QoS-BASED MULTIMEDIA SERVICES ROUTING IN MOBILE AD-HOC NETWORKS

TAING NGUON

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENT FOR THE DEGREE OF

MASTER OF ENGINEERING IN COMPUTER ENGINEERING

SCHOOL OF GRADUATE STUDIES

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2006

ISBN 974-15-2337-8

หัวข้อวิทยานิพนธ์	**การค้นหาเส้นทางเพื่อการรับประกันคุณภาพบริการของมัลติมีเดียในเครือข่ายเคลื่อนที่แบบแอดฮอค**
นักศึกษา	TAING Nguon
รหัสนักศึกษา	47060836
ปริญญา	วิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชา	วิศวกรรมคอมพิวเตอร์
พ.ศ.	2006
อาจารย์ผู้ควบคุมวิทยานิพนธ์	ผศ.ดร.ศักดิ์ชัย ทิพย์จักษุรัตน์

บทคัดย่อ

ปัจจุบันอุปกรณ์ไร้สายต่างๆ เช่น คอมพิวเตอร์แบบพกพา ได้มีการพัฒนาให้มีประสิทธิภาพที่สูงขึ้นประกอบกับอุปกรณ์ไร้สายต่างๆ เหล่านี้ เริ่มเข้ามามีบทบาทอย่างมากในชีวิตประจำวันในปัจจุบัน ในวิทยานิพนธ์ฉบับนี้ เราจะพิจารณาเครือข่ายเคลื่อนที่แบบแอดฮอค การติดต่อสื่อสารของโมไบล์โหนดต่างๆ ในเครือข่ายเคลื่อนที่แบบแอดฮอค จะติดต่อสื่อสารระหว่างกันผ่านทางสัญญาณวิทยุ โดยปราศจากสื่อนำสัญญาณประเภทสาย สถานีฐาน หรือศูนย์กลางการควบคุมการสื่อสาร คุณลักษณะที่สำคัญของโมไบล์โหนดที่อยู่ในเครือข่ายคือ โมไบล์โหนดจะสามารถเคลื่อนที่ได้อย่างอิสระ จึงทำให้โครงสร้างของเครือข่ายที่เชื่อมต่ออยู่เกิดการเปลี่ยนแปลงโครงสร้างค่อนข้างบ่อย ดังนั้นเส้นทางการสื่อสารที่ถูกกำหนดขึ้นอาจจะเกิดความเสียหาย ทำให้ไม่สามารถส่งผ่านไปยังโมไบล์โหนดปลายทางได้ ก่อให้เกิดความล่าช้าของการส่งข้อมูล ซึ่งสิ่งนี้มักจะมีผลกระทบค่อนข้างมากสำหรับการส่งข้อมูลประเภทมัลติมีเดีย

ดังนั้นในงานวิจัยนี้ เราได้นำเสนอการพิจารณาเส้นทางที่สามารถให้การประกันคุณภาพของการบริการสำหรับแอพพลิเคชันประเภทมัลติมีเดียหรือแอพพลิเคชันอื่นๆ ที่มีผลกระทบจากความล้าช้าของการส่งข้อมูล โดยการพิจารณาเส้นทางที่สั้นที่สุดที่มีแบนด์วิธที่เพียงพอต่อความต้องการของการส่งข้อมูลมัลติมีเดีย เราเรียกโปรโตคอลที่เรานำเสนอว่า เอ็มดีเอสอาร์ โดยการประยุกต์ใช้ความแรงของการส่งสัญญาณวิทยุร่วมกับกระบวนการสำรองแบนด์วิธ

เราได้ทำการวัดประสิทธิภาพของโปรโตคอล เอ็มดีเอสอาร์ ด้วยการจำลองการทำงานของระบบเพื่อเปรียบเทียบกับประสิทธิภาพของโปรโตคอลดีเอสอาร์ จากผลการทดลองเราพบว่าโปรโตคอลเอ็มดีเอสอาร์ สามารถลดจำนวนฮอพระหว่างโมไบล์โหนดต้นทางและโมไบล์โหนดปลายทางลดได้ นอกจากนี้ยังสามารถลดความล่าช้าของการส่งข้อมูลลงได้ ซึ่งทำให้ได้ค่าทรูพุทที่สูงขึ้น ดังนั้นเราจะสามารถสรุปได้ว่าโปรโตคอลเอ็มดีเอสอาร์สามารถให้การประกันคุณภาพบริการของการส่งข้อมูลมัลติมีเดียได้

Thesis Title	QoS-Based Multimedia Services Routing in Mobile Ad-Hoc Networks
Student	TAING Nguon
Student ID	47060836
Degree	Master of Engineering
Programme	Computer Engineering
Year	2006
Thesis Advisor	Assist. Prof. Dr. Sakchai Thipchaksurat

ABSTRACT

Wireless devices such as portable or handheld computers have been developing and getting more powerful in their capabilities. These devices called mobile node are now playing important role in our daily life. These products communicate with each other through a shared radio link without cable, backbone, based station or centralized administration which commonly called Mobile Ad Hoc Network (MANET). Every device in this network scenario can be moved freely and randomly that causes network topology change constantly, so available routes may fail at any times then packets being sent are delay on any intermediate nodes. Therefore it might not satisfy some applications such as multimedia application because multimedia stream is delay-sensitive. In this research, we propose Qos-Based Multimedia Service Routing that Modifies Dynamic Source Routing (MDSR) protocol. MDSR protocol selects the route for transmitting packets by considering the high signal strength and bandwidth reservation mechanism in order to avoid collision in the network.

We evaluate performance of MDSR protocol comparing with DSR protocol by means of the simulation. The simulation results show that MDSR can provide the smaller number of hop along the part from source mobile node to destination mobile node, lower delay and also higher throughput comparing with DSR protocol. So, we can conclude that MDSR protocol can provide the QoS for multimedia services.

Acknowledgements

This research has been supported and helped from professors. At first, I would like to thank Assist. Prof. Dr. Sakchai THIPCHAKSURAT, thesis advisor for his support, suggestion and help during my study as graduate student at King Mognkut's Institute of Techonology Ladkrabang. I also would like to thank Assoc. Prof. Dr. Ruttikorn VARAKULSIRIPUNTH, director of Computer Networks Laboratory of Research Center for Communications and Information Technology (ReCCIT), KMITL, for his comment on my doing research as well as my thesis. I also would like to thank Prof. Hiroshi ISHII for his valuable time to comment and suggest me. He is currently a professor of the Department of Communication Engineering, School of Information Technology and Electronics, Tokai University, Japan.

I would like to thank (1) AUN/SEED-Net and JICA for financial supports, (2) The department of Computer Engineering, Faculty of Engineering, King Mongkut's Institute of Technology for providing me a good education (3) Institute of Technology Cambodia for giving me the best opportunities to pursue my master degree.

Finally, I deeply thank my parents, Taing Siek Hong and Ly Chhun Lang for giving me life, for their love, and support for my study.

TAING Nguon

CONTENTS

CONTENTS (CONTINUE)

List of Tables

List of Figures

List of Figures (Continue)

Figure Page

Chapter 1

Introduction

1.1 Literature Review

Mobile Ad Hoc Network (MANET) is a collection of wireless mobile nodes in which each node communicates with each other through shared, limited radio channel in peer to peer fashion that makes MANET to be an infrastructure-less networking as shown in an example in Figure 1. Each mobile node or intermediate node in the network operates as routers forwarding packet to other nodes in order to establish two pairs of communication nodes.

MANET can be categorized into flat architecture and hierarchical architecture [3, 6], called cluster. In flat architecture, all nodes have the same rank, but in hierarchical architecture, nodes are separated into masters and slaves. Master nodes, cluster heads are selected among the nodes in the network and the remaining nodes, slave nodes, register themselves in one cluster of more cluster heads. A cluster head is responsible for communicating with its registered cluster nodes and communicating with other cluster heads in the network. By contrast, slave nodes in each cluster communicate with each other via their respective master nodes.

Because MANET does not rely on wired backbone, based station or centralized administration, the growing demands of roaming users, and the availability of wireless devices, MANET has been received an increased attention for commercial purpose especially for small area wireless networks (home office, building, organization conferences, etc.). Moreover, military communication and disaster recovery operation in the area where there is no based station or access point, MANET is useful in those environments.

In MANET, the nodes are mobile hosts and the topology of the network may change rapidly and unexpectedly. Therefore, the packets transferring from end-to-end may be lost as the result of broken link of the availability routes because of deletions of nodes, contention or interfering. Quality of Services (QoS) for MANET have been done by many researchers [1, 2, 5], such as QoS starting from physical layer up to application layer [4], QoS Medium Access Control (MAC), Genetic Algorithm QoS Routing Method [9], and QoS of Routing [7, 8, 10, 11].

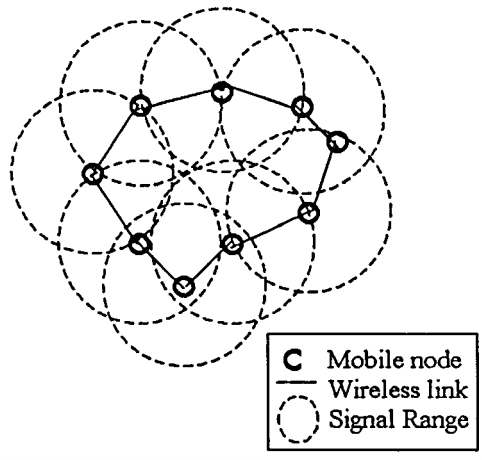


Figure1.1: An Example of Mobile Ad Hoc Network

Due to the limited and shared resources in MANET, Several routing protocols have been developed or proposed in order to adapt to the nature changes of wireless network such as nodes are mobile, routes change frequently, and the topology is unpredictable. These routing protocols [10] can be classified into three different groups: global/proactive, on-demand/ reactive and hybrid. Proactive routing protocols find routes in the own maintaining route table in each node while the reactive routing protocols find route only one node need to send data to the other node. Hybrid routing protocols combine both proactive and reactive routing protocols together. That are; proactive is used in a cluster called zone whereas reactive is used to find nodes outside. This kind of routing protocol is used in hierarchical architecture.

1.2 Statement of problem

Multimedia application such as video conference, tv on web, radio on web or voice over ip, is important for our daily life. It is delay sensitive. In order to guarantee Multimedia traffic in MANET where nodes can be moved randomly, Route used to transmit this kind of traffic, must be able to reduce delay of packet on any intermediate nodes. Some researchers have done research on find route with limited route request packets, which consumes less bandwidth. In this research, we consider the shortest bandwidth path for carrying multimedia traffic. In this shortest bandwidth path scheme, we not only search for the shortest path, but also provide bandwidth mechanism or reservation for nodes along the path from source node to destination node. Therefore the route found must the shortest bandwidth path for transmit multimedia traffic.

1.3 Existing Proposed Solutions

Ref [14], authors have proposed different power level to support higher priority traffic such as multimedia traffic, which is delay-sensitive, and lower priority traffic such non real-time traffic, which delay is not sensitive. In order to reduce delay for multimedia traffic, the authors of Ref.[14] have proposed (1) higher power level for a node that is to transmit multimedia packets or other words its coverage area can be increased so that the number of nodes inside its coverage area is also increased. Therefore it can select a node, which is the farthest one to forward route request packets. As result, the shortest path can be found for multimedia traffic and delay can also be decreased. On the other hand, the authors also proposed (2) lower power level for non real-time traffic such as text application so a node that carries this kind of packet has smaller range. In other word, the number of nodes in its coverage area is decreased. As result, longer path can be found for this kind of traffic and It can save energy, which is the main power for mobile node.

1.4 Thesis Organization

The thesis is organized as bellows:

Chapter 2 explains about existing routing protocols have been done by researchers such table-driven routing protocol, Even-driven routing protocols and hybrid- routing protocols, which combines table-driven and even-driven protocols.

Chapter 3, we will explain in detail our proposed scheme, called Quality of Services Based Multimedia Services Routing in Mobile Ad Hoc Network. Our proposed mechanism will search for a shorter bandwidth route to guarantee multimedia traffic. This shorter bandwidth route is not the shortest path, but nodes along this shorter bandwidth route have available bandwidth to transmit multimedia packets.

Chapter 4, we present our implementation or our proposed scheme by using freeware network simulator called ns2, which is currently developed under supporting through DARPA with SAMAN and through NSF with CONSER, both in collaboration with other researchers including ACIRI [15].

Finally, Conclusions and future researches are given in chapter 5.

Chapter 2

Routing Protocols in Mobile Ad Hoc Network

Due to the characteristic of Mobile Ad Hoc Network as mentioned in Section I, routing protocols have been a rapid growth of research interests among researchers around the globe. There have been a large number of routing protocols designed for multi-hop mobile ad hoc network. The main development goals of these routing protocols are illustrated as below:

- Minimal Overhead: Reduce control overhead as little as possible because it consumes bandwidth, which is the main medium in wireless network. While transmitting, a node consumes much more power than receiving. Anyway both transmitting and receiving processes are power consumers. Therefore they will decrease the life time of battery, which is the only source power of a mobile node.

- Dynamic Topology: This kind of topology does not exist in Wired Local Area Network. Once a route has been established, it is commonly broken as result of node mobility. In order to have a smooth end-to-end pair of communication between source and destination, a viable routing path must be maintained. When the link is broken, an action must be done immediately with a minimum of overhead or Secondary path can be used in stead of the primary broken link.

- Multi-hop routing capability: As communication between end-to-end pair through a radio channel, the wireless range is limited so source and destination are not adjacent nodes. Hence routing protocol must be able to search for multi-hop route from source to destination.

- Loop prevention: Routing loop occurs when data and control packets traverse the path multiple times until either path is fixed, and loop is terminated or time to live (TTL) of the packet becomes zero. Routing loops are wasteful of resources such as bandwidth and power consumption and they also degrade the network performance. Therefore loop should be avoided.

- Power limitation: Battery is the main power source for mobile node so that the life of battery is limited. Any solutions that reduce power consumption will often be favored. The much more use of battery power, the shortest life time of a mobile node is. If battery power is waste consumed, the node will fail quickly then it will affect the network availability and functionality.

A number of routing protocols have been developed or proposed in order to adapt to the nature changes of wireless network such as nodes are mobile hosts, routes change frequently, and the topology is unpredictable. These protocols can be classified into three categories [10]. (1) Proactive routing protocol, all routes to all destinations are determined at start up and these routes are maintained by periodical route update packet. (2) Reactive routing protocol, routes are established when they are required by a source node using route discovery process. (3) Hybrid routing protocol, this protocol combines proactive and reactive routing protocol together where proactive routing protocol is used in a cluster and reactive routing protocol is used to discover route between clusters. Normally Hybrid routing protocol is used in hierarchical network structure and the first two routings protocols are used in flat network structure.

2.1 Table driven routing protocols (Proactive)

Proactive routing protocols maintain information in a table on each node about the routing to the other node in the network. Although the topology of the network does not changed, this information must be updated periodically. Many proactive routing protocols have been proposed, for example, Destination Sequence Distance Vector (DSDV), Wireless Routing Protocol (WRP), Global State Routing (GSR), and Optimized Linked State Routing (OLSR) and so on. These protocols have some differences in term of maintaining updating routing information in the table [10]. There are pros and cons in the proactive routing protocol. The advantage is that when a node needs to send or forward data packets, route can be immediately used and the disadvantage is that this protocol allows significantly overhead and consumes bandwidth in the network. Further, the amount of routing state maintained at each node scales as O(n), where n is the number of nodes in the network

2.1.1 Destination-Sequence Distance Vector Routing (DSDV)

Because DSDV [16] is a proactive routing protocol, each node stores a routing table, which maintains routes to all destinations in the network. The routing table contains the following information for each entry: destination IP address, destination sequence number, next-hop IP address, hop count and install time. DSDV uses both periodic and event-triggered routing table updates. Every time interval, each node broadcasts to its neighbors its current

sequence number, along with any routing table updates. The routing table contains the following fields:

< destination IP address, destination sequence number, hop count >

When a node receives an update message, the node utilizes the information on that update message to compute their routing table entries by comparing the new information with the old information already available in its routing table. Any route with a more recent sequence (greatest) number is used whereas routes with old sequence numbers are discarded. This ensures the utilization of the most recent routing information to the destination. A route with sequence number equal to an existing route is chosen only if it has smaller or better metric and the existing route will be discarded or stored as a less preferable route. In addition to periodical updates, DSDV also utilizes event-trigger updates to announce important link changes, such as link removal. This event-triggered updates ensure timely discovery of routing path changes.

In order to improve performance in mobile networks when it is bigger, DSDV implements two optimizations, full dump and incremental. Full dump updates are transmissions of a node's entire routing table. Because these update size scales of the network size, Full dump updates are performed relatively infrequently. Incremental updates are transmitted more frequently in order to reduce overhead and bandwidth consumption. These incremental updates carry only information changed since the last full dump updates. When movement becomes frequent and the size of an incremental update approaches the size of a Network Protocol Data Unit (NPDU), then a full dump update can be schedule so that the size of the next incremental update will be smaller.

As mentioned above how to select a better route by comparing the new information in the update message and the old information available in its routing table, Fluctuation problem may arise by the following criteria of route updates:

- Routes are always preferred if the sequence numbers are newer, routes with older sequence numbers are discarded.
- A route with a sequence number equal to that of an existing route is preferred if it has a better metric, and the existing route is discarded or stored as less preferable.

When mobile nodes independently transmitting update messages and having markedly different transmission intervals, it may turn out that a particular mobile node receives new update packets in a way that causes this mobile node to consistently change route back and forth

between different next hops, even though no network topology change has taken place. This fluctuation happens because of the above two route selection criteria. Conceivably, a mobile node can always receive two routes with equal sequence numbers or with a newer sequence number one after the other via different neighbors to the same destination, but the mobile node always gets the route with the worse metric first. This situation leads to the fluctuation with a continuing burst of new update packets.

Finally, to solve the problem of fluctuation problem, DSDV requires that a mobile node receives a new update message, the route with the later sequence number must be available for use for the mobile node forwarding decisions, but it does not have to announce the update immediately unless the route is to a previously unreachable destination, that the mobile node has to wait for ***settling time*** before advertising the update message a better metric for a destination. The ***settling time*** is the average time to get all the updated advertisement for a route. In this way, the node can be sure to receive all routing path change for a destination before propagating any of those changes. As result, bandwidth utilization and power consumption can be reduced by neighboring nodes.

2.1.2 Optimized Link State Routing (OLSR)

The Optimized Link State Routing (OLSR) protocol [17] is a proactive protocol and it inherits the stability of the link state algorithm. OLSR is an optimization of of a pure link state protocol for mobile ad hoc networks. The key feature of OLSR is the use of ***multipoint relays*** (MPRs) to reduce the overhead of the network flood and the size of the link control packet. MPRs of a node is a set of its neighborhood; for example in Figure 2.1, {A, B, C, D} is MPRs of node S. Each node in this MPRs can retransmits any broadcast control packet sent from node S, whereas the neighbor nodes ({M.N} in Figure 2.1) are not in this MPRs can read and process the packets but not to retransmit the broadcast control packet from node S. In other word, MPRs of node can be called MPRs selectors.

Each node selects its multipoint relay set among it one hop neighbors in such manner that the set covers all the nodes that are two hops away. The smaller is the multipoint relay set, the more optional is the routing protocol as shown in Figure 2.1. Nodes learn their set of two-hop neighbors through the periodic exchange of Hello messages. Each node periodically transmits a Hello message that contains a list of all neighbors. Associated with each neighbor is an attribute indicating the directionally of the link of that neighbor. The node is label *symmetric* if the link to

the neighbor is bidirectional, or *asymmetric* if a Hello has been received from that node but the link has not been confirmed as bidirectional. When a node receives this hello message from each of its neighbors, it obtains complete knowledge of its two-hop neighbor set at that point in time. Further, if its own address is listed in the Hello message, it knows the link with that neighbor is bidirectional. Then it can update of that neighbor to be symmetric.

Once each node's MPR set is selected, routing path within the network can be determined. In order to adapt to the topology change in the network, nodes periodically exchange Topology Control (TC) message with their neighbors. The TC message for a given node lists the set of neighbors that have selected the sending node as an MPR. Only this set of nodes (MPR set) is advertised within the network. As a node receives TC message from the other network nodes, it can create or modify routing entries to each node in the network using any shortest path routing algorithm, such as a variation of Dijkstra's algorithm.

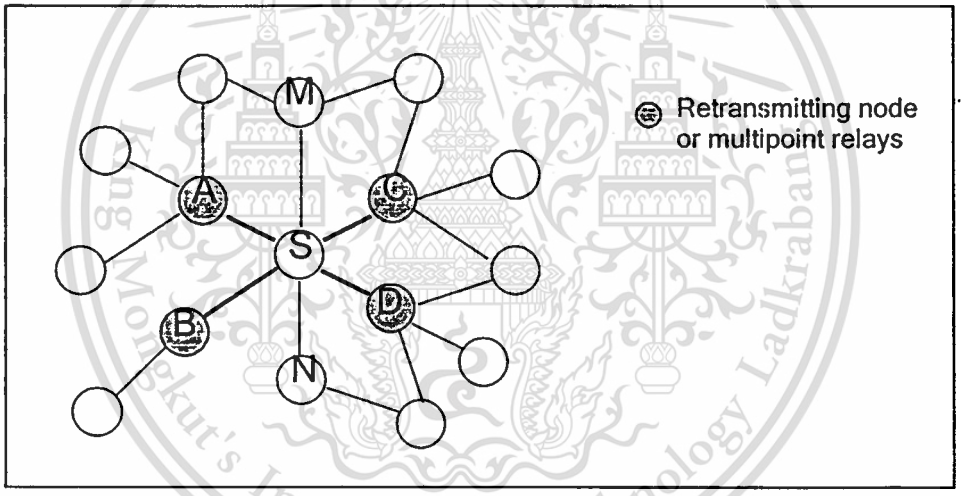


Figure 2.1: Multipoint relays

2.2 On-demand driven routing protocols (Reactive)

Unlike proactive routing protocols, the reactive routing protocols create routes once a node wants to transmit data to a destination. The source node initiates route discovery process by flooding route query with in the network. When the destination is reached, route reply request will be sent back to the source. Once the route has been found, it is maintained until either destination becomes inaccessible or the route is no longer desired then route discovery process will be invoked again. Several reactive protocols have been proposed such as Dynamic Source Routing protocol (DSR), Ad hoc On-demand Distance Vector (AODV), Light-weight Mobile Routing (LMR), Temporary Ordered Routing Algorithm (TORA), Relative Distance Micro-

discovery Ad hoc Routing (RDMAR) [10] and so on. Those protocols can be classified into two groups, source routing and hop-by-hop routing [10]. The advantage of reactive routing protocols is to overcome the overheads, occurred in proactive routing protocols.

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

The Ad Hoc On-Demand Distance Vector Routing [2] is one reactive routing protocol and it is similar to Dynamic Source Routing (Chapter 3) that finds routes based on a route discovery process involving a broadcast network search and a unicast reply containing discovered paths. Similar to DSDV, AODV relies on per-node sequence numbers for loop freedom and for ensuring selection of the most recent routing path. AODV nodes maintain a route table in which next-hop routing information for destination nodes is stored. Every route in the route table has lifetime value. Within the lifetime value, any route is not used then it is expired. Therefore the algorithm used in AODV is based on DSDV and DSR algorithm.

When a node has packets to send to some destination, it does the following steps:

Step 1) Checks in its route table for a route to desired destination node. If any route exists, it uses that route for data packets transmission.

Step 2) Otherwise, the node must invoke Route Discovery Process to find a new route. In order to do so, route request (RREQ) packet must be created and contains information such as the source node's IP address and current sequence number, hop count (initialized to 0), destination node's IP address, the last known sequence number for that destination, and a unique RREQ ID. Or in other word RREQ contains the following fields:

< source addr; source sequence #; broadcast id; dest addr; dest sequence #; hop cnt >

Step 3) Finally, broadcasts RREQ to its neighbors.

When any node (intermediate or destination node) receives route request (RREQ) packets, it does the below step:

Step 1) It generate route reply only if this node is the destination of this RREQ or this node has an unexpired route to the destination node in its route table, which means that the sequence number of that route must be at least as great as the sequence number in the route request.

Step 2) RREQ is already seen or received (the same broadcast id and source address), it drops the redundant RREQ and does not rebroadcast it. If a node cannot satisfy the RREQ, it keeps track of the following information (Destination IP address, Source IP address, Broadcast id, Expiration time for reverse path route entry, Source node's sequence number) in order to implement the *reverse path setup*, as well as the *forward path setup* that will accompany the transmission of the eventual RRPL. Figure 2.2(a) illustrates this flooding procedure.

Step 3) Finally, re-broadcast route request packet.

When a node receives the RRPL, it first creates a *forward route entry* for the destination node. It uses the node from which it received the RRPL as the next hop toward the destination. The hop count for that route is the hop count in the RRPL, incremented by one. This *forward route entry* is used if the source node selects this path for data packet transmissions to the destination. The route reply (RRPL) is forwarded hop by hop back to the source as illustrated in Figure 2.2 (b). If the source node receives the RRPL more than one, then it selects the route with the greatest sequence number and smallest hop count to transmit data packets.

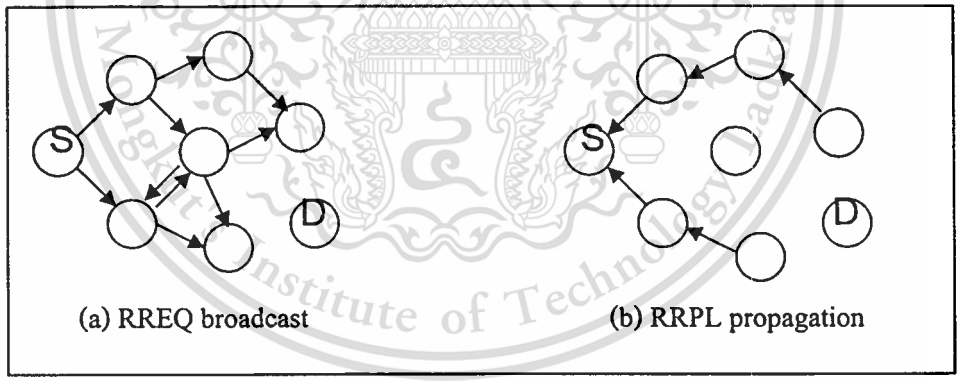


Figure 2.2: AODV route discovery

Another mechanism of AODV is route maintenance for an established connection of pair nodes. A route that has been recently used for the transmission of data packets is called *active route*. Because of nodes' mobility, that active route can be easily broken and this route must be quickly repaired so that packets are not dropped. When the connection is broken, the node upstream (closer to the source node) validates the routes to each of those destinations in its route table. It then creates *route error (RERR)* packet to send back to the source node. This packet contains a list of the destinations that now unreachable due to the broken link. ODV also

contains some features so that it can reduce overhead by using Time to Live (TTL) of the route request (RREQ) to limit the flooding of this packet in the network.

