A Research on Multi-path Routing Protocol for Video Streaming over Ad hoc Networks

アドホックネットワーク上のビデオストリーミング向け
複数経路ルーティングプロトコルに関する研究

By

Palarp Prapatsaranon
(46343)

A thesis submitted to the Department of Frontier Informatics, Graduate School of Frontier Science for the degree

of

Master of Science

at

The University of Tokyo
Thesis Supervisor
Professor Hitoshi Aida

Abstract

Much effort has recently been reported to improve reliability and performance of the video streaming over the mobile ad hoc networks. The multi-path routing protocol for video streaming over mobile ad hoc networks has been developed to achieve higher throughput and reliability than the current routing algorithm. The multi-path routing allows a sender to transmit data via multiple paths to the same destination. Although several studies of multi-path routing have been recently reported, the results of the simulation are not satisfied the video streaming. Therefore, this thesis proposes and implements routing protocol for video streaming in the congested network. The algorithm to select the paths is very important for the multi-path. The algorithm to avoid the node which holds many paths to the destination (common node) is used in the thesis. If two transmission paths are close to each other, the signal interference may occur and make the degradation in the transmission. Additionally, the preemptive technique is also used to improve the multi-path routing to switch to the new path before the path break occurs. This technique prevents the loss of packets because of the new paths will be available for the packet transmission. Performance studies of the multi-path routing were conducted by using ns2 network simulator. The performance of multi-path routing was compared with the one-path routing. Simulation results showed that the multi-path routing achieved higher throughput than one-path routing in the mobility environment.

Dedication

To everyone in my family.

Acknowledgements

The author would like to express his sincere appreciation to his supervisor, Prof. Hitoshi Aida for his kind and continuous guidance, great support, valuable advices and encouragement throughout his research study. The most valuable part in the entire period of this research was his understanding and encouragement given to him to carry out the study independently.

Also, the author frankly gives his special gratefulness to Mr. Shingo Chiba, Ms. Kultida Rojviboonchai, Mr. Wittaya Apirakviriya, Mr. Nguyen Minh Tuan and all members of the Aida Laboratory for their assistance in this study.

The author wishes to express his gratitude to Mr. Chakrit Suwannachote, Mr. Pichit Kiatphotha, Mr. Worawut Temphuwapat, Mr. Aniwat Tandaechanurat, Mr. Worawut Sae-kok and everyone in Thai Todai Community for their helpful discussion, useful comment, and encouragement.

Special thanks to the government of Japan for granting him Monbukagakusho scholarship and giving him a chance to study in the University of Tokyo, which made it possible for him to complete the study.

More than anything else, the author would like to express all of his deepest appreciation and gratitude to his beloved father, mother and brother for their immeasurable courage and contribution to his study.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

In the recent years, the wireless communication technology has increased growing interests in the use of mobile ad hoc wireless networks. [1] Mobile ad hoc network is developed to support robust and efficient operation in mobile wireless networks by using routing functionality into mobile nodes. Such networks are envisioned to have dynamic, sometimes rapidly-changing, random, multi-hop topologies which are likely composed of relatively bandwidth-constrained wireless link.

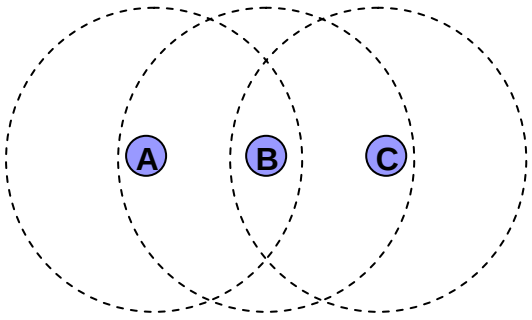


Figure 1.1: A simple ad hoc network of three wireless mobile hosts

Mobile ad hoc network is the group of wireless mobile hosts forming a temporary network. All of the nodes in the network connect to each other via wireless link without the aid of any established infrastructure or centralized administration. If only two hosts, located closely together, are involved in the ad hoc network, no real routing protocol or routing decision are necessary. In many mobile ad hoc networks, though two hosts want to communicate may not be within wireless transmission range

of each other, but could communicate if other hosts between them also participating in the ad hoc network are willing to forward packets for them.

For example, in the network illustrated in Figure 1, mobile node C is not within the range of node A's wireless transmitter. If node A and node C want to exchange packets, they may in this case enlist the services of node B to forward packets for them, since node B is within the overlap between node A's range and node C's range. Indeed, the routing problem in a real mobile ad hoc network may be more complicated than this example suggests, due to the inherent non-uniform propagation characteristics of wireless transmissions and due to the possibility that any or all of the nodes involved may move at any time.

Due to the nature of mobile ad hoc networks, a wireless link have high transmission error rate because of shadowing, fading, path loss and interference from other transmitting users. An end-to-end path found in ad hoc networks has an even higher error rate since it is the concatenation of multiple wireless links. The frequent link failures and route changes cause packet losses and reduce the received video quality. To provide an acceptable received video quality in mobile ad hoc networks, several researches had proposed approaches to improve the quality of the video streaming.

1.2 Objective and Scope of Thesis

The objective of this research is to use multi-path routing protocol to achieve higher throughput and reliability in video streaming over mobile ad hoc network than the current one-path routing protocol. Smoothness transmission with less jitter is needed for the video streaming over mobile ad hoc network. The thesis is then required for the multi-path routing protocol with preemptive technique to avoid jitter in the mobility and congested mobile ad hoc network.

1.3 Overview of Thesis

Recent research approach to improve performance of video streaming over mobile ad hoc networks is described in chapter 2. Jointcount-based multi-path routing protocol is then introduced in chapter 3. Chapter 4 proposes the multi-path routing

protocol with preemptive technique. In chapter 5, simulation-based results of the multi-path routing and result analysis are explained. Chapter 6 describes further discussion about the proposed algorithms compared to the one-path approach. Finally, the conclusion and future work is presented in section 7.

Chapter 2

Recent Research Approach to Improve Performance of Video Streaming over Ad hoc Networks

This chapter describes the mobile ad hoc networks and the recent research approach to improve performance of video streaming over mobile ad hoc networks. In section 2.1, the multi-path technique is presented. Then, the concept of Ad hoc On-demand Distance Vector routing is described in section 2.2.

2.1 Multi-path Technique

The User Datagram Protocol (UDP), typically used in almost all real-time multimedia applications, only extends the best-effort, host-to-host IP service to the process-to-process level. When congestion occurs, an unlimited amount of UDP datagrams may be dropped since UDP is non-adaptive. Real-time multimedia applications must implement additional rate control and error control mechanisms in order to cope with network congestion.

In mobile ad hoc networks, a wireless link have high transmission error rate because of shadowing, fading, path loss and interference from other transmitting users. An end-to-end path found in mobile ad hoc networks has an even higher error rate since it is the concatenation of multiple wireless links. Moreover, user mobility makes the network topology change constantly. Mobile ad hoc networks also need to reconfigure

themselves when users join or leave the network. The frequent link failures and route changes cause packet losses and reduce the received video quality. This is different from wired networks, where packet loss is mainly caused by congestion and buffer overflow. To provide an acceptable received video quality in mobile ad hoc networks, there should be effective error control to reduce packet losses to a certain level.

The [5] examine the problem of using multi-path transport, by which multiple paths are used to transfer data, for a video streaming.

2.1.1 Application Scenarios

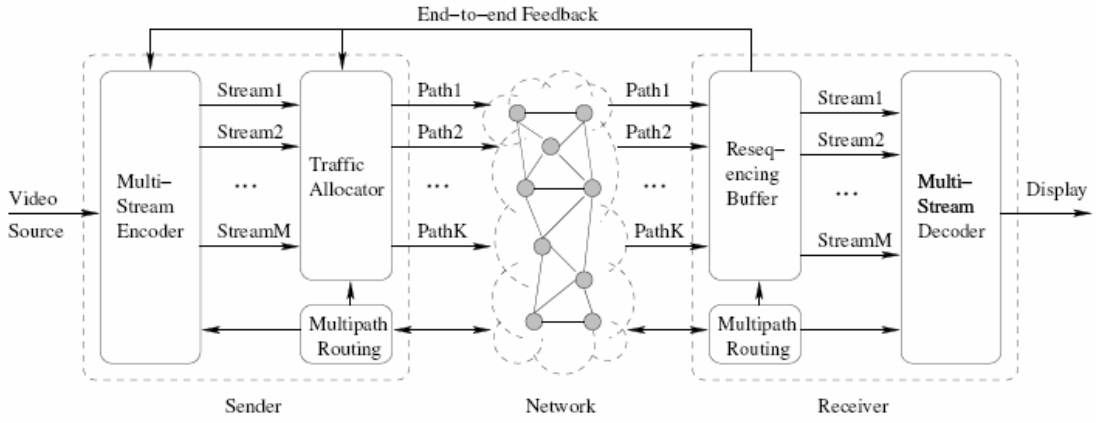


Figure 2.1: The general architecture for the multi-path transport of video streaming applications.

Figure 2.1 illustrates the general architecture for the multi-path transport of video streaming. At the sender, a raw video is first compressed by a video encoder. The encoder may generate a single compressed video flow, or multiple compressed video flows. The latter case is called a multistream coder. [5] Then the flows are partitioned and assigned to the multiple paths by a traffic allocator. These paths are maintained by a multi-path routing protocol. When the flows arrive at the receiver, they are first put into a resequencing buffer to restore the original order. Finally, the video data is extracted from the buffer to be decoded and displayed.

2.1.2 Advantage of using Multi-path Technique

The multi-path transport distributes traffic load in the network more evenly. For example, a large burst of data can be partitioned into several smaller bursts, each transmitted on a different path. A high rate video flow can be partitioned into several subflows, each with a lower rate and sent on a different path. Such balanced load results in less congestion inside the network. Thus the video packet losses caused by router buffer overflow can be effectively reduced.

The multi-path transport provides a larger aggregate capacity for a multimedia session. In a mobile ad hoc network, since the available link bandwidth may be limited and time varying, a high rate flow may not find enough capacity on a single path. With multi-path transport, the flow can be partitioned into several thinner subflows, each of which can be accommodated by a path.

If a set of disjoint paths are used in multi-path transport, losses experienced by the subflows may be independent to each other. When a path is down because of a link failure, which happens more often in a mobile ad hoc network, it is likely that some other paths are still in good condition. Thus the receiver can always receive some data during any period. With proper error concealment schemes applied, the display will not be interrupted by link failures, although certain degradation in the video quality will be observed. Furthermore, with path diversity, error control schemes can be designed jointly with the traffic allocator, making traditional error control schemes more effective and resulting in better error resilience.

The multi-path transport facilitates load balancing for the servers. A client can download video from multiple servers when multi-path transport is used. A high rate session can be partitioned into several lower rate ones, each with a smaller server processing time. The smaller granularity of per user loads allows for more even load balancing, which can translate into either more clients supported or lower response time.

To summarize, the use of multi-path transport for real-time multimedia applications in ad hoc networks can effectively reduce packet losses, provide better scalability, and provide un-interrupted display of video even with the presence of frequent link failures.

2.1.3 Type of Multi-path Routing

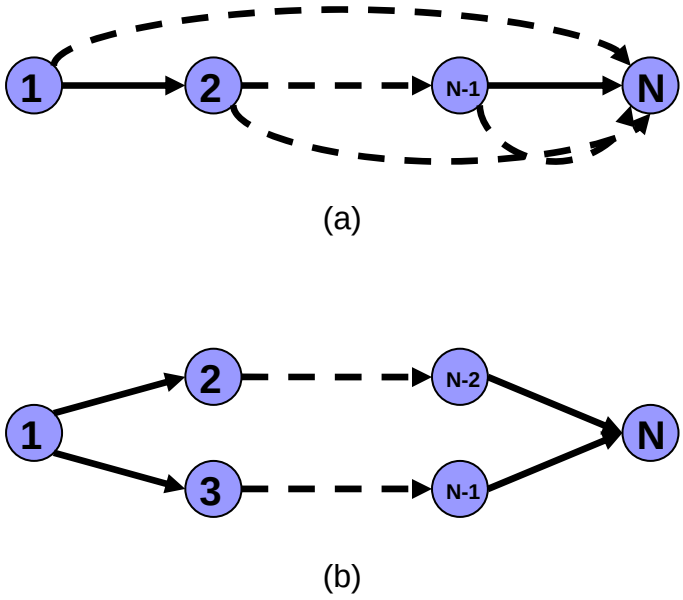


Figure 2.2: Two types of multi-path routing:
(a) Braided multi-paths, (b) Node disjoint multi-paths.

As illustrated in Figure 2.2, in order to use multi-path transport, the underlying routing protocol must provide and update the multiple paths between the source and the destination node.

From the [5], there are two types of multi-path routing protocols, as illustrated in Figure 2.2. A set of braided paths is shown in Figure 2.2 (a), where each node maintains a backup path to the destination node. Figure 2.2 (b) shows two nodes disjoint paths. There is no common node between these paths, except for the source and the destination nodes. A relaxed type of disjoint paths is link disjoint paths, where sharing of nodes, but not links, is allowed. The braided multi-path routing is a relaxed version of disjoint multi-path routing, since the latter may be more difficult to implement or unavailable in some network topology. However, the benefits of using multi-path transport are generally maximized when disjoint paths are in use. For example, when a common node, shared by two paths, is congested or is unavailable, both paths will fail and the receiver video display will be interrupted until a new set of paths are found.

2.1.4 Split Multi-path Routing Algorithm

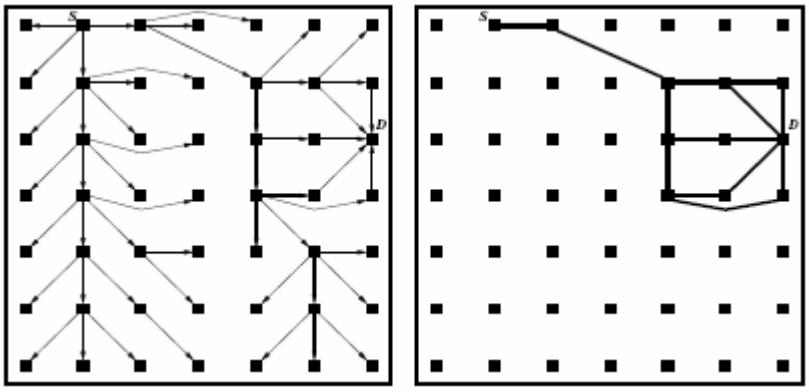
The [6] has presented Split Multi-path Routing (SMR) protocol that builds maximally disjoint paths. Multiple routes, of which one is the shortest delay path, are discovered on demand. Established routes are not necessarily of equal length. Data traffic is split into multiple routes to avoid congestion and to use network resources efficiently.

A. Split Multi-path Routing Algorithm

Split Multi-path Routing (SMR) is an on-demand routing protocol that builds multiple routes using request/reply cycles. When the source needs a route to the destination but no route information is known, it floods the Route Request (RREQ) packet to the entire network. Because this packet is flooded, several duplicates that traversed through different routes reach the destination. The destination node selects multiple disjoint routes and sends Route Reply (RREP) packets back to the source via the chosen routes.

