國立成功大學

資訊工程學系

碩士論文

在無線行動式隨意網路上對多媒體通訊做保證服務品質之隨選混合多重路徑繞路協定

An On-demand Hybrid Multipath Routing Protocol for Multimedia Communication with QoS Guarantees in Mobile Ad Hoc Networks

研究生：邱仁傑

指導教授：蘇銓清博士

中華民國九十五年七月

國立成功大學

資訊工程研究所

碩士論文

在無線行動式隨意網路上對多媒體通訊做保證服務品質之

隨選混合多重路徑繞路協定

An On-demand Hybrid Multipath Routing Protocol for Multimedia Communication with QoS Guarantees in Mobile Ad Hoc Networks

研究生：邱 仁 傑

此論文經審查及口試合格特此證明

論文口試委員

______ ______

______ ______

______

指導教授：______博士

系(所)主管：______博士

中華民國 九十五年 七月

An On-demand Hybrid Multipath Routing Protocol for Multimedia Communication with QoS Guarantees in Mobile Ad Hoc Networks

By

Ren-Jie Chiou

A thesis submitted to Graduate Division in partial fulfillment of the requirements for the degree of

Master of Department of Computer Science and Information Engineering,

National Cheng Kung University,

Tainan, Taiwan, R.O.C.

July 2006

Approved by Chuan-Ching Sue

Jichiang Tsai

Sun-Yuan Hsieh

Advisor : Chuan-Ching Sue

Chairman : Chiang

在無線行動式隨意網路上對多媒體通訊做保證服務品質之隨選混合多重路徑繞路協定

研究生: 邱仁傑　　　　　　　　指導教授: 蘇銓清 博士

國立成功大學資訊工程研究所

<u>中文摘要</u>

行動式隨意網路為沒有固定的基礎架構的無線網路，每一個無線節點可透過直接無線鏈結的方式通訊，或是利用中間節點以連續多點跳躍的無線鏈結方式來通訊。由於節點的任意地移動，使得網路的拓撲變動頻繁，因應這種網路特性需要設計一個有效率的繞路協定。在行動式隨意網路上 AODV 和 DSR 是兩個被廣為應用的隨選繞路協定，然而，之前的許多研究發現了兩者的限制。例如，當在傳輸封包的節點移動，節點之間因為距離過遠無法通訊而造成傳輸中斷，必須重新找尋路徑。一般來說，隨選繞路協定皆使用氾濫法來找尋路徑，而氾濫法會大量地浪費網路的頻寬。為了防止這樣的傳輸錯誤和減少氾濫法的使用，這篇論文提出一個隨選混合多重路徑繞路協定，它最大的特色是結合了節點不相交多重路徑和交錯多重路徑，透過理論分析和模擬結果，隨選混合多重路徑繞路協定可以減少氾濫法的使用頻率、提高封包傳輸率和縮短平均的點對點傳輸延遲。這篇論文還延伸隨選混合多路徑繞路協定對多媒體通訊做保證服務品質，我們提出一個多媒體串流分配方法，它依據多媒體串流的優先權，分配多媒體串流於不同的多路徑上，透過模擬結果顯示，隨選混合多路徑繞路協定利用多媒體串流分配方法對於多媒體通訊可達到較好的影像傳輸品質。

關鍵詞：行動式隨意網路，隨選繞路協定，氾濫法，多重路徑繞路協定，節點不相交多重路徑，交錯多重路徑，多媒體串流

An On-demand Hybrid Multipath Routing Protocol for Multimedia Communication with QoS Guarantees in Mobile Ad Hoc Networks

Ren-Jie Chiou*   Chuan-Ching Sue**

Department of Computer Science and Information Engineering,
National Cheng Kung University

Abstract

A Mobile ad hoc network (MANET) is a collection of wireless mobile computers forming a temporary network without existing wire line infrastructures. Due to the dynamic nature of the network topology and resource constraints, designing an efficient routing in MANETs is challenging. AODV and DSR are two most widely studied on-demand ad hoc routing protocols with low routing overheads. However, previous studies have identified various limitations of these protocols. For example, whenever a node moves and its link breaks on the active route, it can cause a communication fault and then invokes a route discovery process. In general, on-demand protocols use query flooding to discover routes. Such flooding consumes a substantial portion of the network bandwidth. To tolerate communication faults and decrease flooding, this study explores the network redundancy through multipath routing. The designated on-demand hybrid multipath routing (OHMR) features a novel characteristics; it establishes the node-disjoint multipath and the braided multipath between a source-destination pair. Through theoretical analysis and simulation results, we show OHMR can reduce the frequency of route discoveries and achieve a higher packet delivery ratio. Furthermore, the average end-to-end delay for OHMR is shorter than single path, braided multipath and node-disjoint multipath routing schemes. We also extend OHMR with a multimedia traffic allocation strategy to classify multimedia sub-streams among multiple paths according to different priority levels. Our experiments show that the proposed protocol for multimedia communication can improve the performance of the fraction of decodable frames and achieve better performance in terms of video quality over the node-disjoint multipath.

Keyword: MANETs, Multipath, AODV, DSR, Node-disjoint multipath, Braided multipath, multimedia communication

* Author
** Advisor

誌 謝

時光飛逝，在成大的求學生涯竟然足足有六個年頭了，回憶起在成大資訊系直到資訊所的點點滴滴，有吃喝玩樂的歡笑、埋頭苦讀的衝刺、撰寫論文的辛酸，心中盡是無限的感激與不捨，而隨著論文的完成，緊接著是另一段新的生活、新的挑戰。

碩士班兩年的修習期間，首應感謝指導教授蘇銓清博士的悉心指導及鼓勵，使本篇論文得以順利付梓。

在實驗室的研究生涯，因為許多同學的陪伴而豐富更加多采多姿，感謝陪我走過這兩年的研究所同門勇慶、韋仁和世璋，以及研究室室友嘉農、俊永、佩娟、朝陽、思翰和韋志，在學業上的砥礪，共同經歷求學的苦與樂，你們讓我的生活帶來許多歡笑、學習與成長，這段時光我終身難忘。在此，也感謝所有曾經陪我走過這段旅程的朋友們。

最後，僅以本文獻給長年辛勤栽培撫育我的父母親與親愛的弟弟，感謝他們多年來對我無怨無悔的付出與關懷，在物質上及精神上，有形和無形的幫助和支持，使我後顧之憂，全心全意投入學業，並時時鼓勵我，讓我勇於面對困難，才能順利完成研究所的學業，感激之心，溢於言表。也感謝所有關心我的人，謝謝你們！

仁傑 2006 年 7 月于成大

Contents

List of Tables

List of Figures

Chapter 1

Introduction

A Mobile Ad hoc NETwork (MANET) is a collection of wireless mobile computers forming a temporary network, with neither a fixed base station infrastructure nor centralized management function. Each node in MANETs acts both as a host and a router. If two nodes are out of radio range, all message communications between them must pass through one or more intermediate nodes.

The routing protocols for MANET must be adaptive and capable of maintaining routes as the network topology changes and avoid network congestion. Of the various routing protocols, on-demand routing protocols are particularly widely developed because they are efficient in reducing routing overhead. In order to reduce routing overheads, on-demand routing protocols build and maintain routes only when they require routes. Ad hoc On-Demand Distance Vector (AODV) [1] and Dynamic Source Routing (DSR) [2] are two of the most widely studied on-demand ad hoc routing protocols. However, these protocols also have certain performance limitations. For on-demand protocols, whenever a route is required, the route discovery process triggers a flooding process where the source node floods the entire network with query packets to search for a route to the destination. This flooding operation consumes a substantial amount of the available network bandwidth, which clearly is the most important resource in wireless networks. Whenever a node moves and its links break on the active route, it can cause a communication fault and then invoke a route discovery process and result in a loss of a large number of packets and latency. Communication faults can thus significantly affect the performance of routing in MANET.

Multipath routings which can tolerate communication faults establish multiple paths between a source and a destination in a single route discovery attempt. Multipath routing protocols in MANETs can also provide load balancing and higher aggregate bandwidth by making use of the availability of multiple route paths. In these protocols, a new route discovery operation is invoked only when all of the routing paths in the network fail. It has been shown that multipath routing yields significant benefits. Multipath routing has been proposed for ad hoc networks such as MANETs [4, 5, 6, 7, 15, 16, 17]. Although multipath protocols build multiple routes on demand, the majority of them establish only node-disjoint multipath. These node-disjoint alternate paths could prevent communication faults. Because the low relationship between the primary path and node-disjoint alternate paths, link faults on them do not affect each other. A multipath routing technique using braided multipath has been proposed [8]. The multiple paths in a braided multipath are only partially node-disjoint from each other, as opposed to completely node-disjoint. These braided alternate paths which are not node-disjoint with the primary path are used to prevent link faults and would expend energy comparable to the primary path. The set of paths comprising multiple node-disjoint and braided routing paths (including the primary path) is designated as the hybrid multipath. We propose a novel and practical routing protocol, which combines the designated On-demand Hybrid Multipath Routing (OHMR) to identify the hybrid multipath extension to Ad hoc On-Demand Vector (AODV).

Multimedia communication is expected to become popular in MANETs in the future. But multimedia communication has strict delay and loss requirements. Multimedia communication in MANETs faces a number of technical challenges due to the nodes are free to move around randomly and bandwidth constraints caused by the shared medium, and hence the established connection route between a source-destination pair could be broken or congestion during the transmission, which may cause communication faults in the received multimedia data. Several researchers have proposed to use multipath routing for multimedia transport [14, 18, 19, 20, 21]. Multipath routing techniques in these studies are useful for finding multiple node-disjoint paths between a source and a destination. These node-disjoint multipaths are used to provide load balancing. There are node-disjoint paths in the hybrid multipath. OHMR not only achieves load balancing but also has higher reliability than the node-disjoint multipath. We extend OHMR with a multimedia traffic allocation strategy to support multipath multimedia communications in MANETs. Through theoretical analysis and simulation results, we show OHMR can reduce the frequency of route discoveries and achieve a higher packet delivery ratio. Furthermore, the average end-to-end delay for OHMR is shorter than single path, braided multipath and node-disjoint multipath routing schemes. Our protocol uses a multimedia traffic allocation strategy to classify multimedia sub-streams among multiple paths according to different priority levels. The compressed multimedia stream can be segmented into several sub-streams. Each of these sub-streams takes a particular service class. The strategy is to allow braided paths to protect more important sub-streams, and node-disjoint paths to provide load balancing. Our experiments show that the proposed protocol for multimedia communication can improve the performance of the fraction of decodable frames and achieve better performance in terms of video quality over the node-disjoint multipath.

The remainder of this paper is organized as follows. In Chapter 2, we present the background and preliminaries. Chapter 3 introduces the proposed OHMR protocol. Chapter 4 develops OHMR protocol with a multimedia traffic allocation strategy. In Chapter 5, performance evaluation by Theoretical analysis and simulations are presented. Finally, Chapter 6 concludes the paper.

Chapter 2

Background and Preliminaries

2.1 Multipath Routing

Multipath routings which can tolerate communication faults establish multiple paths between a source and a destination in a single route discovery attempt. In these protocols, a new route discovery operation is invoked only when all of the routing paths in the network fail. A wireless topology is shown in Figure 2.1. An example of a node-disjoint multipath is shown in Figure 2.2. In this multipath, the source, S, sends data to the destination node, D, using one primary path, i.e. S->A->B->D, and a alternate path, i.e. S->F->G->H->D. The alternate path is node-disjoint with the primary path. The multiple paths in a braided multipath are only partially node-disjoint from each other, as opposed to completely node-disjoint. Figure 2.3 presents an example of a braided multipath, in which the node, S, sends data to the destination, D, using one primary path, i.e. S->A->B->D, and two alternate paths, i.e. S->C->B->D and S->A->E->D.

The Temporally Ordered Routing Algorithm (TORA) presented in [9] established multiple alternate paths by maintaining a “destination-oriented” directed acyclic graph (DAG) from the source. However, simulation studies [10] have shown that TORA compares unfavorably with DSR and AODV in that the packet delivery ratio is lowest and the average end-to-end delay performance is poor. The performances of AODV are greater than these of TORA. If AODV performs worst than OHMR, then OHMR has much better performances than TORA. In this paper, we did not consider TORA in the analytical and simulation results even though it can maintain multiple redundant paths.