2.2.2 TORA

The Temporally Ordered Routing Algorithm (TORA) [19] is a highly adaptive loop free distributed routing algorithm based on the concept of link reversal is proposed to operate in a highly dynamic mobile networking environment. It is source initiated and provides multiple routes for any desired source/destination pair. The key design concept of TORA is the localization of control messages to a very small set of nodes near the occurrence of a topological change. To accomplish this, nodes need to maintain routing information about adjacent hop nodes. The protocol performs three basic functions:

- Route creation
- Route maintenance
- Route erasure

During the route creation and maintenance phases nodes use a height metric to establish a directed acyclic graph DAG rooted at the destination. Thereafter, links are assigned a direction (upstream or downstream) based on the relative height metric of neighboring nodes as shown in Figure 2.3. This process of establishing a DAG is similar to the query reply process proposed in LMR (Lightweight Mobile Routing). In times of node mobility the DAG route is broken and route maintenance is necessary to re-establish a DAG rooted at the same destination. As shown in Figure 2.4 upon failure of the last downstream link, a node generates a new reference level which results in the propagation of that reference level by neighboring nodes effectively coordinating a structured reaction to the failure. Links are reversed to reflect the change in adapting to the new reference level. This has the same effect as reversing the direction of one or more links when a node has no downstream links

Timing is an important factor for TORA because the height metric is dependent on the logical time of a link failure TORA assumes all nodes have synchronized clocks (accomplished via an external time source such as Global Positioning System). TORA's metric is a quintuple comprised of five elements namely:

- Logical time of a link failure
- The unique ID of the node that defined the new reference level
- Reflection indicator bit

- A propagation ordering parameter and
- The unique ID of the node

The first three elements collectively represent the reference level. A new reference level is defined each time a nodes loses its last downstream link due to a link failure TORA's route erasure phase essentially involves flooding a broadcast clear packet (CLR) throughout the network to erase invalid routes.

In TORA, there is a potential for oscillations to occur, especially when multiple sets of coordinating nodes are concurrently detecting partitions, erasing routes, and building new routes based on each other. Because TORA uses internodal coordination, its instability problem is similar to the count-to-infinity problem in distance vector routing protocols except that such oscillations are temporary and route convergence will ultimately occur.

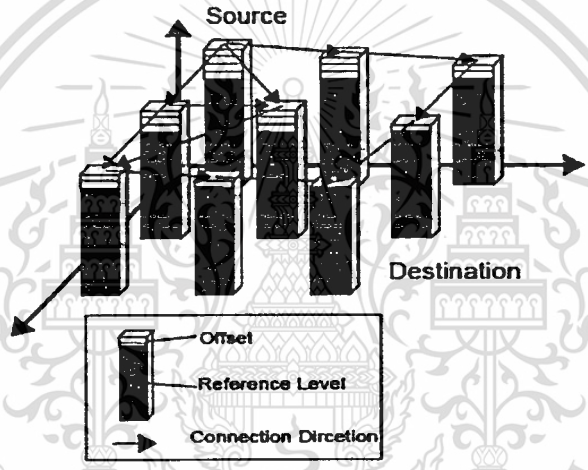


Figure 2.3: Route Creation (TORA principle)

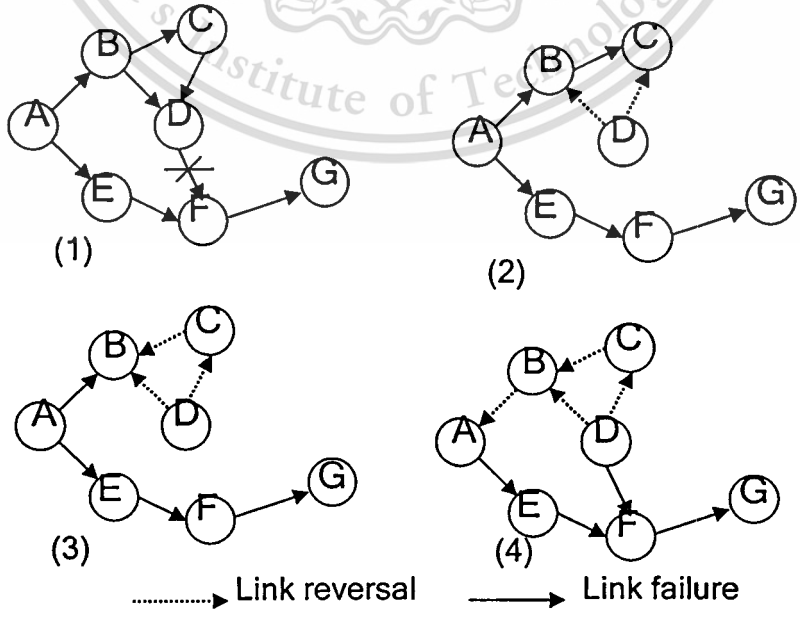


Figure 2.4: Route maintenance

2.3 Hybrid routing protocols

When wireless network becomes larger, it is complex to design routing protocols. All nodes in the network are separated into groups, called cluster. All clusters form a hierarchical infrastructure. In such network, hybrid routing protocols, i.e., combining proactive and reactive routing protocols, are used in order to take advantages on these two routing protocols where proactive maintains route in a cluster and reactive maintains route between clusters. Several hybrids routing protocols have been proposed such as Zone Routing Protocol (ZRP), Zone-based Hierarchical Link State (ZHLS), Distributed Dynamic Routing (DDR) and so on, but the most popular protocol is ZRP.

2.3.1 Zone Routing Protocol

The Zone Routing Protocol (ZPR) [18] integrates both proactive and reactive routing components into a single protocol. Around each node, ZPR defines a zone whose radius is measured in term of hops. Each node utilizes proactive routing within its zone and reactive routing outside its zone. Hence, a given node knows the identity of and a route to all nodes within its zone. When the node has data packets for a particular destination, it checks route table for a route. If the destination lies within the zone, a route will exist in the route table. Otherwise if the destination is not within the zone, a search to find a route to the destination is needed. Figure 2.3 illustrates the zone concept. In this figure, the zone radius is two hops.

For intrazone routing, ZPR defines the Intrazone Routing Protocol (IARP). IAPR is a link state protocol that maintains up-to-date information about all nodes within the zone. For any given node X, X's peripheral are defined to be those nodes whose minimum distance to X is the zone radius. In figure 10.5, S's peripheral nodes are nodes A, B, C and D. These peripheral nodes are important for reactive route discovery. ZPR utilizes the Interzone Routing Protocol (IERP) for discovering routes to destination outside of the zone. For route discovery, the notion of *bordercasting* is introduced. Once a source node determines the destination is not within its zone, the source broadcast a route query to its peripheral nodes. During the bordercast, the packet is delayed toward these peripheral nodes using tree constructed within the intrazone topology. After receiving the message, the peripheral node, in turn, check whether lies within their zone. If the destination is not located, the peripheral nodes, in turn, bordercast the query

message to their peripheral nodes. This process continues until either the destination is found or until the entire network is searched.

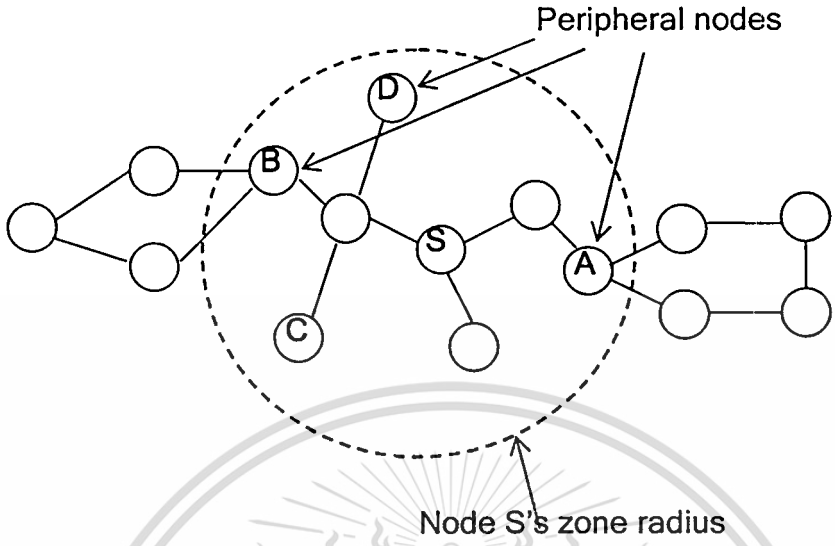


Figure 2.5: ZPR zone radius

Figure 10.6 illustrates an example of the bordercast discovery procedure. In the figure, node S performs a query for the destination X. By using the IARP, it learns that X is not within its zone. it bordercasts the query message into its peripheral nodes. In the Figure, the dotted circle represents the radius of S's zone. The peripheral nodes, in turn, check their zone, and after not finding the destination, bordercast the query message to peripheral nodes. The solid circles in the figure, represents the forward propagation of the query messages to each peripheral nodes. Hence, only the portion of each node's zone that have not been previously traversed by the query message is shown. Eventually, node G discovers X within its zone, and then unicasts a reply back to node S.

To improve query efficiency, a random query processing delay can be used as effective query control mechanism. By waiting a random interval between query exception and query forwarding, the chance of collisions during forwarding is reduced and, therefore, the effectiveness of the protocol is improved. In addition, ZPR defines other optimizations to reduce the messaging and processing overhead. In particular, these include early termination of queries by preventing a query from propagating into a zone that has already been searched for a destination.

Chapter 3

Dynamic Source Routing Protocol

Dynamic Source Routing (DSR) protocol is one of the most popular on-demand routing protocols, proposed by David A.Maltz, David B.Johnson and Yin-Chun Hu. The purpose of DSR is to reduce overhead and to provide highly reactive services to help ensure successful delivery of data packets to reach a destination even though topology of the network is changed or there is another condition in the network. Using DSR, the network is completely self-organizing and self-configuring, requiring no existing network infrastructure or administration. Network nodes cooperate to forward packets for each other to allow communication over multiple "hops" between nodes not directly within wireless transmission range of one another. As nodes in the network move about or join or leave the network, and as wireless transmission conditions such as sources of interference change, all routing is automatically determined and maintained by the DSR routing protocol.

DSR protocol is composed of two main mechanisms that work together to allow the discovery and maintenance of source routes in the ad hoc network. The first mechanism is Route Discovery. When a node, source node, wishing to send data packet to destination node, Route Discovery will be processed only the source node does not have a route to the destination. The second is Route Maintenance. When a route being used to send data packet from source node to destination is broken, the source node invokes Route Maintenance to find a new route for the rest of the packets on the source node to send to the destination node. For example in Figure 3.1, suppose a node A is attempting to discovery a route to node E.

In Route Discovery process, Node A transmits a route query packet to all nodes within the wireless transmission range in this example node B. Each Route Request contains the following information:

- Source node address: the node that originate RREQ packet (In this example, node A)
- Destination Node Address: the target node of RREQ as well as data packets once route found. (In this example, Node E)
- Unique Request Identification: The source node determines this identification in order to have a unique route request packet. (In this example, number 2)

- List of address: a record of addresses of nodes, which the route request packet has traversed. (In this example, Node B, C and D in addition the source node A and destination node E)

Any Intermediate node receives this route request packet (Node B in this example), if it is the target of the Route Discovery, it returns a route reply packet (RRPL) to the source node by reversing the addressed recorded in the route request packet. Once this route reply packet arrives at the source node, then the route found is cached in the source node's Route Cache for further use. If this node receiving the Route Request has recently seen another Route Request message from this initiator bearing this same request identification and target address, or if this node's own address is already listed in the route record in the Route Request, this node discards the Request. Otherwise, this node appends its own address to the route record in the Route Request and propagates it by transmitting it as a local broadcast packet (with the same request identification). In this example, node B broadcast the Route Request, which is received by node C; nodes C and D each also, in turn, broadcast the Request, resulting in a copy of the Request being received by node E.

Once node E, destination node, receives route request packet, it will send back route reply packet to the source node. In this example, Node E replies back to node A, "route reply" by examining its route cache for a route back to A, and if found, will use it for the source route for delivery of the packet containing the Route Reply. Otherwise, node E should perform its own Route Discovery for target node A, but to avoid possible infinite recursion of Route Discoveries, it must piggyback this Route Reply on the packet containing its own Route Request for A. Node E could instead simply reverse the sequence of hops in the route record that it is trying to send in the Route Reply, and use this as the source route on the packet carrying the Route Reply itself.

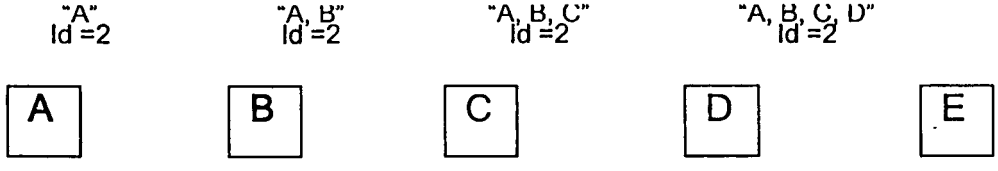


Figure 3.1: A simple Ad Hoc network topology

In Route Maintenance Process, after route found from either route discovery process or Route Cache, The source starts transmitting data packets to the destination by using that

route, each node transmitting the packet is responsible for confirming that data can flow over the link from that node to the next hop. In this case, node A is responsible for the link from A to B, node B is responsible for the link from B to C, node C is responsible for the link from C to D, node D is responsible for the link from D to E.

An acknowledgement can provide confirmation that a link is capable of carrying data, and in wireless networks, acknowledgements are often provided at no cost, either as an existing standard part of the MAC protocol in use [1].

After the acknowledgement request has been retransmitted the maximum number of times, if no acknowledgement has been received, then the sender treats the link to this next-hop destination as currently "broken". It should remove this link from its Route Cache and should return a "Route Error" to each node that has sent a packet routed over that link since an acknowledgement was last received. For example in Figure 3.2, If node C does not receive an acknowledgement from node D after some number of requests, it would return a Route Error packet to node A, as well as any other node that may have used the link from node C to node D since C last received an acknowledgement from D. Node A then removes this broken link from its cache. For sending another packets to the same target (node E), if A has another route in its Route Cache to node E, it can send the packet using the new route immediately. Otherwise, it should perform a new Route Discovery for this target (node E).

Figure 3.2: An example of broken route

3.1 Route Discovery Process

As briefly explained above, Route Discovery is the mechanism of a source node to obtain a route to destination for packets, which do not have a source route to the destination. This mechanism is invoked when a source node does not have a route to the destination. Route Discovery operates entirely on demand and it does not depend on any periodic or background exchange of routing information or neighbor node detection at any layer in the network protocol stack at any node.

The Route Discovery procedure utilizes two types of messages, a Route Request (Table 3.1) and a Route Reply (Table 3.2) to actively search the ad hoc network for a route to the

desired destination. Nodes related to Route Discovery can be classified into three groups, source node, intermediate node and destination node that are described in the section 3.1.1, 3.1.2 and 3.1.3 respectively. As shown in Figure 3.3 with a simple network scenario, the source node (A), intermediate node (B, C, and D), and Destination node (E) have different tasks in the Route Discovery process.

Route Discovery process for finding a destination should not be invoked unless a node has a packet in its Send Buffer requiring delivery to that destination. After each Route Discovery attempt, the interval between successive Route Discoveries for this target should be doubled, up to a maximum of MaxRequestPeriod, until a valid Route Reply is received for this target.

Table 3.1: Route request packet fields

Fields	Description
Source address	The address of the node that originating this packet
Dest. Address	The IP limited broadcast address
Hop Limit (TTL)	Varies from 1 to 255 to limit flooding of the route request packet.
Option Type	Nodes not understanding this option will ignore this option
Opt Data Len	8-bit unsigned integer. Length of the option excluding the Option Type and Opt Data Len fields.
Identification	A unique value generated by the source node of the Route Request. Nodes initiating a Route Request generate a new Identification value for each Route Request. This value allows a receiving node to determine whether it has recently seen a copy of this Route Request: if this Identification value is found by this receiving node in its Route Request Table (in the cache of Identification values in the entry there for this initiating node), this receiving node must discard the Route Request.
Target Address	The address of the node that is the target of the Route Request.
Address[1.....n]	Addresses of a route request packet has traversed in the network. The source address field is the address of the initiator of the Route Discovery and it must not be listed in the Address[i] fields. Thus the address given in Address[1] is the address of the first node on the path after the source node.

Table 3.2: Route reply packet fields

Fields	Description
Source address	The address of the node that originating this packet
Dest. Address	The IP limited broadcast address
Option Type	Nodes not understanding this option will ignore this option
Opt Data Len	8-bit unsigned integer. Length of the option excluding the Option Type and Opt Data Len fields.
Last Hop External	Indicates the last hop given by the route reply.
Reserved	Must be sent as 0 and ignored on reception
Target Address	The address of the node that is the target of the Route Request.
Address[1.....n]	The source route being returned by the route reply. These addresses come from the reversing Address[1..n] of route request packet.

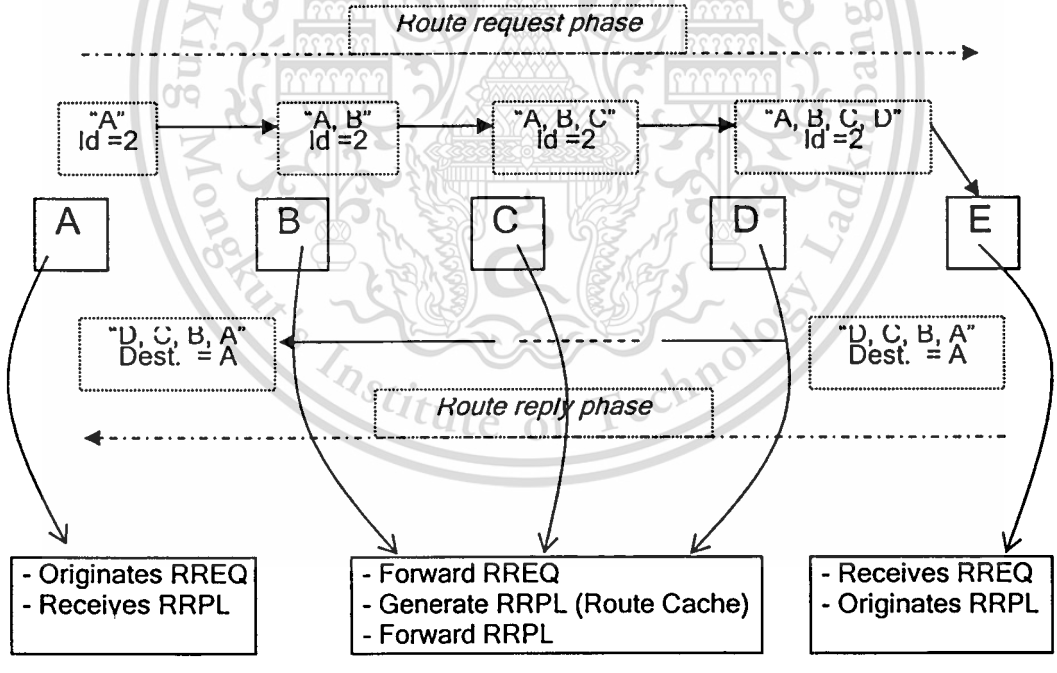


Figure 3.3: An example of Route Discovery Process

3.1.1 Source node

As depicted in Figure 3.3, a source node involving with Route Discovery has two main tasks, originating route query (RREQ) and receiving route reply (RRPL).

3.1.1.1 Originating a Route Request (RREQ)

Before broadcasting a route request packet for discovering destination node, the source node initializes a Route Request option (Table 3.1) in a DSR options header. The Route Request option must be included in a DSR options header in the packet. To initialize the Route Request option, the source node performs the following sequence of steps:

Step 1) The Option Type must be set to the value 2.

Step 2) The Opt Data Len field must be set to the value 6. The total size of the Route Request option when initiated is 8 octets; the Opt Data Len field excludes the size of the Option Type and Opt Data Len fields themselves.

Step 3) The Identification field must be set to a new value, different from that used for other Route Requests recently initiated by this node for this same target address. For example, each node MAY maintain a single counter value for generating a new Identification value for each Route Request it initiates.

Step 4) The Target Address field must be set to the IP address that is the target of this Route Discovery.

Step 5) Broadcast route request packet

Besides the fives steps above, the source node must maintain the information about Route Request in its Route Request table. When initiating a new Route Request, the node must use the information recorded in the Route Request Table entry for the target of that Route Request. The Route Request Table entry for the target records two important information, (1) Time To Live (TTL) field in the Route Query for the last Discovery, which allows the source node to implement various algorithms to control the spread of its route request and (2) the number of consecutive Route Requests initiated for this target since receiving a valid Route Reply giving a route to that target node, and the remaining amount of time before which this node may next attempt at a Route Discovery for that target node.

3.1.1.2 Receiving Route reply (RRPL) packet

When receiving route reply packet, the source node must do the following steps:

Step 1) Add the new route to its Route Cache for further use

Step 2) Check each packet in its Send Buffer, a queue of packets before sending, to determine whether a route to that packet's IP Destination Address now exists in the node's Route Cache including the route just added.

Step 3) If so, the packet should be sent using the route found and remove from the node's Send Buffer.

3.1.2 Intermediate Node

Intermediate nodes involving with Route Discovery have three main tasks, (1) receiving and forwarding route request, (2) generating route reply by using the Route Cache and (3) receiving and forwarding route reply (RRPL) as illustrated in Figure3.3. These three tasks are going to be described in the following sections.

3.1.2.1 Receiving and forwarding route request (RREQ)

When a node receiving route request packet, it performs the following steps:

Step 1) If the Target address field in the Route Request matches this node's IP address, then the node should originate a Route Reply (Section 3.1.3) and return route reply packet to the source node.

Step 2) Else, the node must examine the route recorded in the Route Request option to determine whether if this node's IP address already appears in the list of addresses. If so, RREQ must be dropped and won't be re-broadcast any more.

Step 3) Else decrease TTL value to control the flooding of route request in the mobile ad hoc network. If TTL countdown to 0 then, the node must drop this route request packet and stop re-broadcast further.

Step 4) Else, the node must search its Route Request table for the entry of source node (the IP source address field). If there is such entry, then the node must search the node's cache for identification (Identification field) values and the target node address of the route request to determine whether they just recently received. If so, the node must drop the route request packet (RREO).

Step 5) Else, the node must process the route query request according to the following steps:

- Copy identification and target address of this route query into its cache.
- Append this node's IP address to the list of Address[1..n] in the route request packet.
- Search its own route cache for a route to the target of this route request packet. If a route is found in its route cache, this node must generate a

route reply from route cache (next section) to return route reply packet to the source node.

- If no route found from its cache to the target, then this node should broadcast this route request packet to its neighbors.

3.1.2.2 Generating a route reply by using the Route Cache

Once a node receives route request packet and in order to avoid propagating the route request packet toward to the target node, this node checks in its Route Cache a route from this node to the target node. If there is such route, then Route Reply is generated by the node. This task is called "cache Route Reply" and this mechanism can greatly reduce the overall overhead of Route Discovery on the network by reducing the flood of Route Request.

If there is a route from the Route Cache for the route request to the target node, the node should construct and return a cached Route Reply as follows:

Step 1) The source route for this reply is the sequence of hop addresses:

initiator, Address[1], Address[2], ..., Address[n], c-route

Where initiator is the address of the initiator of this Route Request, each Address[i] is an address from the Route Request, and c-route is the sequence of hop addresses in the source route to this target node, obtained from the node's Route Cache.

Step 2) Send a Route Reply to the source node of the Route Request. The initiator of the Route Request is indicated in the Source Address field in the packet's IP header.

3.1.2.3 Receiving and forwarding route reply (RRPL)

Once a node receives any route reply packet, it then adds some new information gathered from that packet to its own cache. Every time that a node adds new information to its route cache, the node checks each packet in its own Send Buffer to determine whether a route to that packet's IP Destination Address now exists in the node's Route Cache. If so, the packet should be sent by using that route and removed from the Send Buffer.

3.1.3 Destination node

As depicted in Figure 3.3, the destination node involving with Route Discovery has two tasks, receiving RREQ and Originating Route Reply Packet (RRPL). Route Request packet can reach at any node in wireless range of the node that broadcasts or forwards RREQ, but every

node receives this RREQ, it will check whether this RREQ is searching for itself as mentioned in section 3.1.2. If the node is the target of RREQ, then the node must originate Route Reply in order to send back route reply packet to the source and control information is included in the Route Reply packet's field as described in Table 3.2. The process of originating route reply can be described as below:

Step 1) The Option Type in the option MUST be set to the value 3.

Step 2) The Opt Data Len field in the option MUST be set to the value (n * 4) + 3, where n is the number of addresses in the source route being returned (excluding the Route Discovery initiator node's address).

Step 3) The Last Hop External bit in the option MUST be initialized to 0.

Step 4) The Reserved field in the option MUST be initialized to 0.

Step 5) The Route Request Identifier must be initialized to the Identifier field of the Route Request that this reply is sent in response to.

Step 6) The sequences of hop addresses in the source route are copied into the Address[i] fields of the option. Address[1] must be set to the first-hop address of the route after the initiator of the Route Discovery, Address[n] must be set to the last-hop address of the source route (the address of the target node), and each other Address[i] must be set to the next address in sequence in the source route being returned.

3.2 Route Maintenance Process

Route Maintenance is the mechanism by which a source node is able to detect, while using a source route to some destination node, if the network topology has changed such that it can no longer use its route to the destination because a link along the route no longer works. When Route Maintenance indicates that a source route is broken, the source node can attempt to use any other route it happens to know to the destination, or can invoke Route Discovery again to find a new route for subsequent packets to the destination. Route Maintenance for this route is used only when the source node is actually sending packets to the destination.

