (Pc), but will be saturate beyond Pc>0.5. And the AQD is not affected too much, while changing the DNS numbers (Figure 3- 37).

As the simulation time elapses, we find that AQD increases, and then after 3000 iteration, it doesn't change too much (Figure 3- 41 and Figure 3- 42).

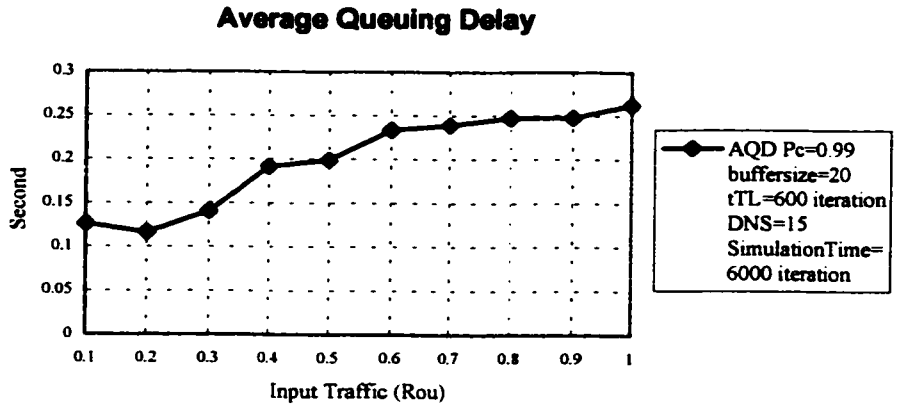


Figure 3- 33:Average Queuing Delay vs. Input Traffic

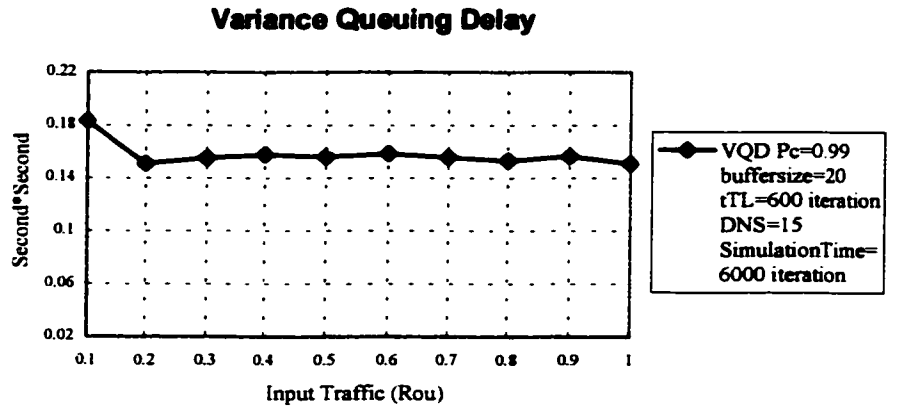


Figure 3- 34: Variance Queuing Delay vs. Input Traffic

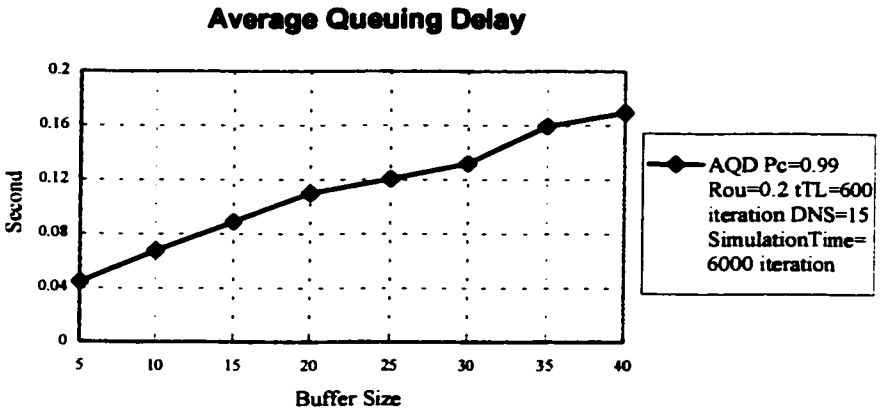


Figure 3- 35: Average Queuing Delay vs. Buffer Size

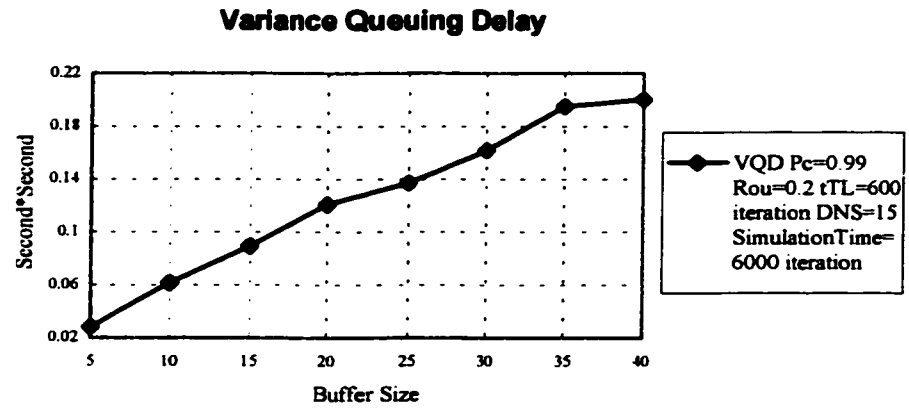


Figure 3- 36: Variance Queuing Delay vs. Buffer Size

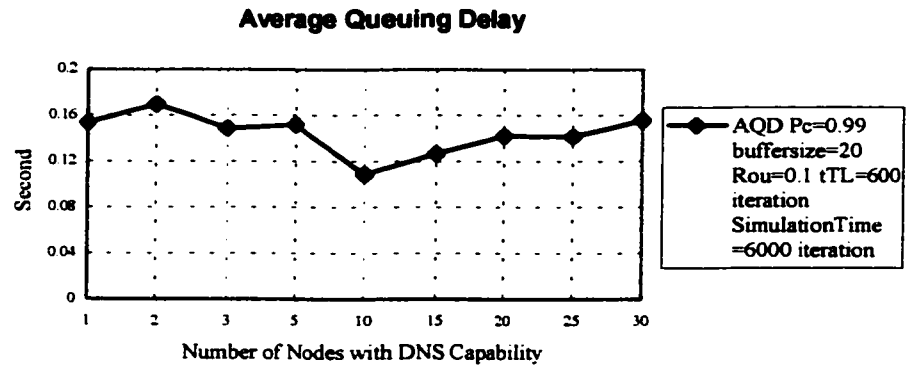


Figure 3- 37: Average Queuing Delay vs. DNS Numbers

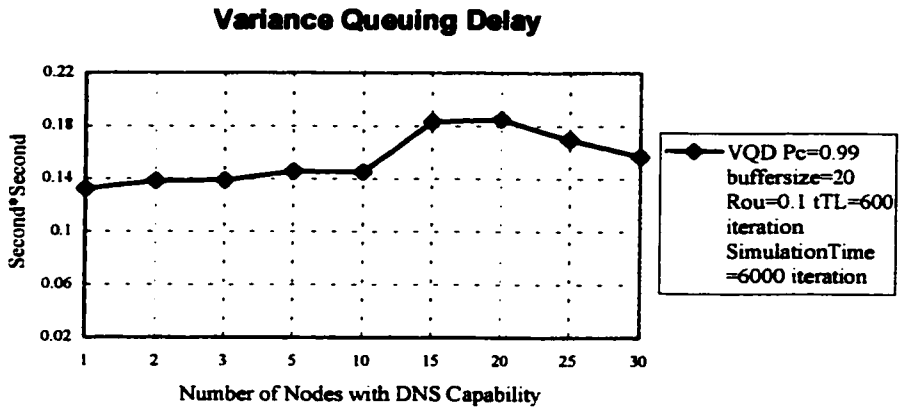


Figure 3- 38: Variance Queuing Delay vs. DNS Numbers

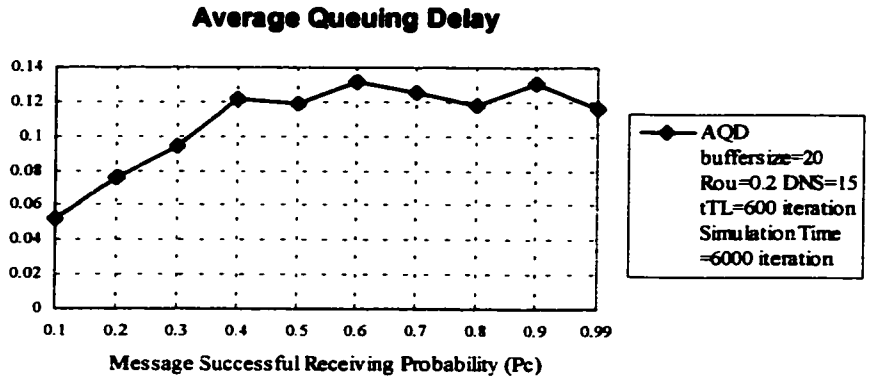


Figure 3- 39: Average Queuing Delay vs. Pc

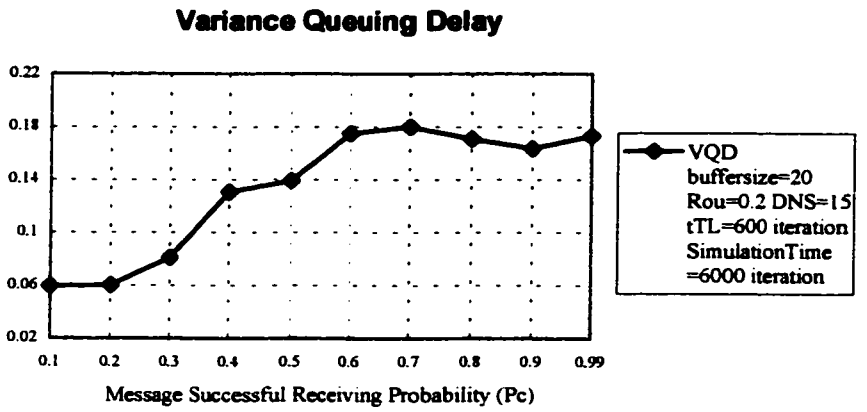


Figure 3- 40: Variance Queuing Delay vs. Pc

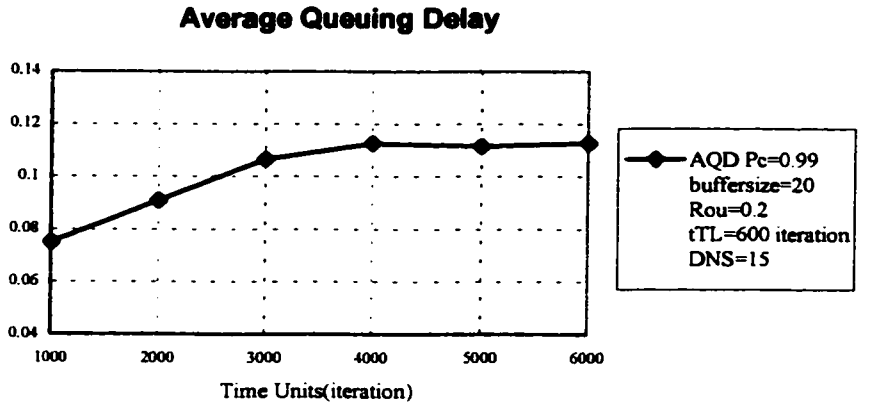


Figure 3- 41: Average Queuing Delay vs. Time Units

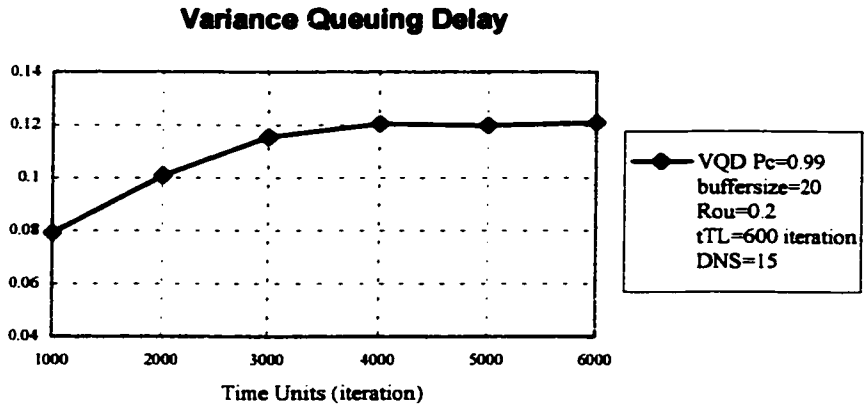


Figure 3- 42: Variance Queuing Delay vs. Time Units

3.4.3 Average Buffer Overflow (ABO)

This performance criterion is directly related to the average number of messages in a node buffer. It is seen that when the buffer size is small, there is more chance for the buffer to overflow, when the buffer size is gradually increased, less messages will encounter overflow, but the variance doesn't change too much (Figure 3- 45 and Figure 3- 46).

From Figure 3- 43 and Figure 3- 44, we can see the linear increase in ABO with input traffic and the variance is approximately linearly increasing as well.

By increasing DNS numbers, we find that the ABO decreases, especially when DNS numbers change from 5 to 10, ABO decreased sharply (Figure 3- 47 and Figure 3- 48).

When Pc increases, the ABO increases too, because more messages succeeded causes to intermediate nodes congested. (Figure 3- 49)

When simulation time elapses, the ABO increases, (Figure 3- 51) and its variance increase too (Figure 3- 52).