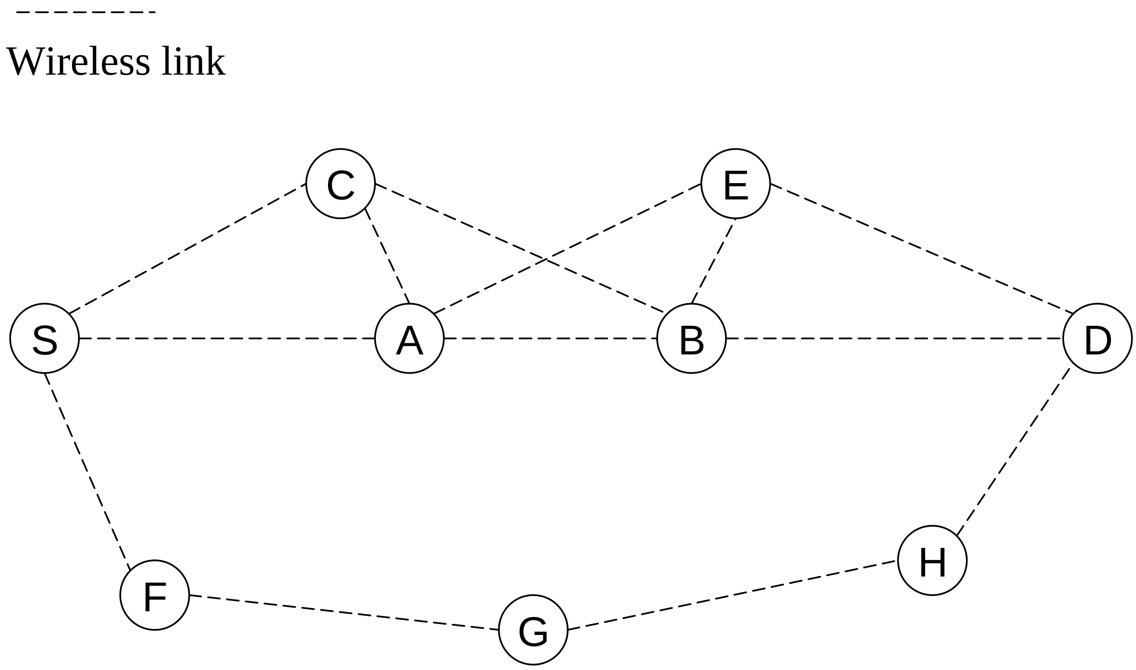


Figure 2.1 Wireless topology

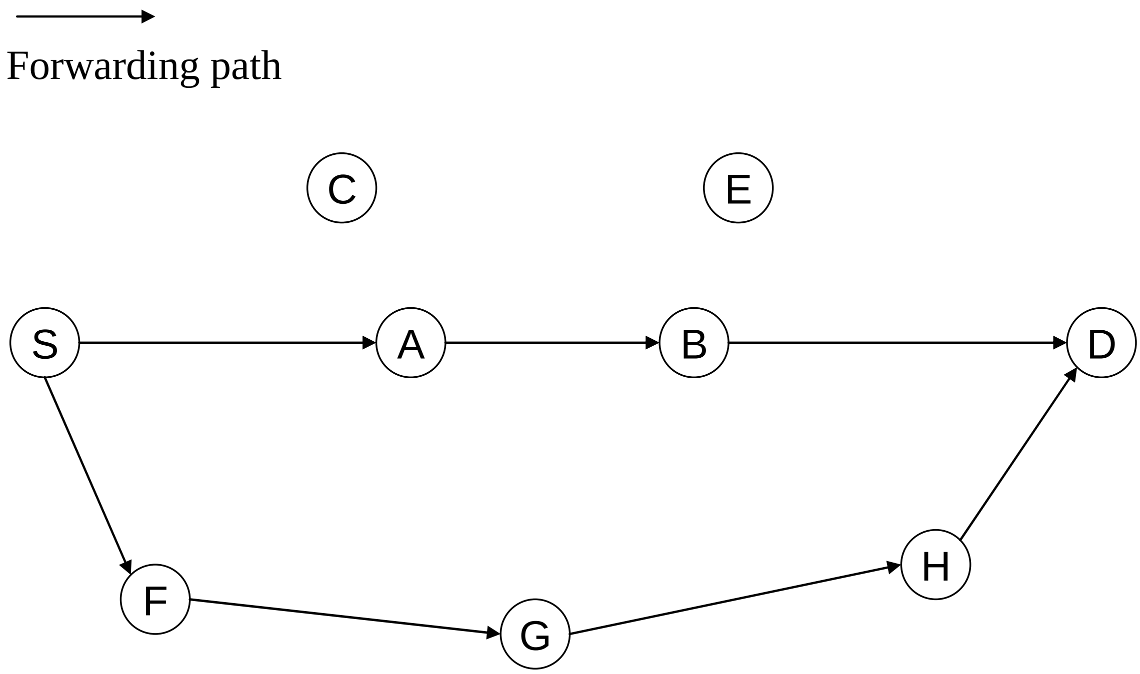


Figure 2.2 Node-disjoint multipath

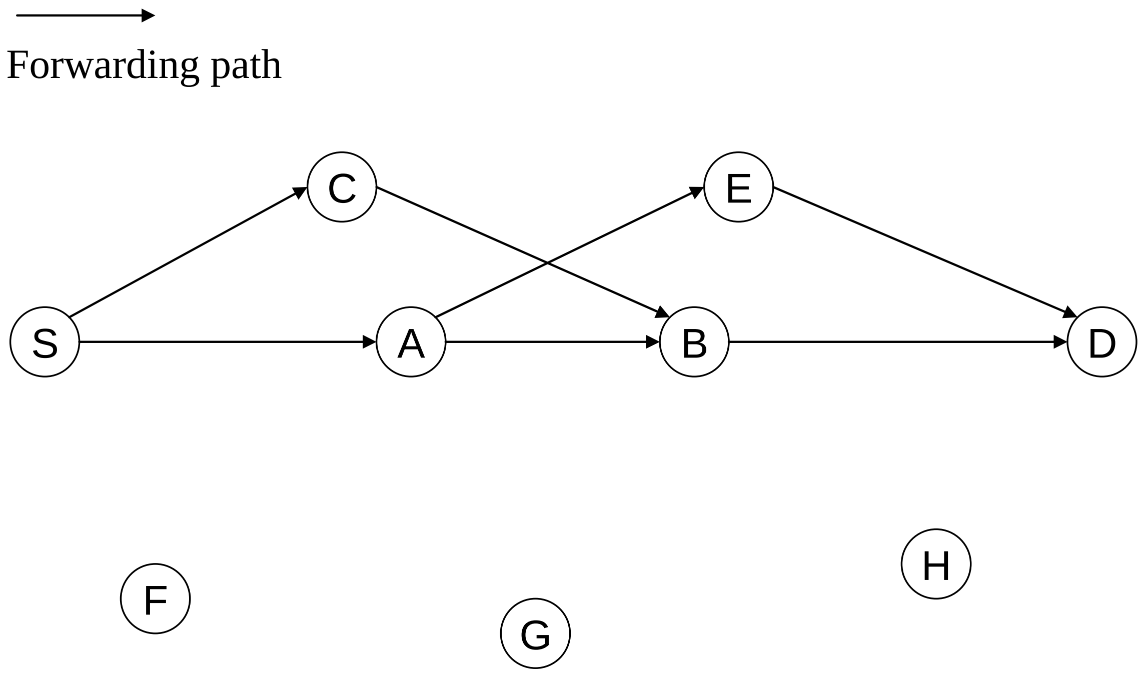


Figure 2.3 Braided multipath

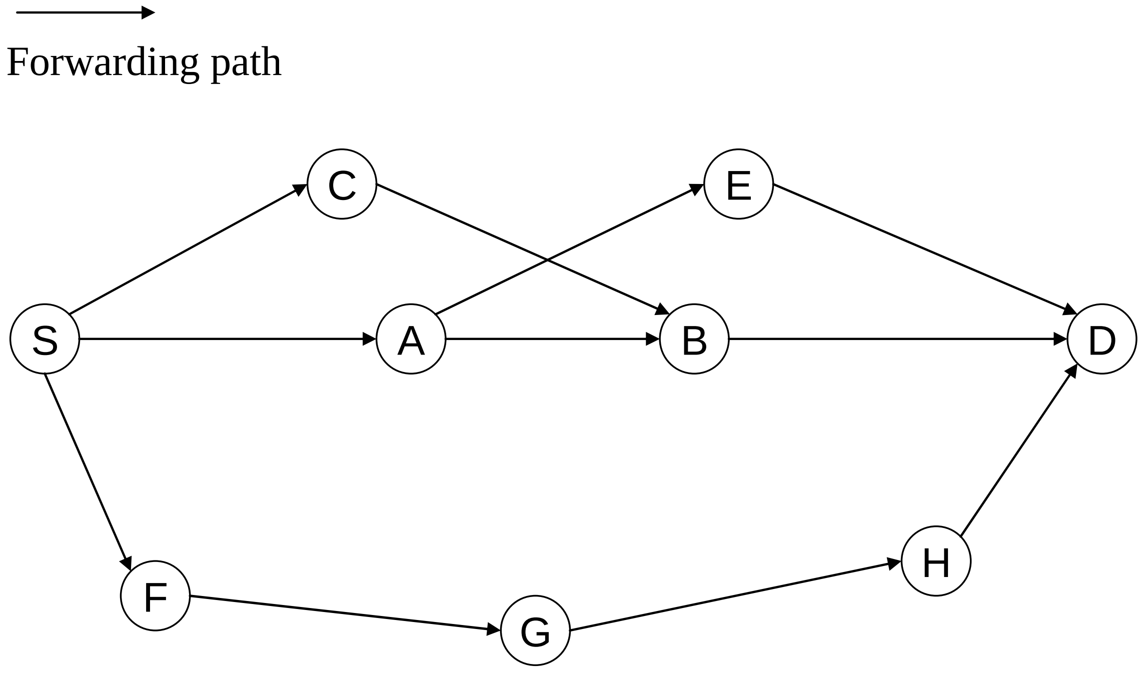


Figure 2.4 Hybrid multipath

This paper proposes a novel and practical routing protocol designated On-demand Hybrid Multipath Routing (OHMR). This protocol modifies and extends AODV to identify hybrid multipath comprising multiple node-disjoint and braided routing paths. An example of a hybrid multipath is shown in Figure 2.4. In this multipath, the source node, S, sends data to the destination node, D, using one primary path, i.e. S->A->B->D, and three alternate paths, i.e. S->C->B->D, S->A->E->D and S->F->G->H->D. One of the alternate paths, i.e. S->F->G->H->D, is disjointed with the primary path, while the other two are non-disjointed.

2.2 Multipath Transport

Several researchers have proposed to use multipath routing for multimedia transport [14, 18, 19, 20, 21]. According to the availability of multiple paths, the compressed video stream can be segmented into several sub-streams. Each of these sub-streams takes a particular service class. The strategy is to allow more important sub-streams to travel over higher QoS path, and less important sub-streams to travel over lower QoS path. The higher QoS path is usually the primary path and the lower QoS path is usually the alternate node-disjoint path. All of them use the node-disjoint multipath routing. With the node-disjoint multipath transport, link/node failure events on different paths are not entirely correlated. Alternate paths in the node-disjoint multipath are unaffected by route failures along the primary path. Even though disjointed paths have attractive resilience properties, they can be energy inefficient in their transmission of data packets. Alternate node-disjoint paths tend to be longer, and therefore consume significantly more energy than the primary path. Since this energy inefficiency can adversely impact the lifetime of MANETs and the transmission of higher priority sub-stream. This study constructs both node-disjoint multipath and braided multipath simultaneously. OHMR finds an alternate braided path for each node on the primary path. The braided alternate paths can avoid link/node failures on the primary path breaks. The alternate paths in a braided multipath are only partially node-disjoint from the primary path, as opposed to completely node-disjoint. Hence, the alternate paths of the braided multipath consume a comparable energy to that of the primary path. And then we let less important sub-streams to travel over the alternate node-disjoint path to provide load balancing. The primary path, the braided alternate paths, and the node-disjoint alternate path can protect each other to achieve better performance in terms of video quality.

In this paper, we present a practical On-demand Hybrid Multipath Routing protocol for multimedia communication with QoS guarantees in MANETs, which combines the node-disjoint multipath and the braided multipath with multimedia traffic allocation strategy for QoS guarantees in MANETs.

Chapter 3

On-demand Hybrid Multipath Routing

This paper proposes a novel and practical routing protocol designated On-demand Hybrid Multipath Routing (OHMR). This protocol modifies and extends AODV to identify hybrid multipath comprising multiple node-disjoint and braided routing paths. OHMR which can tolerate communication faults establish multiple paths between a source and a destination in a single route discovery attempt. A new route discovery operation is invoked only when all of the routing paths in the network.

The basic principle of OHMR is to identify multiple paths during route discovery. OHMR is designed primarily for highly dynamic ad hoc networks in which communication faults occur on a frequent basis. When a single path on-demand routing protocol such as AODV is used in such networks, a new route discovery must be launched each time a communication fault occurs. Each route discovery induces high overheads and latency. This inefficiency can be avoided by maintaining multiple redundant paths such that a new route discovery process is initiated only when all of the paths to the destination are broken. OHMR searches for hybrid multipath comprising both node-disjoint and braided multipaths. The proposed hybrid multipath can be summarized as follows: For each node on the primary path, find the alternate path from the source to the destination that does not contain that node. So we extend AODV which is a distance vector protocol, nodes can use the braided alternate paths to stride unreachable neighbors for tolerating link faults immediately. Comparing source routing, OHMR is such a multipath distance vector protocol that can decrease the end-to-end delay. These braided alternate paths which are not node-disjoint with the primary path are used to prevent these link faults and would expend energy comparable to the primary path. Furthermore, OHMR also discover multiple node-disjoint paths. These node-disjoint alternate paths could prevent communication faults when all of braided alternate paths are not available. Because the low relationship between the primary path and node-disjoint alternate paths, node faults on them do not affect each other. The resulting set of paths (including the primary path) is designated as the hybrid multipath.

3.1 Routing Table

OHMR is a routing protocol, and it deals with route table management. OHMR shows some fields with each route table entry:

Table 3.1 Some fields with each route table entry and meaning of fields

Field	Meaning
rt_dst	Destination IP Address
rt_seqno	Destination Sequence Number
rt_hops	Hop Count (number of hops needed to reach destination)
rt_nexthop	Next Hop
rt_flags	The routing table entry state (e.g., valid, invalid, repair)
rt_mpath	Path type
rt_full	The number of intermediate nodes that are not protected in the route record
rt_braid	Record intermediate nodes in the route record which are protected
rt_passroute	Route record
rt_nop	The number of neighbors which sent the RREQ to the current node

Managing the sequence number is crucial to avoiding routing loops [1], even when links break and a node is no longer reachable to supply its own information about its sequence number. A destination becomes unreachable when a link breaks or is deactivated. When these conditions occur, the route is invalidated by operations involving the sequence number and marking the routing table entry state as invalid.

3.2 OHMR Terminology

This section defines terminology used with OHMR.

Active route:

If the rt_flags field of the entry in the routing table is marked as valid, routes with the entry are called active routes. Only active routes can be used to forward data packets.

Broadcast:

Broadcasting means transmitting to the IP Limited Broadcast address, 255.255.255.255. A broadcast packet may not be blindly forwarded, but broadcasting is useful to enable dissemination of control packets throughout the ad hoc network.

Destination:

Destination is an IP address to which data packets are to be transmitted. A node knows it is the destination node for a typical data packet when its address appears in the appropriate field of the IP header. Routes for destination nodes are supplied by action of the OHMR protocol, which carries the IP address of the desired destination node in route discovery messages.

Forward route:

Routes which are used to send data packets now are called forward routes.

Invalid route:

A route has expired, denoted by a state of invalid in the routing table entry. An invalid route is used to store previously valid route information for an extended period of time. An invalid route cannot be used to forward data packets, but it can provide information useful for future RREQ messages.

Originating node:

A node initiates a RREQ message. For instance, the node which initiates a route discovery process and broadcasts the RREQ message is called the originating node of the RREQ message.

Reverse route:

A route set up to forward a reply (RREP) packet back to the originator from the destination or from an intermediate node having a route to the destination.

Sequence number:

A monotonically increasing number is maintained by each originating node. In routing protocol control packets, it is used by other nodes to determine the freshness of the information contained from the originating node.

Valid route:

See active route

Source:

See originating node.

3.3 Control Packet Structures

3.3.1Route Request (RREQ) Packet Structure

Table 3.2 Some fields of RREQ and meaning of fields

Field	Meaning
rq_dst	Destination IP Address; The IP address of the destination for which a route is desired.
rq_dst_seqno	Destination Sequence Number; The latest sequence number the originator recorded the route to the destination. If the originator never gets the route to the destination, rq_dst_seqno is set to zero.
rq_src	Originator IP Address; The IP address of the node which originated the Route Request.
rq_src_seqno	Originator Sequence Number; The current sequence number to be used in the route entry pointing towards the originator of the route request.
rq_bcast_id	RREQ ID; rq_bcast_id is a sequence number. rq_bcast_id and the originating node's IP address can identify the RREQ.
rq_hop_count	Hop Count; The number of hops from the Originator IP Address to the current node handling the request.
rq_passroute[]	Route record; Each node generating or receiving RREQ appends its IP address to the route record of the route request.

3.3.2Route Reply (RREP) Packet Structure

Table 3.3 Some fields of RREP and meaning of fields

Field	Meaning
rp_dst	Destination IP Address; The IP address of the destination for which a route is supplied.
rp_dst_seqno	Destination Sequence Number; The destination sequence number associated to the route.
rp_src	Originator IP Address; The IP address of the node which originated the RREQ for which the route is supplied.
rp_hop_count	Hop Count; The number of hops from the Originator IP Address to the Destination IP Address.
rp_passroute[]	Route record; It is copied by destination from the route record of RREQ
rp_mpath	Path type; It is set by the destination.

3.3.3Route Error (RERR) Packet Structure

Table 3.4 Some fields of RERR and meaning of fields

Field	Meaning
DestCount	The number of unreachable destinations included in the message.
unreachable_dst[]	Unreachable Destination IP Address; The IP address of the destination that has become unreachable due to a link break.
unreachable_dst_seqno[]	Unreachable Destination Sequence Number; The sequence

	number in the route table entry for the destination listed in the previous Unreachable Destination IP Address field.

The RERR message is sent whenever a link break causes one or more destinations to become unreachable from some of the node's neighbors.

3.4 Route Discovery

3.4.1 Generating Route Requests

A node disseminates a RREQ when it determines that it needs a route to a destination and does not have one available. This can happen if the destination is previously unknown to the node, or if a previously valid route to the destination is marked as invalid. The Destination Sequence Number field in the RREQ message is the last known destination sequence number for this destination and is copied from the Destination Sequence Number field in the routing table. The Originator Sequence Number in the RREQ message is the node's own sequence number, which is even and is incremented by two prior to insertion in a RREQ. The RREQ ID is incremented by one when the current node initiates a RREQ. Each node maintains only one RREQ ID. The Hop Count field is set to one. OHMR extends AODV and adds a Route Record field to record the path in a RREQ. The originating node adds its own address to the Route Record field in the RREQ message. Further, when a RREQ is generated or forwarded by a node in the network, each node appends its address to the Route Record field in the RREQ message.

We give below the pseudo code for packet reception routine function “recv(Packet *p)” and other utility functions, respectively. Function “recvAODV(Packet *p)” is called when packet reception routine function receives a packet with type of AODV. Function “rt_resolve(Packet *p)” is a route handling function. Functions “rt_lookup(nsaddr_t id)”, “rt_lookupmpath(nsaddr_t id, int mpath)” and “rt_lookupnp(nsaddr_t id, nsaddr_t nexthop)” are some utility functions that look up the entry in the routing table where incoming parameters match the fields of the entry. Note that the pseudo code for Function “recvMPEGpkt(Packet *p)” called in the Function “rt_resolve(Packet *p)” is presented in section 4.2.

```
State:
Index           // IP Address of the current node
Sender          // Neighbor which sent the RREP to this node
Source          // Originating node; source address in IP header of the packet
Destination     // Destination address in IP header of the packet
INFINITY        // Maximum expected network diameter
Bid             // RREQ ID of the current node

Function recv(Packet *p)
Begin
    If the type of p is AODV
        recvAODV(p)
        Return
    End If
    If p originated with the current node
        Add ttl(time to live) into IP header of p
    Else If p has been sent by the current node
        Drop p
        Return
    End If
```

```
    If p needs to be broadcasted
        Broadcast p
    Else
        rt_resolve(p)
End

Function recvAODV(Packet *p)
Begin
    Switch the type of p
        case AODVTYPE_RREQ:
            recvRREQ(p)
            Break;
        case AODVTYPE_RREP:
            recvRREP(p)
            Break;
        case AODVTYPE_RERR:
            recvRERR(p)
            Break;
        default:
            The type of p is invalid AODV type
            Exit(1)
    End Switch
End

Function rt_resolve(Packet *p)
Begin
    Set rt is an entry in the routing table
    If the type of p is MPEG
```

```
            rt = recvMPEGpkt(p)
        Else
            rt = rt_lookup(destination)
        If rt == NULL
            Add an entry in the routing table where rt_dst field is filled with destination
        End If
        If rt->rt_flags == RTF_UP && rt->rt_hops != INFINITY
            Forward p using the entry rt in the routing table
        Else If index == source
            sendRREQ(destination)
        Else
            Set rerr is a RERR packet
            Add destination into unreachable_dst field of rerr
            sendRERR(rerr)
End

Function rt_lookup(nsaddr_t id)
Begin
    For each entry in the routing table
        If id == rt_dst field of the entry
            Return the entry
        End If
    End For
    Return NULL
End

Function rt_lookupmpath(nsaddr_t id, int mpath)
Begin
```

```
    For each entry in the routing table
        If id == rt_dst field of the entry && mpath == rt_mpath field of the entry
            Return the entry
        End If
    End For
    Return NULL
End
```

```
Function rt_lookupnh(nsaddr_t id, nsaddr_t nexthop)
Begin
    For each entry in the routing table
        If id == rt_dst field of the entry && nexthop == rt_nexthop field of the entry
            Return the entry
        End If
    End For
    Return NULL
End
```

Figure 3.1, Figure 3.2 and Figure 3.3 display the flowcharts of Function “recv(Packet *p)”, “rt_resolve(Packet *p)” and “recvAODV(Packet *p)”.

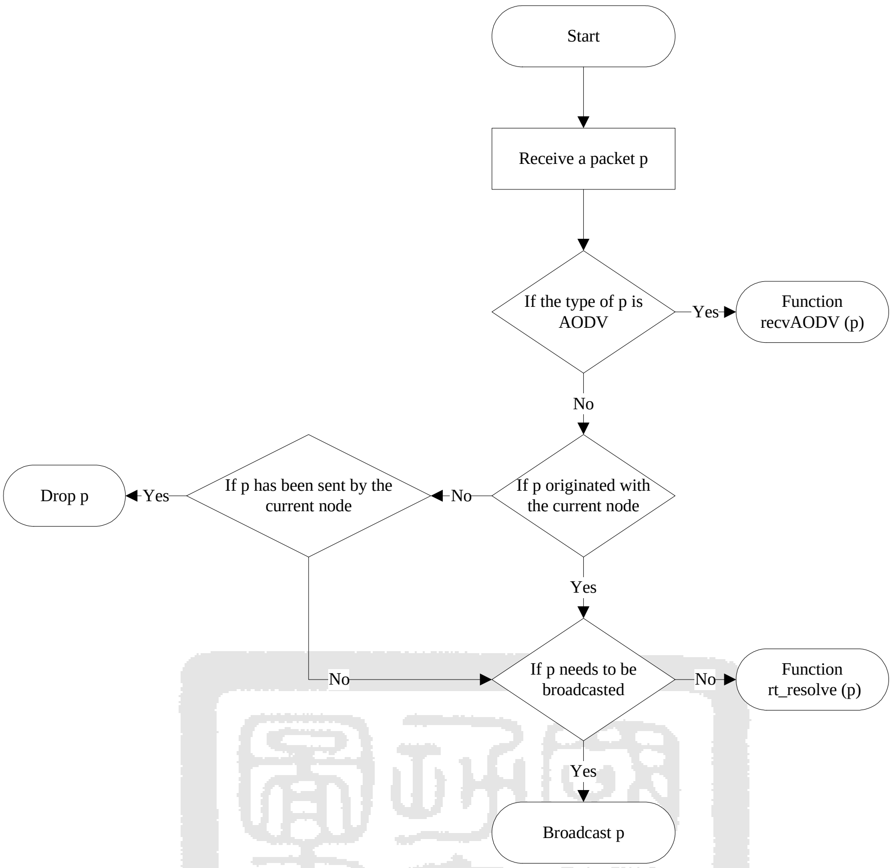


Figure 3.1 Flowchart of Function recv(Packet *p)

Function “recv(Packet *p)” is a packet reception routine. There are three important function blocks in the flowchart of function “recv(Packet *p)” . We describe them below.

1. If the type of packet p is AODV, function “recvAODV(Packet *p)” is called. Function “recvAODV(Packet *p)” handles the packet which type is AODV.

2. If packet p has been sent by the current node, it would be dropped. Probably a routing loop.

3. If packet p doesn’t need to be broadcasted, function “rt_resolve(Packet *p)” is called. Function “rt_resolve(Packet *p)” is a route handling function.

