東海大学大学院平成19年度博士論文

無線アドホックネットワーク上の画像転送におけるパスおよびサーバ分散方式に関する研究

Path and Server Diversities for Video Transmission over Mobile Ad Hoc Networks

指導 石井 啓之 教授

東海大学大学院理工学研究科
総合理工学専攻

周 志 安

CHOW CHEE ONN
ABSTRACT

As mobile ad hoc networks emerge as a promising solution for future ubiquitous communications, there is a compelling demand to design a reliable method for supporting good quality video considering the increasing importance of video applications over the cyberspace nowadays. As commonly known, video application is usually associated with rigid requirements on bandwidth, lost, delay and jitter. Moreover, it is a great challenge to support video in ad hoc networks due to the frequent link breakage and the lack of centralized management system. In this dissertation, we tackle the problem of video unicast and multicast over mobile ad hoc networks using the concept of path and server diversities. The potential of this concept lies on the mesh topology of a mobile ad hoc network that allows each node to be virtually connected to any other node as long as a valid single hop or multiple hops connection can be established between them.

To tackle the issue of video unicast in mobile ad hoc networks, we propose a framework that combines Multipoint-to-Point (MP2P) communication and Multiple Description Coding (MDC) scheme to support best-effort real-time video transmission in mobile ad hoc networks. In this framework, the video is encoded into several independent video descriptions using the MDC scheme, and each video sender is instructed to send a discrete video description. For this purpose, a receiver-initiated source searching mechanism with synchronization feature is proposed to locate and to select the most suitable video senders. In order to obtain the optimum performance, the routing protocol is extended to encourage disjointedness in video transmission from different video senders. The extended Multipoint-to-Point Dynamic Source Routing (MP2P-DSR) protocol consists of two mechanisms: preventive and corrective mechanisms. The former encourages node disjointedness during route discovery and the latter supports link disjointedness when node disjointedness is failed to be achieved. In summary, the proposed MP2P framework supports both path and server diversities to improve the fault-tolerance of the video transmission system. Simulation shows that a great improvement in the video quality is achieved over the conventional point-to-point transmission with a small additional cost on the overall control overhead.

One possible solution to support video multicast over mobile ad hoc networks is by using the multiple tree video multicast model. This method transmits the discrete video descriptions along different multicast trees to provide better fault-tolerance. In this dissertation, we further accomplish this model by proposing a tree-based multiple tree multicast routing protocol called Multiple Tree Multicast Ad Hoc On-demand Distance Vector (MT-MAODV). This protocol constructs two highly disjoint multicast trees in a single routine to minimize the additional control overhead introduced. From the simulation results, using the multiple tree multicast model with the proposed routing protocol gives a better video quality than the conventional single tree video multicast framework does. As expected, this improvement is associated with a small increase in the control overhead.
CONTENTS

Abstract ii
Contents iii
Abbreviations v

Chapter 1: Introduction 1
1.1 Overview of Mobile Ad Hoc Networks 1
1.2 Video Transport over Mobile Ad Hoc Networks 4
   1.2.1 Challenges of Video Transport over Mobile Ad Hoc Networks 4
   1.2.2 Multiple Description Coding Scheme 5
   1.2.3 Video Unicast over Mobile Ad Hoc Networks 7
   1.2.4 Video Multicast over Mobile Ad Hoc Networks 9
1.3 Objectives of the Dissertation 10
1.4 Dissertation Outline 11
References 12

Chapter 2: Multipoint-to-Point Video Transmission 15
2.1 Motivation 15
2.2 Multipoint-to-Point Video Transmission Framework 16
   2.2.1 Potential Applications 16
   2.2.2 Novel Architecture 18
   2.2.3 Source Searching Stage 19
   2.2.4 Route Discovery Stage 22
   2.2.5 Video Streaming Stage 24
2.3 Simulation Study Using NS-2 25
   2.3.1 Simulation Settings 25
   2.3.2 Results and Discussion 26
2.4 Performance Evaluation using Ad Hoc Network Testbed 30
   2.4.1 Setup of the Testbed 30
   2.4.2 Results and Discussion 32
2.5 Conclusions 34
References 35

Chapter 3: Multipoint-to-Point Dynamic Source Routing Protocol 37
3.1 Motivation 37
3.2 The MP2P-DSR Protocol 38
   3.2.1 Preventive Mechanism 39
   3.2.2 Corrective Mechanism 41
3.3 Performance Evaluation 43
   3.3.1 Simulation Settings 43
   3.3.2 Performance Parameters 45
   3.3.3 Results and Discussion 46
3.4 Conclusions 53
References 54
Chapter 4: Multiple Tree Video Multicast

4.1 Motivation

4.2 Multicast Ad Hoc On-Demand Distance-Vector (MAODV) Protocol

4.3 Multiple Tree Multicast Ad Hoc On-demand Distance Vector Protocol
   4.3.1 The Core of MT-MAODV
       i. Multiple Tree Construction
       ii. Multiple Tree Maintenance
       iii. Tree Optimization
   4.3.2 Video Multicast with MT-MAODV

4.4 Performance Evaluation
   4.4.1 Simulation Settings
   4.4.2 Performance Parameters
   4.4.3 Results and Discussion

4.5 Conclusions

References

Chapter 5: Conclusions and Future Work

5.1 Conclusions

5.2 Future Work

List of Publications

Acknowledgments
## ABBREVIATIONS

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMRIS</td>
<td>Ad Hoc Multicast Routing protocol utilizing increasing Id-numbers</td>
</tr>
<tr>
<td>AND</td>
<td>Average Node Degree</td>
</tr>
<tr>
<td>AODV</td>
<td>Ad Hoc On-demand Distance Vector routing</td>
</tr>
<tr>
<td>CAMP</td>
<td>Core-Assisted Mesh Protocol</td>
</tr>
<tr>
<td>CSMA/CA</td>
<td>Carrier Sense Multiple Access/Collision Avoidance</td>
</tr>
<tr>
<td>D-PRMA</td>
<td>Distributed Packet Reservation Multiple Access</td>
</tr>
<tr>
<td>DLSP</td>
<td>Distributed Laxity-based Priority Scheduling</td>
</tr>
<tr>
<td>DSR</td>
<td>Dynamic Source Routing</td>
</tr>
<tr>
<td>DWOP</td>
<td>Distributed Wireless Ordering Protocol</td>
</tr>
<tr>
<td>FAMA</td>
<td>Floor Acquisition Multiple Access</td>
</tr>
<tr>
<td>FORP</td>
<td>Flow-Oriented Routing Protocol</td>
</tr>
<tr>
<td>GOP</td>
<td>Group of Pictures</td>
</tr>
<tr>
<td>GL</td>
<td>Group Leader</td>
</tr>
<tr>
<td>GL table</td>
<td>Group Leader table</td>
</tr>
<tr>
<td>GRPH</td>
<td>Group Hello</td>
</tr>
<tr>
<td>IPTV</td>
<td>Internet Protocol Television</td>
</tr>
<tr>
<td>ITAMAR</td>
<td>Independent-Tree Ad Hoc Multicast Routing</td>
</tr>
<tr>
<td>LAN</td>
<td>Local Area Network</td>
</tr>
<tr>
<td>LSP</td>
<td>Length of Longest Shortest-path</td>
</tr>
<tr>
<td>MAC</td>
<td>Medium Access Control</td>
</tr>
<tr>
<td>MACA/PR</td>
<td>Multiple Access Collision Avoidance with Piggy-backed Reservation</td>
</tr>
<tr>
<td>MACAW</td>
<td>Multiple Access Collision Avoidance for Wireless LANs</td>
</tr>
<tr>
<td>MACT</td>
<td>Multicast Activation</td>
</tr>
<tr>
<td>MANETs</td>
<td>Mobile Ad Hoc Networks</td>
</tr>
<tr>
<td>MAODV</td>
<td>Multicast Ad Hoc On-demand Distance Vector</td>
</tr>
<tr>
<td>MBWA</td>
<td>Mobile Broadband Wireless Access</td>
</tr>
<tr>
<td>MDC</td>
<td>Multiple Description Coding</td>
</tr>
<tr>
<td>MIMO</td>
<td>Multiple-Input Multiple-Output</td>
</tr>
<tr>
<td>MOS</td>
<td>Mean Opinion Score</td>
</tr>
<tr>
<td>MP2MP</td>
<td>Multipoint-to-Multipoint</td>
</tr>
<tr>
<td>MP2P</td>
<td>Multipoint-to-Point</td>
</tr>
<tr>
<td>MP2P-DSR</td>
<td>Multipoint-to-Point Dynamic Source Routing</td>
</tr>
<tr>
<td>MR table</td>
<td>Multicast Routing table</td>
</tr>
<tr>
<td>MT-MAODV</td>
<td>Multiple Tree Multicast Ad Hoc On-Demand Distance Vector</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>ODMRP</td>
<td>On-Demand Multicast Routing Protocol</td>
</tr>
<tr>
<td>OFDM</td>
<td>Orthogonal Frequency Division Multiplexing</td>
</tr>
<tr>
<td>OLSR</td>
<td>Optimized Link State Routing</td>
</tr>
<tr>
<td>P2P</td>
<td>Point-to-Point</td>
</tr>
<tr>
<td>P2MP</td>
<td>Point-to-Multipoint</td>
</tr>
<tr>
<td>PSNR</td>
<td>Peak Signal-to-Noise Ratio</td>
</tr>
<tr>
<td>qcif</td>
<td>Quarter Common Intermediate Format</td>
</tr>
<tr>
<td>RERR</td>
<td>Route Error</td>
</tr>
<tr>
<td>RLC</td>
<td>Rate of Link Changes</td>
</tr>
<tr>
<td>RREP</td>
<td>Route Reply</td>
</tr>
<tr>
<td>RREQ</td>
<td>Route Request</td>
</tr>
<tr>
<td>SDC</td>
<td>Single Description Coding</td>
</tr>
<tr>
<td>SINR</td>
<td>Signal to Interference plus Noise Ratio</td>
</tr>
<tr>
<td>SnR</td>
<td>Shared-node Ratio</td>
</tr>
<tr>
<td>TORA</td>
<td>Temporary Ordered Routing Algorithm</td>
</tr>
<tr>
<td>UDP</td>
<td>User Datagram Protocol</td>
</tr>
<tr>
<td>UR table</td>
<td>Unicast Routing table</td>
</tr>
<tr>
<td>UWB</td>
<td>Ultra-Wideband</td>
</tr>
<tr>
<td>VREP</td>
<td>Video Reply</td>
</tr>
<tr>
<td>VREQ</td>
<td>Video Request</td>
</tr>
<tr>
<td>VSEL</td>
<td>Video Select</td>
</tr>
<tr>
<td>WiMax</td>
<td>Worldwide Interoperability for Microwave Access</td>
</tr>
<tr>
<td>Wi-Fi</td>
<td>Wireless Fidelity</td>
</tr>
</tbody>
</table>
CHAPTER 1
INTRODUCTION

