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

UMI®

Performance Analysis of a Modified Zone-Based Hierarchical Link State Routing

Shaopeng Zhu

A Thesis

in

the Department

of Electrical and Computer Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of Master of Applied Science at

Concordia University

Montreal, Quebec, Canada

October 2001

© Shaopeng Zhu, 2001

National Library of Canada
Bibliothèque nationale du Canada

Acquisitions and Bibliographic Services
Acquisitions et services bibliographiques

395 Wellington Street
Ottawa ON K1A 0N4
Canada

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

Our file Notre référence

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-68449-0

Canada

Abstract

Performance Analysis of a Modified Zone-Based
Hierarchical Link State Routing

Shaopeng Zhu

Existing IEEE 802.11 provide minimal connectivity for wireless network, while a number of different routing protocols are proposed for wireless ad-hoc network connectivity in recent years. Traditional routing protocols are classified in two categories: proactive and reactive. More efficient routing protocols combine both features to minimize network overhead and transfer delay.

In this work, a modified zone-based hierarchical link state routing protocol (MZHLS) is presented. In MZHLS, the network is divided into non-overlapping zones. Link state routing is performed on two levels: node level and zone level. MZHLSP is proactive at node level and reactive when destination node exists in different zones. Only zone ID and node ID of a destination are needed for routing and zone ID is found by searching specific zones, for example, periphery zones.

Performance analysis of MZHLS is emphasized. To evaluate the performance of MZHLS, a simulation model of file transfer process is built in C programming language. Performance analysis of MZHLS focuses on transfer delay, overhead, throughput, buffer overflow, etc. The protocol's behavior and changes introduced by variations on some of the parameter settings and the mechanisms that makes up the protocol are examed. Simulation results show the parameter and mechanisms that have the greatest impact and the trade off that exist between them.

Acknowledgements

I would like to express sincere appreciation and gratitude to my thesis supervisor, Dr. A. K. Elhakeem, for his constant support and patient guidance during the entire preparation of this thesis and for his precious suggestions and encouragement in the work.

Also I would like to express my deep gratitude to my family for encouraging me towards graduate studies and for their constant support throughout my studies.

Table of Contents

List of Acronyms and Abbreviations

WLAN	Wireless Local Area Networks
ISM band	Industrial, Scientific, and Medical band
FCC	Federal Communications Commission
DSSS	Direct Sequence Spread Spectrum
FHSS	Frequency Hopping Spread Spectrum
DBPSK	Differential Binary Phase Shift Keying
DQPSK	Differential Quadrature Phase Shift Keying
MAC	Medium Access Control
BSS	Basic Service Set
IBSS	Independent Basic Service Set
DS	Distributed System
AC	Access Point
DIFS	Distributed Inter Frame Space
SIFS	Short Inter Frame Space
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
CAMA/CD	Carrier Sense Multiple Access with Collision Detection
DBF	Distributed Bellman-Ford Algorithm
DU	Data Unit
NLSP	Node Link State Packet
ZLSP	Zone Link State Packet
LKR	Link Request
LKRS	Link Request Response
LR	Location Request
LRS	Location Request Response

List of Figures

List of Tables

Chapter 1

Introduction to IEEE 802.11

Wireless local area networks (WLANs) provide cable-free access to data rates of 1 megabits per second or higher in a limited geographical area. A WLAN offers the flexibility to relocate equipment or to reconfigure and add more nodes to the network.

The IEEE 802.11 is developed to define a medium access control and physical layer specification for wireless connectivity for fixed, portable and moving stations within in a local area. The purpose of the 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"[1].

A brief introduction to IEEE 802.11 physical layer and medium access control sublayer is presented in this chapter.

1.1 Physical Layer

There are three media that can be used for transmission over wireless LANs: radio frequency, microwave, and infrared. In 1985 the United State released the industrial, scientific, and medical (ISM) frequency bands. These bands are 902-928MHz, 2,4-2,4853GHz, and 5.725-5.85GHz [13][26] and do not require licensing by the Federal Communications Commission (FCC). Spread spectrum technology is required for implementation with the radio frequency in United States. Two types of spread spectrum

are used: direct sequence spread spectrum (DSSS) and frequency hopping spread spectrum (FHSS).

1.1.1 Direct Sequence Spread Spectrum (DSSS):

With DSSS, the transmission signal is spread over a certain band. Bandwidth spreading is achieved by using a wide band spreading code to modulate the data-modulated carrier. A data bit is mapped into a chip that represents the spreading ratio.

In IEEE 802.11, a spread ratio of eleven for DSSS is required. DSSS IEEE 802.11 WLAN supports mandatory bit rates of 1 and 2 Mbps. DSSS IEEE 802.11 WLAN use the 2.4GHz ISM frequency band, where the 1Mb/s basic rate is encoded using differential binary phase shift keying (DBPSK), and a 2Mbps enhanced rate used differential quadrature phase shift keying (DQPSK). An 11-chip Barker sequence is used to spread the data symbol and 1 MHz data modulated carrier spectrum is spread to 11MHz [14]. The maximum operating range is 60 meter under close environment and 430 meter under open environment.

1.1.2 Frequency Hopping Spread Spectrum (FHSS)

With FHSS, a set of carrier frequencies is selected and they are separated from each other by approximately the data modulation bandwidth. The spreading code is used to control the sequence of the carrier frequencies.

FSHH IEEE 802.11 WLAN uses the 2.4GHz ISM band. A maximum of 79 channels can be selected as the frequency hopping set in the United States [14]. The first channel has a center frequency of 2.402GHz, and the subsequent channels are 1MHZ apart from each other. FCC dictates a 1MHz separation for 2.4GHz ISM band which corresponds to 1Mbps data band. FSHH IEEE 802.11 WLANs can be co-located if the frequency hopping sequences are orthogonal. Specifically, Three orthogonal frequency hopping sequence sets are established and each has 26 hopping sequences. Co-existence of the

multiple FHSS WLAN helps to alleviate congestion and enhance the throughput. The minimum hop rate permitted is 2.5 hop/s. The basic access rate of 1Mbps uses two-level Gaussian frequency shift keying (2-Gaussian). The maximum operating range is 50 meter under close environment and 600 meter under open environment.

1.1.3 Infrared

The infrared system uses signal frequencies with wavelength range from 850nm to 950nm.These systems are not bandwidth limited and can achieve greater transmission rate. But they require line of sight or reflected transmissions. Only the amplitude of the signal is detected and the pulse amplitude modulation is used.

1.2 Medium Access Control Sublayer(MAC)

The IEEE 802.11 defines both physical layer and MAC sublayer. MAC layer is responsible for the channel allocation, frame formatting, protocol data unit addressing, error checking, and fragmentation and reassembly.

1.2.1 IEEE 802.11 Architecture

Each mobile, portable or fixed computer is referred to a station in IEEE 802.11. If two or more stations can directly communicate with each other, they form a basic service set (BSS), the fundamental block in IEEE 802.11.

An independent basic service set (IBSS) is that Stations grouped into a BBS are not connected to an infrastructure network. An IBSS is also referred to an ad-hoc network. All communications in an ad-hoc network are peer to peer. Any station can establish or reset a link with any other stations without the attendance of a centralized access point or base station.

BSSs are interconnected by distributed system (DS) through access point (AP). AP is an addressable station similar to a base station in a cellular network. DS functions as a backbone network that is responsible for delivery of MAC service data units to extend the network coverage.

Multiple BSSs and DSs consist the extended service set (ESS). ESS makes the entire network looks like an independent basic service set to the logical link control layer. An ESS can also provide gateway access for wireless users into a wired network such as the Internet. Such function is achieved by using of a portal, which is a logical integration between 802.11 LANs and other wired or wireless network.

1.2.2 IEEE 802.11 MAC Frame Format

IEEE 802.11 supports three types of frames[14]: management, control, and data. The management frames are used for station association and disassociation with the AP, timing and synchronization, and authentication and de-authentication. Control frames are used for handshaking and positive acknowledgement (ACK). Data frames are used to transmit data and can be combined with ACK. IEEE 802.11 frame formats are shown in fig. 1.1:

Protocol version	Type	Subtype	To DS	From DS	More flag	Retry	Pwr. mgt.	More data	WEP	Order
2 bits	2 bits	4 bits	1 bit	1 bits	1 bit	1 bit	1 bit	1 bit	1 bit	1 bit

IEEE 802.11 MAC frame –frame control

Frame control	Duration ID	Address	Address	Address	Sequence control	Address	Frame body	CRC
2 bytes	2 bytes	6 bytes	6 bytes	6 bytes	2 bytes	6 bytes	0-2312 bytes	4 bytes

IEEE 802.11 MAC frame

Figure 1.1: IEEE 802.11 MAC frame format

The protocol version field carries the version of the 802.11 standard. Type and subtype fields determine the function of frame. To DS and From DS field specifies the direction a frame enters or exits the distributed system. More flag deal with the fragmentation of the frame. Retry is set to 1 if this frame is a retransmission. Power management indicates the power save mode. More data field indicates the following frames. WEP indicates the encryption algorithm. Order field indicates whether the frame must be strictly ordered.

The frame body is a variable-length field consisting of the data payload and 7 octets for encryption. The following 6-octet address fields are used to identify the basic service set, the destination station, the source address, and the receiver and transmitter addresses. Duration field indicates the time the channel will be allocated for successful transmission of a frame. CRC field is for error detection.

1.2.3 Medium Access Control Protocol

The IEEE 802.11 MAC protocol is carrier sense multiple access with collision avoidance (CSMA/CA). It is similar to the carrier sense multiple access with collision detection (CSMA/CD) protocol which is employed by Ethernet.

In CSMA/CD protocol, stations listen to the medium before transmission to determine if the channel is idle. If the medium is idle, they transmit their frames. Collision could happen if two or more station sense the medium to be idle and transmit at the same time. In this case, stations will use back off algorithm to reschedule their transmissions.

In CSMA/CA protocol, if the channel is sensed idle for an amount of time equal to or greater than the distributed inter frame space (DIFS)[15], a station is then allowed to transmit. When a intended receiving station has correctly and completely received a frame, it waits a time of short inter frame space (SIFS) and then sends an explicit acknowledgment frame back to the sender. Unlike CSMA/CD, this explicit ACK is needed because the sender itself can not determine whether the frame is successful.

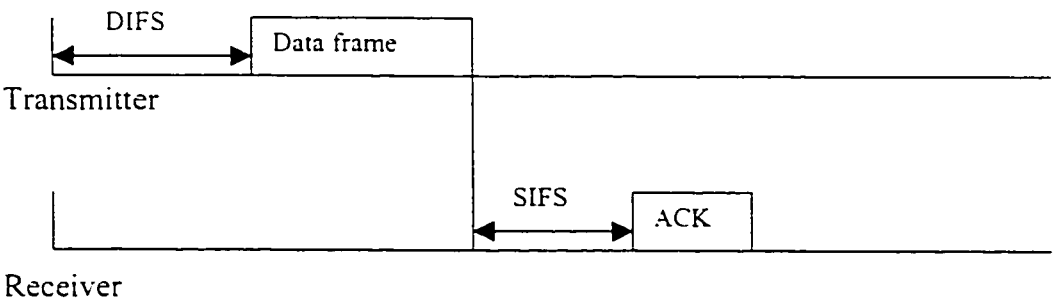


Figure 1.2 Data transmission and ACK in IEEE 802.11 without RTS/CTS

By using of Request To Send (RTS) and Clear To Send (CTS) frames, CSMA/CA can reserve access to the medium. RTS and CTS frames contain a duration field that defines the period of time that the medium is to be reserved for the transmission of the data frame and the returning ACK frame. This also helps to reduce the "hidden node" problem[26].

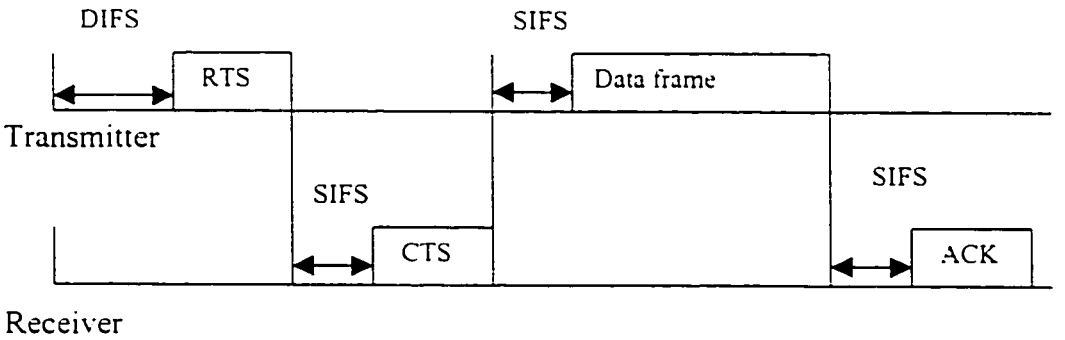


Figure 1.3 Data transmission and ACK in IEEE 802.11 with RTS/CTS

Priority access to the wireless medium is controlled through the use of inter frame space. Stations only required to wait a SIFS before transmission have highest priority access to the medium. Basic access requires stations to wait for DIFS idle time. The source station maintains control of the channel throughout the transmission of frame by waiting only an SIFS period after receiving an ACK and transmitting the next fragment.

In the discussion above, only some of the key aspects of the IEEE 802.11 are highlighted. These features are foundation of designing upper layer routing protocols used for wireless ad-hoc networks.

Chapter 2

Modified Zone-Based Hierarchical Link State Routing

The routing protocols designed for wired networks can not be used for wireless ad hoc networks because of the mobility of networks. Different characteristics of these networks have resulted in different solutions to the routing problems. This chapter gives a brief survey to existing routing protocols for ad hoc networks in section 2.1. Following that, a modified zone hierarchical link state routing is presented in section 2.2. Finally in section 2.3, some parameters are defined for evaluation of routing performance.

2.1 Wireless Ad Hoc Routing Protocols

In wireless ad hoc networks, all nodes forward packets to other nodes and take part in the discovery and maintenance of routes to other nodes. The network is dynamically self-organizing and self-configuring. Different routing protocols use a variety of different routing algorithms and approaches. Based on the network structure and when and how the routes are discovered, these routing protocols can be divided into following categories[19]: 1, table-driven routing (proactive routing); 2, on-demand routing (reactive routing); 3, flooding and limited flooding routing techniques; 4, hierarchical routing.

In table-driven routing protocols, each node maintains one or more tables containing routing information to every other node in the network. All nodes update these tables so that they maintain a consistent and up-to-date knowledge of link topology of the network. When the network topology changes, the nodes propagate update messages throughout the network so as to update their routing information database. These routing protocols

differ in the method by which the topology update information is distributed across the network and the number of necessary routing-related tables.

In the past, routing was based on shortest path routing algorithms, such as the distributed Bellman-Ford (DBF) [16] algorithm. These algorithms suffer from slow convergence in rapidly changing topology (count-to-infinity problem). Besides, DBF-like algorithms incur large update message penalties.

Based on DBF, protocols such as destination sequenced distance vector routing protocol (DSDV)[2][17][18], and global state routing protocol (GSR)[3] are proposed to eliminate the problems that occur in DBF. Link state protocols converge more rapidly than distance vector protocols, but they do so at the expense of significantly more control traffic. Optimized link state protocol (OLSP)[4] utilizes a multicast-like mechanism to reduce the amount of control traffic. Protocols like WRP[5] are able to eliminate the count-to-infinity problem and reduce the temporary loops with less control traffic overhead than distance vector schemes.

In contrast to table-driven routing protocols, all up-to-date routes are not maintained at every node, instead the routes are created when required. When a source wants to send to a destination, it invokes the route discovery mechanisms to find the path to the destination. Typically route discovery relies on flooding of queries. The route remains valid till the destination is reachable or until the route is no longer needed.

Temporally ordered routing algorithm (TORA)[6] is prorosed for highly dynamic multihop wireless networks. The control messages are localized to a very small set of nodes near the occurrence of a topology change.

Protocol such as dynamic source routing (DSR)[7] and ad hoc on demand distance vector (AODV)[8] unicast the route reply back to the querying source along a path specified by a sequence of node addresses accumulated during the route query phase. In the case of DSR, the node addresses are accumulated in the query packet and are returned

to the source to be used for source routing. AODV distributes the discovered routes in the form of next hop information stored at each node in the route. Similarly, associativity based routing (ABR)[9] records link stability for each link that a query packet traverses so as to find longer-lived routes for networks.

The on demand discovery of routes can result in much less traffic than standard distance vector or link state schemes, especially when innovative route maintenance schemes are employed. However, delay of routing discovery may lead to significant delay of packet delivery and the reliance on packet flooding may still lead to considerable control traffic on highly dynamic network.

In flooding and limited flooding techniques, all nodes carry the same responsibility; a node hearing a data packet rebroadcasts to all neighbors and so on. To limit the amount of overhead traffic, limited flooding techniques are proposed[19].

In hierarchical routing protocols, the detail of the network topology is concealed by aggregating nodes into clusters and clusters into super clusters and so on[21]. Two kinds of nodes exist, end points and switches. End points select the switches (similar to the base station 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. Switch nodes have a higher computation and communication burden than other nodes. Both proactive and reactive schemes can be implemented at different levels.

Proactive protocols attempt to continuously determine the network connectivity so that the route is already available before required. The advantage of the proactive scheme is that when a route is needed there is little delay until the route is determined. But more control traffic is required to continuously keep the routing information current.

Reactive protocols, on the other hand, invoke a route discovery procedure only on demand. Because route information may not be available at the time a route request is received, the delay to determine a route can be quite significant. Further more, the global search procedure of the reactive protocols requires significant control traffic.

Because of the delay and excessive control traffic, pure reactive or proactive routing protocols are likewise not appropriate for the dynamic wireless network. Protocols such as zone routing protocol (ZRP)[10][11] integrating the two different classes of traditional schemes are proposed in recent years.

Mario Joa-NG proposed a peer-to-peer zones-based two-level link state routing (ZHLS)[12] for mobile ad hoc networks. It is proactive if the destination is within the same zone of the source and it is reactive if the destination in a different zone.

Following a brief introduction to some typical routing techniques and the performance of those techniques, a modified ZHLS routing protocol will be presented in the last section of this chapter.

2.1.1 Destination-Sequenced Distance-Vector Routing (DSDV)

DSDV is based on the Routing Information Protocol[17], RIP used within the internet. Each node keeps a routing table containing routes (next hop nodes) to all possible destination nodes in the wireless network. Data packets are routed by consulting this routing table. Each entry in the table contains a destination node address, the address of the next hop node in route to this destination, the number of hops (metric) between destination and next hop node, and the lifetime of this entry (called sequence number). Each node must periodically transmit its entire routing table to its neighbors using update packets. Build up and update of routing table is done via the update packets.

The computer simulations in [18] are conducted for 5 kinds of routing techniques including DSDV(Fig. 2.1 to 2.5). The network consists of 200 nodes. Data packets were generated from a Poisson kind of traffic with inter-arrival period of 2.5 seconds

amounting to a data rate of 4Kbps per source/destination pair. Each data packet consists of 10K bits, buffer size at each node is 15 packets, nodes moved at a speed of 60km/hr, maximum radio range is 120m and data rate on channel is 2Mbps.

Performance of DSDV is shown in fig. 2.1 to 2.5. The overhead performance is worse than FSR, HSR...etc. Fig.2.2 shows that the DSDV overhead performance versus speed of nodes is prohibitive. Fig. 2.1 and 2.2 shows that the DSDV performance is somewhat independent of the nodes traffic conditions, this is due to the fact that all nodes are busy anyhow with the transmission of their routing tables. DSDV exhibits good delay and average number of hops (Fig. 2.3 and 2.4). However, for highly deployed networks, where the number of nodes increases as the area enlarges, the overhead ratio is much worse than FSR, HSR...etc (Fig. 2.5). Figs 2.1 to 2.5 also show the performance of simulated on-demand routing techniques such as ADOV and DSR. In type A and B, the on-demand routing tables are timeout every 3 and 6 second, respectively.

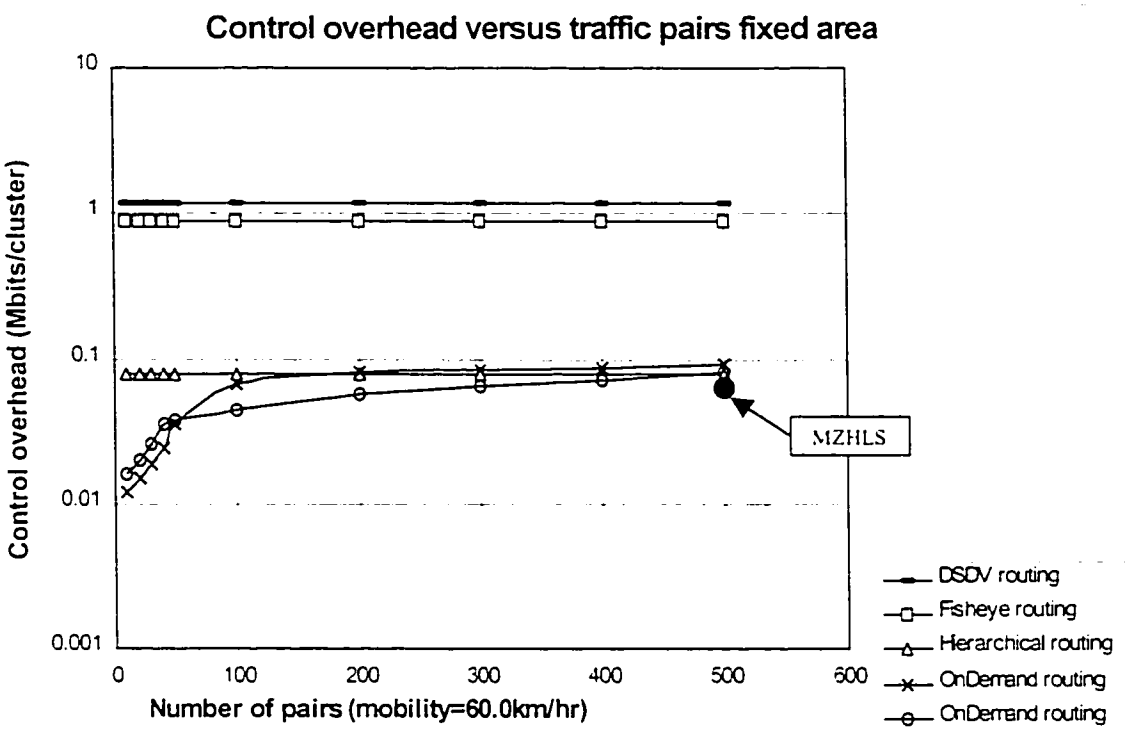


Figure 2.1: Control overhead versus traffic pairs fixed area

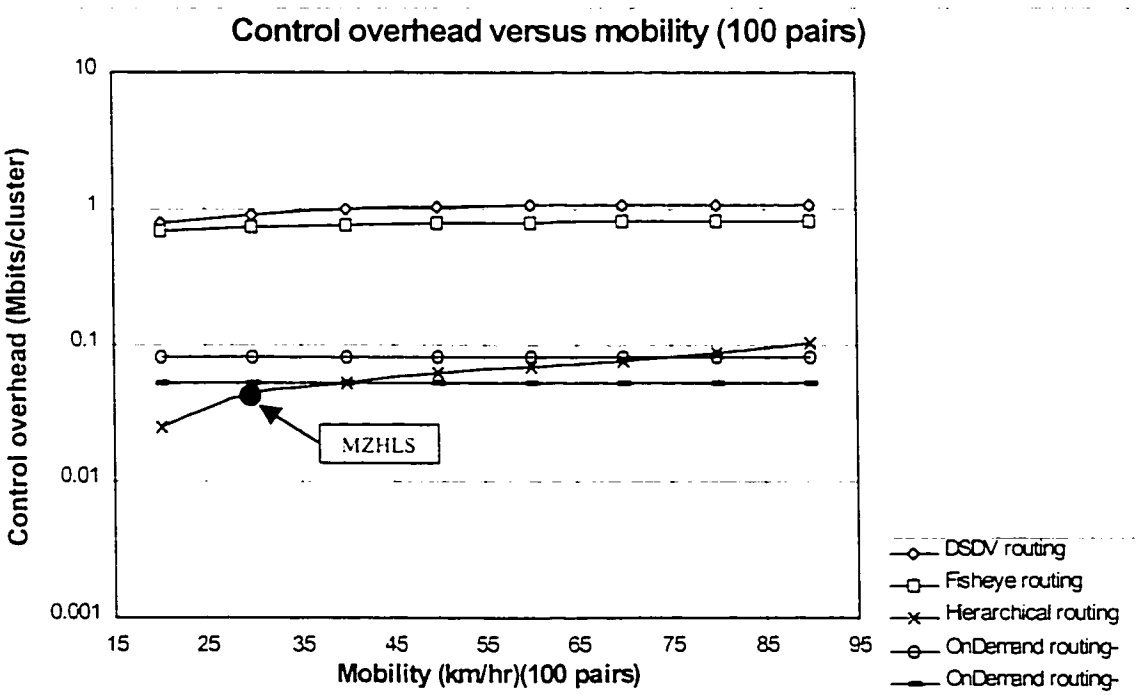


Figure 2.2: Control overhead versus mobility (100 pairs)

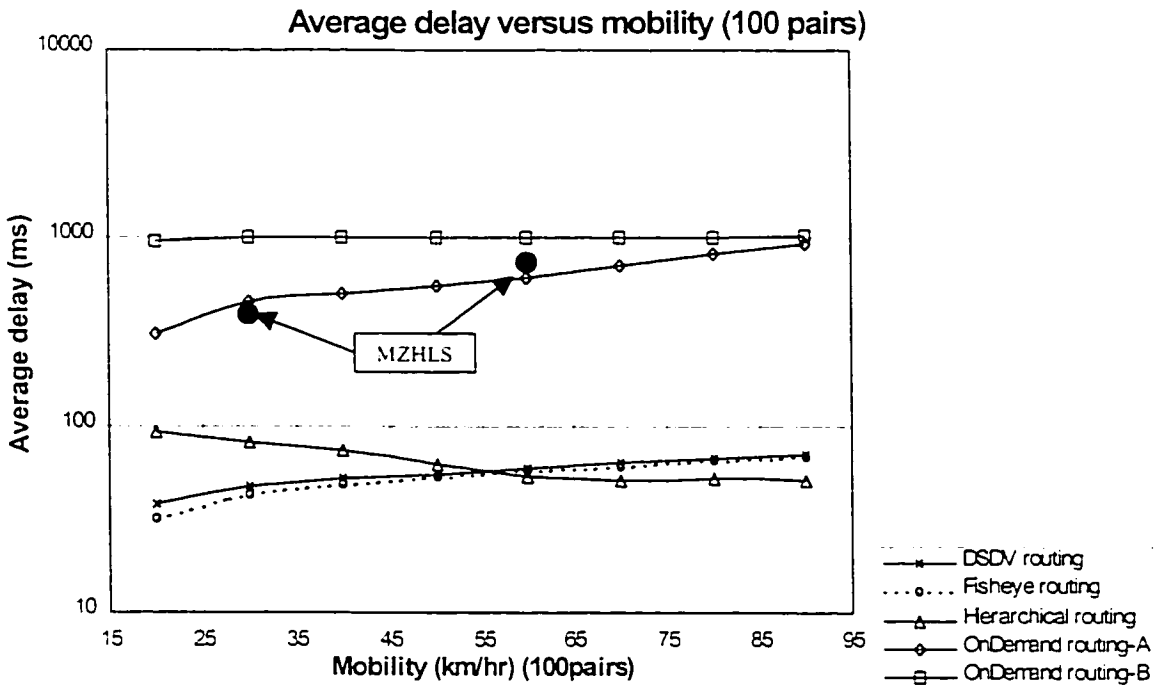


Figure 2.3: Average delay versus mobility (100 pairs)

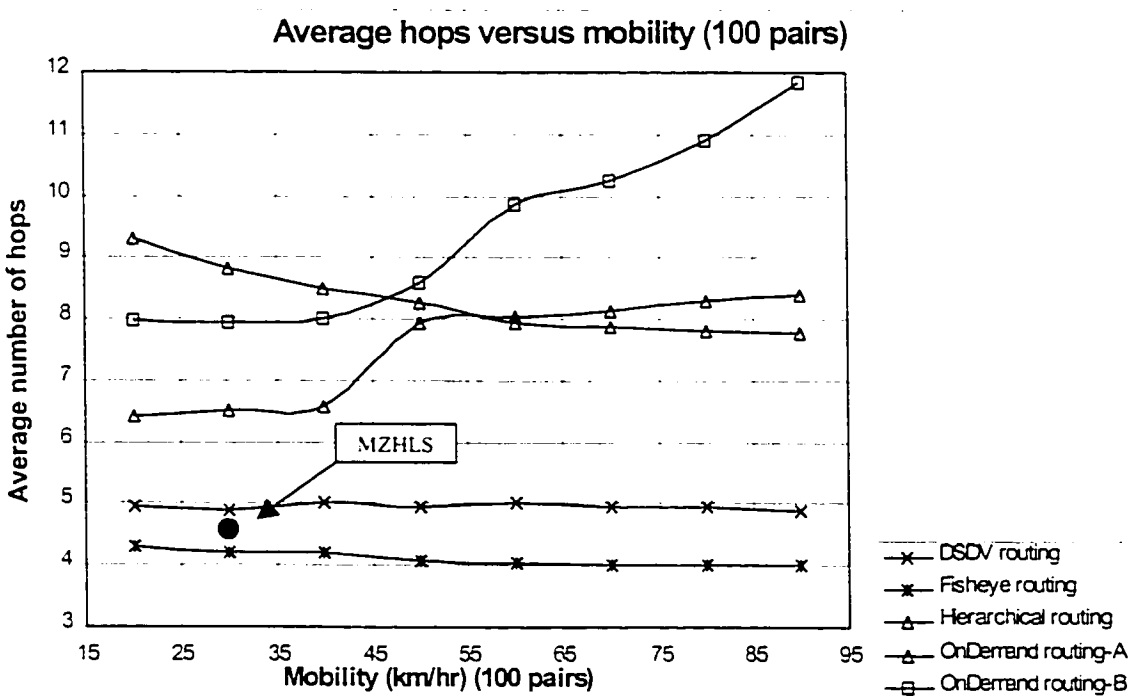


Figure 2.4: Average hops versus mobility (100 pairs)

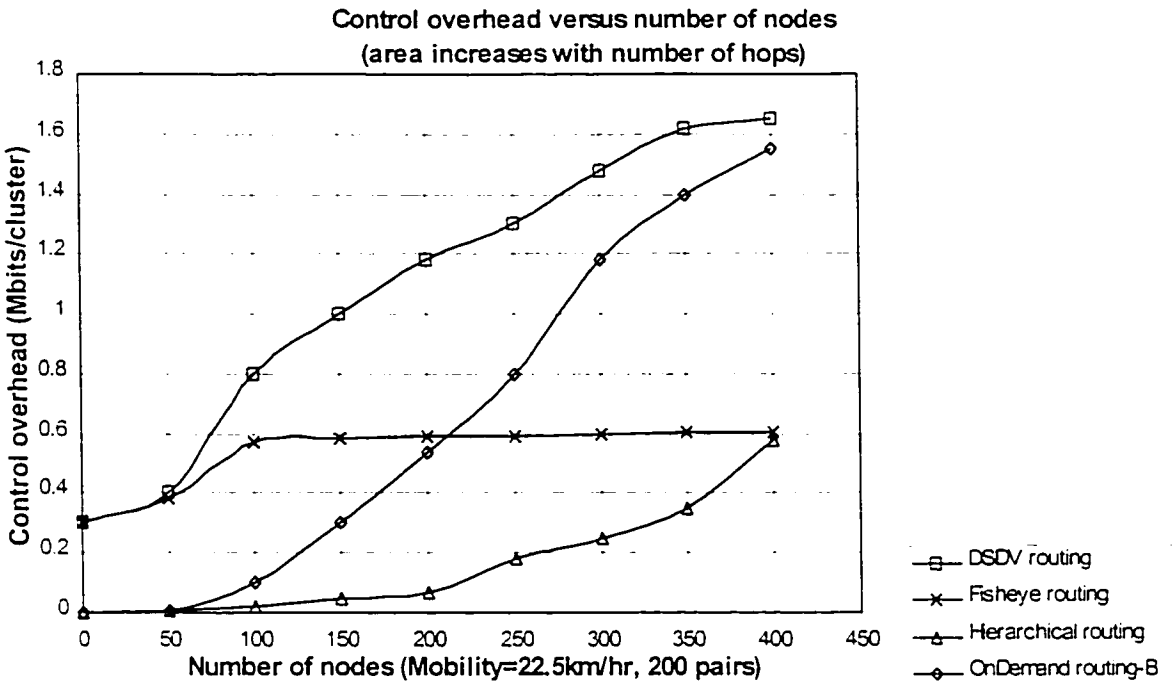


Figure 2.5: Control overhead versus number of nodes

2.1.2 Fisheye Routing (FSR)

In FSR[18][20], each update message does not contain information about all nodes. Instead, it exchanges information about closer nodes more frequently than it does about farther nodes thus reducing the update message size. So each node gets accurate information about neighbors and the detail and accuracy of information decreases as the distance from node increases. Figure 2.6 defines the scope of fisheye for the center node. The scope is defined in terms of the nodes that can be reached in a certain number of hops. The center node has most accurate information about all nodes in the white circle (1st scope) and so on. Even though a node does not have accurate information about distant nodes, the packets are routed correctly because the route information becomes more and more accurate as the packet moves closer to the destination. FSR scales well to large networks as the overhead is controlled in this scheme.