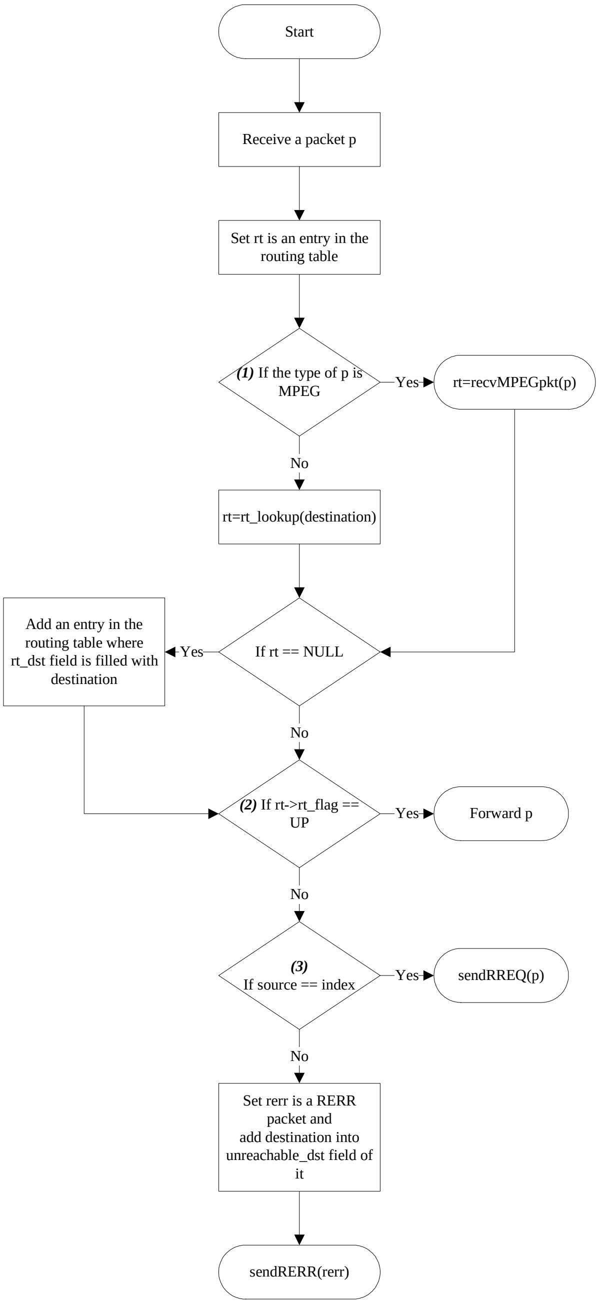


Figure 3.2 Flowchart of Function rt_resolve(Packet *p)

Function "rt_resolve(Packet *p)" is a route handling function. There are three important function blocks in the flowchart of function "rt_resolve(Packet *p)". We describe them below.

1. If the type of packet p is MPEG, function “recvMPEG(Packet *p)” is called. Function “recvMPEG(Packet *p)” handles the packet which type is MPEG.

2. If there is a valid entry to the destination of packet p, packet p would be forwarded.

3. If there isn’t a valid entry to the destination of packet p and the source of packet p is the current node, function “sendRREQ(Packet *p)” is called. Else function “sendRERR(Packet *p)” is called.

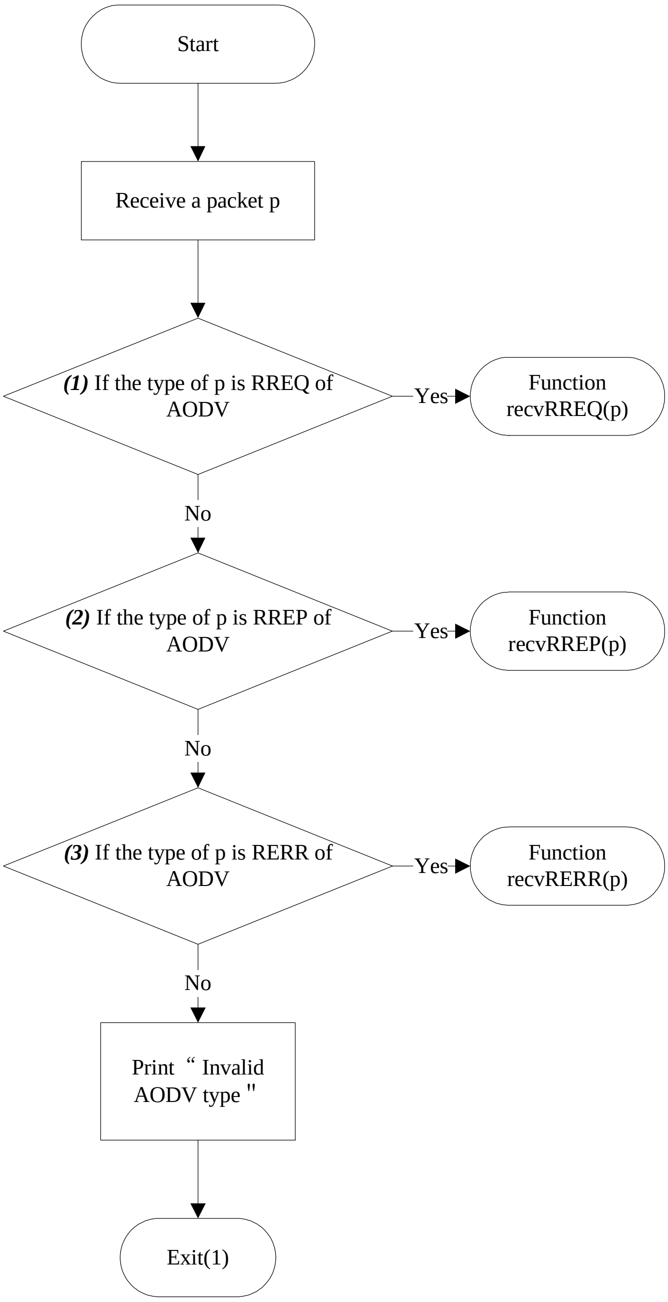


Figure 3.3 Flowchart of Function recvAODV(Packet *p)

Function "recvAODV(Packet *p)" handles the packet which type is AODV. There are three important function blocks in the flowchart of function "recvAODV(Packet *p)". We describe them below.

1. If the type of packet p is RREQ of AODV, function “recvRREQ(Packet *p)” is called.
2. If the type of packet p is RREP of AODV, function “recvRREP(Packet *p)” is called.
3. If the type of packet p is RERR of AODV, function “recvRERR(Packet *p)” is called.

3.4.2Processing and Forwarding Route Requests

When a node receives a RREQ, it checks to determine whether it has received a RREQ with the same Originator IP Address and RREQ ID.

If a node receives a RREQ for the first time, we describe actions below. The node searches for a reverse route to the Originator IP Address. If there is no reverse route to the Originator IP Address, the route is created. If there is a reverse route to the Originator IP Address, the node updates the Originator Sequence Number from the RREQ in its routing table. This reverse route will be needed if the node receives a RREP back to the node that originated the RREQ. When the reverse route is created or updated, the following actions on the route are also carried out:

1. The Originator Sequence Number from the RREQ is compared to the corresponding destination sequence number in the route table entry and copied if greater than the existing value there.

2. The Originator Sequence Number from the RREQ is compared to the corresponding destination sequence number in the route table entry and the hop count from the RREQ is lesser the corresponding hops in the route table entry.

The next hop in the routing table becomes the node from which the RREQ was received (it is obtained from the source IP address in the IP header and is often not equal to the Originator IP Address field in the RREQ message) the hop count is copied from the Hop Count in the RREQ message. The current node can use the reverse route to forward data packets in the same way as for any other route in the routing table.

Notably, RREQ forwarding approach of OHMR is different from AODV. If additional copies of the same RREQ are later received, these packets are discarded in the approach of AODV. In the approach of OHMR intermediate nodes forward the duplicate RREQ that came from two different neighbors. Though the approach would transmit more RREQ packets than the approach of AODV, it enables us to discover both braided and node-disjoint paths. When RREQ packets arrive at its destination, the destination is responsible for judging whether or not the routing path is a node-disjoint path or a braided path.

We give below the pseudo code for Function “recvRREQ (Packet *p)” when a node receives a packet that the type of it is RREQ of AODV. And then Figure 3.4 shows the flowchart of Function “recvRREQ(Packet *p)”.

```
Function recvRREQ (Packet *p)
Begin
    If index is in the route record of p
        Drop p
        Return
    End If
    Add index into the route record of p
    If the RREQ with the same Broadcast ID of packet p has been received
        Set rt0 = rt_lookupmpath(source,1)
        If index == destination
            If CompareNode-Disjoint(rt0,p)
                If rt_lookupnh(source, sender) == NULL
                    Add an entry in the routing table where rt_dst field is filled with
```

```
                source and rt_nexthop field is filled with sender.
            End If
            Generate a RREP and initial its rp_mpath field to 3 and copy the route
            record of p to the route record of RREP
            Send the RREP to source along the route record of it
            Return
        Else If CompareBraided(rt0, p)
            If rt_lookupnh(source, sender) == NULL
                Add an entry in the routing table where rt_dst field is filled with
                source and rt_nexthop field is filled with sender.
            End If
            Generate a RREP and initial its rp_mpath field to 2 and copy the route
            record of p to the route record of RREP
            Send the RREP to source along the route record of it
            Return
        Else
            Drop p
    Else If index != destination && rt0->rt_nop == 1
        If rt_lookupnh(source, sender) == NULL
            Add an entry in the routing table where rt_dst field is filled with source
            and rt_nexthop field is filled with sender.
            rt0->rt_nop++
        End If
        Broadcast p
    Else
        Drop p
Else
    Set rq = AODV RREQ header of p
    Set rt0 = rt_lookup(source)
```

```
    If rt0 == NULL
        Add an entry in the routing table where rt_dst field is filled with source
    End If
    If ( (rq->rq_src_seqno > rt0->rt_seqno ) || ((rq->rq_src_seqno == rt0->rt_seqno)
    && (rq->rq_hop_count < rt0->rt_hops)) )
        Update rt0 in the routing table
        rt0->rt_nop++
        If there are packets queued in the sendbuffer destined for destination
            Send all packets queued in the sendbuffer destined for destination
        End If
    End If
    If index == destination
        Copy the route record of p to the route record of rt0 and set the rp_mpath
        field of rt0 to 1
        Generate a RREP and initial its rp_mpath field to 1 and copy the route
        record of p to the route record of RREP
        Send the RREP to source along the route record of it
    Else If rt->rt_seqno >= rq->rq_dst_seqno
        Generate a RREP and initial its rp_mpath field to 1 and copy the route
        record of p to the route record of RREP
        Send the RREP to source along the route record of it
    Else
        Broadcast p
End
```

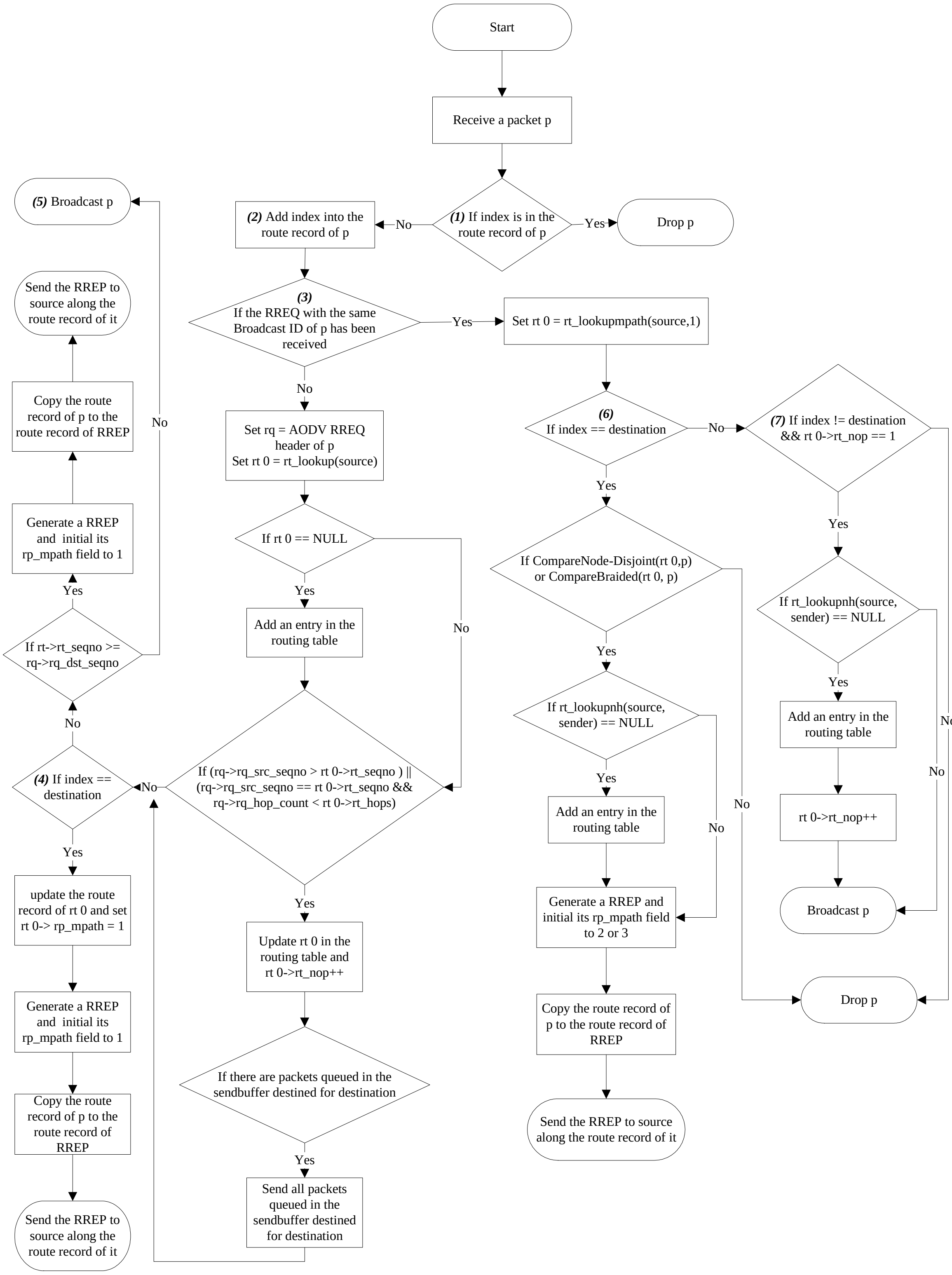


Figure 3.4 Flowchart of Function recvRREQ(Packet *p)

Function “recvRREQ(Packet *p)” is called when the type of the receiving packet is RREQ. There are seven important function blocks in the flowchart of function “recvRREQ(Packet *p)”. We describe them below.

1. This function block checks the IP address of the current node isn’t in the route record of packet p and is used to avoid loop routes.
2. OHMR extends AODV to record the path in RREQ packets. This function block adds the IP address of the current node to the route record of packet p.
3. This function block checks if the RREQ with the same Broadcast ID of packet p has been received. The packet forwarding approach of OHMR is different from AODV. If additional copies of the same RREQ are later received, these packets are discarded in the approach of AODV. In the approach of OHMR intermediate nodes forward the duplicate packets that came from two different neighbors. Though the approach would transmit more RREQ packets than the approach of AODV, it enables us to discover both braided and node-disjoint paths.
4. If the current node receives the RREQ for the first time and the destination of packet p is the current node, the current node sends the RREP to source along the route record of packet p.
5. If the current node receives the RREQ for the first time and the destination of packet p isn’t the current node, the current node broadcasts packet p.
6. If the RREQ with the same Broadcast ID of packet p has been received and the destination of packet p is the current node, the current node is responsible for selecting multiple braided and node-disjoint route paths.
7. This function block checks the RREQ came from two different neighbors at most.

3.4.3Route Selection Method

The destination is responsible for selecting multiple braided and node-disjoint route paths. When receiving the first RREQ, the destination records the route record of RREQ to the route record of the entry to originating node. The route record of RREQ is copied to a RREP and sends the RREP to originating node via the route record of it. Hence the intermediate nodes can forward this packet using the route record of RREP. When the destination receives a duplicate RREQ, it will compare the route record of RREQ to that of the entry to originating node in routing table. If only source node and destination node are the same between them, the path is node-disjoint with the primary path and the destination will set the type of the path to three. If one of intermediate nodes in the route record of the entry to originating node in the routing table is different from all of nodes in the route record of the RREQ, the route is a braided path and the destination will set the type of the route to two. If any node-disjoint or braided path is received, the destination sends the RREP to the source along the route record of RREP. Otherwise, the received RREQ is discarded.

We give below the pseudo code for when destination receives a duplicate RREQ. The destination use Functions “CompareNode-Disjoint(aodv_rt_entry *ety, Packet *p)” and “CompareBraided(aodv_rt_entry *ety, Packet *p)” to judge whether the route record of the packet is a node-disjoint or braided path.

```
Function CompareNode-Disjoint(aodv_rt_entry *ety, Packet *p)
Begin
    For each intermediate node in the route record of ety
        For each intermediate node in the route record of p
            If the node in the route record of ety == the node in the route record of p
```

```
                Return TRUE
            End If
        End For
    End For
    Return FALSE
End

Function CompareBraided(aodv_rt_entry *ety, Packet *p)
Begin
    If ety->rt_full > 0
        For each intermediate node in the route record of the entry
            If the node in the route record of ety is not protected
                For each intermediate node in the route record of p
                    If the node in the route record of ety != the node in the route
                    record of p
                    Else
                            Break
                End For
                If all intermediate nodes in the route record of p are different from the
                node in the route record of the entry
                    Set the node in the route record of ety is protected
                    ety->rt_full --
                    Return TRUE
                End If
            End If
        End For
    Else
        Return FALSE
End
```

3.4.4Generating Route Replies

A node generates a RREP if either:

(i) It is itself the destination.

(ii) It has an active route to the destination, the destination sequence number in the node's existing route table entry for the destination is valid and greater than or equal to the Destination Sequence Number of the RREQ.

When generating a RREP message, a node copies the Destination IP Address, the Originator Sequence Number and the Route Record from the RREQ message into the corresponding fields in the RREP message. Processing is slightly different, depending on whether the node is itself the requested destination, or instead if it is an intermediate node with a fresh enough route to the destination.

Once created, the RREP is unicast to the next hop according to route record field toward the originator of the RREQ. As the RREP is forwarded back towards the node which originated the RREQ message, the Hop Count field is incremented by one at each hop. Thus, when the RREP reaches the originator, the Hop Count represents the distance, in hops, of the destination from the originator.

3.4.5Receiving and Forwarding Route Replies

When a node receives a RREP message, it searches for a route to the previous hop. If needed, a route is created for the previous hop. Next, the node then increments the hop count value in the RREP by one. Then the forward route for this destination is created if it does not already exist. Otherwise, the node compares the Destination Sequence Number in the message with its own stored destination sequence number for the Destination IP Address in the RREP message. Upon comparison, the existing entry is updated only in the following circumstances:

(i) The Destination Sequence Number in the RREP is greater than the node's copy of the destination sequence number and the known value is valid, or

(ii) The sequence numbers are the same, and the previous hop is different from the next hop to destination in route table entry.