1.1 Overview of Mobile Ad Hoc Networks

A mobile ad hoc network is formed when a group of wireless devices organize themselves in a mesh topology for packet relaying without the support of a pre-installed infrastructure. The principle behind ad hoc networking is to utilize multi-hop radio relaying for transmitting data packets from a node to any other node within the same network, provided a valid route can be established between them. The absence of a central coordinator to monitor the changes in the network topology makes the routing issues more complicated than the conventional infrastructure-dependent cellular networks [1]. Figure 1.1 illustrates two types of wireless networks. Conventional cellular network is given in Figure 1.1(a). For a connection to be established between two mobile nodes, one of the mobile nodes must send a request to the mobile switching center via the base station so that the mobile switching center can initiate the connection establishment request. If a mobile node is located outside of the coverage of any base station, no connection can be established between this node with any other node, even if they are located at close proximity to each other. For example, node X is beyond the coverage area and although it is physically very close to node A, no connection can be established between them. Figure 1.1(b) shows the equivalent topology for a mobile ad hoc network. A connection between any two nodes can be established through some intermediate nodes. For example, node X can communicate with node D via nodes A, B and C.

The initial deployment of ad hoc networks is expected to take place in critical environments such as during disaster, in search and rescue, and on battlefields, where a preinstalled infrastructure is not readily available. Recently, mobile ad hoc networks have emerged as a promising solution for providing untethered, ubiquitous service in future wireless system due to the increase in the computing power and decrease in the production cost of mobile devices [2-4]. Moreover, the development of new technologies has made available higher bandwidth for wireless channel. For example, the deployment of Orthogonal Frequency Division Multiplexing (OFDM) has increased the data rate of IEEE 802.11a/g standard to 54Mbps [5]. Meanwhile, the deployment of Multiple-Input Multiple-Output (MIMO) increases the theoretical data rate to 540Mbps in the latest 802.11n standard [6].
Chapter 1. Introduction   Section 1.1

Figure 1.1 Wireless networks:
(a) A cellular network; (b) An ad hoc network.

The deployment of mobile ad hoc networks is associated with some major issues and challenges due to the nature of this type of networks, among which are discussed briefly below.

- **Medium Access Control (MAC) Protocol** – the MAC protocol should be fully distributed considering that there is no centralized coordination is available. Besides, the issues of hidden and exposed terminals should also be taken into consideration when designing the MAC protocol. Generally, the currently available MAC protocols for ad hoc networks can be divided into three types:
  (i) Contention-based protocols: Multiple Access Collision Avoidance for Wireless LANs (MACAW) [7] and Floor Acquisition Multiple Access (FAMA) protocol [8].
  (ii) Contention-based protocols with reservation mechanisms: Distributed Packet Reservation Multiple Access (D-PRMA) protocol [9] and MACA with Piggy-backed Reservation (MACA/PR) [10].
  (iii) Contention-based protocols with scheduling mechanisms: Distributed Wireless Ordering Protocol (DWOP) [11] and Distributed Laxity-based Priority Scheduling scheme (DLPS) [12].

- **Security** – the lack of central coordination and the nature of shared wireless medium have made the system more vulnerable to attack. With this in mind, a distributed authentication system is needed. This issue is still an open and active research topic.
• **Energy Management** – mobile devices usually do not have access to the power supply at all time and so a well-designed energy management scheme is essential in order to maximize the battery lifetime. Considering that every mobile device contributes to the establishment of the mesh topology, prolonging the battery lifetime is equivalent to lengthening the lifetime of the whole network. Consequently, this issue is essential to maintain the functionality of the network.

• **Unicast Routing** – the routing protocol is another active research topic considering the unique characteristics of ad hoc networks. Conventional proactive approach is not suitable for mobile ad hoc networks because continuous changes in network topology may create excessive routing overhead. In order to overcome this weakness, reactive routing protocol is proposed. Protocols in this category are Ad Hoc On-Demand Distance Vector (AODV) routing protocol [13], Dynamic Source Routing (DSR) protocol [14], Temporary Ordered Routing Algorithm (TORA) protocol [15], and Flow-Oriented Routing Protocol (FORP) protocol [16], just to name a few.

• **Multicast Routing** – multicast over wireless networks has an advantage over wire-line networks due to the innate broadcast nature of the wireless networks. As a result, multicast over wireless networks does not involve the placement of multicast node to duplicate the data packets. However, multicast over mobile ad hoc networks is still a challenging task considering the rapid changes in the network topology that requires the multicast tree to be repaired frequently. Some commonly known multicast routing protocols for mobile ad hoc networks are On-Demand Multicast Routing Protocol (ODMRP) [17], Multicast Ad Hoc On-Demand Distance Vector (MAODV) routing protocol [18], Independent-Tree Ad Hoc Multicast Routing (ITAMAR) protocol [19], Ad Hoc Multicast Routing protocol utilizing increasing Id-numbers (AMRIS) [20], and Core-Assisted Mesh Protocol (CAMP) [21].

In this dissertation, the Dynamic Source Routing (DSR) protocol and the Multicast Ad Hoc On-demand Distance Vector (MAODV) routing protocol are used for video unicast and multicast, respectively, because these protocols are suitable to be used. Most importantly, both protocols are well-developed and proven to give good performance under a variety of network conditions.
1.2 Video Transport over Mobile Ad Hoc Networks

1.2.1 Challenges of Video Transport over Mobile Ad Hoc Networks

Nowadays, video applications have contributed to a great portion of the networking traffic over the Internet and the cellular system. This implies the increasing importance of video applications in the cyberspace and providing good quality video over various types of networks is crucial. Recently, supporting video applications over wireless networks becomes a topic of intense interest due to the advance in wireless technologies that increase greatly the bandwidth of the wireless channel and the computation power of mobile devices. These factors have created a platform in the wireless domain similar to the wire-line networks to support various types of video applications, such as video-on-demand, video teleconferencing, and download-and-play video.

Video application is itself very sensitive due to the rigid requirements on bandwidth, loss, delay and jitter. Packet losses cause significant impairment in the video quality due to the propagation of error to the successive frames within the same group of picture (GOP) that share the same prediction. Besides, a video packet arrives after its deadline becomes useless to the decoder, and so re-transmission is generally not helpful in real-time video transmission. Moreover, the inter-arrival time between consecutive video frames also plays an important role in ensuring smooth signal because a long jitter between successive frames may cause jerkiness during the video viewing.

As compared to wire-line networks, video transmission over wireless networks faces some additional problems. First, the wireless medium is more prone to transmission loss due to fading, interference, shadowing and attenuation, and the resultant error rate is usually higher. Besides, proper energy management scheme is also needed because mobile devices may not be able to access to the power supply at any time. Furthermore, if the wireless networks are not supported by a preinstalled infrastructure, for example in mobile ad hoc networks, great challenges is experienced to maintain the network connectivity due to node mobility, which causes continuous changes in network topology. Moreover, transmission loss mentioned previously is also higher in ad hoc networks than in conventional single-hop wireless networks because an end-to-end path in ad hoc networks is usually composed of several wireless connections.