When forwarding a packet, a node must attempt to confirm the reach-ability of the next-hop node, unless such confirmation had been received in the last MaintHoldoffTime. If no confirmation is received after the retransmission of MaxMaintRexmt acknowledgement requests, after the initial transmission of the packet, and conceptually including all

retransmissions provided by the MAC layer, the node determines that the link for this next-hop node of the source route is "broken". This confirmation from the next-hop node for Route Maintenance can be implemented using:

- Link-layer acknowledgement (Section 3.2.1)
- Passive acknowledgement (Section 3.2.2),
- Network-layer acknowledgement (Section 3.2.3)

If no acknowledgement is received after MaxMaintRexmt retransmissions, the node should originate a Route Error (Section 3.2.4) to the original sender of the packet (the source node).

3.2.1 Link Layer Acknowledgment

If the MAC protocol in use provides feedback as to the successful delivery of a data packet, then the use of the DSR Acknowledgement Request (Table 3.3) and Acknowledgement (table 3.4) options are not necessary. If such link-layer feedback is available, it should be used instead of any other acknowledgement mechanism for Route Maintenance, and the node should not use either passive acknowledgements or network-layer acknowledgements for Route Maintenance.

When using link-layer acknowledgements for Route Maintenance, the retransmission timing and the timing at which retransmission attempts are scheduled are generally controlled by the particular link layer implementation in use in the network. For example, in IEEE 802.11, the link-layer acknowledgement is returned after the data packet as a part of the basic access method of the IEEE 802.11 Distributed Coordination Function (DCF) MAC protocol; the time at which the acknowledgement is expected to arrive and the time at which the next retransmission attempt will occur are controlled by the MAC protocol implementation.

When a node receives a link-layer acknowledgement for any packet in its Maintenance Buffer, that node should remove that packet, as well as any other packets in its Maintenance Buffer with the same next-hop destination, from its Maintenance Buffer.

3.2.2 Passive acknowledgement

Passive acknowledgement is used for Route Maintenance when originating or forwarding packets along any hop other than the last hop (the hop leading to the IP Destination Address node of the packet). In particular, passive acknowledgement is used for Route

Maintenance in such cases if the node can place its network interface into "promiscuous" receive mode, and network links used for data packets generally operate bi-directionally.

In using passive acknowledgements for a packet that it originates or forwards, a node considers the later receipt of a new packet (e.g., with promiscuous receive mode enabled on its network interface) to be an acknowledgement of this first packet if both of the following two tests succeed:

- The Source Address, Destination Address, Protocol, Identification, and Fragment Offset fields in the IP header of the two packets must match, and
- If either packet contains a DSR Source Route header, both packets must contain one, and the value in the Segments Left field in the DSR Source Route header of the new packet must be less than that in the first packet.

When a node hears such a passive acknowledgement for any packet in its Maintenance Buffer, that node should remove that packet, as well as any other packets in its Maintenance Buffer with the same next-hop destination, from its Maintenance Buffer.

3.2.3 Network-layer acknowledgement

If the two mechanisms above are not available for a node to determine the reach-ability of the next hop when originating or forwarding a packet, the node should request a network-layer acknowledgement from that next hop node. In order to do that, the node must insert an Acknowledgement Request into the packet The Identification field in that Acknowledgement Request option must be set to a value unique over all packets transmitted by this node to the same next-hop node that are either unacknowledged or recently acknowledged.

When a node receives a packet containing an Acknowledgement Request option, then that node performs the following tests on the packet:

- If the indicated next-hop node address for this packet does not match any of this node's own IP addresses, then this node must not process the Acknowledgement Request option. The indicated next-hop node address is the next Address[i] field in the DSR Source Route option in the DSR Options header in the packet, or is the IP Destination Address in the packet if the packet does not contain a DSR Source Route option or the Segments Left there is zero.

- If the packet contains an Acknowledgement option, then this node must not process the Acknowledgement Request option.

Table 3.3: Acknowledgement Request packet fields

Fields	Description
Option Type	Nodes not understanding this option will ignore this option
Opt Data Len	8-bit unsigned integer. Length of the option excluding the Option Type and Opt Data Len fields.
Identification	The Identification field is set to a unique value and is copied into the Identification field of the Acknowledgement option when returned by the node receiving the packet over this hop.

Table 3.4: Acknowledgement packet fields

Fields	Description
Option Type	Nodes not understanding this option will ignore this option
Opt Data Len	8-bit unsigned integer. Length of the option, in octets, excluding the Option Type and Opt Data Len fields.
Identification	Copied from the Identification field of the Acknowledgement Request of the packet being acknowledged.
ACK Source Address	The address of the node originating the acknowledgement.
ACK Destination Address	The address of the node to which the acknowledgement is to be delivered.

3.2.4 Originating a Route Error

Once a connection has established and if a node (Node C in an example in Figure 3.2) along the established route, could not verify reach-ability of a next-hop node (Node CD in an example in Figure 3.2) after reaching a maximum number of retransmission attempts, the node should send a Route Error as described in Table 3.5, to the IP Source Address (Node A in an example in Figure 3.2) of the packet.

A node transmitting a Route Error must perform the following steps:

- Create a new packet which its Source Address field is set by the value of the address of this node.
- If the Salvage field in the DSR Source Route option in the packet triggering the Route Error is zero, then copy the Source Address field of the packet triggering the Route Error into the Destination Address field in the new packet's IP header; otherwise, copy the Address[1] field from the DSR Source Route option of the packet triggering the Route Error into the Destination Address field in the new packet's IP header
- Insert a DSR Options header into the new packet.
- Add a Route Error Option to the new packet, setting the Error Type to NODE_UNREACHABLE, the Salvage value to the Salvage value from the DSR Source Route option of the packet triggering the Route Error, and the Unreachable Node Address field to the address of the next-hop node from the original source route. Set the Error Source Address field to this node's IP address, and the Error Destination field to the new packet's IP Destination Address.
- If the packet triggering the Route Error contains any Route Error or Acknowledgement options, the node may not append to its Route Error each of these options.
- Send the packet back to the source node.

Besides originating route error of a node that detects broken route, another node that receives this route error packet that node must process the Route Error option as below:

- Remove some dead links in this node's Route Cache which are identified by the Error Source Address field to the node identified by the Unreachable Node Address field.
- If the option following the Route Error is an Acknowledgement or Route Error option sent by this node, copy the DSR options following the current Route Error into a new packet with IP Source Address equal to this node's own IP address and IP Destination Address equal to the Acknowledgement or Error Destination Address. Transmit this packet with the salvage count in the DSR Source Route option set to the Salvage value of the Route Error.

Table 3.5: Route Error packet fields

Fields	Description
Option Type	Nodes not understanding this option will ignore this option
Opt Data Len	8-bit unsigned integer. Length of the option excluding the Option Type and Opt Data Len fields.
Error Type	The type of error encountered. Currently, the following type values are defined: 1 = NODE_UNREACHABLE 2 = FLOW_STATE_NOT_SUPPORTED 3 = OPTION_NOT_SUPPORTED
Reserved	Must be sent as 0 and ignored on reception
Salvage	A 4-bit unsigned integer. Copied from the Salvage field in the DSR Source Route option of the packet triggering the Route Error.
Error Source Address	The address of the node originating the Route Error. The node that attempted to forward a packet and discovered the link failure.
Error Destination Address	The address of the node to which the Route Error must be delivered For example, when the Error Type field is set to NODE_UNREACHABLE, this field will be set to the address of the node that generated the routing information claiming that the hop from the Error Source Address to Unreachable Node Address was a valid hop.
Type-Specific Information	Information specific to the Error Type of this Route Error message.

Chapter 4

Modification of Dynamic Source Routing Protocol

As described in previous chapters, many routing protocols have been done by researchers in order to solve some problems in Mobile Ad Hoc Network. These routing protocols have pros and cons depending on network environments and these routing protocols do not consider or differentiate real-time traffic and non-real time traffic. Moreover Multimedia traffic is delay sensitive so these routing protocols might not satisfy such sensitive delay-traffic. In order to support this kind of traffic, we propose a Modification of Dynamic Source Routing (MDSR) protocol, which modified some mechanisms in Dynamic Source Routing protocols (Chapter 3). We are going to describe the mechanism of MDSR on supporting multimedia services in wireless network in the following sections.

4.1 Design Methodology

In MDSR protocol, Data transmission in MANET is separated into two groups. The first group is priority transmission of real-time data stream which is delay-sensitive data such as on-demand multimedia stream, video conference etc., and the second group is normal data stream which allows the delay in during transmission. We apply our algorithm differently for these data transmissions. When the multimedia traffic is applied and in order to reduce query response time, our algorithm uses larger transmission power level or in other words, the number of nodes in the range of the source is increased. Therefore, source node can select one farthest node to forward route request. In turn, once the farthest node receives the route request packet, It processes bandwidth reservation mechanism, described in the next section, in order to allocate bandwidth for this link whether it satisfy QoS requirement or not. If available bandwidth, then the node re-broadcast the route request packet otherwise drop route request packet. On the other hand, for non-real time traffic which delay is not sensitive, smaller transmission power level is used. Each node receiving route request packet must process bandwidth reservation so that the route being established will satisfy QoS requirement. In Figure 4.1, an example of the proposed routing algorithm, we can see that the dotted line (S-A-B-D) representing a multimedia transmission has shorter path than the route used for non-real time data transmission that is

represented by solid line (S-M-N-O-Q-D). In this example, we assume that all nodes receiving request packet have available bandwidth to guarantee all kind of traffic.

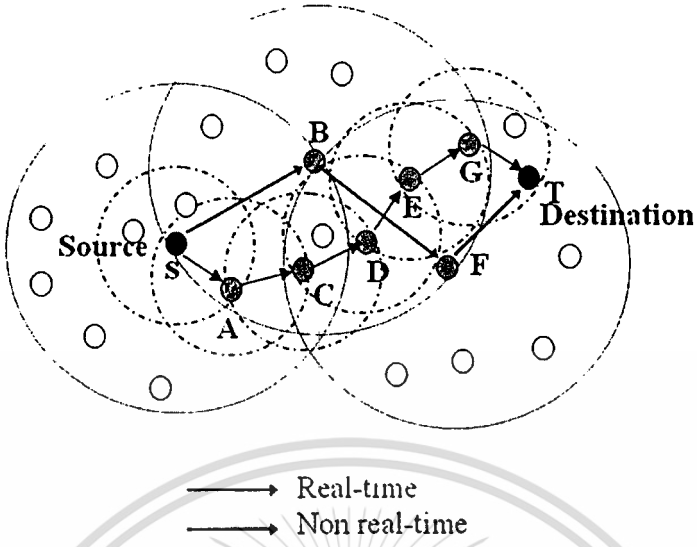


Figure 4.1: An Example of proposed scheme.

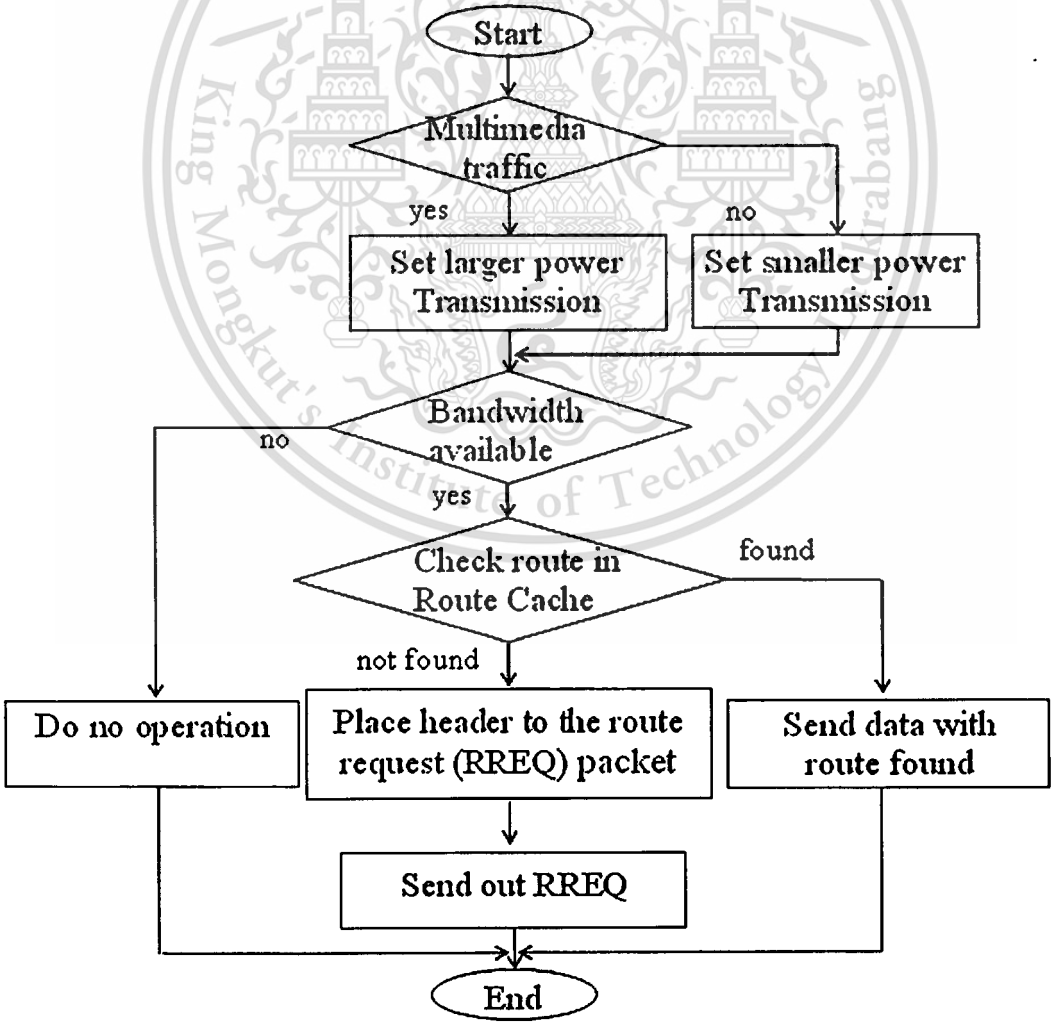


Figure 4.2: Flowchart of source node wishing to send data packets

Figure 4.2 shows the process of a source node that wishes to send data packet. At first it checks to see what kind of traffic. If multimedia traffic the node sets higher power strength, else set lower power strength. Second, it checks whether bandwidth is available for this connection. If bandwidth is not available, it does no operation. Otherwise it searches for a route in the Route Cache previously learned whether there is route to the destination to satisfy the traffic. If a route is found, the source node uses it. In this research, we added more three fields to the existing original packet header of Dynamic Source Routing Protocol (DSR) such as (1) ***slot_list*** is to keep status of slots being used by a node, which a route request packet has traversed. (2) ***require_bw*** is a field of a packet. Its task is to keep how big bandwidth is required for this new link. (3) ***power***, the last added field, differentiates traffic in the network. These three fields are going to be used so often in the bandwidth reservation mechanism. Finally, the source node floods route request (RREQ) packet to its neighbors which can be intermediate node or destination node.

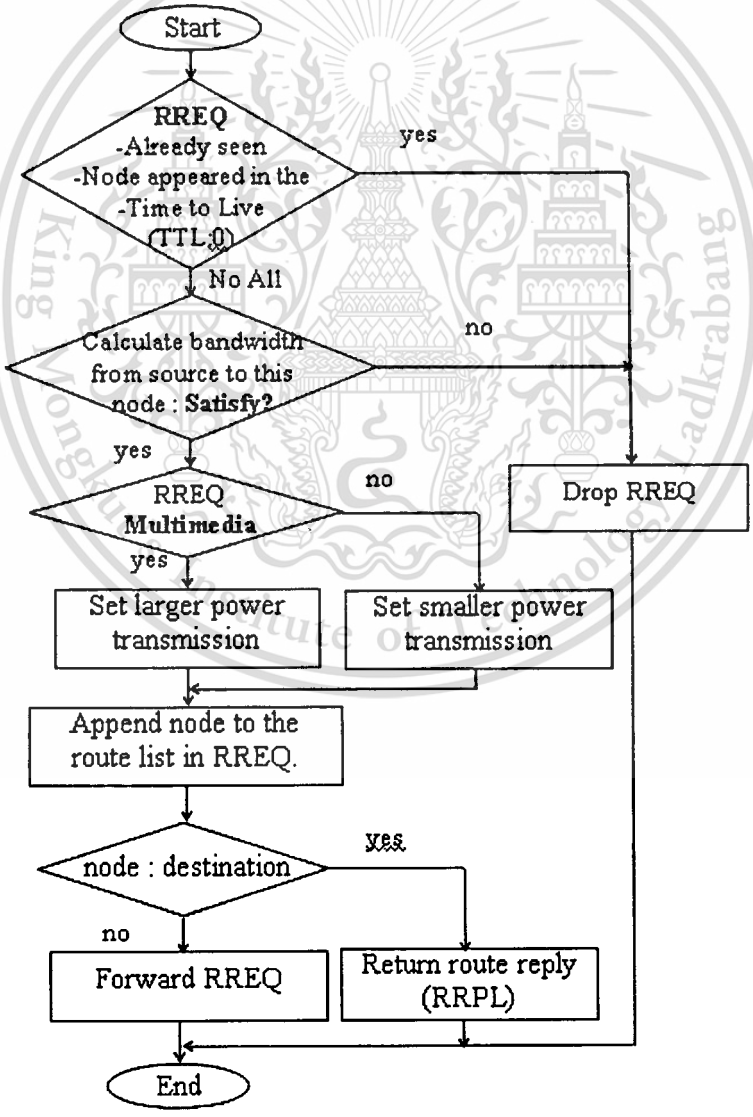


Figure 4.3: Flowchart of an intermediate node or destination node,

wishing to send data packets

Figure 4.3 shows additional mechanism of this research that is a process of an intermediate node or destination node that receives route request packet. At first, it checks to see that if the pair (dest-addr, sequence#) for this RREQ is seen recently, discard this packet and do not process it further. Or if the address of this node appeared in the route-list in the RREQ, we drop this RREQ (do not re-broadcast it) and do not process it further. Second, TTL (time to live) is decremented by one. If TTL counts down to zero, we drop this RREQ and do not process it further. TTL can limit the length of the delivery path. However, this path will be difficult to be maintained within a dynamic environment. In addition, unlimited packet flooding will deteriorate the network performance. The use of TTL can control the flooding traffic. Third, it checks whether the route request is searching for the path for multimedia traffic or datagram traffic (text application traffic). If multimedia traffic, the node set higher power level strength otherwise set lower power strength. Finally it appends its address such as IP the route-list to track the route which the packet has traversed if this node is destination, send route reply back to the source otherwise re-broadcast this route request.

4.2 Bandwidth Reservation Mechanism

Multimedia applications such as digital audio and video have much more stringent QoS requirements than traditional datagram applications. In cellular networks, all stations learn of each other's requirements, either directly or through a control station such as base station in cellular systems. However, the solution used in cellular systems cannot be applied in MANET where all elements are mobile. To support QoS for multimedia applications, we need to know not only the minimal delay path to the destination, but also the available bandwidth on it. A pair of connection should be accepted only if there is enough available bandwidth. We only consider "bandwidth" as QoS because bandwidth guarantee is one of the most critical requirements for multimedia applications. "Bandwidth" in time slotted network systems is measured in terms of the amount of "free" slots. The goal of the QoS routing algorithm is to find a shortest path such that the available bandwidth on the path is above the minimal requirement. To compute the "bandwidth"-constrained shortest path, we not only have to know the available bandwidth on each link along the path, but we also have to determine the scheduling of free slots.

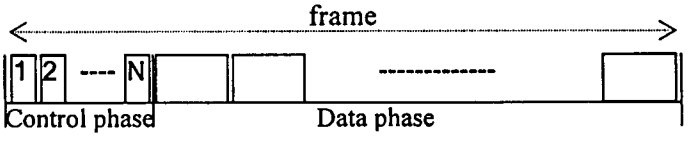


Figure 4.4: Frame structure

4.2.1 Bandwidth Calculation

The transmission time scale is organized in frames, each containing a fixed number of time slots. The entire network is synchronized on a frame and slot basis. Each frame is divided into two phases, namely, the control phase and the data phase, as shown in Figure 4.4. The size of each slot in the control phase is much smaller than the one in the data phase. The control phase is used to perform all the control functions, such as slot and frame synchronization, power measurement, code assignment, slots request. The amount of data slots/frame assigned to a pair of connection is determined according to a QoS requirement. As depicted in Figure 4.4, the control phase uses pure Time Division Multiple Access (TDMA) with full power transmission in a common code. That is, each node takes turns to broadcast its information to all of its neighbors in a predefined slot, such that the network control functions can be performed distributively.

Because only adjacent nodes can hear the reservation information and the network is multihop, the free slots recorded at every node may be different. We define the set of the common free slots between two adjacent nodes to be the *link bandwidth*. Consider the example shown in Figure 4.5, in which node C intends to compute the bandwidth to node A. We assume the network topology is shown in Figure 4.5. If node B can compute the available bandwidth to node A, then node C can use this information and the "link bandwidth" to node B computes the bandwidth to node A.

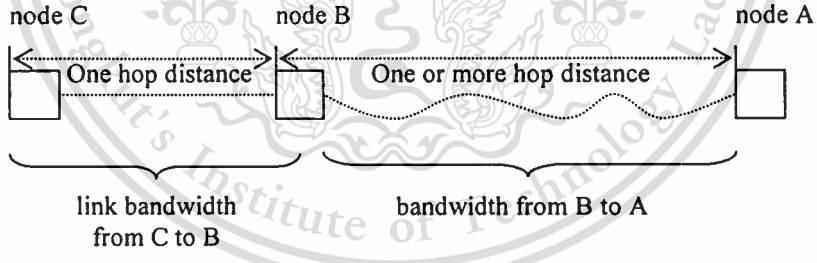


Figure 4.5: End-to-end bandwidth calculation (from C to A)

We define the path bandwidth (which can be called end-to-end bandwidth) between two nodes, that are not necessarily adjacent, to be the set of available slots between them. If two nodes are adjacent, the path bandwidth is the link bandwidth. Consider the example in Fig. 4.5, and assume that one hop distance is between node A and node B. If node C has free slots {1, 3,4}, and node B has free slots {1, 2, 3}. Then the link bandwidth between and is {1, 3}. This means that we can only exploit slots 1 and 3 for packet transmission from node C to node B. Thus, if a connection needs more than two slots in a time frame, then it will be rejected to pass through (C, B). We can observe that link_BW(A,B) = free_slot(A) ∩ free_slot(B). The

definition of free_slot(X) is the slots which are not used by any adjacent host of X to receive or to send packets from the point of view at node X. Next, we can further employ link bandwidth to compute end-to-end bandwidth. We will use the following four cases as examples to show how to calculate the path bandwidth.

Case 1: Assume the link bandwidth of both (A, B) and (B, C) are the same, say {1, 2, 3, 4}, as in Figure 4.6. If node C uses slots 1, 2 to send packets to node B , then node B can only use slots 3, 4 to forward packets to node A. This is because node B cannot be in transmitting mode and listening mode simultaneously. So the path bandwidth from node C to node A, denoted as path_BW(C, A), can be {1, 2}, and its size is two. In this case, four free slots can only contribute two slots for path bandwidth. Namely, [4/2] =2. Similarly, if there are only three free slots on both links, then the size of path bandwidth is [3/2] =1.

Case 2: Assume link_BW(A, B) ={2,3} and link_BW(B, C) ={1,2,3,4}, as in Figure 4.7. Namely, link_BW(A, B) c link_BW(B, C). If node C uses slot 2, then node B cannot use slot 2 any more. So in this case, node C should first use slots in link_BW(B, C) – link_BW(A, B) = { 1,4} to maximize system utilization. Therefore, if node C uses slots 1, 4, then node B can use slots 2, 3. So path_BW(C, A) = {1,4} , and its size is two. Similarly, we can use the same way to process the case of link_BW(A, B) C link_BW(B, C). In this case, node B must use slots in "link_BW(A, B) – link_BW(B, C)" first.

Case 3: If link_BW(A, B) ∩ link_BW(B, C) = ϕ, no conflict will occur. Figure 41.0 shows this example. Node C can choose either slot 3 or 4, and node B can choose slot 2, so path_BW(C, A) = {3} and its size is one.

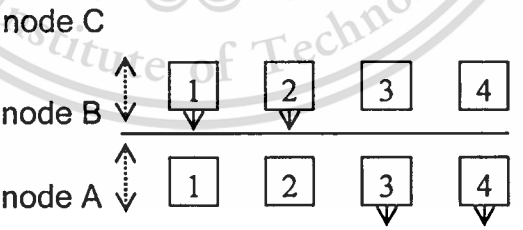


Figure 4.6: Equal case

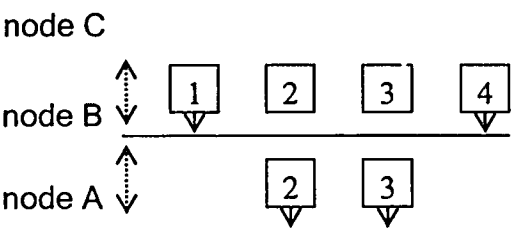


Figure 4.7: Containing case

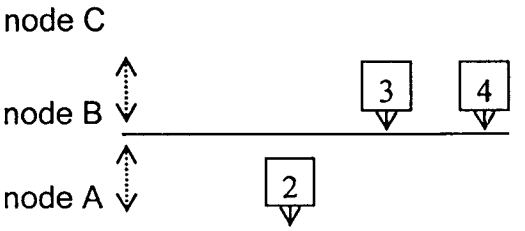


Figure 4.8: Exclusive case

Case 4: This is a general case, as shown in Figure 4.9. We will find any general case can be regarded as a combination of the previous three cases. Follow the slot assignment policy in Case 2. We assign slot 9 to node C and slot 1 to node B first. Figure 4.10 shows the slots left. Next, we assign slot 10 to node C and slot 4 to node B. Figure 4.11 shows only slots {5, 6, 7, 8} left (slot 4 cannot be used by node C any more since node B is in transmitting mode). At present, this is the same situation as in Case 1. So node C can be assigned slots 5, 6, and node B is assigned slots 7, 8, as shown in Figure 4.12. Here we let the path_BW(C, A) = {5, 6, 9, 10}. That is, node C can use slots {5, 6, 9, 10} to send packets to node B, and then node B uses {1, 4, 7, 8} to forward packets to node A. The size of path bandwidth from to is four.