If the route table entry to the destination is created or updated, then the following actions occur:

- The route is marked as active,
- The destination sequence number is marked as valid,
- The next hop in the route entry is assigned to be the node from which the RREP is received, which is indicated by the source IP address field in the IP header,
- The hop count is set to the value of the New Hop Count,
- And the destination sequence number is the Destination Sequence Number in the RREP message.

The current node can subsequently use this route to forward data packets to the destination. We give below the pseudo code for Function “recvRREP (Packet *p)” when a node receives a packet that the type of it is RREP of AODV. And then Figure 3.5 shows the flowchart of Function “recvRREQ(Packet *p)”.

```
Function recvRREP (Packet *p)
Begin
    If the RREP with the same sequence number of packet p has been received
        If index == source
```

If rt_lookupnh(destination, sender) == NULL

Add an entry in the routing table where rt_dst field is filled with destination, rt_nexthop field is filled with sender, and rt_mpath field is filled with rp_mpath of p.

Return

Else

If rt_lookupnh(destination, sender) == NULL

Add an entry in the routing table where rt_dst field is filled with destination, rt_nexthop field is filled with sender, and rt_mpath field is filled with rp_mpath of p.

End If

Send p to source along the route record of it

Else

Add an entry in the routing table where rt_dst field is filled with destination , rt_nexthop field is filled with sender and rt_mpath field is filled with rp_mpath of p.

If there are packets queued in the sendbuffer destined for destination

Send all packets queued in the sendbuffer destined for destination

End If

If index == source

Drop p

Else

Send p to source along the route record of it

End

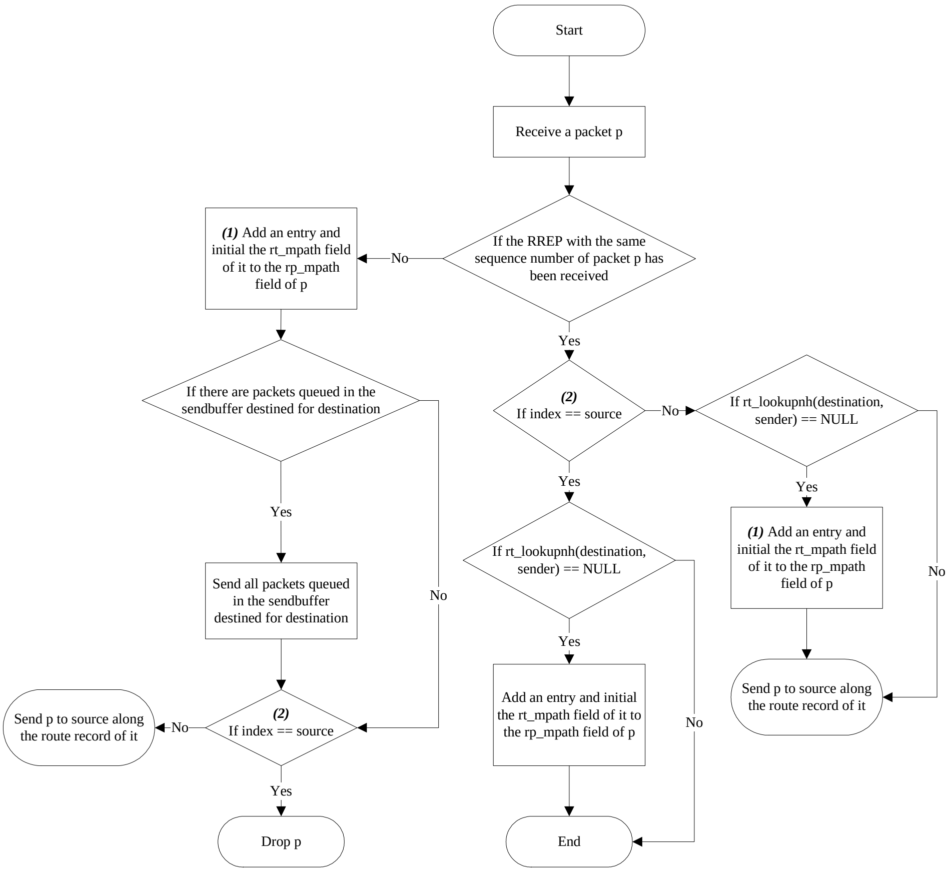


Figure 3.5 Flowchart of Function recvRREP(Packet *p)

Function “recvRREP(Packet *p)” is called when the type of the receiving packet is RREP.. There are two important function blocks in the flowchart of function “recvRREP(Packet *p)”. We describe them below.

1. This function block adds an entry and initials the rt_mpath field of the entry to the rp_mpath field of p.
2. If the current node isn’t the source of packet p, the current node sends packet p to the source along the route record of packet p.

3.5 Route Maintenance

Source node delivers data packets on primary route. Because of mobility, congestion, and packet collisions, primary route can be disconnected. OHMR can recover broken routes immediately. When a node fails to deliver the data packets to the next hop of the route by receiving a link layer feedback from link layer or receives the RERR packet, it removes entries in its route table that uses the broken link and looks up its routing table if there is another entry for the destination. If it has another entry for the destination, data packets therefore can be delivered through the alternate route and are not dropped when route breaks occur. If it has no another entry for the destination, it sends a route error (RERR) packet to the upstream node. When the source has no entry for the destination and the session is still active, it would initiate a new route discovery.

A node initiates processing for a RERR message in three situations:

(i) If it detects a link break for the next hop of an active route in its routing table and it has no other entry to the same destination while transmitting data.

(ii) If it gets a data packet destined to a node for which it does not have an active route.

(iii) It receives a RERR from a neighbor for one or more active routes but it can not protect all of unreachable destinations.

For case (i), the node first makes a list of unreachable destinations consisting of the unreachable neighbor and any additional destinations in the local routing table that use the unreachable neighbor as the next hop.

For case (ii), there is only one unreachable destination, which is the destination of the data packet that cannot be delivered. For case (iii), the list should consist of those destinations in the RERR for which there exists a corresponding entry in the local routing table that has the transmitter of the received RERR as the next hop.

Some of the unreachable destinations in the list could be used by neighboring nodes, and it may therefore be necessary to send a RERR. The RERR should contain those destinations that are part of the created list of unreachable destinations.

The neighboring node(s) that should receive the RERR are all those that belong to at least one of the unreachable destination(s) in the newly created RERR. In case there is only one unique neighbor that needs to receive the RERR, the RERR should be unicast toward that neighbor. Otherwise the RERR is typically sent to the local broadcast address (Destination IP == 255.255.255.255) with the unreachable destinations, and their corresponding destination sequence numbers, included in the packet. The DestCount field of the RERR packet indicates the number of unreachable destinations included in the packet.

Just before transmitting the RERR, certain updates are made on the routing table that may affect the destination sequence numbers for the unreachable destinations. For each one of these destinations, the corresponding routing table entry is updated as follows:

1. The destination sequence number of this routing entry, if it exists and is valid, is incremented for cases (i) and (ii) above, and copied from the incoming RERR in case (iii) above.

2. The entry is invalidated by marking the route entry as invalid

Route maintenance is performed using route error (RERR) packets. When a link failure is detected via link layer feedback, a RERR packet is sent to all of the sources which use that failed link. When a source node receives a RERR, it look up another entry to the destination. If the source has no other used entry to the destination, it initiates a new route discovery. Unused routes in the routing table expire and are deleted.

We give below the pseudo code for Function “recvRERR (Packet *p)” and “handle_link_failure(nsaddr_t broken_neighbor)”. When a node receives a packet that the type of it is RERR of AODV, it implements Function “recvRERR (Packet *p)”. And then Figure 3.5 shows the flowchart of Function “recvRREQ(Packet *p)”. If the link layer feedback is used to detect loss of link, Function “handle_link_failure(nsaddr_t broken_neighbor)” would be called.

```
Function recvRERR(Packet *p)
Begin
    For each unreachable destination called un_dst in the unreachable_dst field of p
        Set hbp is a Boolean variable, initialize to 0
        For each entry called rt in the routing table
            If rt->rt_dst == un_dst && rt->rt_nexthop != sender
                Delete the entry in the routing table where rt_dst is un_dst and
                rt_nexthop is sender
                Set hbp=1
                Break
            End If
        End For
        If hbp == 0
            Delete un_dst in the unreachable_dst field of p
        End If
    End For
    If the number of unreachable destinations in the unreachable_dst field of p > 0
        sendRERR(p)
    Else
        Drop p
End
```

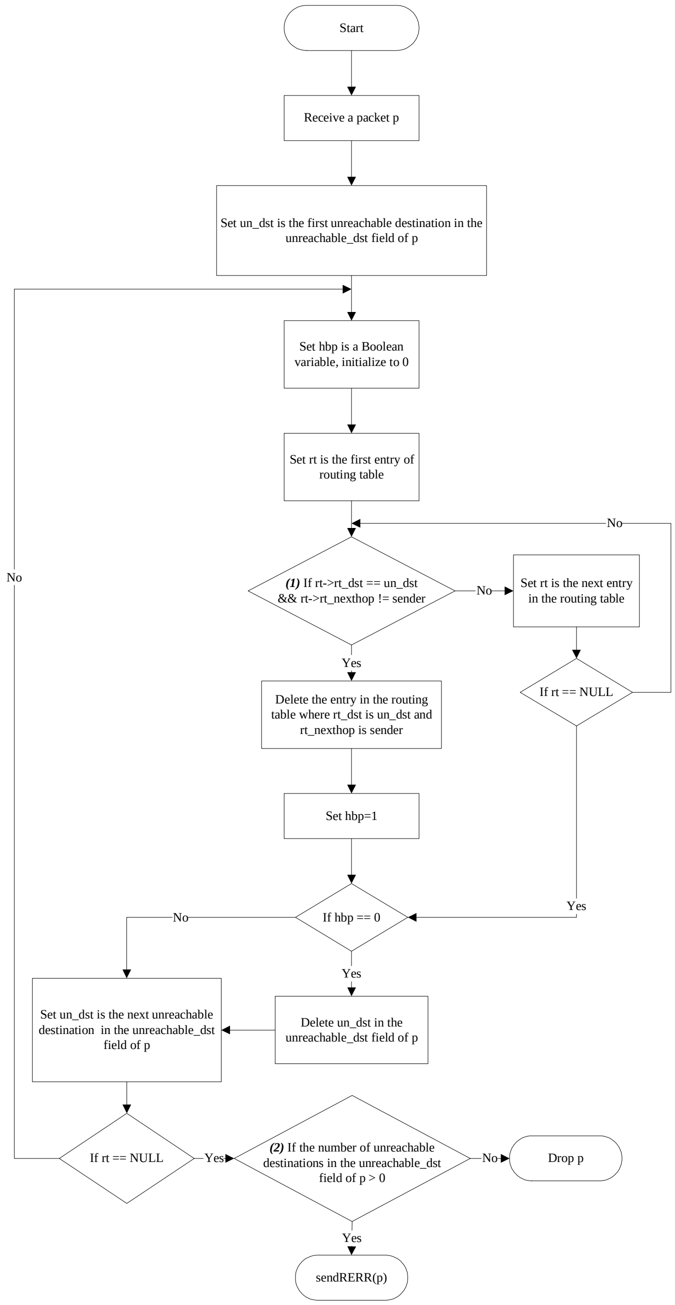


Figure 3.6 Flowchart of Function recvRERR(Packet *p)

Function "recvRERR(Packet *p)" is called when the type of the receiving packet is RERR.. There are two important function blocks in the flowchart of function "recvRERR(Packet *p)". We describe them below.

1. This function block checks if the current node has an alternate path for the unreachable

destinations in the unreachable_dst field of packet p.

2. If there are unreachable destinations which have no alternate path, function "sendRERR(Packet *p)" is called.

```
Function handle_link_failure(nsaddr_t broken_neighbor)
Begin
    For each entry called rt in the routing table
        If rt->rt_nexthop == broken_neighbor
            Set hbp is a Boolean variable, initialize to 0
            For each entry called rt1 in the routing table
                If rt1->rt_dst == rt->rt_dst && rt1->rt_nexthop != broken_neighbor
                    Delete rt
                    Set hbp=1
                    Break
                End If
            End For
            If hbp == 0
                Set rt->rt_flag to invalid
                Set rerr is a RERR packet
                Add broken_neighbor into unreachable_dst field of rerr
                sendRERR(rerr)
            End If
        End If
    End For
End
```

Chapter 4

Multimedia Traffic Allocation Strategy

4.1 Basic Concept of MPEG

Standard MPEG encoders generate three distinct types of frames, namely I, P, and B frames. I frame is encoded in Intra mode, and is essential for the prediction coding of other frames. If part of an I frame is lost, then all frames in the group of pictures (GoP) including this particular frame are impaired. The P frame is encoded in prediction mode, while the B frame is encoded in double prediction mode. As with the I frame, the P frame is also important. If part of a P frame is lost, the impairment propagates the particular P frame, previous B frames, and the following frames in the GoP that includes this P frame. Conversely, if part of a B frame is lost, the impairment propagates solely within that frame.

We illustrate precisely with an example. Every different frame is marked as a serial number in Fig. 4.1. The order of frames is “I1 B1 B2 P1 B3 B4 B5 P2”. I1 is encoded in Intra mode and does not need the prediction coding of other frames. I1 is the prediction coding of P1 and P1 is the prediction coding of P2. I1 and P1 are the prediction coding of B1 and B2. P1 and P2 are the prediction coding of B3, B4, and B5. Note that the order in Fig. 4.1 is the display order of MPEG video frames and it is not the transmission order of MPEG video frames. The transmission order of MPEG video frames is “I1 P1 B1 B2 P2 B3 B4 B5” in Fig. 4.2. Because the P frame is the prediction coding of the B frame when decoding MPEG frames, later P frame is transmitted first. When the receiver receives the B frame, it can decode the B frame directly.

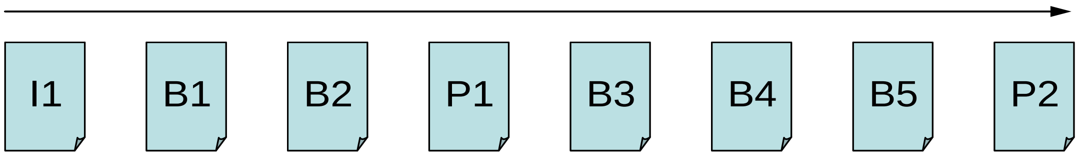


Figure 4.1 An example of the display order for MPEG video frames

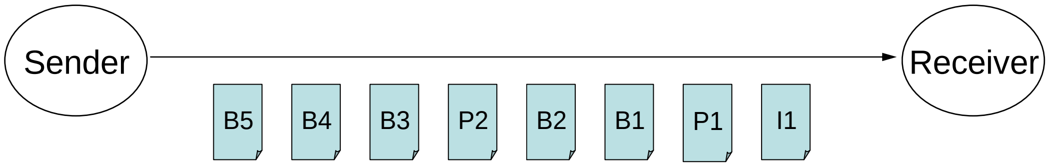


Figure 4.2 The transmission order of Figure 4.1

4.2 Allocation Strategy

Our multimedia traffic allocation strategy is based on the popular standard MPEG coding technique, where a video frame is coded into three distinct types of frames, namely I, P, and B frames. Reception of the I-frame or P-frame can provide low but acceptable quality, while reception of the B-frame can further improve the quality over the base layer alone, but the B-frame cannot be decoded without the I-frame and B-frame. When the I-frame, P-frame and B-frame are transmitted over multiple paths (e.g., two paths), the traffic allocator sends the I-frame and P-frame packets on the primary path and the B-frame packets on the node-disjoint alternate path.

An example of multimedia traffic allocating under OHMR is shown in Figure 4.3. There are three type of path in Figure 4.3. Three type of path are the primary path, the braided alternate path and the node-disjoint alternate path. The primary path is "S->A->B->D" and the rt_mpath field of nodes on the primary path is 1. The node-disjoint alternate path is "S->F->G->H->D" and the rt_mpath field of nodes on the primary path is 3. The braided alternate paths are "S->C->B->D" and "S->A->E->D". The rt_mpath field of nodes which are on the braided alternate path and not on the primary and node-disjoint paths is 2. Source begins to send the I-frame and B-frame packets on the primary path and the B-frame packets on the node-disjoint alternate path. When forwarding paths break, nodes receiving I-frame and P-frame packets or receiving B-frame packets use different order of looking up path in the routing table to forward packets.

We give below the pseudo code for multimedia packet forwarding function "recvMPEGpkt(Packet *p)"

```
Function recvMPEGpkt(Packet *p)
Begin
    If the frame type of p is I-frame or P-frame
        Set rt=rt_lookupmpath(destination,1)
        If rt == NULL
            Set rt = rt_lookupmpath(destination,2);
            If rt == NULL
                Set rt = rt_lookupmpath(destination,3)
            End If
        End If
        Return rt
Else If the frame type of p is B-frame
        Set rt=rt_lookupmpath(destination,3)
        If rt == NULL
            Set rt = rt_lookupmpath(destination,1);
            If rt == NULL
                Set rt = rt_lookupmpath(destination,2)
        End If
        Return rt
    End If
End
```

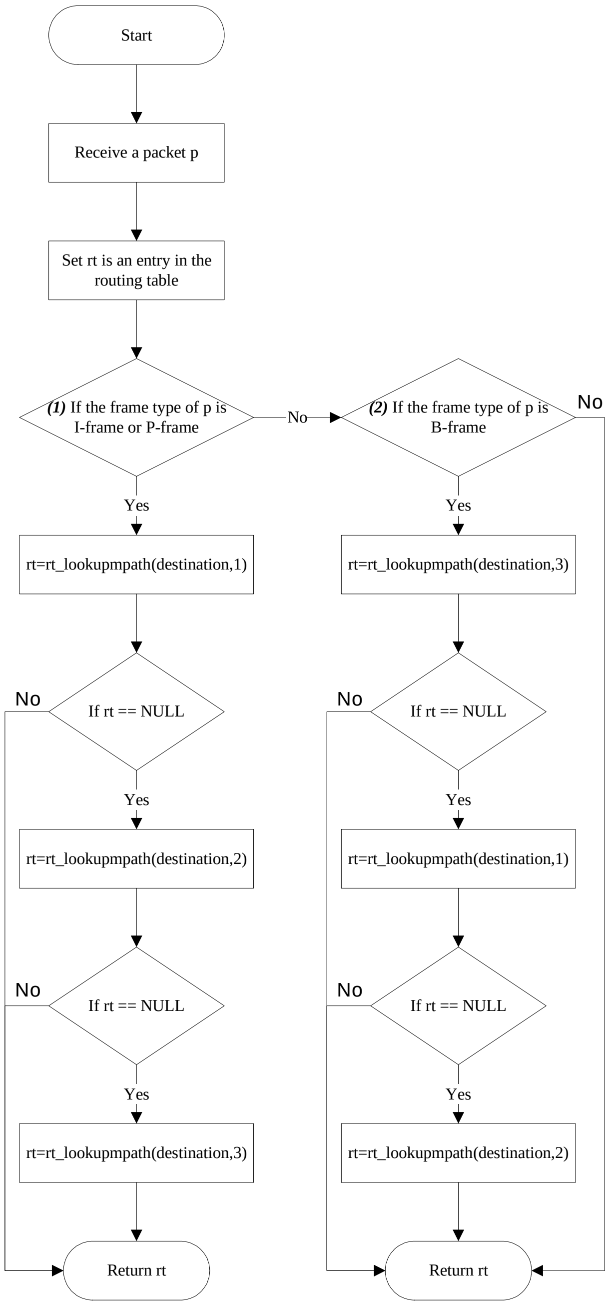


Figure 4.3 Flowchart of Function recvMPEGpkt(Packet *p)

Function "recvMPEG(Packet *p)" is called when the type of the receiving packet is MPEG. There are two important function blocks in the flowchart of function "recvMPEG(Packet *p)". We describe them below.