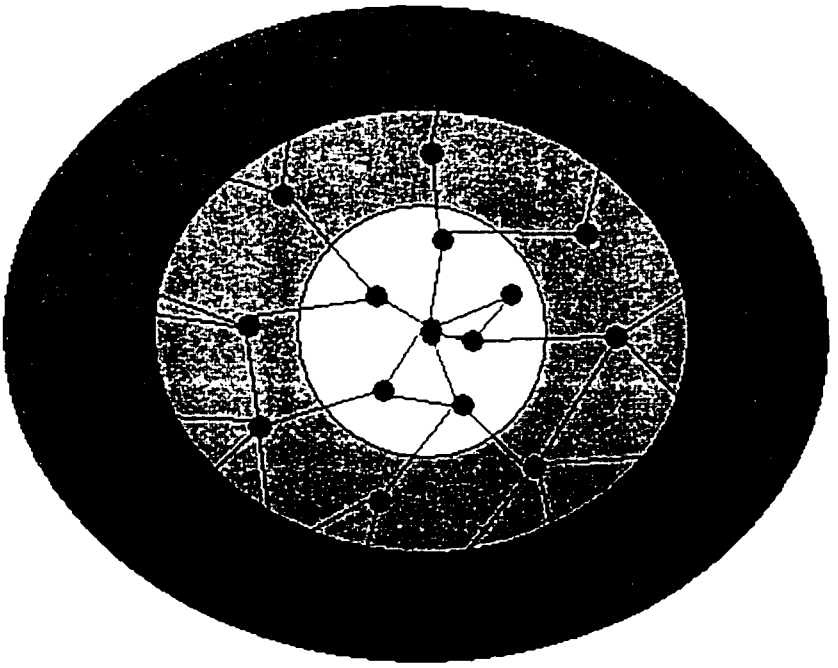


Figure 2.6: Accuracy of information in FSR

Performance of FSR is also shown in fig. 2.1 to 2.5. The routing tables are refreshed at a rate of once every 2 seconds for 1st scope and once every 6 seconds for 2nd scope nodes.

Fig. 2.1 and 2.2 show that the routing overhead (control) per cluster does not change much with active nodes number or their mobility (speeds). However, FSR control overhead is relatively higher than the other routing techniques.

Fig.2.3 shows that the average data packet delay does not rise much with nodes mobility, also FDR performance is the best in this regard compared to other techniques. Similar results are drown from fig. 2.4 which shows the average number of hops traveled by a typical data packet.

Fig. 2.5 shows that the FSR overhead ratio does not increase as much as other techniques as the network size increases.

2.1.3 Hierarchical State Routing (HSR)

The characteristic feature of Hierarchical State Routing (HSR)[18] is multilevel clustering and logical partitioning of mobile nodes. The network is partitioned into clusters and a cluster-head elected as in a cluster-based algorithm[22][23]. In HSR, the cluster-heads again organize themselves into clusters and so on. The nodes of a physical cluster broadcast their link information to each other. The cluster-head summarizes its cluster's information and sends it to neighboring cluster-heads via a gateway node. As shown in the figure 2-7, these cluster-heads are member of the cluster on a level higher and they exchange their link information as well as the summarized lower-level information among each other and so on. A node at each level floods to its lower level the information that it obtains after the algorithm has run at that level. So the lower level has a hierarchical topology information. Each node has a hierarchical address. One way to assign hierarchical address is the cluster numbers on the way from root to the node as shown in figure 2.7. A gateway can be reached from the root via more than one path, so gateway can have more than one hierarchical address. A hierarchical address (a sequence of MAC addresses) is enough to ensure delivery from anywhere in the network to the host.

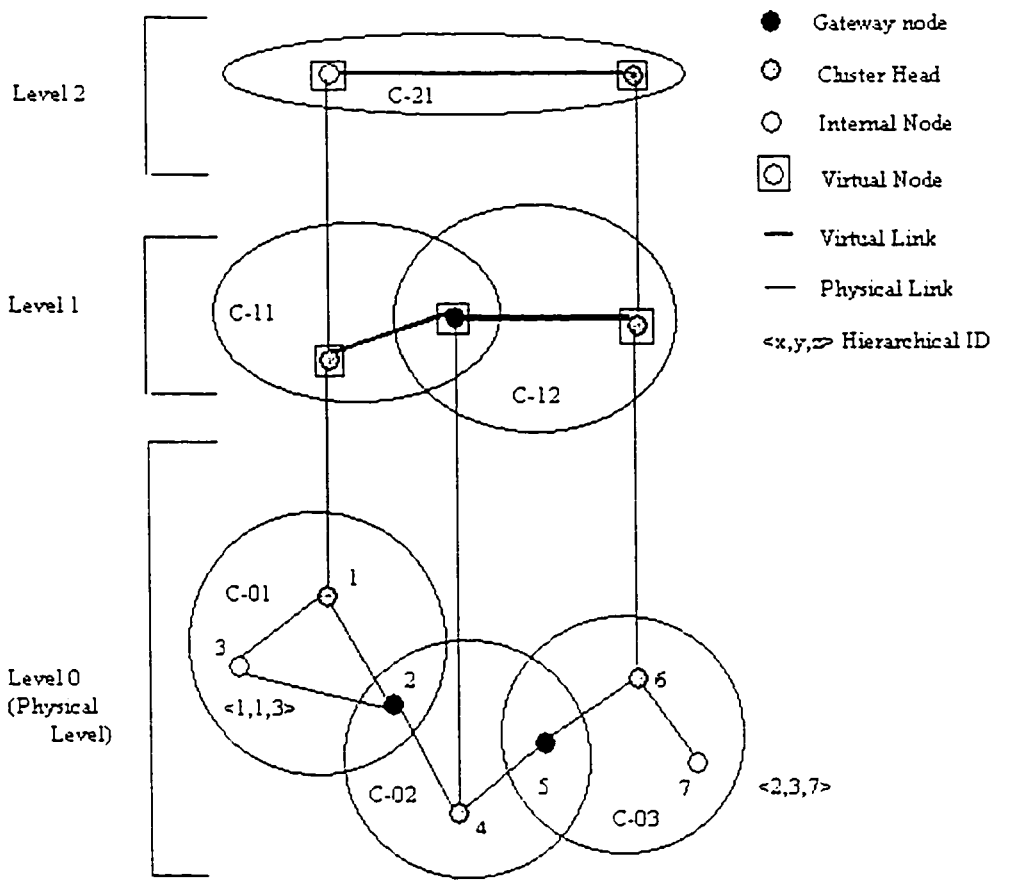


Figure 2-7: An example of physical/virtual clustering in HSR

In addition, nodes are also partitioned into logical sub-networks and each node is assigned a logical address <subnet, host>. Each sub-network has a location management server (LMS). All the nodes of that subnet register their logical address with the LMS. The LMS advertise their hierarchical address to the top levels and the information is sent down to all LMS too. The transport layer sends a packet to the network layer with the logical address of the destination. The network layer finds the hierarchical address of the destination's LMS from its LMS and then sends the packet to it. The destination's LMS forwards the packet to the destination. Once the source and destination know each other's hierarchical addresses, they can bypass the LMS and communicate directly. Since logical address/hierarchical address is used for routing, it is adaptable to network changes.

Performance of HSR is shown in fig. 2.1 to 2.5. Refresh rate of routing tables is 2 seconds. Fig. 2.1 shows that overhead ratio in HSR is relatively constant as the number of communication pairs changes. Also the overhead ratio in HSR is better than that of FSR, DSDV. Fig. 2.2 shows that HSR overhead ratio is better than FSR, DSDV as the nodes increase their speed. Fig. 2.3 also show better delay for HSR, but the average number of hops in fig. 2.4 does not reveal marked improvement compared to other routing techniques. Finally, fig. 2.5 shows a superior overhead ratio performance of HSR, as the number of communication nodes rises (while keeping the user density the same).

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

Ad hoc On-demand Distance Vector Routing (AODV) [8][24] is an improvement on the DSDV algorithm discussed in section 2.1.1. AODV combines proactive and reactive scheme thus to minimize the number of broadcasts by creating routes on-demand as opposed to DSDV that maintains the list of all the routes.

To find a path to the destination, the source broadcasts a route request packet (shown in fig. 2.8a). The neighbors in turn broadcast the packet to their neighbors till it reaches an intermediate node that has a recent route information about the destination or till it reaches the destination (Fig. 2.9a). A node discards a route request packet that it has already seen. The route request packet uses sequence numbers to ensure that the routes are loop free and to make sure that if the intermediate nodes reply to route requests, they reply with the latest information only.

When a node forwards a route request packet to its neighbors, it also records in its tables the node from which the first copy of the request came. This information is used to construct the reverse path for the route reply packet (shown in fig. 2.8b). AODV uses only symmetric links because the route reply packet follows the reverse path of route request packet. As the route reply packet traverses back to the source (Figure 2.9b), the nodes along the path enter the forward route into their tables.

If the source moves then it can reinitiate route discovery to the destination. If one of the intermediate nodes move then the moved node's neighbor realizes the link failure and sends a link failure notification to its upstream neighbors and so on till it reaches the source upon which the source can reinitiate route discovery if needed.

Source	Destination	Source sequence number	Destination sequence number	Number of hops	Previous hop

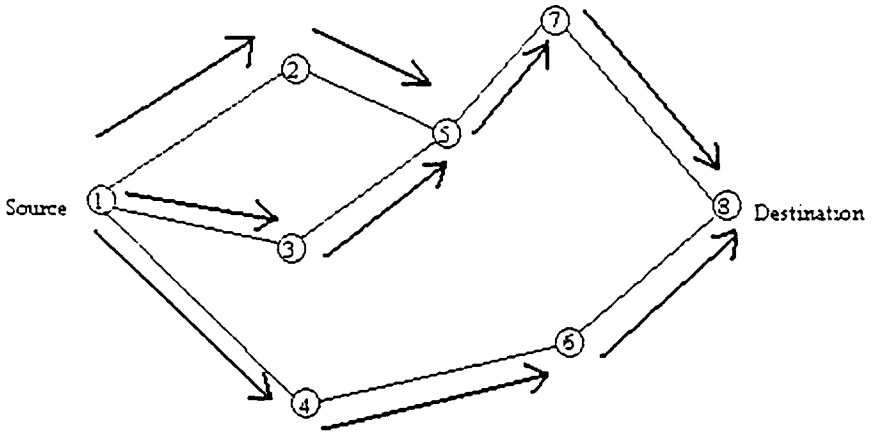
(a) Route request packet

Reply to	Source	Destination	Destination sequence number	Number of hops	Previous hop

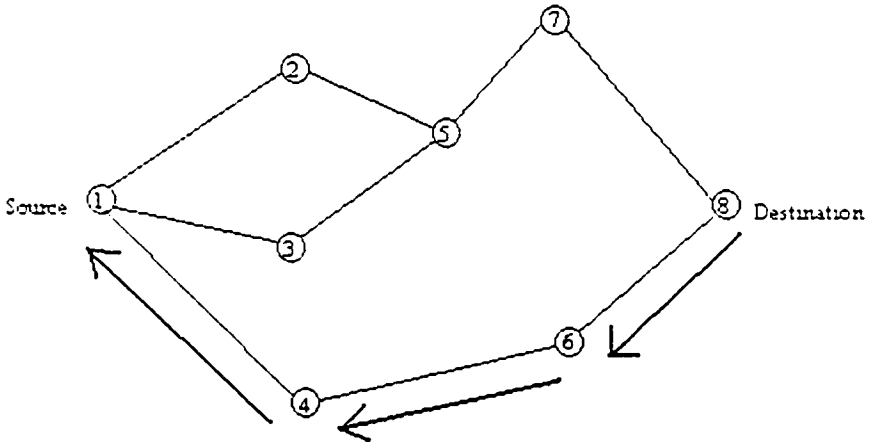
(b) Route reply packet

Figure 2.8: Typical packet format in AODV

Note: AODV has other types of packets such as "hello" packet, "RERR" packet, etc.



(a) Propogation of Route Request (RREQ) Packet



(b) Path taken by the Route Reply (RREP) Packet

Figure 2.9: Route discovery in AODV

Fig.2.10 and 2.11 show the simulation results in [24] namely the control and data performances of AODV. The average mobility is the node speed averaged over all nodes and all simulation iterations (time). The number of "hello" packets does not change much with mobility as compared to numbers of route request and route reply packets. Comparing fig. 2.10 and 2.11, it seems that the number of control (routing) packets are higher than that of data packets.

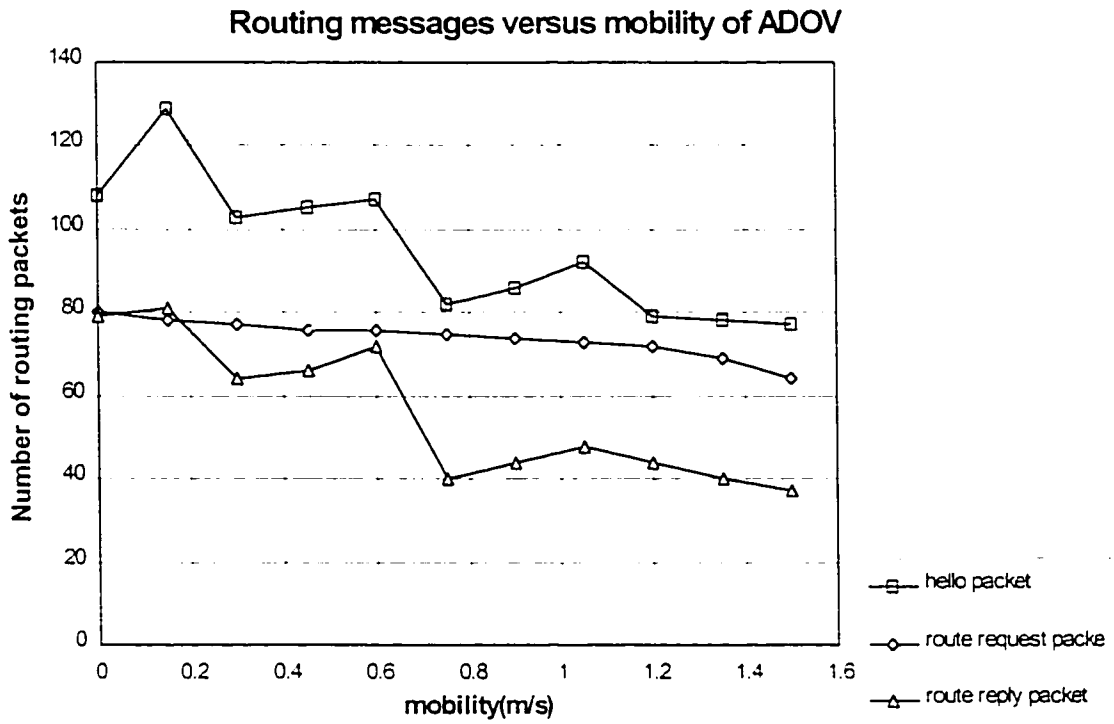


Figure 2-.10: Routing messages versus mobility of ADOV

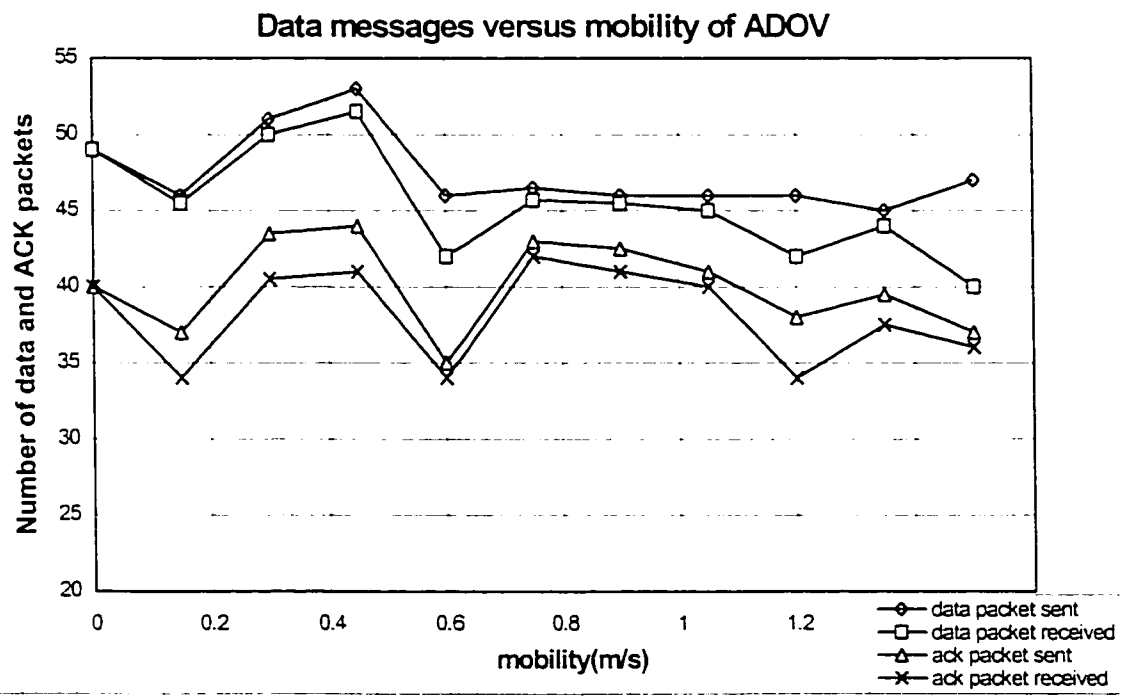


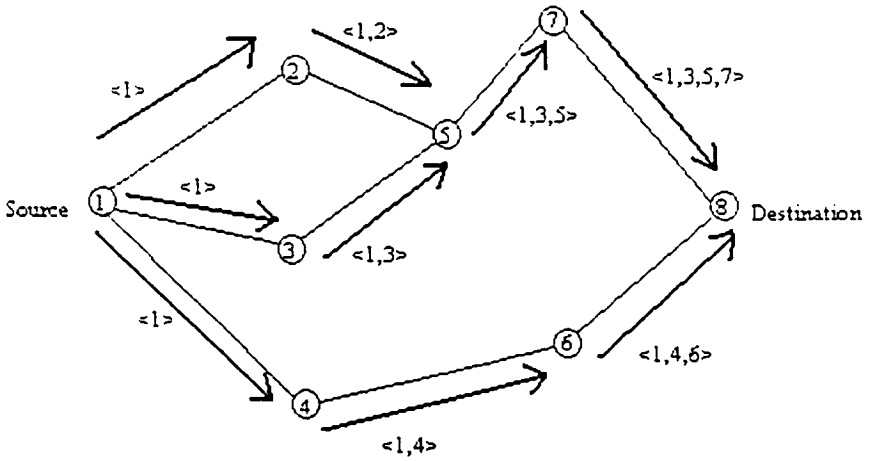
Figure 2.11: Data messages versus mobility of ADOV

2.1.5 Dynamic Source Routing (DSR)

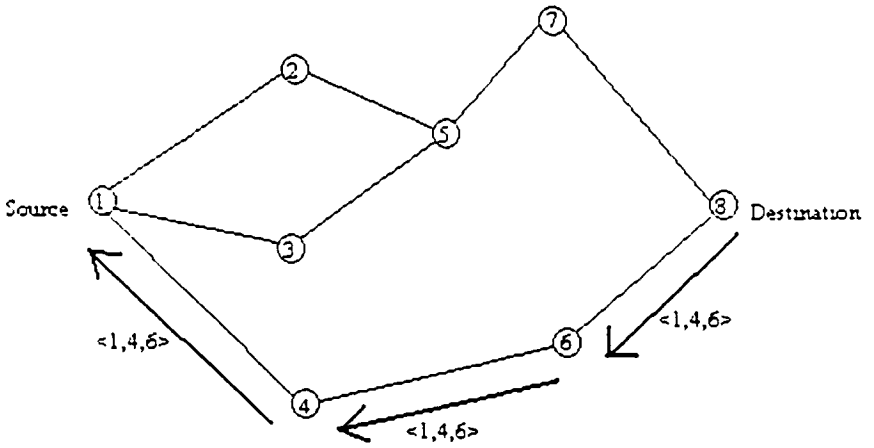
The Dynamic Source Routing Protocol[7][25] is a source-routed on-demand routing protocol. A node maintains route caches containing the source routes that it is aware of. The node updates entries in the route cache as and when it learns about new routes.

The two major phases of the protocol are: route discovery and route maintenance. When the source node wants to send a packet to a destination, it looks up its route cache to determine if it already contains a route to the destination. If it finds that an unexpired route to the destination exists, then it uses this route to send the packet. But if the node does not have such a route, then it initiates the route discovery process by broadcasting a route request packet. The route request packet contains the address of the source and the destination, and a unique identification number. Each intermediate node checks whether it knows of a route to the destination. If it does not, it appends its address to the route record of the packet and forwards the packet to its neighbors. To limit the number of route requests propagated, a node processes the route request packet only if it has not already seen the packet and it's address is not present in the route record of the packet.

A route reply is generated when either the destination or an intermediate node with current information about the destination receives the route request packet. A route request packet reaching such a node already contains, in its route record, the sequence of hops taken from the source to this node.



(a) Building Record Route during Route Discovery



(b) Propogation of Route Reply with the Route Record

Figure 2.12: Creation of record route in DSR

As the route request packet propagates through the network, the route record is formed as shown in fig. 2.12(a). If the route reply is generated by the destination then it places the route record from route request packet into the route reply packet. On the other hand, if the node generating the route reply is an intermediate node then it appends its cached route to destination to the route record of route request packet and puts that into the route reply packet. Fig. 2.12(b) shows the route reply packet being sent by the destination itself. To send the route reply packet, the responding node must have a route to the

source. If it has a route to the source in its route cache, it can use that route. The reverse of route record can be used if symmetric links are supported. In case symmetric links are not supported, the node can initiate route discovery to source and piggyback the route reply on this new route request.

DSR uses two types of packets for route maintenance: Route Error packet and Acknowledgements. When a node encounters a fatal transmission problem at its data link layer, it generates a Route Error packet. When a node receives a route error packet, it removes the hop in error from it's route cache. All routes that contain the hop in error are truncated at that point. Acknowledgment packets are used to verify the correct operation of the route links. This also includes passive acknowledgments in which a node hears the next hop forwarding the packet along the route.

Performance of DSR was simulated in [25]. A network is consisted of 50 nodes that move at variable speed from 0 to 20m/s. Routing overhead of DSR, e.g., is high but can be reduced at the expense of memory and processing time of nodes (in fig. 2.13, 0 represents speed 20m/s and 900 represents speed 0m/s). Yet the greatest overhead loss of DSR is attributed to the long route path in each packet.

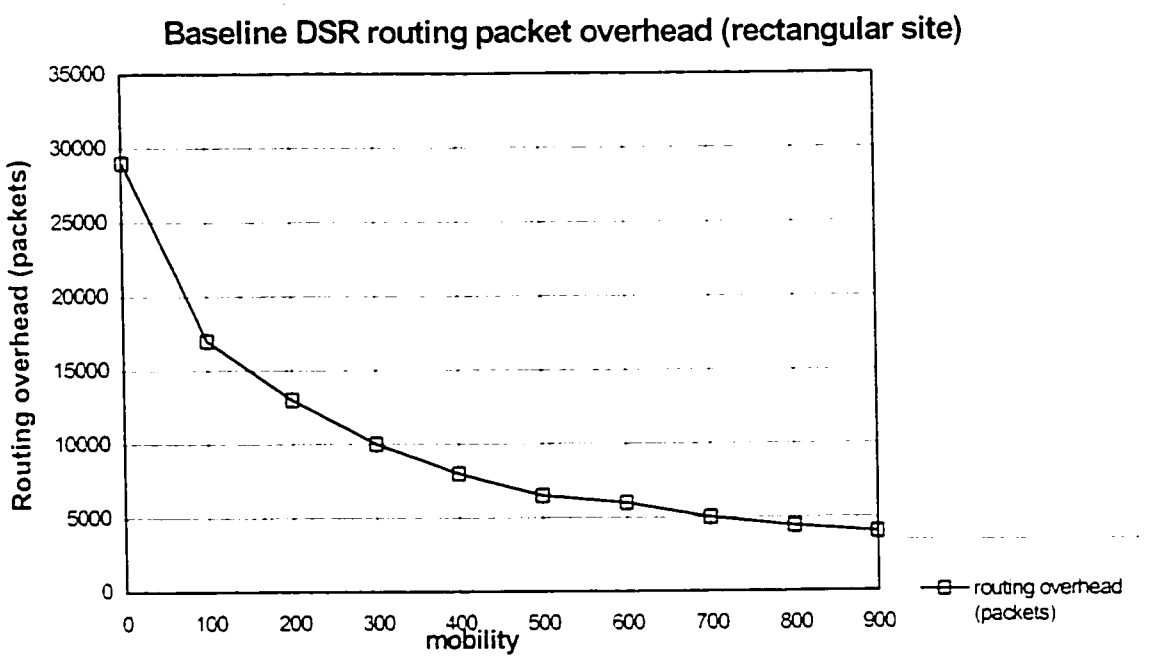


Figure 2.13: Routing packet overhead of DSR

2.1.6 Zone Routing Protocol (ZRP)

Zone routing protocol[10][11] is a zone or cluster based routing protocol that is a hybrid between proactive and reactive routing. It is targeted for very large networks and divides the network into zones or cluster of nodes. The nodes within a zone are close to one another. Use of proactive routing is advocated within a zone and reactive routing across zones.

The approach of ZRP to routing is based on the notion of a routing zone, which is defined for each node and includes the nodes whose distance is equal to or less than a predefined number of hops. This distance is referred to the zone radius, r_{zone}. Each node is required to know the topology of the network within its routing zone only and nodes are

updated about topological changes only within their routing zone. The routes within the network are specified as a sequence of nodes separated by approximately the zone radius.

Route Discovery in ZRP is illustrated by an example in fig. 2.14. First, the source S verifies that the destination D is not within its routing zone. Then, S sends a query to all the nodes on the periphery of its zone; i.e., C, G, and H.

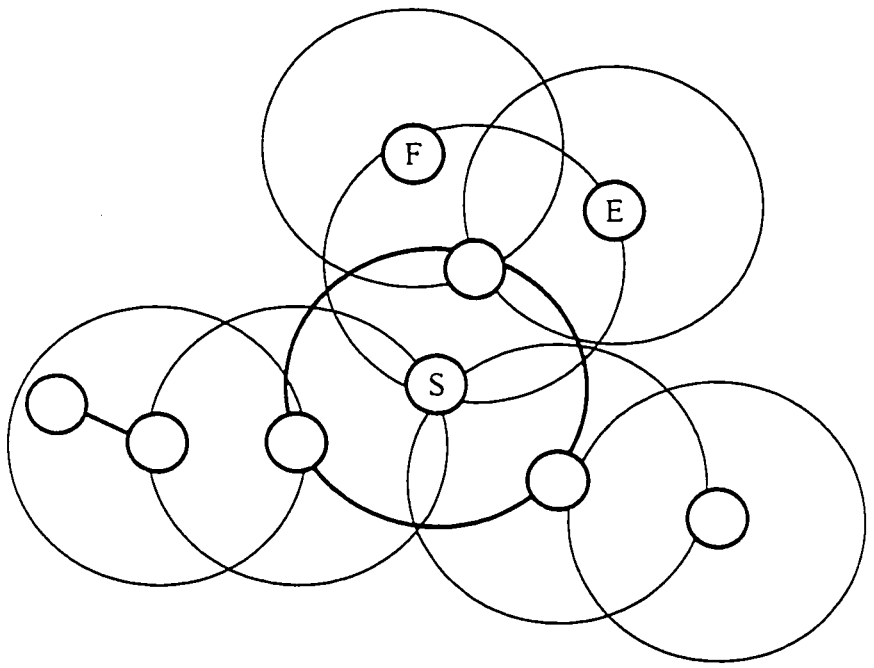


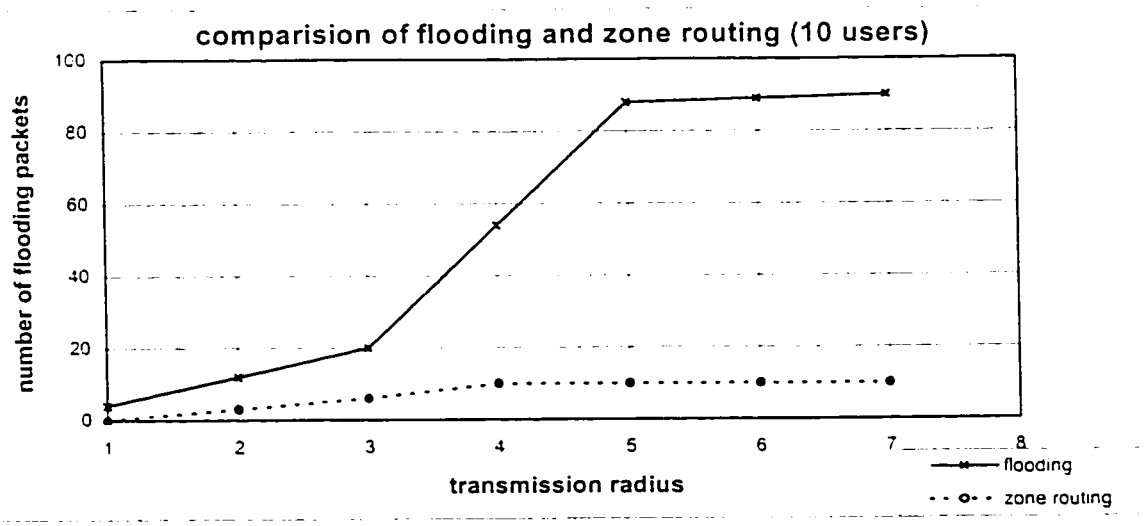
Figure 2.14: An example of Zone Routing

Now, in turn, each one of these nodes, after verifying that D is not in their routing zone, broadcast the query to their "peripheral" nodes. In particular, H sends the query to B, which recognizes D as being in its routing zone and responds to the query, indicating the forwarding path: S-H-B-D.

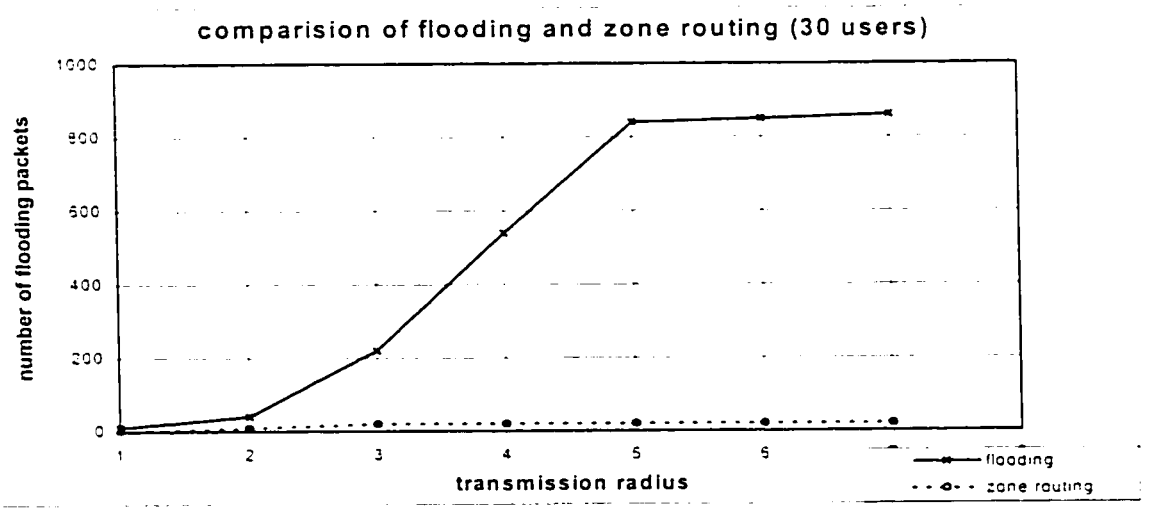
In order to limit the message size and to bound the Route Discovery process, a hop-count is included within the query messages. The value of the hop-count in the initial query message is set to some maximal value and is decreased by one, each time a query message is forwarded. When the hop-count reaches zero, the copy of the query message

is discarded. If the destination node is within maximum hop-count from the source node, the algorithm will discover at least one path between the two nodes, no matter what the value of the zone radius is.

Fig. 2-15 show some of the simulation results in [10]. ZRP requires a smaller number of query messages than pure flooding, as these messages are routed only to peripheral nodes, omitting all the nodes within the routing zones. As the zone radius is significantly smaller than the network radius, the cost of learning the zones' topologies is a small fraction of the cost required by a global proactive mechanism. Furthermore, the amount of data stored at each node is similarly reduced. On the other hand, ZRP is faster than a global reactive route discovery mechanisms, as the number of nodes queried in the process is on the order of $(r_{zone}/r_{net})^2$ of the number of nodes queried by a global flooding process [10]. Additionally, ZRP discovers multiple routes to the destination.



(a) Comparison of flooding and zone routing for a 10 user ad hoc network



(b) Comparison of flooding and zone routing for a 10 user ad hoc network

Figure 2.15: Comparison of the number of control messages in flooding and ZRP

The Route Discovery process in ZRP can be made more efficient in resources, at the expense of longer latency. Instead of querying simultaneously all the peripheral nodes at the boundary of the routing zone, these nodes can be queried either sequentially, one-by-one, or in groups. Thus, there is a tradeoff between the cost and latency of the ZRP Route Discovery protocol.

ZRP is a good example of an ad-hoc routing technique that tries to strike the balance between pro-activity and re-activity. Optimizations to the ZRP protocol can be incorporated. However, it was noted in [11] that accuracy of the optimal zone radius is a very complex function of too many parameters of underlying wireless internet. The description of the mechanisms and more simulation results are presented in [11].

2.2 Modified Zone Based Hierarchical Link State Routing

In zone-based hierarchical link state routing protocol, the wireless network is divided into non-overlapping zones. Initially, each node knows its own position and therefore zone ID through global positioning system (GPS). After the network is established, each node knows the low level (node level) topology about zone connectivity of the whole network. A packet is forwarded by specifying the hierarchical address--zone ID and node ID of a destination node in the packet header. The high level topological information is distributed to all nodes. To find the zone ID of a new destination, a location search is needed. Modified zone hierarchical link state routing (MZHLS) keeps all these features unchanged while differs from ZHLS in routing algorithm at both node level and zone level. Location search is performed by unicasting one location request to one specific zone. Normally these specific zones are peripheral zones. Because all zones have same potentiality whether peripheral zones or on the paths to peripheral zones from certain source node, all zones don't have to be searched before a destination is found. By searching fewer zones than ZHLS, location search delay may be reduced.