The last case is a general case. Observe that it is, in fact, a combination of Cases 1–3. Our slot assignment policy is to consider the slots not in link_BW(B,C) ∩ link_BW(A,B) first, until one of the special cases (i.e., Cases 1–3) occurs. The detail of the bandwidth calculation algorithm is shown in Figure 4.14. To further generalize Case 4, if the distance between and is no longer one hop, as shown in Figure 4.5, the same algorithm can still work. Node C can calculate path_BW(C, A) by using link_BW(C, B) and path_BW(B, A) by using the algorithm in Figure 4.15.

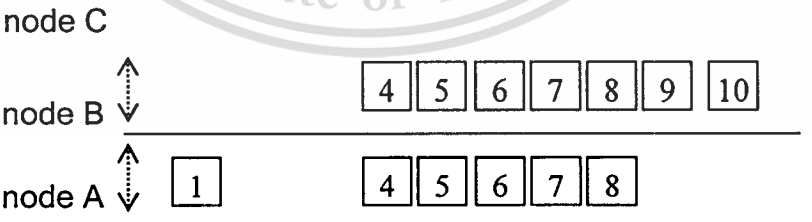


Figure 4.9: General case

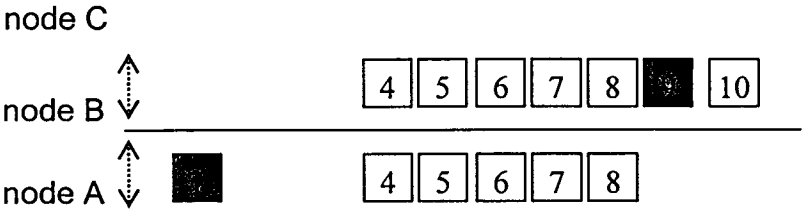


Figure 4.10: Step 1 bandwidth calculation in node C

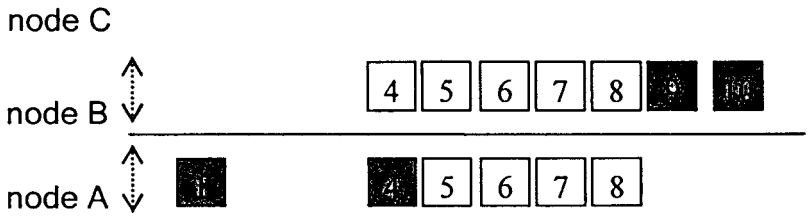


Figure 4.11: Step 2 bandwidth calculation in node C

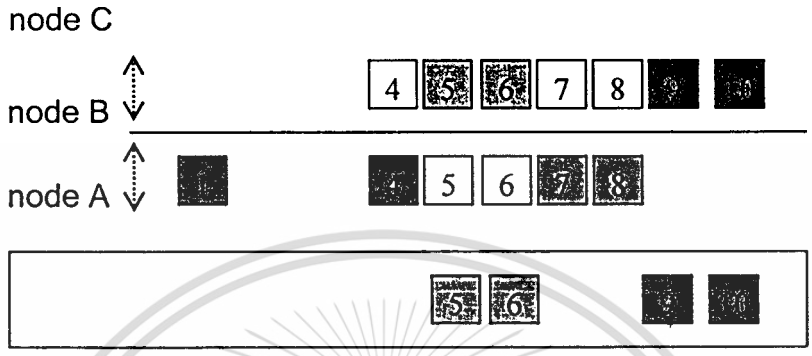


Figure 4.12: Final result of bandwidth in node C

As mentioned in Section 4.1, three fields are added to an original packet of Dynamic Source Routing protocol, such as **slot_list, require_bw, power** where *slot_list* is to keep all state of slots of every node that a route request packet has traversed, *require_bw* is to keep what size of a connection is, and *power* field is used to separate multimedia traffic and datagram traffic (text application). Therefore a packet header has fields as below:

<packet-type, source-add, dest-add, sequence#, route-list, slot_list, require_bw, power, data, TTL>

The algorithm in Figure 4.2 and Figure 4.3 can be replaced by DSR algorithms shown in Figure 4.13 and 4.15 respectively.

Figure 4.13 shows the process of a node that wish to send data packets. First it checks whether the traffic is multimedia traffic. If so, the node sets higher power strength, else set lower power strength and the radius of that node is modified according to what kind of traffic is. Second it checks its bandwidth to see whether it is available. If its bandwidth is not available, that node takes no actions and waits for reserved slots to be released. But in contrast, available bandwidth, this node searches for a route in its Route Cache, a place to store route information to another node in wireless network. If it found a route to destination that satisfies the traffic including bandwidth, the node uses that route to transmit data packets. Otherwise, the node has no route to destination, then it processes route query request by adding related information to the header of RREQ and flooding this RREQ to its neighbor nodes.

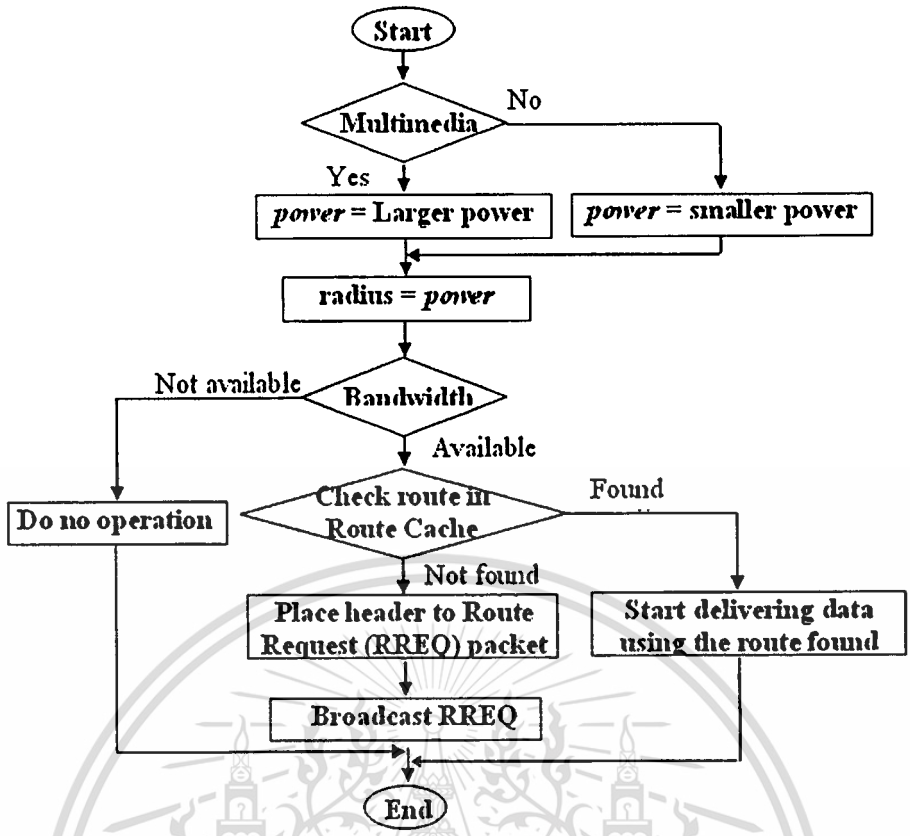


Figure 4.13: The algorithm for a node whishing to send data packets.

Figure 4.15 shows process of that receives route request packet. This node can be any intermediate node or destination node. At first, the node receiving route request packet checks whether this route request packet (<dest-addr, sequence#>) has seen recently, then this route request packet is dropped. Second, if the node's address already appeared in the *route-list* field in RREQ, this node does not rebroadcast and drop this RREQ. Third, Time To Live (TTL) is decremented by one. If TTL counts down to zero, we drop this RREQ and do not process it further. TTL can limit the length of the delivery path. Fourth, if this node is destination, the go to process the algorithm in Figure 4.18. Fifth, Calculate the bandwidth from source to this node following the algorithm provided in Figure 4.14. If bandwidth resulting from bandwidth calculation algorithm is not available, drop the route request packet (RREQ). The status of slots is not modified at this time. Otherwise, available bandwidth, the state of data slot of this node is recorded into *slot_list* field of RREQ. After that, the node uses power level according to what kind of traffic that this RREQ is searching a route for. Finally, route request packet (RREQ) is re-broadcasted to its neighbor nodes.

```
int bandwidth_calculation( int i){
if (the numbers record in slot_list or i  == 0)
        printf("Never occours, Error \n")
        return -2;
else if (the numbers record in slot_list or i == 1 ) {
         link_BW = slot_list[1] & slot_list[2] ;
           // slot_list[1] & state of slots of the node receiving route query request(RREQ)
         return link_BW;
}
else if (the numbers record in slot_list or i == 2){
        i = i – 1;
        link_bw_i = slot_list()[i] & slot_list()[i+1] ( state of slots of node receiving RREQ)
        bandwidth = bandwidth_calculation(i);
        common_BW = bandwidth & link_bw_i;
        common_BW_size = sizeb(common_BW);
        diff2 = bandwidth ^ common_BW;
        difference_BW1 = sizeb(link_bw_i ^common_BW); // the symbol ^  means XOR
        difference_BW2 = sizeb(bandwidth ^ common_BW);
        if ( difference_BW1 <= difference_BW2) {
                bandwidth_size = difference_BW1;
                remain_BW_size = difference_BW2-difference_BW1;
        }
        else {
                bandwidth_size = difference_BW2;
                remain_BW_size = difference_BW1 - difference_BW2;
               }
       if ( ( remain_BW_size > 0 ) || ( common_BW_size > 0)) {
            if ( common_BW_size <= remain_BW_size)  bandwidth_size = bandwidth_size
                + common_BW_size;
           else {
                 bandwidth_size = bandwidth_size + remain_BW_size;
                 common_BW_size = ( common_BW_size - remain_BW_size)/2;
                 if ( common_BW_size > 0)  bandwidth_size= bandwidth_size +
                     common_BW_size;
                 }
        }
        if (bandwidth_size >= requested bandwidth) return link_bw_i;
        else return -2; // no bandwidth available
    }
}
```

Figure 4.14: Path bandwidth calculation algorithm for a node receiving route query request

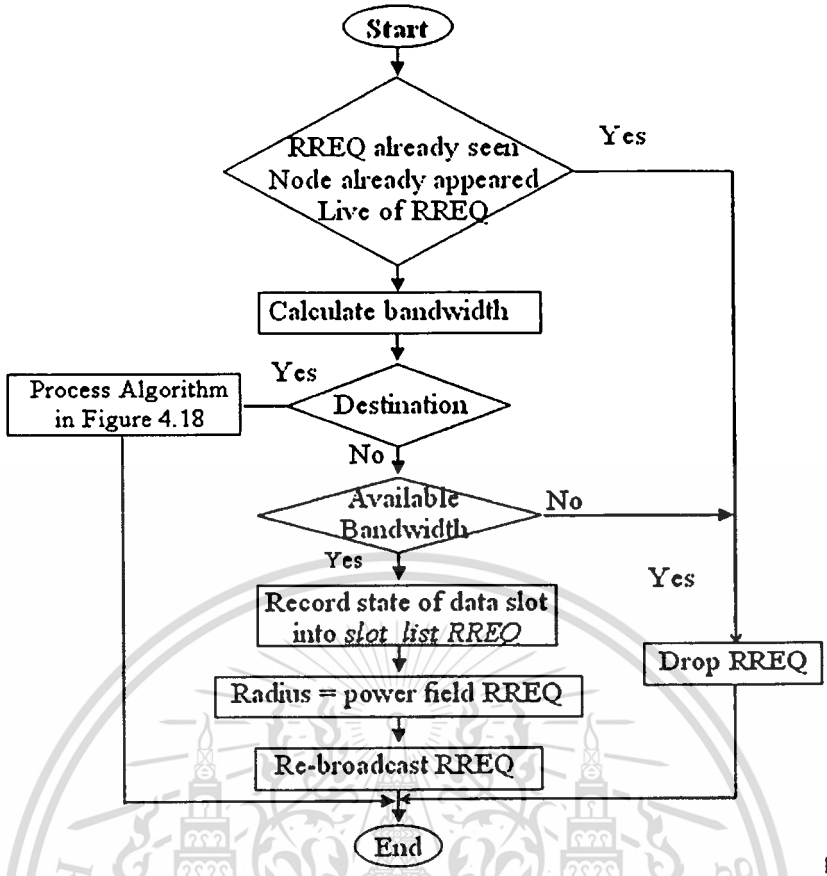


Figure 4.15: The algorithm for a node receiving route request packet.

4.2.3 Slot Assignment

In previous section, we have described how to calculate path bandwidth from source to destination during route query request process mechanism of Dynamic Source Routing protocol. The algorithm in Figure 4.13 only calculates the size of path bandwidth and end-to-end path bandwidth by using the bandwidth information in route request packet. However, it does not tell us how to assign the available slots efficiently during the call setup in each host. In this section, we will discuss how to do the slot assignment. Once a route request packet arrives at destination, the process of slot assignment must be done in order to assign slots to every node that the route request packet has traversed. As result, the information of slot assignment is placed in the header of route reply packet that is going to be sent back to the source. Once any intermediate node receives route reply packet, it must set its slots to be occupied according to the slot assignment information in route reply packet header. We use Figure 4.9 as an example to describe how the slot assignment algorithm works. For a given path, the source node, immediate nodes, and the destination node will do different work. At first, we are going to describe the process of destination node because it is the only node that does the process of slot assignment algorithm which information of slot assignment is placed

in the packet header of route reply. From this route reply packet, any intermediate or the source node can set their slots' status according to the slot assignment information in the route query packet.

Destination Node: When the destination receives route request packet, first it calculates bandwidth from source to this node by using state of slot information in route request packet. If the bandwidth from source to destination does not satisfy, the destination node drops route request packet (RREQ). Otherwise (available bandwidth), it processes of assigning slots to every node that RREQ has traversed. As result, the information of this process will be used in route reply packet (RREP) as information for every intermediate node to set their status to be occupied by this connection. According to link_BW(C, B) and link_BW(B,A), we can compute path bandwidth as illustrated in Figure 4.12. Slots {5, 6, 9, 10} and Slots {1, 4, 7, 8} are recorded in an array of slot_list of route reply packet. Finally the destination node does some other tasks such reserving slots {1, 4, 7, 8} for this connection, setting power field of RREP, and sending out route reply packet (RREP) back to the source by using reversing route in RREQ. This algorithm is given in Figure 4.18.

Intermediate Nodes: When node B receives route reply packet (RRPL), containing bandwidth formation from node A, then it uses this information to check its slots' status. First it changes the power level to forward data packets according to the information of power field in route reply packet (RRPL). Second it checks whether the incoming slots {5, 6, 9, 10} are free. If so, it can receives data packets from node C. In addition, must check if there are free slots such as {1, 4, 7, 8}, outgoing slots, which can be used for forwarding packets to next hop. If any one slot in {5, 6, 9, 10} is busy or if there are no enough free slots {1, 4, 7, 8} to forward the packets, this bandwidth reservation will fail. So the connection must be rejected as result of lacking of available either incoming free slots or outgoing free slots then node B must send a control packet, say RESET, to node A ask to free the slots {1, 4, 7, 8}. If all slots are available, then this intermediate node reserves all these slots for this connection or in other word, the status of these slots are set to be occupied. These checking operations have to be done because the topological change may affect the free slots and this algorithm is given in Figure 4.17.

Source Node: Once route reply packet arrives at the source node, first the source node checks and uses power according to the power field of this packet. Because the source node needs to reserve slots to be used for transmitting, it checks only outgoing slots. In this example, the source node checks only {5, 6, 9, 10}. If these slots are available, the source node reserves these slots for outgoing transmission. After reserving these slots, it records this route in its Route Cache for further use and after that it starts transmitting data packets. If either one of {5, 6, 9, 10} is not free, a control packet (RESET) is going to be sent back to the node that sent RRPL in order for this node to release its slots reservation. The algorithm of this process is given in Figure 4.16.

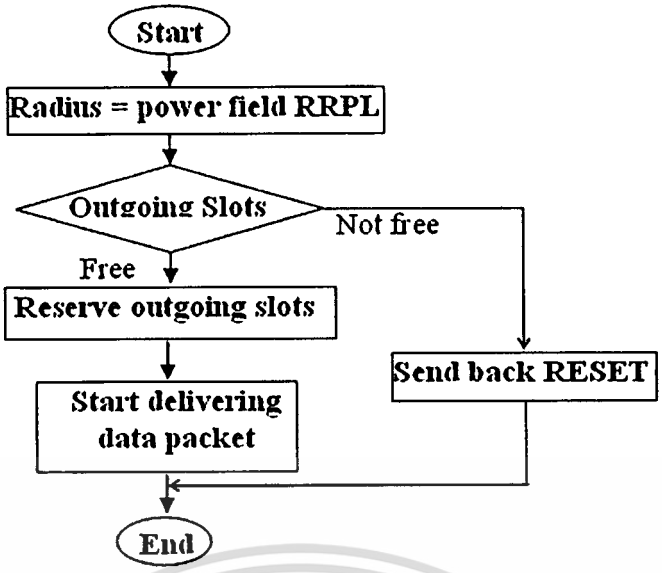


Figure 4.16: The process of slot assignment for a source node that receives route reply packet (RRPL)

Figure 4.16 shows slot assignment process of source node. This process happens when the source node receives route reply packet (RRPL). At first, the source node checks power field of RRPL to see whether it is multimedia traffic. If so, larger power level is used otherwise smaller power level is used instead. After that the source node also checks its outgoing slots that are used to send out data packets. If there are available outgoing slots, then those slots are reserved for this connection. On the other hand, this route found is added to Route Cache for future use. Data packet is being delivered afterward. In case Outgoing slots are not available, the source node sends back RESET packet to released all reserved slots.

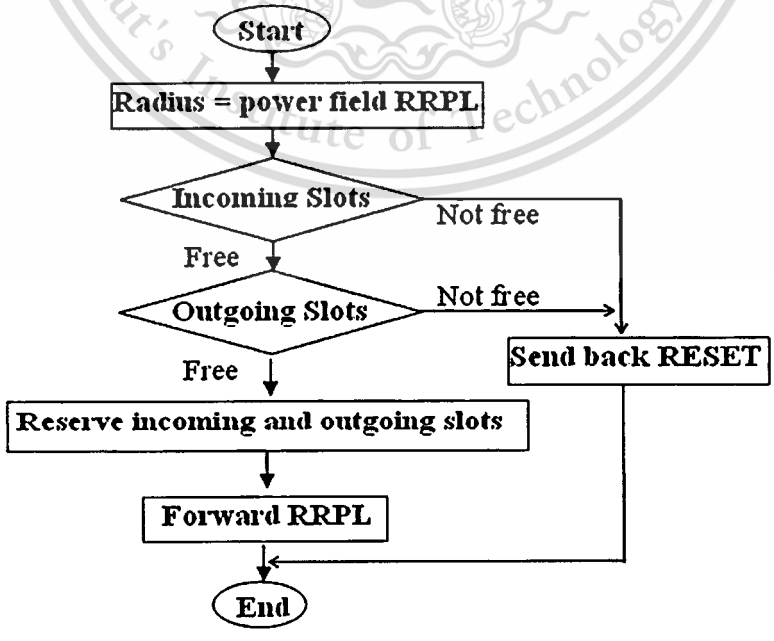


Figure 4.17: The process of slot assignment for an intermediate node that receives route reply packet (RRPL)

When any intermediate node receives route reply packet, slot assignment process is invoked as shown in Figure 4.17. At first, that intermediate node checks power field of RRPL whether it is multimedia traffic. If so, larger power level is applied otherwise smaller power level is applied. After that, incoming slots (slots used to receive data packets) and outgoing slots (slots used to forward data packets) are also checked to see whether they are free. If both incoming slots and outgoing slots are free, these slots are reserved for this connection and route reply packet is forwarded to the source node. Otherwise, RESET packet is sent back to release all reserved slots.

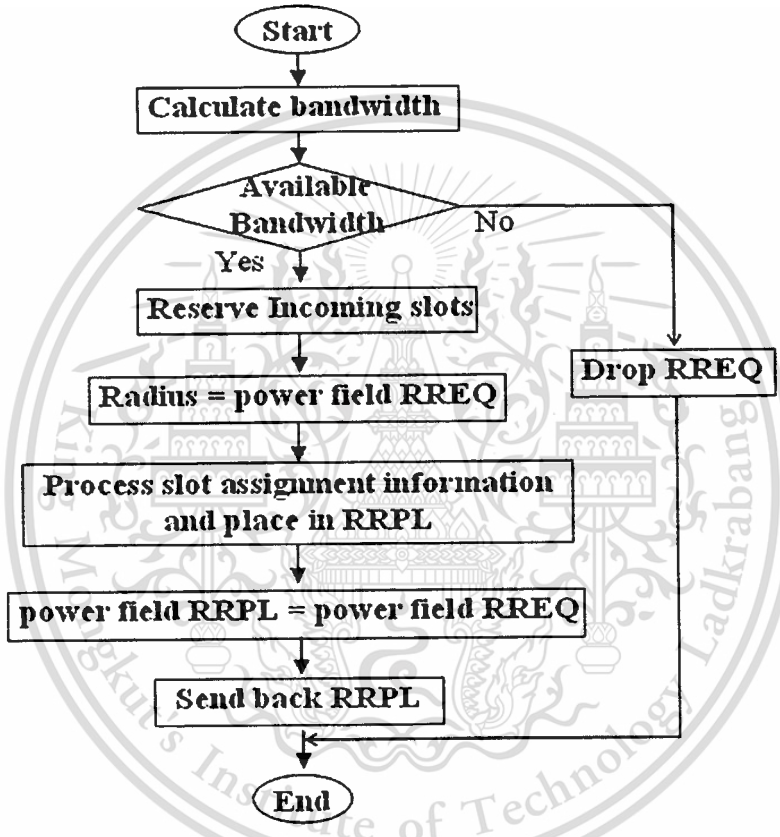


Figure 4.18: The process of slot assignment for destination node that receives route request packet (RREQ)

Destination node is the most important node for slot assignment process because it is the only node that produces slot assignment information by using information in *slot_list* field of RREQ. Moreover it also invoked Route Reply process. As result, route reply packet placed with slot assignment information is sent back to a node that initiates the route request packet. The detail of destination's slot assignment process is described as below.

At first, Destination node invokes a process to calculate bandwidth from source to this node (Figure 4.14). If Bandwidth does not satisfy the connection, route request packet is dropped. Otherwise (available bandwidth), Slot assignment information is produced by using slots' status information recorded in *slot_list* field of RREQ. This slot assignment information is

placed in route reply packet (RRPL) that is the most important information for any intermediate node or source node, which is involved in doing slot assignment process. Second, destination node reserves incoming slots, which are used to receive data packets. Third, it applied power level according to what kind of traffic that the route request packet is searching for. If the RREQ is searching for a route for multimedia traffic, larger power level is applied otherwise smaller power level is applied. Finally, route reply packet (RRPL) embedded with slot assignment information is sent back to the source node.

Chapter 5

Performance Evaluation

Network Simulation version 2 (NS2) is a well-known network simulation application that has been designing and developing by organizations and universities [15]. We use ns2 to evaluate the performance of MANET based on simulation environment (Section 5.1) and simulation assumptions (Section 5.2). Finally, the results of simulations are given in Section 5.3.

5.1 Simulation Environment

Network simulation resides on the area of 800 meters by 800 meters, where 20 mobile nodes can be moved randomly and freely at the speed up to 2 meters per second. Radio transmission range of each node varies from one node to another, depending on what kind of traffic that node is involving to transmit. If nodes involving with multimedia traffic, their radio transmission ranges are 300 meters. Otherwise their radio ranges are 250 meters. The data rate is 2 bps.

Table 5.1: Simulation Parameters

Parameters		**Values**
Simulation Area		800 m x 800m
Number of nodes		20
Speed of nodes		0-2 m/s
Node movement		Randomly
Transmission range	Real-time	300 meters
	Non real-time	250 meters

5.2 Simulation Assumptions

As described in the section bandwidth calculation (Section 4.2.2), the transmission time scale is organized in frames, each containing a fixed number of time slots. The entire network is synchronized on a frame and slot basis. Each frame is divided into two phases, namely, the control phase and the data phase, as shown in Figure 4.6. In this time frame, the data slot in the data phase is 5 ms and the control phase is 0.1 ms. we assume that there are 16 slots in data phase. We can calculate the frame length as the equation below:

Frame length = number of mobile nodes * Control phase + number of slot * data slot

Frame length = 20 * 0.1 + 16 * 5 = 82 ms

There are two types of Quality of Service (QoS) for the offered traffic. QoS_1 and QoS_2 need one and two data slots respectively. The total simulation is 2000 s (2 * 10 ^6 ms). Since the number of data slots is less than the number of mobile nodes, mobile nodes need to compete for these data slots. Once a connection has been established, a transmission window (data slot) is reserved automatically for all the subsequent packets of the established connection.

As described the simulation parameters in section 5.1, we also assume that all nodes wishing to communicate with other nodes within the ad hoc network are willing to participate fully in the protocols of the network. In addition, each node participating in the ad hoc network should also be willing to forward packets for other nodes in the network. Moreover, those nodes can move at any time without notice and move continuously.

Finally, we also assume that a node must be able to receive packets if it is in the radius of sender node. Moreover these packets may be lost or corrupted in transmission on the wireless network. We assume that a node receiving a corrupted packet can detect the error and discard the packet.

5.3 Simulation Results

We use ns2 to simulation two different algorithms. The first algorithm is to use higher power level for multimedia traffic and smaller power level for datagram traffic (text traffic). The aim of this algorithm is to find the shortest path for Multimedia traffic. The simulation results of this first algorithm are in section 5.3.1. Another algorithm is MDSR algorithm that is described in Chapter 4. The aim of MDSR is to find the shortest bandwidth path for guarantee multimedia traffic. It means that the route found is the shortest and each node along the route has enough bandwidth to send or forward multimedia packets. The results of simulation will be shown in section 5.3.2.

5.3.1 Result of Power Level

In this algorithm, traffic is separated into two types. The first type of traffic is Multimedia traffic, which is delay-sensitive. The second type of traffic is datagram traffic (text traffic), which is non-real time traffic and delay is allowed. In order to reduce for Multimedia traffic, higher power level (larger radius) is applied by any node involving receiving and sending packets of this kind of traffic. Consequently, the shortest path is found to guarantee

multimedia application. We have done simulation based on the parameters on Table 5.1, but in this simulation number of nodes varies from 5 to 50. The results of simulation are shown in Figure 5.1 and Figure 5.2.