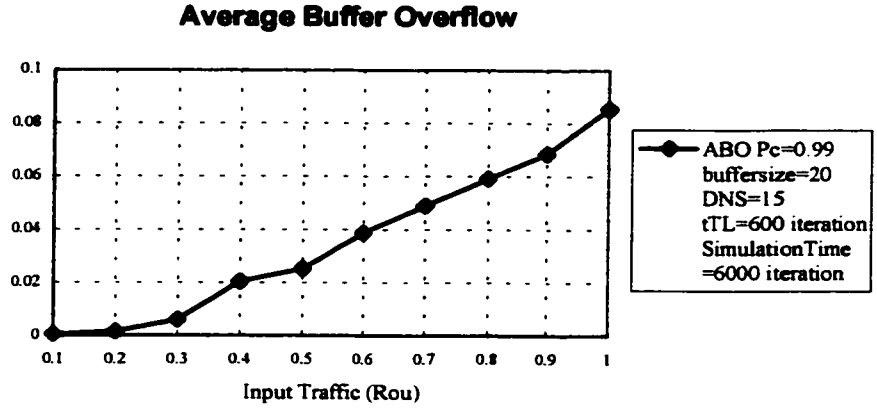


Figure 3- 43: Average Buffer Overflow vs. Input Traffic

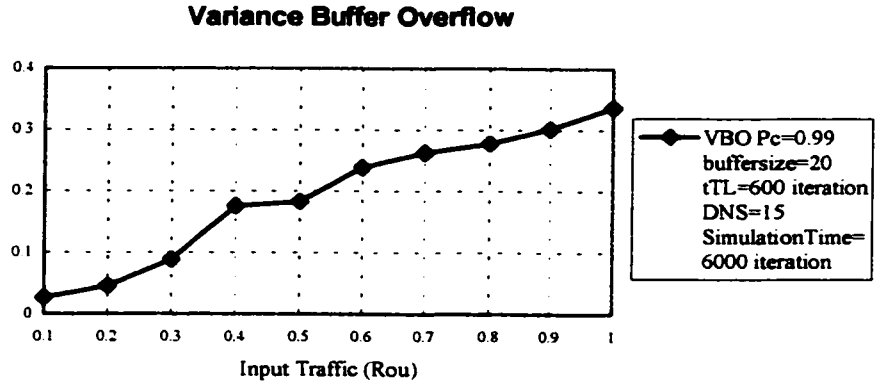


Figure 3- 44: Variance Buffer Overflow vs. Input Traffic

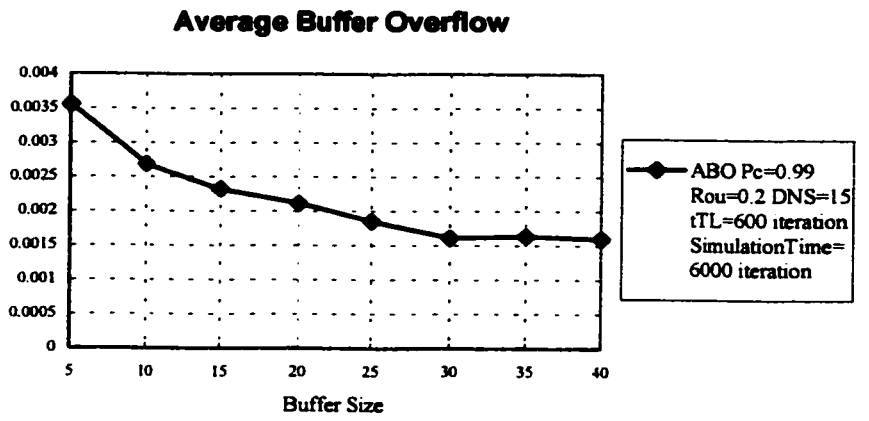


Figure 3- 45: Average Buffer Overflow vs. Buffer Size

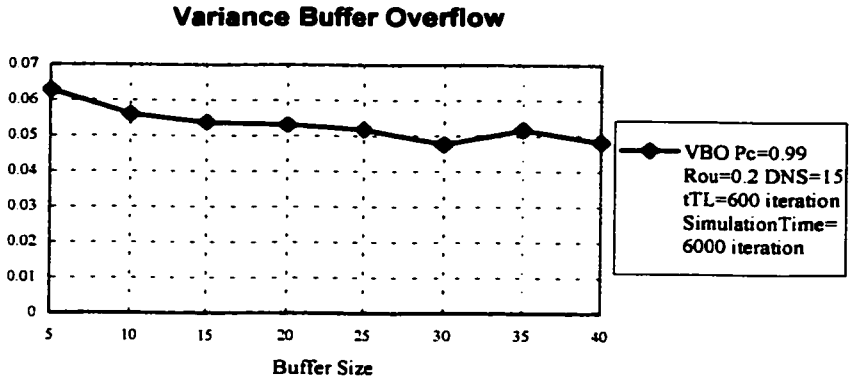


Figure 3- 46: Variance Buffer Overflow vs. Buffer Size

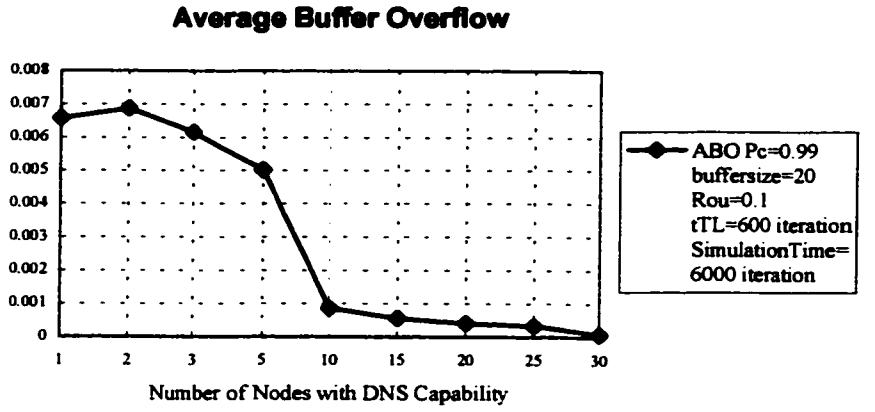


Figure 3- 47: Average Buffer Overflow vs. DNS Numbers

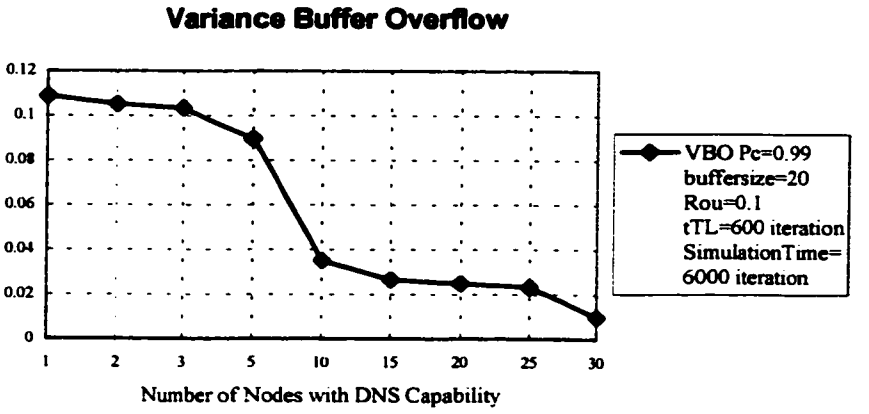


Figure 3- 48: Variance Buffer Overflow vs. DNS Numbers

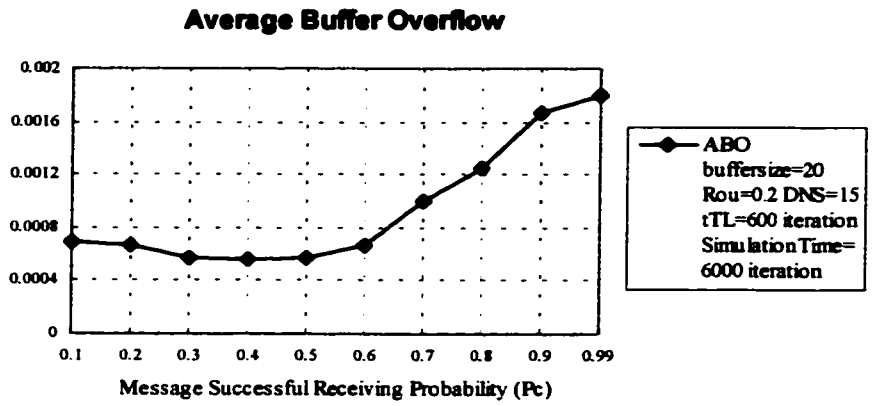


Figure 3- 49:Average Buffer Overflow vs. Pc

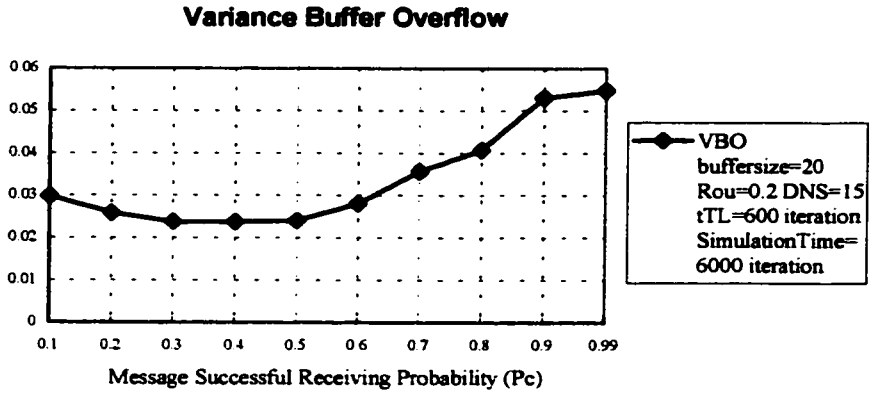


Figure 3- 50: Variance Buffer Overflow vs. Pc

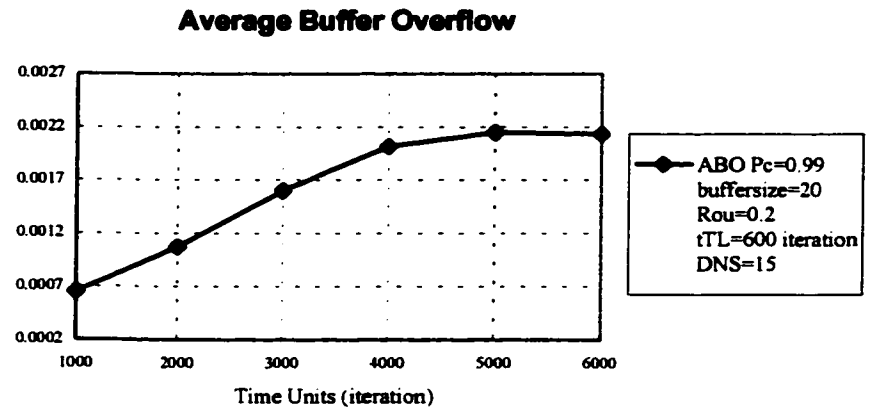


Figure 3- 51:Average Buffer Overflow vs. Time Units

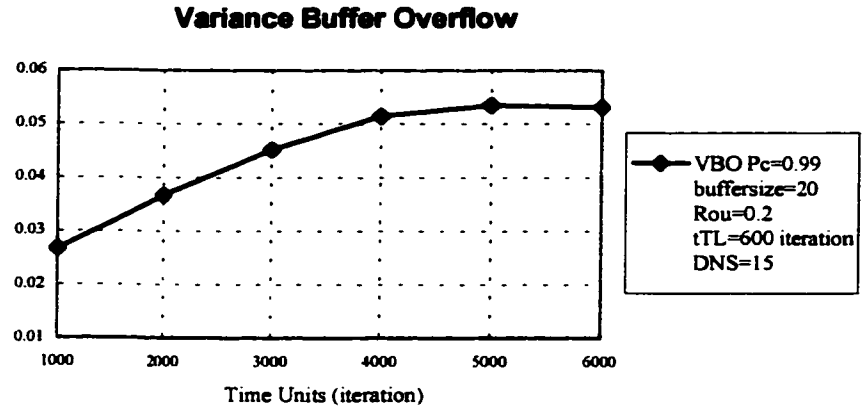


Figure 3- 52: Variance Buffer Overflow vs. Time Units

3.4.4 Overhead

Overhead is the percent of all control messages (Hello, Query and Response).

From Figure 3- 53 and Figure 3- 54, we see that Class 1 nodes overhead decreases with increasing the input traffic and Class 2 nodes overhead doesn't change too much.

According to Figure 3- 55 and Figure 3- 56, we can see that buffer size doesn't affect overhead criterion.

When increasing DNS numbers, we find that Class 1 nodes overhead decreases gradually but saturates for a DNS number>10; on the other hand, Class 2 nodes overhead decreases as DNS numbers increases (Figure 3- 57 and Figure 3- 58).

From Figure 3- 59 and Figure 3- 60, we find that the overhead of class 1, class 2 nodes decreases by increasing message successfully receiving Probability (Pc).

With the elapse of simulation time, the overhead of the two classes decreases a little bit (Figure 3- 61 and Figure 3- 62).

From this group of figures, we also can find that the overhead of Class 1 nodes is larger than Class 2, the reason is that Class 1 nodes query policy is more elaborate than that of Class 2 nodes, so there are more Query and Response messages.