2.2.1 Network Structure of MZHLS

The network is divided into zones in MZHLSP. A node knows its physical location by GPS and maps its physical location to a zone map. The zone size depends on factors such as node mobility, network density, transmission power and propagation characteristics.

MZHLS has a hierarchical structure where two levels of topology are defined: node level topology and zone level topology. If any two nodes are within the direct communication range, a physical link exists. The node level topology provides the information on how nodes are connected together by these physical links. In fig. 2.16, node a can send a packet to node e through node b. So a virtual link exits between node a and e. The zone level topology (Fig. 2.17) tells how the zones are connected by these virtual links. A virtual link exits zone 3 and zone 6 through zone 1 and 2.

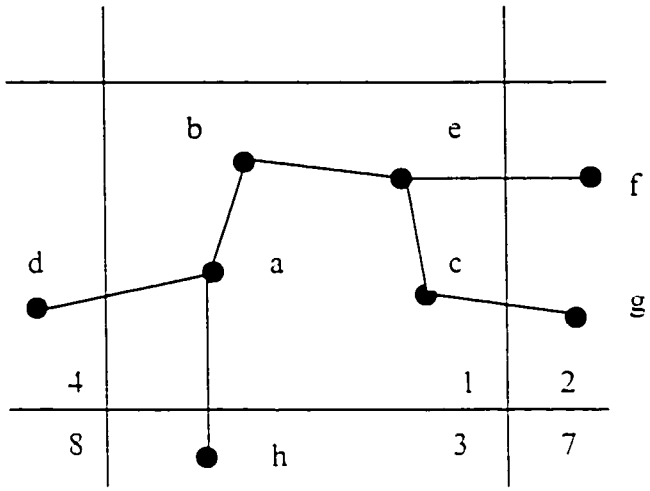


Figure 2.16: Node level topology

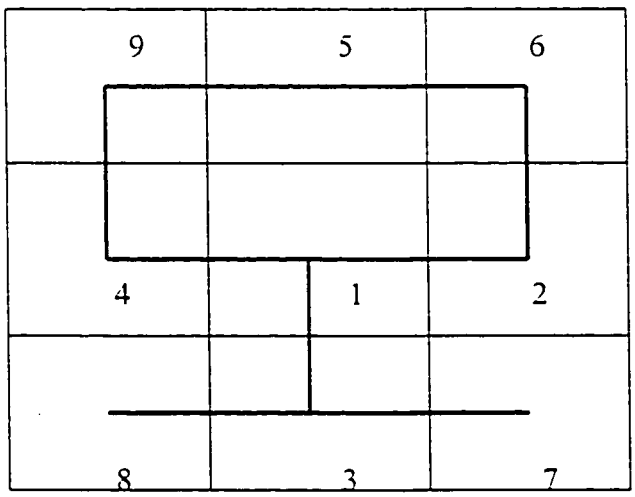


Figure 2.17: Zone level topology

Two types of control packets are used to describe the link topology of different levels: node link state packet (NLSP) and zone link state packet (ZLSP). A NLSP contains information of all direct links from a node to its neighbor nodes while a ZLSP contains information of all links within one hop from a zone to its neighbor zones.

2.2.2 Intrazone Routing

Every node periodically repeats a link procedure by broadcasting a link request packet (LKR) (On the assumption that group address of nodes is available). All neighboring nodes that hear this packet reply with link response packet (LKRS). After a certain period (defined as Link Time), all link response packets are supposed to be collected and this node will know its immediate neighboring nodes. It then includes these neighboring nodes into its direct link list and generates an NLSP. The NLSP will be flooded within the local zone.

Upon receiving NLSPs from all nodes in that zone, every node will know the node level link topology of that zone. Finally, they will build up their node level link information database as shown in table 2.18:

Source	Connect to local nodes:	Connect to nodes of different zones:
a	b	d-4, h-3
b	a, e	
c	b, c	f-2
e	e	g-2

Table 2.18: NLSP in a typical node a's database

Due to node mobility, each NLSP has a timeout counter. If the counter exceeds a certain threshold, this NLSP will be eliminated from the node's link information database or from the network.

Nodes perform shortest path routing at node level. Every node will know the shortest path to any other node within the same zone or any other neighbor node which has direct link to the same zone. It builds up its intrazone routing table as shown in table 2.19:

Destination node	Next node
b	b
d	d
c	b
e	b
f	b
g	b
h	h

Table 2.19: Intrazone routing table of a typical node a

2.2.3 Interzone Routing

Because NLSP contains not only connected node IDs but also zone IDs that the connected nodes belong to, each node knows all neighbor zones that have connection to local zone and periodically generates a ZLSP including all these neighboring zones.

Each node that has connections to nodes of different zones will periodically broadcast the ZLSP through out the whole network. Such a node acts as a gateway node. But unlike ZHLS, if two or more nodes have connections to the same zone, any of them can be randomly selected to forward packets to that zone. Delivery of interzone packets by multiple gateway nodes instead of single gateway node helps to avoid network bottleneck.

As the above procedure is repeated, all nodes will collect ZLSP from all zones of the network and include them into their ZLSP database as shown in table 2.20.

Source zone	Connect to zones:
1	2, 3, 4
2	1, 6
3	1, 7, 8
4	1,9
5	6, 9
6	2, 5
7	3
8	3
9	4, 5

Table 2.20: ZLSP in a typical node's database

Due to node mobility, different nodes of the same zone may generate different ZLSPs at the same time. Synchronization of generating ZLSP by nodes of the same zone is required. A timeout field is added to ZLSP to ensure that only one ZLSP be repeated by other nodes. If two ZLSP are received, the last one will be eliminated by the same timeout value with the same zone ID. The timeout field serves another purpose that out of order ZLSP will be dropped.

When node density is high, zone level topology changes infrequently[12]. Gateway nodes compare the new generated ZLSP and the old ZLSP in their ZLSP databases; if the two have same connection content, the new generated ZLSP will not be broadcast. Nodes will use the old ZLSP to do routing algorithm till the new ZLSP with different content replaces the old one. Thus channel capacity is saved by broadcast of less ZLSPs.

As the procedure repeats periodically, each node collects ZLSPs from all zones and knows the network topology. Shortest path algorithm based on minimum hops (or other

cost functions) is performed the first time at the zone level to find the next zone to destination zone, and the second time at the node level to find the next node to the next zone. The interzone routing table is shown in table 2.21.

Destination zone	Next zone	Next node
2	2	B
3	3	H
4	4	D
5	4	D
6	2	B
7	3	h
8	3	h
9	4	d

Table 2.21 Interzone routing table of a typical node a

2.2.4 Location Search and Routing Mechanism

Before any data transmission can start, a source node needs to find the path to the destination node. Unlike the ZHLS, in modified ZHLS, routing is performed at three steps. In case of continuous or frequent data transmission, location search is not necessary to be performed for every packet. Based on this fact, each node records the last location (destination zone ID) of the destination node as a historic path. At the first step, the source node checks its history path database to find if the destination zone ID is available. If yes, the source node extracts the next node ID to that zone from its interzone routing table, attach both destination zone ID and next node ID to the data packet. All intermediate nodes will route this packet to the destination zone according its routing table.

If history path is not available, the source node performs the second step by checking its intrazone routing table to find out if the destination node is in the table. If yes, the source node extracts the destination zone ID and next node ID and attaches them to the data packet.

In MZHLS, intrazone routing table contains not only the routes to the same zone node, but also the routes to the neighbor nodes in different zone that have direct connection to local zone. Because nodes receive LKR from such neighbor nodes and include them in their NLSPs, the routes to these neighbor nodes are discovered by shortest path algorithm similarly as routes to other nodes in the same zone. By incorporating more nodes to intrazone routing table, the range of intrazone search is enlarged and routing is more adaptive to more dynamic networks. For example, in fig. 2.22 node A continuously sends data packets to node B, and node C is the last intermediate node to forwarding packets to node B. Node B travels from zone 1 to zone 2. When node B leaves zone 1 and enters zone 2, it is still in node B's intrazone routing table because node B is a neighbor node to zone 1. So node B will still get the packet and acknowledge node A with its new zone ID. Node A then modifies its history path and all following packets will be forwarded to zone 2. Thus an extra location search is avoided by the extended intrazone routing algorithm.

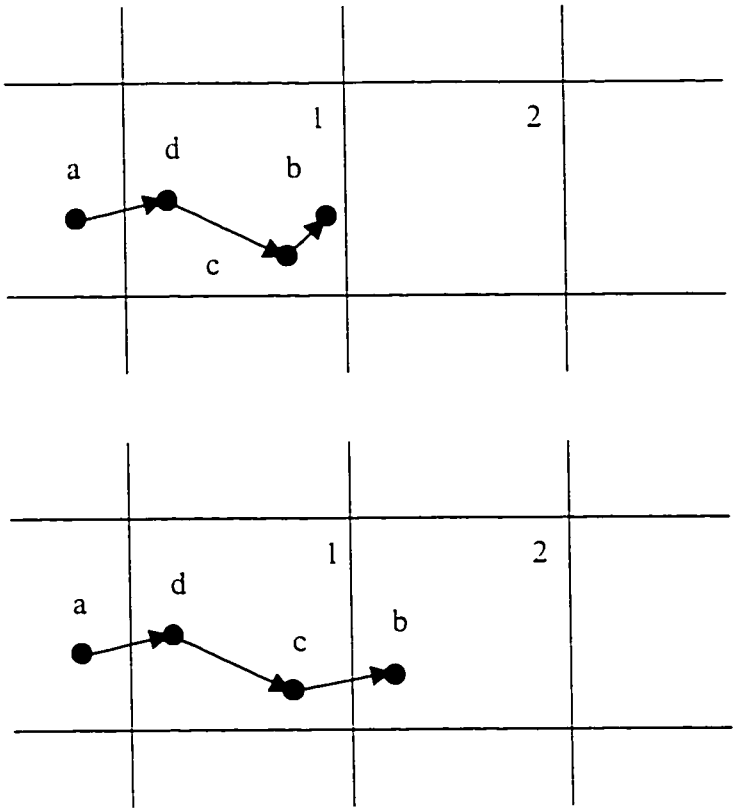


Figure 2.22: Extended intrazone routing

If the source node can not find destination node in its intrazone routing table, it performs the third step by sending a location request to each periphery zone of the network. Because some zones are located in the center of the network, location request packets destined for the periphery zones may pass through these center zones. In this case, the gateway node of that zone issues a location request response to the source node and the location search procedure is completed. Instead of sending a specific query to every zone as in ZHLS, only peripheral zones are queried in MZHLS. As mentioned above, a destination node which is a neighbor to any searched zone will also be found in the second routing step.

If the destination is not found in step 3, source node will repeat step 3 until the route to the destination node is finally found.

Chapter 3

Simulation Model

To evaluate the performance of MZHLS, a simulation model is established in this chapter. The simulation model is implemented in C. A description to the simulation model is given in the first section. Some input parameters are then defined for network configuration. Finally, some performance criteria are introduced.

3.1 Introduction to MZHLS Simulation Model

The network spans an area of 6km X 6km and 200 nodes can randomly move inside this area at speeds from 0km/h to 90km/h.

The transmission rate of a node is assumed to be 1Mbps. A packet size of length of 6000 channel symbols is selected. A node at speed 10km/h can move 0.016m during the transmission of a packet. Thus the unit distance is defined as 0.1m in simulation model and each program iteration time corresponds to 0.006second of real time. Every 100 iterations, each node randomly selects a new moving direction.

The network is further partitioned into 36 zones, each with 10000 X 10000 square unit distance. Fig. 3.1 shows the zone map of simulation model. The transmission radius is assumed to be 8000 unit distance (800m). Because the channel access depends on the result of MAC layer contention, the transmission radius is used to elect the successful node winning the contention. To simplify our model, any two simultaneously transmitting nodes are assumed apart from each other at least twice transmission radii in horizontal and vertical directions.

Simulation is based on the assumption of data transfer process mostly non real time, where delivery of packets is guaranteed. A node transmits more packets only if all previous transmitted packets have been received correctly and acknowledged by destination node.

31	32	33	34	35	36
25	26	27	28	29	30
19	20	21	22	23	24
13	14	15	16	17	18
7	8	9	10	11	12
1	2	3	4	5	6

Figure 3.1 Zone map of the simulation model

3.1.1 Node Status Information

Each node maintains a database that records general node information. Fig. 3.1 shows the node information database. The node is associated with a node ID numbered from 1 to 200. A temporary zone ID is assigned to the node based on the zone partition and position of the node. Position X and Y field denotes the current position of the node. Moving direction flag takes 5 value to denote the current move direction of the node, which are up, down, left, right, fixed.

Each node maintains some flags that are associated with the transmission status of that node. Ready to send flag is set if the node is waiting for a chance to send its packet; otherwise, the node is idle. Link state flag is set when the link topology has changed and the node has generated new NLSP or ZLSP to broadcast. Receiving sequence flag and sending sequence flag indicate the current sequence number of data packet received or sent. History path records the last path that was found previously to a specific destination node, specifically, the destination zone ID attached in the last delivered data packet.

node

Node ID		1~200
Zone ID		1~36
X position		0~60000
Y position		0~60000
Moving Direction		0,1,2,3,4
Moving Speed		0,1,...9
Flags	Ready to send	0: silent, 1: packet ready
	Link state change	0: no change, 1: change
	Sending sequence	0~255
	Receiving sequence	0~255
	History path (zone ID)	1~200

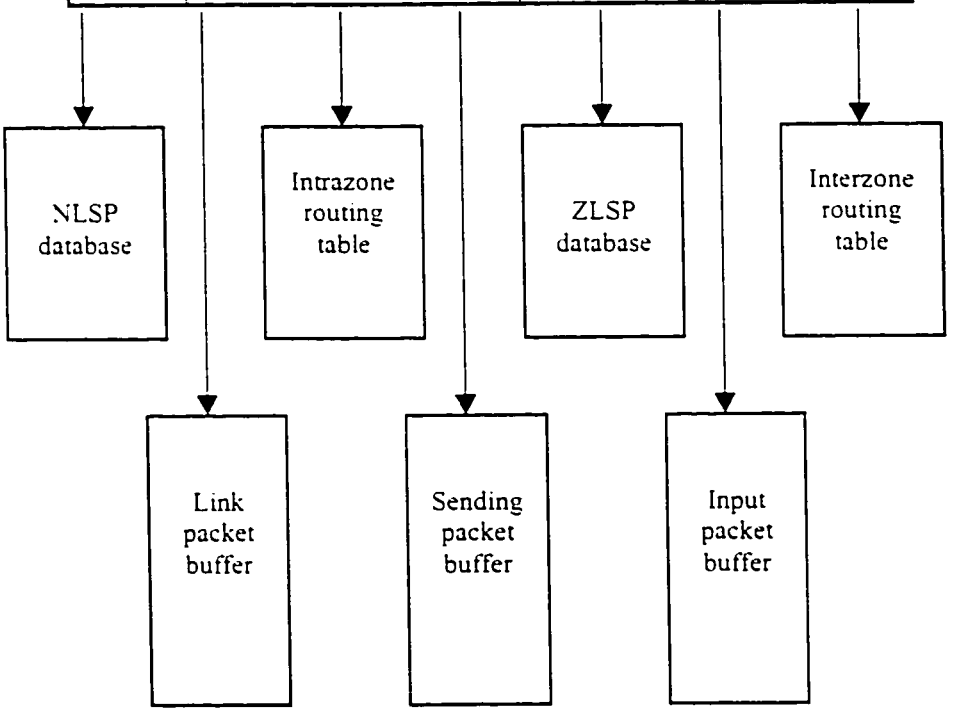


Figure 3.2: General information database handled by nodes

Note:

Moving Direction: 0, stay; 1, up; 2,left; 3, down; 4, right.

Moving Speed: 0, stay, 1: 10Km/hour=1grid/Iteration; 2: 20Km/hour... maximum 9

Different from other flags in database, these two flags are used for simulation only, not in real implementation of the proposed routing techniques.

3.1.2 Routing Information Handled by Nodes

Each node maintains four routing information databases, which are NLSP table, intrazone routing table, ZLSP table, interzone routing table. They are shown in table 3.1 to table 3.4.

	Node ID	Connect to local nodes:			Connect to nodes of different zone					Age timer
Maximum 20 nodes	a	Na1	Na2	...	Za1	N11	Za2	N12	...	AT-a
	b	Nb1	Nb2	...	Zb1	N22	Zb2	N22	...	AT-b
	...	...			...					...
	...	...			...					...

Table 3.1: NLSP data base of simulation model

Maximum 19 nodes	Local node	Destination Node 1	Next Node a	
		Destination Node 2	Next Node b	
		...	...	
	Other zone node	Destination Node 3	Next Node c	Next zone Z1
		Destination Node 4	Next Node d	Next zone Z2
		...	...	...

Table 3.2: Intrazone routing table of simulation model

	Zone ID	Connect to zones (Maximum 8 zones)					Age timer
Maximum 36 zones	1	Z11	Z12	Z13	...	...	AT1
	2	Z21	Z22	...			AT2
	...	...					...
	...	...					...

Table 3.3: ZLSP data base of simulation model

Maximum 35 zones	Destination zone 1	Next zone Z1	Next node N1
	Destination zone 2	Next zone Z2	Next node N2
	...	...	...
	...	...	...

Table 3.4: Interzone routing table of simulation model

NLSP table and ZLSP table contain NLSP or ZLSP packets the node has collected. Each NLSP or ZLSP has an age timer. If NLSP timer expires, the NLSP will be deleted form the table while ZLSP will not be deleted until a new ZLSP of that zone replaces the old one. This is based on the fact that zone link topology changes much slower than node link topology. Age timer is also used to eliminate the duplicate NLSP or ZLSP packets. When a node receives a NLSP or ZLSP, it compares the age timer with the one from same source node or source zone in its table, the later generated one will be included into the table while the older one will be dropped. If the two timers have the same value, it indicates that the two packets are duplicates. The received packet will not be repeated and will be simply dropped. It ensures that no loop occurs during flooding of NLSP or ZLSP.

3.1.3 Packet Type

In this simulation model, there are 6 types of control packet besides data packet. These control packets are: link request packet (LKR), link response packet (LKRS), node link state packet (NLSP), zone link state packet (ZLSP), location request packet (LR), and location request response packet (LRS). The packet formats are shown in table 3.5. To simplify the simulation, we assume that all acknowledgment are received correctly and in time without consuming any channel capacity. So there is no actual acknowledgement packet being transmitted. The source node is acknowledged immediately upon successful delivery.

Data packets have time counter field and hop counter field. If the time counter or hop counter exceeds certain limitation, the data packet will be dropped from the network. Sequence number field of a data packet is used to verify whether the desired data packet is received correctly.

	DU type	Source node	Source zone	Desti-nation node	Desti-nation zone	Next node	Hop count	Sequence number	Time counter	List of link
LKR	1	—	—						—	
LKRS	2	—	—	—	—				—	
NLSP	3	—	—						—	—
ZLSP	4	—	—						—	—
LR	5	—	—	—	—				—	
LRS	6	—	—	—	—					
DU	7	—	—	—	—	—	—	—	—	

Table 3.5 packet format of simulation model

Note: The field is marked by "

LKR: link request packet

LKRS: link request response packet

LR: location request packet

LRS: location request response packet

NLSP: node link state packet

ZLSP: zone link state packet

DU: data packet intrazone

Each node maintains three packet buffers of the same size 20 packets: link buffer, sending buffer, input buffer. The link buffer is used to put in new generated control packets that include LKR, LKRS, NLSP and ZLSP packets. Because these packets are responsible for delivering information of network link topology, they are given higher priority over other packets. Packets in other buffers are transmitted only if the link buffer is empty. The sending buffer is used to insert data, LR and LRS packets. The data packets may be either generated by the node itself or packets transmitted by other nodes. The input buffer is a storage of data packets accepted from upper layer (user). The reason for

setting up the input buffer is that a packet is eliminated from the source node's buffer only after the source node gets an ACK for this packet. By setting up a separate buffer, the packet waiting for an ACK will not block the path through which other packets pass.

3.2 Input Parameters

In this section, we introduce some input parameters that have important effect on network performance. The simulation results of different network performance are based on variation of these input parameters.

- DU timeout: the maximum time during which a data packet can exist in the network. If the time counter of a data packet exceeds the maximum value, the packet will be dropped from the network.
- Input traffic rate: average number of input data packets to the whole network for every simulation iteration. Variation of input traffic rate shows network performance under different traffic load.
- NLSP flooding period: the time interval between two births of NLSP packets generated by the same node. It reflects the frequency of updating node level link information. Each node of the network has the same NLSP flooding period.
- ZLSP flooding period: the time interval between two births of ZLSP packets generated by the same gateway node of the same zone. Each zone has same ZLSP flooding period but different start time.
- Hop limit: maximum hops a data packet is relayed from node to node before which the packet is dropped.
- Link time: each node periodically broadcasts a LKR packet and wait for a certain period to collect the LKRS packets from its neighbor nodes and then broadcast its new generated NLSP. This is defined as link procedure, and the period from the time the source node generates a LKR to the time it stops collecting LKRS is defined as link time.

- Mobility: maximum speed a node can move at.
- Search algorithm: specifies which zones will be searched and the search order.

3.3 Output Parameters for Performance Evaluation.

Some output parameters are defined to evaluate the network performance in this section.

- Overhead (packets/packets): defined as total number of control packets (query, response and hello message) generated from all iterations, divided by total number of packets generated by all users from all iterations including control packets generated.
- Overhead (transmissions/transmissions) defined as total number of transmissions for all control packets divided by the sum of total number of transmissions for all control packets and successful data packets. Total number of transmissions for all successful data packets is calculated by multiplying total number of successful data packets by average number of hops of a typical data packet.
- Network efficiency: defined as total number of successful delivered data packets divided by total number of data packets generated for all users and all iterations.
- Average transfer delay: defined as average of total transfer delay of data packets from generation to final delivery.
- Standard deviation of transfer delay: defined as standard deviation of total transfer delay of data packets from generation to final delivery.
- Average location search delay: defined as average time used by a location search procedure which is initialized by the generation of the first LR packet and ended by a positive LRS to the source node.
- Standard deviation of location search delay: defined as standard deviation of time used by a location search procedure.
- Average path length: defined as average number of hops of all successfully delivered data packets.

- Standard deviation of path length: defined as standard deviation of number of hops of all successfully delivered data packets.
- Average number of packets in link, (in sending and input buffers): average number of packets over all users and all iterations in different buffers.
- Standard deviation of number of packets in link, (in sending and input buffers): standard deviation of the number of packets over all users and all iterations in different buffers.
- Average probability buffer overflow: average probability of user buffer overflow over all users and all iterations.
- Standard deviation of buffer overflow probability: standard deviation of probability of different buffer overflow over all users and all iterations.

Chapter 4

Simulation Results and Evaluations

In this chapter, numerical results of the simulation are presented. Detailed performance evaluation are carried out based on the variation of each set of input parameters. Both advantage and disadvantage of the MZHLS are discussed. Advantage of MZHLS over ZHLS and other routing techniques is discussed through comparison of the results. Weakness of the MZHLS is also pointed out and efforts to alleviate it to some extent will be presented.

4.1 Network Performance under Different Data Packet Timeout

Different settings of data packet timeout have important effect on network performance. If the data packet timeout is short, a data packet exists shortly in the network. Data packets are thus more vulnerable to network congestion. Due to more undesired dropped packets, number of successful delivered data packets decreases. Fig 4.1 clearly shows that the network efficiency is low when timeout is short, and network overhead (transmissions/transmissions) is very high.

As data packet timeout increases, data packets can exist longer in the network, they are less vulnerable to network congestion. But longer timeout means that a node needs to wait for a longer time if for some reason the data packet will never reach its destination, for example, packet loss due to node mobility. In case of location search, the source node needs to wait a longer time for each specific zone if the destination node doesn't reside in that zone. Due to the above reasons, total number of successful data packets decreases.

Fig 4.1 shows that as timeout increases, network efficiency decreases while network overhead (transmissions/transmissions) increases.

It is interesting to observe that overhead ratio (packets/packets) doesn't change much while timeout changes. Because timeout field doesn't affect the number of data packets nodes generated, overhead (packets/packets) will not change significantly if the number of control packets generated doesn't change.

Fig. 4.2 shows that both average and standard deviation of data packets transfer delay increase if the data packet timeout takes a very large or very small value. Minimum transfer delay is achieved only within certain range.

In fig. 4.3, both average and standard deviation of location search delay continuously increase as packet timeout exceeds 500 iterations. Due to the time spent on each zone search increases, location search delay increases approximately linear proportional to packet timeout. Note that when packet timeout is less than 500 iterations, location search delay never decreases. This is because more LR packets are timeout and are dropped before they reach the destination zone.

In fig. 4.4, maximum average number of hops of data packets is achieved when timeout is larger than 500 and less than 2000. Because data packets destined to remote nodes are eliminated by timeout, average number of hops of data packets decreases as timeout decreases. But it is interesting that average number of hops doesn't increase as timeout has a very large value. As shown in fig.4.3, large timeout value leads to long location search delay. Due to the long location search delay and mobility of nodes, less data packets of more hops are successfully delivered. This explains why average number of hops of data packets doesn't increase as timeout has a very large value.

Figures 4.5 and 4.6 show numbers of different packets in different buffer. Because number of control packets is not affected by different data packet timeout, average and standard deviation of the number of packets in link buffer vary slightly.

Comparing figures 4.5 and 4.6, the number of data packets in input buffer has a minimal value while number of packets in sending buffer continuously goes down as timeout increases. Because number of packet in input buffer depends on the network efficiency and input traffic rate. when input traffic rate is fixed, the number of packet in input buffer changes negatively to network efficiency. This is demonstrated by the curve of network efficiency in fig. 4.1. On the other hand, the number of packets in sending buffer indicates how many data (including LR, LRS) packets exits in the network. Because a source node can not send more packets before it gets ACK for the last data packet, total number of data packets existing in the whole network can not exceed the number of all nodes. Longer timeout causes longer waiting time even if the data packet has been dropped from the network. So as timeout increases, number of packets in sending buffer monotonously decreases. (If other flow control protocols such as sliding window protocol are implemented, number of total data packets may exceeds the total number of nodes; channel capacity may be better used and network efficiency may be higher).

Number of packets in a buffer always indicates the overflow probability of the buffer if the buffer size is fixed. Corresponding to fig. 4.5 and fig. 4.6, overflow probability of different buffers vary as the average number of packets in the buffers vary. Fig. 4.7 and fig. 4.8 show these variations.

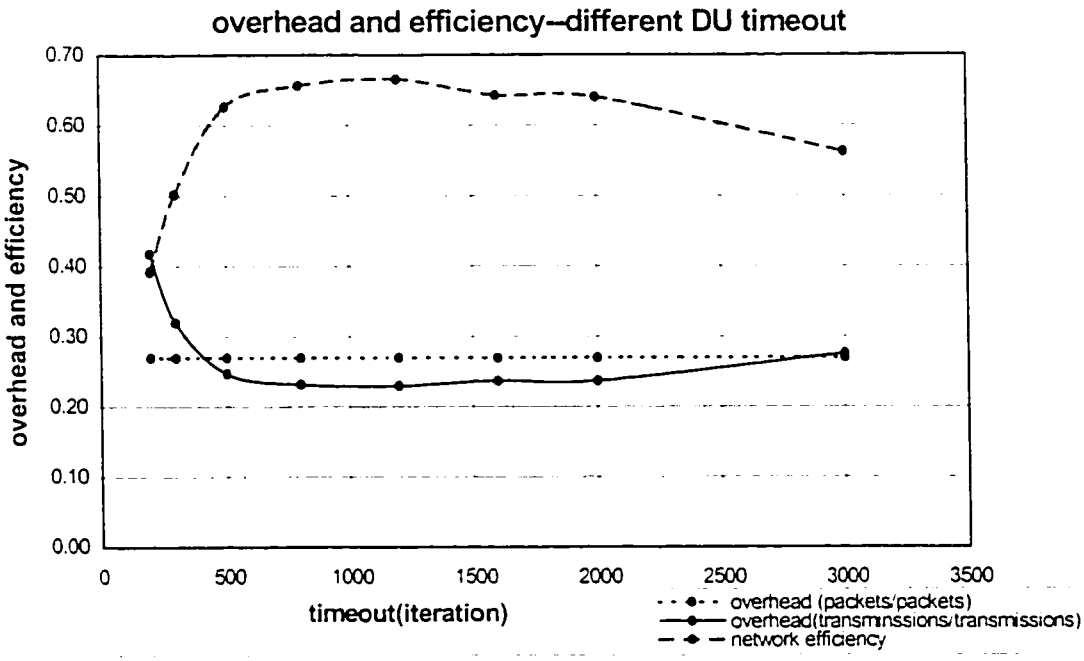


Figure 4.1: Overhead and efficiency-different DU timeout

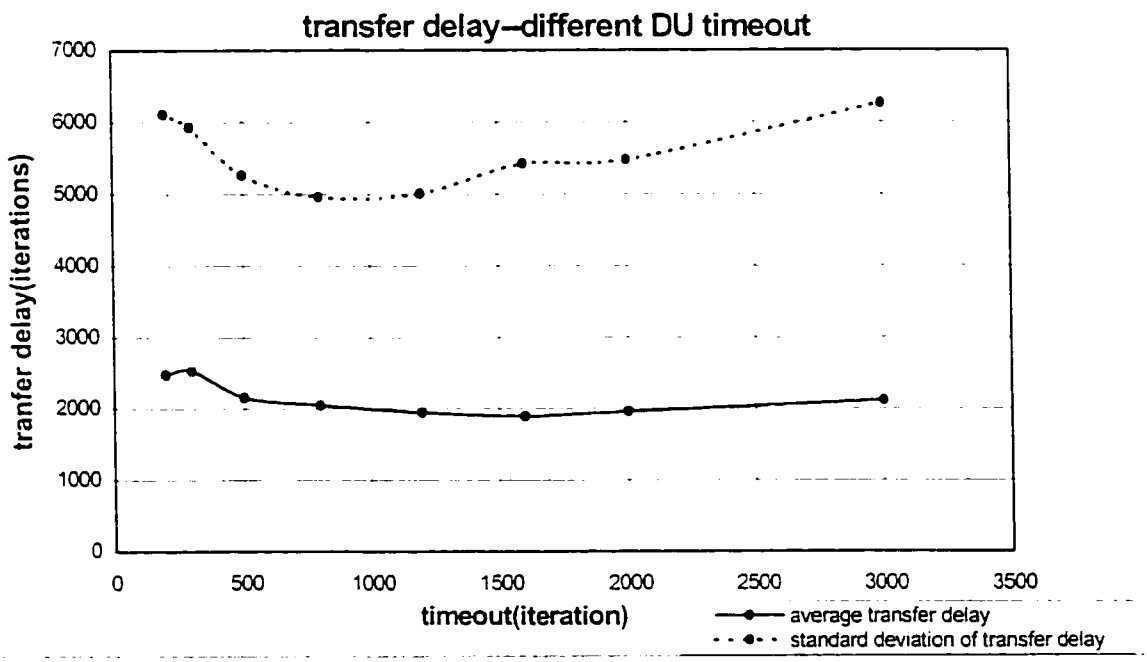


Figure 4.2: Transfer delay-different timeout

Test condition:

Simulation runs 50000 iterations. Maximum speed of nodes: 30km/h. Input traffic rate: 1 packet/iteration. Period flooding NLSP: 10000 iterations. Period flooding ZLSP: 25000 iterations. Hop limit: 20. Link time: 250 iteration.