1. If the frame type of packet p is I-frame or P-frame, the current node looks up the entry which rt_mpath field is one, two or three in turn.
2. If the frame type of packet p is B-frame, the current node looks up the entry which rt_mpath field is three, one or two in turn.

We illustrate some examples when forwarding paths breaks. Following Fig. 4.3, when node A moves and the primary path breaks, node S uses the alternate path which the rt_mpath field is two to forward I-frame and P-frame packets in Fig. 4.4(a). And then, when node C moves and the braided path breaks, node S uses the alternate path which the rt_mpath field is three to forward I-frame and P-frame packets in Fig. 4.4(b).

Following Fig. 4.3, when node G moves and the primary path breaks, node S uses the path which the rt_mpath field is one to forward B-frame packets in Fig. 4.5(a). And then, when node B moves and the primary path breaks, node A uses the alternate path which the rt_mpath field is two to forward B-frame in Fig. 4.5(b).

Generally, a multihop wireless path is up or down for random periods of time, leading to bursty packet losses. A I-frame or P-frame packet loss is likely to be experiencing a packet loss burst. I-frame and P-frame are important. OHMR finds an alternate braided path for each node on the primary path. The primary path with multiple alternate braided paths has higher packet delivery rate than the alternate node-disjoint path so the primary path is used for the I-frame and P-frame. Since the primary path and the alternate node-disjoint path are not correlated. Source uses the alternate node-disjoint path to provide load balancing in the beginning. Moreover, I-frame and P-frame packet transmission using the alternate node-disjoint path could have higher success probability when the primary path and the alternate braided paths can not be used to forward packets.

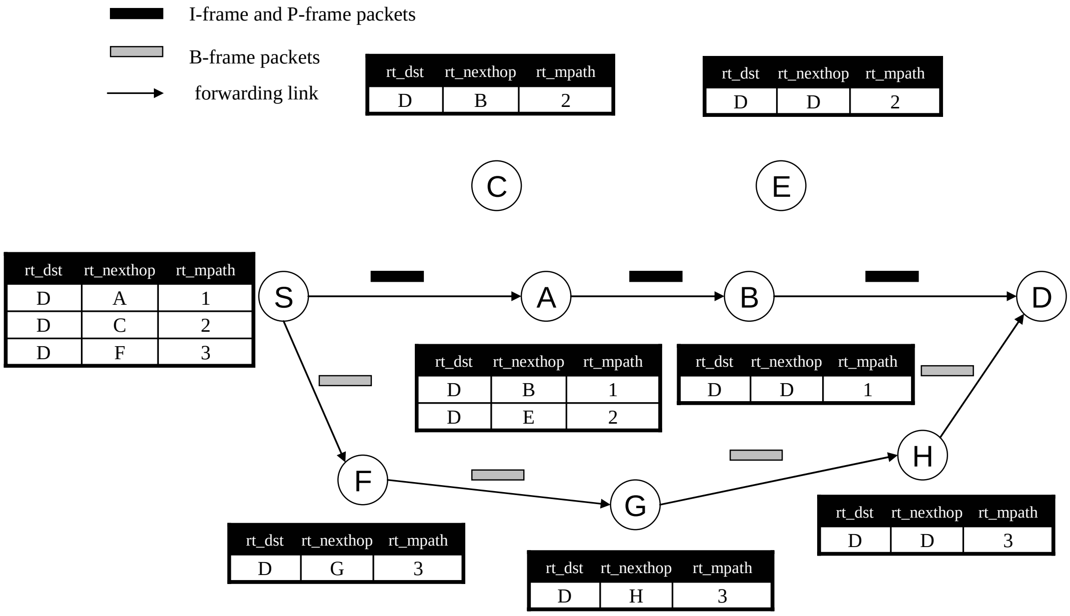
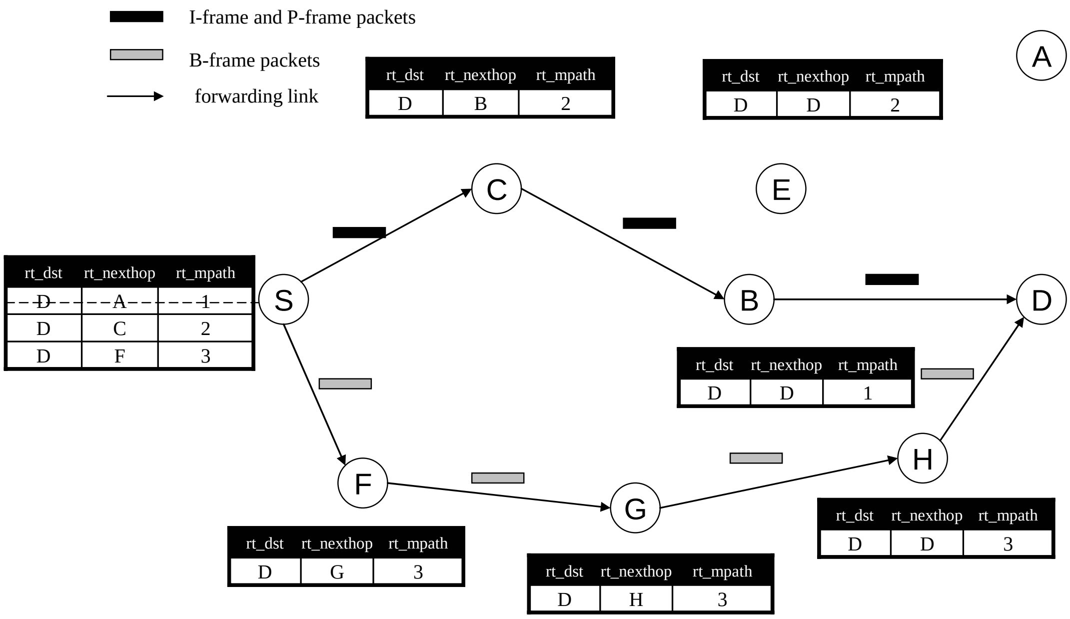
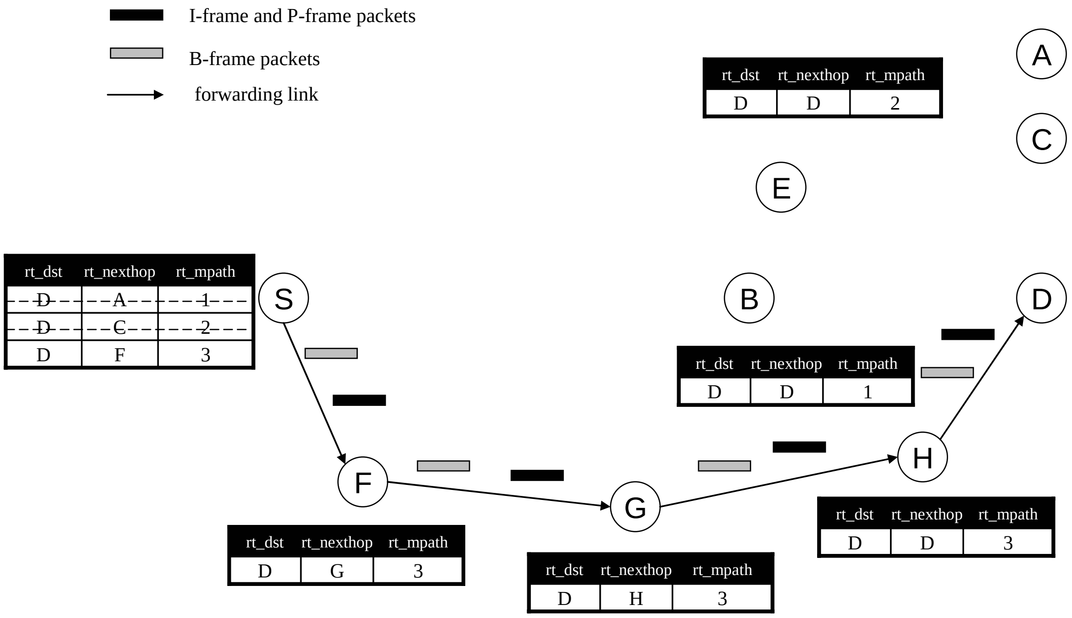


Figure 4.4 An example of multimedia traffic allocating

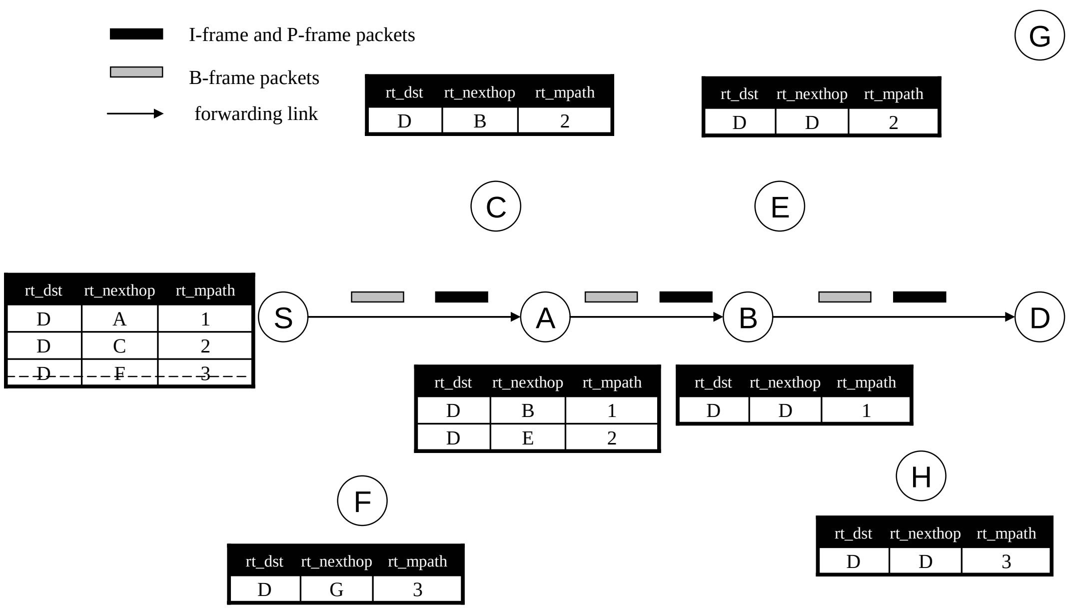


(a)

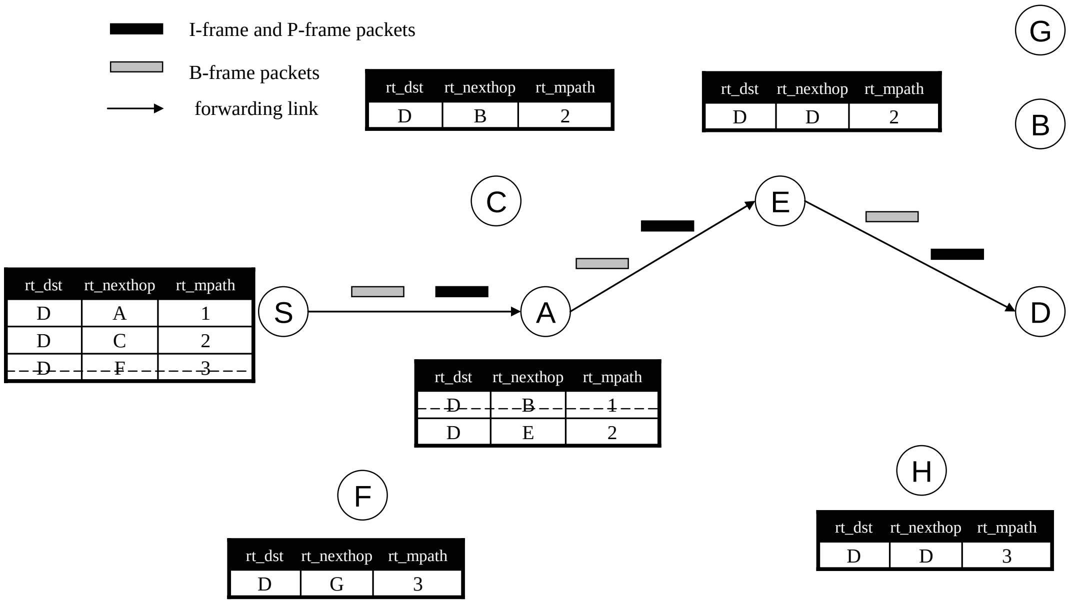


(b)

Figure 4.5 Nodes move on I-frame and P-frame forwarding path



(a)



(b)

Figure 4.6 Nodes move on B-frame forwarding path

Chapter 5

Performance Evaluations

5.1 Theoretical Analysis

5.1.1Evaluation of Query Flooding Frequency

Nasipuri and Das [5] proposed an analytical modeling framework for determining the time interval between successive route discoveries. This study uses a similar analysis technique to analyze the performance of the OHMR protocol under the fault model with unreliable wireless links. Link failures may occur as a result of energy dissipation or localized environmental effects at low deployment densities.

Consider the primary path from source to destination that consists of a sequence of *k* wireless links over *k*-1 intermediate node. Figure 5.1 illustrates a hybrid multipath built by the OHMR protocol. Let N_i be *i*-th node and L_i be *i*-th link in the primary path. P_{Bi} is part of *i*-th braided path and this part of braided path is node-disjoint with the primary path. P_{Bi} can bypass L_i. P_{NDi} is *i*-th node-disjoint path. In our analytical model, the hybrid multipath includes a total of k backup path P_{Bi} and a total of two backup path P_{NDi}. Note that it may not always be possible for all nodes to establish multiple backup paths. This is particularly possible for the case of sparse networks. However, for simplicity, the scenario in Fig. 5.1 is assumed through the present analytical model.

The link L_i on the primary path can be replaced by backup paths, i.e. paths P_{Bi}, P_{Bi-1}, P_{NB1} and P_{NB2}. Let $\overline{L_i}$ denote the event of link L_i failure, $\overline{P_{Bi}}$ the event of path P_{Bi} failure, and $\overline{P_{NDi}}$ the event of P_{NDi} failure. The time until the next route discovery, T, can then be interpreted as the time until event E occurs, where E is described by the following logical expression:

$$E=\overline{L_1}\,\overline{P_{B1}}\,\overline{P_{ND1}}\,\overline{P_{ND2}}+\overline{L_2}\,\overline{P_{B2}}\,\overline{P_{B1}}\,\overline{P_{ND1}}\,\overline{P_{ND2}}+\cdots+\overline{L_{k-1}}\,\overline{P_{Bk-1}}\,\overline{P_{Bk-2}}\,\overline{P_{ND1}}\,\overline{P_{ND2}}+\overline{L_k}\,\overline{P_{Bk-1}}\,\overline{P_{ND1}}\,\overline{P_{ND2}} \qquad (1)$$

The *i*-th term of the right-hand side in this expression represents the event which starts with the failure of L_i and leads to a new route discovery. For example, the second term represents the following sequence of events:

- L_2 breaks on the primary path, prompting N_2 to use the backup path P_{B2} around L_2
- The backup path P_{B2} breaks, prompting N_1 to use another backup path, P_{B1}
- The backup path P_{B1} breaks, prompting N_1 to use the backup path, P_{ND1}
- The backup path P_{ND1} breaks, prompting N_1 to use the backup path, P_{ND2}
- This route fails when P_{ND2} breaks, causing N_1 to initiate a new route discovery

Hence, starting with the breakage of L_2, the events leading to a new route discovery from N_1 are $\overline{L_2}\,\overline{P_{B2}}\,\overline{P_{B1}}\,\overline{P_{ND1}}\,\overline{P_{ND2}}$. The other terms can be derived in a similar way from Fig. 5.1. We also evaluate event E about the braided multipath in Fig. 5.2, the node-disjoint multipath in Fig. 5.3 and the intermediate node-disjoint multipath[5] in Fig. 5.4 respectively. They are described by the following logical expression in turn:

$$E_{Braid}=\overline{L_1}\,\overline{P_{B1}}+\overline{L_2}\,\overline{P_{B2}}\,\overline{P_{B1}}+\cdots+\overline{L_{k-1}}\,\overline{P_{Bk-1}}\,\overline{P_{Bk-2}}+\overline{L_k}\,\overline{P_{Bk-1}} \qquad (2)$$

$$E_{Node-Disjoint}=\overline{P_{Primary}}\,\overline{P_{ND1}}\,\overline{P_{ND2}} \qquad (3)$$

$$E_{IntermediateND}=\overline{L_1}\,\overline{P_{ND1}}+\overline{L_2}\,\overline{P_{ND2}}\,\overline{P_{ND1}}+\overline{L_3}\,\overline{P_{ND3}}\,\overline{P_{ND2}}\,\overline{P_{ND1}}\cdots+\overline{L_k}\,\overline{P_{NDk}}\,\overline{P_{NDk-1}}\cdots\overline{P_{ND1}} \qquad (4)$$

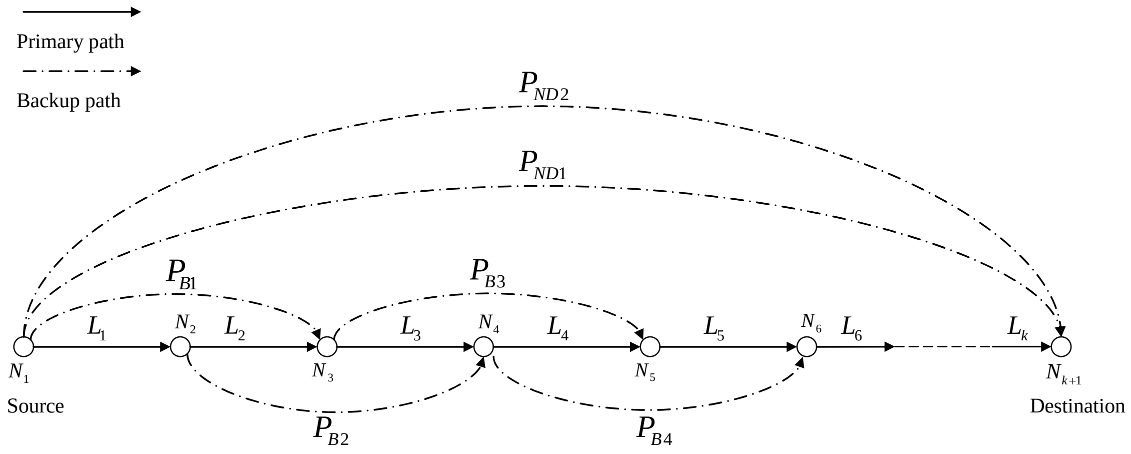


Figure 5.1 Illustration of a hybrid multipath.