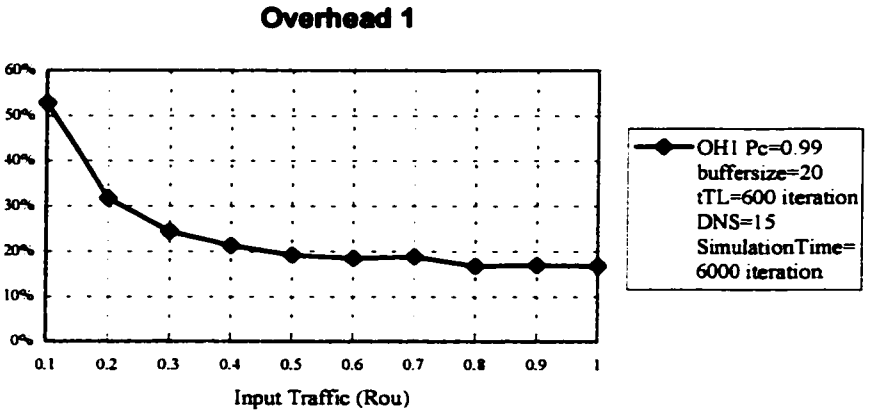


Figure 3- 53: Overhead1 vs. Input Traffic

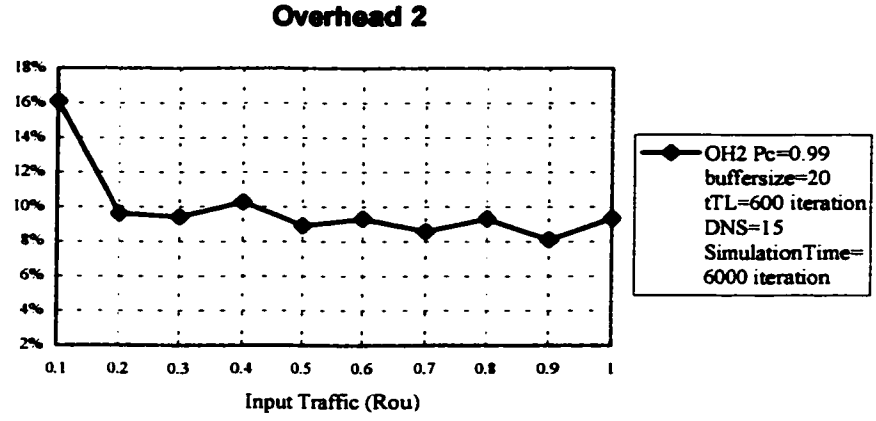


Figure 3- 54: Overhead 2 vs. Input Traffic

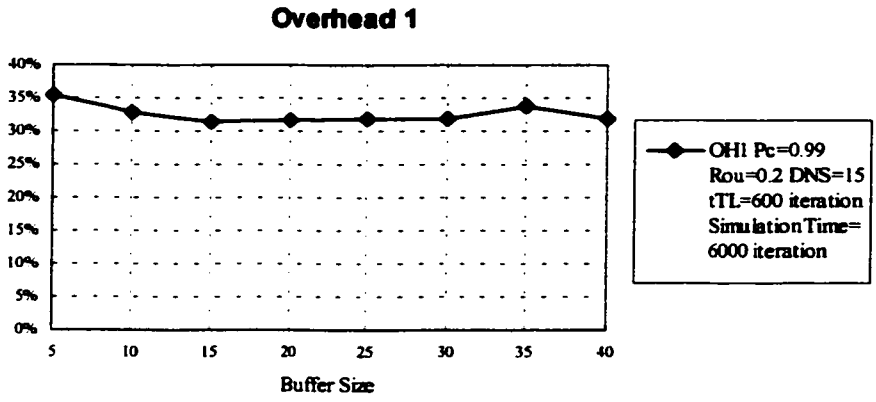


Figure 3- 55: Overhead 1 vs. Buffer Size

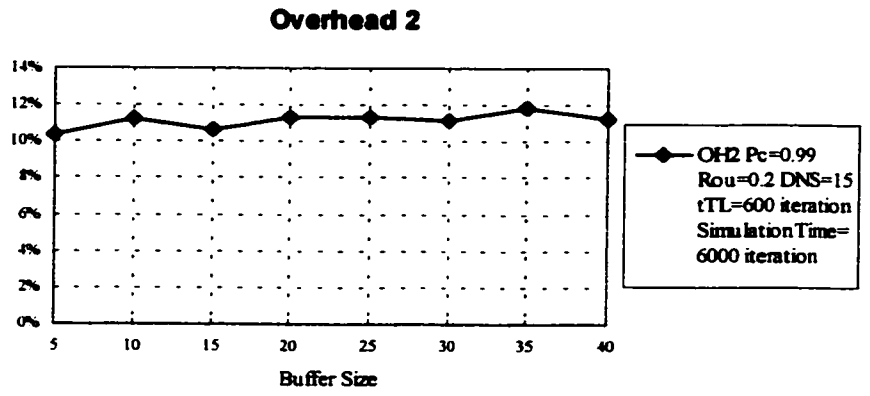


Figure 3- 56: Overhead 2 vs. Buffer Size

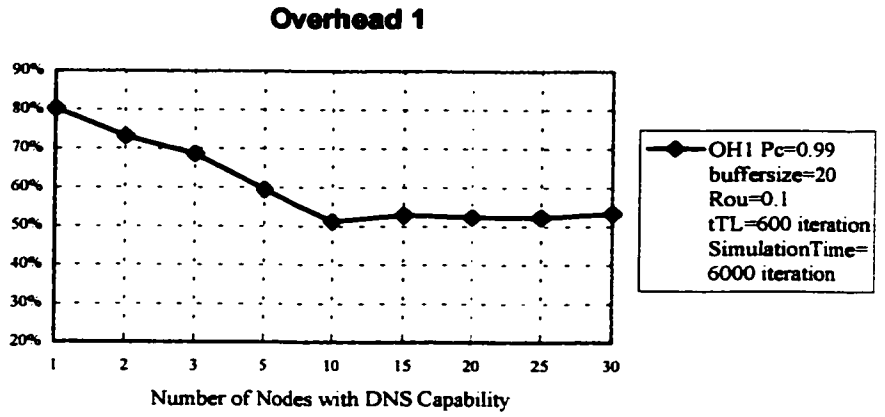


Figure 3- 57: Overhead 1 vs. DNS numbers

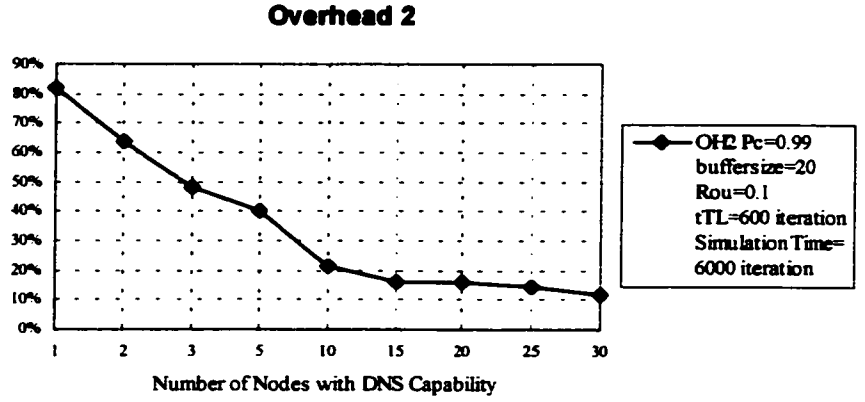


Figure 3- 58: Overhead 2 vs. DNS numbers

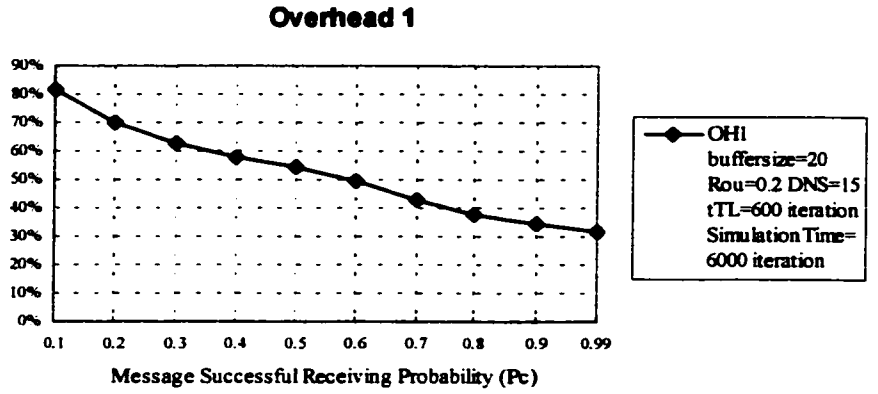


Figure 3- 59: Overhead 1 vs. Pc

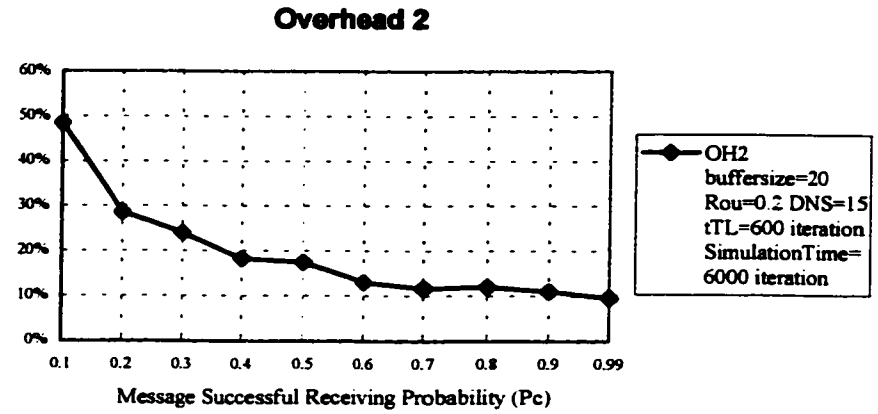


Figure 3- 60: Overhead 2 vs. Pc

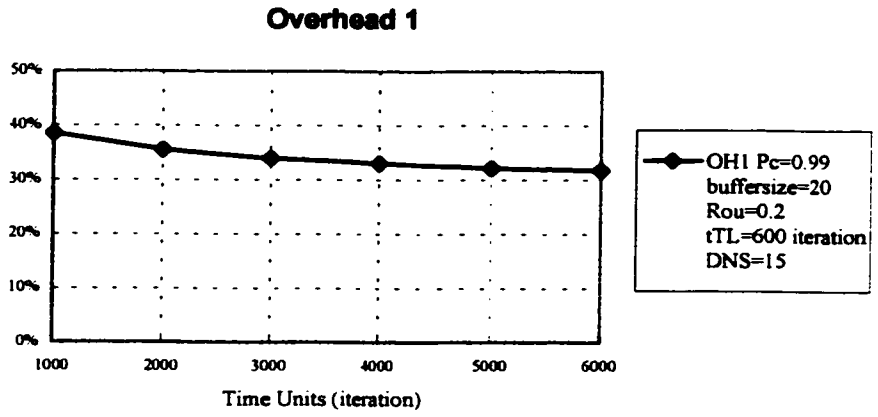


Figure 3- 61: Overhead 1 vs. Time Units

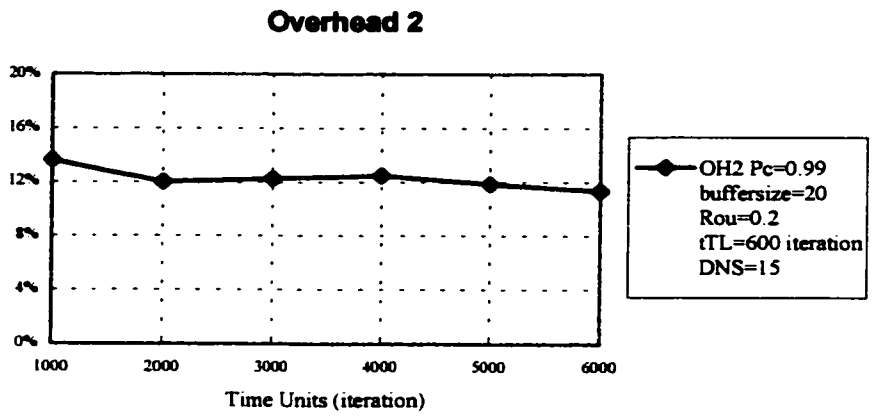


Figure 3- 62:Overhead 2 vs. Time Units

3.4.5 Average Throughput

Average Throughput is a very important criterion, it is the ratio of the number of packets that are successfully transmitted in a very long interval to the maximum number of packets that could have been transmitted with continued transmission on the channel.

As shown in Figure 3- 63 and Figure 3- 64, we can see that the average throughput decreases by increasing the input traffic, because more traffic causes to less opportunity for message to succeed (Figure 3- 65). However, the buffer size doesn't affect this criterion too much, since when enlarging buffer size, the message have less chance to overflow but have more chance to suffer timeout, it is a trade-off between this two criterion, and therefore, no much change of the average throughput (Figure 3- 66).

According to Figure 3- 67 and Figure 3- 68, we can see that increasing DNS numbers largely improves the throughput, the same phoneme is as to increasing

message receiving successfully probability Pc (Figure 3- 69 and Figure 3- 70). But increasing simulation time, the throughput doesn't change too much (Figure 3- 71 and Figure 3- 72).

We also can see that the Class 1 nodes throughput is better than Class 2 nodes, since Class 1 nodes' call is local and Class 2 nodes is terrestrial Global IP.