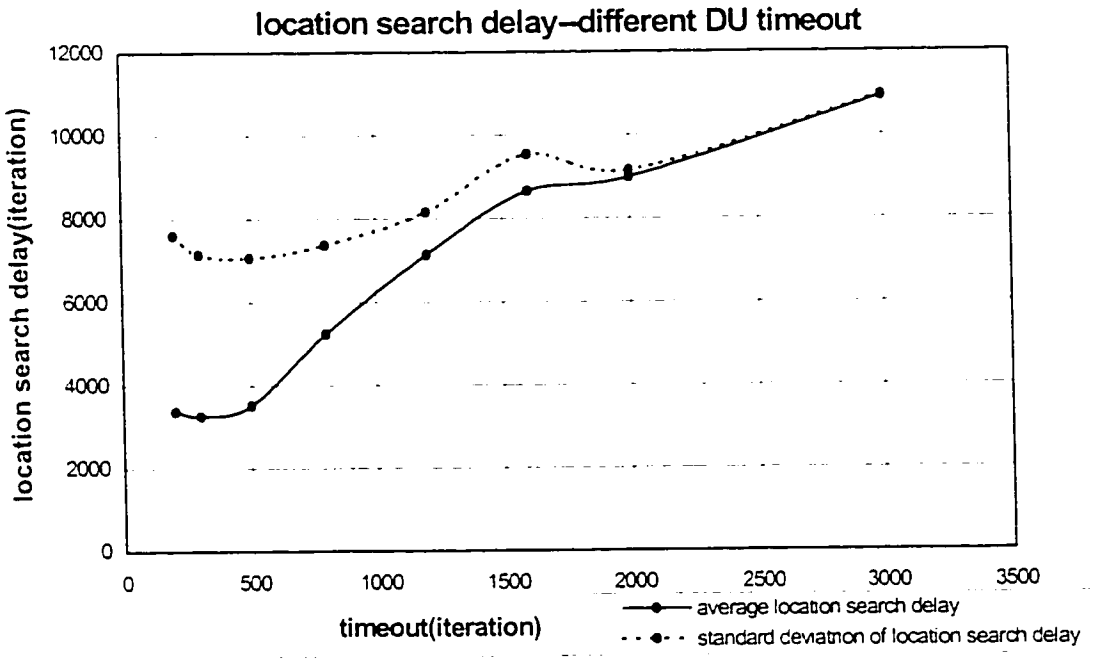


Figure 4.3: Location search delay-different DU timeout

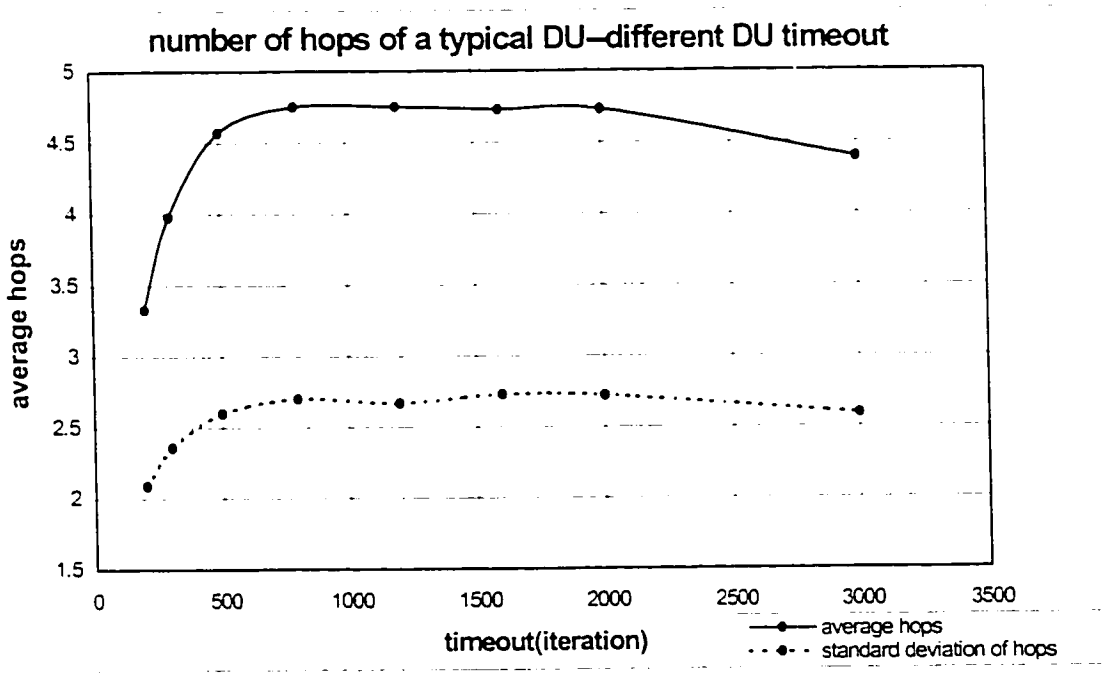


Figure 4.4: Number of hops of a typical DU-different DU timeout

Test condition:
Simulation runs 50000 iterations. Maximum speed of nodes: 30km/h. Input traffic rate: 1 packet/iteration. Period flooding NLSP: 10000 iterations. Period flooding ZLSP: 25000 iterations. Hop limit: 20. Link time: 250 iteration.

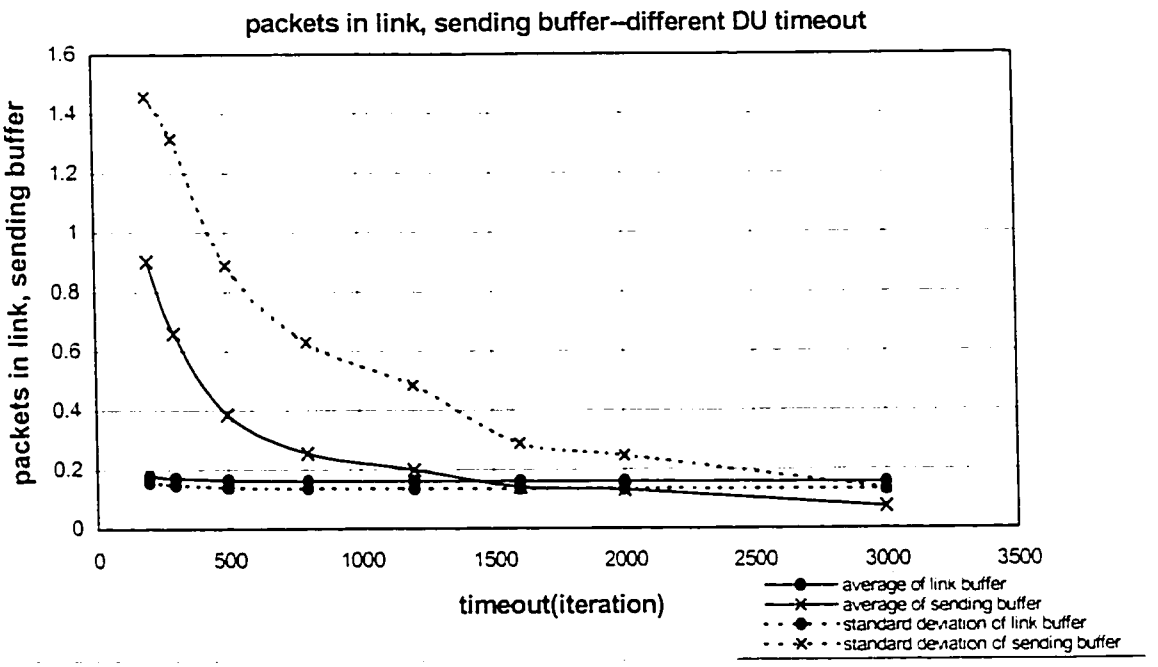


Figure 4.5: Packets in link, sending buffer-different DU timeout

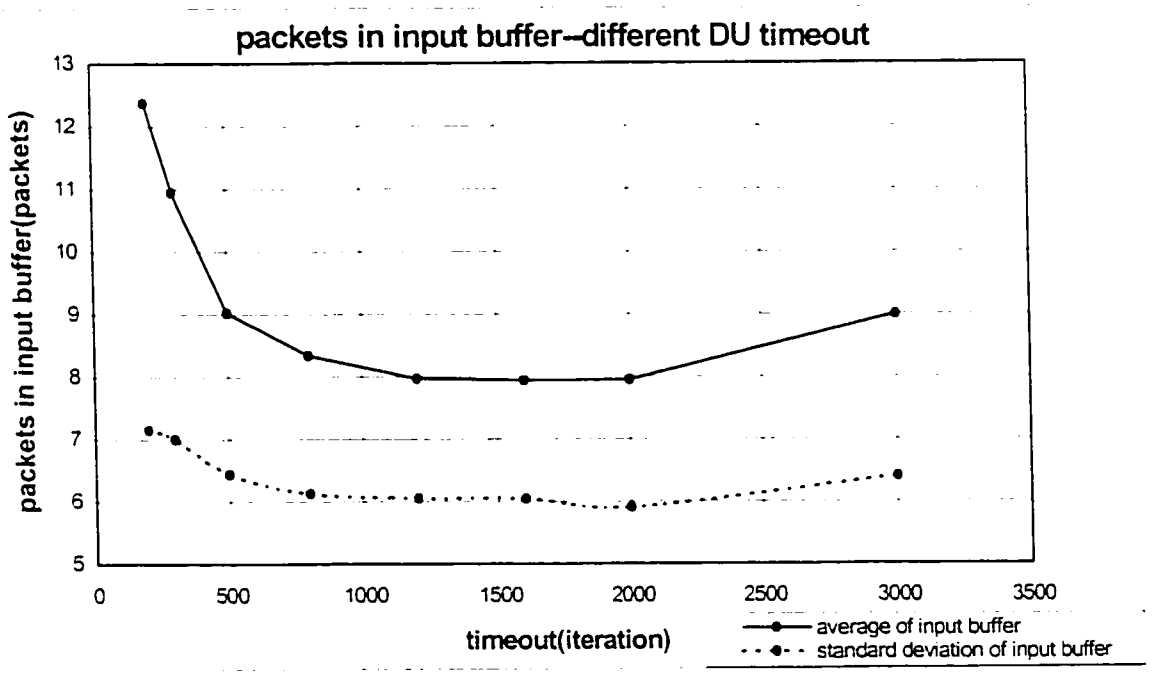


Figure 4.6: Packets in input buffer-different DU timeout

Test condition:
Simulation runs 50000 iterations. Maximum speed of nodes: 30km/h. Input traffic rate: 1 packet/iteration. Period flooding NLSP: 10000 iterations. Period flooding ZLSP: 25000 iterations. Hop limit: 20. Link time: 250 iteration.

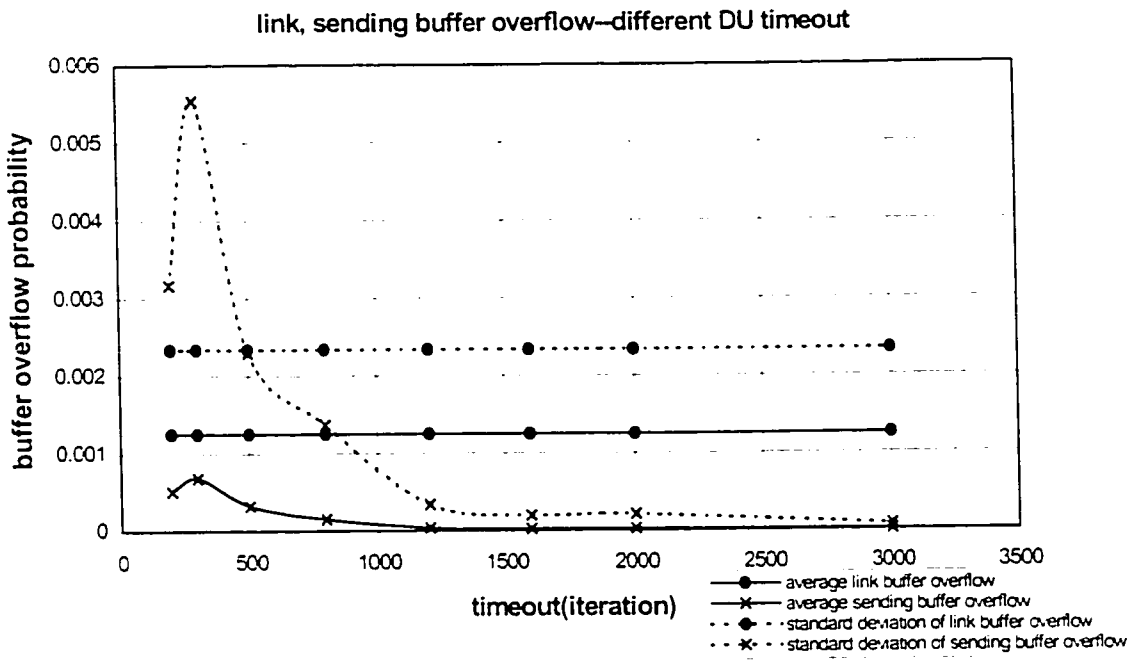


Figure 4.7: Link. sending buffer overflow-different DU timeout

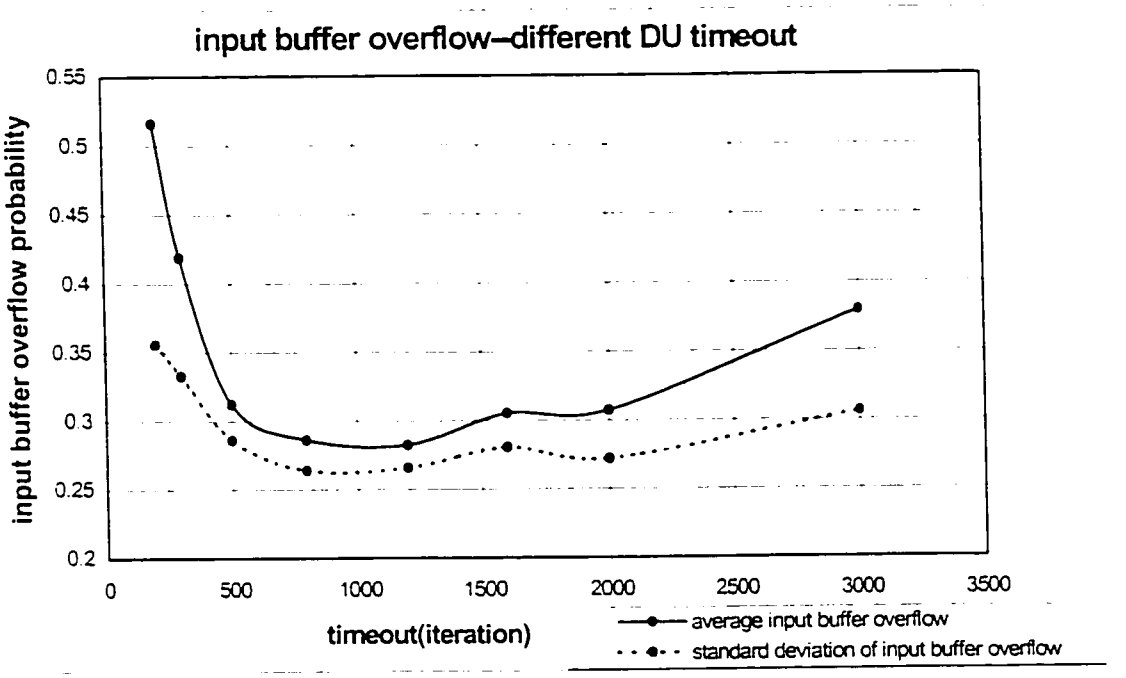


Figure 4.8: Input buffer overflow-different DU timeout

Test condition:
Simulation runs 50000 iterations. Maximum speed of nodes: 30km/h. Input traffic rate: 1 packet/iteration. Period flooding NLSP: 10000 iterations. Period flooding ZLSP: 25000 iterations. Hop limit: 20. Link time: 250 iteration.

4.2 Network Performance under Different Input Traffic Rate

Different input traffic rates cause the network to operate under different traffic load. The input traffic rate to the whole network equals the total number of nodes multiplied by average input traffic rate to each node. By setting different input traffic rates, a set of output parameters are obtained to reflect the network performance under different loads.

Fig. 4.9 shows the network efficiency and overhead as affected by different input traffic rate. Due to more packets transmitted and delivered, it is easy to understand that both network overhead (packets/packets) and overhead (transmissions/transmissions) decrease as input traffic rate increases. Note that network overhead (transmissions/transmissions) saturate when the input traffic rate is very high. This is because further decrement of overhead (transmissions/transmissions) is restricted by maximum network capacity(which means the maximum network capacity has been approached).

Network efficiency also decreases as the input traffic rate increases. It indicates that the number of successfully delivered data packets does not increase proportional to the increment of number of input packets. More input data packets are delayed or lost under heavy network traffic.

Fig. 4.10 shows that transfer delay decreases slightly as input traffic rate increases. This contradicts with our intuition. But if combined with fig. 4.16, it is easy to explain this result. We leave analyses of fig.4.10 and fig. 4.12 shortly after we explain fig 4.16.

Heavy network traffic load causes long location search delay as shown in fig. 4.11. Under heavy network traffic load, more packets may be lost due to timeout. Location search procedure may need to be repeated if LR packet is lost and thus location search delay is prolonged.

Fig. 4.13 and 4.14 show that numbers of packets in sending buffer and input buffer increase as input traffic increases except that the number of packets in link buffer is affected little by input traffic rate. When input traffic rate is very high, average number of packets in input buffer tends to be the buffer size (20) while standard deviation of number

of packets in input buffer decreases. In this case, more and more input buffer are blocked by the very high input traffic rate and thus more input data packets are rejected by nodes. This is also justified by fig 4.16 which indicates more than 70% of all nodes experience input buffers overflow when input traffic rate is 8 packets/iteration for all users. In fig. 4.15, sending buffer overflow probability varies similarly as number of packets in sending buffer varies in fig 4.13. Overflow probability of link buffer is affected little by increasing input traffic rate just as per fig 4.13.

Now it is easy to understand the effect of different input traffic rate on transfer delay and average number of hops of data packets. Because packets may experience a long waiting time at intermediate nodes under heavy network traffic load, packets destined for remote nodes need more hops before arrival and thus are more likely to be dropped due to timeout. So a successful delivered data packets with high hop count may need to be retransmitted twice or more. This causes long transfer delay for data packets to remote destination and nodes accept less data packets from upper layer due to the longer distance to destination.

On the other hand, data packets destined to nearby nodes are less affected by heavy network traffic load. More data packets with low hop count are successfully delivered due to higher input rate. The two factors act together and significantly increase the proportion of low hop count data packets among all successfully delivered packets. As we see in fig 4.12, average and standard deviation of hops of a typical data packet decrease at high input traffic rate. In fig 4-10, the average and standard deviation of transfer delay also decrease at high input traffic rate.

Based on fig.4.10 to fig. 4.16, we can make a conclusion: under low input traffic rate, data packets are likely delivered successfully despite the distance to destination; under high input traffic, priority are given to nodes that have packets to nearby destinations. Remote destination nodes are more difficult to reach.

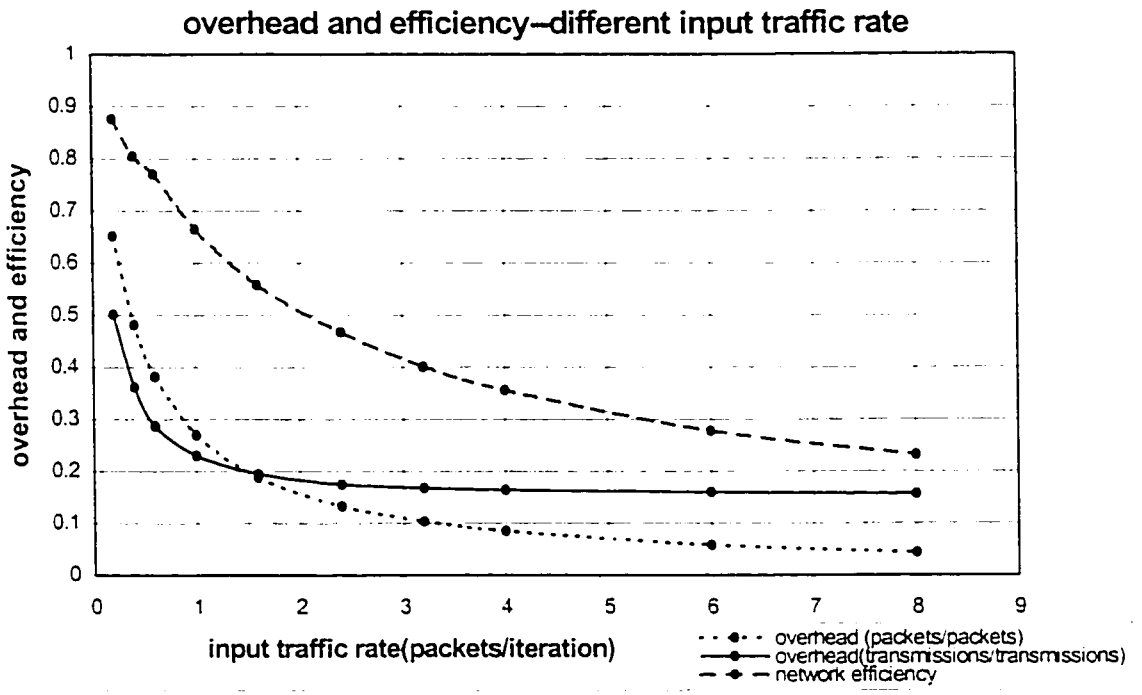


Figure 4.9: Overhead and efficiency-different input traffic rate

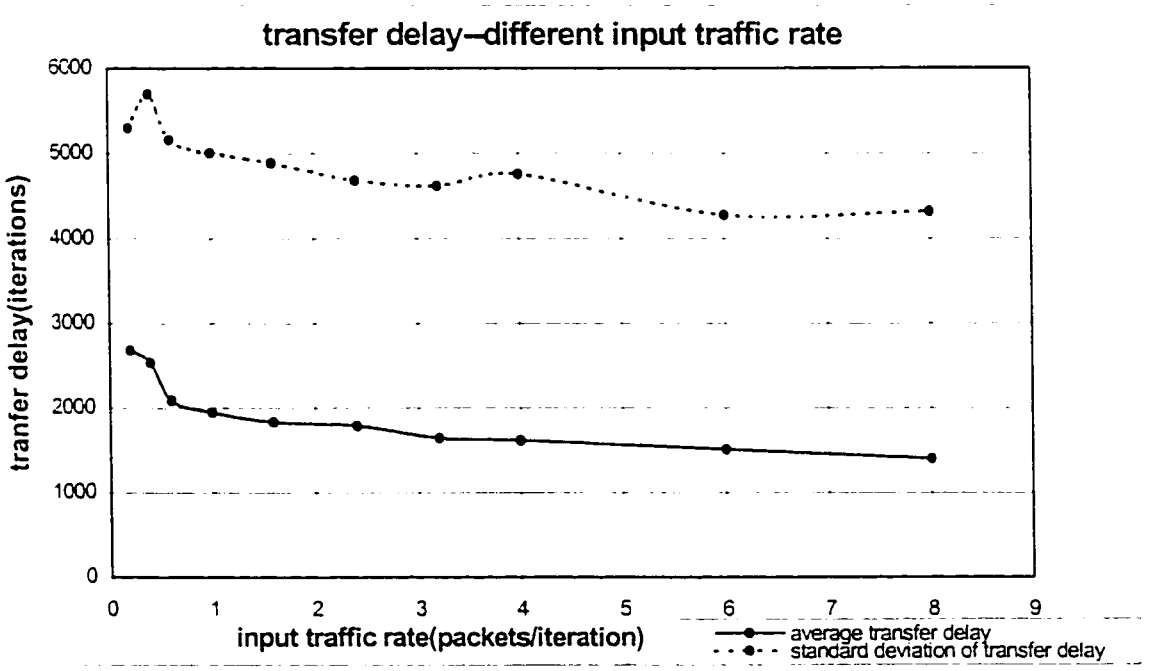


Figure 4.10: Transfer delay-different input traffic rate

Test condition:

Simulation runs 50000 iterations. Maximum speed of nodes: 30km/h. Period flooding NLSP: 10000 iterations. Period flooding ZLSP: 25000 iterations. Data packet timeout: 1200 iterations. Hop limit: 20. Link time: 250 iteration.

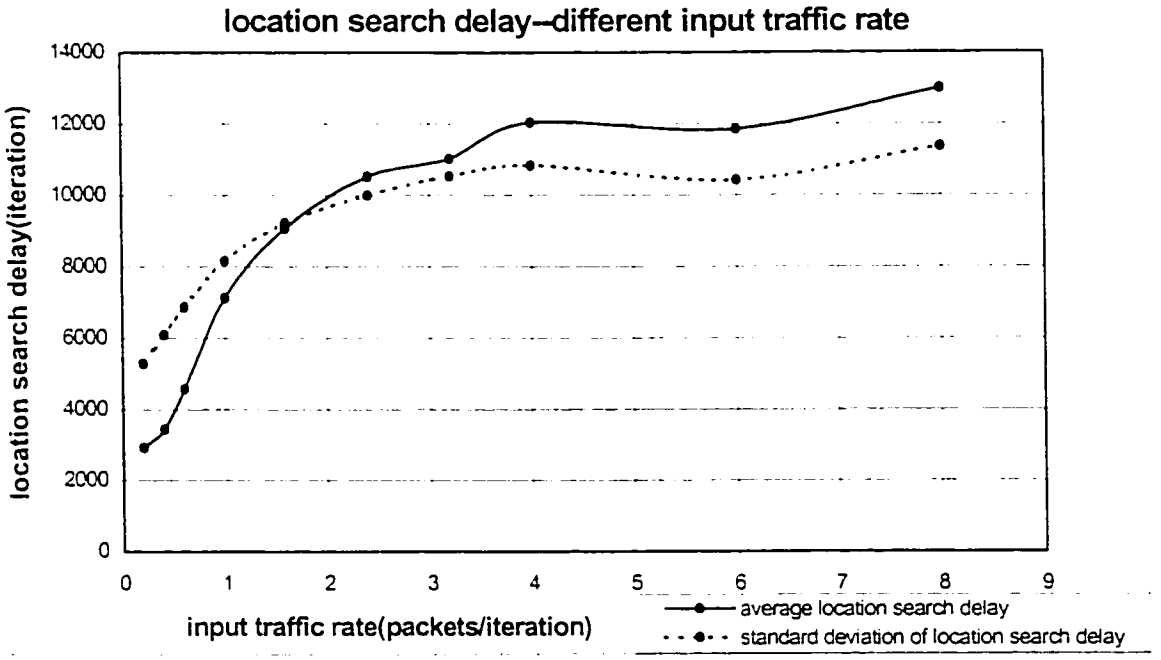


Figure 4.11: Location search delay-different input traffic rate

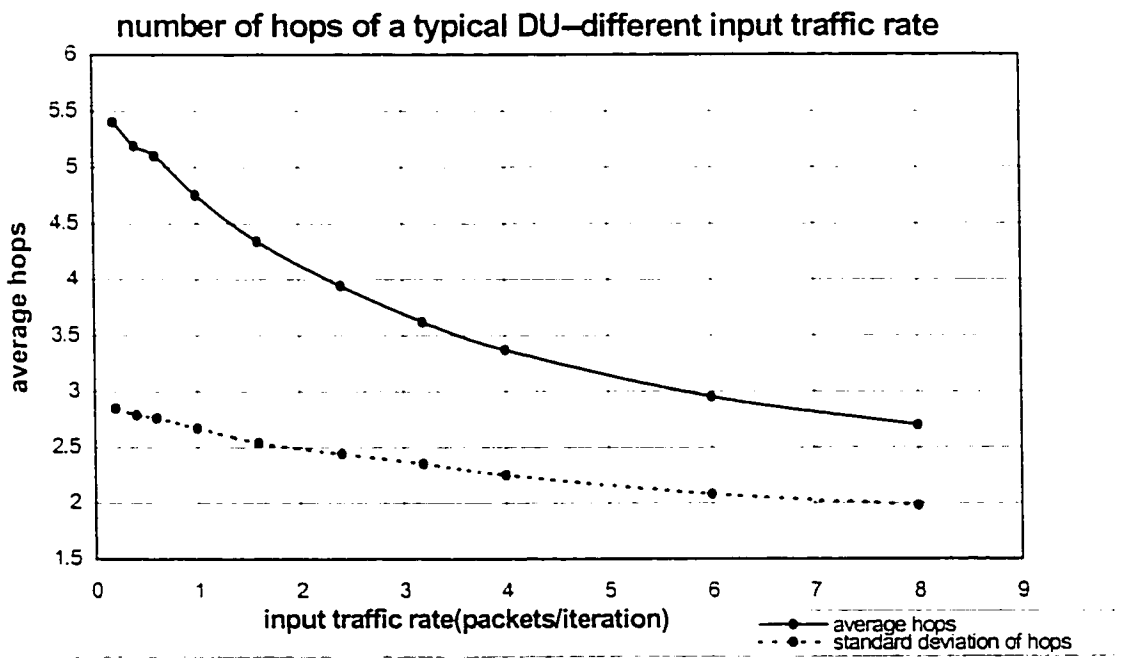


Figure 4.12: number of hops of a typical DU-different input traffic rate

Test condition:
Simulation runs 50000 iterations. Maximum speed of nodes: 30km/h. Period flooding NLSP: 10000 iterations. Period flooding ZLSP: 25000 iterations. Data packet timeout: 1200 iterations. Hop limit: 20. Link time: 250 iteration.

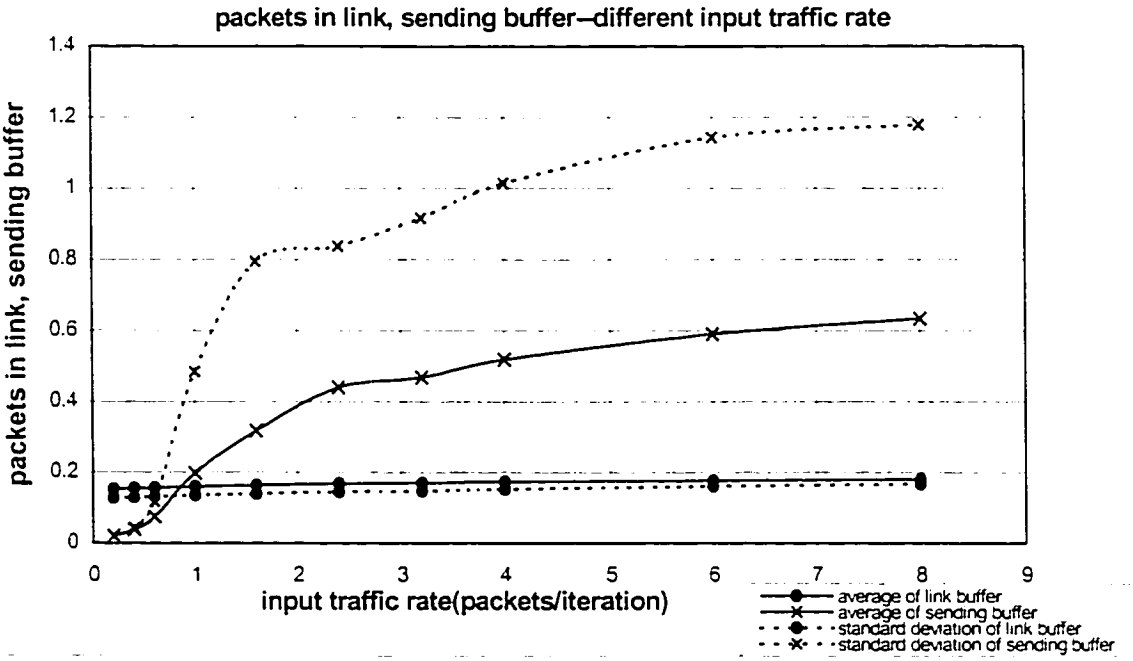


Figure 4.13: Packets in link, sending buffer-different input traffic rate

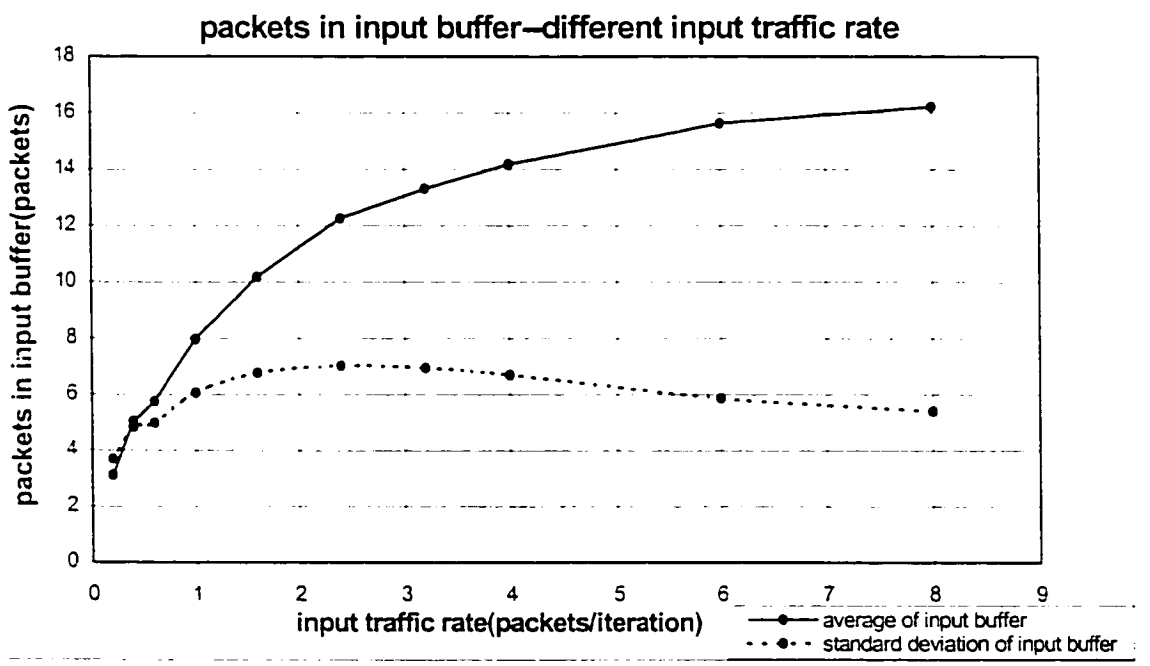


Figure 4.14: Packets in input buffer-different input traffic rate

Test condition:
Simulation runs 50000 iterations. Maximum speed of nodes: 30km/h. Period flooding NLSP: 10000 iterations. Period flooding ZLSP: 25000 iterations. Data packet timeout: 1200 iterations. Hop limit: 20. Link time: 250 iteration.