A.1 RREQ Propagation

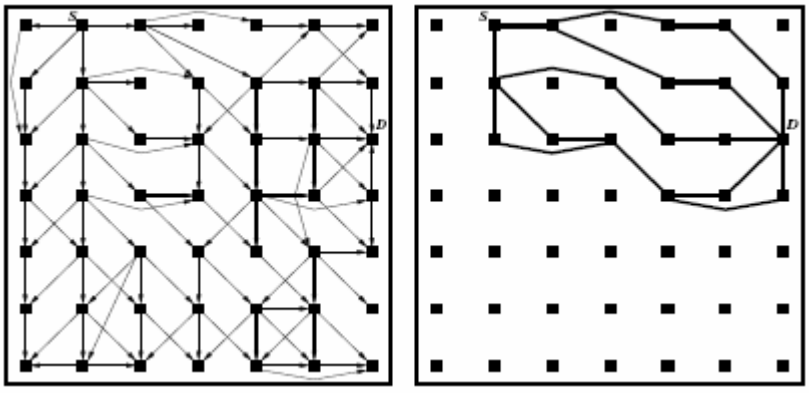
The main goal of Split Multi-path Routing is to build maximally disjoint multiple paths. The SMR will construct maximally disjoint routes to prevent certain nodes from being congested, and to utilize the available network resource efficiently. To achieve this goal in on-demand routing schemes, the destination must know the entire path of all available routes so that it can select the routes. Therefore, the SMR use the source routing approach where the information of the nodes that consist the route is included in the RREQ packet. Additionally, intermediate nodes are not allowed to send RREPs back to the source even they have route information to the destination. If nodes reply from cache as in DSR and AODV routing protocol, it is difficult to establish maximally disjoint multiple routes because not enough RREQ packets will reach the destination and the destination node will not know the information of the route that is formed from the cache of intermediate nodes.



(a) RREQ propagation (b) Available paths

Figure 2.3: Overlapped multiple routes.

When the source nodes has data packets to send but does not have the route information to the destination, it transmits a RREQ packet. The packet contains the source ID and a sequence number that uniquely identify the packet. When a node other than the destination receives a RREQ that is not a duplicate, it appends its ID and re-broadcasts the packet. However, dropping all duplicate RREQs only generate multiple paths that are mostly overlapped. Figure 2.3 (a) shows the paths taken by RREQ from the source node S to the destination node D, and Figure 2.3 (b) depicts the available routes. The figure shows that all five routes share the first two links.



(a) RREQ propagation (b) Available paths

Figure 2.4: Multiple routes with maximally disjoint paths.

In order to avoid this overlapped route problem, the Split Multi-path Routing introduces a different packet forwarding approach. Instead of dropping every duplicate RREQs, intermediate nodes forward the duplicate packets that traversed through a different incoming link than the link from which the first RREQ is received, and whose hop count is not larger than that of the first received RREQ. Figure 2.4 (a) shows the paths taken by RREQs using this technique. The Split Multi-path select more disjoint paths from routes available in Figure 2.4 (b) than those in Figure 2.3 (b). This approach has a disadvantage of transmitting more RREQ packets, but it enables nodes to discover maximally disjoint routes.

A.2 Route Selection Method

In Split Multi-path algorithm, the destination selects two routes that are maximally disjoint. More than two routes can be chosen, but the [6] limit the number of routes to two. One of the two routes is the shortest delay route; the path taken by the first RREQ the destination receives. The Split Multi-path uses the shortest delay path as one of two routes to minimize the route acquisition latency required by on-demand routing protocols. When receiving the first RREQ, the destination records the entire path and sends a RREP to the source via this route. The node IDs of the entire path is recorded in the RREP (for DSR routing protocol), and hence the intermediate nodes can forward this packet using this information. After this process, the destination waits certain duration of time to receive more RREQs and learn all possible routes. It then selects the route that is maximally disjoint to the route that is already replied. The maximally disjoint route can be selected because the destination knows the entire path information of the first route and all other candidate routes. If there is more than one route that are maximally disjoint with the first route, the one with the shortest hop distance is chosen. If there still remain multiple routes that meet the condition, the path that delivered the RREQ to the destination the quickest between them is selected. The destination then sends another RREP to the source via the second route selected. Note that two routes of the session are not necessarily of equal length.

B. Route Maintenance

A link of a route can be disconnected because of mobility, congestion, and packet collisions. It is important to recover broken routes immediately to do effective routing. In Split Multi-path, when a node fails to deliver the data packet to the next hop of the route (by receiving a link layer feedback from IEEE 802.11 or not receiving passive acknowledgement), it considers the link to be disconnected and sends a Route Error (RERR) packet to the upstream direction of the route. The RERR message contains the route to the source, and the immediate upstream and downstream nodes of the broken link. Upon receiving this RERR packet, the source removes every entry in its route table that uses the broken link (regardless of the destination). If only one of the two routes of the session is invalidated, the source uses the remaining valid route to deliver data packets.

When the source is informed of a route disconnection and the session is still active, it may use one of the two policies in rediscovering routes:

- Initiates the route recovery process when any route of the session is broken, or
- Initiate the route recovery process only when both routes of the session are broken.

The first scheme reconstructs the routes more often and produces more control overhead than the second scheme, but the former provides multiple routes most of the time and be robust to route breaks.

C. Allocation Granularity

When the source receives a RREP after flooding the RREQ, it uses the first discovered route to send buffered data packets. When the second RREP is received, the source has two routes to the destination, and can split traffic into two routes. The [6] use a simple per-packet allocation scheme when there are more than one available route to destination. One drawback of this scheme is out of order delivery and re-sequencing burden on the destination.

D. Simulation Analysis

The [7] has uses the Split Multi-path algorithm to apply in routing protocol of the Multimedia Transport. The case of no node mobility simulation result is acceptable.

However, in the case of having node mobility simulation result is not as good as expected. One possible explanation could be the interference between the nodes. Even though the two paths are used but if the nodes are closed to each other, the signal interference will occur and make the degradation in the transmission.

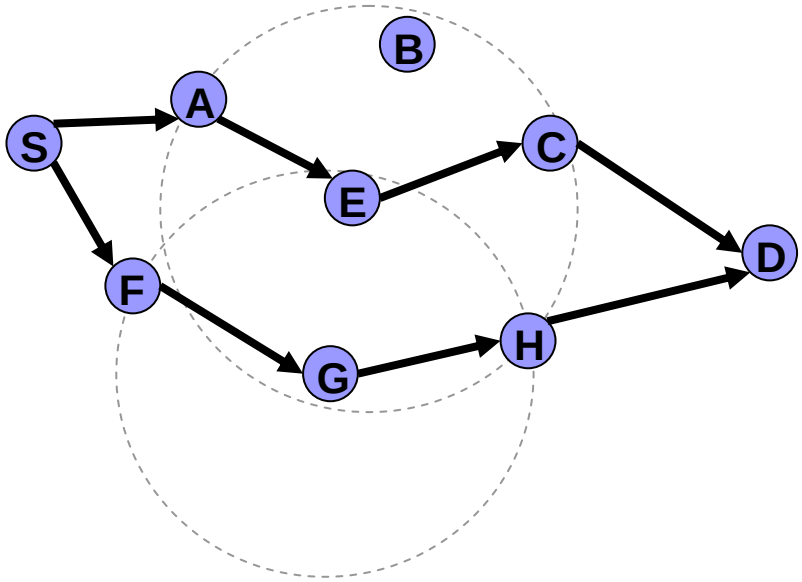


Figure 2.5: Signal interference in the transmission.

Figure 2.5 shows the two paths between the source node S and the destination node D. After the source S sends the packet via both two paths to the node A and node F respectively, node A and node F try to forward the packet to the next node in the path. The problem is that both node A and node F try to send the packet to the next node but the node A and the node F is too close to each other. The node A and the node F are in the transmission range of each other. So there is signal interference between the node A and the node C which will make the degradation in the transmission.

2.2 Ad hoc On-Demand Distance Vector Routing (AODV)

The ad hoc on-demand distance vector (AODV) algorithm [8] can be called a pure on-demand route acquisition system; nodes those do not lie on active paths neither maintain any routing information nor participate in any periodic routing table exchanges. Further, a node does not have to discover and maintain a route to another node until the

two needs to communicate, unless the former node is offering its services as an intermediate forwarding station to maintain connectivity between two other nodes.

When the local connectivity of the mobile node is of interest, each mobile can become aware of the other nodes in its neighborhood by the use of several techniques, including local (not system-wide) broadcasts known as hello messages. The routing tables of the nodes within the neighborhood are organized to optimize response time to local movements and provide quick response time for requests for establishment of new routes. The algorithm's primary objectives are:

- To broadcast discovery packets only when necessary
- To distinguish between local connectivity management (neighborhood detection) and general topology maintenance
- To disseminate information about changes in local connectivity to those neighboring mobiles nodes those are likely to need the information.

AODV uses a broadcast route discovery mechanism [10], as is also used (with modifications) in the Dynamic Source Routing (DSR) algorithm [11]. Instead of source routing, the AODV relies on dynamically establishing route table entries at intermediate nodes. This difference pays off in networks with many nodes, where a larger overhead is incurred by carrying source routes in each data packet. To maintain the most recent routing information between nodes, the concept of DSDV routing is borrowed. Unlike DSDV, each node maintains a monotonically increasing sequence number counter which is used to supersede stale cache routes. The combination of these techniques yields an algorithm that uses bandwidth efficiently (by minimizing the network load for control and data traffic) is responsive to changes in topology, and ensures loop-free routing.

2.2.1 Path Discovery

The Path Discovery process in initiated whenever a source node needs to communicate with another node for which it has no routing information in its table. Every node maintains two separate counters: a node sequence number and a

broadcast_id. The source node initiates path discovery by broadcasting a route request (RREQ) packet to its neighbors. The RREQ contains the following fields:

< source_addr, source_sequence_#, broadcast_id, dest_addr, dest_sequence_#, hop_cnt >

The pair < source_addr, broadcast_id > uniquely identifies a RREQ. broadcast_id is incremented whenever the source issues a new RREQ. Each neighbor either satisfies the RREQ by sending a route reply (RREP) back to the source, or rebroadcasts the RREQ to its own neighbors after increasing the hop_cnt. Notice that a node may receive multiple copies of the same route broadcast packet from various neighbors. When an intermediate node receives a RREQ, if it has already received a RREQ with 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 in order to implement the reverse path setup, as well as the forward path setup will accompany the transmission of the eventual RREP:

- Destination IP address
- Source IP address
- Braodcast_id
- Expiration time for reverse path route entry
- Source node's sequence number.

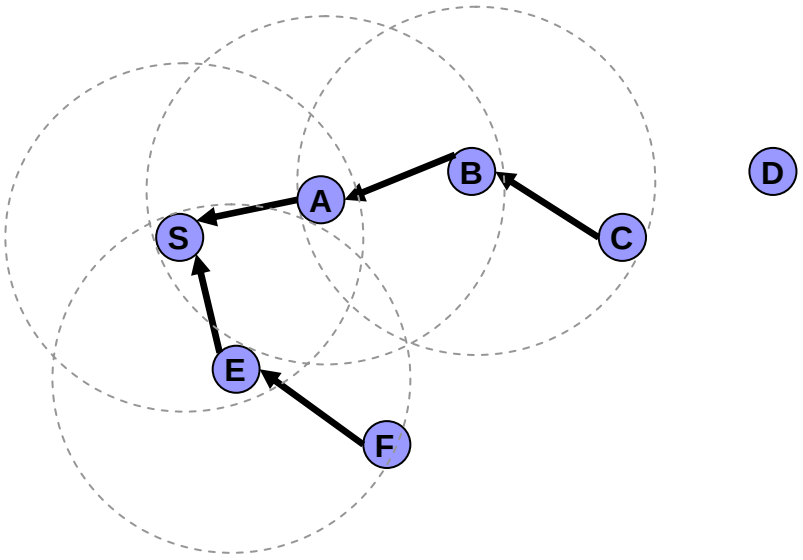


Figure 2.6: Reverse Path Formation.

A. Reverse Path Setup

There are two sequence numbers (in addition to the broadcast_id) included in a RREQ: the source sequence number and the last destination sequence number known to the source. The source sequence number is used to maintain freshness information about the reverse route to the source, and the destination sequence number specifies how fresh a route to the destination must be before it can be accepted by the source.

As the RREQ travels from a source to various destinations, it automatically sets up the reverse path from all nodes back to the source, as illustrated in Figure 2.6. To set up a reverse path, a node records the address of the neighbor from which it received the first copy of the RREQ. These reverse path route entries are maintained for at least enough time for the RREQ to traverse the network and produce a reply to the sender.

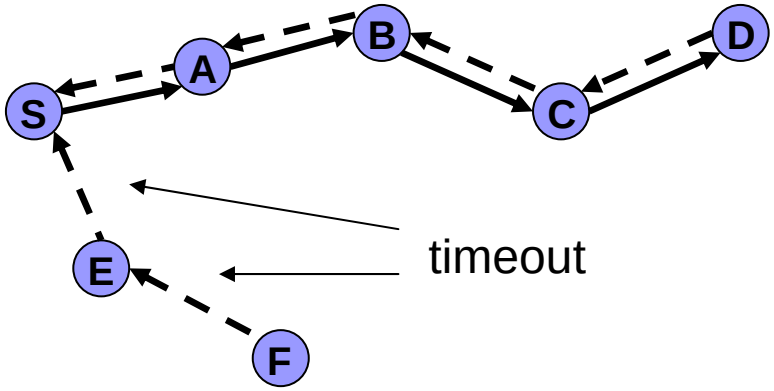


Figure 2.7: Forward Path Formation.

B. Forward Path Setup

Eventually, a RREQ will arrive at a node (possibly the destination itself) that possesses a current route to the destination. The receiving node first checks that the RREQ was received over a bi-directional link. If an intermediate node has a route entry for the desired destination, it determines whether the route is current by comparing the destination sequence number in its own route entry to the destination sequence number in the RREQ. If the RREQ's sequence number for the destination is greater than that

recorded by the intermediate node, the intermediate node must not use its recorded route to respond to the RREQ. Instead, the intermediate node rebroadcasts the RREQ. The intermediate node can reply only when it has a route with a sequence number that is greater than or equal to that contained in the RREQ. If it does have a current route to the destination, and if the RREQ has not been processed previously, the node then unicasts a route reply (RREP) packet back to its neighbor from which it received the RREQ. A RREP contains the following information:

< source_addr, dest_addr, dest_sequence_#, hop_cnt, lifetime >

By the time a broadcast packet arrives at a node that can supply a route to the destination, a reverse path has been established to the source of the RREQ. As the RREP travels back to the source, each node along the path sets up a forward pointer to the node from which the RREP came, updates its timeout information for route entries to the source and destination, and records the latest destination sequence number for the requested destination. Figure 2.7 represents the forward path setup as the RREP travels from the destination D to the source node S. Nodes that are not along the path determined by the RREP will timeout after ACTIVE_ROUTE_TIMEOUT and will delete the reverse pointers.

A node receiving an RREP propagates the first RREP for a given source node towards that source. If it receives further RREPs, it updates its routing information and propagates the RREP only if the RREP contains either a greater destination sequence number than the previous RREP, or the same destination sequence number with a smaller hopcount. It suppresses all other RREPs it receives. This decreases the number of RREPs propagating towards the source while also ensuring the most up-to-date and quickest routing information. The source node can begin data transmission as soon as the first RREP is received, and can later update its routing information if it learns of a better route.

2.2.2 Route Table Management

In addition to the source and destination sequence numbers, other useful information is also stored in the route table entries, and is called the soft-state associated

with the entry. Associated with reverse path routing entries is a timer, called the route request expiration timer. The purpose of this timer is to purge reverse path routing entries from those nodes that do not lie on the path from the source to the destination. The expiration time depends upon the size of the mobile ad hoc network. Another important parameter associated with routing entries is the route caching timeout, or the time after which the route is considered to be invalid.