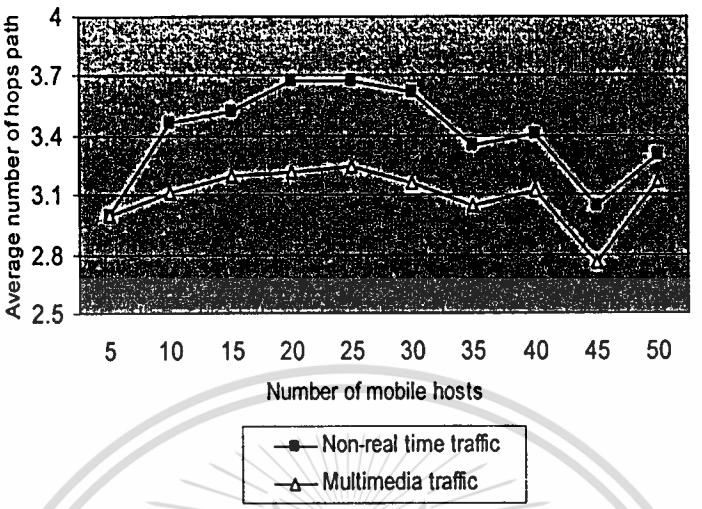


Figure 5.1: Average number of hops versus the number of mobile nodes

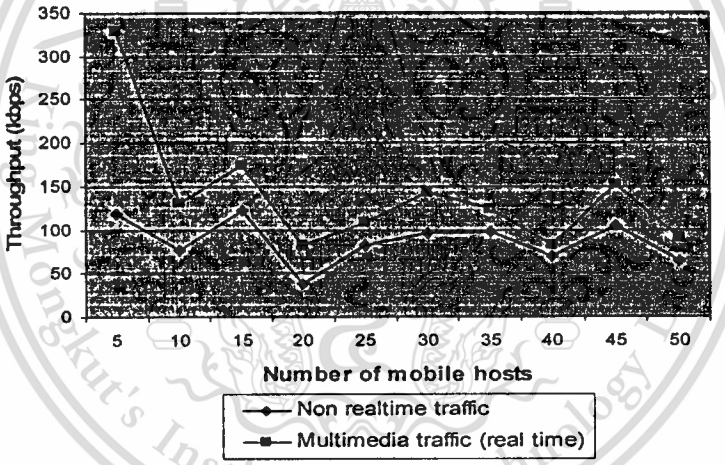


Figure 5.2: Throughput versus number of mobile nodes

As shown in Figure 5.1, we can show that average of hop path for multimedia traffic is always smaller than the one in non-real time traffic (datagram traffic). Therefore the throughput for multimedia traffic is also higher than Non-realtime traffic (datagram traffic) that is illustrated in Figure 5.2.

5.3.2 Results of MDSR

As described in Chapter 4, this algorithm uses not only power level, but also bandwidth reservation mechanism in order to guarantee multimedia traffic. We have done simulation to analyze the performance of MDSR for both Quality of Services, QoS1 and QoS2 that require

one slot and two slots respectively. We have done performance of average delay (section 5.3.2.1) and throughput (section 5.3.2.2).

5.3.2.1 Average delay

The representation of traffics in these simulation results are the same as in section 5.3.1. Figure 5.3 shows that average delay for Quality of Service for Multimedia traffic requiring data slot 1, is lower than the delay of datagram traffic requiring the same data slot (1 data slot). Moreover Figure 5.4 shows the average delay for Quality of Service for Multimedia traffic requiring data slots 2, is lower than the delay of datagram traffic which requires two data slots.

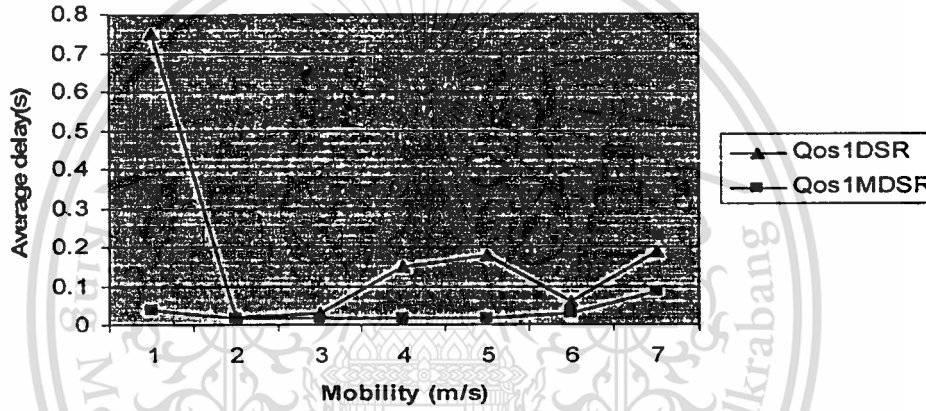


Figure 5.3: Average delay for QoS1 versus mobility of mobile nodes

In summery, our proposed scheme can provide Quality of Service for multimedia traffic the higher throughput and lower delay than the datagram traffic for all connections that requires both one data slot and two data slots.

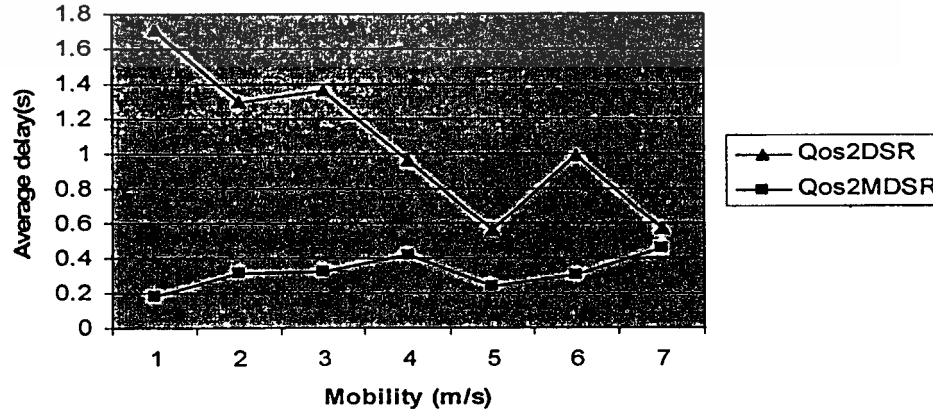


Figure 5.4: Average delay for QoS2 versus mobility of mobile nodes

5.3.2.2 Throughput

As shown in Figure 5.5, the curve QoS1MDSR represents Quality of Service of Multimedia traffic requiring 1 data slot and the curve QoS1 DSR represents Quality of Service datagram traffic requiring 1 data slot. This figure shows that the throughput of multimedia traffic is higher than the throughput of datagram traffic.

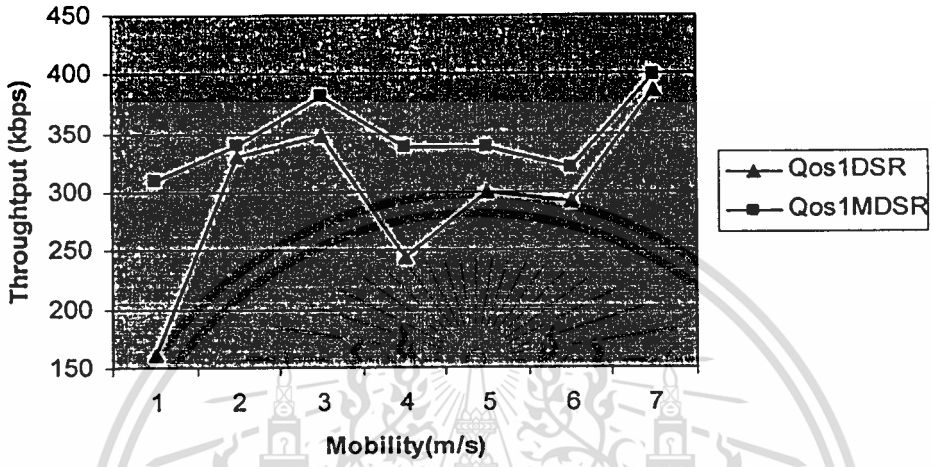


Figure 5.5: Throughput of QoS1 versus mobility of mobile nodes

Moreover, in Figure 5.6, the curve QoS2MDSR represents Quality of Service of Multimedia traffic requiring 2 data slot and the curve QoS2DSR represents Quality of Service datagram traffic requiring 2 data slot. This figure shows that the throughput of multimedia traffic is higher than the throughput of datagram traffic.

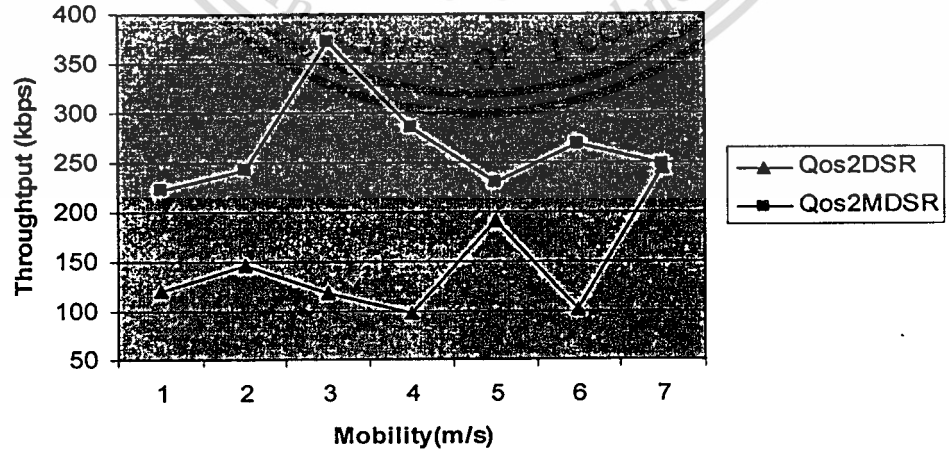


Figure 5.6: Throughput of QoS2 versus mobility of mobile nodes

Every connection can be established only if slots of every node in the path of the connection must be reserved. In the case that there are many connections in Ad-Hoc network, it is likely to be difficult to find the shortest bandwidth path for a connection that requires higher bandwidth since mobile node in the network has allocated some of its slots for other connections.

When a higher bandwidth connection is established so that data packets can be flowed, and in any circumstances that connection is broken then lots of packets residing on any intermediate node can be lost as it is much more difficult to find a new route to destination to satisfy that higher bandwidth connection. In a period of time, if the same input for both multimedia traffic requiring one and two data slots, the throughput of multimedia traffic requiring one data slot is likely to be higher than the throughput of multimedia traffic requiring two data slots. For example in the simulation result of Figure 5.5 and figure 5.6, Qos1MDSR representing the throughput of multimedia traffic requiring 1 data slots is higher than the one requiring 2 data slots (Qos2MDSR).

Chapter 6

Conclusions and Future work

6.1. Conclusions

Mobile Ad Hoc composes of mobile nodes equipped with wireless network interface 802.11x and these mobile nodes communicate to each other through radio channel, which can be easily interfered by the environment. Furthermore, these mobile nodes can also be moved freely and randomly. Therefore the available route may fail at any time so that Multimedia traffic, which is real time application or delay-sensitive traffic, might have difficulty in transmitting from end-to-end nodes. In order to guarantee this traffic to flow smoothly, we have proposed a routing scheme called "Quality of Service Based Multimedia Service Routing in Ad Hoc", which composed algorithms of power level, and bandwidth calculation and bandwidth reservation mechanism. In this proposed scheme, we have considered a shortest-bandwidth path for transmitting multimedia packets.

In this routing scheme, traffic in MANET is classified into two groups, Multimedia traffic and datagram traffic. In order to differentiate these two traffics, Different power levels are applied.

During Route Discovery process (1) Multimedia traffic, if a node broadcast RREQ for this traffic, then that node uses larger power level. In other word, the number of nodes in its coverage area can also be increased then it is possible for this node select a node which is at the edge of its coverage area. Any node receives this route request packet also used larger power level. Note that bandwidth information of every node is embedded in RREQ so that bandwidth can be calculated hop by hop from source to destination. If there is no enough bandwidth, the RREQ is dropped or the call is rejected. If any connection can be established, the route of that connection must be the shortest bandwidth path. (2) Datagram traffic or text application, when a node broadcasts RREQ for this kind of traffic, smaller power level is used. In other word, the number of node in its coverage area can be also decreased. Bandwidth information of every node is also embedded in RREQ packet then bandwidth can be calculated. If no bandwidth available then the call is rejected.

We have shown that throughput of Quality of service of Multimedia traffic is higher that throughput quality of service of datagram traffic or text application. Furthermore, the delay of multimedia traffic is lower than the delay of datagram traffic or text application.

6.2. Future works

Look at Figure 6.1, suppose a connection has already been established from the source node A to node C. Each node has 8 slots (slot 0 to 7). As shown in Figure 6.1, Node A uses slot number 0 and slot number 1 to transmit packets to node B. Node B uses its slot number 2 and slot number 3 to forward packets to node C.

While node B is in process of forwarding data packets to node C by using its slot 2 and slot3, Node D MUST not be able to transmit any packets to node E by using its slot 2 and slot 3 because node D overhears data transmission from node B.

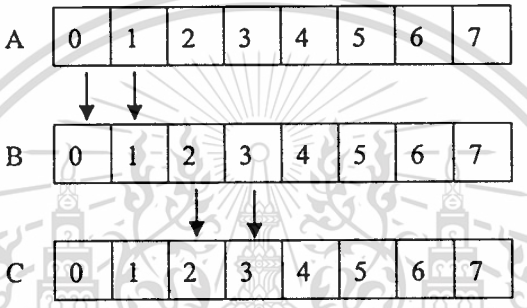


Figure 6.1: Bandwidth assignment from node A to node C

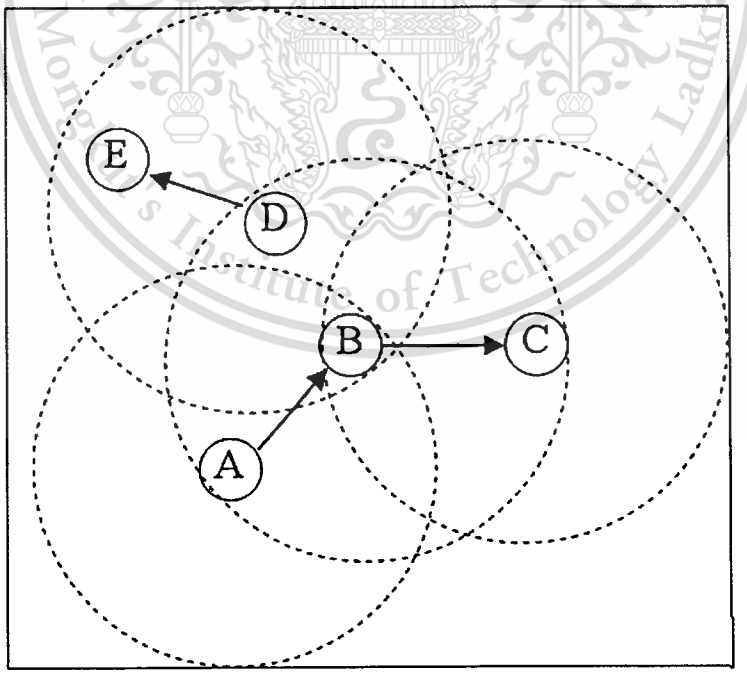


Figure 6.2: Overheard of bandwidth information

In the future research, we will improved our routing scheme by creating a table for every node in the network to keep record of slots being used to transmit packets by nodes in its coverage area. Therefore, the case above can be solved; consequently, the performance of this routing scheme can also be improved.

REFERENCES

[1] David B. Johnson, David A Maltz and Yin-Chun Hu, **"The Dynamic Source Routing Protocol for Mobile Adhoc Network,"** IETF internet draft 19 July, 2004.

[2] Charles Perkins, **"Ad hoc On-Demand Distance Vector (AODV) Routing,"** IETF Internet draft July 2003.

[3] George Aggelou,**"On the performance Analysis of the Minimum- Blocking and Bandwidth-Reallocation Channel-Assignment (MBCA/BRCA) Methods for Quality-of-Service Rouitng Support in Mobile Multimedia Ad Hoc Networks"**, IEEE Transactions on, Volume: 53, Issue: 3, May2004, Pages:770 – 782.

[4] Prasant Mohapatra, Jian Li, and Chao Gui, **"QoS in Mobile Ad hoc Network,"** Special Issue on QoS in Next-Generation Wireless Multimedia Communicati-ons Systems in IEEE Wireless Communications Magazine, June 2003.

[5] N. Bambos and G. J. Pottie, **"On power control in high capacity cellular radio networks,"** in Proc. IEEE GLOBECOM'92, pp. 863–867.

[6] Turgut D., Das S.K., Elmasri, R., Turgut B., **"Optimizing clustering algorithm in mobile ad hoc networks using genetic algorithmic approach,"** Global Telecommunications Conference, 2002. GLOBECOM '02. IEEE, Volume: 1, 17-21 Nov. 2002 Pages: 62 – 66.

[7] L. Hu, **"Distributed code assignments for CDMA packet radio networks,"**IEEE/ACM Trans. Networking, pp. 668–677, Dec. 1993.

[8] Ying-Hong Wang, Hung-Zu Lin, Shu-Min Chang, **"Interfering-aware QoS multipath routing for ad hoc wireless network,"**Advanced Information Networking and Applications, 2004 AINA2004. 18th International Conference, Volume: 1, Pages: 29-34, 2004.

[9] Barolli L, Koyama A, Shiratori N, **"A QoS routing method for ad-hoc networks based on genetic algorithm,"**Database and Expert Systems Applications, 2003. Proceedings. 14th International Workshop on, 1-5 Sept. 2003, Pages: 175–179.

[10] Mehran Abolhasan, Tadeusz Wysocki and Eryk Dutkiewicz, **"A review of routing protocols for mobile ad hoc networks,"**Ad Hoc Networks, Volume 2, Issue 1, January 2004, Pages 1-22.

[11] Aggelou G.N., Tafazolli, R.,"**QoS support in 4th generation mobile multimedia ad hoc networks,**" 3G Mobile Communication Technologies, 2001. Second International Conference on (Conf. Publ. No. 477), 26-28 March 2001, Pages: 412– 416.

[12] Shigang Chen, Nahrstedt K, "**Distributed quality-of-service routing in ad hoc networks,**" Selected Areas in Communications, IEEE Journal on , Volume: 17, Issue: 8 , Aug. 1999 Pages:1488 – 1505.

[13] Chunhung Richard Lin; Jain-Shing Liu, "**QoS routing in ad hoc wireless networks,**"Selected Areas in Communications, IEEE Journal on Volume 17, Issue 8, Aug. 1999 Page(s):1426 - 1438

[14] Nguon Taing, Sakchai Thipchaksurat, Ruttikorn Varakulsiripunth, Hiroshi Ishii, **"Routing Scheme for Multimedia Services in Mobile Ad Hoc Network,"** Fifth International Conference on Information Communications and Signal Processing.

[15] http://www.isi.edu/nsnam/ns/

[16] Charles E. Perkins and Pravin Bhagwat,"**Highly Dynamic Destination Sequenced Distance-vector Routing (DSDV) for mobile computers,**" in ACM SIGCOMM'94 Conference on Communications Architectures, Protocols and Applications, 1994, pp. 234-244.

[17] Jacquet, P.; Muhlethaler, P.; Clausen, T.; Laouiti, A.; Qayyum, A.; Viennot, **"Optimized link state routing protocol for ad hoc networks,"** in the proceedings of Multi Topic Conference, IEEE INMIC December 2001. pp. 62 – 68.

[18] Zygmund J. Haas and Marc R. Pearlman, **"Determining the optimal con_guration for the zone routing protocol,"** IEEE Journal on Selected Areas Communications, vol. 17, no. 8, Aug. 1999.

[19] Elizabeth M. Royer, C-K.Toh, **"A review of current routing protocols for ad hoc mobile wireless networks",** IEEE Personal Communication, volume 6, pages 46 - 55, April 1999.

Appendix

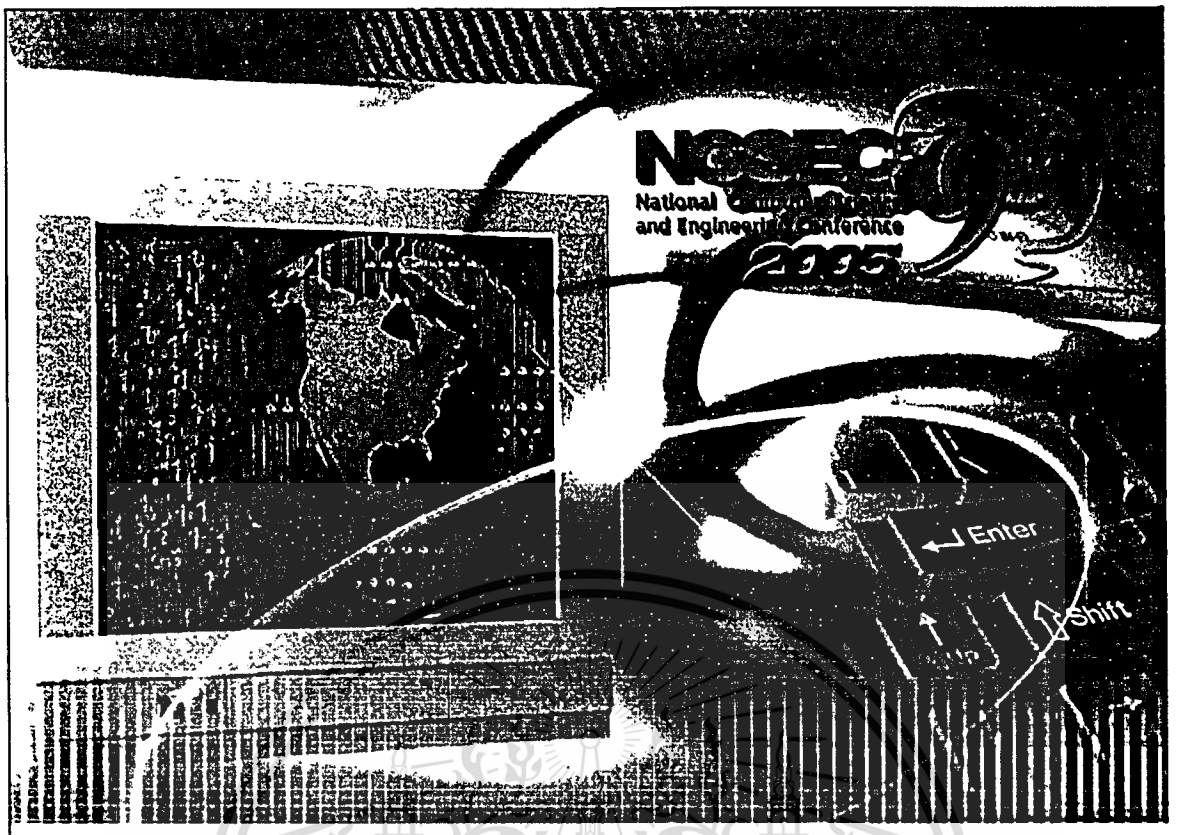
Appendix A

List of Publications

[1] N. Taing, S. Thipchaksurat, R. Varakulsiripunth, and H. Ishii, **"Modification of Dynamic Source Routing based on Signal Strength for Multimedia Traffic in Mobile Ad-hoc Network,"** The 9th National Computer Science and Engineering Conference (NCSEC 2005), page 101-108, October 2005, Bangkok Thailand .

[2] N. Taing, S. Thipchaksurat, R. Varakulsiripunth, and H. Ishii, **"Routing Scheme for Multimedia Services in Mobile Ad Hoc Network,"** Proceeding of the Fifth International Conference on Information Communications and Signal Processing, December 2005, Bangkok, Thailand, page 11-15.

[3] N. Taing, S. Thipchaksurat, R. Varakulsiripunth, and H. Ishii, **"Performance Improvement of Dynamic Source Routing Protocol for Multimedia Services in Mobile Ad Hoc Networks,"** IEEE conference on International Symposium on Wireless Pervasive Computing (ISWPC 2006), January 2006, Phuket Thailand.



The 9th National Computer Science and Engineering Conference

October 27-28, 2005

University of Thai Chamber of Commerce, Bangkok Thailand

Organized by:

Department of Computer Engineering, School of Engineering,
University of Thai Chamber of Commerce

In Cooperation with:

Electrical Engineering; Electronics, Computer, Telecommunications and Information Technology Association of Thailand (ECTI)

IEEE Communications Society, Thailand Chapter

Sponsored by:

University of Thai Chamber of Commerce

National Electronics and Computer Technology Center (NECTEC)

Sun Microsystems (Thailand)

CS Loxinfo Public Company Limited

Pearson Education Indochina Limited

The OGA Group

▷ Modification of Dynamic Source Routing based on Signal Strength for Multimedia Traffic in Mobile Ad-hoc Network

Nguon TAING*, Sakchai THIPCHAKSURAT*,
Ruttikorn VARAKULSIRIPUNTH*, Hiroshi ISHII**
*Faculty of Engineering and Research Center for Communications and Information Technology (ReCCITT), King Mongkut's Institute of Technology Ladkrabang, Chalongkrung Road, Ladkrabang, Bangkok, Thailand 10520
**Department of Communication Engineering, School of Information Technology and Electronics, Tokai University, Japan
Email: romdoul2002@yahoo.com, {ktsakcha, kvruttik} @kmitl.ac.th

Modification of Dynamic Source Routing based on Signal Strength for Multimedia Traffic in Mobile Ad-hoc Network

Nguon TAING*, Sakchai THIPCHAKSURAT*, Ruttikorn VARAKULSIRIPUNTH*, Hiroshi ISHII**
*Faculty of Engineering and Research Center for Communications and Information Technology (ReCCITT), King Mongkut's Institute of Technology Ladkrabang, Chalongkrung Road, Ladkrabang, Bangkok, Thailand 10520
**Department of Communication Engineering, School of Information Technology and Electronics, Tokai University, Japan
Email: romdoul2002@yahoo.com, {ktsakcha, kvruttik} @kmitl.ac.th

Abstract

In Mobile Ad hoc Network (MANET), all nodes communicate with one another through radio links without cable and can move freely that make MANET to be an infrastructureless network. Since the network topology may change constantly and the available routes may fail many times, Quality-of-Service (QoS) of real-time multimedia application may have difficulty in satisfying their services. In this paper, we propose a Modification of Dynamic Source Routing (MDSR) based on Signal Strength for Multimedia services, which selects the shortest path by using power level. We show by means of the simulation that MDSR provides the lower average number of hops from source node to destination node resulting in the lower delay comparing with those of DSR.