In light of this, video transmission over mobile ad hoc networks can be tackled from two perspectives. First, the video coding scheme can be further improved to increase its error resilience, for example by using layered coding [22] or Multiple Description Coding (MDC) scheme [23, 24]. Second, a more reliable packet transmission technique is needed to guarantee at least minimum quality of services at all time, for example, the use of path diversity [25-31] and server diversity [32, 33].
1.2.2 Multiple Description Coding Scheme

The main objective of video coding is to compress the raw video into a form suitable and convenient for storage and transmission without significantly reducing its quality. Generally, a video encoder consists of three main functional modules: a temporal model that reduces temporal redundancy between neighboring video frames, a spatial model that reduces spatial redundancy between neighboring samples in the residual frame of the temporal model, and an entropy encoder that converts a series of symbols representing elements of the video sequence into a compressed bit stream. A major problem of this coding scheme is the propagation of error where a single frame error may affect not only the current frame but also all subsequent frames, until a frame with new prediction is received. In order to combat this problem, layered coding is introduced to encode the video stream into two layers: a base layer and an enhancement layer. The base layer alone is enough to reconstruct the original video but at lower quality. The enhancement layer is used to improve the video quality further. In normal circumstance, both layers are sent, but under a heavy traffic load, only the base layer is sent [22]. This method is more robust but the same problem may occur if the base layer is lost, since the enhancement layer is totally useless by itself. Consequently, with the Multiple Description Coding (MDC) scheme it is proposed to generate several equally important and independent video descriptions. Each description can be decoded independently to obtain the original video at low but acceptable quality, and every additional description received improves the quality further [23]. MDC is more error-resilient and it has been proven to outperform conventional coding schemes in most lossy networks.

A few MDC schemes have been proposed to date. Figure 1.2 shows a general architecture of one of the simplest forms, the frame-based approach [33]. The encoder consists of two parts: a frame separator and a video encoder, as shown in Figure 1.2(a). The raw video frames are divided into several sub-streams in a per frame basis. Next, these sub-streams are encoded separately to obtain independent video descriptions using a standard video codec such as MPEG and H.264. As compared to a single video stream, a higher bandwidth is required to host the MD coded video due to the larger difference between neighboring frames within each sub-stream. This results in lower compression efficiency and the resulting data size is usually larger. In [34], a comprehensive analysis of video encoded by the MDC scheme shows that the average frame size increases as the number of descriptions increases. The size of this increase depends on the content of the video as well; normally, high motion video produces a larger increase as the number of descriptions increases. The decoding process is illustrated in Figure 1.2(b). Each description is decoded separately and the reconstructed video sub-streams are merged for viewing purpose. At this stage, a compensation scheme may be used to replace any missing frame with the one displayed previously. This compensation scheme can reduce distortion during display of the video, as given in Figure 1.2(c).
There are also specifically designed MD codec, which involves a modification of the video coding algorithm [24, 25]. For example, in [24] a second-order predictor for motion compensation was used to predict a frame from two previously coded frames; while in [25] a matching pursuits MDC scheme was created by using a three-prediction-loop scheme to replace the discrete-cosine transfer structure in the matching pursuits video codec framework. Literature [23] presents a comprehensive analysis on MDC.

![Diagram](image)

**Figure 1.2** General architecture of Multiple Description codec: (a) Encoder; (b) Decoder; (c) Frame compensation.
1.2.3 Video Unicast over Mobile Ad Hoc Networks

A considerable amount of work has been done in adopting the concept of diversity for video transmission over mobile ad hoc networks. Diversity can be achieved through multiflow transport, for example, through path diversity or server diversity. The concept of path diversity, commonly known as multipath transport, involves the splitting of the video into several parts and transmitting them via different paths [25-31].

Congestion-optimized multipath routing is proposed to minimize global congestion by finding multiple routes and performing traffic partitioning [36]. Besides, Zhu and Girod also proposed a distributed algorithm for congestion-minimized multipath routing in [31] and a congestion-aware rate allocation for multipath video streaming in [30]. However, the above mentioned work used a traffic model that assumes all nodes transmit simultaneously and the capacity of link is measured using signal to interference plus noise ratio (SINR), which is different with the flow model used in 802.11 MAC protocol. Consequently, they may not suite the real situation in mobile ad hoc networks.

In [26], Mao et al. proposed three point-to-point video transport techniques that combined multistream video coding together with multipath transport. These techniques are Feedback Based Reference Picture Selection Scheme (RPS), Layered Coding (LC) with selective Automatic Repeat Request (ARQ) scheme, and Multiple Description Motion Compensation Coding (MDMC) scheme. Their simulation demonstrated that the key factor for successful multipath transport is the use of disjoint routes to ensure that the losses experienced by each video stream are independent. For this purpose, they proposed an extended Dynamic Source Routing (DSR) protocol in order to ensure that two disjoint routes are maintained continuously between the video source and receiver. In order to minimize packet loss due to topology changes, Wei and Zakhor proposed the Robust Multipath Source Routing Protocol (RMPSR), which built a set of disjoint routes for a source-receiver pair instead of just maintaining two disjoint routes only [29]. In summary, both [26] and [29] have demonstrated the implementation of path diversity together with multistream coding scheme in improving the video quality.

Supporting multiflow transport not only involves issue at the network layer, but also at the higher layers, especially the application and transport layers. The existing real-time transport protocol (RTP) may not be able to support this framework efficiently due to the rapid topology changes and lack of reliable real-time connections in a mobile ad hoc network. In order to tackle this problem, Mao et al. proposed a multiflow real-time transport protocol (MRTP), which is an extension of RTP, to facilitate multiflow real-time video transmission over ad hoc networks [28]. MRTP is implemented at the application layer to provide essential support including session and flow management, data partitioning and reassembly, traffic dispersion, data framing, timing and quality-of-service feedback. The
authors suggested three application scenarios of this protocol, including point-to-point multipath transmission, multipoint-to-point communication and multicast.

Apart from path diversity, the concept of server diversity, more commonly known as multipoint-to-point communication, is also a feasible solution for improving video quality. In [32] and [33], server diversity has been employed to improve video quality over packet-lossy wire-line networks. However, the implementation of server diversity to achieve good quality video streaming over mobile ad hoc networks has not been studied yet. Figure 1.3 reveals the concept of diversities.

![Figure 1.3 Concepts of diversities: (a) Path diversity; (b) Server diversity.](image-url)
1.2.4 Video Multicast over Mobile Ad Hoc Networks

The introduction of the MDC scheme has spurred research interest in the area of video transport with path diversity. The concept of multicasting MD video was first introduced in CoopNet to prevent a web server from being overwhelmed by a large amount of requests from its clients [37]. In CoopNet, the live or on-demand video is encoded into several video descriptions and each description is sent along different trees. CoopNet uses a centralized tree management scheme and a logical link level routing approach that models each logical link as several physical links. Thus, it is not suitable for the infrastructureless mobile ad hoc networks because wireless medium cannot be modeled in such a way.

The concept of CoopNet was modified to tackle video multicast over mobile ad hoc networks in [38], where a multiple tree video multicast scheme was introduced. The authors modeled video multicast over mobile ad hoc network as an optimization problem and proposed a solution based on genetic algorithm to construct a pair of disjoint trees. Simulation study shows that a significant gain in the overall video quality can be achieved. However, the implementation of this proposal is rather difficult due to two reasons. First, it is assumed that the characteristics of all valid links are known. Second, the genetic algorithm involves a complicated computation. Generally, this paper introduced the concept of multiple tree video multicast but a proper routing protocol to construct multiple trees was not given.

Two multiple trees multicast routing protocols were presented in [39]. Serial Multiple Disjoint Tree Multicast Routing (Serial MDTMR) protocol constructs multiple disjoint trees in a sequential manner. This protocol creates completely-disjoint trees but it cannot guarantee full connectivity of a group member to all multicast trees, and the routing load is also doubled. To overcome this weakness, the Parallel Multiple Nearly-Disjoint Tree Multicast Routing (Parallel MNTMR) protocol was proposed. This protocol allows the overlapping of these multicast trees under inevitable conditions to ensure full connectivity. In addition, the construction of multiple trees is carried out in a single routine to reduce the overall control overhead. This is done by dividing the network virtually into two parts and tree construction is carried out simultaneously at both virtual topologies. Simulation has shows that the combination of these protocols with MDC scheme can improve the video quality in term of reducing the number of interruptions of the received video. Both protocols discussed above are source-initiated mesh-based protocols. The multicast group members do not re-broadcast the received video, therefore, a reasonably high density of mobile nodes is required to ensure good connectivity of the multicast tree.
1.3 Objectives of the Dissertation

The main objective of this dissertation is to address the problem of supporting best-effort video transmission over infrastructureless mobile ad hoc networks, motivated by the increasing popularity of these networks in future ubiquitous communication environment. More specifically, we outline the following targets:

- To analyze the challenges of supporting video streaming over mobile ad hoc networks that experience rapid changes in its topology.
- To design a framework for reliable video unicast over mobile ad hoc networks.
- To propose a framework to support reliable video multicast over mobile ad hoc networks.
1.4 Dissertation Outline

In this chapter a brief introduction on mobile ad hoc networks and its related issues are presented. Besides, we also analyze the challenges faced in dealing with supporting video transmission over mobile ad hoc networks. A complete literature review on the previous work related to this dissertation is also given. The rest of the dissertation is organized as follows.