In each routing table entry, the address of active neighbors through which packets for the given destination are received is also maintained. A neighbor is considered active (for that destination) if it originates or relays at least one packet for that destination within the most recent active_timeout period. This information is maintained so that all active source nodes can be notified when a link along a path to the destination, which is followed by packets along active route entries, is called an active path.

A mobile node maintains a route table entry for each destination of interest. Each route table entry contains the following information:

- Destination
- Next Hop
- Number of hops
- Sequence number for the destination
- Active neighbors for this route
- Expiration time for the route table entry

Each time a route entry is used to transmit data from a source toward a destination, the timeout for the entry is reset to the current time plus active_route_timeout.

If a new route is offered to a mobile node, the mobile node compares the destination sequence number of the new route to the destination sequence number for the current route. The route with the greater numbers are the same, then the new route is selected only if it has a fewer number of hops to the destination.

2.2.3 Path Maintenance

Movement of nodes not lying along an active path does not affect the routing to that path's destination. If the source node moves during an active session, it can reinitiate the route discovery procedure to establish a new route to the destination. When either the destination or some intermediate node moves, a special RREP is sent to the affected source nodes. Periodic hello messages can be used to ensure symmetric links, as well as to detect link failures. Alternatively, and with far less latency, such failures could be detected by using link-layer acknowledgements (LLACKS). A link failure is also indicated if attempts to forward a packet to the next hop fail.

Once the next hop becomes unreachable, the node upstream of the break propagates an unsolicited RREP with a fresh sequence number (a sequence number that is one greater than the previously known sequence number) and hop count of ∞ to all active upstream neighbors. Those nodes subsequently relay that message to their active neighbors and so on. This process continues until all active source nodes are notified; it terminates because AODV maintains only loop-free routes and there are only a finite number of nodes in the mobile ad hoc network.

Upon the receiving notification of a broken link, source nodes can restart the discovery process if they still require a route to the destination. To determine whether a route is still needed, a node may check whether the route has been used recently, as well as inspect upper level protocol control blocks to see whether connections remain open using the indicated destination. If the source node (or any other node along the previous route) decides it would link to rebuild the route to the destination, it sends out an RREQ with a destination sequence number of one greater than the previously known sequence number, to ensure that it builds a new, viable route, and that no nodes reply if they still regard the previous route as valid.

Chapter 3
Jointcount-based Multi-path Routing Protocol

In the previous chapter, the research approaches to improve performance of video streaming over mobile ad hoc networks are stated. Nevertheless, the smoothness of the video streaming using the stated multi-path approach is not satisfied. One possible major problem could be the signal interference. The Split Multi-path approach will use the two paths which are close to each other for transmitting the data packet. The signal interference will occur and make the degradation of the transmission. Therefore, the new multi-path approach is introduced in order to avoid the signal interference and the AODV routing protocol is used to do the experiment.

This chapter is organized as follows. In section 3.1, the concept of Ad hoc On-demand Distance Vector routing is described. The AODV-based multi-path routing protocol using jointcount is presented in section 3.2. Finally, the implementation of the multi-path approach is stated in the section 3.3.

3.1 AODV-based Multi-path Routing Protocol

The [12] has proposed the routing protocol uses a new method to find a pair of link-disjoint paths by selecting a route having a small number of common intermediate nodes on its path by using jointcount.

3.1.1 Design Principal

The [12] proposed a new AODV-based multi-path routing protocol for mobile ad hoc networks. Specifically, the routing protocol works based on the following primary design principles.

A. Setting up multiple reverse routes

To set up multiple reverse routes, the nodes initiate a route update process whenever they receive a RREQ. This process allows accepting alternative routes with the same or lower hop counts than the previous one.

B. Introducing a new method to find link-disjoint paths

The [12] proposed a new method for finding a pair of link-disjoint paths, which do not have any common link between the source and the destination. The common links are formed when multiple nodes use a common intermediate node, which has only one intermediate node to the destination. A path with more such common nodes shows that there are many opportunities to form the common links. To avoid having common links on the path, when multiple route choices are available the intermediate nodes select a route that has a smaller number of common nodes on the path than other routes. To count the number of common nodes on the path, the [12] added a new field called a jointcount to RREP for indicating the number of common nodes.

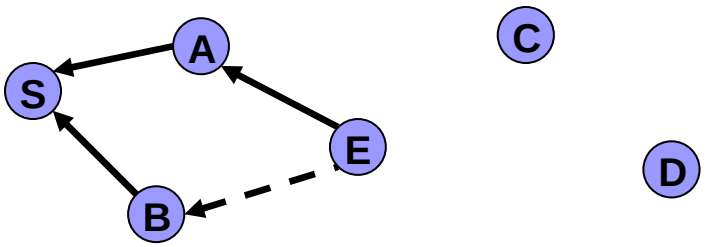


Figure 3.1: Example scenario for route discovery.

3.1.2 Setting up Multiple Reverse Routes

Based on the design principle A, the nodes invoke a route update process when they receive not only the first arriving RREQ, but also duplicate RREQs. Therefore, as in Figure 3.1, the intermediate node E invokes a route update process when it receives

duplicate RREQ from node B. The process allows accepting alternative routes using duplicate RREQs with the same or lower hop counts than the previous route, which was set up using the first arriving RREQ. The hop count of an alternative routing using the duplicate RREQ from node B is the same as that of the previous routes. Thus, node E adds a new reverse route using duplicate RREQ from node B. In this way, the node E can set up two reverse routes.

Although a method is also available for setting up many routes using any duplicate RREQ without the hop count restriction, this method causes higher data packet end-to-end delay because it allows a longer hop count.

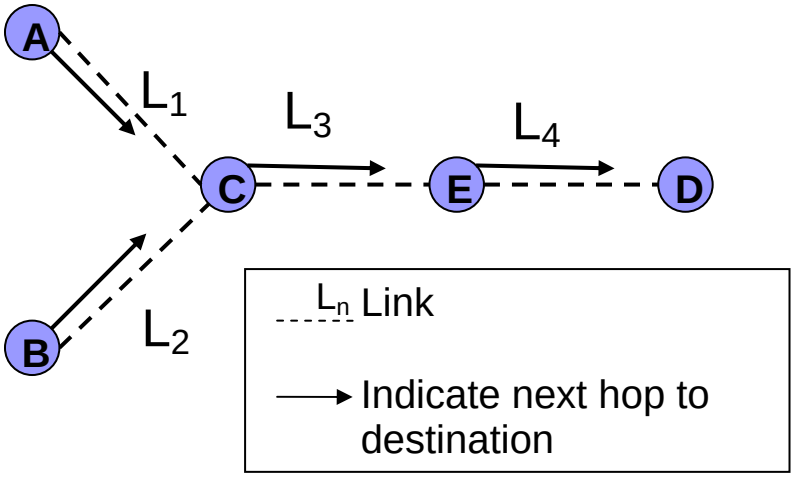


Figure 3.2: Example of causing common links.

3.1.3 Finding Link-Disjoint Paths

Based on design principle B, The [12] introduced a method for selecting a route. Generally, common links are formed when multiple nodes use a common intermediate node on the path and the common node uses only one intermediate node on the path. In Figure 3.2 for example, the node A sets up intermediate node C as the next hop on path to the destination D. Also, the node B sets up node C as the next hop. The node A and B use a common intermediate node C as the next hop on the path to destination node D, so the path from node A to node D and the path from node B to node D share links from the common intermediate node C to node D. Thus, a path with more such common nodes shows that there are many opportunities to share common links.

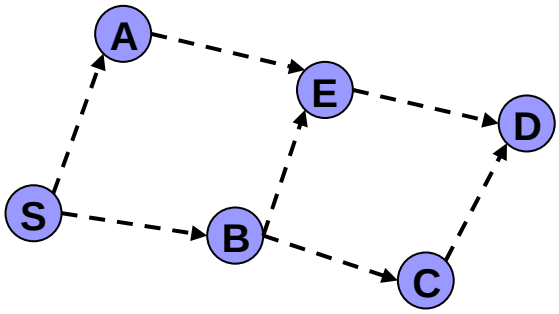


Figure 3.3: Available next hop on the path.

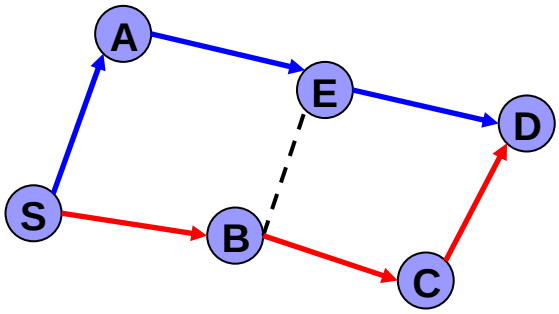


Figure 3.4: Link disjoint paths by selecting intermediate nodes.

For the reason mentioned above, the [12] suggest that the intermediate nodes select a route that has a small number of common nodes on its path when the multiple route choices are available. In figure 3.3, the node B has multiple routes to the destination D, so it selects a route C and does not select a route E, which has a common node E on the path. Nodes such as A and C have only one route, so they select that route. A series of selected routes offer a pair of link-disjoint paths between the source and destination as shown in figure 3.4.

3.1.4 Route Discovery of Proposed Routing Protocol

When a traffic source needs a route to the destination, it initiates a route discovery. An intermediate node receiving RREQs sets up multiple reverse routes. To find link-disjoint paths, each node performs a route selection method. Details of the route discovery are as follows:

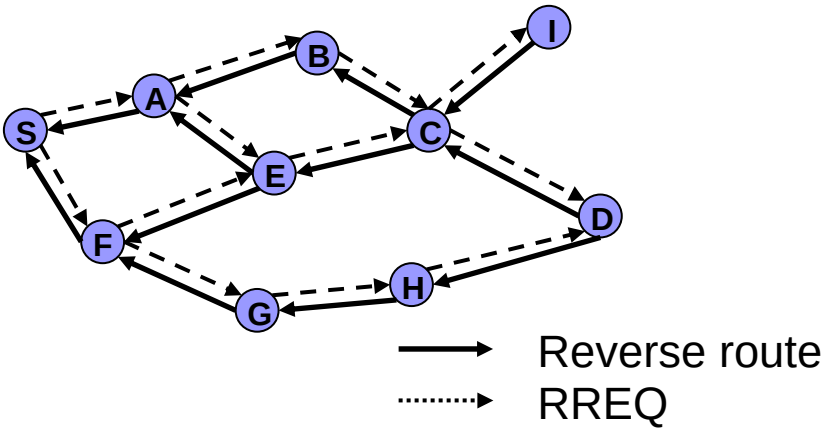


Figure 3.5: Multiple reverse routes set up using RREQ.

A. Setting up multiple reverse routes (Figure 3.5)

The source node S broadcasts a RREQ. An intermediate node receiving the first arriving RREQ, sets up a reverse route to the source. This is the same step as the AODV, which sets up a reverse route using the first arriving RREQ. However, unlike the AODV, the intermediate node may set up alternative routes using duplicate RREQs as described in section 3.1.2.

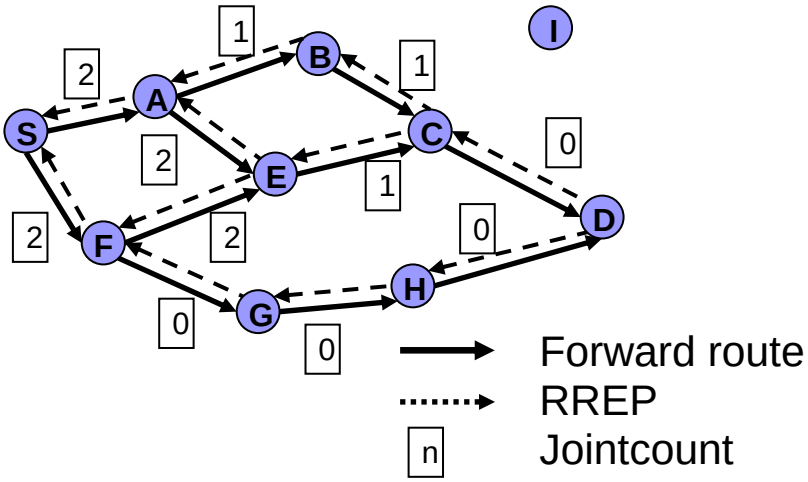


Figure 3.6: Multiple forward routes set up using RREP.

B. Setting up multiple forward routes (Figure 3.6)

When the destination node D receives the first arriving RREQ, it sets up a reverse route. It sends back an RREP to the source node S via the reverse route. Assume

that destination node D receives duplicate RREQs. If node D sets up a new reverse route using the duplicate RREQ on the basis of the hop count and sequence number, it then sends an RREP via the new reverse route. If node D does not set up a new reverse route, then it does not send the RREP. Each RREP carries a jointcount field introduced at the design principle B. At the destination node, the jointcount is always set to zero.

The intermediate node receiving the RREPs invokes a route update process. The route update process judges whether a new forward route using the RREP is added in the same way as the reverse route setup. If it allows accepting the new route, then the intermediate node forwards the RREP. When an intermediate node such as node C in Figure 3.5 has multiple reverse routes, it forwards the receiving RREP to each reverse route (to node B and node E). The nodes B and node E receive the RREP from the common intermediate node C, so they set up common intermediate node C as a forward route. This means that a common intermediate node is a node that forwards the receiving RREP to multiple reverse routes. Each node can therefore judge itself as the common node, if it has multiple reverse routes. To count the number of common nodes on the path, the jointcount of the RREP is increased by one at the common nodes. A non-common node, namely one that has only one reverse route, forwards an RREP without modifying the jointcount field of the receiving RREP. For example, in Figure 3.6, a common node C increases the jointcount (0) of the receiving RREP and forwards the RREP with jointcount (1) to B and E. In the same way, a common node E forwards an RREP with jointcount (2). An intermediate node such as node A receives the multiple RREPs. The intermediate node forwards the RREP that arrives first. The node may then receive additional RREPs. One trouble is that additional RREPs may have a different jointcount. Therefore, the node must be consistent regarding which one of these multiple jointcounts is notified to the others. The node must notify others of the greatest jointcount, because the notified jointcount can never be changed after this route discovery. The node therefore forwards an additional RREP only when the jointcount of the RREP is greater than that of the previous RREPs.

Since the process for the first arriving RREQ and the first arriving RREP is the same as that of AODV, the path first set up between S and D is the same as that of AODV.

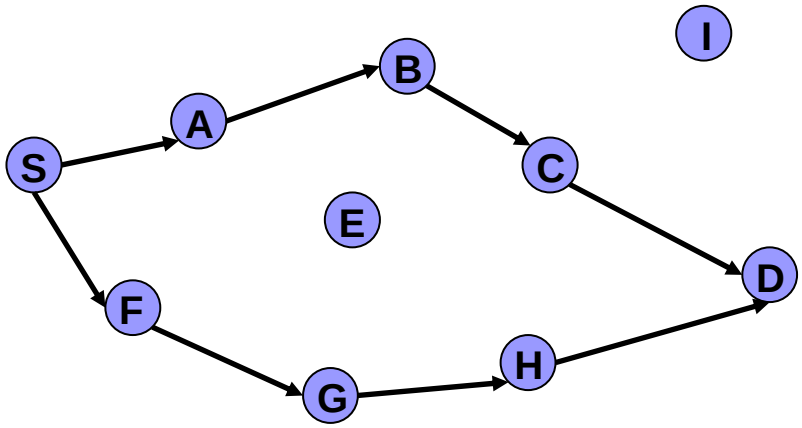


Figure 3.7: Route selection using jointcount.