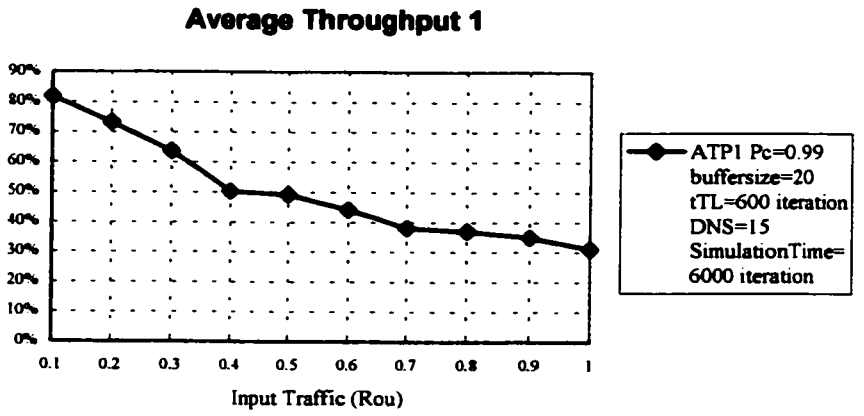


Figure 3- 63: Average Throughput 1 vs. Input Traffic

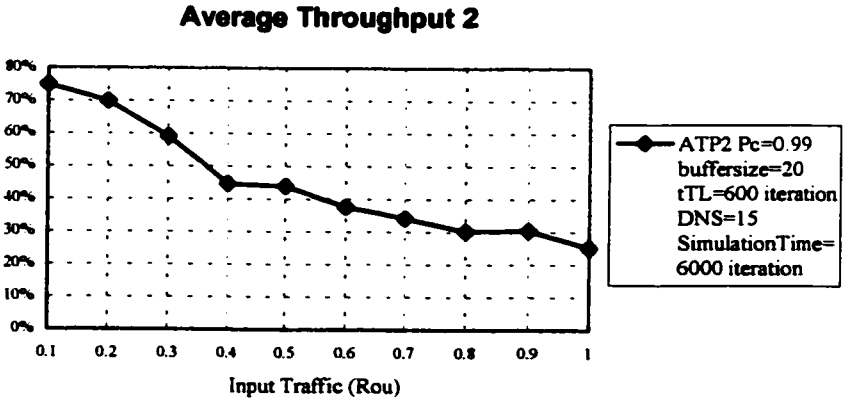


Figure 3- 64: Average Throughput 2 vs. Input Traffic

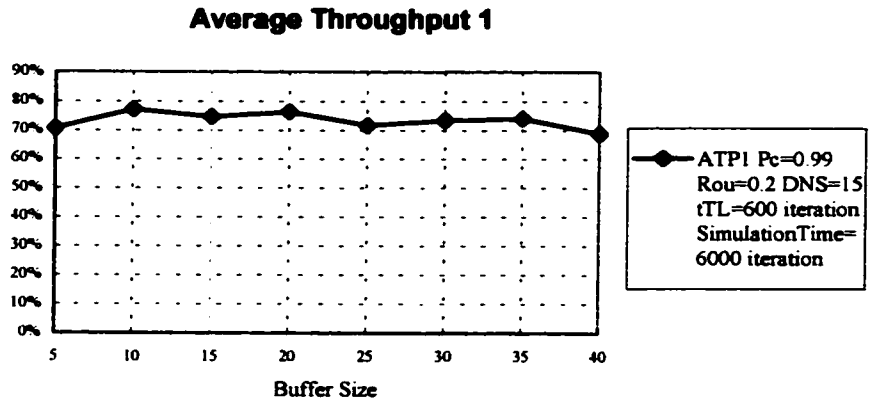


Figure 3- 65: Average Throughput 1 vs. Buffer Size

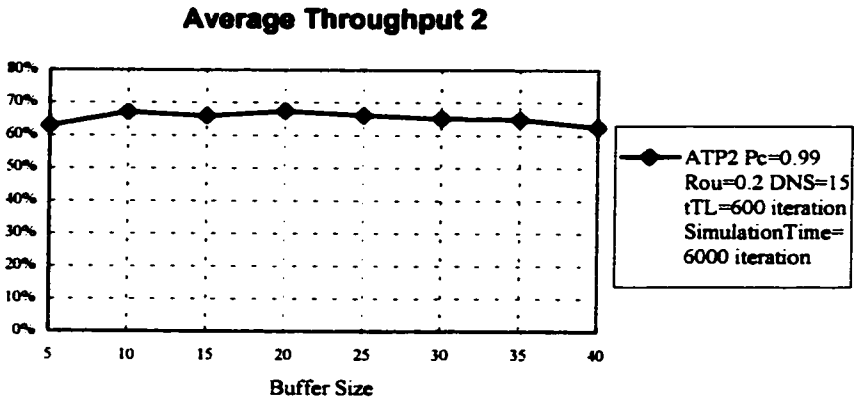


Figure 3- 66: Average Throughput 2 vs. Buffer Size

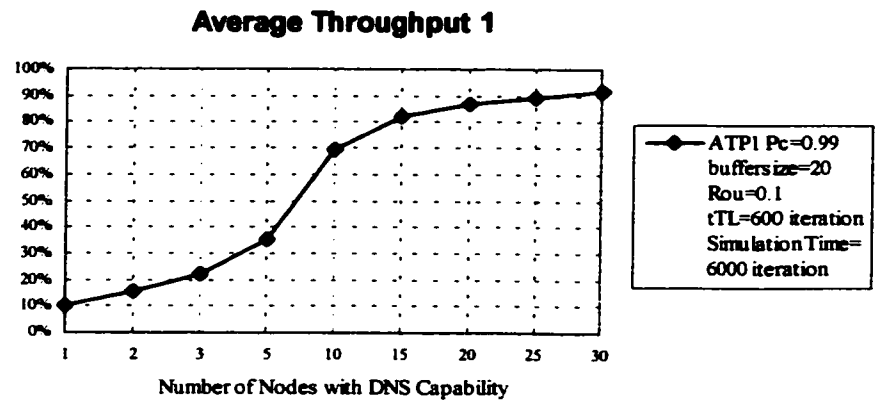


Figure 3- 67: Average Throughput 1 vs. DNS numbers

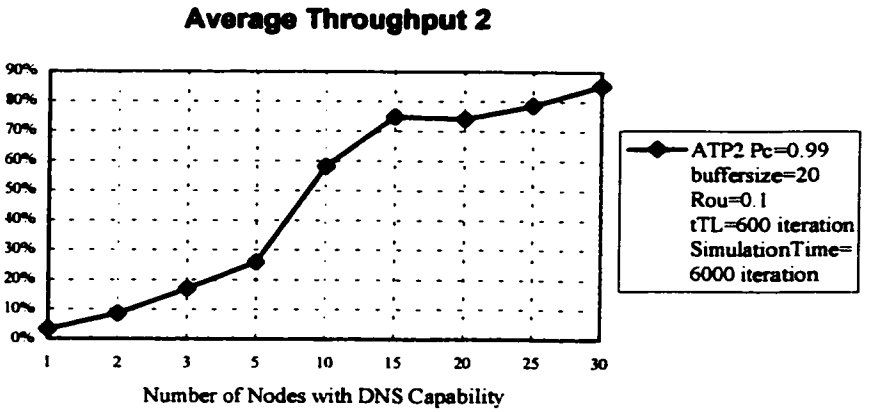


Figure 3- 68: Average Throughput 2 vs. DNS numbers

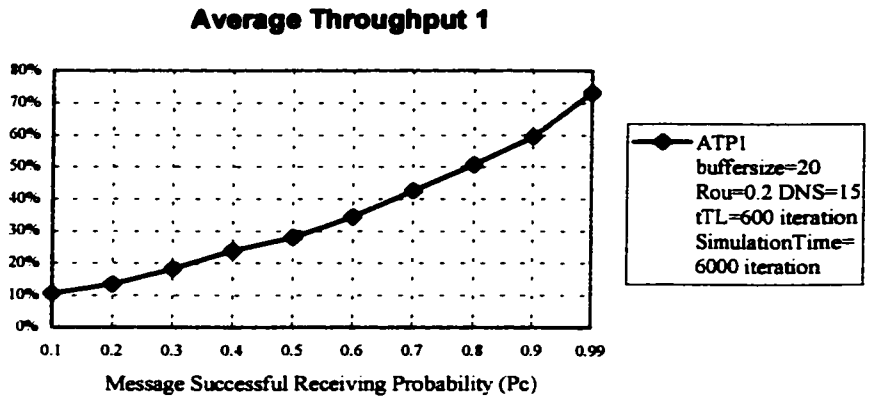


Figure 3- 69: Average Throughput 1 vs. Pc

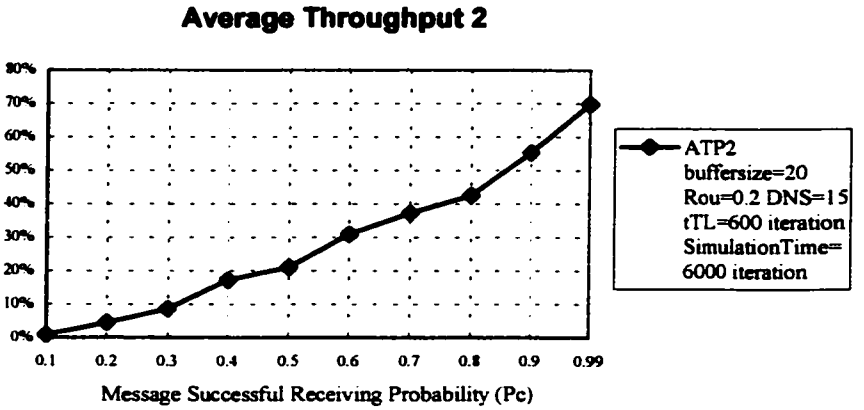


Figure 3- 70: Average Throughput 2 vs. Pc

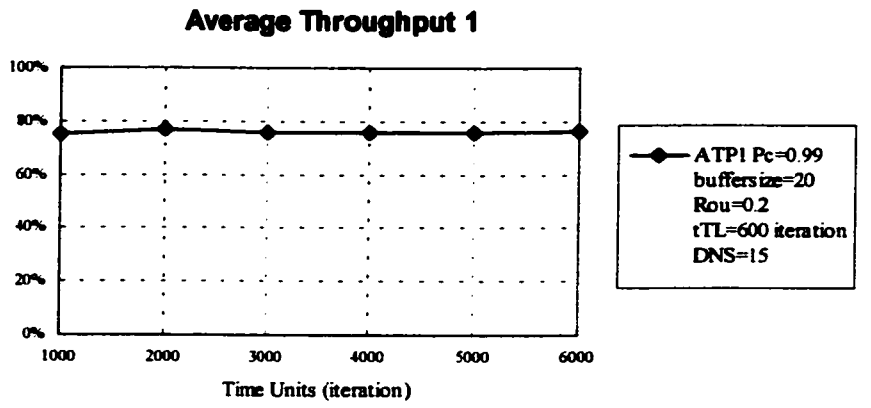


Figure 3- 71: Average Throughput 1 vs. Time Units

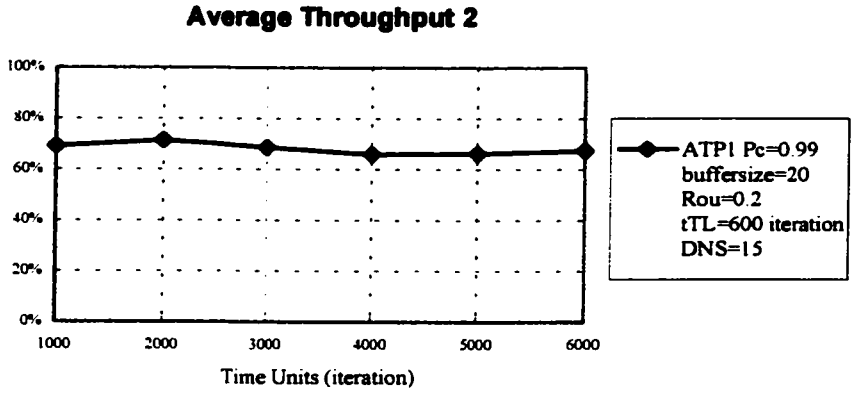


Figure 3- 72: Average Throughput 2 vs. Time Units

3.4.6 Average Latency of First Route Reply

The average latency of the first route reply of class 1 and class 2 nodes are shown in Figure 3- 73 and Figure 3- 75, it is easy to see that both classes of node takes more time to get the first route when increasing the input traffic, and the variance is also becoming larger (Figure 3- 74 and Figure 3- 76). However, the nodes buffer size has less effect on this criterion (Figure 3- 77,Figure 3- 78,Figure 3- 79 and Figure 3- 80).

By increasing the number of nodes having DNS capability in the whole network, we can see from Figure 3- 81 and Figure 3- 83 that both two classes nodes spend less time to get the route and the variance also decreases (Figure 3- 82 and Figure 3- 84).

We also find that increasing the message receiving successfully probability Pc leads to less latency of the first route reply, as well as the variance. (Figure 3- 85,Figure 3- 86,Figure 3- 87 and Figure 3- 88).

From this group of figures, we find that Class 1 nodes spend more time to get first route reply than Class 2 nodes, just because of the fact that Class 1 nodes route discovering is more complicate and takes more time than Class 2 nodes.