In Chapter 2, we present a framework for multipoint-to-point (MP2P) video transmission. The proposed framework consists of three stages: source searching stage, routing discovery stage and video streaming stage. We propose a receiver-initiated source searching mechanism with synchronization ability to ensure the video transmission at different video senders are started at close interval in order to avoid excessive frames jitter. For the coding of video frames, we adopt the concept of Multiple Description Coding (MDC) scheme to generate several independent and equally important video descriptions to increase the robustness of the framework. We perform preliminary simulation study to evaluate the performance of the proposed framework using an event-driven simulator called NS-2 and also over a small testbed comprising of several laptop computers.

In Chapter 3, we complete the MP2P framework by proposing an extension to the Dynamic Source Routing (DSR) protocol in order to discover optimally-disjoint routes for different video sources; it is called Multipoint-to-Point Dynamic Source Routing (MP2P-DSR) protocol. Specifically, this protocol is designed to encourage the discovery of disjoint routes for different video senders through a preventive mechanism, and to diverse video streams from different senders transmitting via the same intermediate node using a corrective mechanism. The complete framework is evaluated extensively to demonstrate the outstanding performance of the proposed framework.

In Chapter 4, we tackle the problem of video multicast over mobile ad hoc networks. We propose an extension to the well-known Multicast Ad Hoc On-Demand Distance Vector (MAODV) routing protocol to construct multiple disjoint trees for distributing video coded using Multiple Description Coding (MDC) scheme. Specifically, we modify the tree construction and maintenance mechanisms to allow a mobile node to connect a multicast group via multiple disjoint paths in a single routine; by doing this, multiple disjoint trees can be constructed with minimum increase in the control overhead. We also perform extensive simulation using NS-2 to verify the performance of the proposed routing protocol.

In Chapter 5, we conclude this dissertation by emphasizing on the contributions of this dissertation. Besides, we also identify some open issues for further research.
References


CHAPTER 2
MULTIPOINT-TO-POINT VIDEO TRANSMISSION

2.1 Motivation

The concept of multipoint-to-point (MP2P) communication has been implemented in wire-line networks such as Internet and peer-to-peer networks to provide best effort real-time video transmission over lossy networks without QoS guarantee [1-4]. In [1], the authors proposed an approach to distribute video coded with Multiple Description Coding (MDC) scheme from several servers in a content delivery network. Their work concentrated on the problem of server coloring (distribution of video descriptions), server placement, and server selection for optimum performance. In [2], multiple distributed video servers were used to transmit video over the Internet. This approach requires periodical synchronization among the servers to coordinate the data rate and the video sequence. The concept of media streaming over peer-to-peer networks was introduced in [3] assuming that the bandwidth available is predetermined. In [4], a receiver-driven multistate transport protocol called R^2CP was introduced to support both unicast and multipoint-to-point real-time video communications over peer-to-peer networks.

Undoubtedly, multipoint-to-point communication provides a feasible solution for reliable video transmission over lossy networks because as long as a receiver is still connected to one of the servers, a continuous video streaming can be guaranteed. We believe this concept is suitable for ad hoc networks as well because the mesh topology of an ad hoc network, where each node is virtually connected to nodes within its transmission range, is itself a readily available platform for multipoint-to-point communications. However, the models presented above may not suitable for mobile ad hoc networks due to two main reasons. First, the network topology of ad hoc networks changes regularly due to the node mobility. Second, an end-to-end connection in ad hoc networks consists of several wireless connections and the rate of link breakage is higher than the single-hop wireless networks. Taking into account the above challenges, in this chapter we design a multipoint-to-point communication framework for video streaming over mobile ad hoc networks [5-7].
2.2 Multipoint-to-Point Video Transmission Framework

2.2.1 Potential Applications

The ultimate objective of the proposed MP2P framework is to provide a reliable means of supporting best-effort real-time video streaming that fulfills the basic requirements on bandwidth, delay and jitter in the mobile ad hoc networks. Due to the unique characteristics of ad hoc networks, the currently used video streaming mechanisms may not suitable in terms of fault-tolerance and robustness. In this section, the potential applications of the proposed MP2P framework on several types of video applications are described.

Contributed by the peer-to-peer technology, download-and-play video has become increasingly popular. It was estimated that more than 60% of the traffic on the Internet is contributed by the peer-to-peer traffic, which mainly is audio and video traffic. The feasibility of using peer-to-peer technology to support live video streaming over wire-line networks has also been widely studied [8]. With the developing of portable multimedia players with large storage capacity, high processing power and wireless communication technology, download-and-play video in wireless domain is expected to be one of the killer applications for the multimedia industry in the near future. Besides, the mesh topology of an ad hoc network is itself a readily available platform for multipoint-to-point transmission. However, due to the absence of a centralized indexing server in ad hoc network, a fast and efficient information searching mechanism is needed. In addition, the proposed MP2P framework in this dissertation is designed to allow the video to start after a short waiting time, instead of having to wait until the download is completed. Figure 2.1a illustrates download-and-play video application; node C receives the same video file stored in nodes A and B, both have the same video file stored their media library. This method allows uninterrupted video streaming as long as node C is at least connected to either node A or node B.

Figure 2.1b illustrates live video streaming from an infrastructure network to a wireless community network, which is created using the concept of ad hoc networking. One practical application of this concept is to provide video streaming, such as IPTV, to area that has difficulty to build the necessary infrastructure, for example the rural area. This solution is in parallel with the deployment of wireless ad hoc network as the last-mile solution for rural area. The use of MP2P in this scenario allows the distribution of workload among the members of the ad hoc network. It is important to note that the participant of a member to the ad hoc network is completely voluntary, and hence it is a fairer approach to share the workload among the members to avoid excessive demand to a particular member. Besides, this approach ensures continuous video streaming when some of the members leave the ad hoc network.
There is no limitation on the use of the MP2P framework in the above video applications only. However, for some video applications with more rigid requirements, the above MP2P model may not suitable or should be enhanced accordingly. For example, the MP2P system may become resource consuming and complicated in videoconference due to multiple concurrent video streaming sessions. Besides, it is also important to note that ad hoc networks provide no guarantee on the video quality due to the unpredictable nature of these networks. For instance, the limitation on supporting high-definition video-on-demand with rigid requirement on bandwidth is not the transmission model but the network itself. Therefore, this dissertation focuses on best-effort video streaming that fulfills the basic requirements only.

![Figure 2.1 Various types of video applications using MP2P framework: (a) Download-and-play video streaming; (b) Live video streaming.](image)
2.2.2 Novel Architecture

The basic idea behind the proposed method is to implement concurrent video streaming from several nodes to a single receiver. The novel architecture is given in Figure 2.2. In this, the receiver is given the responsibility to select the video sources, and to instruct these video sources to encode the video accordingly and to send out different video descriptions. The same MD codec is assumed to be installed in every mobile node so that identical video descriptions are created and the total time elapsed for video coding is approximately the same. In addition, the receiver also ensures that the received video frames are encoded correctly in the right sequence for viewing purpose. For this purpose, a playout buffer is used. Besides, this buffer also absorbs the transmission delay and frame jitter from different video sources.

In general, the MP2P framework is divided into three stages: source searching stage, route discovery stage, and video streaming stage. In this chapter the focus is on the source searching stage and the video streaming stage. For the route discovery stage, the Dynamic Source Routing (DSR) protocol is used without modification because it is desired to evaluate the suitability of using currently available routing protocols for the proposed MP2P framework.

![Figure 2.2](image) Novel architecture of the MP2P framework.
2.2.3 Source Searching Stage

Both source-initiated and receiver-initiated searching mechanism can be used to locate the video sources but the receiver-initiated approach is used in this dissertation in order to synchronize the starting time of the subsequent stages and to avoid large difference in the starting time of the video transmission of the different video sources. Three types of packet are used in this stage: video request packet (VREQ), video reply packet (VREP), and video select packet (VSEL). For this purpose, a special header carrying the control information is added after the IP header. Figure 2.3 gives the additional header for each type of packets.

(a) VREQ packet

(b) VREP packet

(c) VSEL packet

Figure 2.3 Video Request Option header.
Since the DSR protocol is used as the routing protocol, the packet headers for source searching stage are designed based on the standard format of the DSR protocol in order to standardize the format of the control packets for all the three stages. Therefore, these packets also carry in their header the complete addresses of the route. For VREQ, every node that forwards the packet must record its address in the packet header. For VREP and VSEL, these packets are forwarded according to the addresses recorded in the packet header. An 8-bit unsigned integer called Type field is used to differentiate these packets, and an 8-bit Data Length field is used to record the length, in octets, of the option. For VREQ and VSEL, the required data, for example the video filename or the video stream ID, is recorded using a 32-bit unsigned integer. For VSEL, the parameters used for video coding and the identity of the other video sources are also carried in its header.

The source searching mechanism is given in Figure 2.4. The receiver initiates a source searching by broadcasting a VREQ to the networks and a timer that expires after VREQ_WAIT_TIME is activated. Nodes with the desired video may reply by using VREP, which can be broadcast or unicast using the reversed route. Since the proposed method records the route traversed by VREQ in its header, the VREP is returned unicast along the reverse route in order to reduce the control load. Upon the timeout of the waiting period, the receiver selects the most suitable nodes to become the video sources. For example, two video sources, named nodes S_1 and S_2, are selected based on the arrival time of their VREP, where the reply from S_1 arrives before S_2. If the receiver receives only one reply before the timeout of the timer, normal point-to-point video transmission is executed. Finally, the selected video sources are informed using VSEL. Each VSEL carries information used for MD coding, i.e. the number of video descriptions to be created and which description is to be sent. For point-to-point transmission, both values are set to one. In order to help in the discovery of disjoint routes in the next stage, the identity of the other video source (for example, the IP address) is carried by the VSEL packets as well.