C. Selecting link-disjoint paths (Figure 3.7)

To find a pair of link-disjoint paths, each node uses the route selection method which is described in section 3.1.3. Each node selects a route whose jointcount is smaller than the others when multiple route choices are available. If all routes have the same jointcount, the node selects a route whose hopcount is smaller than the others. If both the jointcount and hopcount are the same, then the nodes select a route whose lifetime is longer than the others. If the intermediate node has only one forward route, then it selects that route. This route selection is invoked when a new route is added or the primary route breaks. For example, since the jointcount of a forward route B is smaller than that of a forward route E, the node A selects the forward route B. In the same way, node B selects node C, node C selects node D. The pair of link-disjoint forward paths A-B-C-D and F-G-H-D is found on this way.

The source node S receives the first arriving RREP and sets up a forward route via the RREP. The source node S immediately sends data packets using the forward route without waiting for arrival of additional RREPs. Since the source node does not wait for the arrival of additional RREPs, the time from invoking the route discovery to the time first data packet is sent is the same as that of AODV. For example, in Figure 3.6, the node S receiving a RREP from node A sets up a forward route (in this scenario, the node S first receives a RREP from node A, and then receives it from node I). The node S then starts sending data packets are forwarded on the path S-A-B-C-D.

The source node S then receives an additional RREP from node F and adds a new forward route F as an alternative route. The path S-F-G-H-D is used when the primary path breaks. The source node S can send data packets using the alternative path. This alternative path does not use the same broken link as the primary path.

3.2 Implementation of Multi-path approach

The author would like to propose the algorithm of multi-path routing based on the jointcount value with more improvement and specification in detail. Here are the details of the improvement.

3.2.1 Setting up multiple routes

In order to improve the AODV to the multi-path AODV approach, the multiple routes should be prepared for selecting paths. The author would like to change the flow of the receive request packet from the figure 3.8 to the figure 3.9.

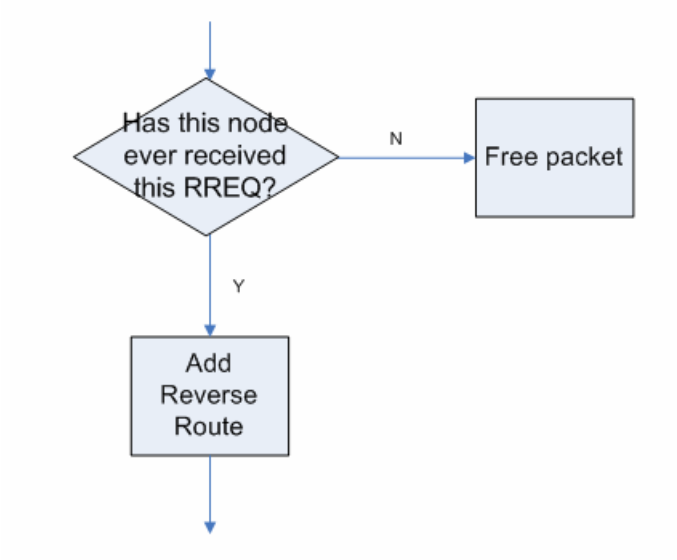


Figure 3.8: Flow of Receive Request method.

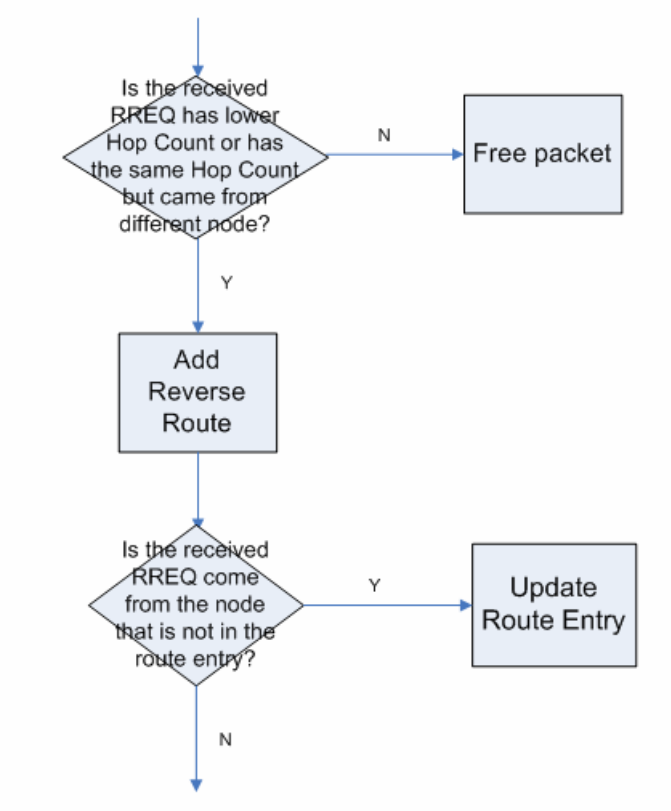


Figure 3.9: New Flow of Receive Request method.

The node has to accept the RREQ even though the RREQ has the same ID as the previous RREQ. If the new RREQ has a better route or have the same hop count as one in the route entry, the new route is accepted and is added as another reverse route.

The flow of the receive reply packet should be modified from the figure 3.10 to the figure 3.11.

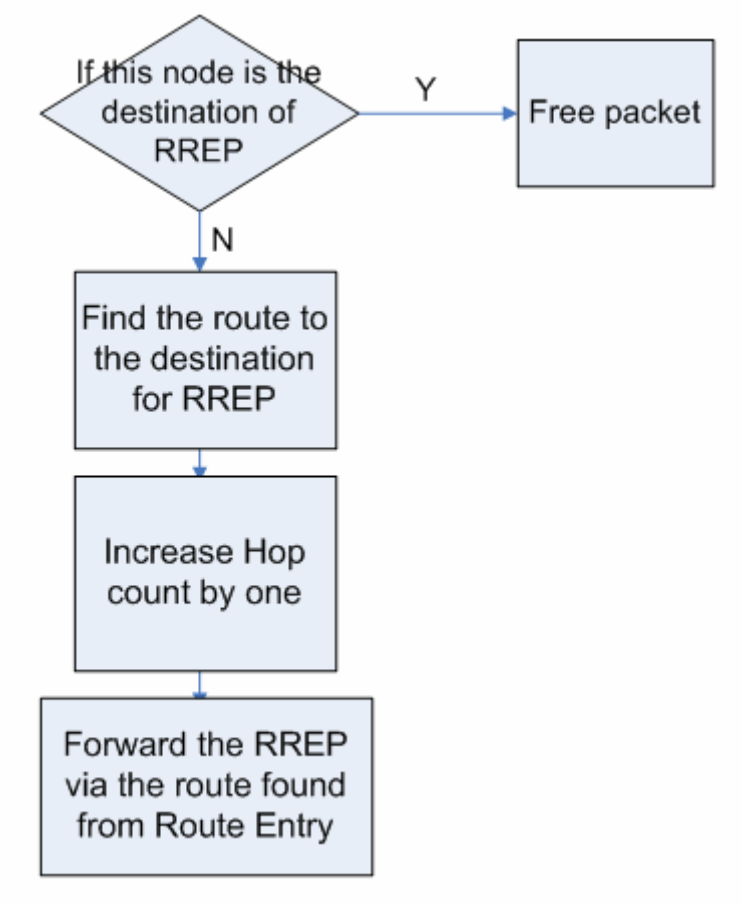


Figure 3.10: Flow of Receive Reply method.

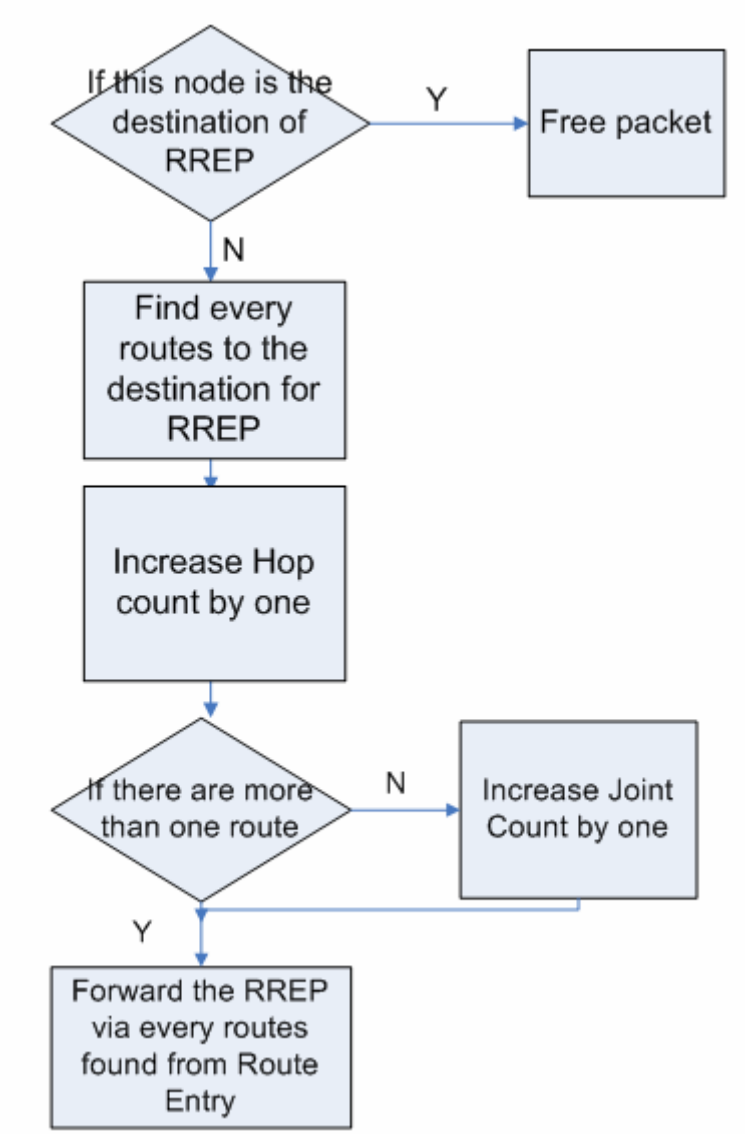


Figure 3.11: New Flow of Receive Reply method.

If the node accepts the new RREP packet, the node has to pick up every reverse route. If there is more than one reverse route, the jointcount value is increased by one and the node forwards the RREP packet to each reverse routes.

3.2.2 Selecting Link-Disjoint Paths

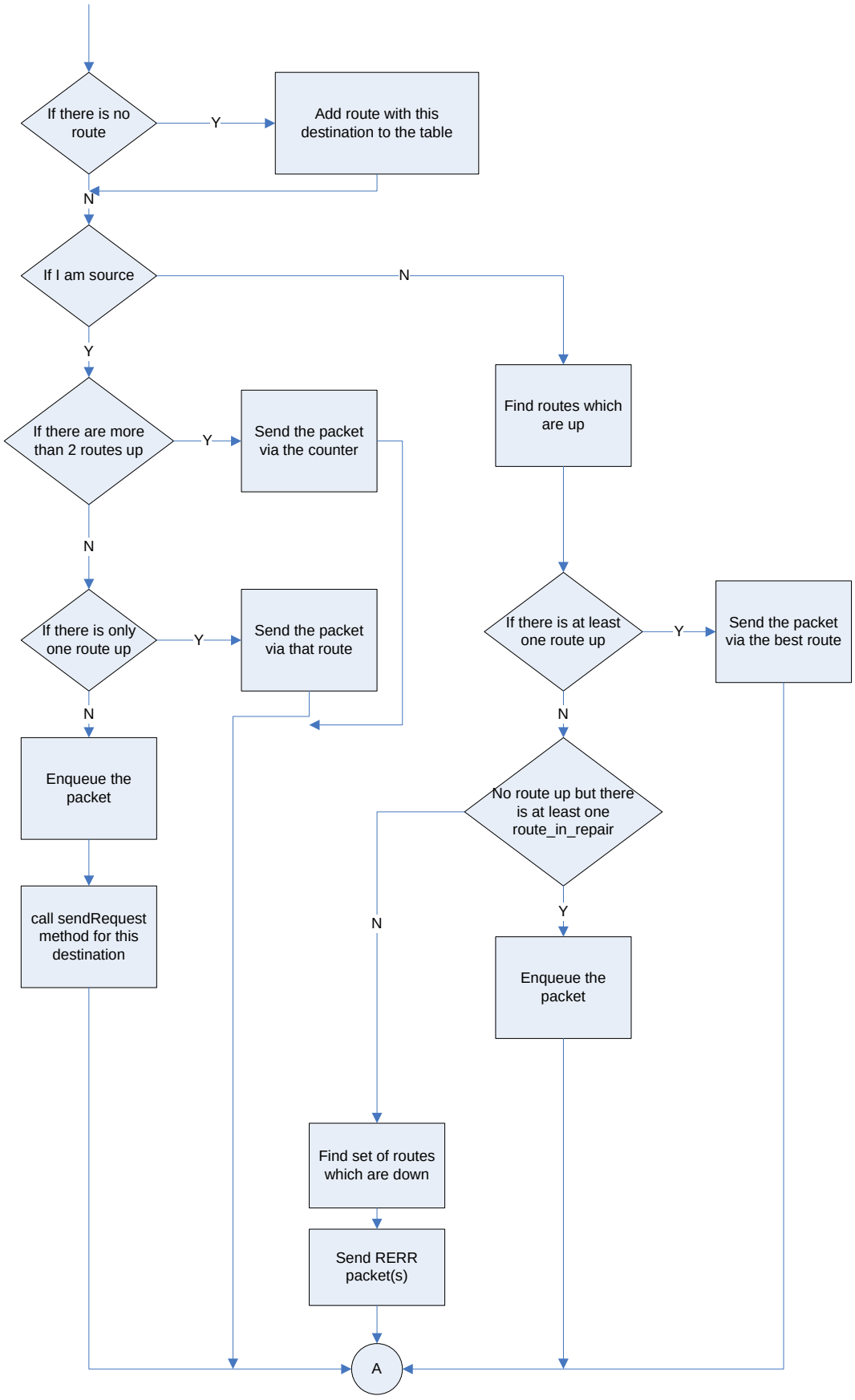

If there is no route
Y
Add route with this destination to the table
N
If I am source
N
Y
If there are more than 2 routes up
Y
Send the packet via the counter
N
If there is only one route up
Y
Send the packet via that route
N
Enqueue the packet
call sendRequest method for this destination
Find routes which are up
If there is at least one route up
Y
Send the packet via the best route
N
No route up but there is at least one route_in_repair
Y
Enqueue the packet
N
Find set of routes which are down
Send RERR packet(s)
A

Figure 3.12: New Flow of selecting paths.

If the source node has more than one path to the destination, the source node can switch from the first path to the second path and switch back every time the source node wants to transmit the packet to the destination. For the programming, the counter is added to switch the paths. The flow of the selecting path should be like figure 3.12. The intermediate node picks up only one best route and forward data packet via that route.

3.2.3 Jointcount

The proposed multi-path algorithm is based on finding the link-disjoint paths. So the jointcount value is very important for selecting the paths. The detail of adding the jointcount is described below.