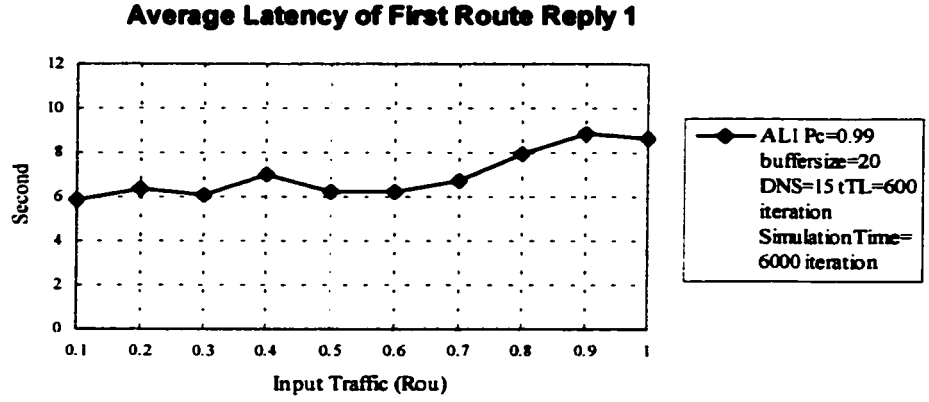


Figure 3- 73: Average Latency of First Route Reply 1 vs. Input Traffic

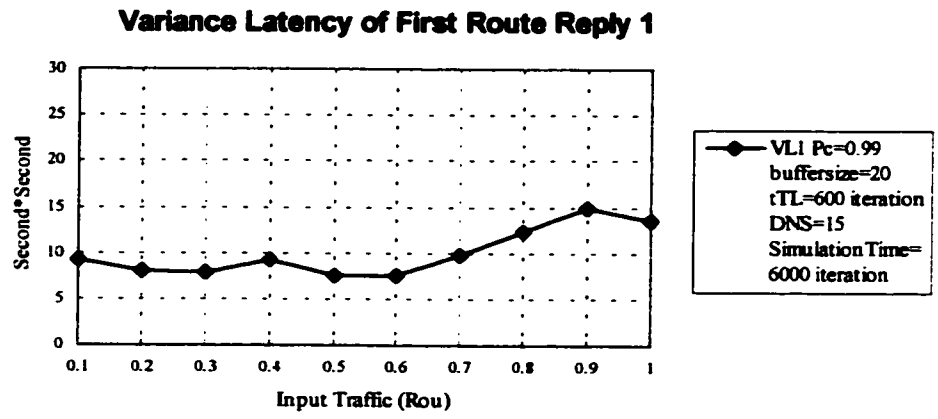


Figure 3- 74: Variance Latency of First Route Reply 1 vs. Input Traffic

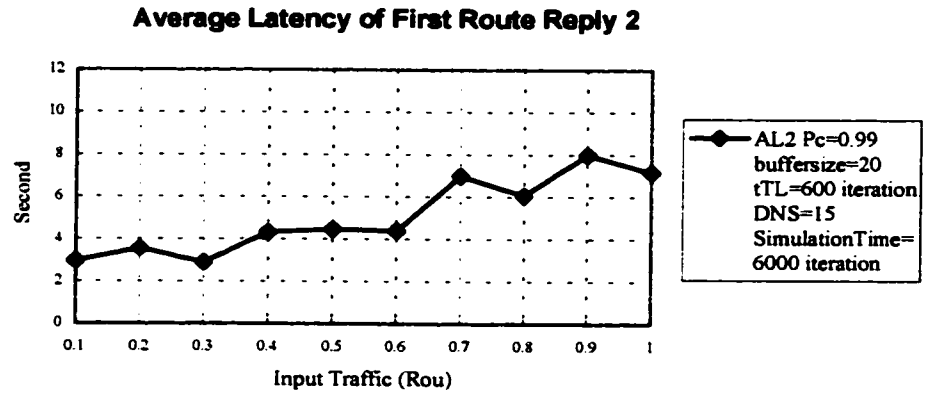


Figure 3- 75: Average Latency of First Route Reply 2 vs. Input Traffic

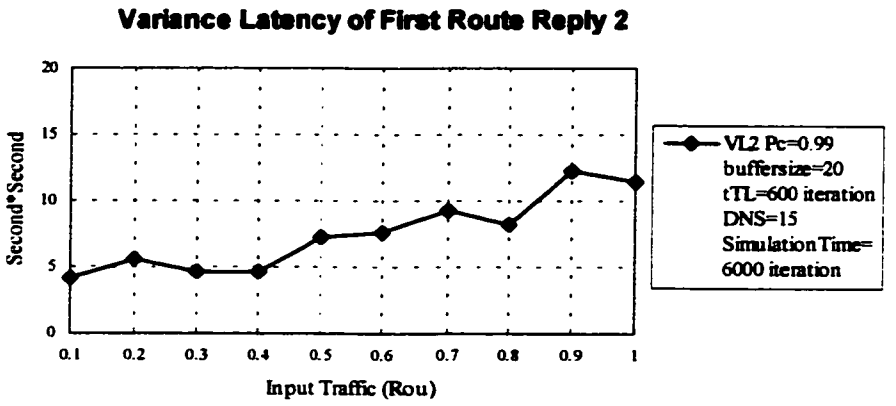


Figure 3- 76: Variance Latency of First Route Reply 2 vs. Input Traffic

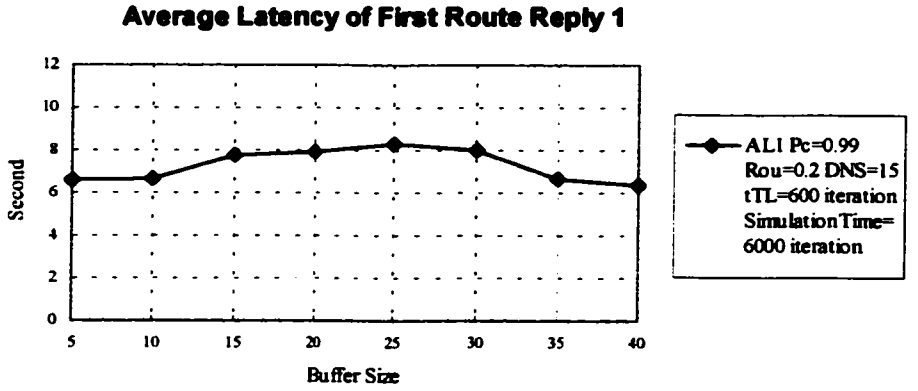


Figure 3- 77: Average Latency of First Route Reply 1 vs. Buffer Size

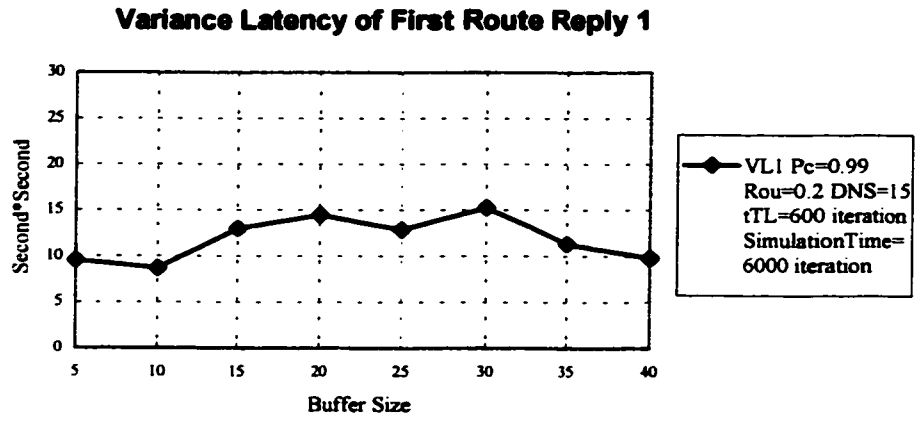


Figure 3- 78: Variance Latency of First Route Reply 1 vs. Buffer Size

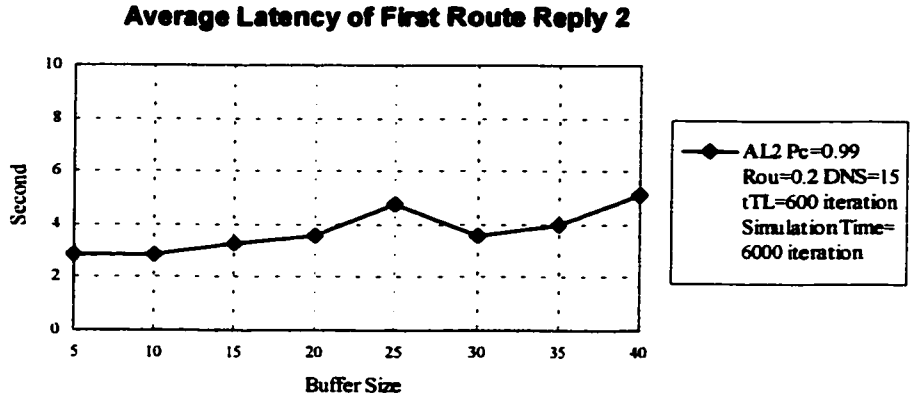


Figure 3- 79: Average Latency of First Route Reply 2 vs. Buffer Size

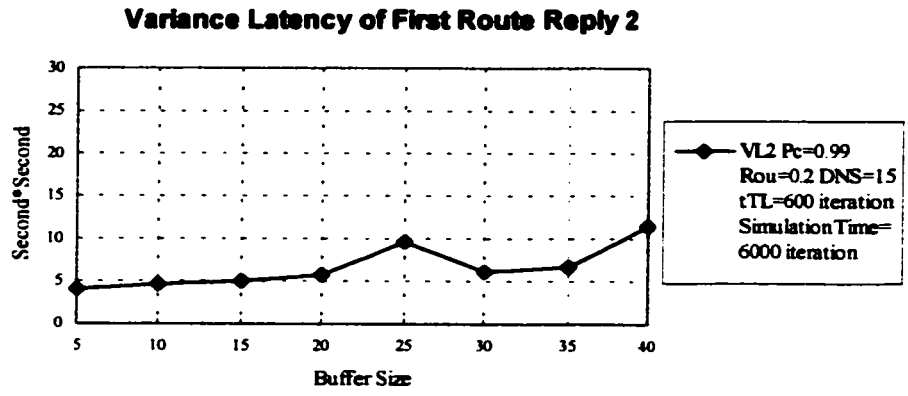


Figure 3- 80: Variance Latency of First Route Reply 2 vs. Buffer Size

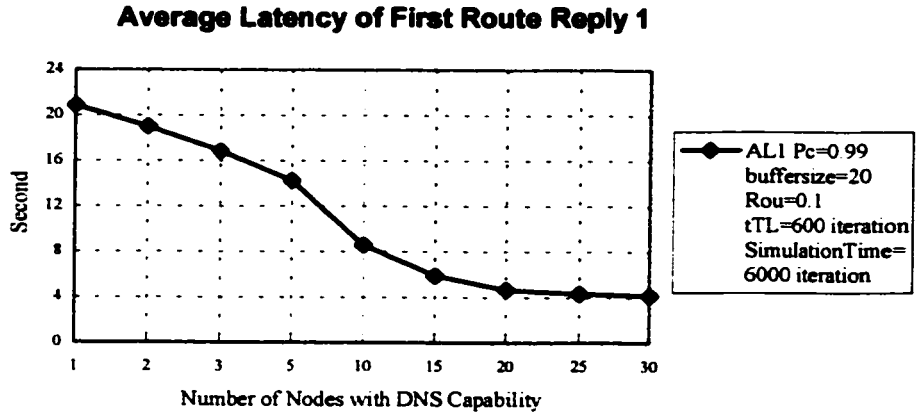


Figure 3- 81: Average Latency of First Route Reply 1 vs. DNS numbers

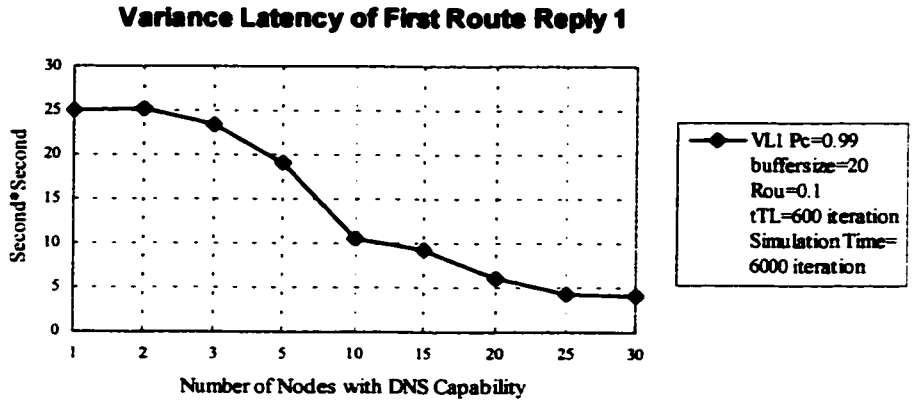