Keywords: Routing Protocol, Multimedia, Mobile Ad hoc Network

1. Introduction

Mobile Ad Hoc Network (MANET) is a collection of wireless mobile nodes in which each node communicates with each other through shared, limited radio channel in peer to peer fashion that make MANET to be an infrastructures-less networking as shown in Figure 1. Each mobile node or intermediate node in the network operates as routers forwarding packet to other nodes in order to establish two pairs of communication nodes.

MANET can be categorized into flat architecture as shown in Figure 1and hierarchical architecture [3, 6], called cluster as shown in Figure 2. In flat architecture, all nodes have the same rank, but in hierarchical architecture, nodes are separated into masters and slaves. Master nodes, cluster heads are selected among the nodes in the network and the remaining nodes, slave nodes, register themselves in one cluster of more cluster heads. A cluster head is responsible for communicating with its registered cluster nodes and communicating with other cluster heads in the network. By contrast, Slave nodes in each cluster communicate with each other via their respective master nodes.

Because MANET does not rely on wired backbone, based station and centralized administration, the growing demands of roaming users, and the availability of wireless devices, MANET has been received an increased attention for commercial purpose especially for small area wireless networks (home office, building, organization conferences..etc..). Moreover military communication and disaster recovery operation in the area where there is no based station or access point, MANET is useful in those environments.

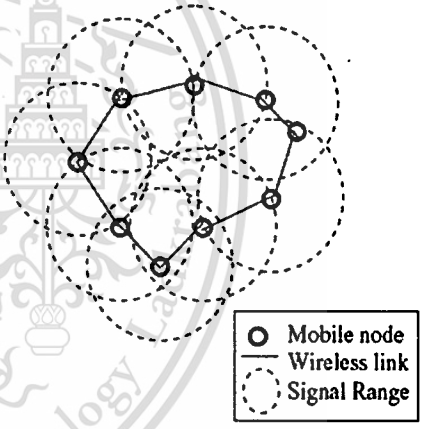


Figure 1: Flat architecture of MANET

In MANET; the nodes are mobile, the topology of the network may change rapidly and unexpectedly. Therefore the packets transferring from end-to-end may be lost as the result of broken link of the availability routes because of deletions of nodes, contention or interfering so Quality of Services (QoS) have been done by many researchers [1,2,5] including QoS starting from physical layer up to application layer [4], QoS Medium Access Control

(MAC), Genetic Algorithm QoS Routing Method [9], and QoS of Routing [7, 8, 10,11].

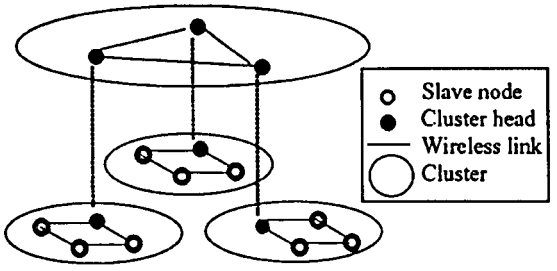


Figure 2: Cluster architecture of

Multimedia stream and some real time application are delayed sensitive such as live voice and video conferencing. In order to guarantee the multimedia stream, so in this paper we propose Modification of Dynamic Source Routing (MDSR) based on Signal Strength for Multimedia services which select the shortest path by using the largest power level which decreases the number of hops of the route. The rest of the paper is organized as follows: In section 2, we briefly present routing protocols that have been proposed, Section 3, related work. In section 4, we describe our method, Modification of Dynamic Source Routing (MDSR) based on Signal Strength for Multimedia services. Simulation and results are given in section 5 and finally conclusion in section 6.

2. Current Routing protocols

Due to the limited and shared resources in MANET, Several routing protocols have been developed or proposed in order to adapt to the nature changes of wireless network such as nodes are mobile, routes change frequently, and the topology is unpredictable. These routing protocols [10] can be classified into three different groups: global/proactive, on-demand/ reactive and hybrid. Proactive routing protocols find routes in the own maintaining route table in each node while the reactive routing protocols find route only one node need to send data to the other node. Hybrid routing protocols combine both proactive and reactive routing protocols together. That are; proactive is used in a cluster called zone whereas reactive is used to find nodes out size. This kind of routing protocol is used in hierarchical architecture.

2.1. Proactive routing protocols

Proactive routing protocols maintain information in a table on each node about the routing to the other node in the network. Although the topology of the network does not changed, this information must be updated periodically. Many proactive routing protocols has been proposed; for example; Destination Sequence Distance Vector (DSDV), Wireless Routing Protocol (WRP), Global State Routing (GSR), and Optimized Linked State Routing (OLSR) and so on. These protocols have some differences in term of maintaining updating routing information in the table [10]. There are pros and cons in the proactive routing protocol. The advantage is that that when a node needs to send or forward data packets route can be immediately used and the disadvantage is that this protocol allows significantly overhead and consumes bandwidth in the network.

2.2. Reactive protocols

Unlike proactive routing protocols, Reactive routing protocols creates routes once a node wants to transmit data to a destination. The source node initiates route discovery process by flooding route query with in the network. When the destination is reached, route reply request will be sent back to the source. Once the route has been found, it is maintained until either destination becomes inaccessible or the route is no longer desired then route discovery process will be invoked again. Several reactive protocols have been proposed such as Dynamic Source Routing protocol (DSR), Ad hoc On-demand Distance Vector (AODV), Light-weight Mobile Routing (LMR), Temporary Ordered Routing Algorithm (TORA), Relative Distance Micro-discovery Ad hoc Routing (RDMAR) [10] and so on. Those protocols can be classified into two groups, source routing and hop-by-hop routing [10]. The advantage of reactive routing protocols is to overcome the overheads, occurred in proactive routing protocols.

2.3. Hybrid routing protocols

When wireless network becomes larger, it is complex to design routing protocols. All nodes in the network are separated into groups, called cluster. All clusters form a hierarchical infrastructure. In such network, Hybrid routing protocols combining proactive and reactive routing protocols, are used in order to take advantages on these two routing protocols where proactive maintains route in a cluster and reactive maintains route between clusters. Several Hybrids routing protocols have been proposed such as Zone Routing Protocol (ZRP), Zone-based Hierarchical Link State (ZHLS), Distributed Dynamic Routing (DDR) and so on, but the most popular protocol is ZRP.

As briefly described all routing protocols above, the routing of MANET can be classified as proactive routing protocols, re-active routing protocol and Hybrid routing protocol. These routing protocols have pros and cons depending on network environments and these routing protocols do not consider or differentiate real-time traffic and non-real time traffic so we propose Modification of Dynamic Source Routing (MDSR) based on Signal

Strength for Multimedia services which selects the shortest path using higher power level and the algorithm is described in the section 4. The result of this proposed scheme, the path found must be shorter.

3. Related works

Dynamic Source Routing (DSR), one of the most popular on-demand routing protocols, proposed by David A.Maltz, David B.Johnson and Yin-Chun Hu. The purpose of DSR is to reduce overhead and to provide highly reactive services to help ensure successful delivery of data packets to reach a destination even though topology of the network is changed or there is another condition in the network.

The two common mechanisms of DSR are:

+ Route Discovery: when a node, source node, wishing to send data packet to destination node. Route Discovery will be processed only the source node does not have a route to the destination.

+ Route Maintenance: when a route being used to send data packet from source node to destination is broken, the source node invokes Route Maintenance to find a new route for the rest of the packets on the source node to send to the destination node.

Both mechanisms of DSR operate on-demand that is different from other routing protocols, which use periodic routing packet that causes overhead in the network. Moreover, in DSR, a node may learn and cache multiple routes to any destination. This caching of multiple of routes avoids the overhead of performing a new Route Discovery.

4. Modified Dynamic Source Routing (MDSR) protocol

In MANET, data transmissions should be separated into two groups. The first group is priority data transmission, real-time data stream which is delay-sensitive such as on-demand multimedia stream, video conference .etc... and the second group is normal data stream which allows the delay in during transmission. We apply our algorithm differently for these data transmissions. When a multimedia traffic is applied and in order to reduce query response time, our algorithm uses larger transmission power level or in other words, the number of nodes in the range of the source is increased. Therefore source node can select one node which is farthest to forward route request. On the other hand, for non-real time traffic which delays is not sensitive, smaller transmission power level is used. In Figure 5, an example of the proposed routing algorithm, we can see that the dotted line(S-A-B-D) representing a multimedia transmission has shorter path than the route used for non-real time data transmission which is represented by solid line(S-M-N-O-Q-D). As multimedia stream is on-demand and in order to adapt to the dynamic topologies of MANET, DSR[1] is suitable for this data traffic. The process of a source mobile host that wants to send data packet, one mechanism in MDSR, can be described in Figure 3.

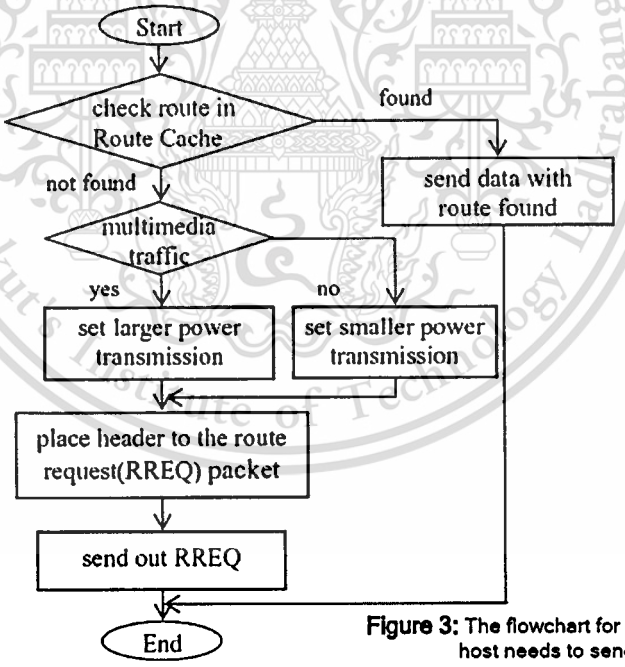


Figure 3: The flowchart for a source mobile host needs to send data.

From Figure 3, we can explain the detail of the process of a source mobile host that wishes to send data packets as follows:

1. Check route in the Route Cache previously learned whether there is route to the destination to satisfy the traffic. If found, the source node uses it
2. Otherwise and if it is real-time traffic, the node sets higher power strength, Else set lower power strength.
3. In our routing protocol, a packet has the following fields: packet-type, source-add, dest-add, sequence#, route-list, traffic_type, data, TTL. It places header to route request packet such as source node address, destination address, sequence number and traffic type to differentiate traffic in the network.
4. Finally, the source node floods route request (RREQ) packet to its neighbors.

When the source mobile host sends route request (RREQ) to its neighbors, which can be intermediate mobile host or destination mobile host. Another mechanism in MDSR for those mobile hosts which receive route request can be described in Figure 4.

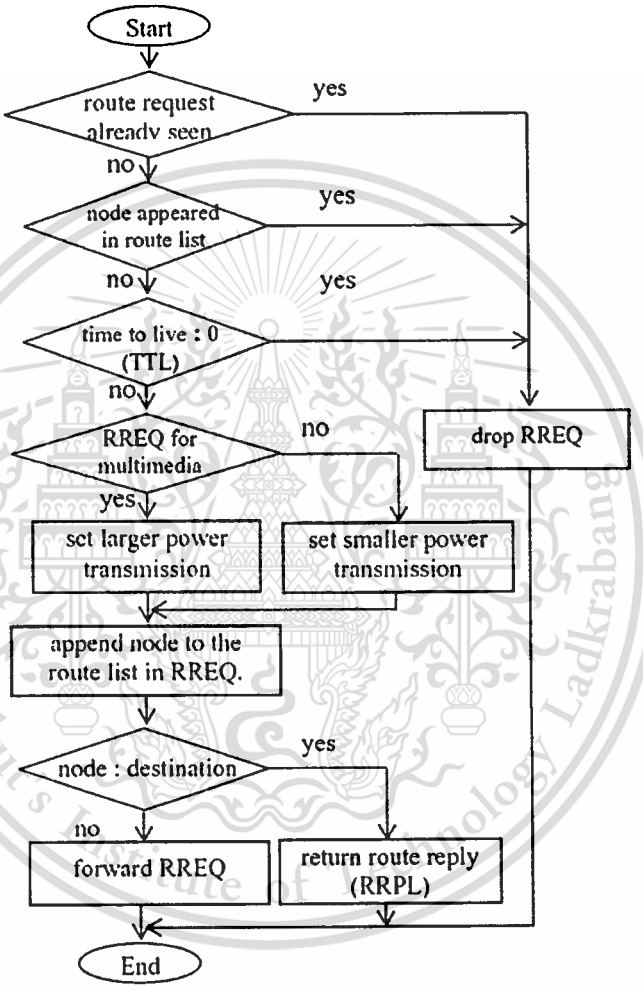


Figure 4: The flowchart for an intermediate mobile host or a destination mobile host receiving route request.

From Figure 4, we can explain the detail of the process of an intermediate mobile host or a destination mobile host as following steps:

1. If the pair <dest-addr, sequence#> for this RREQ is seen recently, discard this packet and do not process it further.
2. If the address of this node appeared in the route-list in the RREQ, we drop this RREQ (do not re-broadcast it) and do not process it further.
3. Check whether the route request is searching for the path for multimedia traffic or normal data traffic. If multimedia traffic, the node set higher power level strength otherwise set lower power strength.
4. Decrement TTL by one. If TTL counts down to zero, we drop this RREQ and do not process it further. TTL can limit the length of the delivery path. However, this path will be difficult to be maintained within a dynamic environment. In addition, unlimited packet flooding will deteriorate the network performance. The use of TTL can control the flooding traffic.
5. Append the address of this node to the route-list to track the route which the packet has traversed If this node is destination, send route reply back to the source otherwise re-broadcast this route request.

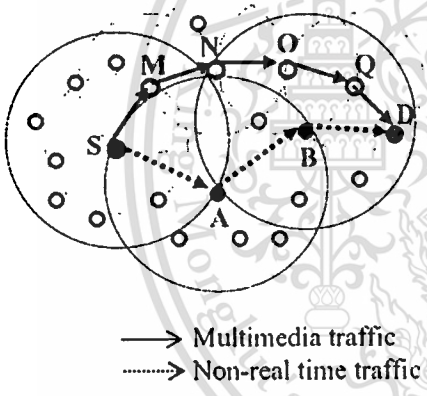


Figure 5: Example of proposed algorithm

5. Simulation and Results

5.1 Model and parameters

We use network simulator called ns2 to analyze the performance of the proposed scheme. Our network consists of 5 nodes to 50 nodes that move randomly at maximum speed 2m/s over an area 700x700 meters. The details of simulator parameters are shown in Table 1. We assume that packets can be lost or corrupted in transmission and a mobile host receiving these corrupted packets can detect the error and discard them.

Table 1: Simulation Parameters

Parameters		Values
Simulation Area		700 x 700
Number of nodes		5 10 15 20 25 30 35 40 45 50
Speed of nodes		0-2 m/s
Node movement		randomly
Transmission range	Real-time	300 meters
	Non real-time	250 meters

5.2 Simulation result

The result of simulation are shown in Figure 6, we can notice that for multimedia transmission, the mean number of routing hops is always smaller than non-real time, that is, the route query delay time is reduced. Because we use a higher transmission power level, the coverage area of each mobile node is extended. Thus, it makes possible to reduce average number of routing hops. For example; as the number of mobile nodes is increased from 5 to 50, mean number of routing hops used for real-time is smaller than mean number of routing hops.used by non-real time

As shown on the simulation result, we see that when the number of nodes is increased, it does not mean that the average number of hops path is increased because it depends on the topologies of the network and the characteristic of DSR routing protocols. Anyway, from point to point of number of nodes in the networks, we see that the average number of nodes path for multimedia traffic is smaller than the average number of nodes path for non-real time traffic because for multimedia data transmission, larger power level is used, so the number of mobile hosts in the coverage area is also increased. Consequently it makes possibility for routing algorithm to select one mobile host which is near the edge of its coverage area to forward multimedia data packets.

In summary, for non real-time traffic, since it is not sensitive to delay, with smaller power level, we can get several advantages, that is, low probability of failure and low power consumption. For multimedia traffic, we can reduce the route query delay time in place of high power consumption.

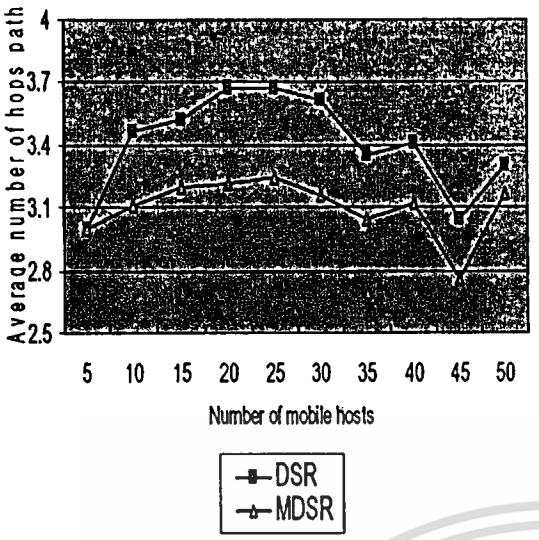


Figure 6: The average number of hops versus the number of mobile nodes for multimedia traffic

6. Conclusions

We proposed a Modification of Dynamic Source Routing (MDSR) based on signal strength to support real-time traffic such as multimedia or video stream. This proposed scheme selects a shortest path for multimedia traffic by applying larger power level as the delay is sensitive for such kind of traffic. On the other hand, for non-real time traffic, this algorithm will use smaller power level longer path for non-real time. We have shown that MDSR provides the lower average number of hops for multimedia traffic than the one of DSR. As the result, the transmission delay of multimedia traffic can be decreased.

7. Future works:

In the future works, we are going to consider the bandwidth calculation during route query request. Therefore the nodes in the routes found for both data traffic have available bandwidth to guarantee or transfer real-time data packet.

8. References

[1] David B. Johnson, David A Maltz and Yin-Chun Hu, "The Dynamic Source Routing Protocol for Mobile Adhoc Network", *IETF internet draft 19 July, 2004.*

[2] Charles Perkins, "Ad hoc On-Demand Distance Vector (AODV) Routing", *IETF Internet draft July 2003.*

[3] George Aggelou, "On the performance Analysis of the Minimum- Blocking and Bandwidth-Reallocation Channel-Assignment (MBCA/BRCA) Methods for Quality-of-Service Routing Support in Mobile Multimedia Ad Hoc Networks", *IEEE Transactions on, Volume: 53, Issue: 3, May2004, Pages:770 – 782.*

[4] Prasant Mohapatra, Jian Li, and Chao Gui, "QoS in Mobile Ad hoc Networks," *Special Issue on QoS in Next-Generation Wireless Multimedia Communications Systems* in *IEEE Wireless Communications Magazine, June 2003.*

[5] Nilufar Baghaei and Ray Hunt, "Review of quality of service performance in wireless LANs and 3G multimedia application services", *Computer Communications, Volume 27, Issue 17, 1 November 2004, Pages 1684-1692.*

[6] *Turgut D., Das S.K., Elmasri, R., Turgut B.*, "Optimizing clustering algorithm in mobile ad hoc networks using genetic algorithmic approach", *Global Telecommunications Conference, 2002: GLOBECOM '02. IEEE, Volume: 1, 17-21 Nov. 2002 Pages: 62 – 66.*

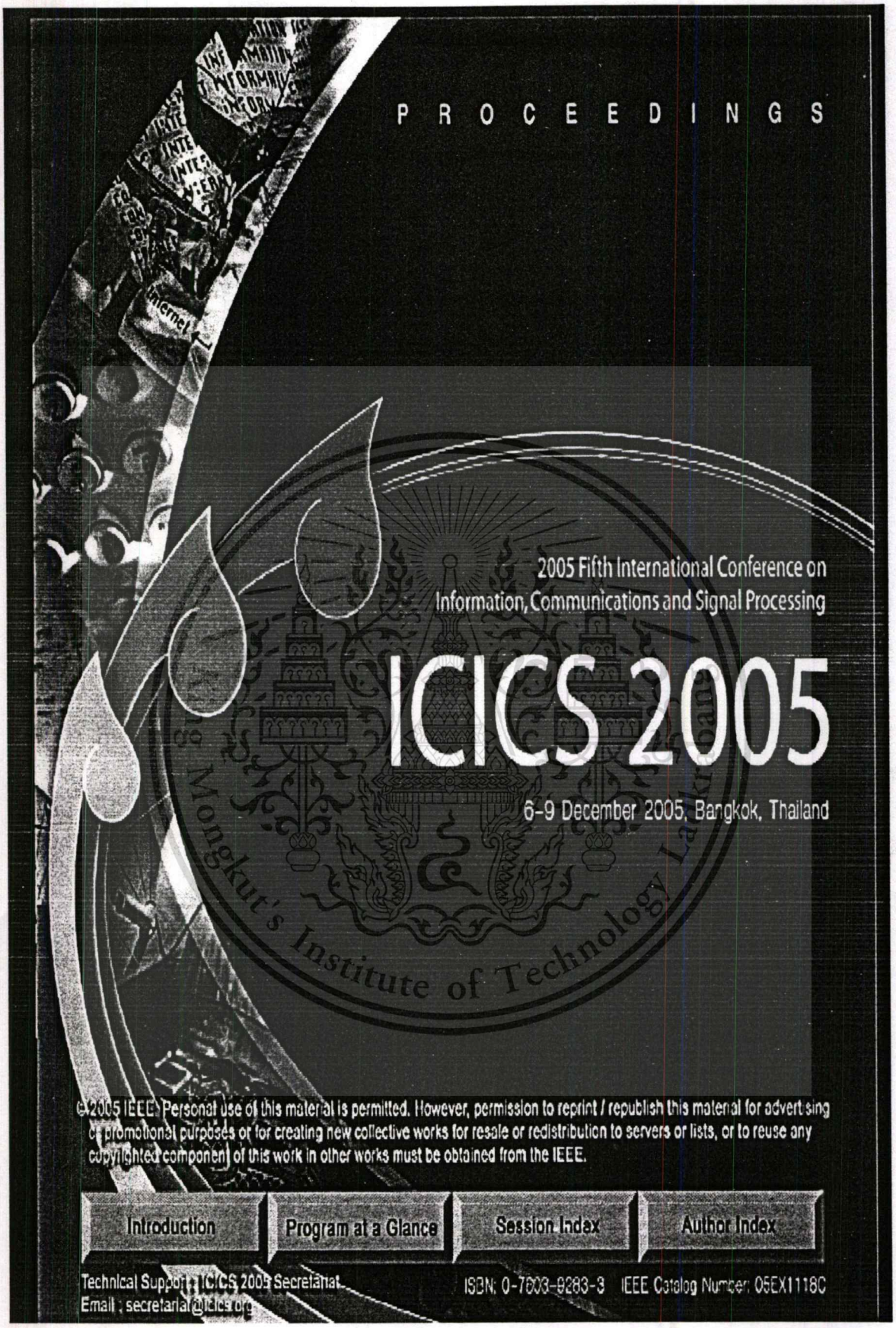
[7] Shigang Chen, Nahrstedt K. "Distributed quality-of-service routing in ad hoc networks", *Selected Areas in Communications, IEEE Journal on , Volume: 17, Issue: 8 , Aug. 1999 Pages:1488 – 1505*

[8] Ying-Hong Wang, Hung-Zu Lin, Shu-Min Chang, "Interfering-aware QoS multipath routing for ad hoc wireless network", *Advanced Information Networking and Applications, 2004 AINA2004. 18th International Conference, Volume: 1, Pages: 29-34, 2004.*

[9] Barolli L, Koyama A, Shiratori N, "A QoS routing method for ad-hoc networks based on genetic algorithm", *Database and Expert Systems Applications, 2003. Proceedings. 14th International Workshop on, 1-5 Sept. 2003, Pages: 175–179.*

[10] Mehran Abolhasan, Tadeusz Wysocki and Eryk Dutkiewicz, "A review of routing protocols for mobile ad hoc networks ", *Ad Hoc Networks, Volume 2, Issue 1, January 2004, Pages 1-22.*

[11] Aggelou G.N., Tafazolli, R.,"QoS support in 4th generation mobile multimedia ad hoc networks", *3G Mobile Communication Technologies, 2001. Second International Conference on (Conf. Publ. No. 477), 26-28 March 2001, Pages: 412– 416.*



Introduction | Program at a Glance | Session Index | Author Index

Technical Support : ICICS 2005 Secretariat
Email : secretariat@icics.org

ISBN: 0-7803-9283-3 IEEE Catalog Number: 05EX1118C

Routing Scheme for Multimedia Services in Mobile Ad Hoc Network

Nguon TAING*, Sakchai THIPCHAKSURAT*, Ruttikorn VARAKULSIRIPUNTH*, Hiroshi ISHII**
*Faculty of Engineering and Research Center for Communications and Information Technology (ReCCIT),
King Mongkut's Institute of Technology Ladkrabang (KMITL),
Chalongkrung Road, Ladkrabang, Bangkok, Thailand 10520
**Department of Communication Engineering, School of Information Technology and Electronics, Tokai University, Japan
Email: romdoul2002@yahoo.com, {ktsakcha, kvruttik}@kmitl.ac.th

***Abstract*— All communications in a network made by mobile hosts through a shared radio link without cable or backbone, commonly called Mobile Ad hoc Network (MANET). Since the network topology may change constantly because of mobility of nodes and the available routes may fail many times, Quality-of-Service (QoS) of real-time multimedia application may have difficulty in satisfying their services. In this paper, we propose a Routing Scheme for Multimedia Services, which selects the shortest path by using power level. Our proposed scheme provides not only the lower means number of hops path from source node to destination node, but also higher thoughput than the conventional scheme.**

Keywords— ***Routing Protocol, Multimedia, MANET***

I. INTRODUCTION

Mobile Ad Hoc Network (MANET) is a collection of wireless mobile nodes in which each node communicates with each other through shared, limited radio channel in peer to peer fashion that makes MANET to be an infrastructures-less networking as shown in an example in Figure 1. Each mobile node or intermediate node in the network operates as routers forwarding packet to other nodes in order to establish two pairs of communication nodes.