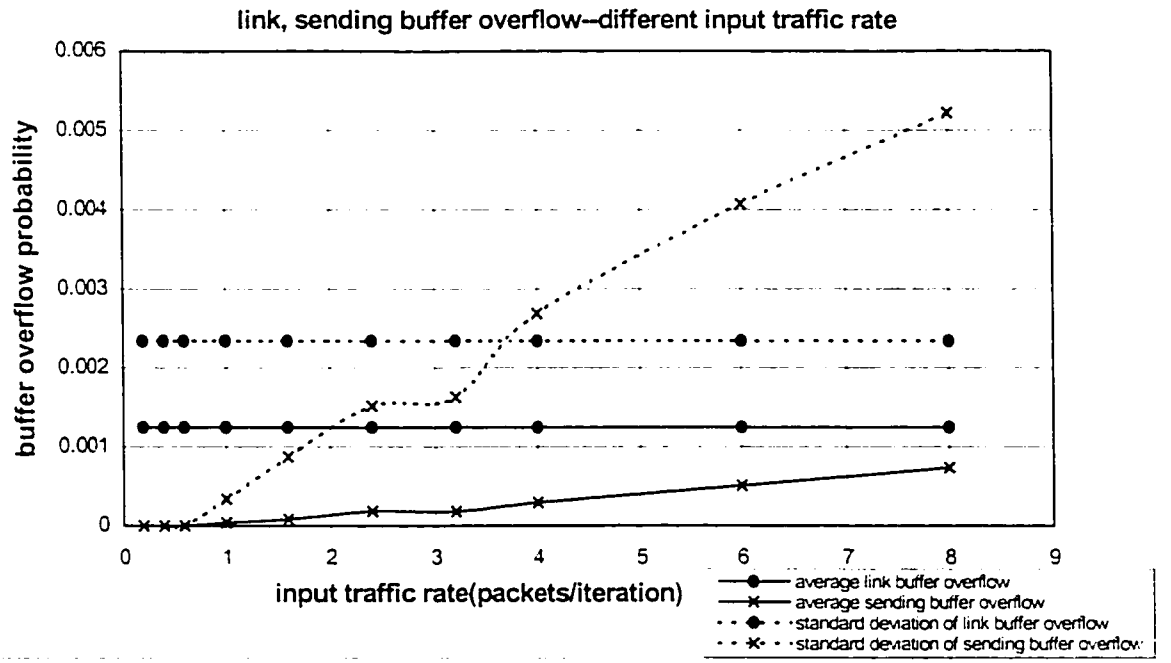


Figure 4.15: Link, sending buffer overflow-different input traffic rate

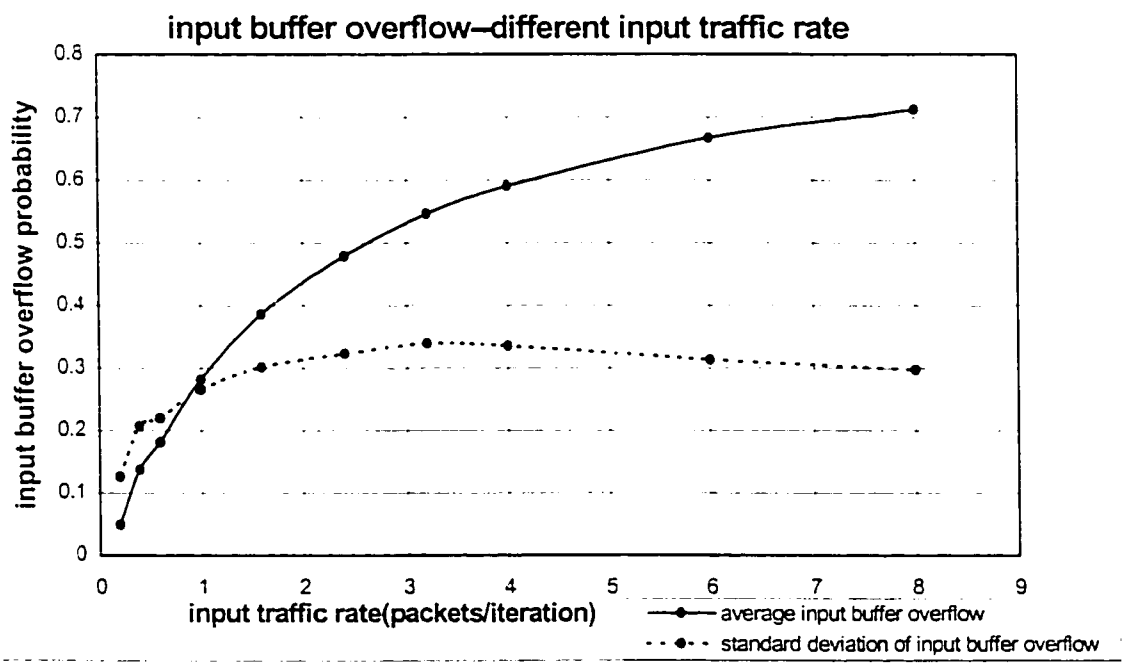


Figure 4.16: Input buffer overflow-different input traffic rate

Test condition:
Simulation runs 50000 iterations. Maximum speed of nodes: 30km/h. Period flooding NLSP: 10000 iterations. Period flooding ZLSP: 25000 iterations. Data packet timecut: 1200 iterations. Hop limit: 20. Link time: 250 iteration.

4.3 Network Performance under Different NLSP Flooding Period

In MZHLS, nodes periodically perform the link procedure which includes transmitting a LKR packet, collecting LKRS packets, flooding NLSP packets within the local zone. NLSP flooding period is defined as the duration between the two link procedure performed by the same node. Because all these three types of packets generated in a link procedure are overhead packets, different NLSP flooding period cause network perform differently.

Consider a network with N nodes partitioned into M zones, R is average number of nodes that in a reachable area of a typical node and A is the reachable area. In our model, N = 200, M = 200/36=5.5 nodes, R= N x A / total network area =200x (2 x800x 2x800) / (6000 x 6000)= 14.2 nodes. The number of overhead transmissions caused by single link procedure for all nodes is estimated as the total number of LKR, LKRS, NLSP packets, which is N(1+R+N/M). Compared with the third term, overhead caused by the LKR and LKRS packets are usually small if the zone size is much larger than the area covered by a transmission. Mario Joa-NG gave only estimation of NLSP overhead in ZHLS[12]. Note that the estimation above is based on that numbers of nodes in each zone are equal. We have seen that overhead caused by NLSP is approximately square proportion to number of nodes in a zone. If number of nodes in different zones deviates greatly from the average number N/M, total overhead to network caused by NLSP flooding will increase.

Because ZLSP packets are repeated by each node in the network, overhead caused by flooding ZLSP packets is estimated as NM.

Different NLSP flooding period has important effects on network performance. NLSP delivers node level topology information as a network vehicle but put extra overheads to network capacity. Short NLSP period means frequent update of node level link topology information but heavy network overhead, and vice versa. So there is a trade off between

network overhead and reliable links. To achieve good network performance, period of flood NLSP should be carefully defined. Fig. 4.17 to 4.24 show these characteristics.

In fig. 4.17, as NLSP flooding period increases, overhead continuously decreases while network efficiency has a maximum value and decreases at both end. It clearly indicates that number of successfully delivered data packets decreases due to heavy network traffic or unreliable path.

To demonstrate our estimation of overhead in transmissions, we arbitrarily select in fig. 4.17 period flooding NLSP = 10000 as an reference point to calculate the overhead (transmissions/transmissions). For each iteration, NLSP overhead is N x (1+R+N/M) /10000 =200x(1+14.2+5.5)/10000=0.414(transmissions/iteration),

period flooding ZLSP is 25000 and ZLSP overhead is NM/25000= 200x36/25000= 0.288(transmissions/iteration).

Total overhead = NLSP overhead + ZLSP overhead=0.414 + 0.288 = 0.7(transmissions/iteration)

Input traffic rate is 1 packet/iteration, network efficiency is 0.65 from fig. 4.17, so successful data packets delivered rate is 1x 0.65= 0.65 (packet/iterations). From fig 4-18, average number of hops is 4.75 in fig.20, so transmissions for successful data packets per iteration is 0.65 x 4.75= 3.09 (transmissions/iteration).

Now we can calculate the overhead(transmissions/transmissions) = overhead transmissions/(successful transmissions for data +overhead transmissions) = 0.7 / (3.09 +0.7) = 18.5 %, the simulation result in fig. 4-17 is 22%, 2.5 percentage higher than calculated result. As we stated above, the 2.5 percentage increment of overhead is caused by unequally distributed number of nodes in different zones. So the calculation of overhead is a very exact estimation of the simulation result.

Two factors, path reliability and overhead traffic, functions together to make location search delay increase when NLSP flooding period is very short or very long. Minimal

location search time is found when NLSP flooding period takes a proper value, such as 5000-10000 iterations in fig. 4.19.

As shown in fig. 4.18, average transfer delay decrease to approximate 2000 iterations when NLSP flooding period is longer than 5000 iterations. Because transfer delay consists of location search time (if necessary) and delivery time, effect of NLSP flooding period on transfer delay is complicate. Transfer delay differs greatly at individual nodes especially when NLSP flooding period is long.

If period flooding NLSP is long enough (> 4000 iterations), overhead caused by NLSP packets is reduced and number of hops of a typical data packet is affected little as shown in fig. 4.20.

Fig. 4.21 shows that both number of packets in link buffer and in sending buffer decrease as NLSP flooding period increases. Long NLSP flooding period reduces the number of packets in link buffer directly and reduces number of packets in sending buffer by releasing channel capacity from overhead. On the other hand, number of packets in input buffer has a minimum value as NLSP flooding period varies. This is correspondent to the effect on network efficiency. Maximum network efficiency and thus minimum queuing packets in input buffer are achieved when period flooding NLSP is set between 4000 and 10000 iterations.

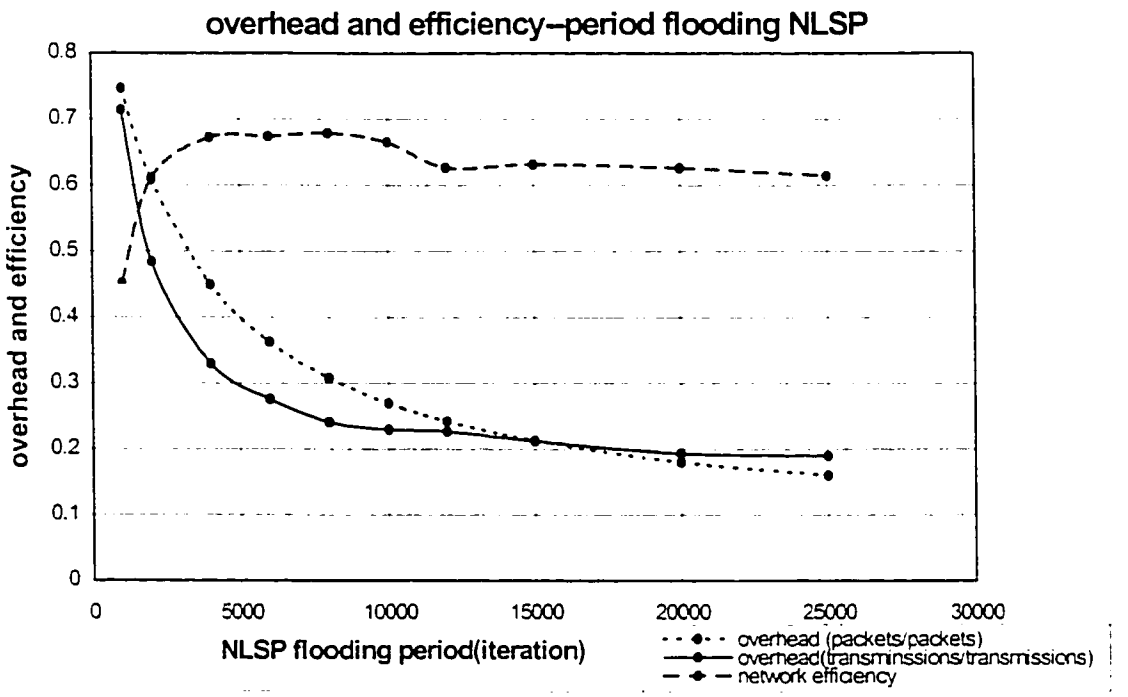


Figure 4.17: Overhead and efficiency-period flooding NLSP

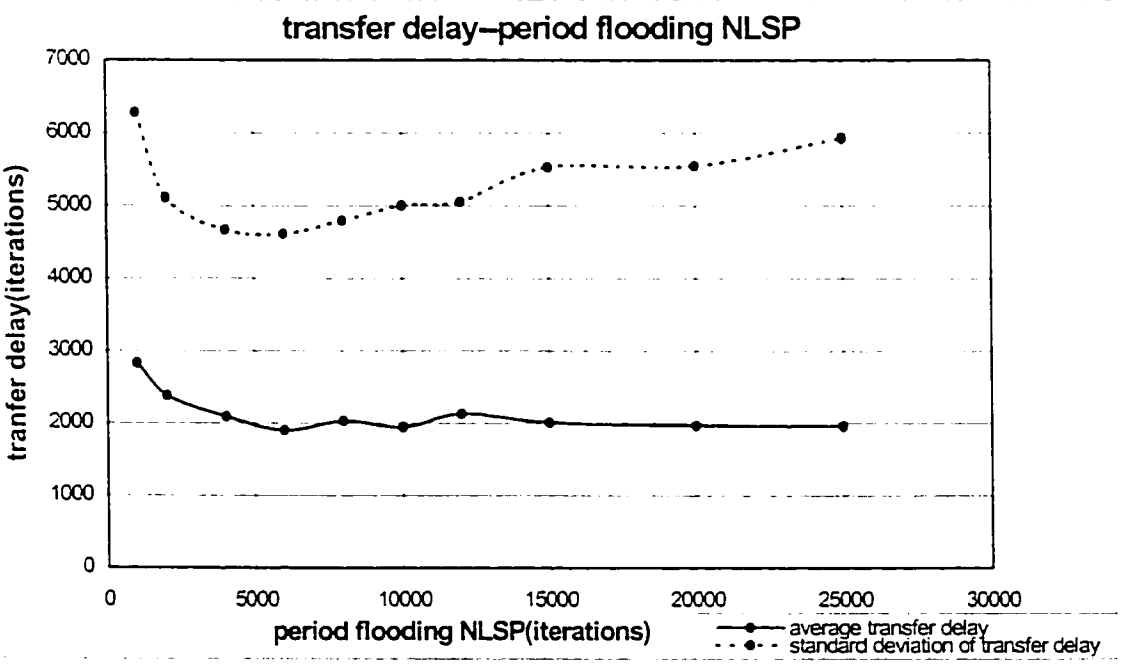


Figure 4.18: Transfer delay-period flooding NLSP

Test condition:

Simulation runs 50000 iterations. Maximum speed of nodes: 30km/h. Input traffic rate: 1 packet/iteration. Period flooding ZLSP: 25000 iterations. Data packet timeout: 1200 iterations. Hop limit: 20. Link time: 250 iteration.

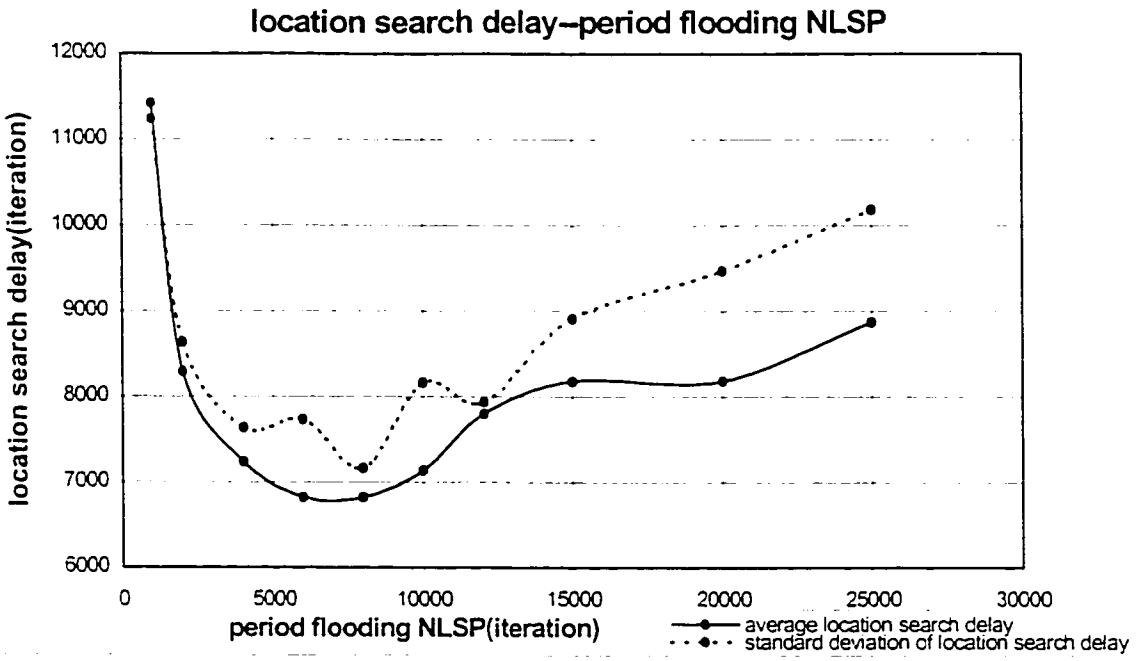


Figure 4.19: Location search delay-period flooding NLSP

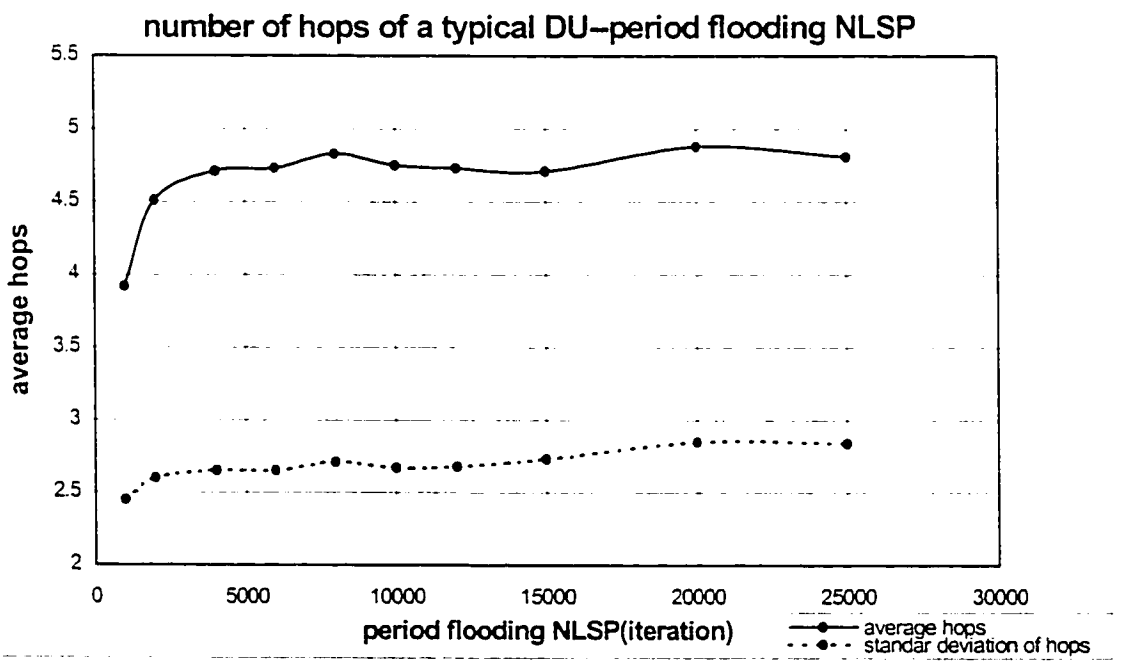


Figure 4.20: Number of hops of a typical DU-period flooding NLSP

Test condition:
Simulation runs 50000 iterations. Maximum speed of nodes: 30km/h. Input traffic rate: 1 packet/iteration. Period flooding ZLSP: 25000 iterations. Data packet timeout: 1200 iterations. Hop limit: 20. Link time: 250 iteration.

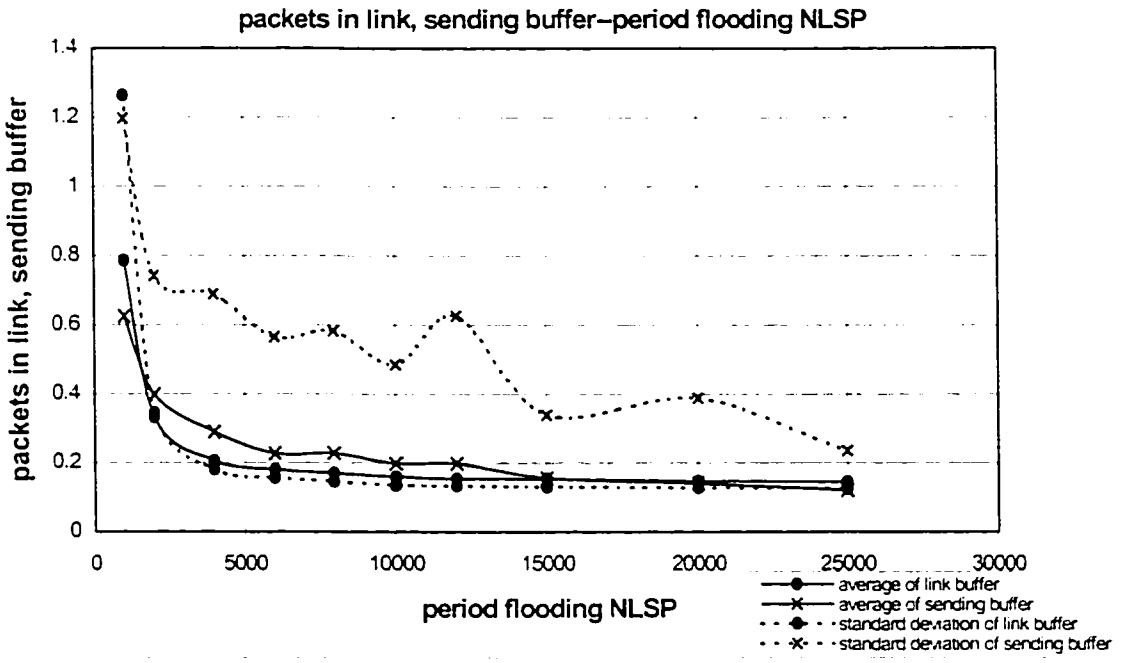


Figure 4.21: Packets in link, sending buffer-period flooding NLSP

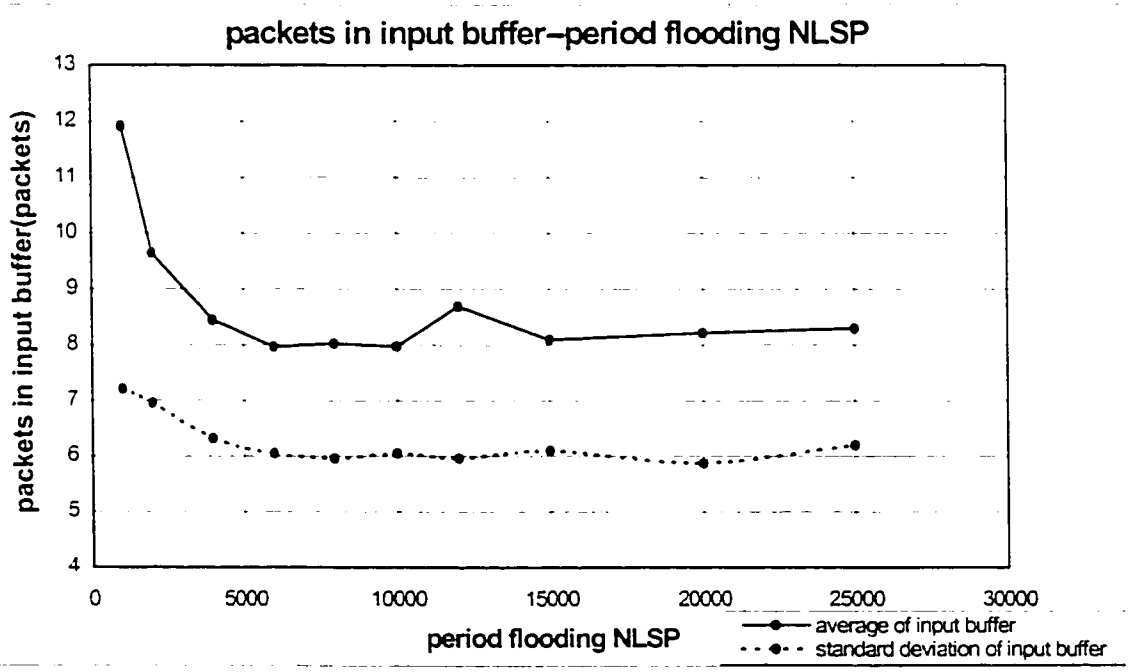


Figure 4.22: Packets in input buffer-period flooding NLSP

Test condition:
Simulation runs 50000 iterations. Maximum speed of nodes: 30km/h. Input traffic rate: 1 packet/iteration. Period flooding ZLSP: 25000 iterations. Data packet timeout: 1200 iterations. Hop limit: 20. Link time: 250 iteration.

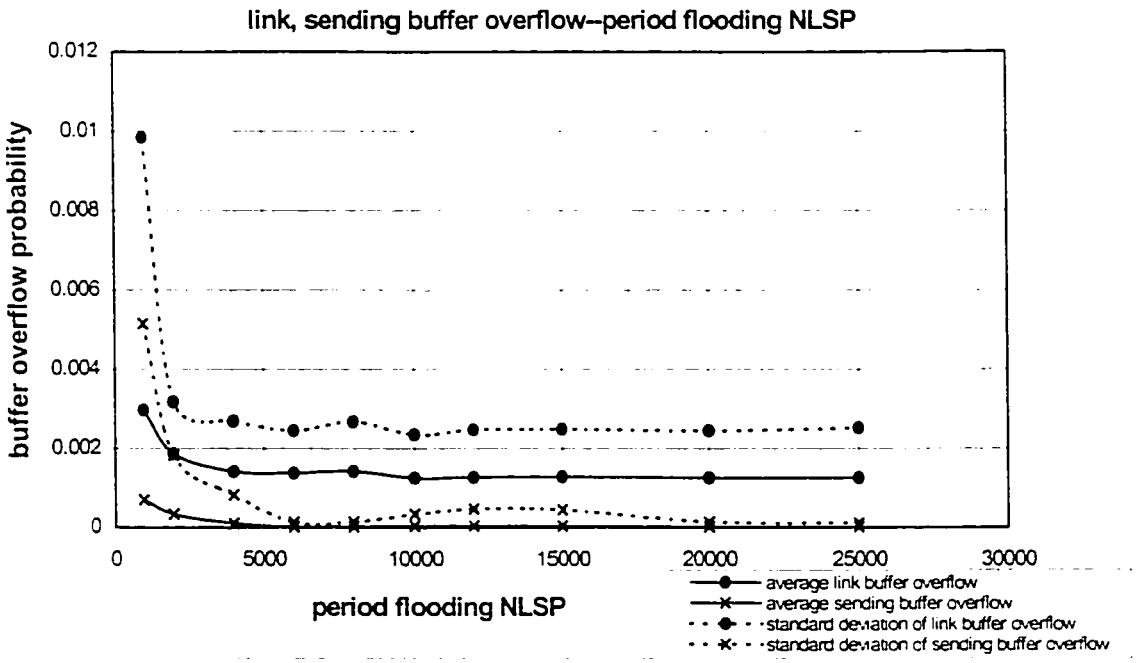


Figure 4.23: Link, sending buffer overflow-period flooding NLSP

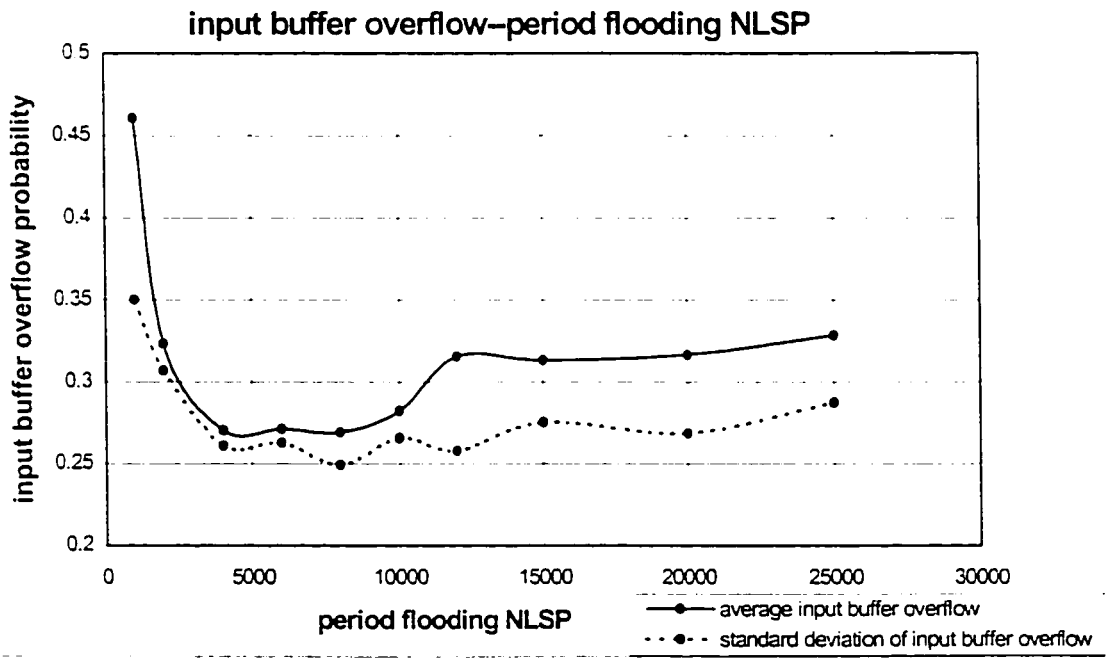


Figure 4.24: Input buffer overflow-period flooding NLSP

Test condition:
Simulation runs 50000 iterations. Maximum speed of nodes: 30km/h. Input traffic rate: 1 packet/iteration. Period flooding ZLSP: 25000 iterations. Data packet timeout: 1200 iterations. Hop limit: 20. Link time: 250 iteration.

4.4 Network Performance under Different ZLSP Flooding Period

Similar to the effect of NLSP flooding period, ZLSP flooding puts extra overhead on network while updating network link information. Because the zone level link topology changes much slower than node level link topology when node density is high, and unlike effect of NSLP flooding, long ZLSP flooding periods may reduce network overhead without obvious damage to network performance.

We observe this phenomenon clearly in fig. 4.25. As flooding period of ZLSP increases, the overhead monotonously decreases and network efficiency monotonously increases. This indicates that zone level link information needs to be updated less frequently than node level link information. Transfer delay also monotonously decreases in fig. 4.26.

In fig. 4.27, location search delay fluctuates between period flooding ZLSP = 10000 iterations and 30000 iterations but finally decreases as period flooding ZLSP increases. Because zone level topology information is based on each node level topology information, the relationship between NLSP flooding period and ZLSP flooding period may cause this phenomenon.

To avoid network congestion caused by overhead, we randomly select the starting point of flooding NLSP and flooding ZLSP. Specifically in our model, we select iterations of multiple of node ID as a starting point of flooding NLSP for that node and iterations of multiple of zone ID as a starting point of flooding ZLSP for that zone. If ZLSP flooding is shortly after most nodes in that zone have finished flooding NLSP, the new generated ZLSP is based on up-to-date link information and is reliable. Otherwise, the new generated ZLSP is based on out-of-date link information. Furthermore, if we select flooding period of NLSP and ZLSP in a certain way such the starting points of flooding ZLSP of most zones take palce shortly after node level link information update, reliable paths will be available for LR packet and location search delay will decrease. Different

relationship between NLSP flooding period and ZLSP flooding period may cause the fluctuation of transfer delay in fig. 4.27.

If flooding period of ZLSP is very short (less than 5000), average and standard deviation of the number of hops of a typical data packet decrease due to more overhead traffic as in fig. 4-28. But if flooding period of ZLSP is long enough, the number of hops of a typical data packet does not vary much. In this case, the number of hops of a typical data packets depends more on the location of selected destinations.

Numbers of packets in all three buffers decrease as flooding period of ZLSP increases in fig. 4.29 and fig. 4.30. It is demonstrated again that zone level link topology changes much slower than node level link topology and needs to be updated less frequently.

Variation of buffer overflow probability of different buffers are shown in fig. 4.31 and fig. 4.32. Only overflow probability of input buffer decreases noticeably as flooding period of ZLSP increases. It is correspondent to higher network efficiency in fig. 4.25.