A. Adding jointcount in the RREP packet

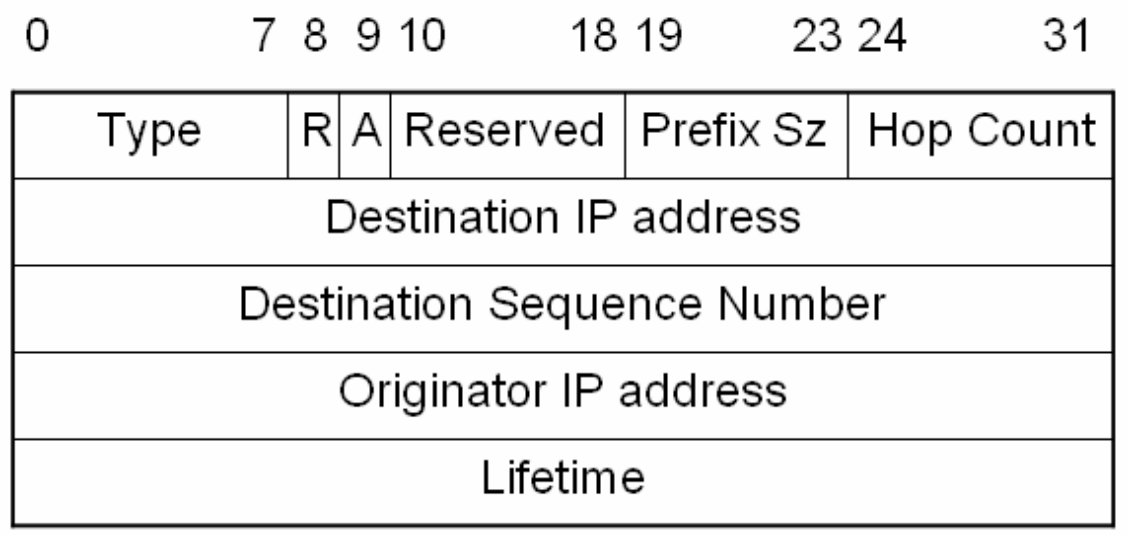


Figure 3.13: RREP packet structure.

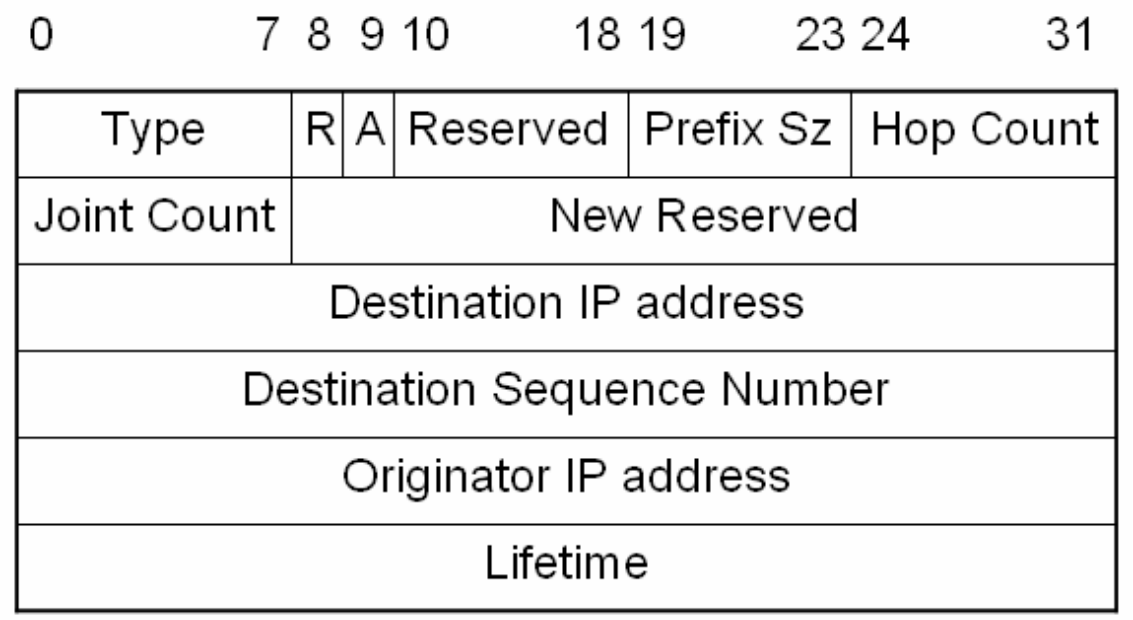


Figure 3.14: RREP packet structure with Jointcount.

The structure of the RREP packet is as shown in the figure 3.13. The author would like to add another 32 bits and use first 8 bits for jointcount value. The left is reserved for other useful objective (Figure 3.14).

B. Adding jointcount in the route entry

When the nodes receive the RREP, the nodes set forward route to the destination and this data is kept in the route entry of the route table. To select the link-disjoint paths, the jointcount value from the RREP should also be added to the route entry. Then the node can select the route to forward the packets from the jointcount value in the route entry.

C. Receiving RREP with different value of jointcount

If the intermediate node receives the RREP with the different value of the jointcount value in the route entry, the intermediate node should follow the below mechanism.

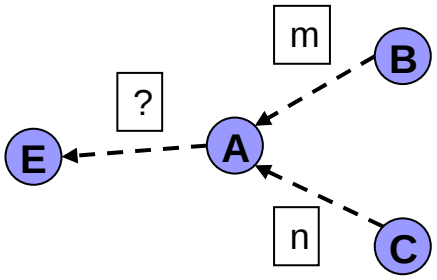


Figure 3.15: Receiving RREP with different value of Jointcount.

1) The RREP packet comes from different node (The intermediate node never receives RREP packet from this node).

The intermediate node A receives the first RREP from the node B with the jointcount value = m. Then the node A accepts the RREP and set the forward route to the node B. Node A forwards the RREP via the reverse route. Then node A receives the next RREP from the node C with the different value of jointcount.

If node A accepts the new RREP, node A updates the route entry as usual. Then before forwarding the RREP, the node checks every active route entries for the jointcount value. Find the max value of jointcount from the route entries (same destination). Then compare to the max jointcount with one in the RREP. If the max jointcount is greater than one in the RREP, put the max jointcount value in the RREP and then forward the RREP. The node must warn other nodes with the greatest value of jointcount. If the max jointcount is smaller than one in the RREP, just forward the RREP without changing the jointcount value.

2) The RREP packet comes from the same node (The intermediate node has received RREP packet from this node before.)

From the figure 3.15, the node E will receive the RREP two times from the node A. If node E accepts new RREP from node A, node E have to compare the jointcount value in the route entry and one in the RREP. If the jointcount value in the route entry is smaller than one in the RREP, node E updates the jointcount value in the route entry with the value from the RREP. Then node E forwards the RREP. If the jointcount value in the route entry is greater than one in the RREP, node E updates the jointcount value in the RREP before forwarding the RREP to the next node.

Chapter 4
Proposal of Multi-path Routing Protocol with Preemptive Technique for Video Streaming

The previous chapter describes the jointcount-based multi-path routing protocol. For this chapter, the author would like to propose a new approach which combines the advantage of multi-path routing and the preemptive technique together. The multi-path can provide more than one route to transmit the data packet. The preemptive technique can seamlessly switch from the current path which is likely to break soon to a new good path.

This chapter is organized as follows. The concept of the preemptive technique is stated in the section 4.1. Then the improvement to the preemptive technique and the detail about finding preemptive threshold are described in the section 4.2.

4.1 Preemptive Technique

Existing on-demand mobile ad hoc routing algorithms initiate route discovery only after a path breaks, incurring a significant cost in detecting the disconnection and establishing a new route. The preemptive technique [2] is the research to investigate adding proactive route selection and maintenance to on-demand mobile ad hoc routing algorithms. When a path is likely to be broken, a warning is sent to the source node indicating the likelihood of a disconnection. The source can then initiate path discovery

early, potentially avoiding the disconnection altogether. A path is considered likely to break when the received packet power becomes close to the minimum detectable power (other approaches are possible). Care must be taken to avoid initiating false route warnings due to fluctuations in received power caused by fading and similar random transient phenomena.

4.1.1 Preemptive Route Maintenance

A preemptive route maintenance algorithm [2] initiates recovery action early by detecting that a link is likely to break soon and finds and uses an alternative path before the cost of a link failure is incurred. This technique is similar to soft-handoff techniques used in cellular phone networks as mobiles move across cells [4]. Thus, the algorithm maintains connectivity by preemptively switching to a higher quality path when the quality of a path in use becomes suspect. More specifically, the algorithm consists of two components:

(i) Detect that a path is likely to be disconnected soon.

(ii) Find a better path and switch to it.

Similar to on-demand protocols, the preemptive technique replace path failure, with the likelihood of failure as trigger mechanism for route discovery. Although continuous update protocols could benefit from preemptive maintenance, their overhead is already too high and will only be increased from it.

A critical component of the proposed scheme is determining when path quality is no longer acceptable (which generates a preemptive warning). The path quality can incorporate several criteria such as signal strength, the age of path, the number of hops and rate of collisions. In [2], the path quality (and hence the preemptive warnings) is determined by a function of the signal strength of received packets with the number of hops being used as secondary measure. Since most breaks can be attributed to link failure due to node mobility in a typical mobile ad hoc scenario, the signal strength offers the most direct estimate of the ability of the nodes to reach each other. It is important that signal power fluctuations due to fading and similar temporary disturbances do not generate erroneous preemptive warnings.

In one-path routing algorithm, using preemptive route maintenance the cost of detecting a broken path (the retransmit/timeout time) is eliminated if another path is found successfully before the path breaks. In addition, the cost for discovering an alternative path is reduced (or eliminated) before the current path was actually broken. This can be expected to reduce the latency and jitter. Among the disadvantages, a higher number of path discoveries may be initiated since a path may become suspect but never break (for example, if the nodes change direction and move towards each other). However, if only high quality paths are accepted; they are likely to live longer reducing the number of re-discoveries needed.

4.1.2 Warn Packet Transmission

Based on the preemptive route maintenance algorithm, Figure 4.1 demonstrates the example of warning packet transmission.

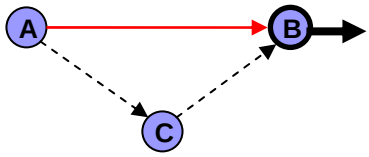


Figure 4.1: The movement of node from source node.

The source node A has the route to node B. Node A sends the data packets to the node B. Then node B moves away from the node A until the node B is in the preemptive region (the region before the transmission range of the node A) When the node B receives the data packet during the node B is in preemptive region, the received power of the packet will below the threshold. If the received power is below the threshold, the node B will send the warn packet back to the node which sent the last data packet (which in this case is the node A). When the node A receives warn packet from node B, the node A does as the node A receives route error packet. The node A does the re-discovery process to find the path to the destination. Then node A sends the data packet via new paths which node C is the intermediate node. These processes will occur before the node B moves away from the transmission range of node A. The node A will seamlessly switch to the new path before the current path breaks.

4.1.3 Preemptive Technique Experimental Study

The [2] has presented a class of algorithms that initiates proactive path switches when the quality of a path in use becomes suspect. The [2] showed that this proactivity avoids using a path that is about to fail and eliminates the associated cost of detecting the failure and recovering from it, significantly improving the performance of the network.

The [2] focused on signal power along each hop of the path as a measure of the quality of the path (a more robust definition of quality could include more factors such as the age of the path, number of hops, congestion). More specifically, using an estimate of the motion patterns of the time needed to complete a path query, and relating that time to the motion patterns of the nodes, the [2] derived a threshold on the signal power that will allow the nodes enough time to recover before the path gets disconnected. When a packet is received with a signal power below this threshold by a packet along a path, it generates a warning packet destined to the source of the path. The source then initiates a search for a higher quality path (a path where all the links are above the threshold) and immediately switches to it, avoiding a path break altogether.

As a case study in the [2], DSR and AODV were extended for proactivity. The Preemptive DSR demonstrated significant improvements over non-proactive DSR. But the Preemptive AODV demonstrated slightly improvements over non-proactive AODV. Both studies are based one the one-path routing algorithm. The author would like to use the preemptive technique in the multi-path routing algorithm.

4.2 Adding Preemptive Technique

As stated in the previous chapter, the preemptive technique is the technique to preemptively switch from the current path which is likely to break to a new good path. However, in the [2], the preemptive technique is used only in one-path routing. The author would like to use the preemptive technique in multi-path routing to improve the quality of the transmission.

4.2.1 Generating the Preemptive Warning

The preemptive warning is generated when the signal power of a received packet drops below a preemptive threshold. The value of this threshold is critical to the efficiency of the algorithm. If the value is too low, there will not be sufficient time to discover an alternative path before the path breaks. However, if the value is too high, the warning is generated too early with three negative side-effects:

- Unnecessary discover: the full life of the path currently use is not exploited. Likewise, the moving nodes may change direction and the current path never breaks, rendering the preemptive action an unnecessary overhead.
- This technique may force to accept a path of a lower quality than the one which is currently used.
- Increasing the preemptive threshold effectively limits the range of the mobiles (a smaller range is now acceptable without generating a preemptive warning).

If the threshold is too high, false connection can occur. Generating the preemptive warning is complicated due to fading that can cause sudden variations in the received signal power.

4.2.2 Preemptive Region

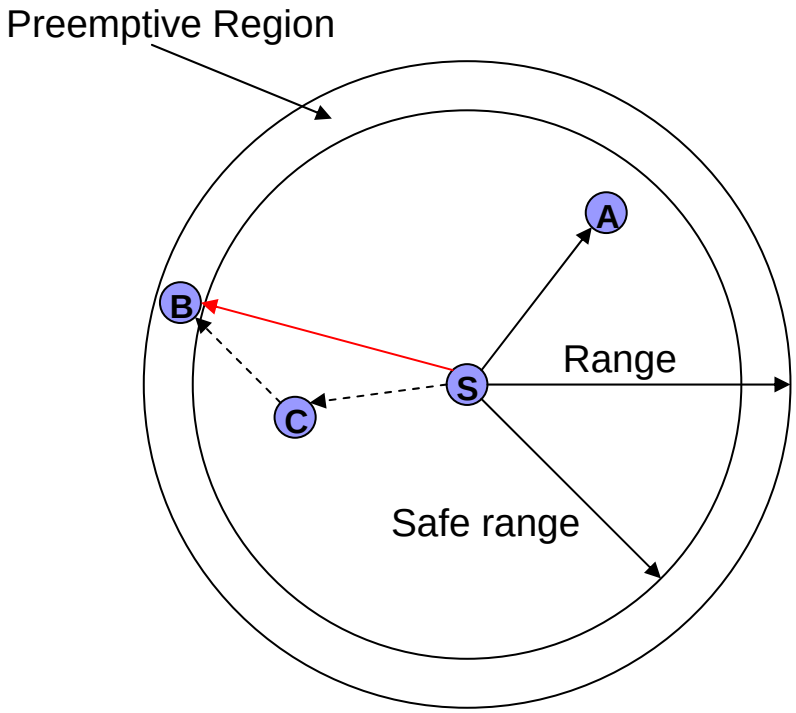


Figure 4.2: Preemptive Region.

Figure 4.2 demonstrates the preemptive region around a source. For example, as node B in the figure enters this region, the signal power of received packets from the source S falls below the preemptive threshold, generating a warning packet to node S. The node S initiates route discovery action, and discovers a route through node C. Node S switches to this route avoiding the failure of the path as node B moves out of direct range of node S.

The recovery time from a broken path, $T_{recover}$, depends on the size and topology of the network, as well as the path being recovered. However, the author assumes that each node keeps a running estimate of this value. The optimal value for the signal threshold will warn the source $T_{recover}$ seconds before the path breaks; this allows just enough time to discover a new path. Hence, the warning interval T_w (which is the time between a warning and a break) should be set to $T_{recover}$.