Figure 3- 82: Variance Latency of First Route Reply 1 vs. DNS numbers

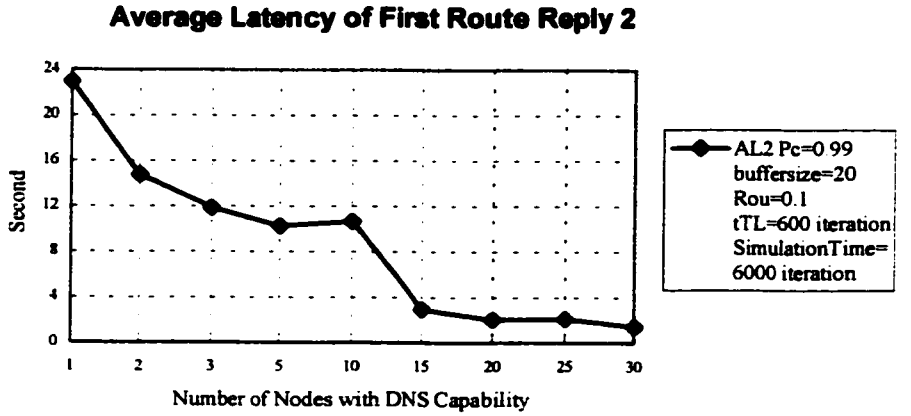


Figure 3- 83: Average Latency of First Route Reply 2 vs. DNS numbers

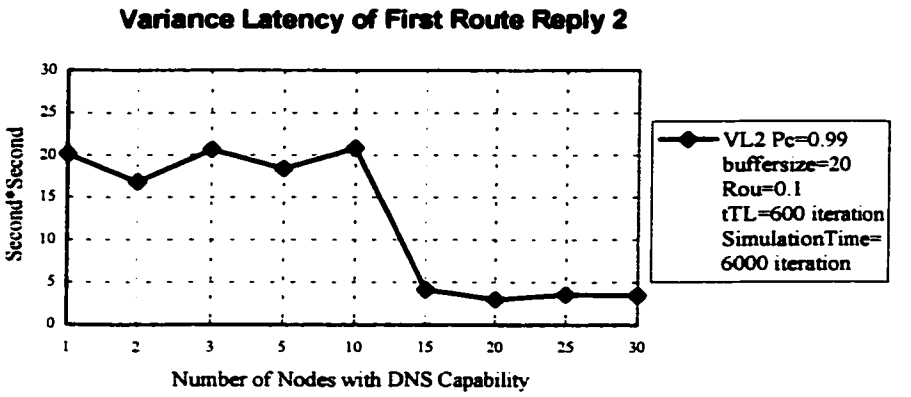


Figure 3- 84: Variance Latency of First Route Reply 2 vs. DNS numbers

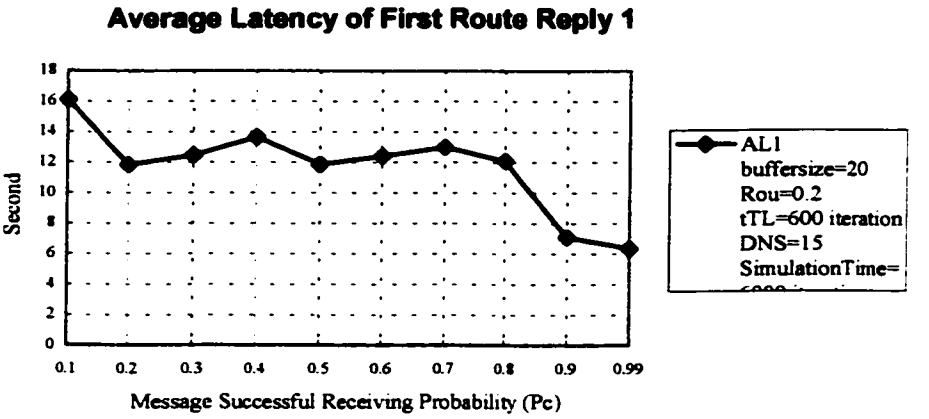


Figure 3- 85: Average Latency of First Route Reply 1 vs. Pc

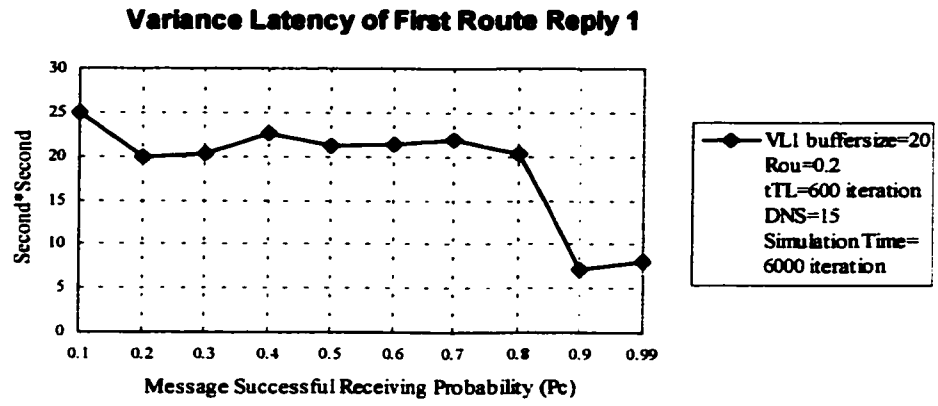


Figure 3- 86: Variance Latency of First Route Reply 1 vs. Pc

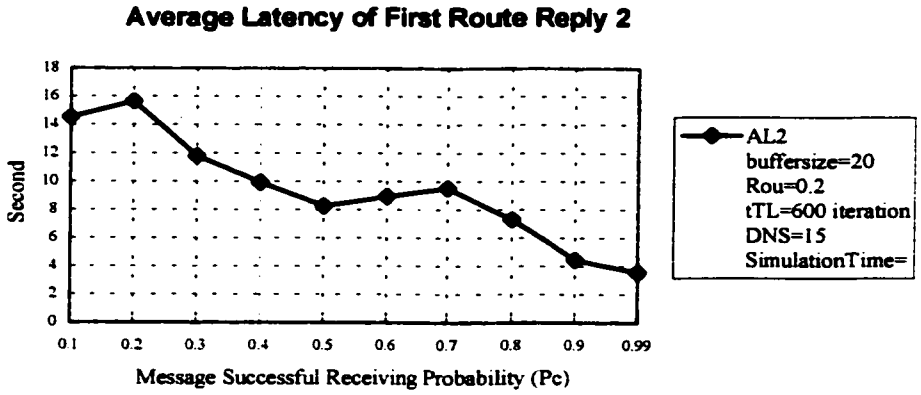


Figure 3- 87: Average Latency of First Route Reply 2 vs. Pc

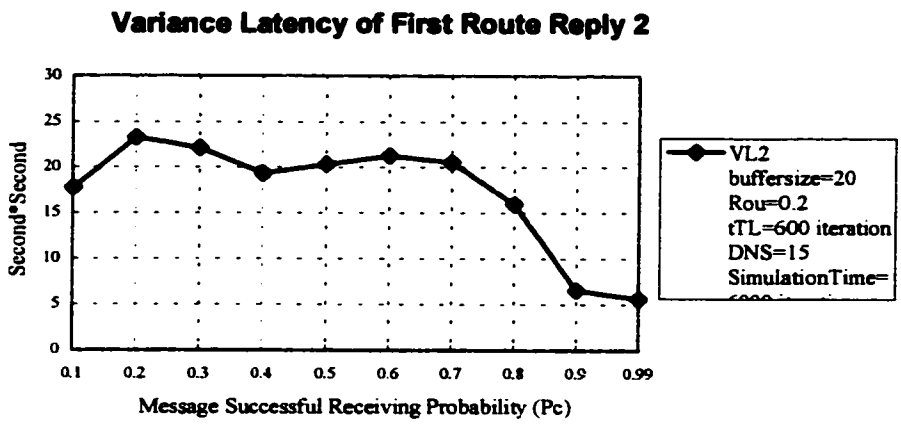


Figure 3- 88: Variance Latency of First Route Reply 2 vs. Pc

CHAPTER 4

4 SUMMARY & CONCLUSIONS

4.1 Summary

In chapter 1 of this thesis, we have given a short general introduction to IEEE 802.11 Wireless LAN Protocol, 802.11 topology, physical layer and MAC layer is presented.

In chapter 2, we gave a detailed review and discussion of the current routing protocol in wireless ad hoc network. We catalogued four types of current ad hoc routing techniques. DSDV, AODV, ZRP and HSR routing protocols' characteristics and functionality are compared.