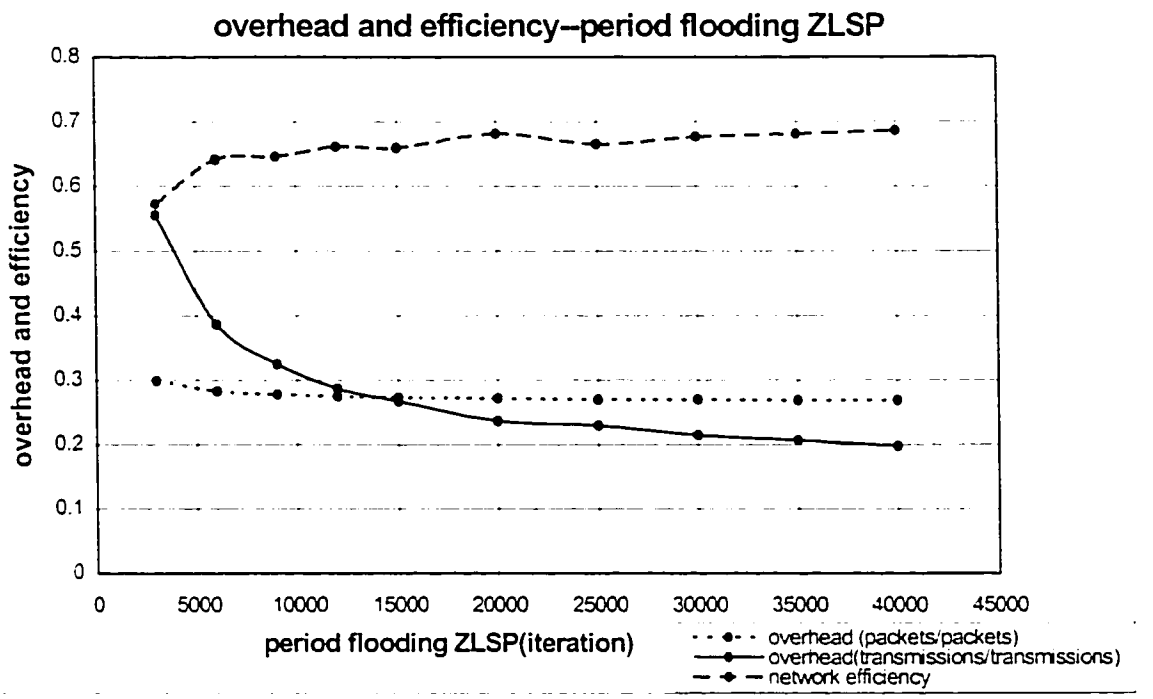


Figure 4.25: Overhead and efficiency-period flooding ZLSP

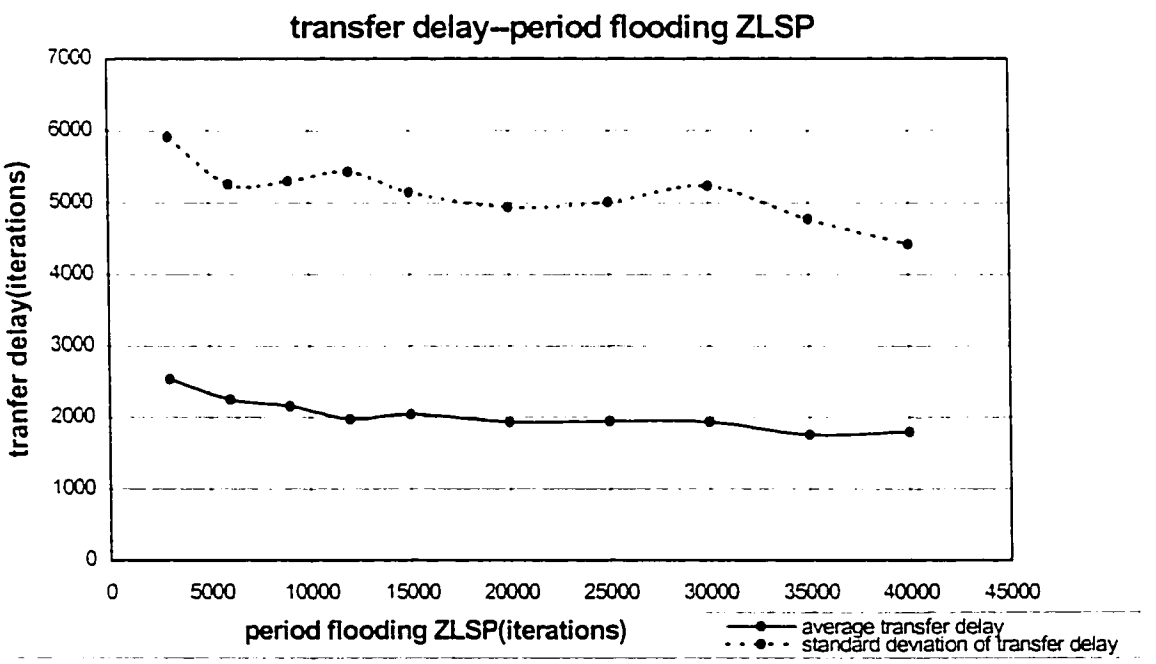


Figure 4.26: Transfer delay-period flooding ZLSP

Test condition:

Simulation runs 50000 iterations. Maximum speed of nodes: 30km/h. Input traffic rate: 1 packet/iteration. Period flooding NLSP: 10000 iterations. Data packet timeout: 1200 iterations. Hop limit: 20. Link time: 250 iteration.

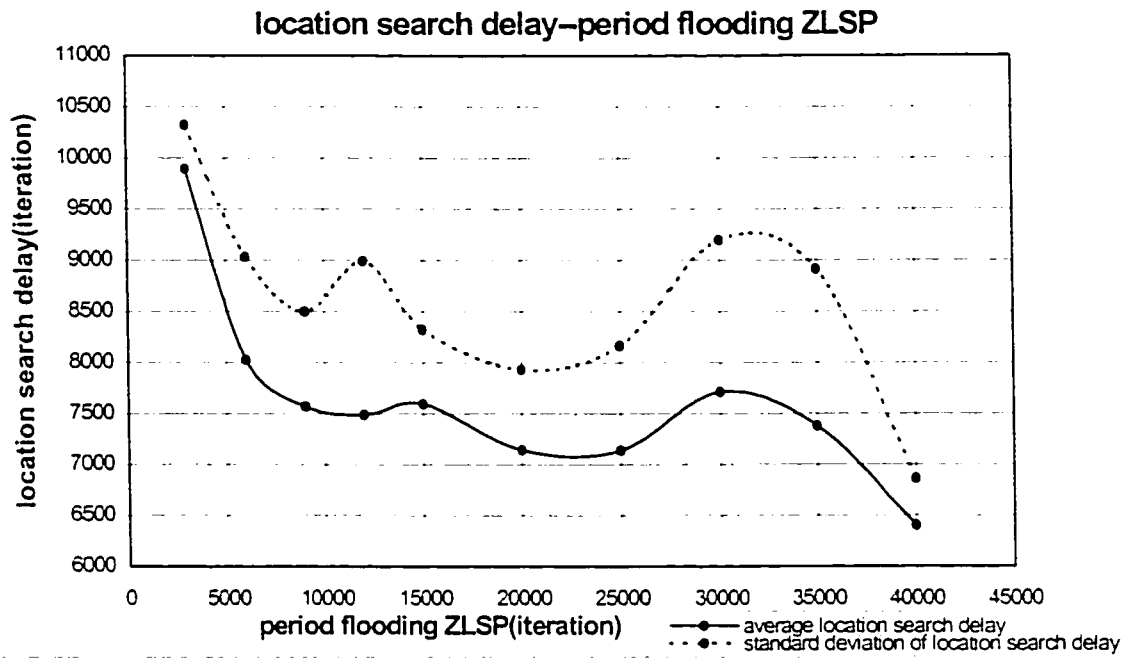


Figure 4.27: Location search delay-period flooding ZLSP

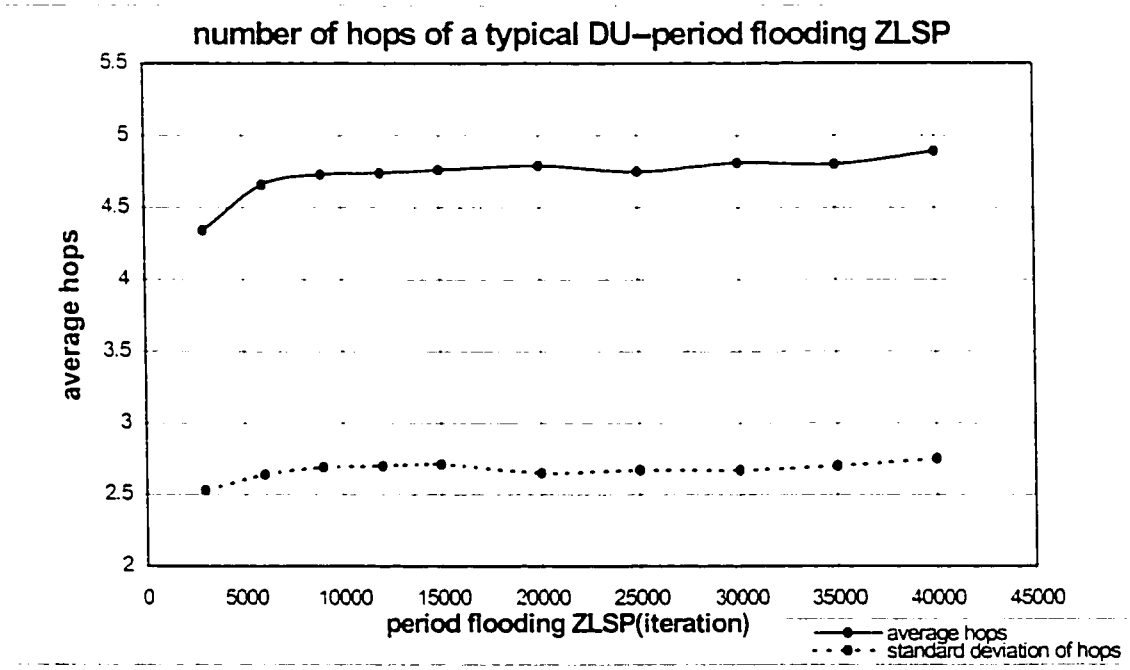


Figure 4.28: Number of hops of typical DU-period flooding ZLSP

Test condition:
Simulation runs 50000 iterations. Maximum speed of nodes: 30km/h. Input traffic rate: 1 packet/iteration. Period flooding NLSP: 10000 iterations. Data packet timeout: 1200 iterations. Hop limit: 20. Link time: 250 iteration.

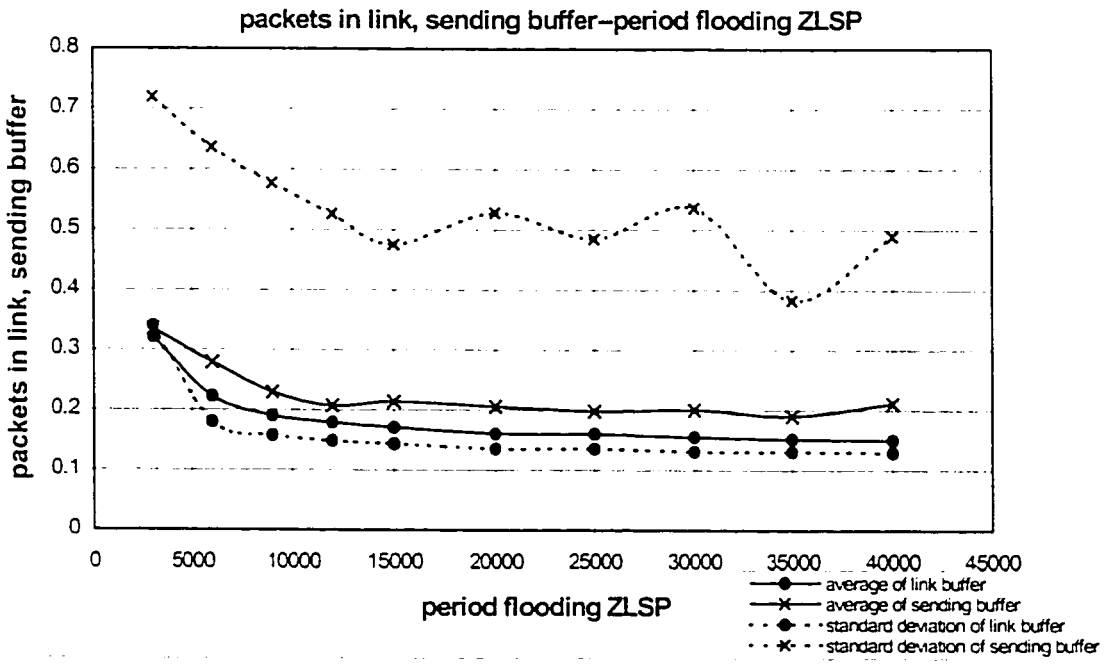


Figure 4.29: Packets in link, sending buffer-period flooding ZLSP

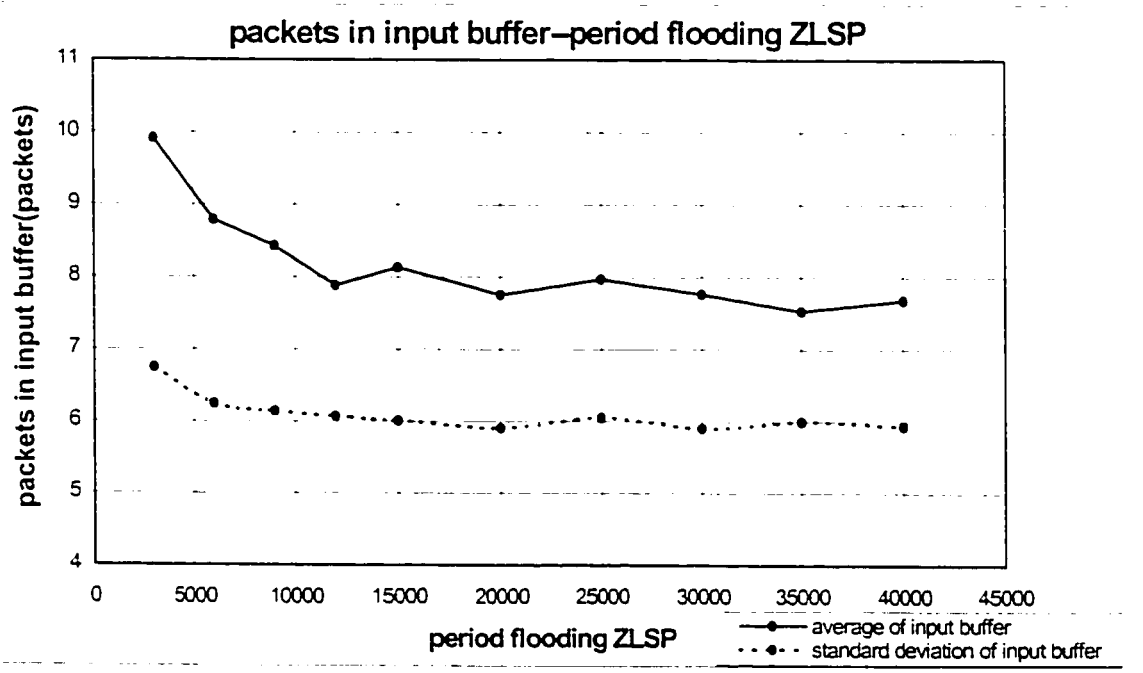


Figure 4.30: Packets in input buffer-period flooding ZLSP

Test condition:
Simulation runs 50000 iterations. Maximum speed of nodes: 30km/h. Input traffic rate: 1 packet/iteration. Period flooding NLSP: 10000 iterations. Data packet timeout: 1200 iterations. Hop limit: 20. Link time: 250 iteration.

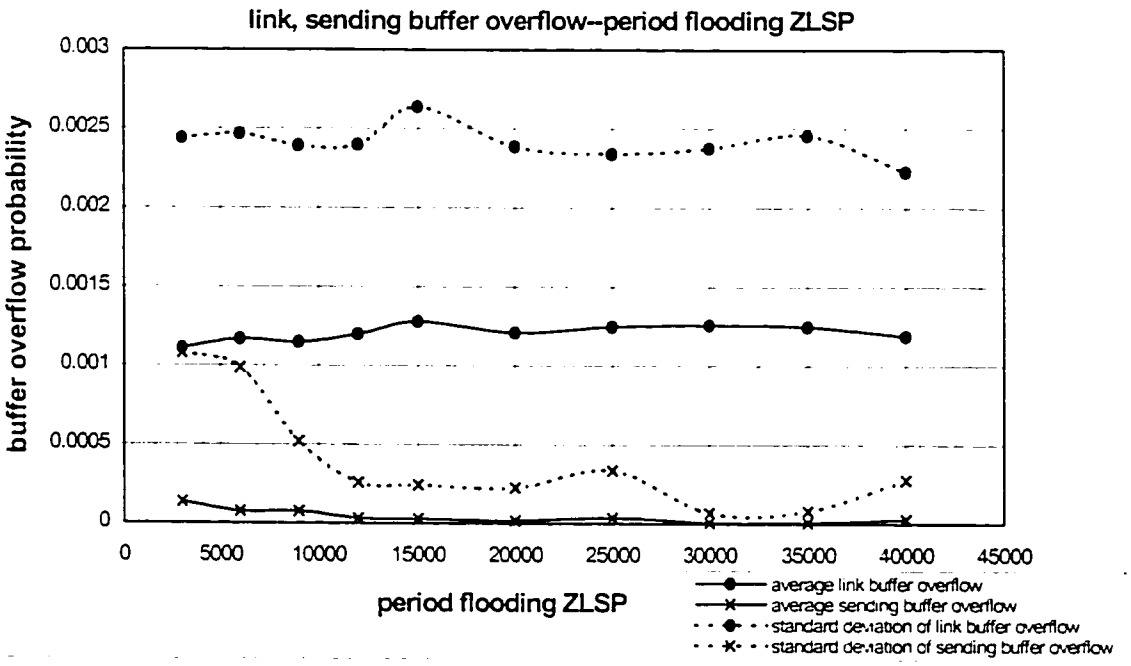


Figure 4.31: Link, sending buffer overflow-period flooding ZLSP

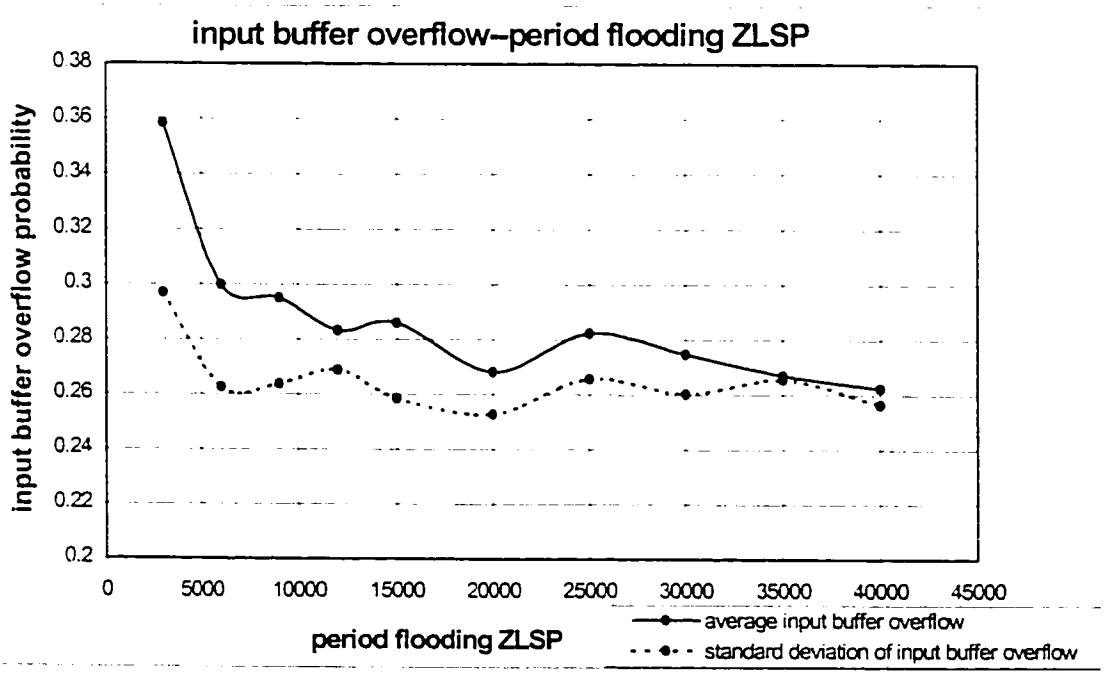


Figure 4.32: Input buffer overflow-period flooding ZLSP

Test condition:
Simulation runs 50000 iterations. Maximum speed of nodes: 30km/h. Input traffic rate: 1 packet/iteration. Period flooding NLSP: 10000 iterations. Data packet timeout: 1200 iterations. Hop limit: 20. Link time: 250 iteration.

4.5 Network Performance under Different Hop Limits

Hop limit is defined as the maximum number of hops after which the packet will be dropped. It is used to limit looping. Packets circulates in a loop will have extraordinary high hop count and thus will be dropped due to hop limit.

Loops may exists in MZHLS due to out-of-date link information. Data packets with extraordinary high hop count are found in our model but they occupy a very small proportion among all data packets (as small as 1/3000 to 1/10000). Fig. 4.33 to fig. 4.40 show simulation results of the effect of different hop limit on network performance.

As we see from fig. 4.33 to fig. 4.36, if hop limit is greater than 20, network performance will not be significantly affected by hop limit. It indicates that the length of normal paths is typically less than 20. Packets that hop above 20 hops may circulate in a loop and may be treated as extraordinary high hop count. Dropping of such packets will not decrease the network performance.

Fig. 4.37 to fig. 4.40 shows the average buffer content. Link buffer is not affected by different hop limits. Because control packets have higher priority over data packets, number of packets in link buffer and overflow probability of link buffer will not vary as hop limit of data packet varies. Packet number and overflow probability of sending buffer and input buffer vary as hop limit changes. But if hop limit is greater than 20, the variation of contents in link and input buffer is very small.

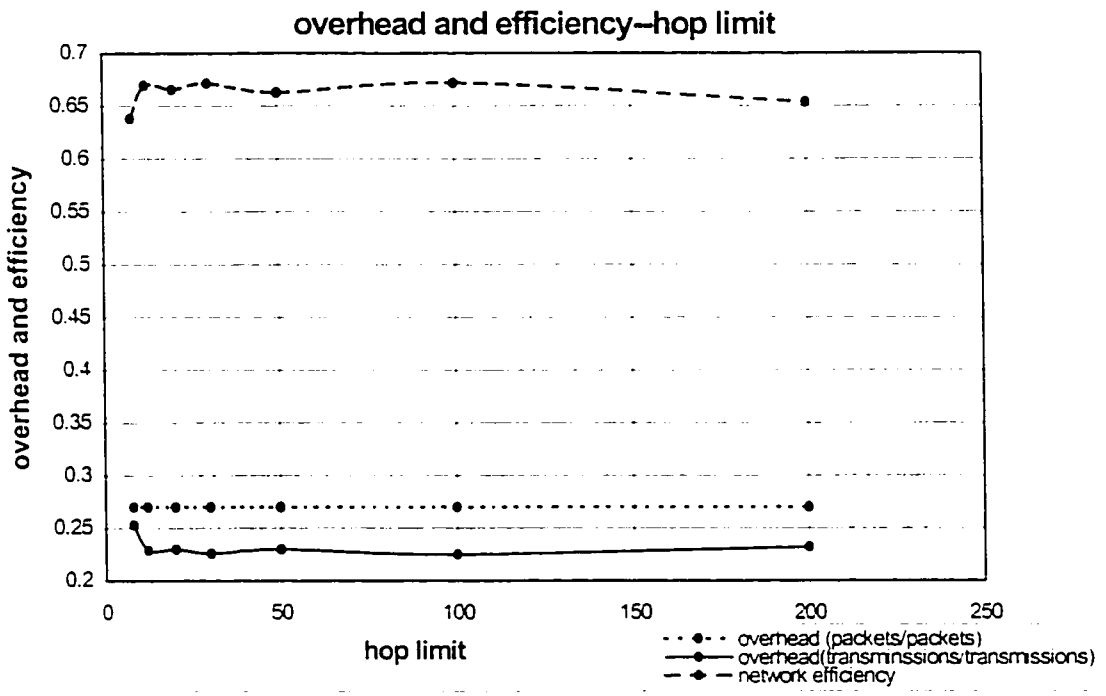


Figure 4.33: Overhead and efficiency-hop limit

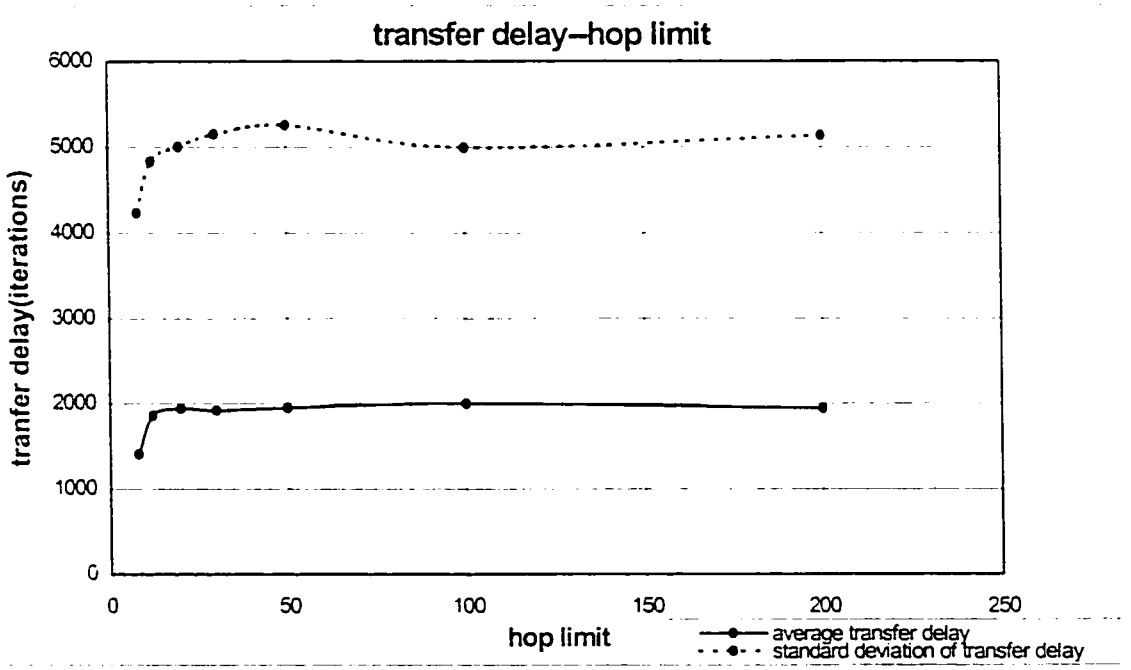


Figure 4.34: Transfer delay-hop limit

Test condition:
Simulation runs 50000 iterations. Maximum speed of nodes: 30km/h. Input traffic rate: 1 packet/iteration. Period flooding NLSP: 10000 iterations. Period flooding ZLSP: 25000 iterations. Data packet timeout: 1200 iterations. Link time: 250 iteration.

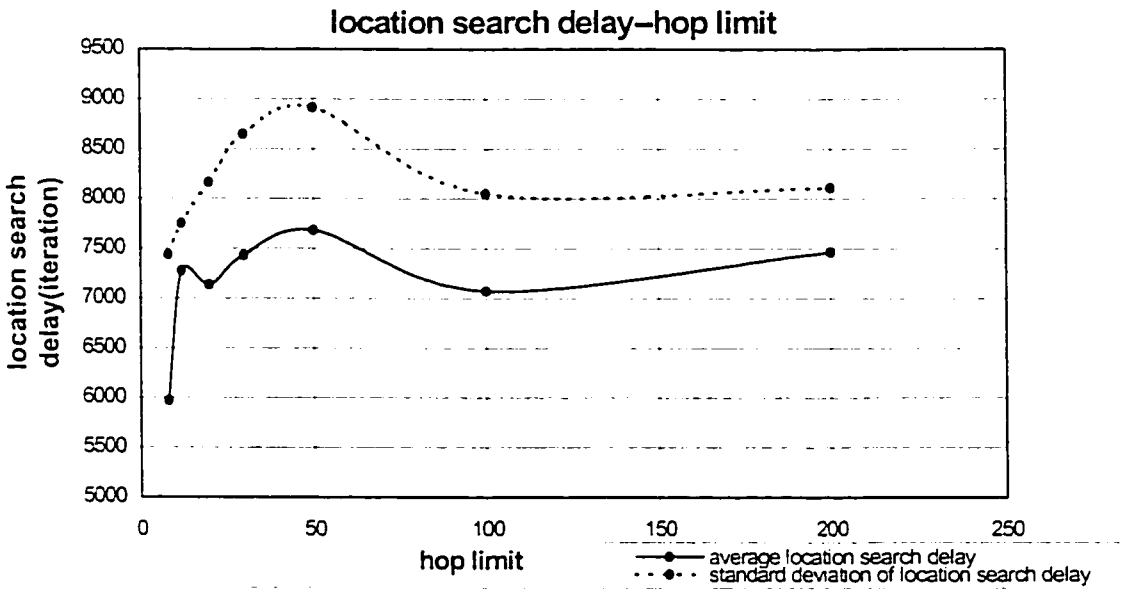


Figure 4.35: Location search delay-hop limit

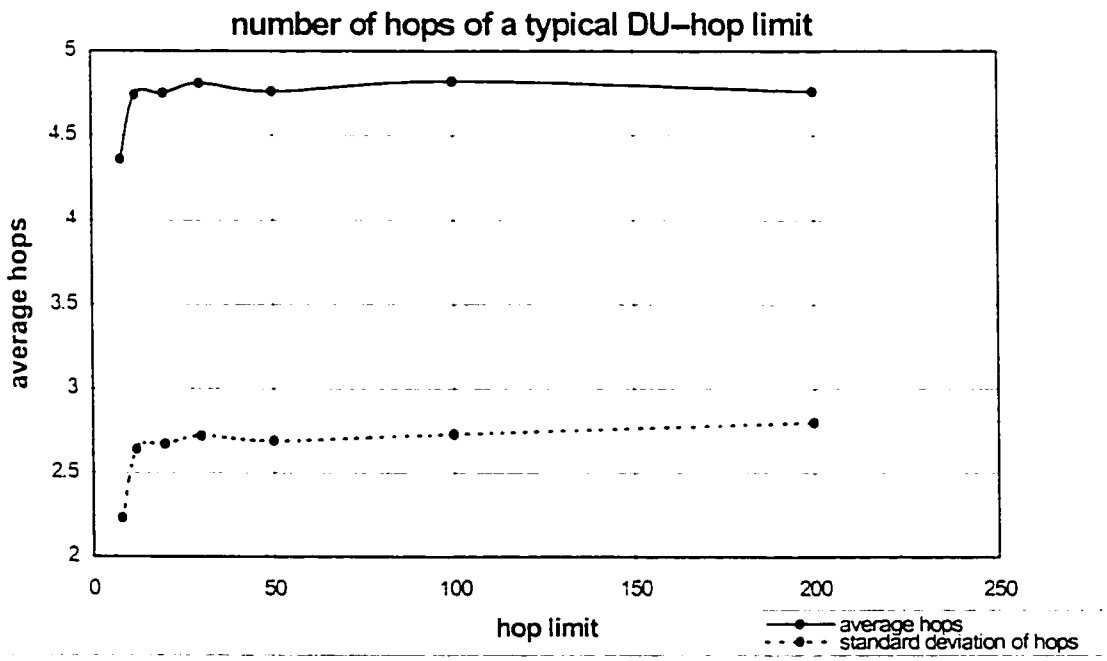


Figure 4.36: Number of hops of a typical DU-hop limit

Test condition:
Simulation runs 50000 iterations. Maximum speed of nodes: 30km/h. Input traffic rate: 1 packet/iteration. Period flooding NLSP: 10000 iterations. Period flooding ZLSP: 25000 iterations. Data packet timeout: 1200 iterations. Link time: 250 iteration.

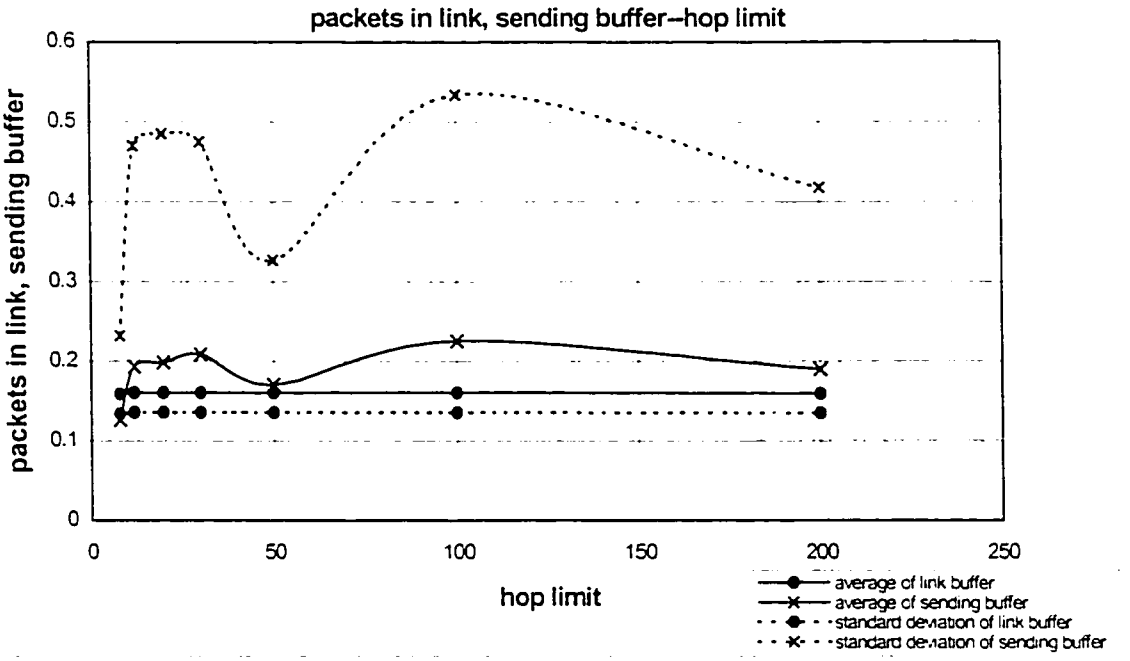


Figure 4.37: Packets in link, sending buffer-hop limit

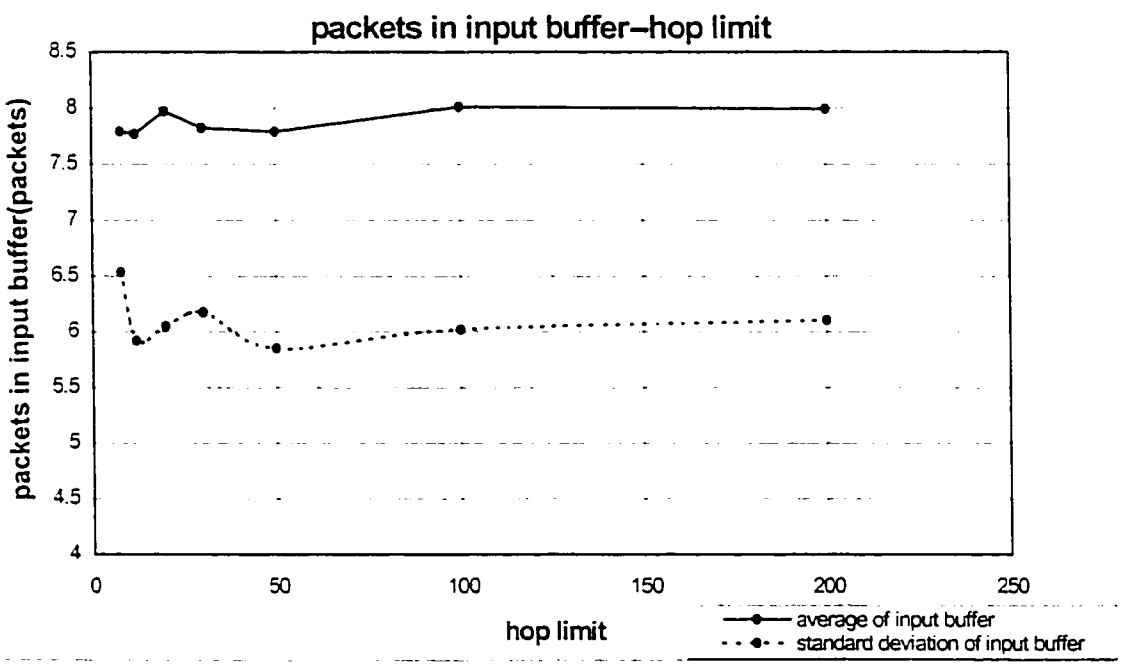


Figure 4.38: Packets in input buffer-hop limit

Test condition:
Simulation runs 50000 iterations. Maximum speed of nodes: 30km/h. Input traffic rate: 1 packet/iteration. Period flooding NLSP: 10000 iterations. Period flooding ZLSP: 25000 iterations. Data packet timeout: 1200 iterations. Link time: 250 iteration.