Given two mobile nodes with a vector distance X between them, moving with vector speeds, V_1 and V_2, the distance between the two nodes is $X+t(V_2-V_1)$. The time until the absolute distance between them becomes greater than the range of source is a function of their relative location and velocity. In the worst case the sources are moving

at their maximum speeds away from each other. This case can be used to derive a conservative estimate on the preemptive region.

Given a typical land-based network where the maximum speed of a node is 5 m/s and a recovery time estimate of 0.1 sec (this is derived empirically, and in the protocol would be based on a running history estimate). The preemptive region would start 1 meter from the maximum range; even if the two nodes are moving away from each other at maximum speeds (combined 10 m/s), the 1 meter distance will give the source 0.1 second necessary to find a new path.

4.2.3 Generating the Preemptive Warning

Because an explicit estimate of the preemptive region requires the nodes to exchange location and velocity information, the power of received packets should be used to estimate the distance between them. The recovery time can be related to the power threshold as follows. This research considers devices operating in the ISM bands (such as Lucent WaveLANs). The signal power drops is

$$P_r = \frac{P_0}{r^n} \tag{4.1}$$

At a distance r from the transmitter, where P_0 is the transmitted power and n is typically between 2 and 4.

The signal power at any point is the sum of the main signal transmitted by the antenna in addition to components of the signal that reflect off-of the surrounding features. In open environment, the main secondary component is the strong reflection of the transmitted signal from the ground. Equation (1) represents an approximate (and idealized) model for the channel with n=2 near the source until a certain point where n becomes 4.

The author would like to assume the $1/r^4$ drop in signal power throughout the preemptive region (since the preemptive region is near the maximum range of the devices). More specifically,

$$P_{received} = \frac{P_0}{r^4} \tag{4.2}$$

P_0 is a constant for each transmitter/receiver pair, based on the antenna gain and height. The minimum power receivable by the device is the power at the maximum

transmission range, P_{range} is $P_0/range^4$. This value is the characteristic of the device (For example 3.65×10^{-10} Watts for WaveLANs [14]). Similarly, the preemptive signal power threshold is the signal power at the edge of the preemptive region. In addition, for a preemptive region of width of *w*, the signal power threshold is

$$P_{threshold} = \frac{P_0}{r^4_{preemptive}} \tag{4.3}$$

The $r_{preemptive}$ is equal to *range-w* where *w* = relative_speed x T_w. The preemptive ratio, δ is defined as

$$\delta = \frac{P_{threshold}}{P_{range}} = \frac{\frac{P_0}{(range - w)^4}}{\frac{P_0}{range^4}} = (\frac{range}{range - w})^4 \tag{4.4}$$

For example, WaveLAN cards have a range of 250 meters in open environments in the 900 MHz band [14]. The preemptive ratio for a preemptive region of width 1 meter is

$$(\frac{250}{250 - 1})^4 = 1.016$$

This value corresponds to a signal of threshold of 1.016 x P_{range} = 3.71 x 10^{-10} Watts.

4.2.4 Preemptive Technique in Multi-path

Although the preemptive technique warn the source node before the path break occurs, the source node has to do the rediscovery process in order to find the new good path to the destination. The author would like to use the preemptive technique in the multi-path routing. Several intermediate nodes between the source and the destination provide multiple forward routes to the destination. If the forward routes are not in used, the routes will be timeout and cannot be used. However, if the intermediate nodes which hold multiple forward routes receive the preemptive warn packet from the next node before the other routes are timeout, those intermediate nodes can simply change to another forward route. These intermediate nodes no need to forward the preemptive warn packet to the source node. This approach will reduce the rediscovery process and decrease the congestion in the mobile ad hoc networks.

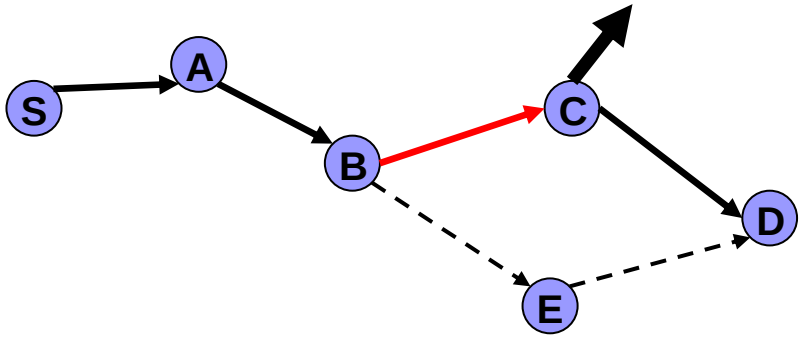


Figure 4.3: Changing Path before to destination.

For example in the figure 4.3, source node S has a path to the destination D which is S-A-B-C-D. Node B has two active routes to the destination node D. Node B can set up two forward routes because the algorithm of the multi-path. As in the figure 4.3, node C is moving away from the transmission range of node B. When the node C move away until reach the preemptive region, node C will send the warn packet to the node B. If the node B holds other active routes to the destination (which in the figure is node E), node B just set the current route as down route and switch the current route to the other active rote. Node B do not have to forward the warn packet.

4.2.5 Warn Packet

The Warn packet should be created as another AODV packet. The Warn packet is sent to warn the previous about the path is likely to break soon. The structure of the Warn packet is as shown in the figure 4.4.

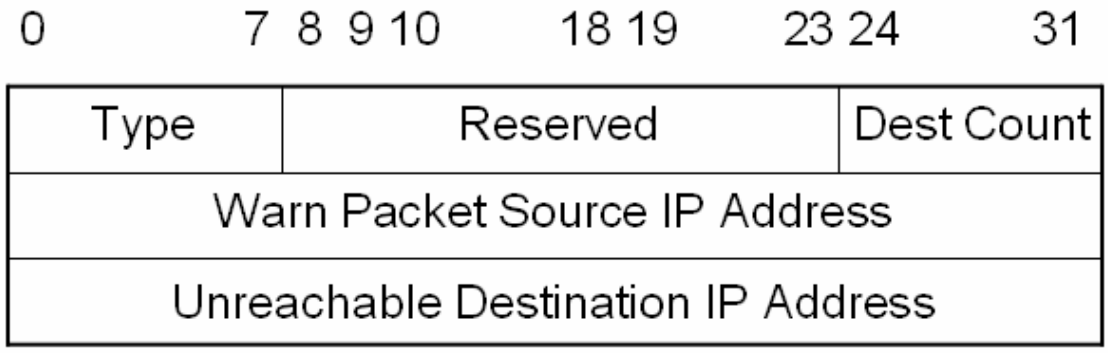


Figure 4.4: Warn packet structure.

Chapter 5
Simulation Performance Evaluation

The previous chapter presented the proposal of multi-path routing protocol with preemptive technique for video streaming over ad hoc networks. This chapter describes the simulation performance evaluation. Performance studies of the proposal were conducted using Network Simulator ns-2 [13]. The reason why the Network Simulator ns-2 was chosen is that, in recent years, the ns-2 is widely used for performance evaluation in ad hoc network field. Moreover, the ns-2 is always improved from the developers that make the simulator always up-to-date.

This chapter is organized as follows. The network simulator ns-2 and its merit are briefly described in section 5.1. Then, necessary modification in source code is explained in section 5.2. Section 5.3 explains about simulation conditions. Finally, section 5.4 discuss about the simulation results.

5.1 Network Simulator ns-2

An extended version of UCB/LBNL network simulator ns-2 is used for experimental evaluation. Ns-2 is a discrete event simulator that is developed as part of VINT project at the Lawrence Berkeley National Laboratory. The extensions implemented by the CMU Monarch project enable it to simulate mobile nodes connected by wireless network interfaces. The simulator provides substantial support for simulation of AODV routing over wireless networks. Moreover, the simulator allows the author to improve the protocol and then compare its performance results with those of the conventional protocol in the same condition. To run a simulation, a script

written by TCL is needed. In the TCL script, the nodes and links have to be set. Then, attach agents to the nodes. After that, the traffics are attached to the agents as well. The simulation allows the author to set time for traffic to start and stop. Finally, simulation is run by using command "ns <filename.tcl>".

5.1.1 AODV Module

The simulator contains AODV's modules. The AODV's related files are as follows.

- aodv.h and aodv.cc contain main mechanism of AODV Routing.
- aodv_packet.h contains AODV packet types.
- aodv_rqueue.h and aodv_rqueue.cc contain packet queue for routing protocol.
- aodv_rtable.h and aodv_rtable.cc contain route entry details.

The default value of parameters for each agent is set in ~ns/tcl/lib/ns-default.tcl. In the following experiments, the author makes a comparison between the results of One-path AODV and Multi-path AODV, with and without Preemptive Technique.

5.1.2 Related files for adding preemptive

As the author has mentioned in the section 4, the power capacity is used to be threshold for each node in the mobile ad hoc networks. The wireless-phy.h and wireless-phy.cc provide the power of the received packet, so the preemptive technique codes can be added in these files.

5.1.3 Trace and Monitoring

The trace support for wireless simulation currently uses cmu-trace objects. The cmu-trace objects are of three types- CMUTrace/Drop, CMUTrace/Recv, and CMUTrace/Send. These are used for tracing packets that are dropped, received and sent by agents, routers, mac layers or interface queue in ns. An example of the new trace format is shown below.

s –t 0.267662078 –Hs 0 –Hd -1 –Ni 0 –Nx 5.00 –Ny 2.00 –Nz 0.00 –Ne -1.000000 –Nl RTR –Nw --- -Ma 0 –Md 0 –Ms 0 –Mt 0 –Is 0.255 –Id -1.255 –It message –Il 32 –If 0 –Ii 0 –Iv 32

The trace format as seen above can be divided into the following fields:

- **Event type** In the traces above, the first field describes the type of event taking place at the node can be one of the four types:

  s send

  r receive

  d drop

  f forward

- **General tag** The second field starting with "–t" may stand for time or global setting.

  -t time

  -t * (global setting)

- **Node property tags** This field denotes the node properties like node-id, the level at which tracing is being done like agent, router or MAC. The tags start with a leading "-N" and are listed as below:

  -Ni: node id

  -Nx: node's x-coordinate

  -Ny: node's y-coordinate

  -Nz: node's z-coordinate

  -Ne: node energy level

  -Nl: trace level, such as AGT, RTR, MAC

  -Nw: reason for the event. The different reasons for dropping a packet are given below:

  "END" DROP_END_OF_SIMULATION

  "COL" DROP_MAC_COLLISION

  "DUP" DROP_MAC_DUPLICATE

  "ERR" DROP_MAC_PACKET_ERROR

  "RET" DROP_MAC_RETRY_COUNT_EXCEEDED

“STA” DROP_MAC_INVALID_STATE

“BSY” DROP_MAC_BUSY

“NRITE” DROP_RTR_NO_ROUTE for example, no route is available.

“LOOP” DROP_RTR_ROUTE_LOOP for example, there is a routing loop.

“TTL” DROP_RTR_TTL for example, TTL has reached zero.

“TOUT” DROP_RTR_QTIMEOUT for example, packet has expired.

“CBK” DROP_RTR_MAC_CALLBACK

“IFG” DROP_IFQ_QFULL for example, no buffer space in IFQ.

“ARP” DROP_IFQ_ARP_FULL for example, dropped by ARP.

“OUT” DROP_OUTSIDE_SUBNET for example, dropped by base stations on receiving routing updates from nodes outside its domain.

- **Packet information at IP level** The tags for this field start with a leading “-I” and are listed along with their explantions as following:

-Is: source address.source port number

-Id: destination address.destination port number

-It: packet type

-Il: packet size

-If: flow id

-Ii: unique id

-Iv: ttl value

- **Next hop info** This field provides next hop info and the tags starts with a leading “-H”.

-Hs: id for this node

-Hd: id for next hop towards the destination

- **Packet info at MAC level** This field gives MAC layer information and starts with a leading “-M” as shown below:

-Ma: duration

-Md: destination’s Ethernet address

-Ms: source's Ethernet address

-Mt: Ethernet type

This trace file is used for tracing AODV as well as multi-path AODV traffic throughout the experiments.

5.2 Necessary Modification in Source Code

The ns-2 network simulator contains implementations of the AODV protocol. The author implements four types of the simulations which needs to modify the code as below.

5.2.1 Multi-path code

To improve the normal AODV to the multi-path AODV, the parts of the codes need to be modified is as follow.

- In the route table file, create a new type of object called multiple route entry. The multiple route entry is an array to keep route entries. The objective of multiple route entry is to keep the routes to the same destination in the array.
- Every method in the AODV main file should change "finding route to the destination" to "finding multiple routes" to the destination.
- The receive request method should be modified to receive the RREQ with the same ID as previous one in order to create the multiple reverse routes.
- In the receive request method, do not forget to add precursor list to every routes in the multiple route entries.
- The receive reply method should be modified to accept the multiple route reply to create the multiple forward routes.
- The receive reply method should be modified to forward RREP packet to every reverse routes
- Add the jointcount field in the RREP packet and route entry file.

- The receive reply method should be modified to upgrade the value of jointcount in the RREP packet.
- The receive error method should be modified to check if the node still has another active route to the destination. If the node still has another active route to the destination, the node no needs to forward the RERR packet.
- In the route resolve method, the flow of the program should be changes as shown in the figure 3.12 in the chapter 3.
- Route resolve method for source node should be set to switch from one active path to another active path and switch back in next transmission. (The 2 best active paths will be used to transmit the data packet). Create counter for switching the paths.
- The set of the route selector counter should be added for the every node in case of one source may have to transmit to more than one destination. The route selector counter is for the source node to switch from the best route to the second route in the next transmission and switch back in the next transmission.

5.2.2 Preemptive code

To add the preemptive technique to the normal AODV, the parts of the codes need to be modified is as follow.

- Create a Warn packet as a new AODV type packet.
- In the wireless-phy file, at the receive method add the condition to check for the received packet power, if the power is under threshold create Warn packet and then send the packet.
- Create receive method for receive warn packet. If the node receive warn packet, do as the node receive the RERR packet.

5.3 Simulation Condition

The author would like to implement the code as described above. Table 5.1 shows parameters that are constantly set for all simulation.

Table 5.1: Evaluation condition for the simulation

Nodes	30 Nodes
Area	800m x 800m
Channel Capacity	2 Mbps
Data Packet Size	1000 bytes
Maximum number of packets in buffer	64 packets
Node's transmission range	250 meter
Simulation Running Time	300 seconds

5.3.1 General Setting

The IEEE802.11 Distributed Coordination Function is used as a Medium Access Control (MAC) protocol. The mobility model uses the random way point model. Nodes move randomly within the field. A node starts its journey from a random location to a random destination at a randomly chosen speed. After it reaches its destination, another random destination is targeted after a pause. The author considered the case of continuous mobility (no pauses). To change node mobility, the author varies the maximum speed of the nodes which is 0, 5, 10, 15 and 20 m/s. Traffic sources are continuous bit rate (CBR). Use UDP as the transport layer. The number of connections is three. The main transmission is in the time interval [0, 300] s. The other is transmitted in the time interval [100, 200] s and [150, 250] s. The traffic rate is 12.5 packets per second (100kbps).

5.3.2 Preemptive Threshold Setting

As stated in the chapter 4, the preemptive threshold value depends on the $T_{recovery}$ (The recovery time from a broken path). This value can be found from the experiments. The author has simulated the network and calculated the average $T_{recovery}$ of each node max speed case. Table 5.2 shows the value of preemptive threshold for each node max

speed case in one-path routing. The Table 5.3 shows the value of preemptive threshold for each node max speed case in multi-path routing.