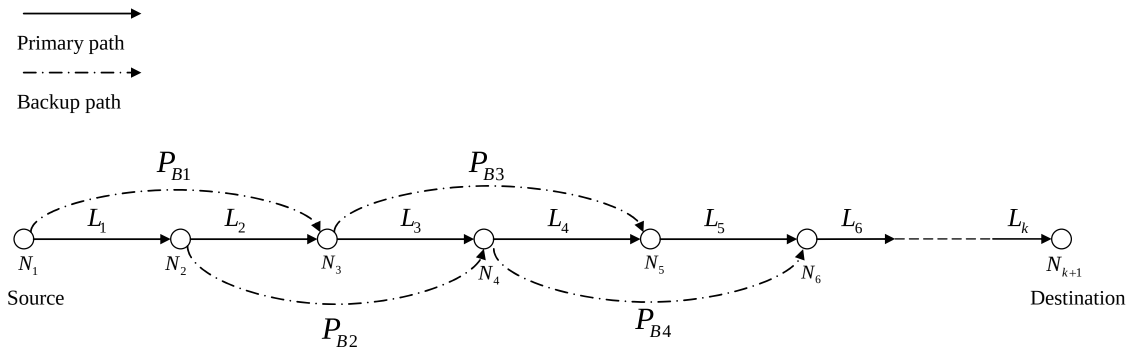


Figure 5.2 Illustration of a braided multipath.

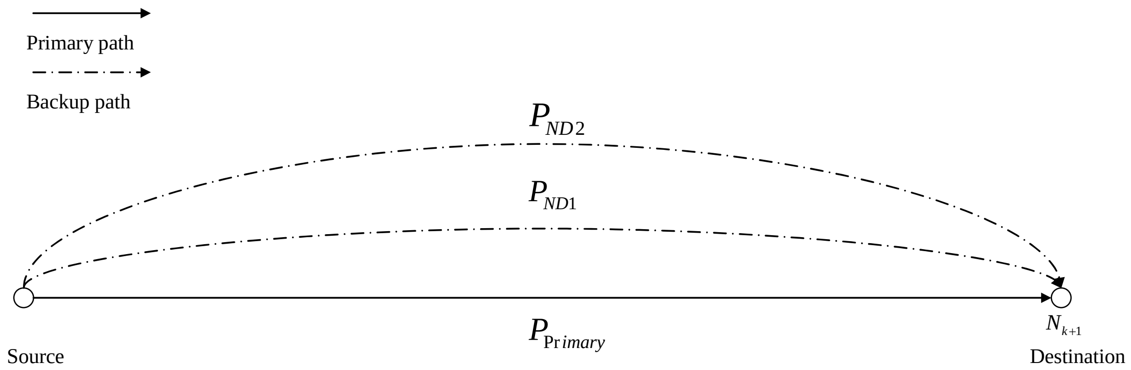


Figure 5.3 Illustration of a node-disjoint multipath.

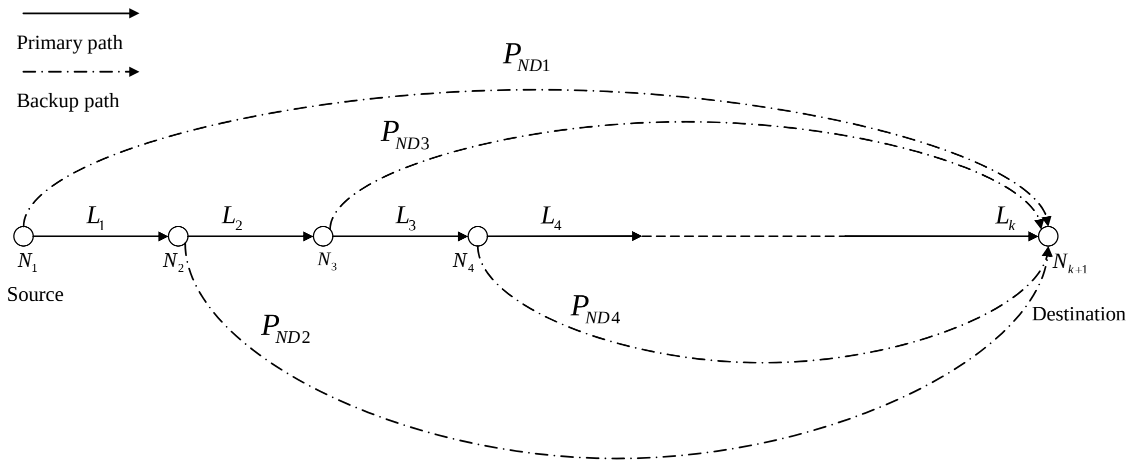


Figure 5.4 Illustration of an intermediate node-disjoint multipath.

In the current analysis, the lifetime of a wireless link is represented by a random variable. Consider a path *P* from the source to the destination composed of a sequence of *k* wireless links. L_i is *i*-th link in the route. The lifetime of L_i is denoted by X_{L_i}. Further, assume that $X_{L_i}, i = 1,2,\cdots,k$, are independent and identically distributed (iid) exponential random variables, each with a mean of l. If backup paths P_{Bi} and P_{NDi} consist of k_{Bi} and k_{NDi} links, respectively, $X_{P_{Bi}}$ and $X_{P_{NDi}}$ are also exponential random variables and have means of l/k_{Bi} and l/k_{NDi}. Note that according to the current assumptions, X_{L_i}, $X_{P_{Bi}}$ and $X_{P_{NDi}}$ are independent.

The time after which none of the routes are useful is represented by a random variable *T*, where:

$$
\begin{aligned}
T = &\min(\max(X_{L_1}, X_{P_{B1}}, X_{P_{ND1}}, X_{P_{ND2}}), \max(X_{L_2}, X_{P_{B2}}, X_{P_{B1}}, X_{P_{ND1}}, X_{P_{ND2}}), \\
&\cdots, \max(X_{L_{k-1}}, X_{P_{Bk-1}}, X_{P_{Bk-2}}, X_{P_{ND1}}, X_{P_{ND2}}), \max(X_{L_k}, X_{P_{Bk-1}}, X_{P_{ND1}}, X_{P_{ND2}})) \qquad (5)
\end{aligned}
$$

T represents the time between successive route discoveries. In the present analysis, it is assumed that the end-to-end packet transmission latency is very small compared to the interval between route changes. Therefore, the time spent in discovering routes is negligible and can be ignored.

For simplicity, the solution of the random variable T expressed in Eq. (5) can be separated into two steps. In the first step, the maximum value of at least two iid exponential random variables is obtained, while in the second step, the minimum value of at least two iid exponential random variables is obtained.

The first step considers the case of M iid exponential random variables, $X_1, X_2, \cdots X_M$, where the pdf of X_i is $f_{X_i}(t) = \lambda_i e^{-\lambda_i t}, i = 1,2,.....,M$. X_i is one of $X_{L_i}, X_{P_{Bi}}$ or $X_{P_{NDi}}$, where these X_i are independent. It is assumed that the pdf of T_{fir} is given by $T_{fir} = \max(X_1, X_2,..., X_M)$. The cumulative distribution function (cdf) of T_{fir}, $F_{T_{fir}}(t)$ is then obtained as:

$$\begin{aligned}
F_{T_{fir}}(t) &= P[T_{fir} \leq t] \\
&= P[\max(X_1, X_2, \cdots X_M) \leq t] \\
&= P[(X_1 \leq t) \cap (X_2 \leq t) \cap \cdots \cap (X_M \leq t)] \\
&= \prod_{i=1}^{M} F_{X_i}(t) \qquad (6)
\end{aligned}$$

where $F_{X_i}(t) = \int_0^t f_{X_i}(t)dt = 1 - e^{-\lambda_i t}$ is the cdf of X_i. Differentiating Eq. (6) with respect to t, gives the pdf of T_{fir}, as shown in Eq. (7), from which the maximum value of at least two iid exponential random variables can be obtained.

$$
\begin{aligned}
f_{T_{fir}}(t) &= F'_{T_{fir}}(t) \\
&= \lambda_1 e^{-\lambda_1 t}(1-e^{-\lambda_2 t})(1-e^{-\lambda_3 t})\cdots(1-e^{-\lambda_M t}) \\
&\quad + \lambda_2 e^{-\lambda_2 t}(1-e^{-\lambda_1 t})(1-e^{-\lambda_3 t})\cdots(1-e^{-\lambda_M t}) + \cdots \\
&\quad + \lambda_M e^{-\lambda_M t}(1-e^{-\lambda_1 t})(1-e^{-\lambda_2 t})\cdots(1-e^{-\lambda_{M-1} t})
\end{aligned} \tag{7}
$$

$$
where\ \lambda_i = \begin{cases} \dfrac{1}{l} & for\ X_i = X_{L_i} \\ \dfrac{k_{Bi}}{l} & for\ X_i = X_{P_{Bi}} \\ \dfrac{k_{NDi}}{l} & for\ X_i = X_{P_{NDi}} \end{cases}
$$

Let the random variable $\max(X_{L_i}, X_{P_{Bi}}, X_{P_{Bi-1}}, X_{P_{ND1}}, X_{P_{ND2}})$ be denoted by Z_i for 0<i<k, $\max(X_{L_1}, X_{P_{B1}}, X_{P_{ND1}}, X_{P_{ND2}})$ be denoted by Z_1, and $\max(X_{L_k}, X_{P_{Bk-1}}, X_{P_{ND1}}, X_{P_{ND2}})$ be denoted by Z_k. Hence, from Eq. (7), the pdf of Z_i, $f_{Z_i}(t)$ is given by:

$$
f_{Z_i}(t) = \begin{cases}
\lambda_{L_1} e^{-\lambda_{L_1} t}(1-e^{-\lambda_{P_{B1}} t})(1-e^{-\lambda_{P_{ND1}} t})(1-e^{-\lambda_{P_{ND2}} t}) + \lambda_{P_{B1}} e^{-\lambda_{P_{B1}} t}(1-e^{-\lambda_{L_1} t})(1-e^{-\lambda_{P_{ND1}} t})(1-e^{-\lambda_{P_{ND2}} t}) \\
\quad + \lambda_{P_{ND1}} e^{-\lambda_{P_{ND1}} t}(1-e^{-\lambda_{L_1} t})(1-e^{-\lambda_{P_{B1}} t})(1-e^{-\lambda_{P_{ND2}} t}) + \lambda_{P_{ND2}} e^{-\lambda_{P_{ND2}} t}(1-e^{-\lambda_{L_1} t})(1-e^{-\lambda_{P_{B1}} t})(1-e^{-\lambda_{P_{ND1}} t}) & for\ \ i=1 \\
\lambda_{L_i} e^{-\lambda_{L_i} t}(1-e^{-\lambda_{P_{Bi}} t})(1-e^{-\lambda_{P_{Bi-1}} t})(1-e^{-\lambda_{P_{ND1}} t})(1-e^{-\lambda_{P_{ND2}} t}) + \lambda_{P_{Bi}} e^{-\lambda_{P_{Bi}} t}(1-e^{-\lambda_{P_{L_i}} t})(1-e^{-\lambda_{P_{Bi-1}} t})(1-e^{-\lambda_{P_{ND1}} t})(1-e^{-\lambda_{P_{ND2}} t}) \\
\quad + \lambda_{P_{Bi-1}} e^{-\lambda_{P_{Bi-1}} t}(1-e^{-\lambda_{P_{L_i}} t})(1-e^{-\lambda_{P_{Bi}} t})(1-e^{-\lambda_{P_{ND1}} t})(1-e^{-\lambda_{P_{ND2}} t}) + \lambda_{P_{ND1}} e^{-\lambda_{P_{ND1}} t}(1-e^{-\lambda_{P_{L_i}} t})(1-e^{-\lambda_{P_{Bi}} t})(1-e^{-\lambda_{P_{Bi-1}} t})(1-e^{-\lambda_{P_{ND2}} t}) \\
\quad + \lambda_{P_{ND2}} e^{-\lambda_{P_{ND2}} t}(1-e^{-\lambda_{P_{L_i}} t})(1-e^{-\lambda_{P_{Bi}} t})(1-e^{-\lambda_{P_{Bi-1}} t})(1-e^{-\lambda_{P_{ND1}} t}) & for\ \ 0<i<k \\
\lambda_{L_k} e^{-\lambda_{L_k} t}(1-e^{-\lambda_{P_{Bk-1}} t})(1-e^{-\lambda_{P_{ND1}} t})(1-e^{-\lambda_{P_{ND2}} t}) + \lambda_{P_{Bk-1}} e^{-\lambda_{P_{Bk-1}} t}(1-e^{-\lambda_{L_1} t})(1-e^{-\lambda_{P_{ND1}} t})(1-e^{-\lambda_{P_{ND2}} t}) \\
\quad + \lambda_{P_{ND1}} e^{-\lambda_{P_{ND1}} t}(1-e^{-\lambda_{L_1} t})(1-e^{-\lambda_{P_{Bk-1}} t})(1-e^{-\lambda_{P_{ND2}} t}) + \lambda_{P_{ND2}} e^{-\lambda_{P_{ND2}} t}(1-e^{-\lambda_{L_1} t})(1-e^{-\lambda_{P_{Bk-1}} t})(1-e^{-\lambda_{P_{ND1}} t}) & for\ \ i=k
\end{cases}
$$

$$
where \begin{cases} \lambda_{L_i} = \dfrac{1}{l} \\ \lambda_{P_{Bi}} = \dfrac{k_{Bi}}{l} \\ \lambda_{P_{ND1}} = \dfrac{k_{ND1}}{l} \\ \lambda_{P_{ND2}} = \dfrac{k_{ND2}}{l} \end{cases} \tag{8}
$$

In the second step of the solution procedure for the random variable *T*, combining Eqs. (5) and (8) gives the pdf of $T = \min(Z_1, Z_2, ..., Z_k)$. The cdf of T, $F_T(t)$ is then given by:

$$
\begin{aligned}
F_T(t) &= P[\min(F_{Z_1}, F_{Z_2}, ..., F_{Z_k})] \\
&= 1 - P[\max\left((1-F_{Z_1}), (1-F_{Z_2}), ..., (1-F_{Z_k})\right)] \\
&= 1 - \prod_{i=1}^{k}\left(1 - F_{X_{Z_i}}(t)\right)
\end{aligned} \tag{9}
$$

Differentiating Eq. (9) with respect to t, yields the pdf of T, $f_T(t)$, as shown in Eq. (10), from which the minimum value of at least two iid exponential random variables can be obtained.

$$\begin{aligned} f_T(t) &= F'_T(t) \\ &= \sum_{i=1}^{k}\left(f_{Z_i}(t) \prod_{j=1, j\neq i}^{k}\left(1 - F_{Z_i}(t)\right)\right) \end{aligned} \tag{10}$$

From Eqs. (8) and (10), it can be shown that given a knowledge of the hop-wise lengths of all the routes, the expected value of T can be derived from :

$$E[T] = \int_0^{\infty} t * f_T(t)dt \tag{11}$$

From Eqs. (11), the frequency of route discoveries can be derived from:

$$Frequency = \frac{1}{E[T]} \tag{12}$$

Based on the analytical procedure presented above, this section investigates the performance benefits of various multipath routing protocols. In the OHMR protocol, the performance is dependent on the number of links in the backup paths P_{Bi} and P_{NDi}. The actual number of links in the primary and backup paths, i.e. k, k_{Bi}, and k_{NDi}, and the mean value, l, are dependent on the dynamic conditions of the network. To assess the performance improvement obtained from the OHMR protocol, this study assumes that the maximum number of backup paths P_{Bi} is *k* and the maximum number of backup paths P_{NDi} is two, where *k* is the length of the primary path. Note that in the following, k_{Bi} and k_{NDi} represent the lengths of P_{Bi} and P_{NDi}, respectively. In evaluating the performance improvement provided by OHMR, three different values of parameters k_{Bi} and k_{NDi} are considered, i.e. Case A, Case B and Case C. In Case A, we assume that all of backup paths between S and D have the same length as the primary path. This implies the “best case” scenario for the hybrid multipath. In Cases B and C, k_{Bi} and k_{ND} increase by one and two, respectively.

The path lengths of P_{Bi} and P_{NDi} are:

Case A: k_{Bi} =2, for $1 \le i \le k$

k_{NDj} =k, for $1 \le j \le 2$

Case B: k_{Bi} =3, for $1 \le i \le k$

k_{NDj} =k+1, for $1 \le j \le 2$

Case C: k_{Bi} =4, for $1 \le i \le k$

k_{NDj} =k+2, for $1 \le j \le 2$

For each of these three cases, this study determines the frequency of route discoveries from the expected time interval between route discoveries (E[T]), as given by Eq.(12) .

5.1.2Numerical Results

Figure 4 plots the frequency of route discoveries under OHMR at the source with different values of the primary path length for Cases A, B and C. Note that in this figure (and in all subsequent figures), the mean lifetime of a wireless link (l) is assumed to be 5. As expected, the OHMR protocol performs significantly better than the single path routing strategy. In all three cases, route discovery is initiated less frequently under the OHMR protocol than in the single path case. It can be seen that the relative advantage of the OHMR protocol increases as the primary route becomes longer. When the primary path has a length of five links, the frequency of route discoveries under the OHMR protocol relative to that of the single path routing protocol reduces by 58%. Similarly, when the length of the primary path increases to eight links, the frequency of route discoveries under OHMR relative to that of the single path routing protocol decreases by 67%. Therefore, the OHMR protocol is a more appropriate choice for primary paths containing a larger number of links.

It is apparent that the frequency of route discoveries for Case A is lower than that for Case B and that the frequency of route discoveries for Case B is lower than that for Case C. It can be seen that the relative advantage of multipath routing diminishes as the backup path becomes longer. This is reasonable, since backup routes become longer, and longer routes typically break more easily than shorter routes.

In comparing the relative performances of the node-disjoint multipath, the intermediate node-disjoint multipath [5], the braided multipath and the OHMR protocols, Figure 5.6, Figure 5.7 and Figure 5.8 plot the frequency of route discoveries at the source with different values of the primary route path length for Cases A, B and C, respectively. In Fig. 5.6, this study considered Case A and assumed that the "best" backup paths were available for each protocol. The frequency of route discoveries under OHMR is least. Note that this case is very unlikely to occur in practice. So Figure 5.7 and Figure 5.8 plot the worse cases. In all three cases, the frequency of route discovery is smallest under the OHMR protocol than in other multipath protocols.