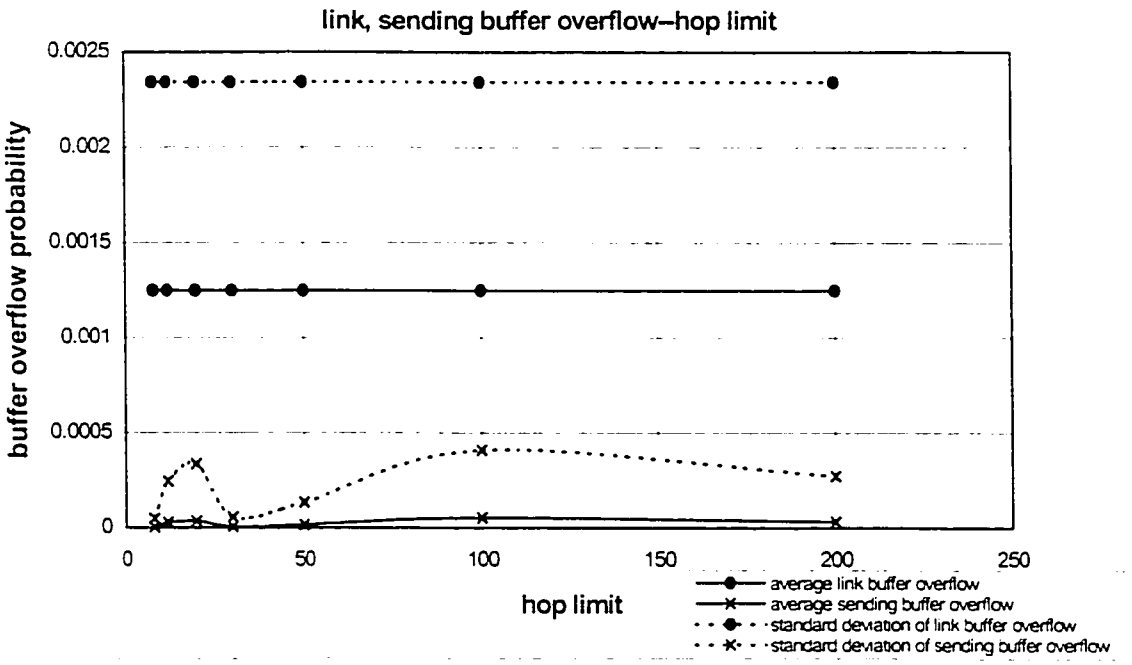


Figure 4.39: Link, sending buffer overflow-hop limit

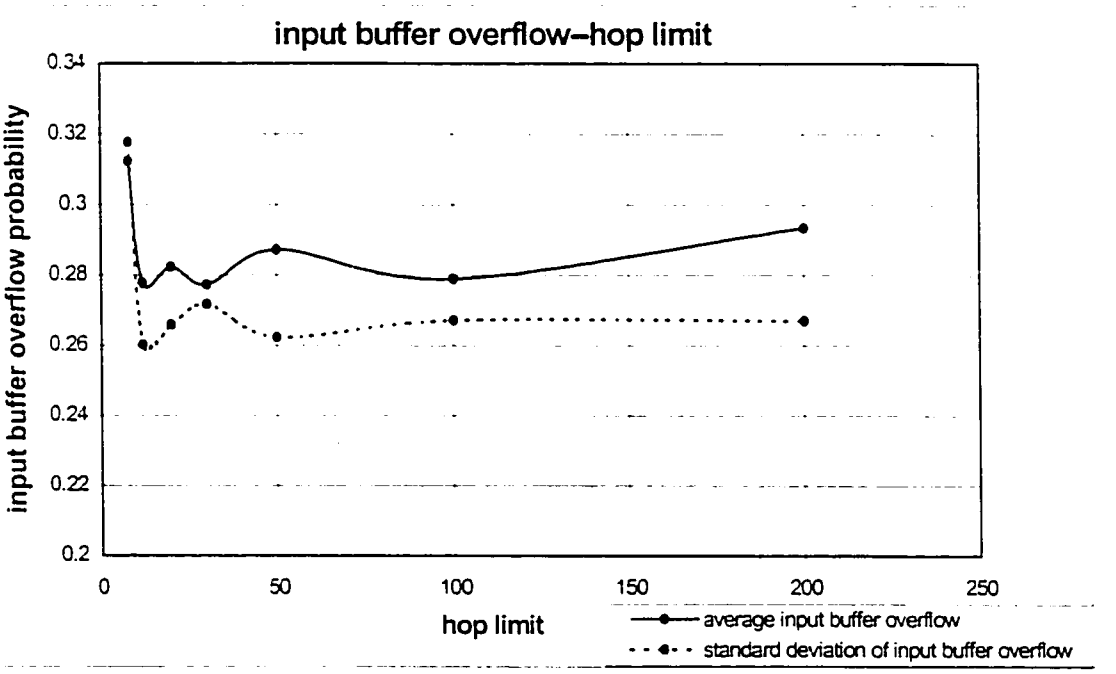


Figure 4.40: Input buffer overflow-hop limit

Test condition:
Simulation runs 50000 iterations. Maximum speed of nodes: 30km/h. Input traffic rate: 1 packet/iteration. Period flooding NLSP: 10000 iterations. Period flooding ZLSP: 25000 iterations. Data packet timeout: 1200 iterations. Link time: 250 iteration.

4.6 Network Performance under Different Link Time Setting

Each node periodically broadcasts a LKR packet, waits for a certain period to collect the LKRS packets from its neighboring nodes and then broadcasts its new generated NLSP. It is defined as link procedure. The period from the time the source node generates a LK to the time it stop waiting for replies from its neighbor nodes is defined as link time.

The amount of time that a neighbor node needs to reply to the LKR packet from a requesting node depends on the instant network traffic, density of neighbor nodes and number of packets queuing in their link buffer. There are so many factors that may delay the reply from a neighboring node, it is very difficult to estimate the average and maximum time before which all LKRS from all neighboring nodes are collected. But best link time for network performance may exist for two reasons: short link time may cause loss of LKRS packets while long link time may make the information obtained out-of-date. Another advantage of limiting link time is: a neighbor node which has more packets in its buffer may experience heavy traffic and thus may need long link time to reply to the link request node. Due to limitation of link time, the requesting node may exclude the LKRS packet from this neighbor node. So future data packets of the requesting node will be routed to other nodes than this neighbor node. Such mechanism helps to distribute network traffic and avoid network bottle necks.

In this simulation model, each LKR packet is set to the maximum link time in its timeout field. Each iteration, link time of LKR and LKRS packets decreases by 1. If a node gets a LKR packet which has non-zero link time, it will insert a LKRS packet to its link buffer and copy the link time value from LKR packet to LKRS packet. LKRS packet will be dropped if its link time reaches zero. The requesting node will refuse all LKRS packets after it waits a maximum link time. Such mechanism ensures that the time used by request-reply procedure is less than maximum link time

By trying different setting of link time, experimental result of best link time is easily found. Fig 4.41 to fig. 4.48 show that network performance under different link time.

Fig. 4.41 shows that network has maximum efficiency and minimum overhead (transmissions/transmissions) if link time is longer than 200 iterations and less than 600 iterations except that overhead (packets/packets) is affected little by link time. Minimum transfer delay and location search delay are also obtained within this range in fig.4.42 and fig. 4.43. Similar situation happens in fig. 4.44 on number of hops of a typical data packet. So link time between 200 to 600 iterations can be thought as best link time.

Number of packets in link buffer and sending buffer increase under best link time in fig.4.45 because more routing information is available . Overflow probability of link buffer and sending buffer also increase in fig. 4.47 due to more packets in these buffers.

As a conclusion, link time is a predefined value of duration. If it is properly defined, a node may get as more available NLSP packets from other nodes as possible while the delay for collecting these packets is not too long. In such a case, number of packets in the input buffer and overflow probability of input buffer have minimum value under best link time (fig. 4.46 and 4.48). It implies that network efficiency is enhanced due to the best selection of link time.

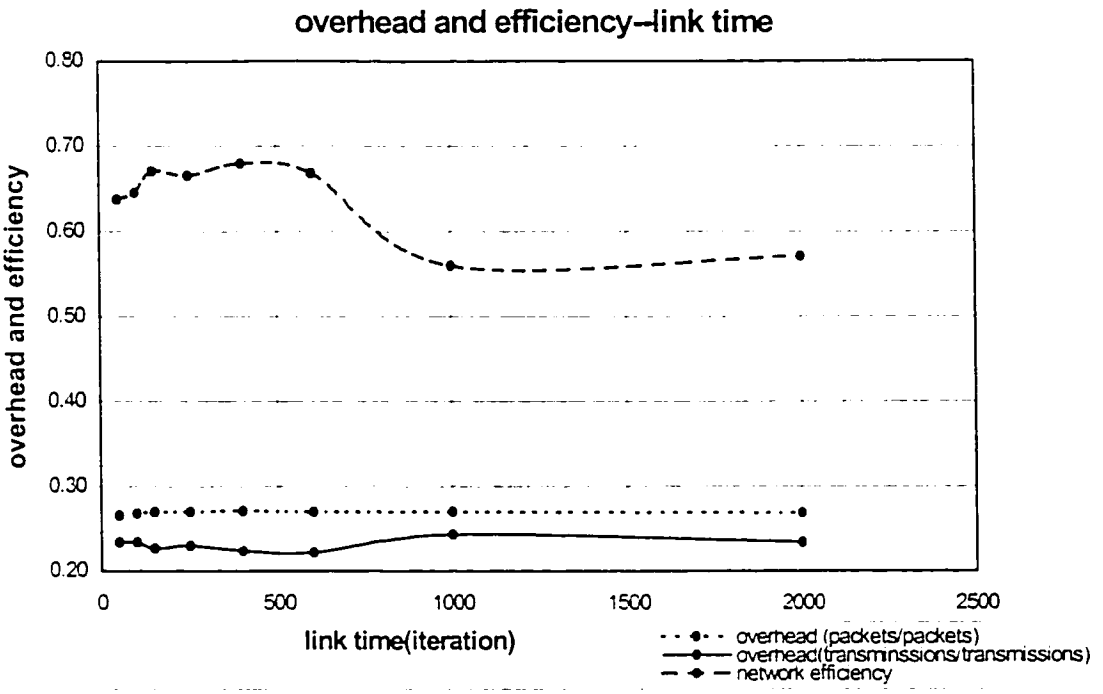


Figure 4.41: Overhead and efficiency-link time

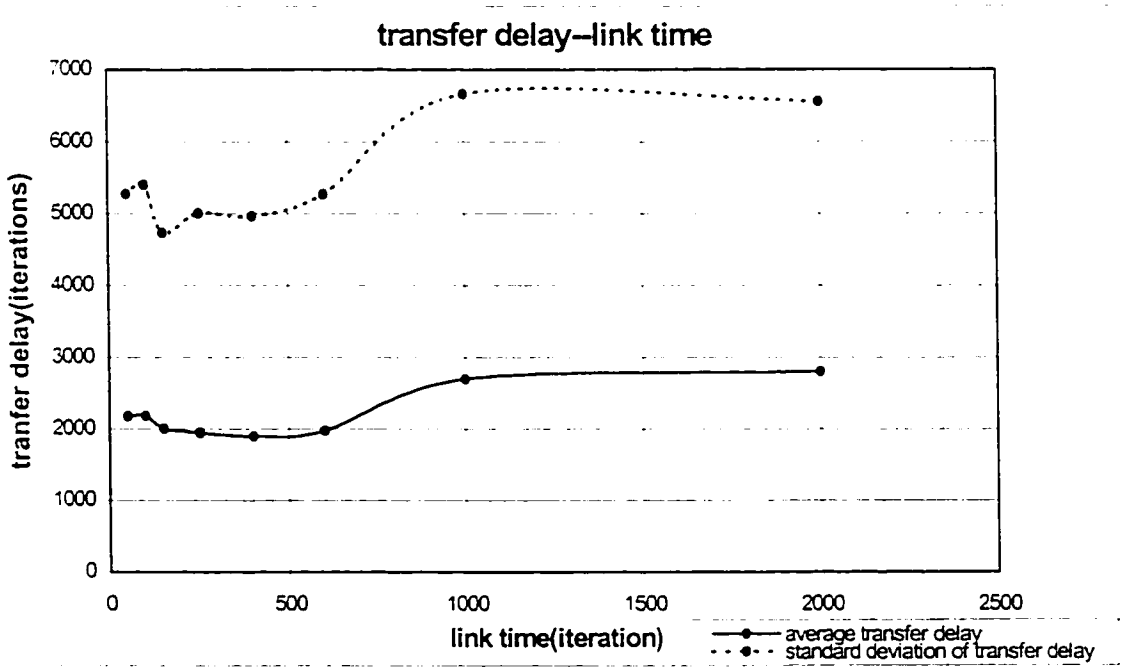


Figure 4.42: Transfer delay-link time

Test condition:

Simulation runs 50000 iterations. Maximum speed of nodes: 30km/h. Input traffic rate: 1 packet/iteration. Period flooding NLSP: 10000 iterations. Period flooding ZLSP: 25000 iterations. Data packet timeout: 1200 iterations. Hop limit: 20

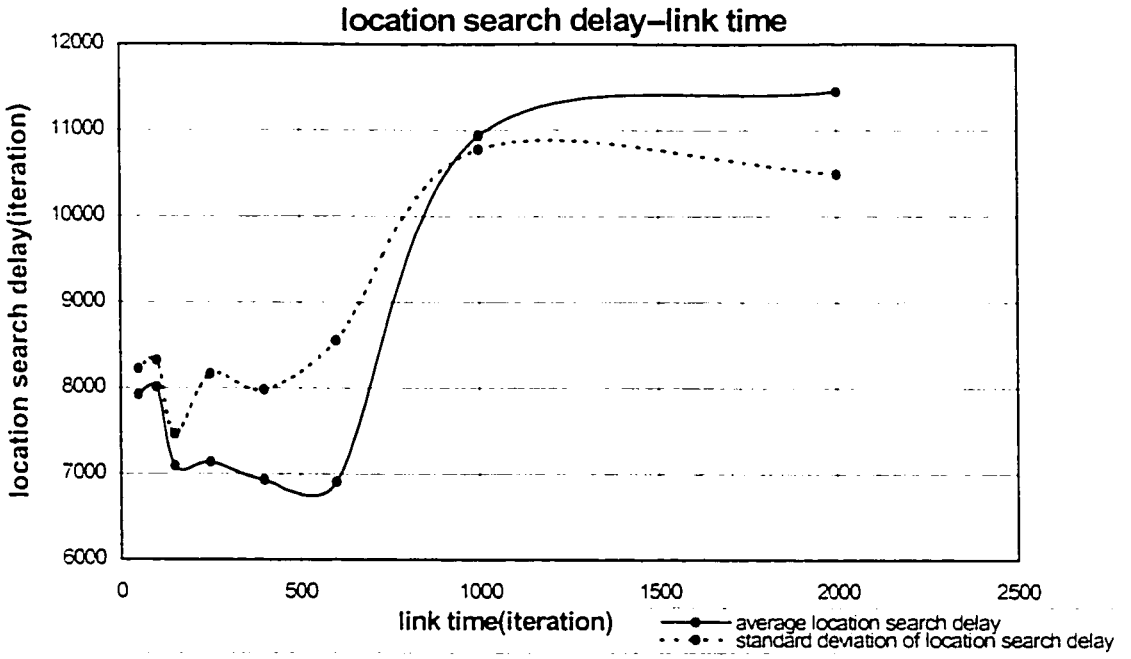


Figure 4.43: Location search delay-link time

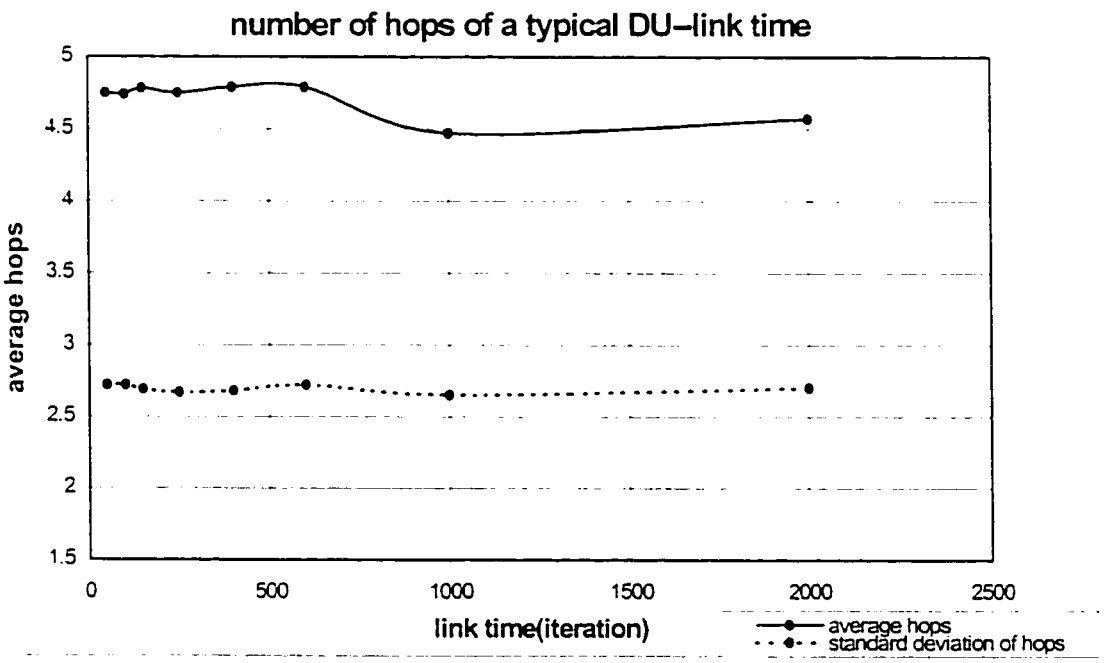


Figure 4.44: Number of hops of a typical DU-link time

Test condition:
Simulation runs 50000 iterations. Maximum speed of nodes: 30km/h. Input traffic rate: 1 packet/iteration. Period flooding NLSP: 10000 iterations. Period flooding ZLSP: 25000 iterations. Data packet timeout: 1200 iterations. Hop limit: 20

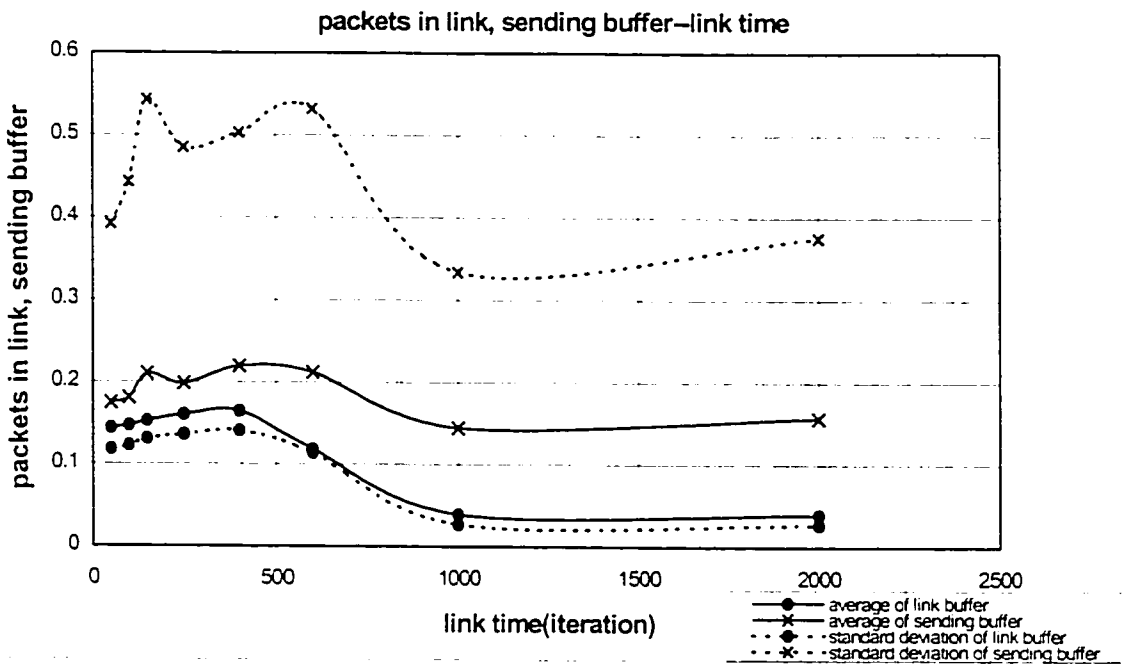


Figure 4.45: Packets in link, sending buffer-link time

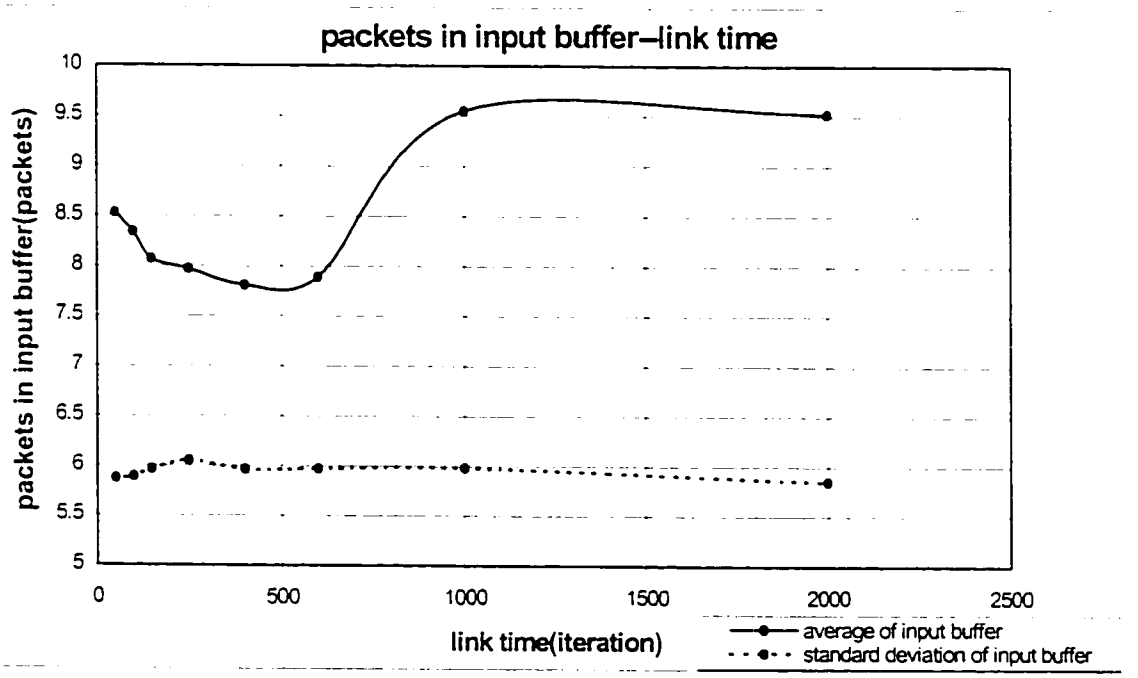


Figure 4.46: Packets in input buffer-link time

Test condition:
Simulation runs 50000 iterations. Maximum speed of nodes: 30km/h. Input traffic rate: 1 packet/iteration. Period flooding NLSP: 10000 iterations. Period flooding ZLSP: 25000 iterations. Data packet timeout: 1200 iterations. Hop limit: 20

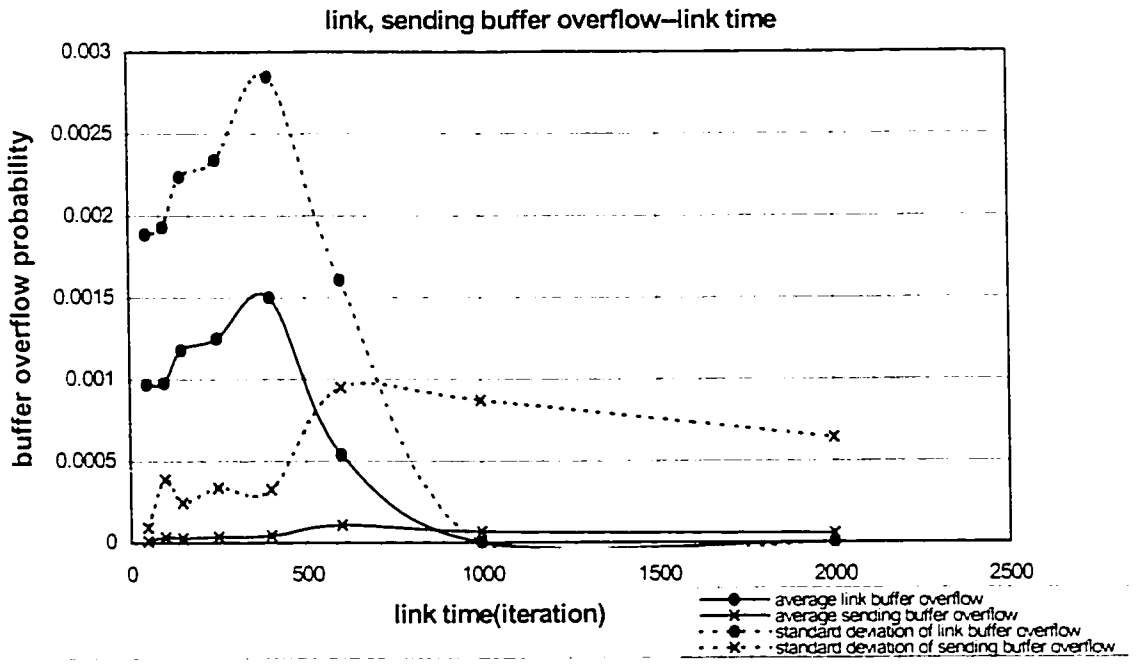


Figure 4.47: Link, sending buffer overflow-link time

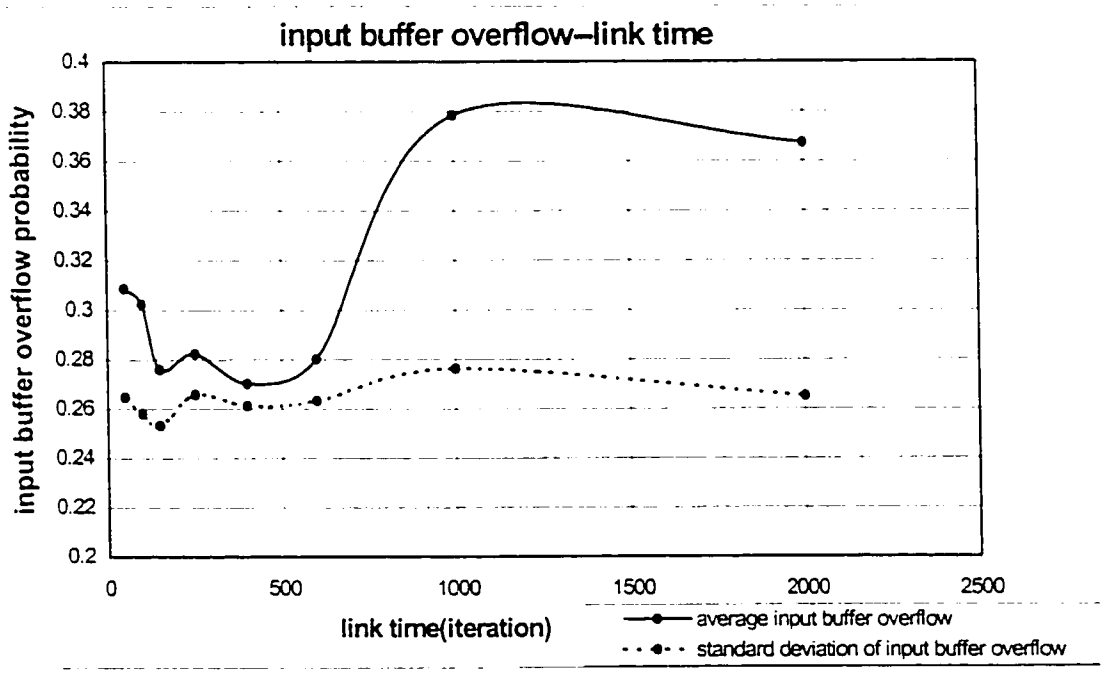


Figure 4.48: Input buffer overflow-link time

Test condition:
Simulation runs 50000 iterations. Maximum speed of nodes: 30km/h. Input traffic rate: 1 packet/iteration. Period flooding NLSP: 10000 iterations. Period flooding ZLSP: 25000 iterations. Data packet timeout: 1200 iterations. Hop limit: 20

4.7 Network Performance under Different Node Mobility

Routing performance degrades in dynamic networks. Network overhead increases due to frequent update of network topology information and packet loss increases due to mobility of nodes. In this section, we study the moving effects of nodes speeds on MZHLS performance.

In the simulation model, each unit speed is defined as a speed of 10km/h and a node at speed of 10km/h moves one unit distance in each iteration. In previous sections, speed is randomly selected from 0, 10km/h, 20km/h, 30km/h for each node (maximum speed is 30km/h). In this section, the maximum speed is increased to 60km/h and 90km/h. Network performance is also tested when all nodes are fixed (where maximum speed is 0).

It is obviously seen in fig. 4.49 that network efficiency decreases and overhead (transmissions/transmissions) increases when maximum speed of nodes increases. Because number of overhead packets generated and number of data packets generated are not affected by node mobility, overhead (packets/packets) keeps constant.

Due to more packet loss, as maximum speed increases, transfer delay increases in fig. 4.50 and location search delay also increases in fig. 4.51.

Because long path to destination is more vulnerable to node mobility, network efficiency decreases faster for packets destined for remote nodes and thus packets with more hop count occupy smaller proportion in more dynamic network. Fig. 4.52 shows the decrement of average and standard deviation of hop count of a typical data packet.

As known from chapter 3, average number of packets in sending buffer depends on the total number of data packets including LR packets that exist in the whole network. In more dynamic network, source nodes frequently wait for timeout to retransmit data packets or LR packets due to packet loss. Those packets actually do not exist in the network, so that total number of data and LR packets decreases in the whole network. This causes decrement of packet number in sending buffer as shown in fig. 4.53. It is also

seen that, on the other hand, node mobility has little effect on packet number in link buffer. Because all control packets are broadcast, no control packet loss will happen whether the nodes move fast or not. Similarly in fig. 4.55, overflow probability of link buffer is affected little by node mobility while overflow probability of sending buffer decrease as node mobility increase.

In fig. 4.54, the number of packets in input buffer and overflow probability of input buffer increase fast as node mobility increases. This corresponds with decrement of network efficiency in fig.4.49.