In chapter 3, a new routing protocol in wireless ad hoc network named DNS Based Wireless Routing Protocol was introduced based on the ideas and intellectual properties of Dr. A.K. Elhakeem. We gave a detailed description of the scheme and its various parameters.

4.2 Conclusion

In this thesis, we propose a new IP routing technique for ad-hoc wireless networks. In DBRP, each node knows the node connectivity within its reachable set; some nodes have DNS capability, which are responsible for route discovery and also

handle the data message forwarding. The routing is performed on two levels: Local node and terrestrial Global IP.

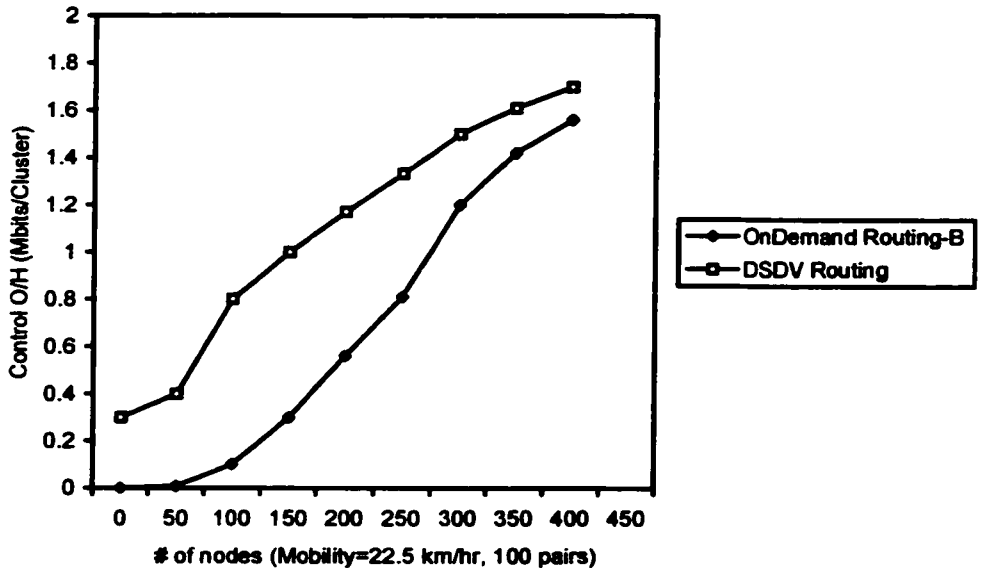


Figure 4- 1:Control O/H over number of nodes [Iwata 1999]

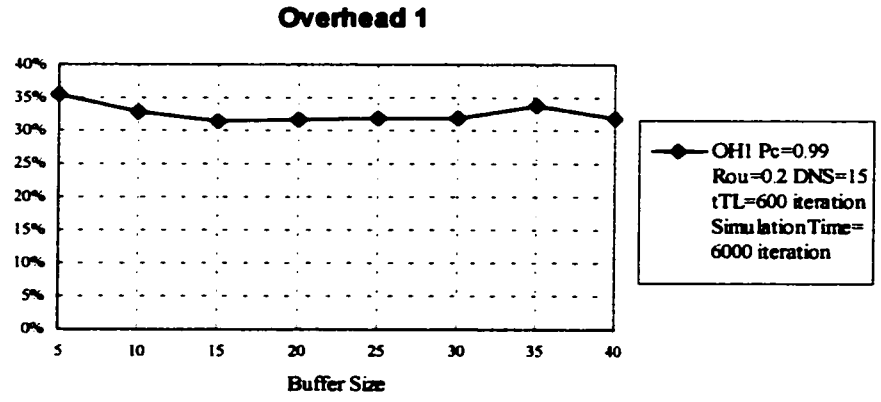


Figure 4- 2: Control O/H over Buffer Size

From the simulation results we conclude that by using DNS technique, the networks' performance is highly improved.