MANET can be categorized into flat architecture and hierarchical architecture [3, 6], called cluster. In flat architecture, all nodes have the same rank, but in hierarchical architecture, nodes are separated into masters and slaves. Master nodes, cluster heads are selected among the nodes in the network and the remaining nodes, slave nodes, register themselves in one cluster of more cluster heads. A cluster head is responsible for communicating with its registered cluster nodes and communicating with other cluster heads in the network. By contrast, slave nodes in each cluster communicate with each other via their respective master nodes.

Because MANET does not rely on wired backbone, based station and centralized administration, the growing demands of roaming users, and the availability of wireless devices, MANET has been received an increased attention for commercial purpose especially for small area wireless networks (home office, building, organization conferences, etc.). Moreover, military communication and disaster recovery operation in the area where there is no based station or access point, MANET is useful in those environments.

In MANET, the nodes are mobile hosts and the topology of the network may change rapidly and unexpectedly. Therefore, the packets transferring from end-to-end may be lost as the result of broken link of the availability routes because of deletions of nodes, contention or interfering. Quality of Services (QoS) for MANET have been done by many researchers [1, 2, 5], such as QoS starting from physical layer up to application layer [4], QoS Medium Access Control (MAC), Genetic Algorithm QoS Routing Method [9], and QoS of Routing [7,8, 10, 11].

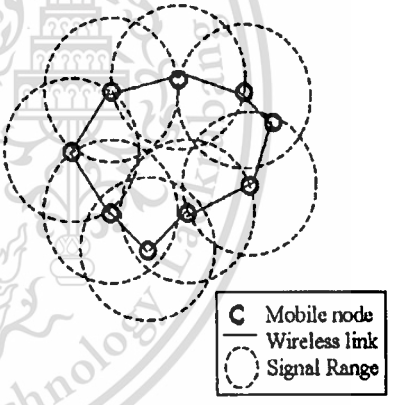


Figure 1. An Example of MANET

Multimedia stream and some real time application are delayed sensitive such as live voice and video conferencing. In this paper we propose Routing Scheme for Multimedia Services that selects the shortest path by using the larger power level in order to guarantee the multimedia stream and decrease the number of hops of the route. The rest of the paper is organized as follows. In section II, we briefly present routing protocols that have been proposed. In section III, we explain DSR protocol which is related to our scheme. In section IV, we describe our method, Routing Scheme for Multimedia Services.

Simulation and results are given in section V. Finally we conclude our paper in section VI.

II. CURRENT ROUTING PROTOCOLS

Due to the limited and shared resources in MANET, several routing protocols have been developed or proposed in order to adapt to the nature changes of wireless network such as nodes are mobile hosts, routes change frequently, and the topology is unpredictable. These routing protocols [10] can be classified into three different groups, i.e., global/proactive, on-demand/reactive and hybrid. Proactive routing protocols find routes in the own maintaining route table in each node while the reactive routing protocols find route only one node need to send data to the other node. Hybrid routing protocols combine both proactive and reactive routing protocols together. The concept is that proactive is used in a cluster called zone whereas reactive is used to find nodes out size. This kind of routing protocol is used in hierarchical architecture.

A. Proactive routing protocols

Proactive routing protocols maintain information in a table on each node about the routing to the other node in the network. Although the topology of the network does not changed, this information must be updated periodically. Many proactive routing protocols have been proposed, for example, Destination Sequence Distance Vector (DSDV), Wireless Routing Protocol (WRP), Global State Routing (GSR), and Optimized Linked State Routing (OLSR) and so on. These protocols have some differences in term of maintaining updating routing information in the table [10]. There are pros and cons in the proactive routing protocol. The advantage is that when a node needs to send or forward data packets, route can be immediately used and the disadvantage is that this protocol allows significantly overhead and consumes bandwidth in the network.

B. Reactive protocols

Unlike proactive routing protocols, the reactive routing protocols create routes once a node wants to transmit data to a destination. The source node initiates route discovery process by flooding route query with in the network. When the destination is reached, route reply request will be sent back to the source. Once the route has been found, it is maintained until either destination becomes inaccessible or the route is no longer desired then route discovery process will be invoked again. Several reactive protocols have been proposed such as Dynamic Source Routing protocol (DSR), Ad hoc On-demand Distance Vector (AODV), Light-weight Mobile Routing (LMR), Temporary Ordered Routing Algorithm (TORA), Relative Distance Micro-discovery Ad hoc Routing (RDMAR) [10] and so on. Those protocols can be classified into two groups, source routing and hop-by-hop routing [10]. The advantage of reactive routing protocols is to overcome the overheads, occurred in proactive routing protocols.

C. Hybrid routing protocols

When wireless network becomes larger, it is complex to design routing protocols. All nodes in the network are separated into groups, called cluster. All clusters form a hierarchical infrastructure. In such network, hybrid routing protocols, i.e., combining proactive and reactive routing protocols, are used in order to take advantages on these two routing protocols where proactive maintains route in a cluster and reactive maintains route between clusters. Several hybrids routing protocols have been proposed such as Zone Routing Protocol (ZRP), Zone-based Hierarchical Link State (ZHLS), Distributed Dynamic Routing (DDR) and so on, but the most popular protocol is ZRP.

As briefly described all routing protocols above, the routing of MANET can be classified as proactive routing protocols, reactive routing protocol and hybrid routing protocol. These routing protocols have pros and cons depending on network environments and these routing protocols do not consider or differentiate real-time traffic and non-real time traffic. So we propose Routing Scheme for Multimedia Services to select the shortest path using higher power level. The algorithm is described in the section III.

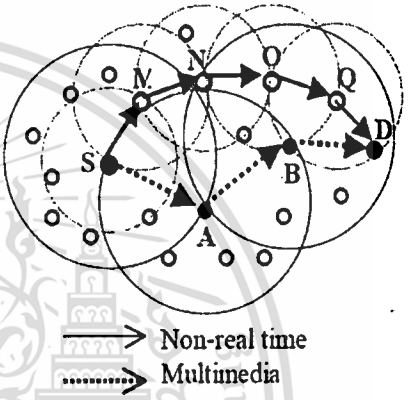


Figure 2. An example of proposed algorithm.

III. DYNAMIC SOURCE ROUTING (DSR) PROTOCOL

Dynamic Source Routing (DSR) protocol is one of the most popular on-demand routing protocols, proposed by David A.Maltz, David B.Johnson and Yin-Chun Hu. The purpose of DSR is to reduce overhead and to provide highly reactive services to help ensure successful delivery of data packets to reach a destination even though topology of the network is changed or there is another condition in the network.

The two common mechanisms of DSR protocol are:

+ Route Discovery: when a node, source node, wishing to send data packet to destination node. Route Discovery will be processed only the source node does not have a route to the destination.

+ Route Maintenance: when a route being used to send data packet from source node to destination is broken, the source node invokes Route Maintenance to find a new route for the rest of the packets on the source node to send to the destination node.

Both mechanisms of DSR protocol operate on-demand that is different from other routing protocols, which use periodic

routing packet that causes overhead in the network. Moreover, in DSR protocol, a node may learn and cache multiple routes to any destination. This caching of multiple of routes avoids the overhead of performing a new Route Discovery.

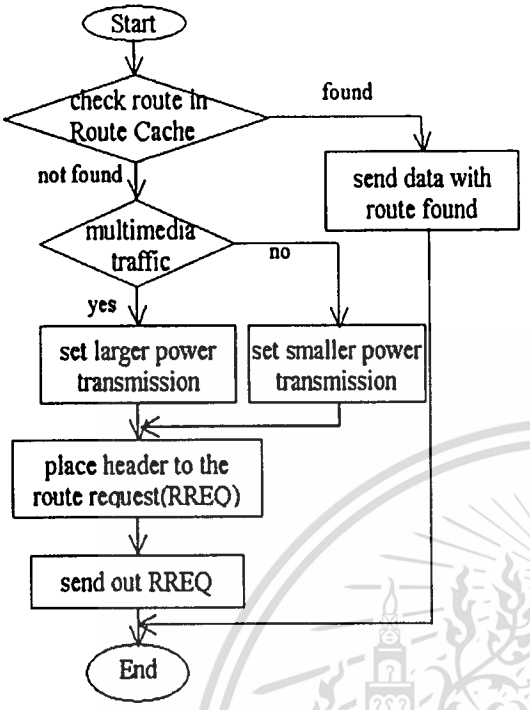


Figure 3. The flowchart for a source node needs to send data.

IV. PROPOSED ALGORITHM

In MANET, data transmissions should be separated into two groups. The first group is priority transmission of real-time data stream which is delay-sensitive data such as on-demand multimedia stream, video conference etc., and the second group is normal data stream which allows the delay in during transmission. We apply our algorithm differently for these data transmissions. When the multimedia traffic is applied and in order to reduce query response time, our algorithm uses larger transmission power level or in other words, the number of nodes in the range of the source is increased. Therefore, source node can select one fastest node to forward route request. On the other hand, for non-real time traffic which delay is not sensitive, smaller transmission power level is used. In Figure 2, an example of the proposed routing algorithm, we can see that the dotted line (S-A-B-D) representing a multimedia transmission has shorter path than the route used for non-real time data transmission that is represented by solid line (S-M-N-O-Q-D). As multimedia stream is on-demand and in order to adapt to the dynamic topologies of MANET, DSR [1] is suitable for this data traffic. The process of a source node that wants to send data packet can be described in Figure 3.

From Figure 3, we can explain the detail of the process of a source node that wishes to send data packets as follows:

- Check route in the Route Cache previously learned whether there is route to the destination to satisfy the traffic. If found, the source node uses it
- Otherwise and if it is real-time traffic, the node sets higher power strength, else set lower power strength.
- In our routing protocol, a packet has the following fields: packet-type, source-add, dest-add, sequence#, route-list, traffic type, data, TTL. It places header to route request packet such as source node address, destination address, sequence number and traffic type to differentiate traffic in the network.
- Finally, the source node floods route request (RREQ) packet to its neighbors which can be intermediate node or destination node.

In addition to the mechanism shown in Figure 3, another mechanism for nodes that receive route request can be described in Figure 4.

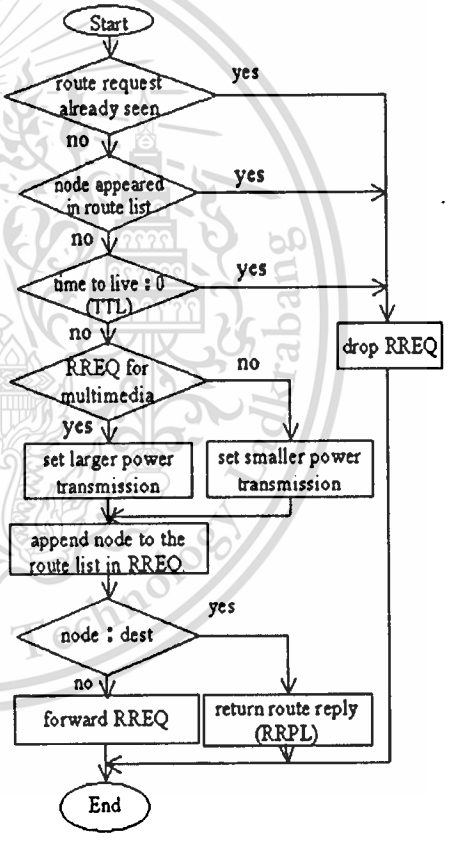


Figure 4. The flowchart for an intermediate node or a destination node receiving route request

From Figure 4, we can explain the detail of the process of an intermediate node or a destination node as following steps:

- If the pair <dest-addr, sequence#> for this RREQ is seen recently, discard this packet and do not process it further.
- If the address of this node appeared in the route-list in the RREQ, we drop this RREQ (do not re-broadcast it) and do not process it further.
- Check whether the route request is searching for the path for multimedia traffic or normal data traffic. If multimedia traffic, the node set higher power level strength otherwise set lower power strength.
- Decrement TTL by one. If TTL counts down to zero, we drop this RREQ and do not process it further. TTL can limit the length of the delivery path. However, this path will be difficult to be maintained within a dynamic environment. In addition, unlimited packet flooding will deteriorate the network performance. The use of TTL can control the flooding traffic.

Append the address of this node to the route-list to track the route which the packet has traversed If this node is destination, send route reply back to the source otherwise re-broadcast this route request.

TABLE I. SIMULATION PARAMETERS

Parameters		Values
Simulation Area		700 x 700
Number of nodes		5 10 15 20 25 30 35 40 45 50
Speed of nodes		0-2 m/s
Node movement		Randomly
Transmission range	Real-time	300 meters
	Non real-time	250 meters

V. SIMULATION AND RESULTS

A. *Model and parameters*

We use network simulator called ns2 to analyze the performance of the proposed scheme. Our network consists of 5 nodes to 50 nodes that move randomly at maximum speed 2m/s over an area 700x700 meters. Two connection were established, one for non-real time and another one for real-time or multimedia traffic. The details of simulator parameters are shown in Table I. We assume that packets can be lost or corrupted in transmission and a node receiving these corrupted packets can detect the error and discard them.

B. *Simulation result*

The result of simulation is shown in Figure 5, we can notice that for multimedia transmission, the mean number of routing hops is smaller than non-real time. Because we use a higher transmission power level, the coverage area of each mobile node is extended. Thus, it makes possible to reduce average number of routing hops. For example: as the number of mobile nodes is increased from 5 to 50, mean number of routing hops used for real-time is smaller than mean number of routing hops used by non-real time.

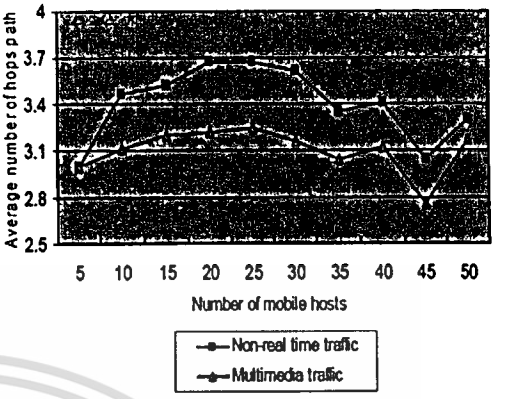


Figure 5. The average number of hops versus the number of mobile nodes

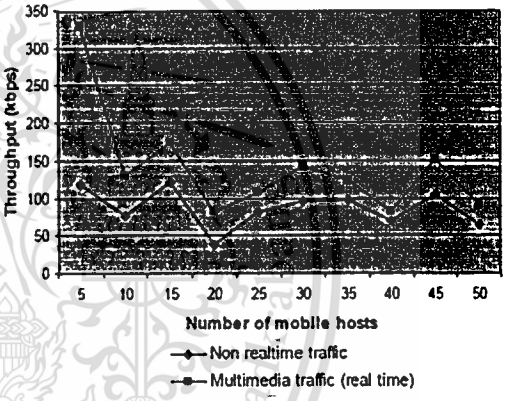


Figure 6. Throughput versuses the number of mobile nodes

Moreover as shown in Figure 6, the throughput of multimedia stream is always higher than the one of non-real-time traffic including route query request time which makes route query request for real-time faster than the route query request of non real-time traffic.

As shown in Figure 5 and 6, we can see that when the number of nodes is increased, it does not mean that the average number of hops path is increased or the throughput is higher than the smaller number of hops because it depends on the topologies of the network from time to time. Besides, a farthest node is selected to forward packets, which causes a route used to be broken easily when the node moves so that it reduces the throughput and a new route must be re-established again.

Anyway, from point to point of number of nodes in the networks, we see that the average number of nodes path for multimedia traffic is smaller than the average number of nodes path for non-real time traffic. Not only that but also the throughput of multimedia traffic is higher than non real-time traffic because for multimedia data transmission, larger power level is used, so the number of nodes in the coverage area is also increased.

Consequently it makes possibility for routing algorithm to select one node which is near the edge of its coverage area to forward multimedia data packet.

VI. CONCLUSIONS AND FUTURE WORKS

We proposed a Routing Scheme for Multimedia Services to support real-time traffic such as multimedia or video stream. This proposed scheme selects a shortest path for multimedia traffic by applying larger power level because the delay is sensitive for such kind of traffic. On the other hand, for non-real time traffic, this algorithm will use smaller power level longer path for non-real time. We can conclude that our propose schemes provides the lower mean number of hops path for multimedia traffic than the mean number of hops path for non-real time traffic. As the result, the transmission delay of multimedia traffic can be decreased. We also show that our proposed scheme can provide higher throughput for multimedia traffic.

In the future works, we are going to consider the bandwidth calculation during route query request. Therefore the nodes in the routes found for both data traffic have available bandwidth to guarantee or transfer multimedia real-time data packet.

REFERENCES

[1] David B. Johnson, David A Maltz and Yin-Chun Hu, "The Dynamic Source Routing Protocol for Mobile Adhoc Network", IETF internet draft 19 July, 2004.

[2] Charles Perkins, "Ad hoc On-Demand Distance Vector (AODV) Routing", IETF Internet draft July 2003.

[3] George Aggelou, "On the performance Analysis of the Minimum-Blocking and Bandwidth-Reallocation Channel-Assignment (MBCA/BRCA) Methods for Quality-of-Service Routing Support in Mobile Multimedia Ad Hoc Networks", IEEE Transactions on, Volume: 53, Issue: 3, May2004, Pages:770 – 782.

[4] Prasant Mohapatra, Jian Li, and Chao Gui, "QoS in Mobile Ad hoc Networks", Special Issue on QoS in Next-Generation Wireless Multimedia Communicati-ons Systems in *IEEE Wireless Communications Magazine*, June 2003.

[5] Nilufar Baghaei and Ray Hunt, "Review of quality of service performance in wireless LANs and 3G multimedia application services", Computer Communications, Volume 27, Issue 17, 1 November 2004, Pages 1684-1692.

[6] Turgut D., Das S.K., Elmasri, R., Turgut B., "Optimizing clustering algorithm in mobile ad hoc networks using genetic algorithmic approach", Global Telecommunications Conference, 2002. GLOBECOM '02. IEEE, Volume: 1, 17-21 Nov. 2002 Pages: 62 – 66.

[7] Shigang Chen, Nahrstedt K, "Distributed quality-of-service routing in ad hoc networks", Selected Areas in Communications, IEEE Journal on , Volume: 17, Issue: 8 , Aug. 1999 Pages:1488 – 1505

[8] Ying-Hong Wang, Hung-Zu Lin, Shu-Min Chang, "Interfering-aware QoS multipath routing for ad hoc wireless network", Advanced Information Networking and Applications, 2004 AINA2004. 18th International Conference, Volume: 1, Pages: 29-34, 2004.

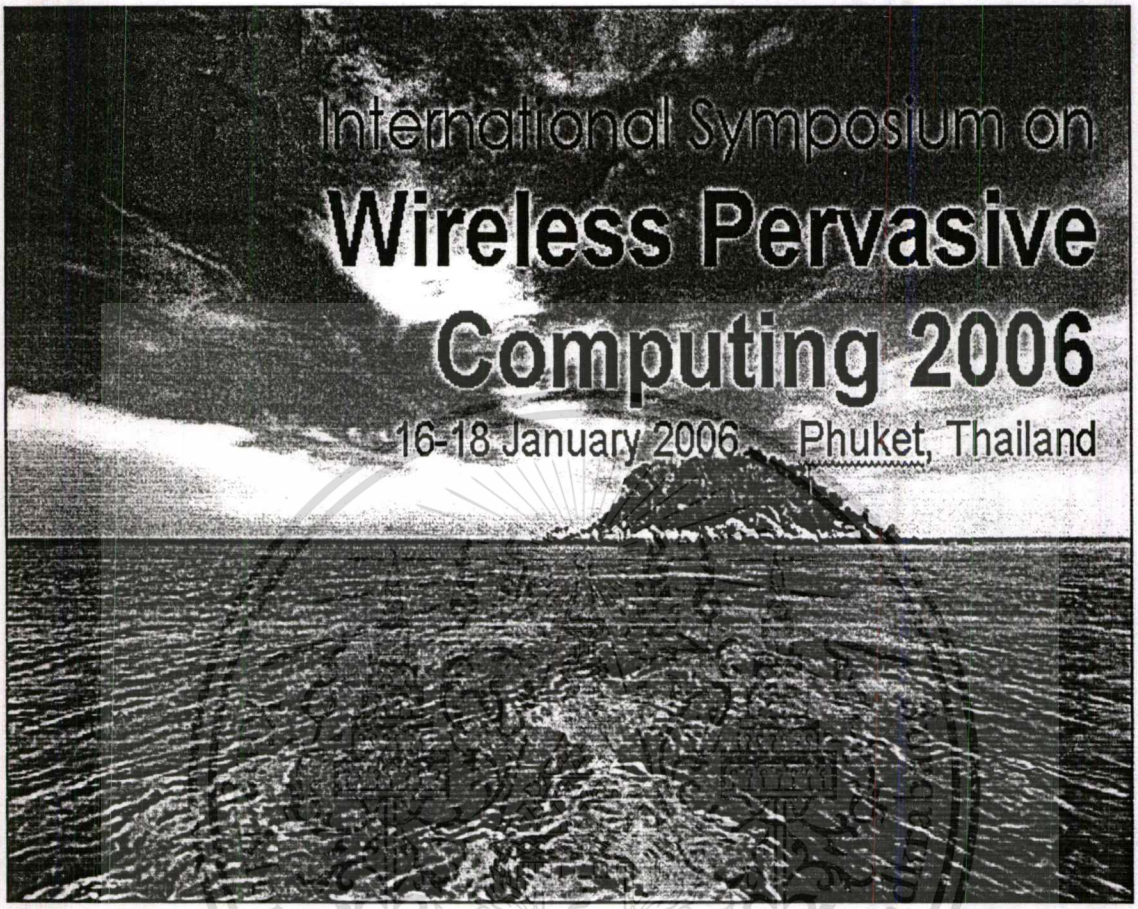
[9] Barolli L, Koyama A, Shiratori N, "A QoS routing method for ad-hoc networks based on genetic algorithm", Database and Expert Systems Applications, 2003. Proceedings. 14th International Workshop on, 1-5 Sept. 2003, Pages: 175–179.

[10] Mehran Abolhasan, Tadeusz Wysocki and Eryk Dutkiewicz, "A review of routing protocols for mobile ad hoc networks ", Ad Hoc Networks, Volume 2, Issue 1, January 2004, Pages 1-22.

[11] *Aggelou G.N., Tafazolli, R.,* "QoS support in 4th generation mobile multimedia ad hoc networks", *3G Mobile Communication Technologies, 2001. Second International Conference on (Conf. Publ. No. 477), 26-28 March 2001, Pages: 412– 416.*

IEEE

IEEE COMMUNICATIONS SOCIETY



Gold Sponsors

Siam Photon

ECTI-21 NECTEC

Performance Improvement of Dynamic Source Routing Protocol for Multimedia Services in Mobile Ad Hoc Network

Nguon TAING*, Sakchai THIPCHAKSURAT*, Ruttikorn VARAKULSIRIPUNTH*, Hiroshi ISHII**
*Faculty of Engineering and Research Center for Communications and Information Technology (ReCCIT),
King Mongkut's Institute of Technology Ladkrabang (KMITL), Chalongkrung Road, Ladkrabang, Bangkok, Thailand 10520
**Department of Communication Engineering, School of Information Technology and Electronics, Tokai University, Japan
Email: roundoul2002@yahoo.com, {ktsakcha, kvruttik}@kmitl.ac.th, ishii@dt.u-tokai.ac.jp

***Abstract*—All mobile nodes participating in a group communicate with one another in the group through a shared radio link without cable or backbone composes an infrastructure-less network, called Mobile Ad hoc Network (MANET). Because network topology may be changed at any time as result of mobility of nodes and the failure of available routes, Quality-of-Service (QoS) of real-time multimedia application may have difficulty in satisfying their services. In this paper, we propose a routing scheme for multimedia services called Modified Dynamic Source Routing (MDSR) protocol. The MDSR selects the shortest path by using power level. Our proposed scheme provides lower average delay and mean number of hop path from source to destination than DSR protocol can provide.**

***Index Terms*—Communication system routing, MANET, Multimedia.**

I. Introduction

In Mobile Ad Hoc Network (MANET), nodes communicate with one another through shared, limited radio channel in peer to peer fashion that makes MANET to be an infrastructure-less network shown in Fig. 1. Moreover each mobile node or intermediate node in the network operates as routers forwarding packet to other nodes in order to establish end-to-end communication.

MANET can be categorized into flat architecture and hierarchical architecture [8], called cluster. In flat architecture, all nodes have the same rank, but in hierarchical architecture, nodes are separated into masters and slaves. Master nodes, cluster heads are selected among the nodes in the network and the remaining nodes, slave nodes, register themselves in one cluster of more cluster heads. A cluster head is responsible for communicating with its registered cluster nodes and communicating with other cluster heads in the network. By contrast, Slave nodes in each cluster communicate with each other via their respective master nodes.

All nodes in MANET can be moved freely or turned off at any time, so network topology may change rapidly and unexpectedly. Therefore, the packets transferring from end-to-end may be lost as the result of broken link of the availability routes. Quality of Services (QoS) for MANET have been done by many researchers [1], [2], such as QoS starting from physical layer up to application layer [3], QoS Medium Access Control (MAC), Genetic Algorithm QoS Routing Method [5], and QoS of Routing [4, 6, 7].

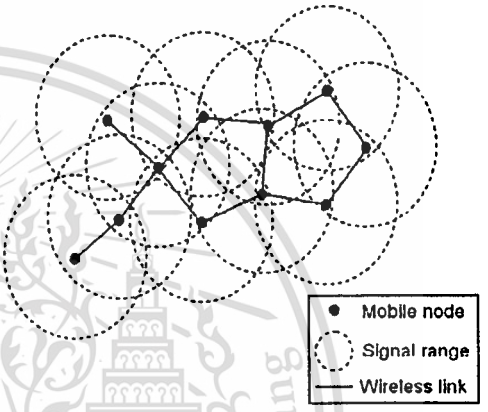


Fig. 1. A Mobile Ad hoc Network

Multimedia stream and some real time applications are delay-sensitive such as live voice and video conference. All routing protocols do not differentiate real-time traffic and non-real time traffic so in order to guarantee real-time traffic, multimedia stream we propose Modified Dynamic Source Routing (MDSR) protocol which selects the shortest path using higher power level. The remainder of this paper is organized as follows: In section II, we describe all related works and describe MDSR in Section III. In section IV, we evaluate the performance of MDSR comparing with DSR. Finally, the conclusion is given in Section V.