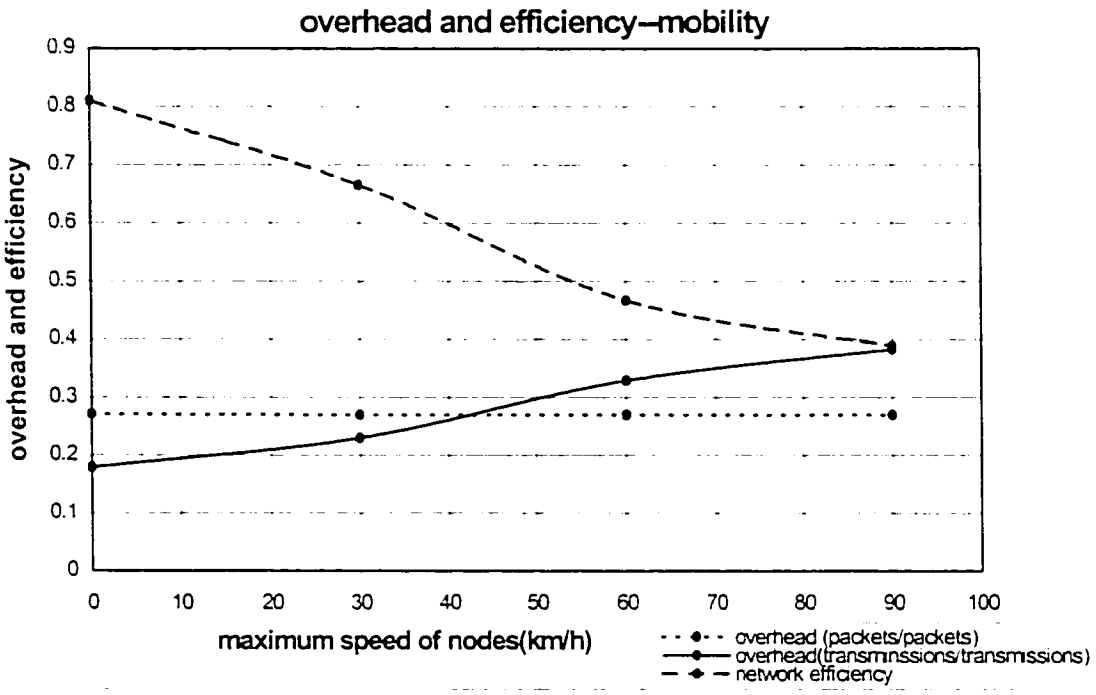


Figure 4.49: Overhead and efficiency-mobility

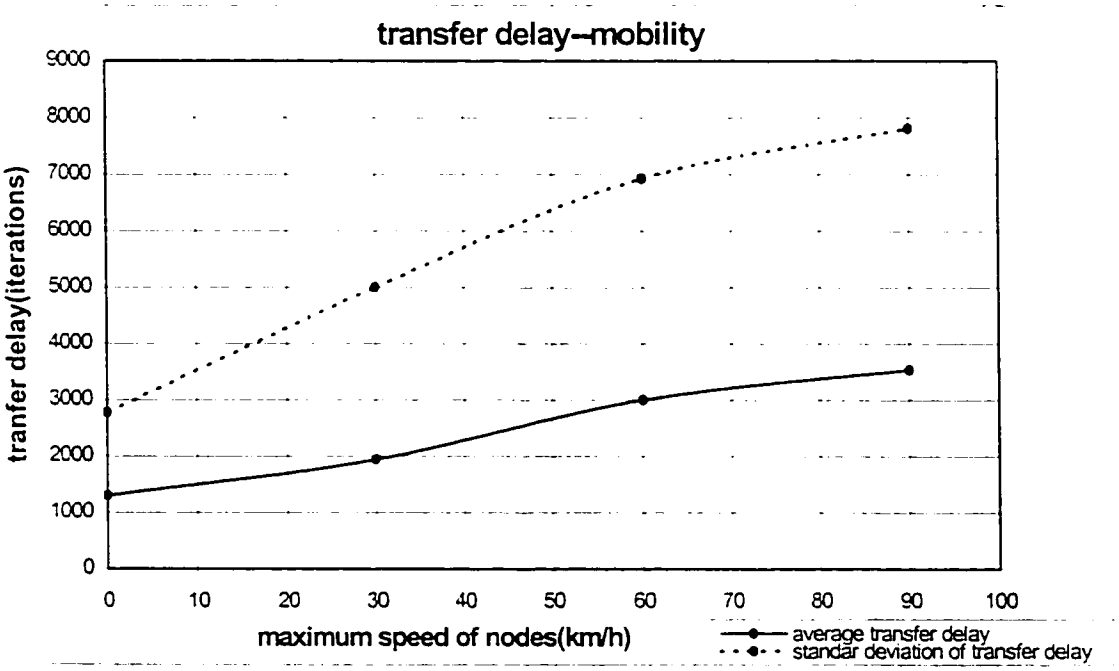


Figure 4.50: Transfer delay-mobility

Test condition:

Simulation runs 50000 iterations. Input traffic rate: 1 packet/iteration. Period flooding NLSP: 10000 iterations. Period flooding ZLSP: 25000 iterations. Data packet timeout: 1200 iterations. Hop limit: 20. Link time: 250 iteration.

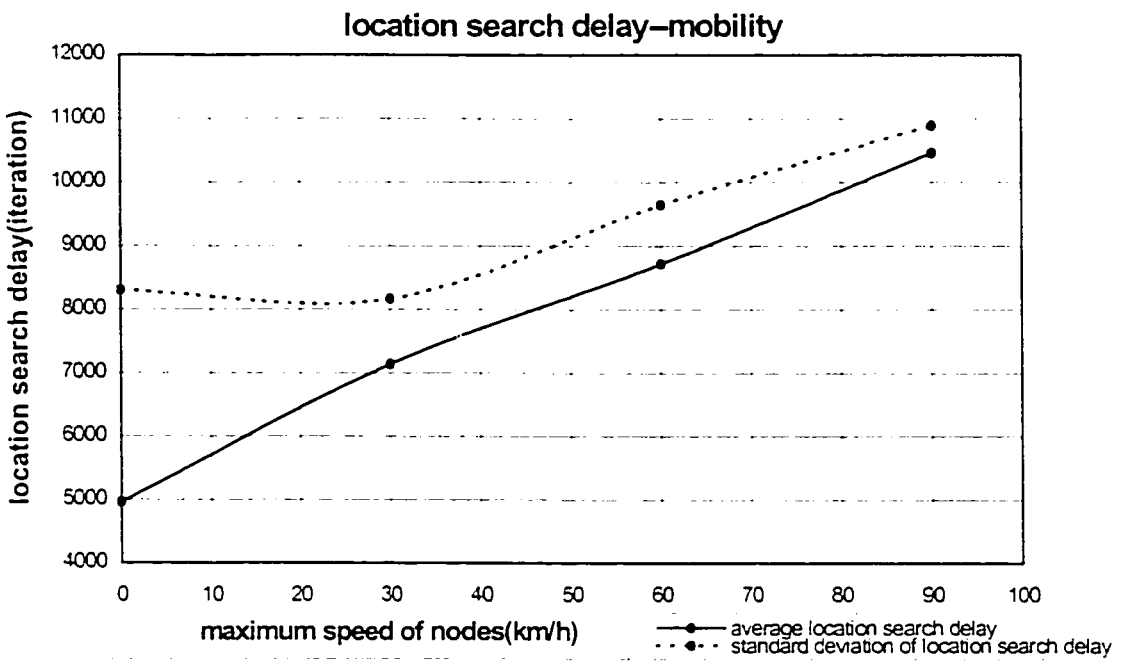


Figure 4.51: Location search delay-mobility

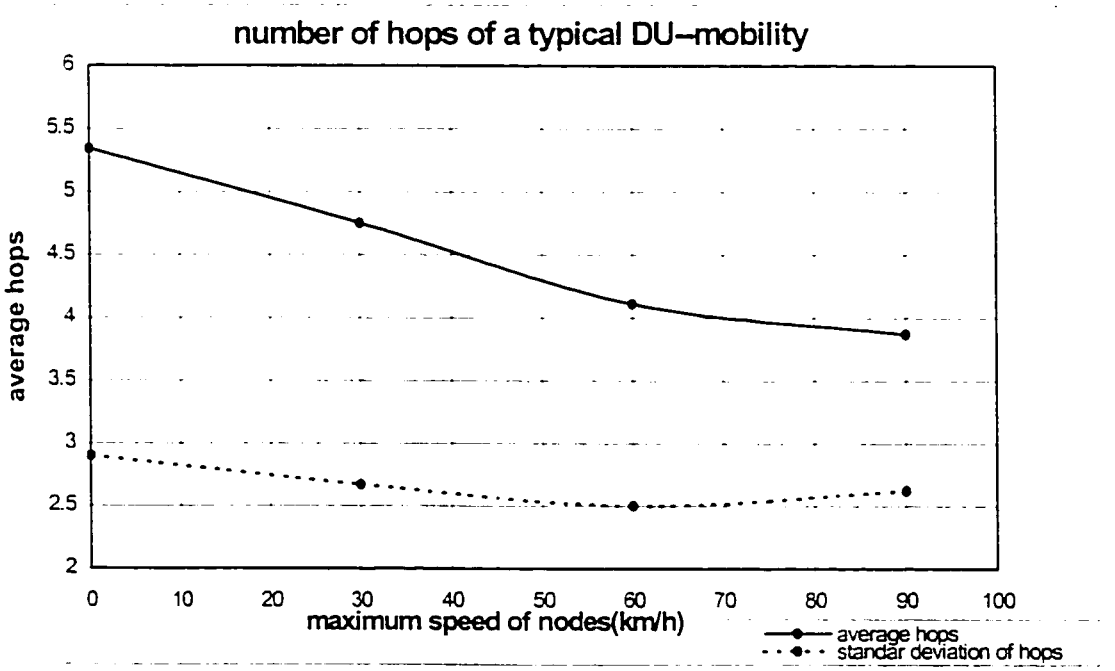


Figure 4.52: Number of hops of a typical DU-mobility

Test condition:
Simulation runs 50000 iterations. Input traffic rate: 1 packet/iteration. Period flooding NLSP: 10000 iterations. Period flooding ZLSP: 25000 iterations. Data packet timeout: 1200 iterations. Hop limit: 20. Link time: 250 iteration.

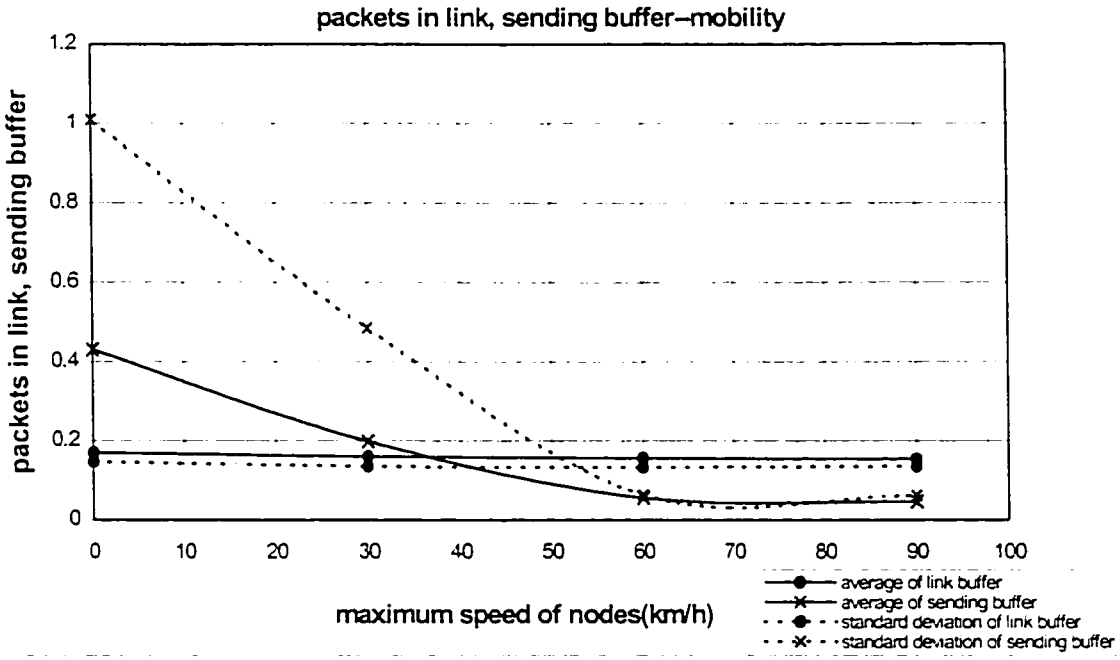


Figure 4.53: Packets in link, sending buffer-mobility

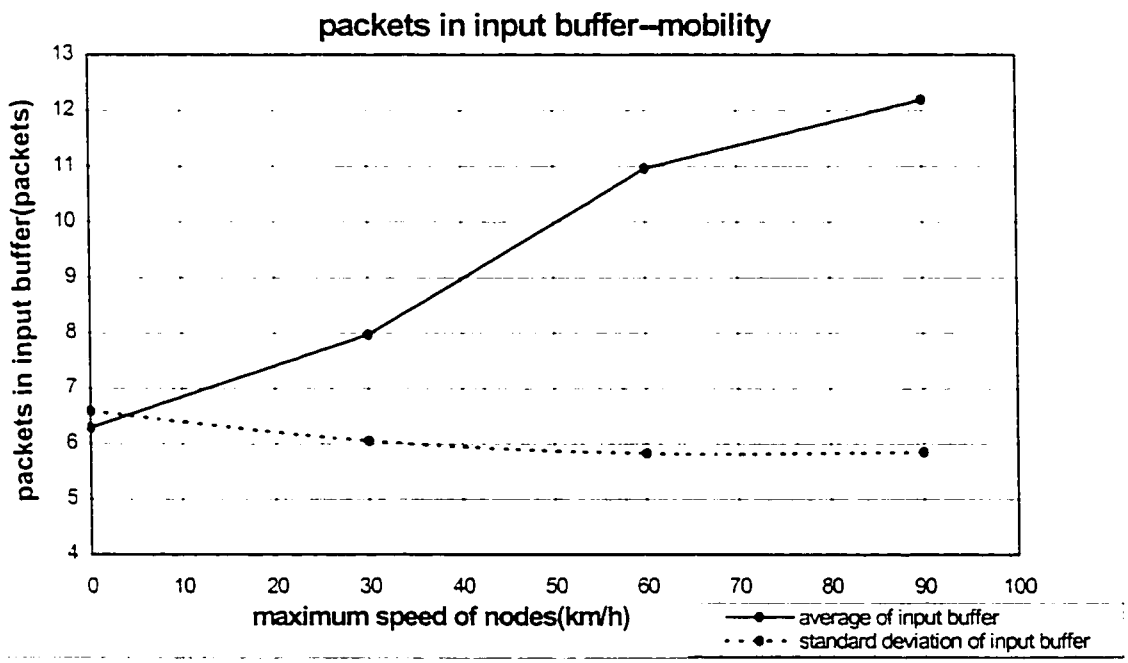


Figure 4.54: Packets in input buffer-mobility

Test condition:
Simulation runs 50000 iterations. Input traffic rate: 1 packet/iteration. Period flooding NLSP: 10000 iterations. Period flooding ZLSP: 25000 iterations. Data packet timeout: 1200 iterations. Hop limit: 20. Link time: 250 iteration.

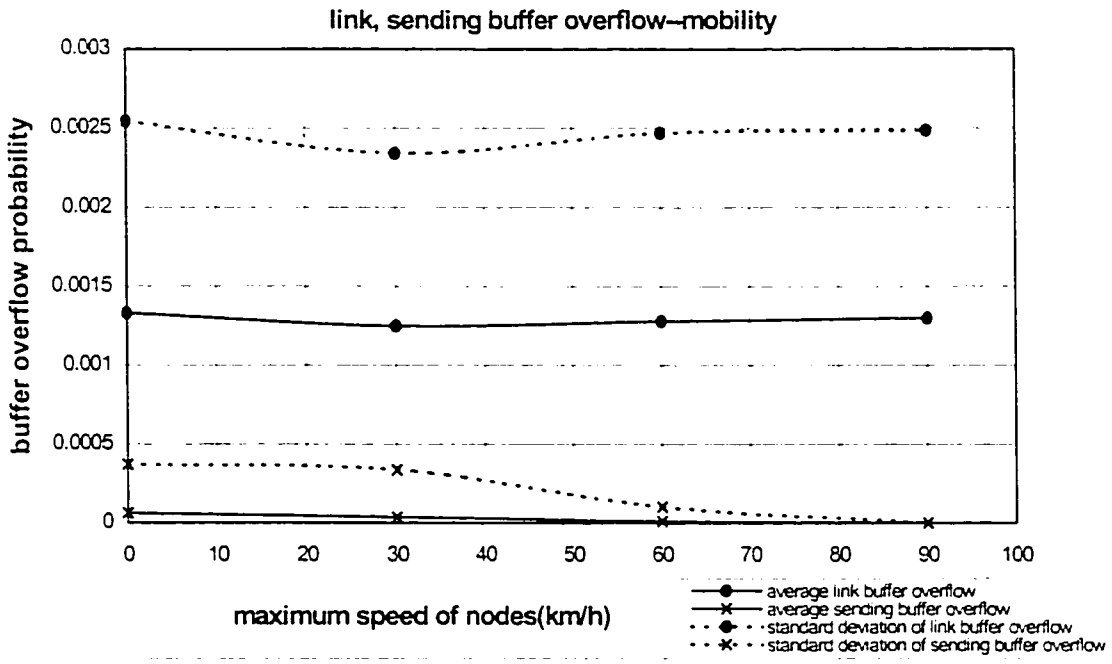


Figure 4.55: Link, sending buffer overflow-mobility

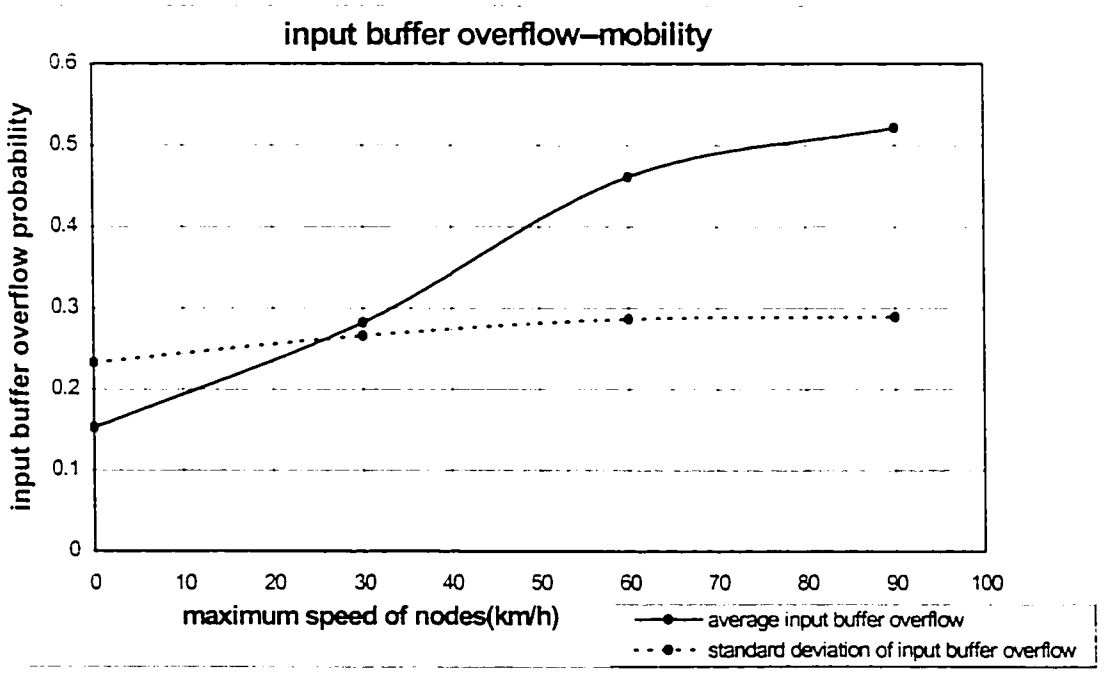


Figure 4.56: Input buffer overflow-mobility

Test condition:
Simulation runs 50000 iterations. Input traffic rate: 1 packet/iteration. Period flooding NLSP: 10000 iterations. Period flooding ZLSP: 25000 iterations. Data packet timeout: 1200 iterations. Hop limit: 20. Link time: 250 iteration.

4.8 Performance Comparison of MZHLS Search Algorithms with ZHLS

Finally in this section, different search algorithms are used to evaluate routing performance and performance of MZHLS is compared with ZHLS.

In this simulation model, network is partitioned into 36 zones. There are 20 periphery zones among all 36 zones. As stated in chapter 3, by searching more nodes (local nodes and neighbor nodes of this zone) during intrazone search, fewer zones need to be searched before a destination is found.

Two search algorithms are designed in our model: partial search/2, full search/4. Partial search/2: a node starts a location search procedure by transmitting two LR packets to two periphery zones on two opposite directions, e.g. zone 1 and zone 36, if destination is not found, source node continues to search two other periphery zones on opposite directions and so on. Zones selected to search are some of periphery zones: zone 1, 36; 6, 31; 3, 34; 18, 19; 4, 33; 24, 13. If destination is not found after all above zones are searched, source node will start over to re-search these zones.

Full search/4: similar as partial search/2 except that the source node search 4 zones on four directions simultaneously, zones selected to be search are all periphery zones.

Performance of simulation result under two different search algorithms is list in table 4.1, performance of ZHLS is also list in the same table.

	Full search/4	partial search/2	ZHLS
Overhead (packets/packets)	0.27	0.27	0.263
Overhead (transmissions/transmissions)	0.217	0.23	0.753
Network efficiency(packets/packets)	0.700	0.666	0.118
Average transfer delay (iterations)	2037	1943	5455
Standard deviation of transfer delay (iterations)	4812.6	5005.4	9113.8
Average location search delay (iterations)	7196	7139	17148
Standard deviation of location search delay(iterations)	8163.3	8165.6	13053.5
Average number of hops of a typical DU	4.86	4.75	3.78
Standard deviation of hops of data packets	2.66	2.67	2.03
Average number of packets in link buffer	0.1637	0.1604	0.1543
Standard deviation of packets in link buffer	0.14	0.136	0.131
Average overflow probability of link buffer	0.00125	0.00125	0.00125
Standard deviation of overflow probability of link buffer	0.002342	0.002342	0.002342
Average number of packets in sending buffer	0.3546	0.1985	0.1068
Standard deviation of packets in sending buffer	0.8	0.485	0.2961
Average overflow probability of sending buffer	0.0001790	0.0000346	0.0000047
Standard deviation of overflow probability of sending buffer	0.001294	0.000336	0.000287
Average number of packets in input buffer	7.87	7.97	16.95
Standard deviation of packets in input buffer	6.112	6.051	3.123
Average overflow probability of input buffer	0.2473	0.2823	0.7802
Standard deviation of overflow probability of input buffer	0.245	0.266	0.189

Table 4.1: network performance under different search algorithms

Simulation runs 50002 iterations under following conditions: maximum speed of nodes: 30 km/h; data packets input rate/iteration : 1 packet/iteration; period of flooding NLSP: 10000 iteration; period of flooding ZLSP: 25000 iteration; link time: 250 iteration; timeout of data packets: 1200 iteration; hop limit of data packets: 20. Link time: 250 iteration.

Comparing the two search algorithm, we find that they have very similar performance except that the number of packets in sending buffer and overflow probability of sending buffer of full search/4 algorithm is higher than those of partial search/2 algorithm. This is caused by more LR packets sent simultaneously. Sending more LR packet does not reduce the transfer delay and location search delay because it increases network traffic and incur more packet loss. It is interesting to observe that network performance does not degrade obviously even if location search does not include all periphery zones. This phenomenon demonstrates the positive effect of our modification: extending intrazone routing to all neighbor nodes of local zone makes a single search covers more area without putting extra overheads to network.

Comparing performance of MZHLS and ZHLS, some improvements are obviously seen in table 4.1. Network efficiency increases significantly while overhead (transmissions/transmissions) and transfer delay are minimized.

The location search delay is approximately estimate as follows: simulation of table 4-1 runs under the condition that timeout of data packets (including LR packets) is 1200 iterations. In case of ZHLS, average number of zones need to be search before a destination node is found is (36-1)/2 =17.5 zones, and the average location search delay is 17.5x1200 =21000 iterations. Destination node may locate on path to a searched zone; it will reduce the delay time. On the other hand, packet loss may increase the delay time. The simulation result in table 4-1 is 17148 iterations. As stated in chapter 2, ZHLS is reactive if the destination locates in a different zone. Normally the disadvantage of

reactive routing protocols is long delay for route discovery. Table 4.1 shows that ZHLS suffers from a delay as long as 17148 iterations in route discovery. Such a long location search delay causes problems as low network efficiency, long transfer delay, and input buffer overflow.

Two efforts to minimize location search delay have been done in this model as stated in chapter 2, extending the intrazone search to all local zone nodes and all neighbor nodes, using different search algorithm. Location search time for partial search/2 in case of MZHLS is also estimated as: average number of zones to be searched before a destination node is found is: 12/2 =6, average location search delay (search two zones each time) is 1200x6/2 = 3600 iterations. Due to packet loss and queuing delay (sending two packets may needs longer queuing time than sending one), actually location search delay may be longer. The simulation result of location search delay is 7139 iterations. Compared with ZHLS, this delay is much shorter. So network efficiency is significantly increased, while overhead (transmissions/transmissions), transfer delay and input buffer overflow are minimized. But the location search delay is still long as a cost of reactive scheme.

As a conclusion, improvement to ZHLS is made by making full use of node level link information; extended intrazone routing enables more efficient location search algorithm in MZHLS routing.

4.9 Performance Comparison of MZHLS with Other Routing Protocols

Figure 2.1 to 2.5 in section 2.1.1 show Iwata's numerical simulation results of 5 different routing techniques. In this section Iwata's results will be used to compare with the performance of MZHLS.

Due to the difference of simulation model, simulation conditions of each model should be elaborated before any fair comparisons could be made. Also some parameters conversions are necessary. Even though, the comparison is approximate.

In MZHLS model, the transmission rate of the channel is assumed to be 1Mbps while in Iwata's model it is 2 Mbps, this may lead to at least 50% performance degradation. In MZHLS, all kinds of packets have the same size of length of 6000 channel symbols so that the iteration time is 6 ms. In Iwata's model, packets have different size, 10 kbits for data packets, 2 kbits and 500 bits for different control packets. Smaller control packet size leads to shorter iteration time, thus benefits of shorter delay time and higher network efficiency. The drawback of different packet size is longer processing time and longer latency.

In Iwata's model, each source/destination pair generates 4Kbps input traffic, and 500 pairs generate an amount of input traffic equivalent to an input traffic rate of 1 packet/iteration in MZHLS model (which is used as normal test condition in MZHLS model). Most of Iwata's simulation results are done with the input traffic of 100 pairs.

To simplify the simulation in MZHLS, active nodes are apart from each at least twice transmission radius in both horizontal and vertical. This causes 50% less active nodes than maximum possible number in each iteration and thus 50% loss in channel capacity.

Fig. 4.49, shows that MZHLS has a very good overhead performance. At mobility 60 km/hr, overhead (transmission/transmission) is 0.33. Under input traffic of 1 packet/iteration, overhead bit rate is equal to 0.33 Mbps and it is half of that of data transmission. If the control packet size is decrease to 1/5, then the overhead bit rate is 0.06Mbps. It is better than the results of all the 5 techniques shown in fig. 2.1. Similarly, at mobility 30 km/hr, MZHLS also exhibits better overhead performance than the five routing techniques show in fig. 2.2.

Fig. 4.50 shows that average transfer delay of MZHLS is 1900 iterations at 30 km/hr and 3000 iterations at 60 km/hr, which is 11.2 seconds and 18 seconds, respectively. Considering 1/5 length of control packet size, 2 Mbps transmission rate, twice more active nodes in each iteration and 1/5 input traffic rate, the average transfer delay of

MZHLS are equivalent to or better than 0.56 seconds and 0.9 seconds, respectively. From fig. 2.3, average transfer delay of MZHLS is better than on-demand routing and worse than the other three techniques. (Because each node needs to initialize a location search procedure at the beginning of simulation, it causes longer transfer delay. MZHLS transfer delay is expected to be better after the network reaches a stable state).

Under conditions of 1 packet/iteration input traffic rate in fig. 4.12 and mobility of 30km/hr, 60km/hr in fig. 4.52, the average hop count of a typical packet of MZHLS is around 4.5, which is as good as FSR and DSDV in fig. 2.4. MZHLS exhibits a good path length performance and its path length is constant in most cases since it always traces the shortest path.

From the comparison above, a conclusion can be made that MZHLS has a very good overall performance among the five different techniques.

Conclusions

In this thesis, a modified zone-based hierarchical link state routing (MZHLS) is suggested for wireless ad-hoc networks. A peer-to-peer zone-based two-level link state routing (ZHLS), which is proactive at node level and reactive when a destination exists in different zones, has significant delay for route discovery as other reactive routings especially in large and dynamic network. To minimize the delay of route discovery in ZHLS, two important modifications have been done in this thesis: Extending the range of intrazone search and using more efficient location search algorithm.

The thesis emphasizes on performance evaluation and analyses of MZHLS. A simulation model in C was set up to study which input parameters have important effects on network performance and how network performance varies under different condition or different input parameter settings. Simulation results has confirmed that MZHLS reduces the location search delay and improves the network performance in some extent versus ZHLS. Comparison with other typical routing techniques also demonstrates that MZHLS has a very good overall performance.

Simulation results have also shown that network need to be optimized before best network performance can be achieved. Various input parameter settings affect each other and thus have significant and complicated effect on network performance.

As the network changes constantly, how to dynamically configure the network is still a challenge for future research. A good routing protocol is difficult to find because it not only performs well under certain network condition but also under most possible network conditions. It should be able to modify the network configuration quickly and thus to work well on changing network.

In this works, a batch of performance criteria are defined for evaluation of routing performance. Some of them are referred to the similar performance criteria defined for

the wired network; others are defined specifically for MZHLS. A set of performance criteria should be defined systematically to best evaluate a wireless ad-hoc routing protocol. It leads to another topic for future research.

Throughout this work, the routing processing time is not touched. Simulation shows that it takes about $8x10^{-4}$ second for a Pentium 100M processor to do the shortest path algorithm for a single node at both node level and zone level. Compared with the $6x10^{-3}$ second transmission time of a whole packet, the processing delay is not negligible in real communication. Specifically in the MZHLS model, the shortest path algorithm is simply based on minimum hops. If a combination factor of distance, cost and reliability is considered, the processing time for shortest path algorithm is expected to be much longer. This implies that powerful algorithms may be achieved at the cost of degrading some other routing performances, e.g. longer latency.

On the other hand, the whole routing processing time for different network model varies on different types of processors. Faster processors require shorter processing time to do the same algorithm. Degrading of performance due to long processing delay may be alleviate by more powerful processors. How the routing processing time on different processors affects the routing performance is again a topic for future research.

References

[1]Wireless medium access control and physical layer WG, IEEE draft standard P802.11, "wireless LAN," IEEE Stds. Dept, D3, Jan. 1996

[2]C.E. Perkins and P.Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers", in Proc. ACM SIGCOMM, vol. 24, no. 4, pp. 234-244, Oct. 1994

[3]Tsu-Wei Chen and Mario Gerla, "Global state routing: a new routing scheme for ad-hoc wirless networks" Proc. IEEE ICC'98,
http://www.ics.uci.edu/~atm/adhoc/paper-collection/gerla-gsr-icc98.pdf

[4]P.Hacquet, P. Muhlethaler, and A.Qayyum, "Optimized link state routing protocol", IETF MANET, Internet Draft, Nov. 1998

[5]S. Murthy and J.J. Garcia-Luna-Aceves, "An efficient routing protocol for wireless communication networks, ACM Mobile networks and App. J., Special issue on routing in mobile communication networks, Oct. 1996, pp. 183-97
http://www.ics.uci.edu/~atm/adhoc/paper-collection/aceves-routing-winet.pdf

[6]V.D. Park and MS Corson "A highly adaptive distributed routing algorithm for mobile wireless networks", Proc. IEEE INFORCOM'97, Kobe, Japan, pp. 1405-1413.
http://www.ietf.org/internet-drafts/draft-ietf-manet-dsr-03.txt

[7]David B. Johnson, Davix A.Maltz, "Dynamic source routing in ad hoc networks", Mobile Computing, T.Imielinski and H.Korth, Eds., Kulwer, 1996, pp.152-81
http://www.ics.uci.edu/~atm/adhoc/paper-collection/johnson-dsr.pdf

[8]Charle E. Perkins, Elizabeth M. Royer, Samir R.Das, "Ad hoc on-demand distance vector routing", Proc. IEEE WMCSA'99, vol.3, New Orleans, LA, pp. 90-100
http://www.ietf.org/internet-drafts/draft-ietf-manet-aodv-04.txt

[9]C.-K. Toh, "Long-lived ad-hoc routing based on the concept of associativity" March 1999 IETF Draft.

http:/ www.ietf.org internet-drafts draft-ietf-manet-longlived-adhoc-routing-00.tx

[10]Z.J. Haas, "The zone routing protocol (ZRP) for ad hoc networks," Internet Draft, Nov. 1997

[11]M.R. Pearlman and Z.J. Haas, "Determining the optimal configuration for the zone routing protocol", IEEE journal on selected areas in communications, Aug. 1999, pp.1395-1414

[12]M Joa-Ng and I.-T. Lu, "A peer-to-peer zone-based two-level link state routing for mobile Ad-Hoc Networks," IEEE Journal on selected areas in communications, Vol.17, No. 8, Aug. 1999, pp.1415-1425

[13]R. Nee and G.Awater, "New high-rate wireless LAN standards", IEEE Communication magazine, Dec. 1999

[14]B. Crow, I. Widjaja, J. Kim and P. Sakai, "IEEE 802.11 wireless local area networks", IEEE communications magazine, Sept. 1997

[15]Benny Bing, "measured performance of the IEEE 802.11 wireless LAN", 1999 IEEE

[16]D. Bertsekas and R. Gallager, Data Networks, 2nd ed. Englewood, NJ: Prentice-Hall, 1992

[17]Hedrick C 1998 Routing Information Protocol, RFC 1058, June

[18]Iwata A. Chiang C.C., Pei G., Gerla M., Chan T.W. 1999 "Scalable routing strategies for ad-hoc wireless networks", IEEE journal on selected areas in Comm., Vol.17, No.8, Aug, pp.158-162

[19]Elhakeem A.K, Ali S.M., Aquil F., Li Z., and Zeidi. S.R.A. 2000 "New forwarding Data basis and ad-hoc routing techniques for nested clusters of wireless LANs", submitted to the wireless Comm. Journal, Kluwer Publisher, Nov.

[20]Kleinrock L. and Stevens K. 1971 "Fisheye: L lenslike Computer Display Transformation", Computer Sci. Dept, University of Californis, Los Angeles, CA, Tech. Rep.

[21]G.S. Lauer, "Packet-radio touting," Routing in Communications Networks, M.E. Steenstrup, Ed. Englewood Cliffs, NJ: Prentice-Hall, 1995, pp.375-379.

[22]Chiang CC., Wu H-K, Liu W., and Gerla M. 1997 "Routing in clustered multihop, mobile, wireless networks", Proc. IEEE Singapore Int. Conf. Networks, pp.197-211

[23]Gerla M. and Tsai J 1995 "Multiuser, mobile, multimedia radio network", ACM Baltzer J. Wireless networks, vol. 1, no. 3, pp.255-265.

[24]Assad. S, 1998 http://www.ctr.columbis.edu

[25]Maltz D.A, Broch J., Jetcheva J. and Johnson D. 1999 "The efforts of On-demand behavior in routing protocols for multihop wireless ad-hoc networks" IEEE journal on selected areas in Comm. Vol.17, No.8, 1999, pp.1439-1453

[26]Edward C. Prem, "Wireless local area networks", ftp://ftp.netlav.ohio-state.edu/pub/jain/wirelesslans/index.htm