The sending of VSELS is associated with a simple mechanism to synchronize the starting time of the route discovery stage at the video sources. In the previous example, VSEL to S_1 is sent out T_d/2 after VSEL to S_2 is sent out, where T_d is the difference between arrival times of VREP_1 and VREP_2. The objective of this mechanism is to ensure that both video sources start their route discovery at virtually the same time and consequently, the starting of video streaming will also be virtually concurrent, as shown in Figure 2.4. Only a loose synchronization is provided here because a more precise synchronization would involve a greater number of packet exchanges and would subsequently increase the control overhead and result in longer delay in the video streaming. Besides, the proposed mechanism together with a small playout buffer at the receiver is sufficient to provide best-effort video at good quality, as demonstrated in the simulation study.
Figure 2.5 gives a general view of the synchronization mechanism that involves N video sources. Upon the timeout of the waiting time, the receiver selects N most suitable video sources and uses Equation 2.1 to calculate the sending time of VSEL to each video source x. The first VSEL is sent to the source that is farthest in terms of travelling time and the subsequent VSELS are delayed by the difference in the one-way travelling time, given as (\(\tau_N - \tau_x\))/2. In this calculation, the average transmission time is used to estimate one-way transmission time that can be derived as number of hops traversed multiplies with NODE_TRAV_TIME, which represents the average one-hop traversal time for a packet including queuing delay, interrupt processing times, transfer times and etc.

\[
T_x = T_N + \frac{(\tau_N - \tau_x)}{2}; \ x = 1, 2, \ldots, N - 1 \quad \ldots \quad (2.1)
\]

Figure 2.5 Sending of VSEL with synchronization.
2.2.4 Route Discovery Stage

This stage aims to search for a valid route or routes connecting the video sources and receiver. The main focus of this chapter is not the routing issue but the searching and streaming issues. For the route discovery stage, the original DSR protocol is used without modification [9]. This protocol is used because it has been proven to perform well under different network conditions [10]. This stage involves sending of RREQ by the video source and replying by the receiver using RREP.

Dynamic Source Routing (DSR) protocol is a simple yet efficient reactive routing protocol that is completely self-organizing and self-configuring. The main feature of DSR is that every data packet carries in its header a complete list of nodes through which the packet must be transmitted; it is called the source route of the packet. This protocol is composed of two mechanisms: route discovery and route maintenance. Route discovery mechanism is started by a node wishing to send data to a destination and having no valid route to this destination. Consequently, this node broadcasts a route request packet (RREQ) to discover a valid route to this destination. Any node that can provide a valid route to this destination and the destination itself can reply to this request by unicast a route reply packet (RREP) along the reverse route back to the requestor. Figure 2.6 demonstrates a RREQ sent by node S to obtain a valid route to the destination node D. Assume both nodes A and B have no knowledge about the destination, the RREQ is rebroadcast after their IDs are recorded at the packet header. Upon arriving at the destination, a RREP is created to carry the source route back to the requestor along the reverse route traversed by the RREQ. On the other hand, assuming node C has a valid route to node D in its route cache, it can then create a complete route to the destination by combining the route traversed by the RREQ and the route in its route cache. This resultant route is carried by a RREP back to the requestor.

![Figure 2.6 Route discovery in the DSR protocol.](image-url)
The confirmation of receipt of a data packet can be handled by the MAC layer or through a passive acknowledgment mechanism. If a node has transmitted a packet for a maximum number of allowed retries but no confirmation of receipt is received, a link breakage is detected. This node should inform the packet sender about the broken link with a route error packet (RERR). Upon receiving a RERR, a node removes routes that consist of the broken link from its route cache. Basically, this protocol does not support local repair of broken route. Some additional features are also included in this protocol to enhance its performance:

- A node forwarding or overhearing packet transmission of its neighboring nodes (promiscuous mode) may add the routing information to its route cache. Figure 2.7(a) reveals this phenomenon. When a data packet arrives at node A, the source route of this data packet is checked and the useful part, \( \text{i.e. \{A, B\}} \), may be stored in its route cache for future use. Besides, when node X, which is a neighbor of node A, overhears the packet transmission of node A, the source route of the packet is observed again and the resultant route, \( \text{i.e. \{X, A, B\}} \), may also be added to its route cache. Overhearing is a unique characteristic of wireless networks due to its inherent broadcast nature.

- Automatic route shortening may be performed when a node detects one or a few intermediate nodes in the route become no longer necessary. For example in Figure 2.7(b), if node C overhears the transmission of packet from node A to node B, which will be received by itself later, node C initial a route shortening and inform node S using a gratuitous RREP.

- Packet salvaging is performed once for every packet failed to be sent out due to a link breakage, after a RERR is sent to the packet sender. An intermediate node may safe the packet by checking its route cache for a valid route to the destination. If a valid route is found, the packet is transmitted along the new route. In Figure 2.7(c), upon detecting a link breakage to node C, which is the next hop recorded in the packet header, node B sends a RERR back to node S, and salvages the packet by overwritten the source route to another valid route, in this case from \( \text{\{S, A, B, C, D\}} \) to \( \text{\{S, A, B, E, F, D\}} \).

![Figure 2.7 Enhancements in the MP2P-DSR protocol.](image-url)
2.2.5 Video Streaming Stage

Again, it is important to note that the same MD codec is assumed to be installed in every node. This dissertation generally deals with N-source and N-description model. The video sources encode the video frames and send out only one video description based on the instruction carried by the VSEL. The video streaming is started once a valid route to the receiver is available and no further synchronization is performed here. Instead, a playout buffer is used at the receiver to absorb the difference in arrival time of video packets, and for re-sequencing purposes. The playout buffer is activated immediately after the first video packet arrives at the receiver. Packets that arrive after their deadline are discarded because only best-effort services are considered. The size of the playout buffer should be carefully designed by taking into considering the maximum number of hops can be handled by the DSR protocol (MAX_SR_LEN). Usually, the delay caused by re-sequencing involves a complicated analysis as given in [11]. However, it is generally safe to design the playout buffer by considering the worst case, in which the nearest source is one hop away and the farthest source is MAX_SR_LEN hops away from the receiver. From Figure 2.4 and by assuming the route discovery stage is started at the same time at both video sources, the maximum difference is given by

$$\Delta t_{\text{max}} = 3 \times (\text{MAX\_SR\_LEN} - 1) \times \text{NODE\_TRAV\_TIME} \quad \ldots \quad (2.2)$$

where NODE\_TRAV\_TIME is a conservative estimate of the average one-hop traversal time for a packet including queuing delays, interrupt processing times, transfer times and etc.; a common value for this parameter is between 20 and 50 milliseconds. The coefficient 3 indicates that this calculation also takes into consideration the time difference for route discovery, as shown in Figure 2.4, by assuming the route discovery is started at the same time in both video sources. Generally, the buffer size should be larger than $\Delta t_{\text{max}}$ in order to accommodate the delay for frames re-sequencing as well.
2.3 Simulation Study Using NS-2

2.3.1 Simulation Settings

This section explains the preliminary simulation using NS-2 [12] with CMU wireless extension [10] to compare three cases. Case I is point-to-point video transmission using SDC; Case II is point-to-point video transmission using MDC with two descriptions of video; and Case III is the proposed mechanism that involves two video sources and one receiver. The simulator uses the IEEE802.11 protocol in the MAC layer working in the Distributed Coordination Function (DCF) mode, a form of Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA). Its physical layer features are not modeled. The channel bandwidth is 2 Mbps and the transmission range is 250 meters. In other words, the topology used in this simulation is a homogenous network.

The simulation topology contained 20 nodes within an area of 1000 meters by 600 meters. A rectangular shape area has longer average length of routes to create more connection breaks during the simulation [13]. The node mobility was modeled using the Random Waypoint Model used in [14]. The level of mobility depends on the maximum speed of the nodes. Six different maximum speeds were used: 5.0 m/s, 10.0 m/s, 12.5 m/s, 15.0 m/s, 17.5 m/s and 20.0 m/s, respectively. Also, only continuous mobility case with zero pause time is considered. In each scenario, each node is randomly assigned with an initial location, a destination and a traveling speed, the speeds being uniformly distributed between the given minimum and maximum speeds. During the simulation, these nodes are assumed to travel from their initial location to the destination at the assigned speed. After they reach the destination, a new destination and traveling speed are assigned, and the movement continues. This is repeated until the end of the simulation. For each maximum speed, 10 scenarios were generated to obtain the average results. Five random cross traffics of 8.2 kbps each were also introduced in the network.