Table 5.2: Preemptive Threshold for each case in one-path routing

Node Max Speed (m/s)	$T_{recover}$ **(s)**	**Preemptive Region Width (m)**	**Preemptive Threshold (Watt)**
5	0.0187	0.187	3.663×10^{-10}
10	0.0195	0.390	3.675×10^{-10}
15	0.0207	0.621	3.689×10^{-10}
20	0.0221	0.884	3.704×10^{-10}

Table 5.3: Preemptive Threshold for each case in multi-path routing

Node Max Speed (m/s)	$T_{recover}$ **(s)**	**Preemptive Region Width (m)**	**Preemptive Threshold (Watt)**
5	0.0189	0.189	3.663×10^{-10}
10	0.0198	0.396	3.675×10^{-10}
15	0.0210	0.630	3.689×10^{-10}
20	0.0231	0.924	3.706×10^{-10}

These preemptive threshold values will be use in the preemptive simulation cases.

5.4 Simulation Result and Discussion

The experiment divides the condition of routing into the following cases.

- One-path approach without preemptive technique
- One-path approach with preemptive technique
- Multi-path approach without preemptive technique
- Multi-path approach with preemptive technique

The author would like to compare between one-path, multi-path and preemptive, non-preemptive approach. The experiment will focus on the average throughput of the simulation, the average delay of the data packets, the route discovery frequency, the average hop count and the number of warn packets.

The preemptive simulations use the same condition as the non-preemptive case, except adding the preemptive threshold and transmitting warn packets. In the case of node max speed = 0m/s (no mobility), there is no need to set the threshold because the nodes do not move away from each other.

Each data point represented an average of four different scenarios.

5.4.1 Simulation Result

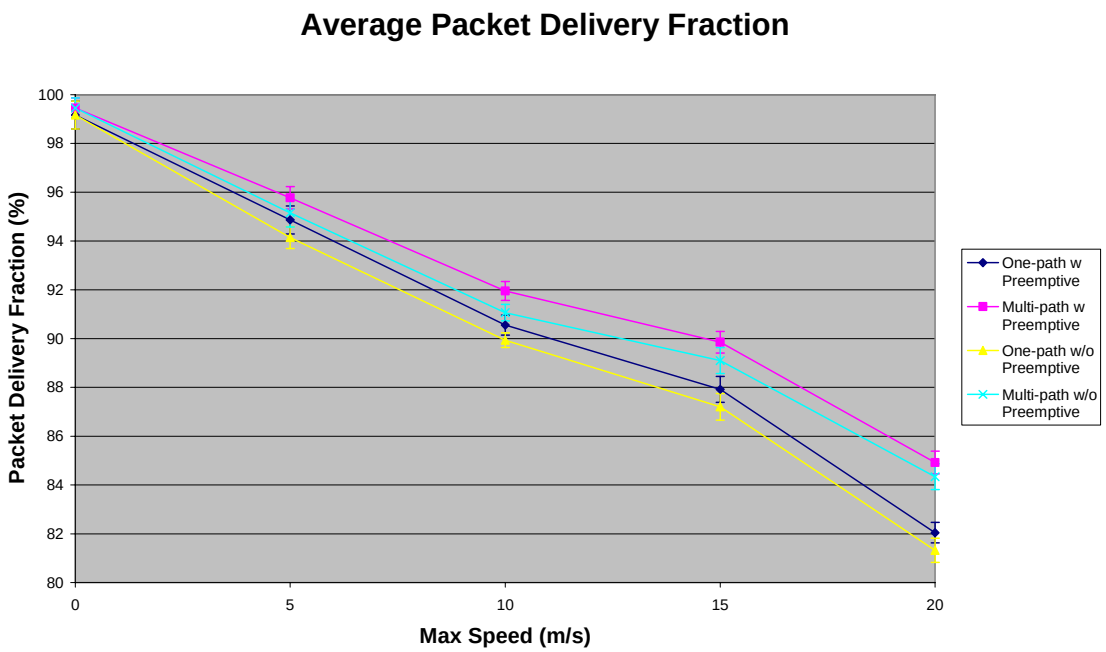


Figure 5.1: Average Packet Delivery Fraction.

Figure 5.1 shows the data packet delivery fraction of each routing approach. Results show that the multi-path routing approach outperforms the one-path routing approach.

Compare between preemptive and non-preemptive approach. The preemptive approach is slightly better than the non-preemptive approach in both one-path and multi-path.

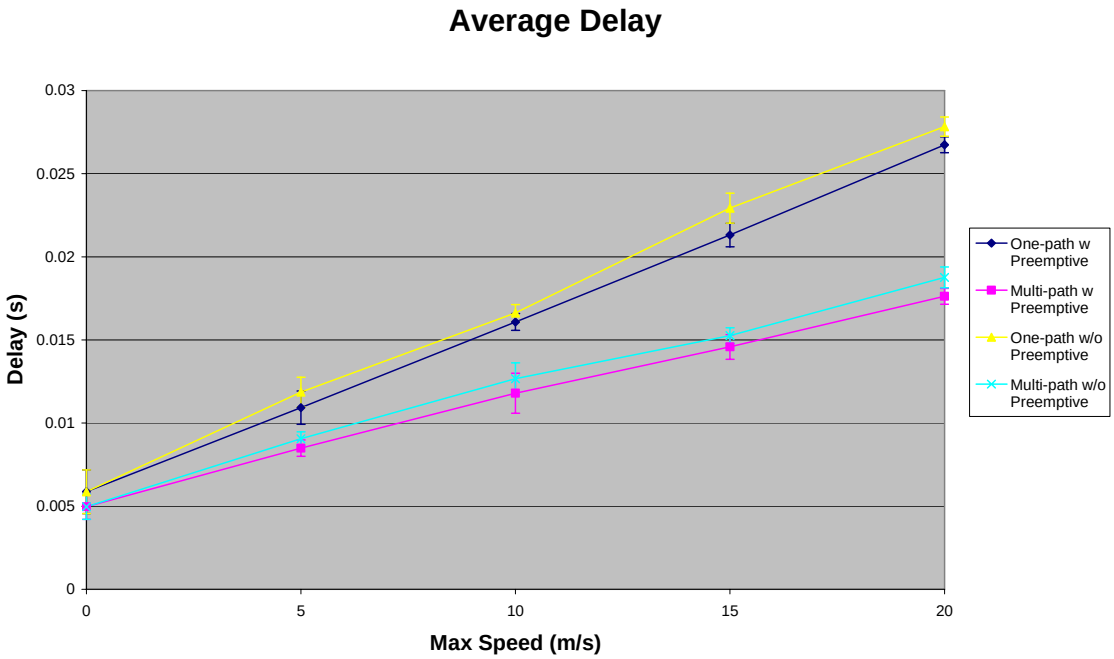


Figure 5.2: Average Delay.

Figure 5.2 shows the average delay of each routing approach. The proposed multi-path approach also exhibits the smaller end-to-end delay than the one-path approach. For example, when the node maximum speed is 10 m/s, the delay of one-path AODV is 0.017 seconds. The delay of proposed multi-path approach is 0.012 seconds. The proposed protocol decreases the delay by 29 %.

Compare between preemptive and non-preemptive approach. The preemptive approach slightly reduces average delay than the non-preemptive approach in both one-path and multi-path. Even though the warn packets are sent, the packet sizes are small which do not affect the other transmission.

The figure 5.2 also shows the standard deviation for each approach. The standard deviation of both one-path and proposed multi-path is almost the same. That means the average delay of the multi-path approach has almost the same range with the one-path approach.

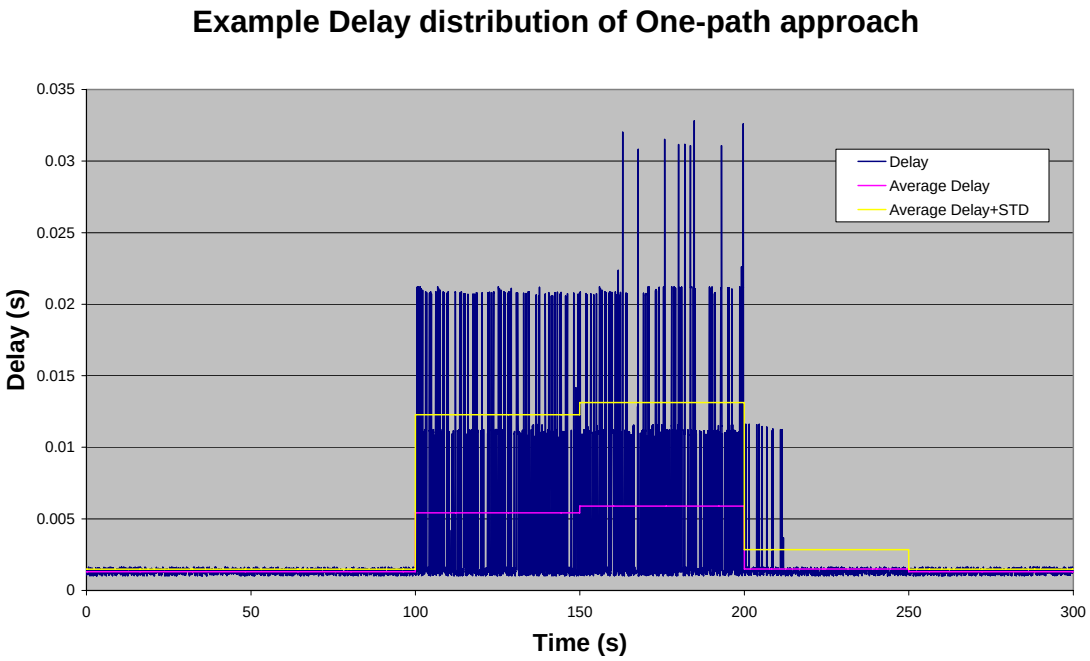


Figure 5.3: Example of Delay distribution of One-path approach.

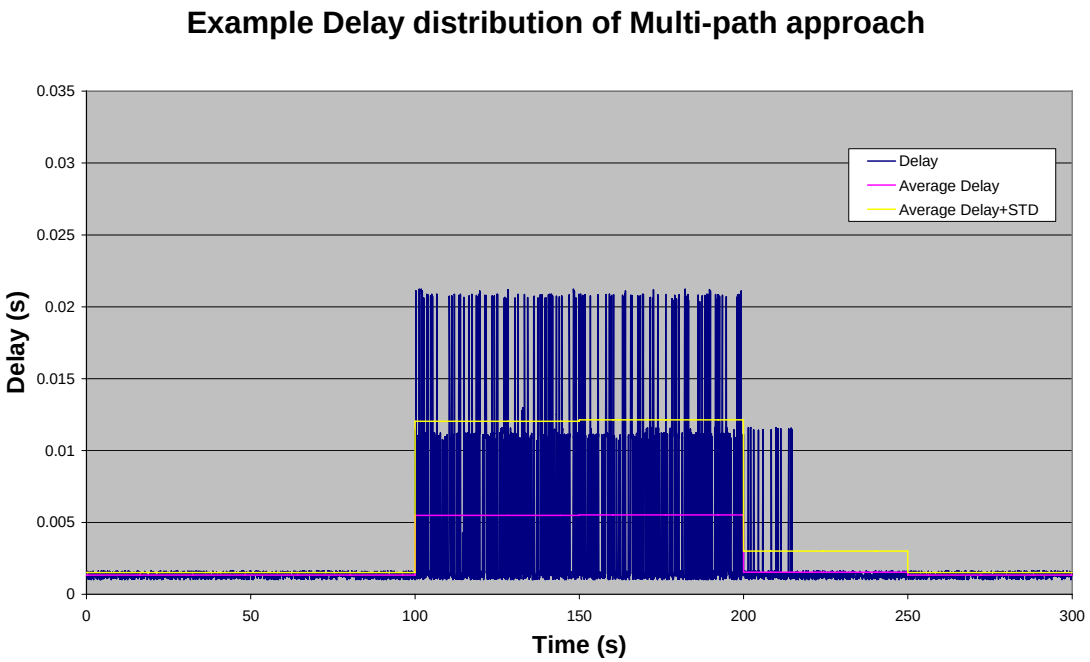


Figure 5.4: Example of Delay distribution of Multi-path approach.

Figure 5.3 shows the example of the delay distribution of one-path approach with the preemptive technique by picking from the node max speed=5m/s. The delay is

from the connection [0, 300] second. Figure 5.4 shows the example of the delay distribution of multi-path approach with the preemptive technique from the same case. In the duration 150 to 200 second which there are three connections in the simulation, multi-path approach can yield the delay smoother than the one-path approach.

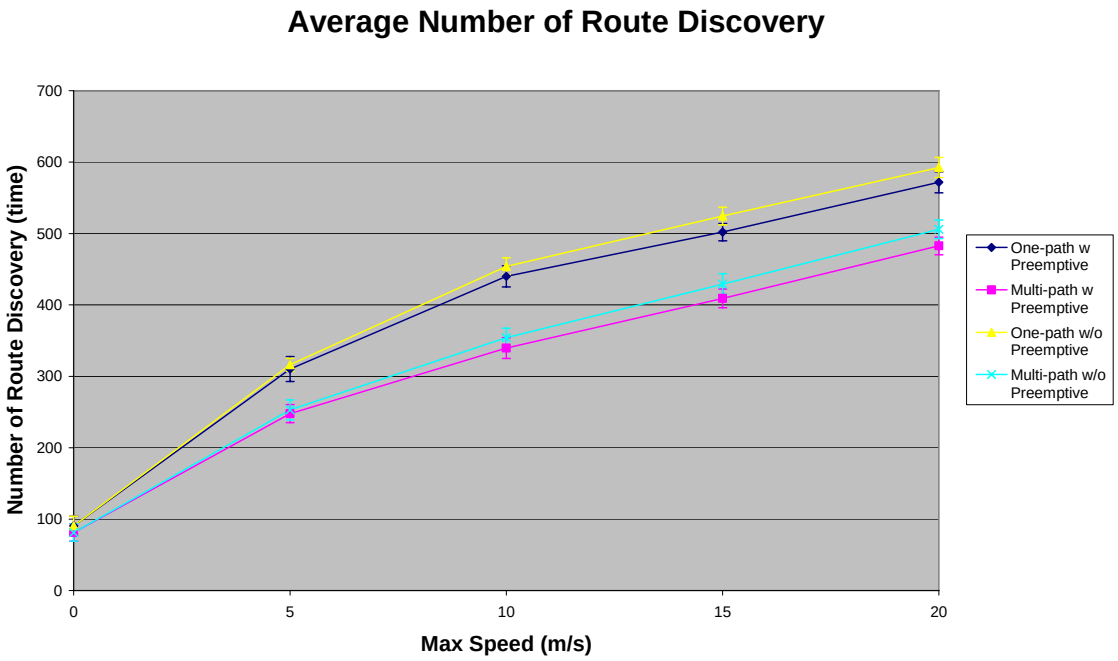


Figure 5.5: Average Number of Route Discovery.

Figure 5.5 shows the average number of route discovery of each routing approach. The proposed multi-path approach has a smaller number of route discoveries than the one-path approach. The preemptive approach still overcomes the non-preemptive approach. This may implies that the preemptive technique can help protect path break before the break occurs.