According to Figure 4-1, when the number of nodes in the networks is 200, the overhead of DSDV is about 40% and that of On-demand Routing is about 110%; but according to Figure 4-2, we see that the overhead of DBRP is only 35%. Compared DBRP routing protocol with other routing protocols, we find that by properly setting of input parameters, DBRP will achieve lower overhead in the ad hoc networks. That means DBRP avoid flooding and network congestion.

Also, we compare ZRP with DBRP. From Figure 4-3 and Figure 4-4, we can see that ZRP's successful datamessage is much less than that of the DBRP. The loss data messages probability of ZRP is about 55%, that means there are about 45% data messages can successfully reach their destination. However, the successful data messages probability of DBRP is above 70%. So, we conclude DBRP has low loss probability.

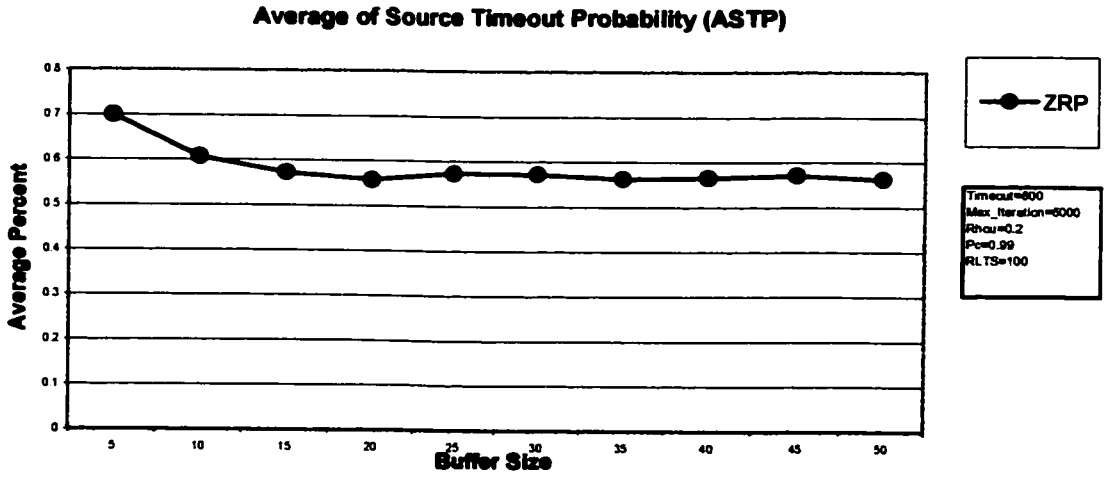


Figure 4- 3:Average of Source Timeout Probability vs. Buffer Size

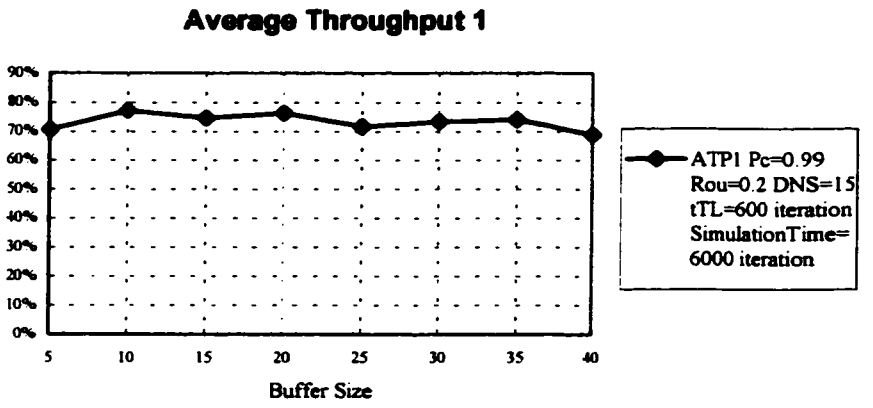


Figure 4-4:Average Throughput vs. Buffer Size

As we conclude, the DBRP has some advantages, such as low loss probability, less transmission and queuing delay, as well as avoiding flooding, network congestion and buffer overflow. So this protocol can support real-time data streams. We also conclude that the higher the complexity of routing, the more costly will be the implementation which should be avoided for all those small devices of Bluetooth and Wireless LANs to be cost effective.

4.3 Suggestion for Future Work

There are still other features which were not explained in detail in this thesis. Also there remains some question which needs further study and investigation.

First, rather than assigning each node the same IP address during the whole simulation time, one may use DHCP (Dynamic Host Configuration Protocol) to assign a new IP address to nodes moving far from one DNS nodes and closer to another one should be studied. It will solve triangle problem.

Second, We assume the number of nodes within networks is fixed, so the scalability of DBRP should be further studied.

At the end, we just briefly compare DBRP with other routing protocols. In order to further prove DBRP's effectiveness,the more detail comparsion and discussion should be further presented.

5 REFERENCES

[1] B. CrowI. Widjaja, J.Kim, P.Sakai, " IEEE 802.11 Wireless Local Area Networks," IEEE Communications Magazine, Sept. 1997, pp.116-126.

[2] P.T. Davis and C.R. Mcguffin, "Wireless Local Area Networks," Mcgraw-Hill, pp.85-115,1994.

[3] L.M. Correia and R.Prasad, " An overview of wireless broadband communications," IEEE Commmun. Mag., vol.35. no.1, pp.38-33,Jan.1997.

[4] Wireless Medium Access Control and Physical LayerWG, IEEE Draft standard P802.11, " Wireless LAN," IEEE Stds. Dept, D3, Jan. 1996.

[5] J. Deng and R.S. Chang, "A Priority Scheme for IEEE 802.11 DCF Access Method" ,IEICE TRANS. COMMUN.,Vol.E82-B. NO.1 Jan. 1999

[6] R.O. LaMaire, A. Krishna, P. Bhagwat, and J. Panian, " Wireless LANs and Mobile Networking Standards and Future Directions", IEEE Communications Magazine, pp.86-95, Aug. 1996

[7] Harshal S.Chhaya, Snjay Gupjay Gupta, "Performance modeling of synchronous data transfer methods of IEEE 802.11 MAC Protocol" Wireless Networks vo.3 no.3 pp.217-234,1997.

[8] G. Bianchi "IEEE 802.11 – Saturation Throughput Analysis", IEEE Communications Letters, Vol. 2, No. 12, pp.318-321 , Dec 1998

[9] T.N. Stern. " Design Issues Relevant to Developing an Integrated Voice/Data Mobile Radio System, "IEEE Transactions on Vehicular Technology, Vol.39, No. 1, pp75-82, Feb. 1990.

[10] Elizabeth M. Royer, Chai-Keong Toh "A review of Current Routing Protocols

for Ad Hoc Mobile Wireless Networks" IEEE Personal Communications, pp.46, April 1999.

[11] O.Andrisano, " An Integrated Approach for the Design of Wide-Band Wireless LAN", IEEE ITC 98, pp.121-126, June 1998.

[12] O.Andrisano, " Wireless Multimedia Assessment with Traffic" in Proceed of IEEE multimedia communicatiob workship 1998, Sept. 1998.

[13] Richard Van Nee Et La, " New High-Rate Wireless LAN Standards," IEEE communications Magazine, Dec 1999. PP.82-88.

[14] Haas,Z.J., " A New Routing Protocol for the Reconfigurable Wireless Networks," ICUPC'97, San Diego, CA, Oct 12, 1997.

[15] Haas, Z.J. , Pearlman, M.R., " The Performance of Query Control Schemes for the Zone Routing Protocol", SIGCOMM'98, Vancouver, BC, Sept 2-4, 1998.

[16] Haas, Z.J., Tabrizi, S., " On Some Chanllenges and Design Choices in Ad-Hoc Communications", MILCOM'98, Boston, MA, October 18-21, 1998.

[17] Haas, Z.J. and Pearlman, M.R. " Determining the Optimal Configuration for the Zone Routing Protocol," to appear in IEEE JSAC issue on Ad-Hoc Networks,June 1999.

[18] Joa-Ng, M., Lu, I.T., " A Peer-to-Peer Zone-Based Two-Level Link Statte Routing for Mobile Ad-Hoc Networks," to appear in IEEE JSAC issue on Ad-Hoc Networks, Jume 1999.

[19] Johnson, D.B., Maltz, D.A., " Dynamic Source Routing in Ad-Hoc Wireless Networks," in Mobile Computing, chapter5, PP.153-181, Kluwer, 1996.

[20] Moy, J., " OSPF Version 2," Internet Draft Standrad, RFC 2178, July 1997.

[21] Zygmunt J.Haas, Marc R. Pearlman, " The Zone Routing Protocol for Ad Hoc Networks" Internet Draft Standard, RFC2026, June 1999.

[22] Murthy, S., Garcia-Luna-Aceves, J.J., " An Efficient routing Protocol for wireless Networks," MONET, vol.1, no.2, PP.183-197, October 1996.

[23] Park, V.D. , Corson, M.S., " A Highly Adaptive Distributed Routing Algorithm for Mobile Networks," IEEE INFOCOM'97, Kobe, Japan, 1997.

[24] Perkings, C.E., Bhagwat, P., " Highly Dynamic Destination-Sequenced Distance-Vector Routing for Mobile Computers," ACM SIGCOMM, vol.24, no.4, October 1994.

[25] PerKins, C.E., Royer, E.M., " Ad-Hoc On-Demand Distance Vector Routing", IEEE WMCSA'99, Feb.1999.

[26] M.Steenstrup, Editor, "Routing in Communications Networks", Prentice Hall, 1995.

[27] N.M Maxemchuk, M. Elzarki, " Routing and Flow Control in High Spread Wide Area Networks", IEEE Proceeding, vol.78, PP.204-221, Jan 1990.

[28] R.Davies " Ad-Hoc Wireless Networking: Contention Free Multiple Access Using Token Passing" in Proc. Of VCT'95, pp.361-365.

[29] A.Hoffmann, " Performance Analysis of a Token Based MAC Protocol with Asymmetric Polling Strategy" in Proc. ICC'94 Louisiana, USA pp.1306-1310.

[30] D.Dardari, " A General approach to the Evalution and Characterization of Packet Radio Networks Performance" , International Journal of Wireless Information Network, vol.3, no.4, 1996, pp.203-217.

[31] L.Kleinrock, "Queueing Systems", John Wiley & Sons 1975.

[32] D.L. Lough, T.K. Blankenship, K. J. Krizman, " A Short Tutorial on Wireless LANs and IEEE 802.11", The Bradley Dept. of ece at Virginia Polytechnic Institute and State University.

[33] Chiang CC., Wu H-K, Liu W., and Gerla M. 1997 " Routing in Clustered Multihop, Mobile, Wireless Networks", Proc. IEEE Singapore Int. Conf. Networks.

[34] Tranenbaum, Andrew S., " Computer Networks – Third Edition", Upper Saddle River, New Jersey: Prentice Hall PTR,1996.

[35] Proakis, John G., " Digital Communications – Third Edition" , McGraw Hill Series in Electrical and Communication Engineering, March 1995.