II. Related work

A. Current routing protocols

Due to the limited and shared resources in MANET, several routing protocols have been developed or proposed in order to adapt to the nature changes of wireless network such as nodes are mobile hosts, routes change frequently, and the topology is unpredictable. These routing protocols [6] can be classified into three different groups, i.e., global/proactive, on-demand/reactive and hybrid. Proactive routing protocols find

0-7803-9410-0/06/$20.00 ©2006 IEEE.

routes in the own maintaining route table in each node while the reactive routing protocols find route only one node need to send data to the other node. Hybrid routing protocols combine both proactive and reactive routing protocols together. The concept is that proactive is used in a cluster called zone whereas reactive is used to find nodes out size. This kind of routing protocol is used in hierarchical architecture.

1) Proactive routing protocols

Proactive routing protocols maintain information in a table on each node about the routing to the other node in the network. Although the topology of the network does not changed, this information must be updated periodically. Many proactive routing protocols have been proposed, for example, Destination Sequence Distance Vector (DSDV), Wireless Routing Protocol (WRP), Global State Routing (GSR), and Optimized Linked State Routing (OLSR) and so on. These protocols have some differences in term of maintaining updating routing information in the table [6]. There are pros and cons in the proactive routing protocol. The advantage is that when a node needs to send or forward data packets, route can be immediately used and the disadvantage is that this protocol allows significantly overhead and consumes bandwidth in the network.

2) Reactive protocols

Unlike proactive routing protocols, the reactive routing protocols create routes once a node wants to transmit data to a destination. The source node initiates route discovery process by flooding route query with in the network. When the destination is reached, route reply request will be sent back to the source. Once the route has been found, it is maintained until either destination becomes inaccessible or the route is no longer desired then route discovery process will be invoked again. Several reactive protocols have been proposed such as Dynamic Source Routing protocol (DSR), Ad hoc On-demand Distance Vector (AODV), Light-weight Mobile Routing (LMR), Temporary Ordered Routing Algorithm (TORA), Relative Distance Micro-discovery Ad hoc Routing (RDMAR) [6] and so on. Those protocols can be classified into two groups, source routing and hop-by-hop routing [6]. The advantage of reactive routing protocols is to overcome the overheads, occurred in proactive routing protocols.

3) Hybrid routing protocols

When wireless network becomes larger, it is complex to design routing protocols. All nodes in the network are separated into groups, called cluster. All clusters form a hierarchical infrastructure. In such network, hybrid routing protocols, i.e., combining proactive and reactive routing protocols, are used in order to take advantages on these two routing protocols where proactive maintains route in a cluster and reactive maintains route between clusters. Several hybrids routing protocols have been proposed such as Zone Routing Protocol (ZRP), Zone-based Hierarchical Link State (ZHLS), Distributed Dynamic Routing (DDR) and so on, but the most popular protocol is ZRP.

B. Dynamic Source Routing Protocol (DSR)

Dynamic Source Routing (DSR) protocol is one of the most popular on-demand routing protocols, proposed by David A.Maltz, David B.Johnson and Yin-Chun Hu. The purpose of DSR is to reduce overhead and to provide highly reactive services to help ensure successful delivery of data packets to reach a destination even though topology of the network is changed or there is another condition in the network.

The two common mechanisms of DSR protocol are:

1) Route Discovery

When a source node, wishing to send data packet to destination node. Route Discovery will be processed only the source node does not have a route to the destination. The process of route discovery is first to check a suitable source route in by searching routes in the route cache, previously learned, of the source node. If no route is found in its cache, it will initiate the route discovery protocol by broadcasting route request packet in order to find a new route to this destination.

When another node in the transmission range of the sender receives this route query request, it checks if it is the target of the route discovery. If so, it returns a route reply packet to the sender, giving a copy of the accumulated route record from the route request; when the sender receives this route reply packet, it caches this route in its cache for the future use in sending other packets to this destination If the node receiving the route request packet is not the target then it checks the route request packet if this packet is already seen or its address appeared in the route list of route request, this node drop route request and does not rebroadcast further. Otherwise, this node appends its address to the route record in the route request packet and propagates it by transmitting by transmitting it as local broadcast packet.

2) Route Maintenance

When a route being used to send data packet from source node to destination is broken, Route Maintenance is invoked by the source node in order to detect the broken link. When Route Maintenance indicates a source route is broken, the source node can attempt any other route it happens to know to the destination or the source node invokes route discovery again in order to find a new route for the rest of the packets on the source node to send to the destination node.

Both mechanisms of DSR protocol operate on-demand that is different from other routing protocols, which use periodic routing packet that causes overhead in the network. Moreover, in DSR protocol, a node may learn and cache multiple routes to any destination. This caching of multiple of routes avoids

the overhead of performing a new Route Discovery.

As briefly described all routing protocols above, the routing of MANET can be classified as proactive routing protocols, re-active routing protocol and hybrid routing protocol. These routing protocols have pros and cons depending on network environments and these routing protocols do not consider or differentiate real-time traffic and non-real time traffic. So we propose a routing scheme for multimedia services, Modified Dynamic Source Routing protocol, which select the shortest path using higher power level. The algorithm is described in the section III.

III. Proposed SCHEME: Modified Dynamic Source Routing Protocol

We propose a routing scheme for multimedia services called Modified Dynamic Routing (MDSR) protocol. We separate data transmissions into two groups. The first group is priority transmission of real-time data stream which is delay-sensitive data such as on-demand multimedia stream, video conference etc.,. When this kind of traffic is applied and in order to reduce query response time, MDSR protocol uses larger transmission power level or in other words, the number of nodes in the range of the source is increased. Therefore, source node can select one fastest node to forward route request. The second group is non-real time data transmission, which allows delay while transmitting. When this kind of traffic is applied smaller transmission power level is used so that energy can be saved.

Fig. 3 presents an example of the proposed routing scheme. We can see that the dotted line (S-A-B-D) representing a multimedia transmission has shorter path than the route used for non-real time data transmission that is represented by solid line (S-K-L-M-N-D).

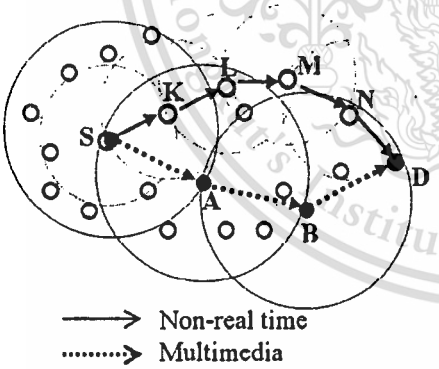


Fig. 2. An example of proposed algorithm

As multimedia stream is on-demand and in order to adapt to the dynamic topologies of MANET, DSR [1] is suitable for this data traffic. There are two mechanisms of this proposed scheme. The first mechanism is for a source node that wishes to send data packets as follows:

step 1) Check route in the Route Cache previously learned whether there is route to the destination to satisfy the traffic. If found, the source node uses it.

step 2) Otherwise and if it is real-time traffic, the node sets higher power strength, else set lower power strength.

step 3) In MDSR, a packet has the following fields: packet-type, source-add, dest-add, sequence#, route-list, traffic type, data, TTL. It places header to route request packet such as source node address, destination address, sequence number and traffic type to differentiate traffic in the network.

step 4) Finally, the source node sends out route request (RREQ) packet to its neighbors which can be intermediate node or destination node.

The second mechanism is for an intermediate node or a destination node that is to forward route request packet as follows:

step 1) If the pair <dest-addr, sequence#> for this RREQ is seen recently, drop route request packet.

step 2) If the address of this node appeared in the route-list in the RREQ, we drop this RREQ.

step 3) Check whether the route request is searching for the path for multimedia traffic or normal data traffic. If multimedia traffic, the node set higher power level strength otherwise set lower power strength.

step 4) Decrement TTL by one. If TTL counts down to zero, we drop this RREQ. TTL can limit the length of the delivery path. However, this path will be difficult to be maintained within a dynamic environment. In addition, unlimited packet flooding will degrade the network performance. The use of TTL can control the flooding traffic.

step 5) Append the address of this node to the route-list to track the route which the packet has traversed If this node is destination, send route reply back to the source otherwise re-broadcast this route request.

IV. Simulation and Results

A. Simulation Model and Parameters

We use network simulator called ns2 to analyze the performance of the proposed scheme. The network consists of 5 nodes to 50 nodes that move randomly at maximum speed 2m/s over an area 700x700 meters. The details of simulator parameters are shown in Table 1. We assume that packets can be lost or corrupted in transmission and a node receiving these corrupted packets can detect the error and discard them.

B. Simulation Result

The result of simulation is shown in Fig. 3,.we can notice that for multimedia transmission, the mean number of routing hops is smaller than non-real time. Because we use a higher transmission power level, the coverage area of each mobile

node is extended. Thus, it makes possible to reduce average number of routing hops. For example; as the number of mobile nodes is increased from 5 to 50, mean number of routing hops used for real-time is smaller than mean number of routing hops used by non-real time.

Table1: Simulation Parameters

Parameters		Values
Simulation Area		700 x 700
Number of nodes		5 10 20 30 40 50
Speed of nodes		0-2 m/s
Node movement		Uniform distribution
Transmission range	Real-time	300 meters
	Non real-time	250 meters

Moreover as shown in Fig 4, the delay of multimedia stream is always higher than the one of non-real-time traffic. Therefore transferring data packets of multimedia traffic from end-to-end nodes is faster than transferring data packets of non-real time traffic.

As shown in Fig. 3 and Fig. 4 respectively, we can see that when the number of nodes is increased, it does not mean that the average number of hops path is increased or the delay is higher than the smaller number of hops because it depends on the topologies of the network from time to time. Besides, a farthest node is selected to forward packets which causes a route used to be broken easily when the node moves so that it reduces the throughput and a new route must be re-established again.

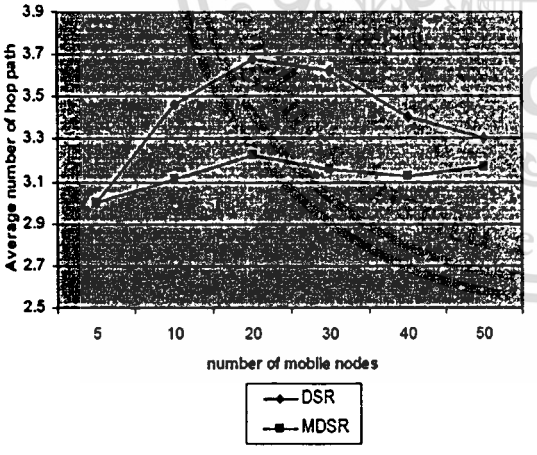


Fig. 3. Mean number of hop versus the number of mobile nodes

Anyway, from point to point of number of nodes in the networks, we see that the average number of nodes path for multimedia traffic is smaller than the average number of nodes path for non-real time traffic. Not only that but also the delay of multimedia traffic is higher than non real-time traffic because for multimedia data transmission, larger power level is used, so the number of nodes in the coverage area is also increased. Consequently it makes possibility for routing algorithm to select one node which is near the edge of its coverage area to forward multimedia data packet.

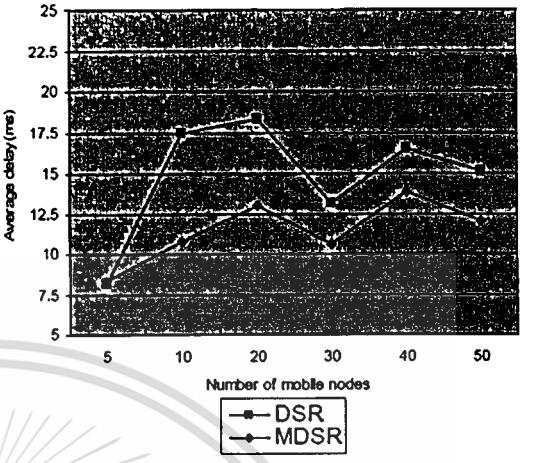


Fig. 4. Average delay versus number of mobile node.

V. Conclusion

We proposed Modified Dynamic Source Routing Protocol for Multimedia Services to support real-time traffic such as multimedia or video stream. This proposed scheme selects a shortest path for multimedia traffic by applying larger power level because the delay is sensitive for such kind of traffic. On the other hand, for non-real time traffic, this algorithm will use smaller power level longer path for non-real time. We can conclude that our propose schemes provides the lower mean number of hops path for multimedia traffic than the mean number of hops path for non-real time traffic. We also show that our proposed scheme can provide lower delay for multimedia traffic than the delay of non-real time traffic.

In the future works, we are going to consider the bandwidth calculation during route query request. Therefore the nodes in the routes found for both data traffic have available bandwidth to guarantee or transfer multimedia real-time data packet.

References

[1] David B. Johnson, David A Maltz and Yin-Chun Hu, "The Dynamic Source Routing Protocol for Mobile Adhoc Network", IETF internet draft 19 July, 2004.

[2] Charles Perkins, "Ad hoc On-Demand Distance Vector (AODV) Routing", IETF Internet draft July 2003.

[3] Prasant Mohapatra, Jian Li, and Chao Gui, "QoS in Mobile Ad hoc Networks," Special Issue on QoS in Next-Generation Wireless Multimedia Communicati-ons Systems in IEEE Wireless Communications Magazine, June 2003.

[4] Ying-Hong Wang, Hung-Zu Lin, Shu-Min Chang, "Interfering-aware QoS multipath routing for ad hoc wireless network", Advanced

Information Networking and Applications, 2004 AINA2004. 18th International Conference, Volume: 1, Pages: 29-34, 2004.

[5] Barolli L, Koyama A, Shiratori N, "A QoS routing method for ad-hoc networks based on genetic algorithm", Database and Expert Systems Applications, 2003. Proceedings. 14th International Workshop on, 1-5 Sept. 2003, Pages: 175–179.

[6] Mehran Abolhasan, Tadeusz Wysocki and Eryk Dutkiewicz, "A review of routing protocols for mobile ad hoc networks ", Ad Hoc Networks, Volume 2, Issue 1, January 2004, Pages 1-22.

[7] Shigang Chen, Nahrstedt K, "Distributed quality-of-service routing in ad hoc networks", Selected Areas in Communications, IEEE Journal on Volume: 17, Issue: 8, Aug. 1999 Pages:1488 – 1505

[8] Turgut D., Das S.K., Elmasri, R., Turgut B., "Optimizing clustering algorithm in mobile ad hoc networks using genetic algorithmic approach", Global Telecommunications Conference, 2002. GLOBECOM '02. IEEE, Volume: 1, 17-21 Nov. 2002 Pages: 62 – 66.

Appendix B

Bandwidth Calculation Modules

This section contains some important modules that are used in MDSR such as bandwidth calculation, slot assignment, creating a *reset* packet and releasing slots being reserved.

B. 1 Bandwidth Calculation Module

This module is called when a node receives a route request packet in order to calculate bandwidth from source node to this node. If no bandwidth available, then drop route request packet. Otherwise rebroadcast route request packet. If the route request packet arrives at its destination node, then this destination node processes slots information recorded in route request packet. As result, slots being used by every node in path from source to destination are assigned and this information is placed in route reply packet, which is sent back by the destination node. The detail of this module is described in section 4.2.2.

```
unsigned int DSRAgent::bandwidthCalculation(SRPacket &p, int i) {
        hdr_sr *srh =  hdr_sr::access(p.pkt);
        unsigned int link_BW=0,link_bw_i,bandwidth,bandwidth_size ,common_BW =0;
        unsigned int common_LAST=0  ,common_BW_size =0, remain_BW_size=0;
        unsigned int difference_BW1=0,difference_BW2=0;
        unsigned int re_diff2=0,diff2=0,out_bw=0,bandwidth_re=0;
        if ( i == 1) {          printf (" Error Never executed \n");
                                return srh->require_slot_array_list()[1];
        }
        else if (i == 2) {
            link_BW = srh->require_slot_array_list()[1] & srh->require_slot_array_list()[2];
            return link_BW;
        }
        else if ( i >=3 ) {
            i = i - 1;
            unsigned int link_bw_i = srh->require_slot_array_list()[i+1] & srh->require_slot_array_list()[i];
            bandwidth = bandwidthCalculation(p,i);
            common_BW = bandwidth & link_bw_i;
             common_BW_size = sizeb(common_BW);
```

```
        diff2 = bandwidth ^ common_BW;
        difference_BW1 = sizeb(link_bw_i ^common_BW);
        difference_BW2 = sizeb(bandwidth ^ common_BW);
        if ( difference_BW1 <= difference_BW2) {
              bandwidth_size = difference_BW1;
              remain_BW_size = difference_BW2-difference_BW1;
        }
        else {
              bandwidth_size = difference_BW2;
              remain_BW_size = difference_BW1 - difference_BW2;
        }
        if ( ( remain_BW_size > 0 ) || ( common_BW_size > 0)) {
              if ( common_BW_size <= remain_BW_size)
                 bandwidth_size = bandwidth_size + common_BW_size;
              else {
                 bandwidth_size = bandwidth_size + remain_BW_size;
                 common_BW_size = ( common_BW_size - remain_BW_size)/2;
                 if ( common_BW_size > 0)
                        bandwidth_size= bandwidth_size + common_BW_size;
                 }
        }
        if ( bandwidth_size >= srh->require_b()) { // compare bandwidth
           if (difference_BW2 >= srh->require_b() ) re_diff2 = selectDigit( diff2,srh->require_b());
           else if (difference_BW2 < srh->require_b() ) re_diff2 = diff2 |
           selectDigit(common_BW,fullfil(difference_BW2,srh->require_b()));
           else re_diff2 = selectDigit(common_BW,srh->require_b());
           common_LAST = re_diff2 & link_bw_i;
           out_bw = common_LAST ^ link_bw_i; // might be error here
           bandwidth_re = selectDigit(out_bw,srh->require_b());
        }
        else bandwidth_re = 0; //bn2
              if (p.dest ==net_id) {
                 if (i == 2) srh->require_slot_array_list()[i-1] = re_diff2;
                 srh->require_slot_array_list()[i] =    bandwidth_re;
              }
     return bandwidth_re;
     }
}
```

B. 2 Slot Reservation

When a node receives a route reply packet and its free slots (incoming and outgoing slots) satisfy the bandwidth information placed in this packet which means that this node is able to guarantee this connection, then some slots will be reserved according to the information in route reply packet. The process of this reservation is in the module below.

```
void DSRAgent::assignedTimeSlotUsed(SRPacket& p,Time todayBandwidth,int nodeLocation){
// if nodelocation is at the first or at the end of a route then just assign
// else combine incoming and outgoing slot used.
    hdr_sr *srh =  hdr_sr::access(p.pkt);
    int location = -1 , remain = -1;
    location = findNodeLocation(p.route.dump());
    if (location == 0) {
        printf("Error assignedTimeSlot  %s nodeid %s\n",p.route.dump(), net_id.dump());
        goto stop;
    }
    if ( location == 1){ // take the first link_bw
        assignedTimeSlotUsedMark(todayBandwidth,stringToInteger(net_id.dump()),
        p.route.BW_path_bandwidth()[location-1]);
    }
    else if ( location == p.route.length() ) { // take the (location -1 )th link_bw
         assignedTimeSlotUsedMark(todayBandwidth,stringToInteger(net_id.dump()),
        p.route.BW_path_bandwidth()[location-2]);
        }
    else { // combine incoming and outgoing slot used.
         unsigned int unionBw = p.route.BW_path_bandwidth()[location-2] |
         p.route.BW_path_bandwidth()[location-1];
         if ((unionBw < 0) || (unionBw >65535 )) {
             printf("Dsragent Error union Bw %d  node id %s route %s ",unionBw,net_id.dump(),
             p.route. dump());
         }
         assignedTimeSlotUsedMark(todayBandwidth,stringToInteger(net_id.dump()), unionBw);
    }
stop: printf("Stop\n");
}
```

B. 3 Send Reset Packet

When a node receives any route reply packet, the node's free slots do not satisfy the slots information in route reply packet or in other word, this node does not have enough bandwidth to guarantee the call connection then this module (send reset packet) is called then a *reset packet* is created and it is sent back to previous node in order to release its reserved slots for this connection.

```
void DSRAgent::sendReset(SRPacket& p){
        hdr_sr *old_srh = hdr_sr::access(p.pkt);
        int location = -1;
        if (p.route.full()) {
                printf("Error sendReset \n");
                return; // alas, the route would be to long once we add ourselves
        }
        SRPacket p_reset;
        p_reset.src = net_id;
        p_reset.dest = p.src;
        p_reset.pkt = allocpkt() ; // 4444
        hdr_ip *new_iph =  hdr_ip::access(p_reset.pkt);
        new_iph->daddr() = Address::instance().create_ipaddr(p_reset.dest.getNSAddr_t(),RT_PORT);
        new_iph->dport() = RT_PORT;
        new_iph->saddr() =  Address::instance().create_ipaddr(p_reset.src.getNSAddr_t(),RT_PORT);
        new_iph->sport() = RT_PORT;
        new_iph->ttl() = 255;
        hdr_sr *new_srh =  hdr_sr::access(p_reset.pkt);
        new_srh->init();

        location = findNodeLocation(p.route.dump());
        if ((location == 0 ) || (location > p.route.length()) ) printf("Error length or node id does not exist in
        the route path \n");
        p_reset.route.appendPath_reset(p.route,0,location);
        for ( int j = 0; j < location ; j++ ) {
                p.route[j].fillSRAddr(new_srh->reset_addrs()[j]);  // copy from p.route to a new header
                        //printf("dsragent sendReset that is ok %d \n",p_reset.route.path_bandwidth[j]);
                        // that is ok because bandwidth of route is not useful in this case.
                        // what we need is information is placed in the header of the packet.
```

```
                    // that is ok if data is inocorrect be cause bandwidth must be minus 1
        }
        p_reset.route.length_reset() = location ; // set length to the route
        new_srh->route_reset_len() = location; // increament 1
        new_srh->route_reset_reset() = 1;
                // this is RESET packet, sent back to release the slots that were assigned
                // propagate the request sequence number in the reply for analysis purposes
        new_srh->rtreq_seq() = old_srh->rtreq_seq();
        hdr_cmn *new_cmnh =  hdr_cmn::access(p_reset.pkt);
        new_cmnh->ptype() = PT_DSR;
        new_cmnh->size() = IP_HDR_LEN;
        p_reset.route.reverseInPlace();
        p_reset.route.appendPath_reset_bw(p.route,0,location);
        for ( int z =1 ; z < location ; z++ ) {
                new_srh->reset_slot_array_list()[z] = p_reset.route.path_bandwidth[z-1];
        }
        p_reset.route.resetIterator();
        p_reset.route.fillSR(new_srh);
        new_cmnh->size() += new_srh->size();
        for( int m = 0; m < location -1; m++ ) {
            printf("route copy %d route %s\n",p_reset.route.path_bandwidth[m], p_reset.route.dump());
        }
        double d = 0.0;  //Random::uniform(RREQ_JITTER);
        Scheduler::instance().schedule(this,p_reset.pkt,d);
}
```

B. 4 Release Reserved Slots

Every node receives a *reset packet* (described in A.3), the node call the module below to release its slots being reserved for this connection. The slots were reserved are incoming slots and outgoing slots.

```
void DSRAgent::releaseSlot(SRPacket& p){
        hdr_sr *srh = hdr_sr::access(p.pkt);
        int location = 0, nodeId = -1;
        unsigned int sum_bandwidth =0 ;
        location = findNodeLocation(p.route.dump());
        nodeId = stringToInteger(net_id.dump());
```

```
if( p.dest == net_id ) { // packet has arrived at the destination
    if ( location <=1) return;
    else if ( location > p.route.length() )  printf ("Error desnitationr release slot location %d route
    %s length %d\n",location,p.route.dump(), p.route.length());
    else {
        printf("Release des %d node id %s route %s srh %d \n",
        routeb[nodeId].free_slots,net_id.dump(), p.route.dump(),srh-
        >reset_slot_array_list()[location -1]);
        routeb[nodeId].free_slots = routeb[nodeId].free_slots | srh->reset_slot_array_list()
        [location -1];
        printf("after Release des %d node id %s route %s \n", routeb[nodeId].free_slots,net_
        id.dump(),  p.route.dump());
        if ( routeb[nodeId].free_slots >65535 ) printf("Error Release slot \n");
    }
}else {
    if ( location <=1) return;
    else if ( location > p.route.length() -1) printf("Error location reset\n");
    else {
        sum_bandwidth = srh->reset_slot_array_list()[location] | srh->reset_slot_array_list()
        [location -1];
        printf("Release des %d node id %s route %s srh %d \n",routeb[nodeId].free_slots,
        net_id.dump(), p.route.dump(), sum_bandwidth);
        routeb[nodeId].free_slots = routeb[nodeId].free_slots | sum_bandwidth;
        if ( routeb[nodeId].free_slots >65535 ) printf("Error Release slot \n");
        }
}
}
```

Author's Biography

Mr. TAING Nguon was born on March 20, 1978 in Tbaung Khmum district, Kampong Cham province, Cambodia. In 1999, He received bachelor degree in computer science from the department of computer science, faculty of science, Royal University of Phnom Penh, Cambodia. In 2000, He received a certificate of pedagogy from Faculty of Pedagogy, Phnom Penh, Cambodia. From the year 2001 to 2004, he was a lecturer in the department of computer and communication engineering, Institute of Technology of Cambodia (ITC), Cambodia. In 2004, he was awarded by AUN/SEED-Net (www.seed-net.org), a Southeast Asia educational development network program, which is financed by JICA, to pursue his master of engineering at King Mongkut's Institute of Technology Ladkrabang, Bangkok, Thailand. As the goal of this program, after his master graduation, he will be working again as a lecturer in the department of Computer and Communication Engineering, Institute of Technology of Cambodia. His research interest is in Wireless Local Area Network and Computer Network Security.