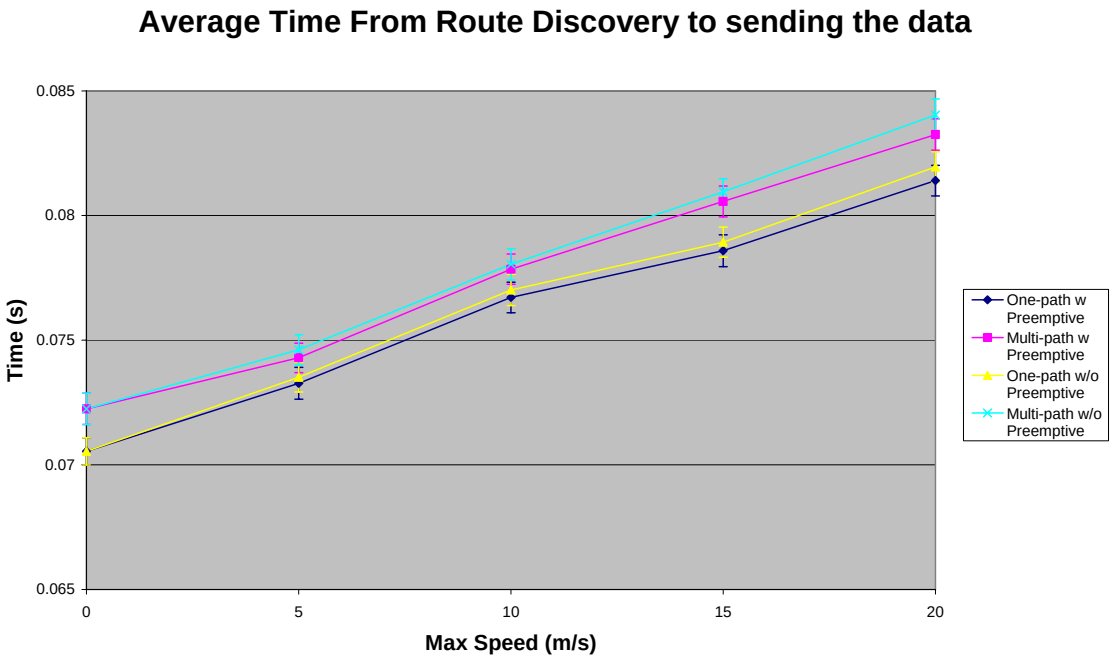


Figure 5.6: Average Time Interval from Route Discovery to sending the data packet.

Figure 5.6 shows the average time interval from route discovery to sending the data packet. The proposed multi-path approach takes a longer time interval than the one-path approach. The multi-path approach may take longer time than the one-path approach because of setting multiple reverse routes. The RREQ packet and RREP packet will be more than the one-path approach. Therefore, the RREP may take longer time to come to the source node.

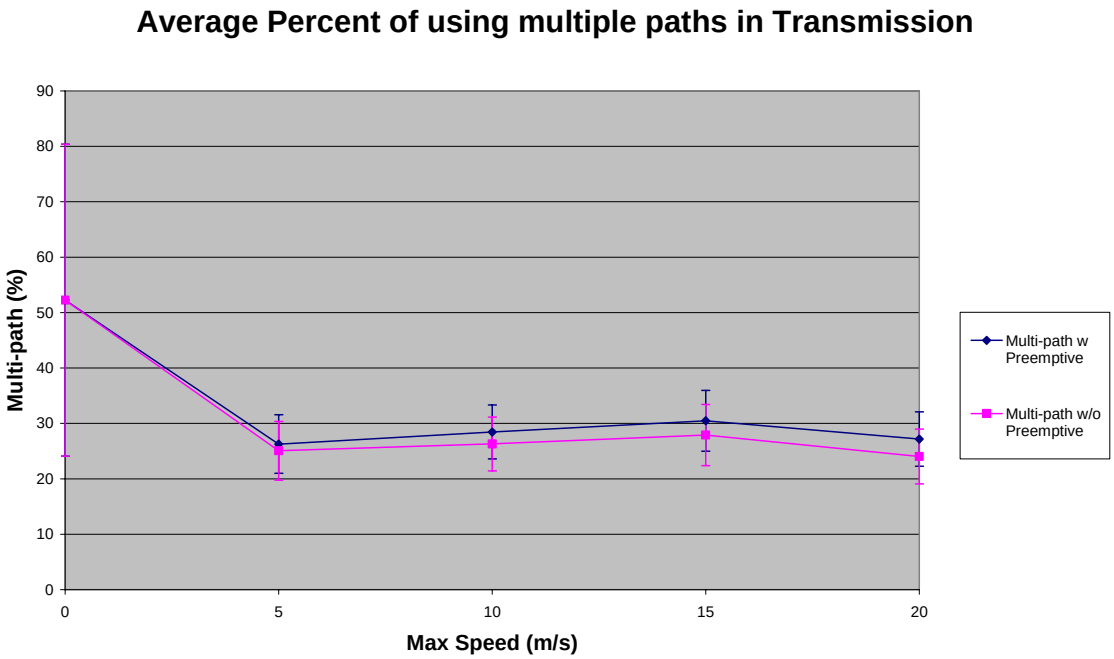


Figure 5.7: Average Percent of using multiple paths in Transmission.

Figure 5.7 shows the percent of using multiple paths in data transmission. These values are evaluated from the source nodes when the source nodes have multi-path to the destinations compare to all of the transmission. The figure 5.7 shows that the multi-path approach can find the multiple paths in the topologies. In no mobility case, the average percent of multi-path is higher than the mobility case because if the source node can find the multiple paths, the paths will not be break. However, if other transmission is added, the path may break and the source node may have to do the route discovery process again. This make the % is not as much as expected. In the mobility case, the average percent is in the range about 25 to 30.

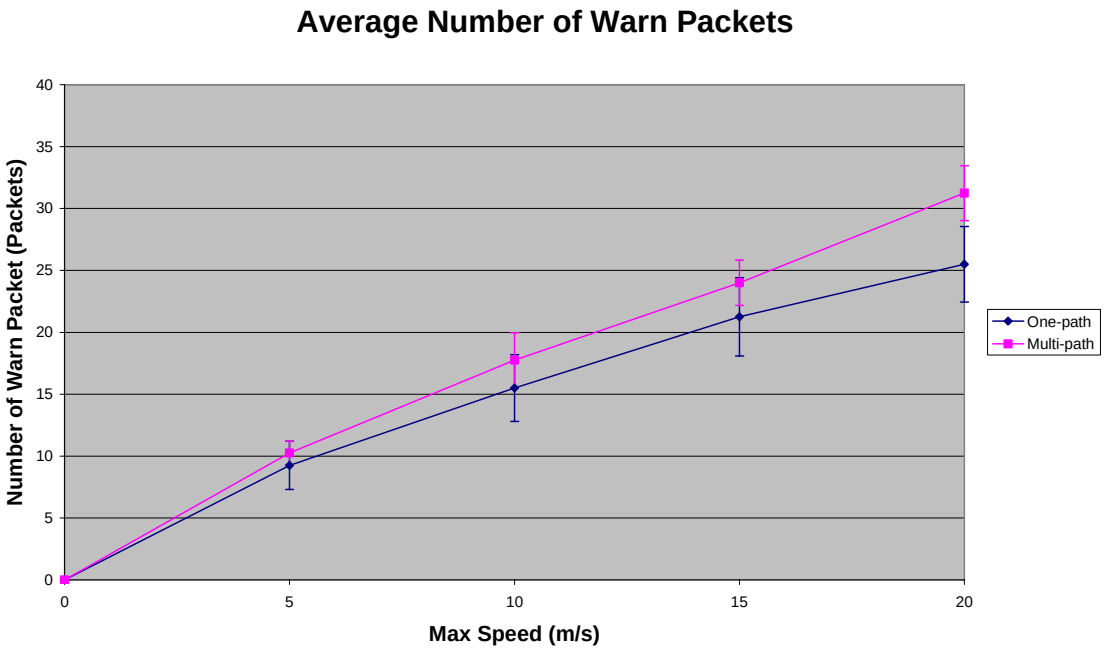


Figure 5.8: Average Number of Warn Packets.

Figure 5.8 shows the average number of warn packet sent by the nodes. The value in the figure has shown that when the node max speed is increasing, the warning packet is sent more often. This can imply that the preemptive technique is more efficient in the high mobility.

5.4.2 Discussion

As seen from the above figures, the performance of the proposed multi-path approach is revealed as follow.

- The proposed multi-path approach achieved better average throughput than the one-path approach. In every node mobility simulations, the average throughputs of the proposed approach are better than the regular AODV routing. The average throughputs of the proposed approach in high node mobility are significantly better than the one-path's average throughput.
- Compare the average throughput between the preemptive and non-preemptive simulation, the proposed multi-path approach with

preemptive average throughputs are slightly better than the non-preemptive multi-path approach. Even though the preemptive approach seems to affect not much for the routing, but in the high node mobility, the preemptive technique helps preventing the path break more than in the low mobility.

- The proposed multi-path approach yields the less average delay in every node max speed. The delay is reduced more in the high node mobility in both non-preemptive and preemptive cases. This means that even though the proposed multi-path approach has more AODV packet transmissions because of setting multiple reverse routes and multiple forward routes. The routes selected by the proposed multi-path approach are better than the routes selected by one-path approach.
- The standard deviation of the average delay in both one-path and proposed multi-path is almost the same. That means the average delay of the multi-path approach has almost the same range with the one-path approach.
- The example delay distribution of the multi-path approach is smoother than the example delay distribution of the one-path approach. This may imply that the multi-path approach is better than one-path approach in video streaming.
- The Route discovery frequency of the proposed approach is smaller than the one-path approach. This can imply that the multi-path can provide the path to the destination better than the one-path. The proposed approach has storages routes to change before the path break occurs.
- The average hop count of the proposed approach is bigger than the one-path approach in some cases. The proposed approach may use the longer path because the proposed approach will select the path which has smaller jointcount value. The hop count value is second condition. However, these longer paths still provide shorter average delay than the one-path approach.

- The proposed multi-path approach takes a longer time interval than the one-path approach from the route discovery to sending the data packet. Because of several RREQ and RREP from the multiple reverse routes and multiple forward routes. The source node may have to wait longer time before receiving the first RREP.
- The author calculates the percent of using multiple paths in data transmission. The results show that the multi-path can be used about 25 to 30% in mobility case. The source node may not find the multiple paths because of several reasons, the RREP from another node may come late, or even there is no multiple paths only one path is possible. However, the average throughputs are quite acceptable because even though the source node does not hold the multi-path, the intermediate nodes may hold the multi-path and can switch the path before the path break occur. In no mobility case, there are some cases that if the node is too close together, that intermediate node is no need. So the percent of multi-path will dramatically drops because there is no movement, the paths cannot be changed until the other transmission disturbs the transmission.
- The number of warn packets is increased when the node mobility is high. This may conclude that, the preemptive can do the warning effectively because they can warn better in the high node mobility which the path tends to break easier.

Chapter 6
Conclusion

6.1 Conclusion

The video streaming over mobile ad hoc networks have some problems in bandwidth, delay and packet loss because of the mobility of the node in the network. In order to solve the problems, the multi-path transmission is recommended. The sender will generate multiple compressed video flows. Then the flows are partitioned and assigned to the multiple paths. In order to use the multi-path transport, the underlying routing protocol has to provide and update the multiple paths between the source and the destination node. The author introduced the multi-path routing protocol for video streaming over ad hoc network. The approach includes the jointcount-based multi-path routing and the preemptive technique.

The author used Ad hoc On-Demand Distance Vector Routing (AODV) for routing protocol. The selections of multiple routes are based on approach to find a pair of link-disjoint paths by selecting a route having a smaller number of common intermediate nodes on its path. The jointcount is introduced to use to keep the number of common nodes along the path. The source node can select the link-disjoint paths by using the jointcount value.

The author also added the preemptive technique to improve the quality of the routing. If the current path is going to break, the preemptive technique will seamlessly switch from the current path to an alternative good path before a break.

The author has implemented four types of the simulations which are one-path with preemptive, one-path without preemptive, multi-path with preemptive and multi-path without preemptive. Each of the simulation is done in the various maximum speeds

of the nodes. The simulation result has shown that the multi-path approach overcome the one-path approach. The preemptive technique yields slightly better than non-preemptive technique. The results for a variety of mobility show that the proposed protocol achieves better performance in terms of average throughput and average delay. This indicates that the proposed protocol can be applied to use for video streaming over mobile ad hoc networks.

List of Publications

---

[1] **P. Prapatsaranon**, K. Rojviboonchai and H. Aida, “Multi-path Routing Protocol for Video Streaming over Ad Hoc Networks”, Proc. of IEICE Society Conference, Session B-15, No.9, September 2005.

[2] **P. Prapatsaranon**, K. Rojviboonchai and H. Aida, “Multi-path Routing Protocol with Preemptive Technique for Video Streaming over Ad Hoc Networks”, Proc. of IEICE Society for the Study of Information Network (IN), March 2006. (To be presented)

[3] Minh T. Nguyen, H. Aida and **P. Prapatsaranon**, “New Multipath Routing Protocol based on AODV for Mobile Ad Hoc Network”, Proc. of IEICE Society for the Study of Information Network (IN), March 2006.

References

[1] S. Corson, J. Macker, "Mobile Ad Hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations", RFC 2501.

[2] Tom Goff, Nael B. Abu-Ghazaleh, Dhananjay S. Phatak and Ridvan Kahvecioglu, "Preemptive Routing in Ad Hoc Networks", Proceedings of ACM Mobicom 2001, July 2001, Rome, pp 43-52

[3] A.S. Tanenbaum, "Computer Network," Fourth edition, 2003.

[4] S.S. Rappaport. "Wireless Communication Systems" Prentice Hall, 1996.

[5] Yihan Li, Shiwen Mao, Shivendra S. Panwar, "The Case for Multipath Multimedia Transport over Wireless Ad Hoc Networks", First International Conference on Broadband Networks (BROADNETS'04) October 2004.

[6] Sung-Ju Lee, Mario Gerla, "Split Multipath Routing with Maximally Disjoint Paths in Ad hoc Networks", Proceedings of ICC 2001, Helsinki, Finland, June 2001.

[7] K. Rojviboonchai, F. Yang, Q.Zhang, H. Aida and W. Zhu, "AMTP: A Multipath Multimedia Streaming Protocol for Mobile Ad Hoc Networks," Proc. of IEEE International Conference on Communications (IEEE ICC 2005), Seoul, Korea, May 2005.

[8] Charles E. Perkins, Elizabeth M. Royer, "Ad-hoc On-Demand Distance Vector Routing", Proc. 2nd IEEE Workshop on Mobile Computing Systems and Applications, pp.90-100, February 1999.

[9] C. Perkins, E. Belding-Royer, S. Das, "Ad hoc On-Demand Distance Vector (AODV) Routing", RFC 3561.

[10] M.S. Corson and A. Ephremides "A Distributed Routing Algorithm for Mobile Wireless Networks", ACM J. Wireless Networks, 1(1), January 1995.

[11] D. Johnson and D. Maltz "Dynamic source routing in ad-hoc wireless networks" In Computer Communications Review – Proceedings of SIFCOMM '96, August 1996.

[12] Shinji MOTEGI, Hiroki HORIUCHI and Members, "AODV-Based Multipath Routing Protocol for Mobile Ad Hoc Networks", IEEE Trans. Commun., Vol.E87-B, No.9 September 2004

[13] A Collaboration between researchers at SAMAN, CONSER and ACIRI, "The Network Simulator – ns-2," http://www.isi.edu/nsnam/ns/index.html

[14] WaveLAN/PCMCIA Card User's Guide – Lucent Technologies.

[15] A Collaboration between researchers at UC Berkeley, LBL, USC/ISI, and Xerox PARC, “The ns Manual”- The VINT Project.