We evaluated the performance of the proposed mechanism using interactive video application, which contains 10,000 frames coded at 128 kbps. For SDC, each video frame was contained into a single packet, which is 1024 bytes in size, for transmission. For MDC, each video frame was encoded into two descriptions; each description was transmitted in a single packet with 512 bytes in size. The frame interval was 64 ms. Each simulation ran for 900 seconds and the results were averaged over the ten scenarios.
2.3.2 Results and Discussion

The quality of a video frame depends on the number of descriptions received. For MDC, a frame is called a good frame if both descriptions are received. If only one of the two packets is received, the decoded frame is of acceptable quality. Otherwise, the frame is declared as a bad frame [4]. For SDC, the frame is either good or bad. When one or a few consecutive bad frames are received, an interruption during the video viewing is observed and it is called an interruption. Besides, the number of consecutive bad frames received indicates the length of an interruption. The performance evaluation is based on the number of interruptions and the average length of these interruptions. The smaller value these parameters are, the better the video quality is.

Figure 2.8 shows the number of interruptions. As may be seen, Case II has the highest number of interruptions, whereas Case III has the lowest. Numerically, the number of interruptions in Case III is about 30% to 45% lower than Case I, and 45% to 55% lower than Case II. Clearly, the proposed MP2P framework successfully reduces the number of interruptions. Besides, the poor performance of Case II implies that the implementation of MDC does not necessarily enhance the video quality, but the additional workload may worsen the network performance. In the MP2P framework, the use of multiple video sources to distribute MD coded video can minimize the effect of this additional workload and subsequently, the strength of MDC is fully utilized, as shown in the simulation results.

![Figure 2.8 Simulation result: Number of interruptions.](image-url)
The average length of interruptions is shown in Figure 2.9. The graph shows that Case III has the shortest average length of interruptions, and this is followed by Case II. Case I, which does not implement MDC, has the longer average length of interruptions. Case III gives the shortest average length of interruptions, which is about 2 to 5 frames shorter than Case I. Therefore, it is fair saying that the implementation of MDC (Case II and III) successfully shortens the length of interruptions. Moreover, when MD coded video are transmitted from different sources, the packet losses become less correlated and the probability of receiving at least one video description is generally higher. Consequently, less and shorter interruptions are observed.

![Figure 2.9 Simulation result: Average length of bad periods.](image)

Next, the quality of the received frames in each case is observed. Figure 6 shows the comparison. For Case I, the frame is either good or bad because there is only one description per frame. For Cases II and III, the frames are categorized as good, acceptable or bad depending on the number of descriptions received, as explained above. In Figure 2.10, it can be seen that the proposed framework has the smallest number of bad frames, which is about 5% less than Case I. Besides, the use of MDC alone as in Case II does not reduce the quantity of bad frames, as shown in Figure 2.10(b). The explanation for this observation is the additional workload that has worsened the network performance. In terms of the quantity of good frames, it can be observed that the proposed mechanism has the lowest percentage of good frames. This situation is explainable because the number of acceptable frames is higher.
Figure 2.10 Simulation result: Frames distribution based on quality: (a) Case I; (b) Case II; (c) Case III.
The main drawback of MDC is the higher overhead due to a greater number of packets sent. In this simulation, the number of packets sent in Cases II and III are doubled as compared to Case I. Figure 2.11 shows the network throughput. As predicted, Case II has the lowest throughput due to the addition overhead that degrades the network performance. This negative impact is made less significant in the proposed MP2P framework because better load balancing is achieved by the use of multiple sources. In the same graph, Cases I and III have almost the same throughput, where the difference is only between 0.2% and 0.6%; Case III is 1% to 2% better than Case II.

![Figure 2.11 Simulation result: Network throughput.](image-url)
2.4 Performance Evaluation Using Ad Hoc Network Testbed

Experiment on real ad hoc networks is a challenging task due to many reasons. First, a large number of mobile nodes, as well as a large area to place these mobile nodes, are required in order to create a mesh topology with good connectivity for packet relaying. Besides, monitoring of these mobile nodes requires a lot of manpower and real-time synchronization among these nodes is rather difficult to be carried out. Furthermore, it is complicated to create a random movement because a pre-designed movement may lack of randomness. General speaking, the development of a real ad hoc networks involves very high cost. This explains the reason why the performance evaluations in this dissertation are mostly based on the NS-2 simulator, which is a reliable event-driven simulator.

In this section, we developed an ad hoc network testbed to demonstrate multipoint-to-point data streaming. A comparison between point-to-point and multipoint-to-point transmission was not carried out because such a small testbed is not suitable for a fair comparison. Therefore, only the characteristics of multipoint-to-point transmission are demonstrated in this section [15].

2.4.1 Setup of the Testbed

The ad hoc network testbed was developed using five Panasonic Let’s Note Light W5 laptops with IEEE 802.11a/b/g cards operated in ad hoc mode. Table 2.1 gives the details of these laptops. The experiment was carried out inside the Ishii Laboratory, Building 9 of Shonan Campus of Tokai University. In the building, there was interference from other wireless LAN access point using IEEE 802.11 standard, and other electronic devices, such as microwave oven. Static Routing was used to force the data transmission following the assigned route as shown in Figure 2.12. For this purpose, the Optimized Link State Routing (OLSR) protocol was used at the network layer [16]. This mechanism allows the experiment to be carried out within a small area. In this experiment, two video sources were used. Generally, Nodes 1 and 5 were data sources, and node 3 was the receiver. Nodes 2 and 4 were the relay nodes for nodes 1 and 5, respectively.

<table>
<thead>
<tr>
<th>Processor</th>
<th>1.06 GHz</th>
</tr>
</thead>
<tbody>
<tr>
<td>Memory</td>
<td>512 MB</td>
</tr>
<tr>
<td>Wireless Network Interface Card (NIC)</td>
<td>IEEE 802.11 g</td>
</tr>
<tr>
<td>Operating System</td>
<td>Windows XP</td>
</tr>
</tbody>
</table>
The video traffic was modeled at 128 kbps; 64 kbps for each video description, using LAN Traffic v2 [17]. Each video description was packetized into a single packet for transmission; the packet size was 512 bytes and the frame interval was 64 ms. UDP was used at the transport protocol. Two sets of experiments were carried out, and each experiment was repeated 5 times to obtain the average results. Each experiment ran for 20 seconds. Table 2.2 summarizes the settings of each experiment. During the experiment, the unavailability of a node, due to movement or other reasons, was simulated by switching off its NIC.

**Table 2.2** Experiment settings.

<table>
<thead>
<tr>
<th>Experiment</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>All nodes were switched ON at the beginning of the simulation. Subsequently, Node 2 and Node 4 were switched OFF together for 2 seconds. Then, Node 2 was turned OFF again for 2 seconds, followed by the same the action at Node 4.</td>
</tr>
<tr>
<td>2</td>
<td>All nodes were switched ON at the beginning of the simulation. Subsequently, Node 2 and Node 4 were switched OFF together for 2 seconds. Then, Node 4 was turned OFF again for 2 seconds, followed by the same action at Node 2. Next, Node 4 was switched OFF for another 2 seconds</td>
</tr>
</tbody>
</table>
2.4.2 Results and Discussion

The performance evaluation is based on the quality of the video frames received. Since the MDC scheme was used in this experiment, a bad quality frame is created only when both video descriptions are lost in transmission. If either one of the two video descriptions is received, the quality of this frame is categorized as acceptable. If both video descriptions are lost, a frame with bad quality is inserted. Figure 2.13 presents the experimental results for both experiments. Generally, the percentage of bad frames is about 20%. It can be observed that the result of each trial is different due to the random effects that influence the data transmission, such as interference from other mobile devices, and time difference in switching the mobile node to ON and OFF.

![Figure 2.13 Simulation result: Quality of video frames.](image)

In order to have a better view of the characteristic of multipoint-to-point transmission with the MDC scheme, a mapping of the results from Trial 5 of Experiment 1, which is among the best performance, to the real video trace, is performed. The first 800 frames of ‘highway’ sequence in quarter common intermediate format (qcif) [18] were selected. The MD codec used were obtained from [19] and [20], based on the per-frame allocation scheme explained in Section 1.2.2 of Chapter 1. A frame is successfully decoded if the packet containing this frame is received successfully. The Peak Signal-to-Noise Ratio (PSNR) of the reconstructed frame is given in Figure 2.14.
Two important observations are obtained here. First, an obvious distortion is observed only when both video descriptions are lost in transmission for a long interval, as shown between frames 230 and 420. When any one of these video descriptions is received successfully, a fluctuation in the frame quality is observed, but not a noticeable distortion during video viewing. Second, by using frame compensation together with MDC, in which an undecodable frame is replaced with its previously decodable frame, can significantly increase the overall quality of the displayed video because the neighboring frames usually have small difference. For frames 230 to 420, the PSNR drops to 25dB but for frame 670 to 800, the lowest PSNR is around 30dB.
2.5 Conclusions