This study assumes the number of node-disjoint backup path in the hybrid multipath and the node-disjoint multipath is one. The frequency of route discoveries at the source with different values of the primary route path length under four different multipath protocols for case A is shown in Fig. 5.9. The frequency of route discoveries under OHMR is close to that under braided multipath and is still less than that under braided multipath. So through this study, we show the hybrid multipath comprising multiple node-disjoint and braided routing paths can reduce the frequency of route discoveries.

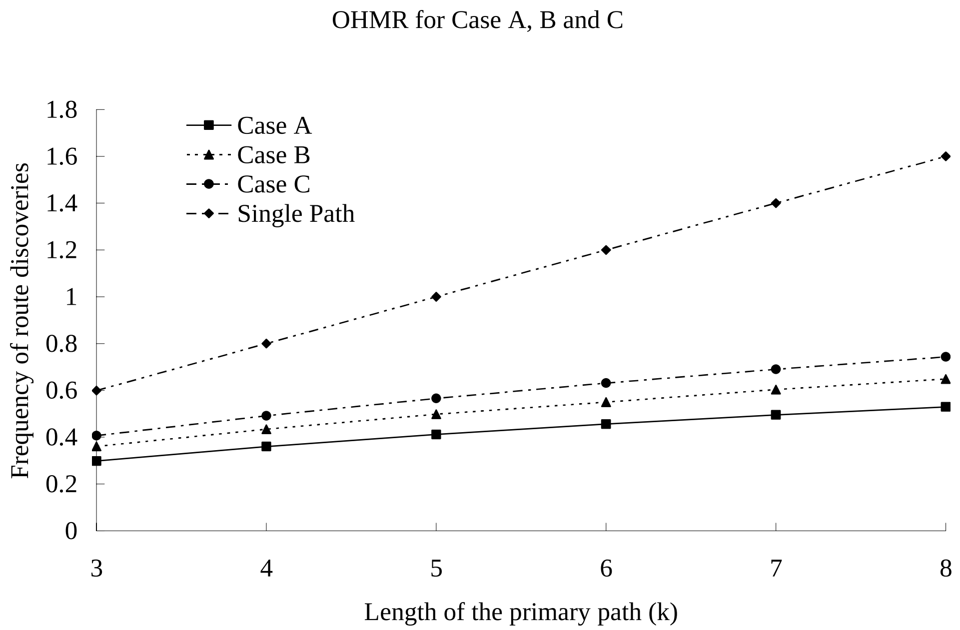


Figure 5.5: Performance of OHMR with different primary path lengths. Three cases are compared with the single path case. Performance of single path routing is shown for reference.

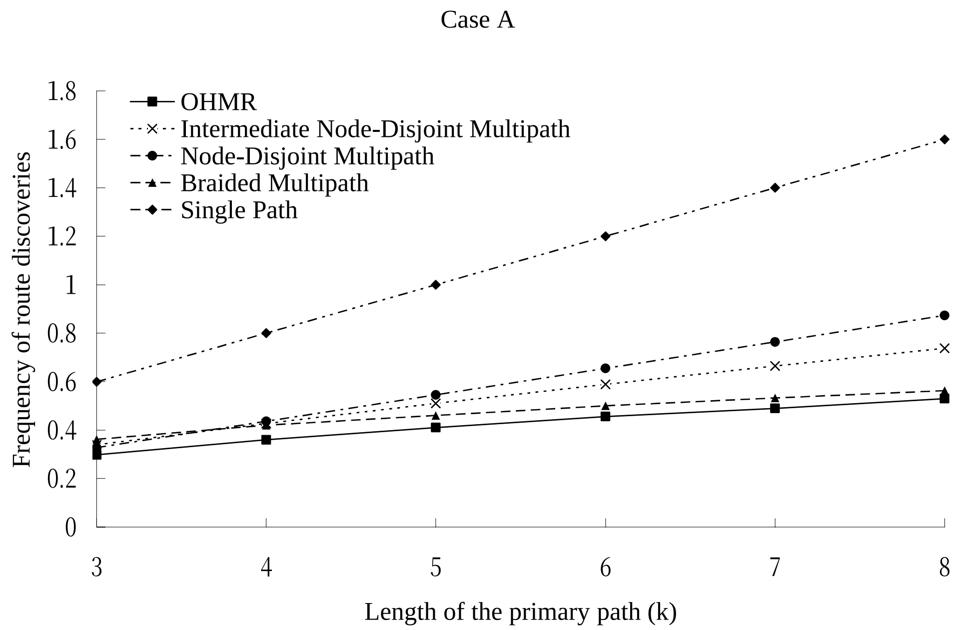


Figure 5.6: Comparison for case A between four different multipath protocols with different primary path lengths.

Case B

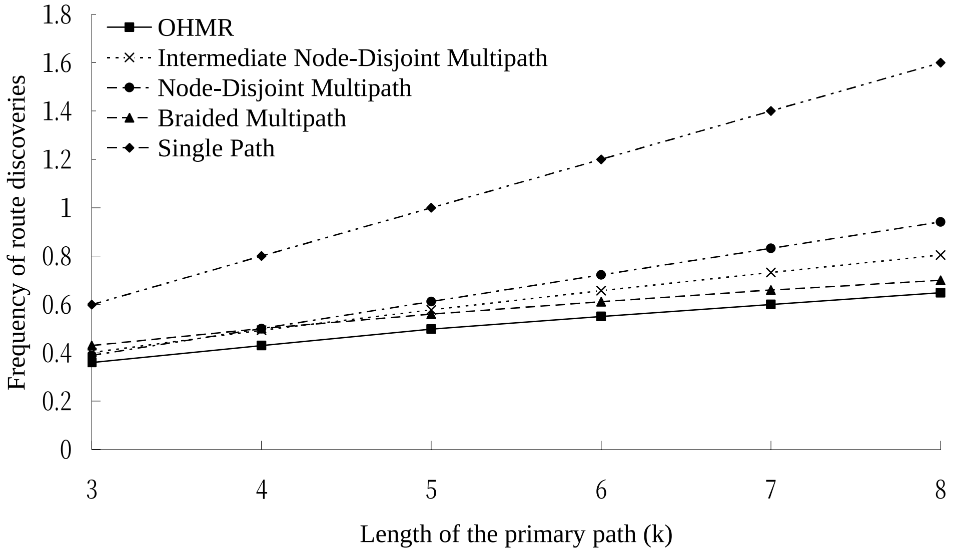


Figure 5.7: Comparison for case B between four different multipath protocols with different primary path lengths.

Case C

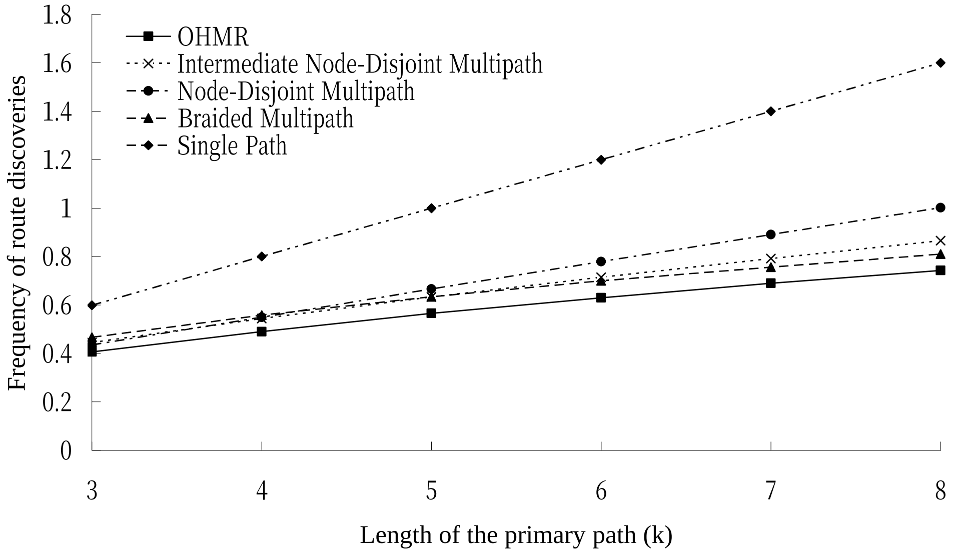


Figure 5.8: Comparison for case C between four different multipath protocols with different primary path lengths.

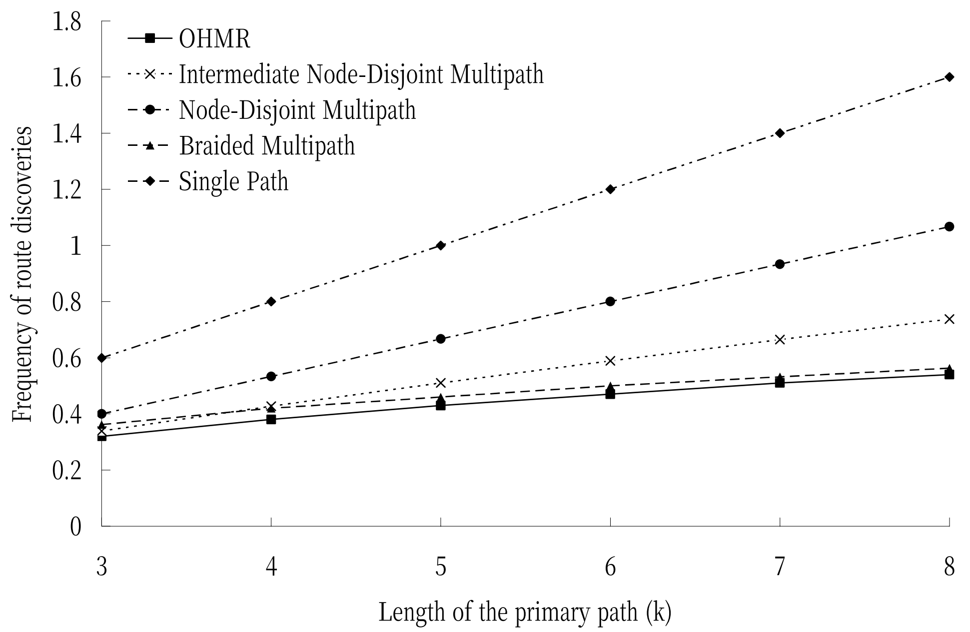


Figure 5.9: Comparison for case A between four different multipath protocols with different primary path lengths. The number of node-disjoint backup path in the hybrid multipath and the node-disjoint multipath is one.

5.2 Simulations with Constant Bit Rate Traffic

5.2.1Simulation Environment

The simulation of this integrated networking system is based on the network simulator (ns-2). It is a discrete event-driven simulator providing support for most of TCP, routing, and multicast protocols over wired and wireless networks. It was developed in the VINT project at UC Berkeley, and is currently maintained at USC. The CMU extension of ns-2[12] provides some wireless supports. It is possible to construct detailed and accurate simulations for wireless LANs and MANETs. Our simulation modeled a network with 50 mobile nodes placed randomly in a rectangular field, 1500mx300m area. A rectangular shape area is chosen to make the average length of routes longer, so as to observe more route breaks during the simulation. Channel capacity was 2 Mb/s. Each run is executed for

300 seconds of simulation time. For scenario creation, two kinds of scenario files are used. The first is a movement pattern file that describes the movement that all nodes should undergo during the simulation. The random waypoint mobility model [11] was used. Each node randomly selects a position, and moves toward that location with a speed between the minimum and the maxi-mum speed. Once it reaches that position, it becomes stationary for a predefined pause time. After that pause time, it selects another position and repeats the process. We varied the pause time to simulate different mobility degrees. Longer pause time implies less mobility. The minimum and the maximum speed were zero and 20 m/s, respectively. The second is a traffic pattern file that uses a traffic generator to simulate constant bit rate (CBR) sources. CBR uses UDP as its transport protocol. The sources and the destinations are randomly selected with uniform probabilities. There were ten data sessions, each with the traffic rate of four packets per second. The size of data payload was 512 bytes.

5.2.2Results Analysis

We begin by examining the effects of the pause time on the frequency of routing discovery of different route protocols relative to AODV. Figure 5.10 shows the result of frequency of routing discovery versus the pause time. The frequency of routing discovery for OHMR is the lowest compared with node-disjoint multipath and braided multipath since OHMR uses hybrid multipath to decrease route discovery frequency. This result is coincident with the previous theoretical analysis.

Figure 5.11 shows the results of average end-to-end delay. The average end-to-end delay is the average elapsed time to deliver a packet from the source node to the destination node. AODV has higher average end-to-end delay compared to multipath routing and the average end-to-end delay of OHMR is lower than those of node-disjoint and braided multipaths. This demonstrates that the multipath routing can improve the end-to-end delay. With the decrease of pause time, the average end-to-end delay for both multipath routing and single path routing increases. This is due to the fact that the network topology changes more frequently. However, the route discovery frequency for OHMR is smaller among other routing schemes. Therefore, OHMR has the lowest end-to-end delay among other schemes irrespective of various pause times.

The average packet delivery ratio is shown in Figure 5.12. The simulation results demonstrate that OHMR has higher packet delivery ratio than AODV and other multipath protocols. OHMR lost fewer packets than AODV, braided multipath and node-disjoint multipath. It is obvious that OHMR provide efficient fault tolerance and efficient recovery from failures resulting from node movement in MANETs.

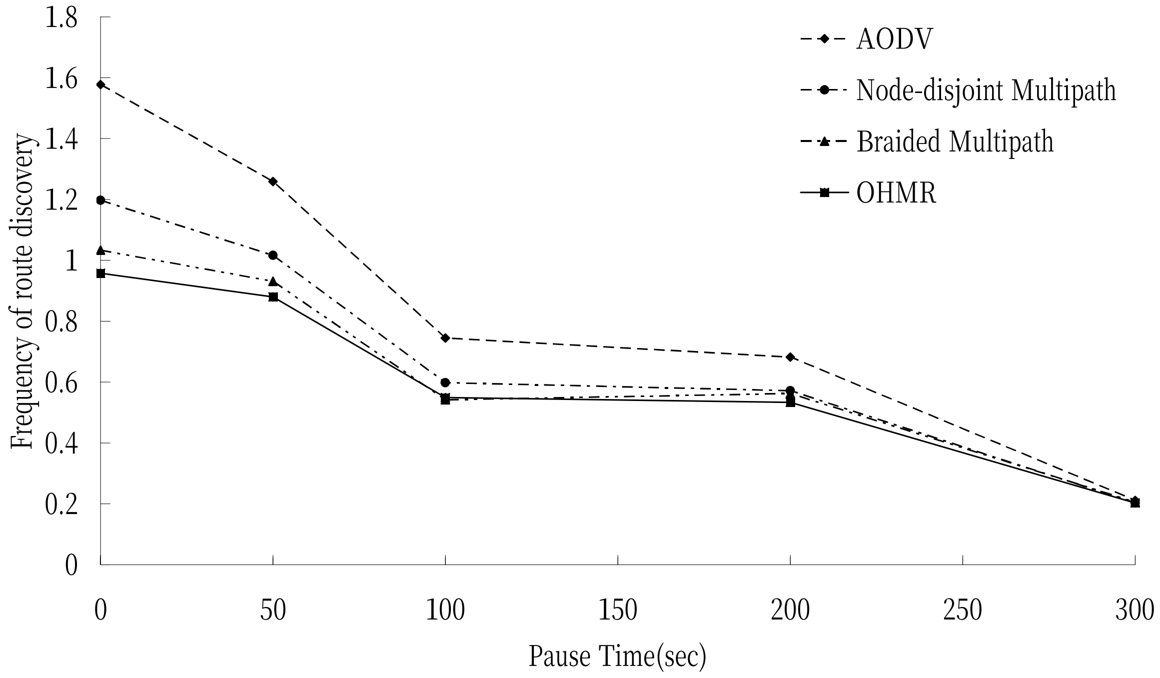


Figure 5.10: Route discovery frequency in simulations with CBR traffic.

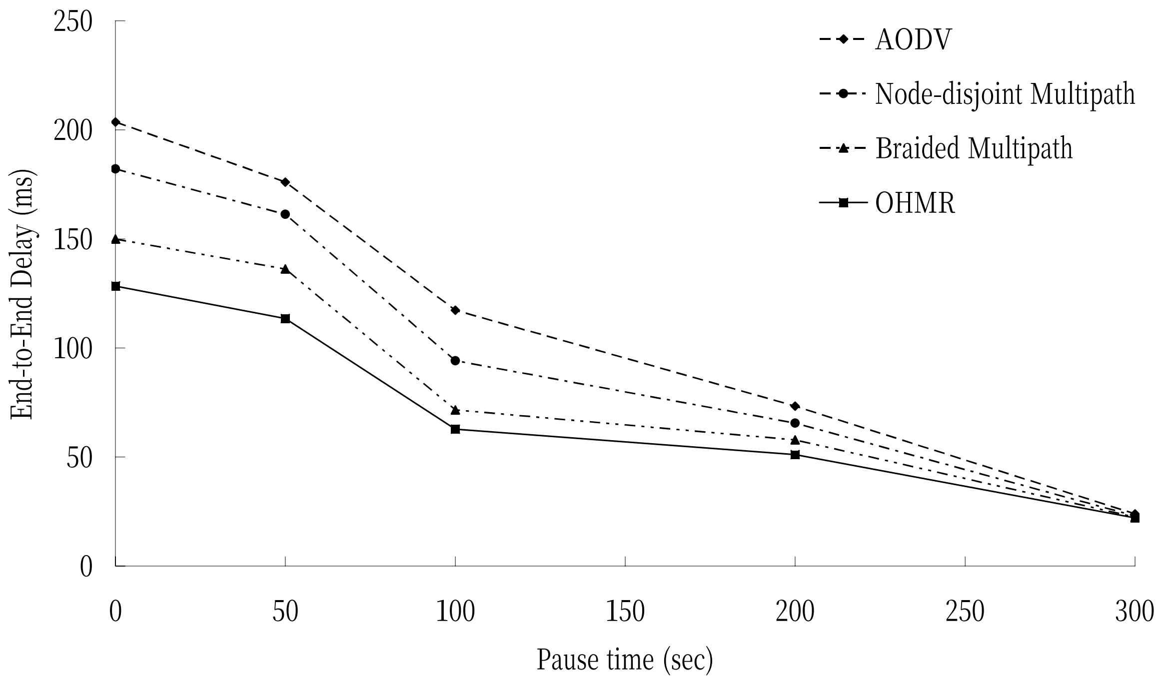


Figure 5.11: The average end-to-end delay in simulations with CBR traffic

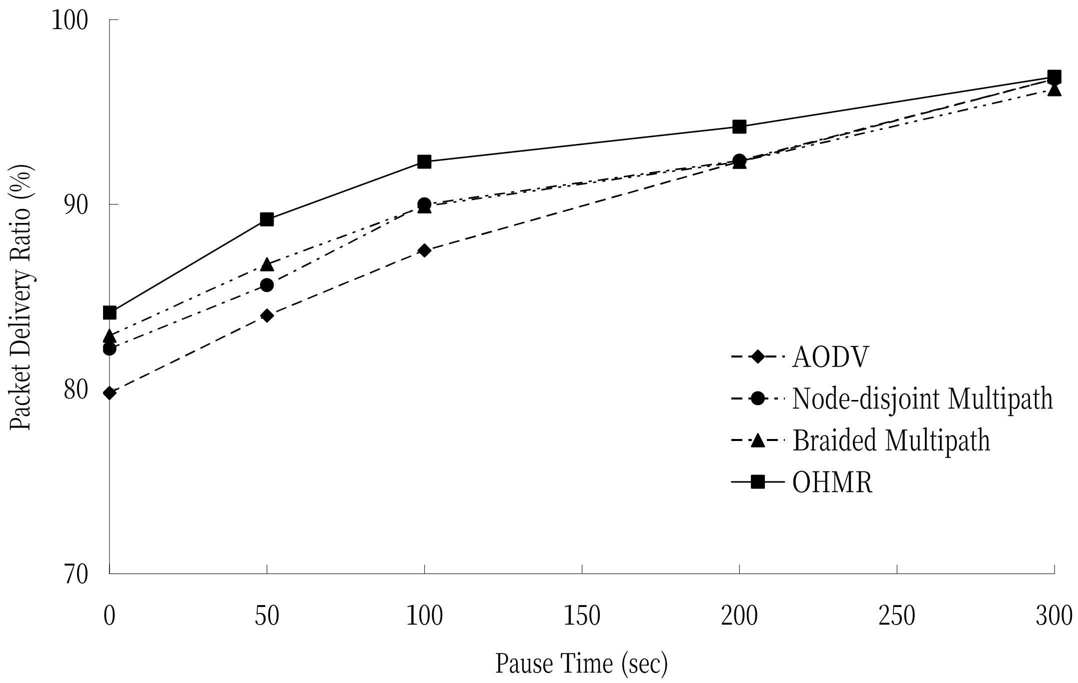


Figure 5.12: Packet delivery ratio in simulations with CBR traffic

In addition, our simulations set pause time to five because the mean lifetime of a wireless link (l) is assumed to be 5 in theoretical analysis and measure the time between successive discoveries in Table 5.1. These results of simulations compare to numerical results of theoretical analysis in Table 5.2. There are two similar aspects in simulations and theoretical analysis. One is that the time between successive discoveries decreases as the length of the primary path increases. The other is that the time between successive discoveries under OHMR is larger than those under braided multipath, node-disjoint multipath and AODV in turn. These results show that OHMR can maintain an end-to-end transmission for a long time. We also observe the time between successive discoveries in simulations is much greater than relative time in theoretical analysis. The average lifetime of a wireless link in simulations is more than five. Each node randomly selects a position every five second, and moves toward that location with a speed between the minimum and the maxi-mum speed. When nodes move and are in transmission range of each other, the lifetime of a wireless link is growing and more than five. The time between successive discoveries gains as the lifetime of a wireless link increases.