Supporting video transmission over mobile ad hoc networks is a challenging task due to the unstable ad hoc connection between nodes and the inflexible characteristics of the conventional video coding scheme. Motivated by the concept of peer-to-peer sharing in the wire-line networks, we proposed to combine multipoint-to-point transmission with Multiple Description Coding to support best-effort video transmission over mobile ad hoc networks in this chapter. The proposed framework consists of three stages: source searching stage, route discovery stage, and video streaming stage. Specifically, we introduced a receiver-initiated source searching mechanism that provides soft synchronization, and analyzed the requirements on play-out buffer at the receiver to support this framework. Preliminary evaluation was carried out using an event-driven simulator called NS-2 and a small ad hoc network testbed. In both cases, the MP2P framework gives better performance than the conventional point-to-point transmission. Moreover, this improvement is mostly contributed by multipoint-to-point transmission because using MDC alone does not improve the video quality as the number of interruptions during video viewing is worse than the point-to-point transmission.
References


http://www.isi.edu/nsnam/ns/


[16] OLSR (Optimized Link State Routing).
   http://hipercom.inria.fr/olsr/

   http://www.zti.fr/public/others/index_gb.htm

[18] QCIF Sequences.
   http://trace.eas.asu.edu/yuv/index.html

   http://iphome.hhi.de/suehring/tml/download/old_jm/

[20] Multiple Description Coding.
   http://140.116.72.80/~smallko/ns2/MDC.htm
CHAPTER 3
MULTIPOINT-TO-POINT DYNAMIC SOURCE ROUTING PROTOCOL

3.1 Motivation

In Chapter 2, we implemented the multipoint-to-point framework based on the concept of server diversity to support best-effort video transmission in mobile ad hoc networks. Besides, we also used Multiple Description Coding (MDC) scheme for video coding to create several independent and equally important video descriptions. It is commonly known that the strengths of MDC can only be fully utilized when these video descriptions have low correlated packet losses because low quality video streaming can be guaranteed as long as any one of the video descriptions arrives at the receiver without error. For that reason, it can be easily predict that there is a bottleneck in the proposed method when one or more video descriptions are transmitted via the same intermediate node, as shown in Figure 3.1. If node X is congested, the receiver can still receive video signal from Source 3, and thus, only a drop in video quality is observed. However, if the congestion occurs at node Y causing all video descriptions being dropped, a service interruption will be observed. In light of this, there is a need to implement path diversity in the proposed framework to encourage the use of disjoint routes for sending these video descriptions. In this chapter we tackle the routing issue at the network layer to further enhance the proposed MP2P framework.

Figure 3.1 Motivation of the MP2P-DSR protocol.
3.2 The MP2P-DSR Protocol

The DSR protocol is an on-demand routing protocol for multi-hops wireless ad hoc networks in which the end-to-end route from source to destination is carried in the packet header [1]. In this chapter we extend the routing protocol to find disjoint routes for different video sources. The extended protocol is called Multipoint-to-Point Dynamic Source Routing (MP2P-DSR) protocol [2]. It is important to note that this extension is only applied to video traffic that implements the MP2P framework, and non-video data should follow the original DSR protocol. It is important to note that the requestors (video sources) are synchronized to start their route discovery at virtually the same time, as discussed in the previous Chapter. Generally, the extension is comprised of two mechanisms: preventive and corrective mechanisms.

3.2.1 Preventive Mechanism

The preventive mechanism aims to avoid the convergence of video streams at the same intermediate node by encouraging the discovery of disjoint routes during the route discovery stage. To assist in this process, Equation 3.1 is introduced to measure the correlation factor of two routes. The shared-node ratio (SnR) of two routes, \( R_1 \) and \( R_2 \), is calculated by dividing the number of common nodes of these two routes to the length of the shorter route [3].

\[
\text{SnR}(R_1, R_2) = \frac{N_{\text{shared node}}(R_1, R_2)}{\text{Length}_{\text{min}}(R_1, R_2)} \quad \ldots (3.1)
\]

The complete algorithm is given in Figure 3.2. In this algorithm, three threshold values, \( i.e. \) TH1, TH2, and TH3, are defined to indicate the level of disjointedness. Generally, TH1 and TH3 can take a large value to indicate a rigid requirement on the disjointedness. However, TH2 should take a moderate value because a large value may make the route discovery mechanism inefficient. The main characteristics of the preventive mechanism are highlight below [4].

i) Each duplicated RREQ is not discarded immediately at the intermediate nodes. Instead, the route it traversed is analyzed to provide a backup route if it is highly disjoint (< TH1) with the route traversed by the first RREQ, and better than the existing backup route, if available. This backup route is used to provide another route for the requestor, as shown in Figure 3.3(a). The second RREQ received at node X provides a backup route \( R_B \). After receiving RREP1, RREP2 is created and forwarded after RREP1 is forwarded. The route carried by RREP2 is Route 2 and the remaining route from node X to node D as recorded in RREP1. This step aims to provide more disjoint routes for the requestor.
Receive RREQ originated by video source A

Yes

Video source?

No

Destination?

Yes

Duplicate?

No

RF = RC

Yes

RF = Route traversed by first RREQ arriving at an intermediate node
RB = Backup route recorded at an intermediate node
RC = Route traversed by current RREQ arriving at an intermediate node
TH1, TH2, TH3 = Threshold values
RD(i, j) = Route i replied to video source j by the destination.

SnR(RC, RD(i, j)) < TH2?

No

j = A

i = 1, 2, 3 …

SnR(RC, RC) < TH1?

Yes

R_B = null?

No

Forward

No

SnR(RC, RC) < SnR(RB, RC) ?

Yes

RB = RC

No

SnR(RC, RD(i, j)) < TH3?

j \neq A

i = 1, 2, 3 …

priority = 0

priority = 1

Reply RREP (priority)

END

R_F = Route traversed by first RREQ arriving at an intermediate node
R_B = Backup route recorded at an intermediate node
R_C = Route traversed by current RREQ arriving at an intermediate node
TH1, TH2, TH3 = Threshold values
RD(i, j) = Route i replied to video source j by the destination.

Figure 3.2 Preventive mechanism algorithm.
ii) The original DSR protocol allows the destination node to reply to all RREQ arrived, and this creates a scenario where most or all immediate (one-hop) neighbors of the destination are involved in the route created for one requestor. Consequently, route discovered by the other requestors may have low disjointedness. In order to overcome this problem, selective reply is introduced. If a route has high ratio of shared-node ($\geq TH_2$) with route which replies have already been sent, this RREQ is discarded. Figure 3.3(b) shows an example of selective reply.

iii) Each route for which the destination sends a reply is given a priority. The priority is determined by comparing the route traversed by the RREQ with routes for which replies have previously been sent to the other requestors (video sources). If the shared-node ratio is low ($< TH_3$), it means that this route is highly disjoint with routes used by the other video sources; therefore, it is given high priority. Otherwise, a low priority is assigned. This priority is carried by the RREP back to the requestor. The video sources always use the shortest high-priority route for video transmission if available.

iv) Video sources cannot use route that includes other video sources in order to avoid an excessive workload at a video source. For this purpose, the video sources are informed with the identity of other video sources via the VSEL packets, as explained in Section 2.2.2 of Chapter 2.

![Figure 3.3 Preventive mechanism in the MP2P-DSR protocol.](image-url)
3.2.2 Corrective Mechanism

Clearly, the preventive mechanism encourages node disjointedness in video transmission, but with no guarantee because the topology may be such that no fully node-disjoint route exists. With this in mind, a corrective mechanism is introduced to encourage link disjointedness in the event that preventive mechanism fails to provide this. As shown in Figure 3.4, the corrective mechanism is activated when more than one video stream is transmitted via the same intermediate node X, which is not an immediate neighbor of the destination, and would continue their flows over the same route to the destination. When this happens, node X should find an alternative route to the destination from its route cache and transmit one of the video streams using this route. In order to avoid unnecessary delay in video transmission, the length of the alternative route should not be longer than the original one.

The algorithm is given in Figure 3.5. For this purpose, each intermediate node that forwards video packets is required to maintain two additional parameters: forward_video flag and current_video ID. The first parameter is set when this intermediate node has recently forwarded a video packet, whereas the second parameter records the address of the video source that originates its recently forwarded video packet. Besides, a timer that expires after EX_TIME is activated. Within the activation period of this timer, this intermediate node is expecting video packets from the video source recorded in current_video. Therefore, if it receives video packet from other video sources, the corrective mechanism is executed.

![Figure 3.4 Corrective mechanism in the MP2P-DSR protocol.](image-url)
Figure 3.5 Corrective mechanism algorithm.
3.3 Performance Evaluation

3.3.1 Simulation Settings

In order to evaluate the proposed framework, the entire MP2P framework was added to the NS-2 [5]. Besides, the DSR protocol was also extended accordingly. A simulation study was carried out to verify its performance. The NS2 is implemented with CMU wireless extension [6]. The channel bandwidth was set to 11Mbps to simulate higher bandwidth of the new IEEE802.11 standard. The simulation topology contained 50 nodes within an area of 1500 meters by 800 meters. Again, homogenous network was used in this simulation. The node movement was modeled using the enhanced Random Waypoint Model [7]. The minimum traveling speed was set to 0.1 m/s and the maximum speed was varied between 2.5 m/s and 15.0 m/s to model different levels of node mobility. For each level of mobility, 30 scenarios were generated prior to the simulation so that identical scenarios could be re-used for each case to ensure fairness in the comparison study. Table 3.1 shows statistical analysis for the scenarios generated for reference in the next session.

1. Average Node Degree (AND) is the number of one-hop neighbors of a node, and is averaged over all nodes and over the duration of the simulation.
2. Rate of Link Changes (RLC) is the number of times the connection between any two nodes is formed and broken divided by the simulation duration.
3. Length of Longest Shortest-Path (LSP) is the distance, in number of hops, of the longest shortest-path encountered by any two nodes within the simulation duration.