Table 5.1 The average time between successive discoveries in simulations

Length of the primary path / Routing Protocol	3	4	5	6	7	8
AODV	9.95	6.04	6.15	4.09	3.65	3.33
Node-Disjoint Multipath	10.76	7.53	6.2	5.87	5.31	4.14
Braided Multipath	15.18	10.82	8.45	6.75	6.03	5.23
OHMR	15.5	13.93	8.69	7.16	6.12	5.33

Table 5.2 The expected time between successive discoveries in theoretical analysis

Length of the primary path / Routing Protocol	3	4	5	6	7	8
AODV	1.67	1.25	1	0.83	0.71	0.63
Node-Disjoint Multipath	2.5	1.88	1.5	1.25	1.07	0.94
Braided Multipath	2.76	2.41	2.17	2.01	1.88	1.78
OHMR	3.09	2.61	2.31	2.11	1.96	1.84

5.3 Simulations with Multimedia Traffic

5.3.1Simulation Environment

We use a simulation model based on NS-2 with CMU wireless extension [12]. In the simulations, the MANET consists of sixteen mobile nodes are located inside a 600m × 600m region. We only consider the continuous mobility case. Each mobile node has a continuous and random waypoint mobility model [11] (0s pause time) with a maximum speed of 5 meter/second. The radio propagation model is the two-ray ground reflection model for longer distance with omnidirectional antenna. The shared radio media has a nominal bit rate of 2 Mbps. UDP is used as transport protocol. Ten UDP traffic flows are introduced as background traffics. Each of these flows has the traffic rate of four packets per second. The size of data payload was 512 bytes. The some, destination and the duration of these background flows are set random. Each of nodes has a queue size of 20 packets. These settings can be easily modified according to the requirements of applications.

We get the video file from the website [13]. There are two format sizes, CIF (352 x 288) and QCIF (176 x 144). The difference of them is video frame size. Here we use “Highway drive” video in CIF format to simulate. We decode thc CIF video using an MPEG codec, at 30 frames per second at various quantization levels and for different

Group of Pictures (GOP) lengths. Decoded video quality is measured in terms of the fraction of decodable frames and Peak-Signal-to-Noise-Ratio (PSNR).

The fraction of decodable frames: The fraction of decodable frames reports the number of decodable frames over the total number of transmitted frames. A frame is considered to be decodable if at least a fraction dt (decodable threshold) of the data in each frame is received. However, a frame is only considered decodable if and only if all of the frames upon which it depends are also decodable. Therefore, when dt=0.75, 25% of the data from a frame can be lost without causing that frame to be considered as undecodable. In the simulations, we set the decodable threshold to one (dt=1).

PSNR (Peak Signal Noise Ratio): PSNR is one of the most widespread objective metrics to assess the application-level QoS of video transmissions. The following equation shows the definition of the PSNR between the luminance component Y of source image S and destination image D:

$$PSNR(n)_{dB} = 20 \log_{10}\left(\frac{V_{peak}}{\sqrt{\frac{1}{N_{col}N_{row}}\sum_{i=0}^{N_{col}}\sum_{j=0}^{N_{row}}[Y_S(n,i,j)-Y_D(n,i,j)]^2}}\right),$$

where $V_{peak} = 2^k-1$ and k = number of bits per pixel (luminance component). PSNR measures the error between a reconstructed image and the original one. Prior to transmission, one may then compute a reference PSNR value sequence on the reconstruction of the encoded video as compared to the original raw video. After transmission, the PSNR is computed at the receiver for the reconstructed video of the possibly corrupted video sequence received. The individual PSNR values at the source or receiver do not mean much, but the difference between the quality of the encoded video at the source and the received one can be used as an objective QoS metric to assess the transmission impact on video quality at the application level.

5.3.2Results Analysis

In the following we present a comparison study of the OHMR with a node-disjoint multipath for video streaming in MANET. The video stream is segmented into two sub-streams based on the quality resolutions. The MPEG codec was used to generate two sub-streams. With both of the OHMR and the node-disjoint multipath, the two sub-streams are sent over two node-disjoint paths. One of sub-stream is labeled as high priority (I-frame and P-frame), and the other is labeled as low priority (B-frame). In the experimental results presented, the performance of OHMR is compared with the performance of the node-disjoint multipath under the same topology and background traffic environment.

Table 5.1 shows that the packet delivery ratio for OHMR has better performance than that of the node-disjoint multipath. The node-disjoint multipath drops a larger fraction of the packets than that of the OHMR. It can be seen that the OHMR has higher reliability than the node-disjoint multipath. We observe that the peak signal-to-noise ratio (PSNR) drops when there is loss in packets. The deepest drop occurs when a large burst of losses in I-frame and P-frame with a loss burst of B-frame. The braided alternate paths in the OHMR can avoid communication failures and a loss burst of B-frame when the primary path breaks. The PSNR curve in Fig. 5.13 has more frequent and a larger burst of frame loss than that in Fig. 5.14. Compared to the fraction of decodable frames and the average PSNR in Table 5.2, OHMR improves the performance by up to 16.89%, and OHMR achieves a significant 1.05 dB gain over the node-disjoint multipath in this experiment.

Table 5.3 Comparison of the Packet delivery ratio between the OHMR and the node-disjoint multipath

	OHMR	Node-disjoint multipath
Packets sent	4607	
Packets received	4558	3940
Packets lost	49	667
Packet delivery ratio	98.9%	85.5%

Table 5.4 Comparison of the fraction of decodable frames and the average PSNR between the OHMR and the node-disjoint multipath

	OHMR	Node-disjoint multipath
Frames sent	2001	
Frames received	1976	1638
Not decoded frames	20	359
Frames miss	4	3
The fraction of decodable frames	98.75%	81.86%
Average PSNR	36.81	35.76

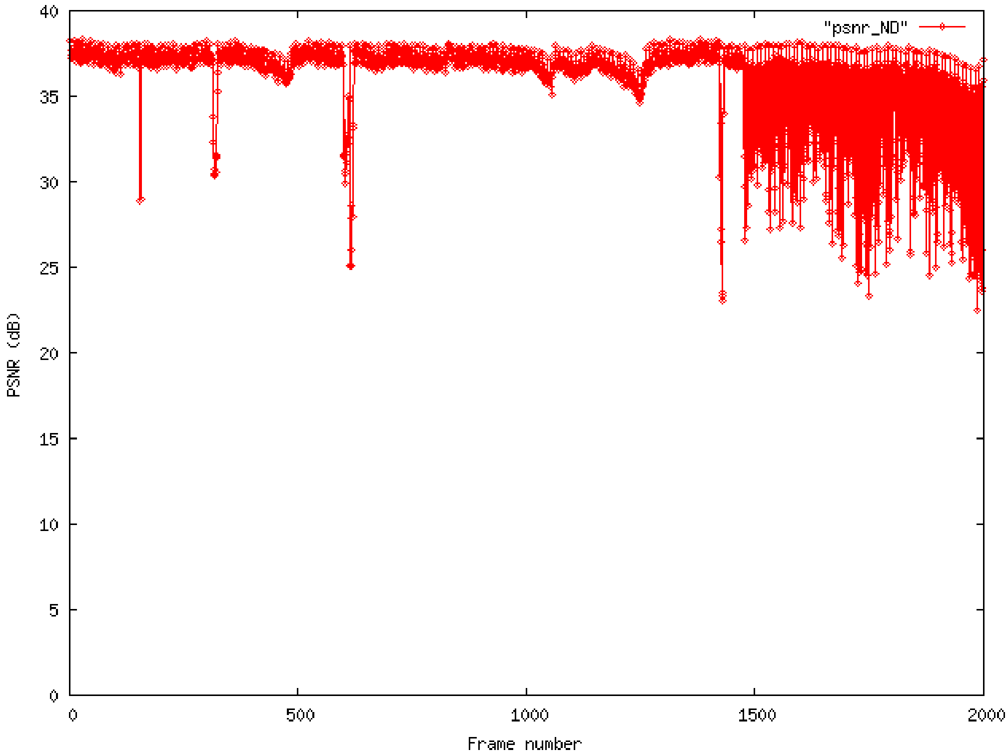


Figure 5.13 The PSNRs of the received video frames in the node-disjoint multipath.

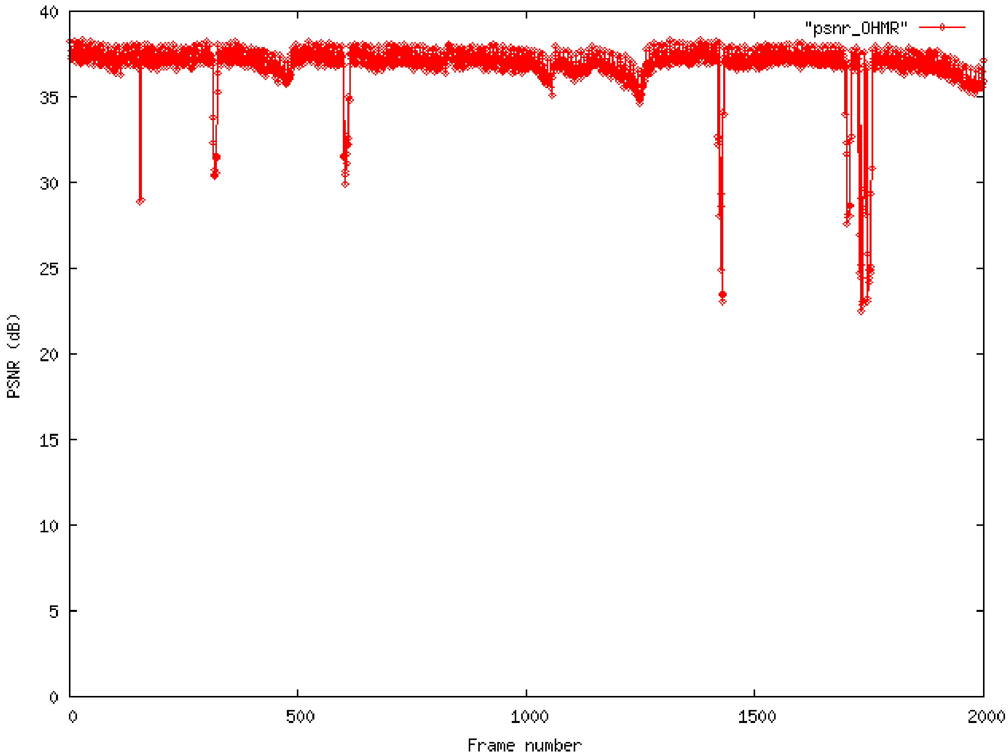


Figure 5.14 The PSNR of the received video frames in the OHMR.

Chapter 6

Conclusions

This study has proposed a multipath extension to the AODV on-demand routing protocol. The OHMR searches for the node-disjoint multipath and the braided multipath using a single flooding query in order to provide sufficient redundancy. The energy consumed by alternate paths of the braided multipath is comparable to that consumed by the primary path. Alternate paths of the node-disjoint multipath are unaffected by link/node failures on the primary path. The key advantage of OHMR is a significant reduction in the frequency of route discovery flooding. The theoretical analysis has shown that the OHMR maintains an end-to-end transmission for a longer period than single path, braided multipath, intermediate node-disjoint multipath and node-disjoint multipath. Simulation results show that the OHMR can reduce the frequency of route discovery, decrease the average end-to-end delay, and increase packets delivery ratio. We then extend OHMR with a multimedia traffic allocation strategy to classify multimedia sub-streams among multiple paths according to different priority levels. The strategy is to allow more important sub-streams to travel over the primary path, and less important sub-streams to travel over the alternate node-disjoint path. Our experiments show that the proposed protocol for multimedia communication can improve the performance of the fraction of decodable frames and achieve better performance in terms of video quality over the node-disjoint multipath. In a future study, we hope to add the queue management for different sub-streams in our proposed protocol and investigate related performances for other multimedia codecs.

Bibliography

[1] E. Perkings, E. M. Belding-Royer, S. R. Das, “Ad Hoc On-Demand Distance Vector (AODV) Routing,” http://www.ietf.org/rfc/rfc3561.txt, IETF RFC, July 2003.

[2] J. Broch, D. Johnson, and D. Maltz, “The Dynamic Source Routing Protocol for Mobile Ad hoc Networks (DSR),” http://www.ietf.org/internet-drafts/draft-ietf-manet-dsr-10.txt, IETF Internet draft, July 2004.

[3] E. Perkings, E. M. Royer and S. R. Das, “Performance Comparison of Two On-Demand Routing Protocols for Ad Hoc Networks,” IEEE Personal Communications, pp. 16–28, February 2001.

[4] S. J. Lee and M. Gerla, “Split Multipath Routing with Maximally Disjoint Paths in Ad Hoc Networks,” IEEE ICC, pp. 3201-3205, June 2001.

[5] A. Nasipuri, R. Castaneda, and S. R. Das, “Performance of Multipath Routing for On-demand protocols in Mobile Ad Hoc Networks,” ACM Mobile Networks and Applications (MONET), pp. 339-349, August 2001.

[6] M. K. Marina, and S. R. Das, “On-demand Multipath Distance Vector Routing in Ad Hoc Networks,” IEEE International Conference on Network Protocols (ICNP), pp. 14-23, November 2001.

[7] X. Li and L. Cuthbert, “On-demand node-disjoint multipath routing in wireless ad hoc networks,” IEEE International Conference on Local Computer Networks, pp. 419 – 420, November 2004.

[8] B. Ganesan, R. Govindan, S. Shenker, and D. Estrin, “Highly-Resilient, Energy-Efficient Multipath Routing in Wireless Sensor Networks,” Mobile Computing and Communications Review (MC2R), Vol. 4, no. 5, October 2001.

[9] V. D. Park and M. S. Corson, “A highly adaptive distributed routing algorithm for mobile wireless networks,” IEEE INFOCOM, pp. 1405–1413, 7-11 April 1997.

[10] S. R. Das, R. Castañeda, J. Yan and R. Sengupta, “Comparative performance evaluation of routing protocols for mobile, ad hoc networks,” IC3N, pp. 153-161, 12-15 Oct. 1998.

[11] J. Broch, D. A. Maltz, D. B. Johnson, Y.C. Hu, and J. Jetcheva, “A performance comparison of multi-hop wireless ad hoc network routing protocols,” In Proc. of the 4th annual ACM/IEEE international conference on Mobile computing and networking, pp.85-97, 1998.

[12] ns-2: network simulator. http://www.isi.eduinsnam/ns/

[13] http://trace.eas.asu.edu/yuv/index.html

[14] J. Chakareski, S. Han, and B. Girod, “Layered coding vs. multiple descriptions for video streaming over multiple paths,” In Proc. of the 11th ACM international conference on Multimedia, pp. 422-431, Berkeley, CA, Nov. 2003.

[15] X. Li and L. Cuthbert, “Stable Node-Disjoint Multipath Routing with Low Overhead in Mobile Ad Hoc Networks,” In Proc. of the IEEE MASCOTS, 4-8 Oct. 2004.

[16] Z. Ye, S. V. Krishnamurthy and S.K. Tripathi, “A Framework for Reliable Routing in Mobile Ad Hoc Networks,” IEEE INFOCOM, Vol. 1, 30 March-3 April, 2003.

[17] Linifang Zhang, Zenghua Zhao, Yantai Shu, Lei Wang, Yang and O.W.W., ”Load balancing of multipath source routing in ad hoc networks,” IEEE ICC, Vol. 5, pp. 3197-3201, 28 April-2 May 2002.

[18] N. Gogate, D. Chung, S. S. Panwar, and Y. Wang, “Supporting image and video applications in a multihop radio environment using path diversity and multiple description coding,” IEEE Trans. Circuits Syst. Video Technol., vol. 12, no. 9, pp. 777–792, Sep. 2002.

[19] S. Mao, S. Lin, S. S. Panwar, Y. Wang, and E. Celebi, “Video transport over ad hoc networks: Multistream coding with multipath transport,” IEEE JSAC, vol. 21, no. 10, pp. 1721–1737, Dec. 2003.

[20] J. Apxtolopulos and S. Wee, “Unbalanced multiple description video communication using path diversity,” In Proc. IEEE ICIP, pp. 966-969, Thessaloniki, Greece, Oct. 2001.

[21] J. Apostolopoulos, T. Wong, W. Tan, and S. Wee, “On multiple description streaming with content delivery networks,” In Proc. IEEE INFOCOM, pp. 1736-1745, New York, June 2002.