These values are the average values for all 30 scenarios. It can be observed that AND and LSP are independent of node mobility, and as expected, RLC increases with node mobility. To represent the background traffic, five constant bit rate (CBR) connections of 18 kbps each were introduced randomly in the networks. UDP was used in the transport layer.

<table>
<thead>
<tr>
<th>Max. Speed (m/s)</th>
<th>2.5</th>
<th>5.0</th>
<th>7.5</th>
<th>10.0</th>
<th>12.5</th>
<th>15.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>AND</td>
<td>7.5</td>
<td>7.8</td>
<td>7.6</td>
<td>8.1</td>
<td>8.0</td>
<td>7.9</td>
</tr>
<tr>
<td>RLC</td>
<td>0.3</td>
<td>0.6</td>
<td>0.9</td>
<td>1.0</td>
<td>1.3</td>
<td>1.5</td>
</tr>
<tr>
<td>LSP</td>
<td>10.3</td>
<td>10.1</td>
<td>10.7</td>
<td>10.7</td>
<td>10.9</td>
<td>11.2</td>
</tr>
</tbody>
</table>

The video sources and receiver were chosen randomly among the 50 nodes and recorded so that the same nodes were used for each case to maintain the fairness of the comparison study. The video sequence used was ‘highway’ in quarter common intermediate format (q CIF), which is 176 x 144 pixels/frame [8].
The first 1800 frames were used and they were encoded at 12 frames per second (f/s) using the H.26L video codec for both the SDC and MDC schemes [9]. The Group of Pictures (GOP) size was 12 frames and the default coding parameters were used for both SDC and MDC schemes to maintain the fairness of the comparison study. For MDC scheme, the per-frame separator in [10] was used to divide the raw video frames into two groups prior to implementing video coding as explained in Section 2.2, and the EvalVid toolset was used to obtain the video trace for the simulation [11]. The average PSNR ratio for the encoded video frame was 43.27 dB, and the average frame sizes were 4143 bits, 4657 bits, 4978 bits, and 5220 bits for one, two, three and four video descriptions, respectively. It is important to note that one video description represents the SDC video. Clearly, the MDC scheme adds redundancy to the encoded frames as explained in [12].

By using MAX_SR_LEN and NODE_TRAV_TIME equal to 20 hops and 40 ms, respectively, \( \Delta t_{\text{max}} \) is equal to 2.28 seconds using Equation 2.2 (in Chapter 2). In order to accommodate time elapsed for frame re-sequencing, which should be of moderate level [13], the buffer size is conservatively set to 5 seconds, a practical waiting time for real-time video transmission. Each simulation ran for 200 s and the results were averaged over the 30 scenarios.

The simulation study can be divided into two parts [14]. It is important to note that this dissertation only deals model with \( N \) video sources and \( N \) video descriptions. In the first part, we compared the performance of DSR and MP2P-DSR. Besides, we also included point-to-point communication in this part to observe the differences. In this second part, we evaluated the impacts of increasing the number of transmission nodes and the number of video descriptions used. For this purpose, we varied \( N \) from two to four, and the MP2P-DSR protocol was used. The default values were used in the simulation to maintain the fairness of the comparison study. As for the threshold values in MP2P-DSR, we set \( \text{TH}_1 \) and \( \text{TH}_3 \) to 0.8, which indicates a strict requirement on node disjointedness. For \( \text{TH}_2 \), the value was 0.5. In order to have a better view, Figure 3.6 summarizes all the cases used in this simulation study.

\[ \text{Figure 3.6 Simulation scenarios.} \]
3.3.2 Performance Parameters

The following parameters are used for performance evaluation:

(a) **Average Peak Signal-to-Noise Ratio (PSNR)** is the average value of the PSNR of the reconstructed frames, which is calculated by comparing the luminance value of each reconstructed frame with its corresponding original frame, as given in Equation 3.2. This parameter represents the quality of the decoded video as a whole.

\[
\text{PSNR} = 10 \log_{10} \left( \frac{V_{\text{peak}}}{\sqrt{\frac{1}{N} \sum_{i} \sum_{j} \left( Y_{\text{ref}}(i,j) - Y_{\text{prec}}(i,j) \right)^2}} \right) \quad \ldots \quad (3.2)
\]

where \( Y_{\text{ref}}(i,j) \) and \( Y_{\text{prec}}(i,j) \) are the pixel value of the reference and reconstructed frames, respectively; \( N \) is the total number of pixels in a frame; and \( V_{\text{peak}} \) equals to 255 for picture coded with 8-bit resolution.

(b) **Normalized Routing Load** is the ratio of total routing packets forwarded at each node to the total data packets received, expressed in percentage. This parameter measures the efficiency of the routing protocols. In the calculation, the data packets include both video and non-video packets because the routing information is used by both types of traffic, so they cannot be differentiated. This calculation does not affect the fairness of the comparison study because the number of data packets is the same for all cases.

(c) **Normalized Control Overhead** is the ratio of the total control overhead created during source searching and route discovery stages to the total number of data packets received. This parameter is also expressed in percentage and it represents the efficiency of the proposed framework as a whole.

(d) **Total Time elapsed for source searching** measures the amount of time spent for source searching.

(e) **Average End-to-End Delay** is the average end-to-end delay experienced by the video packets. This parameter gives an indication on the additional delay caused by the proposed routing protocol.

(f) **Throughput of Non-Video Traffic** is also measured to observe the influence of the proposed method on non-video applications, i.e. the background traffic.

(g) **Number of Bad Frames** specifies the quantity of decoded bad quality frames. For this purpose, we performed a PSNR to Mean Opinion Score (MOS) mapping for the video frames as recommended by ITU-T, and given in Table 2 [15]. We further simplified the classification into two categories only. A frame is considered as bad frame if the MOS is below 4.
(h) **Number of Interruptions** measures how often distortions occur during the video viewing. An interruption is observed when a number of bad frames are decoded consecutively. Naturally, the human visual system cannot notice the distortion if the interruption lasts for a very short period of time. In this part, we assumed an interruption is noticeable only if it lasts for at least 0.5 second.

<table>
<thead>
<tr>
<th>MOS</th>
<th>PSNR (dB)</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>dB &lt; 20</td>
<td>Bad</td>
</tr>
<tr>
<td>2</td>
<td>20 ≤ dB &lt; 25</td>
<td>Poor</td>
</tr>
<tr>
<td>3</td>
<td>25 ≤ dB &lt; 31</td>
<td>Fair</td>
</tr>
<tr>
<td>4</td>
<td>31 ≤ dB &lt; 37</td>
<td>Good</td>
</tr>
<tr>
<td>5</td>
<td>dB ≥ 37</td>
<td>Excellent</td>
</tr>
</tbody>
</table>

### 3.3.3 Results and Discussion

In this first part of the simulation, we compared C1_DSR, C2_DSR and C2_MP2PDSR. Figure 3.7 shows the average PSNR, which represents the overall video quality. As seen, MP2P (C2_DSR and C2_MP2PDSR) provides a better overall performance than P2P transmission (C1_DSR). Further improvement is achieved by using a routing protocol that uses disjoint routes for video transmission, as demonstrated in C2_MP2PDSR, where the improvement over C1_DSR is between 0.2 to 0.95 dB. In addition, C2_MP2PDSR provides a more stable performance over various levels of mobility. The fluctuation is only 0.35 dB for C2_MP2PDSR, and 0.5 dB for C2_DSR, but as high as 0.9 dB in C1_DSR. This also leads to the observation that the complete MP2P framework provides a larger improvement under high mobility. Since the video quality is directly proportional to the quantity of video frames successfully received, there are two factors that contribute to this improvement. First, the route discovery mechanism is enhanced to obtain more disjoint routes for the video sources, and using disjoint routes can achieve better workload balancing within the network, which leads to less congestion. Second, MD coded video has better error resilience and the impact of frames lost is generally smaller. The combination of these two factors contributes to the improvement of the overall video quality.

Figure 3.8 shows the normalized routing load for the three cases. As expected, C2_MP2PDSR has the highest routing load due to the additional routing packets created during the route discovery mechanism to find disjoint routes. The additional normalized routing load is about 2 to 4%. It is also of interest to observe the overall control overhead including both source searching and route discovery, as presented in Figure 3.9. It can be seen that Figures 3.8 and 3.9 exhibit the same pattern with small difference, which is only about 0.5%. The additional control overhead created during source searching stage is relatively
small because the source searching stage is executed only once at the beginning of
the simulation but the route discovery and route re-establishment are carried out
whenever the topology changes. From Figures 3.7, 3.8 and 3.9, a tradeoff between
control overhead and throughput is observed; the higher control overhead in the
proposed method has led to a better video performance.

---

**Figure 3.7** Simulation I result: Average PSNR.

**Figure 3.8** Simulation I result: Normalized routing load.
Chapter 3. Multipoint-to-Point Dynamic Source Routing Protocol  

Section 3.3

Figure 3.9 Simulation I result: Normalized control overhead.

Figure 3.10 shows the total time elapsed for source searching. Since C2_DS and C2_MP2PDSR use the same source searching mechanism, they have the same value for this parameter. Source discovery for the multipoint-to-point framework introduces a slight increase in delay of about 10 ms. This delay is due to the need to locate two video sources and for synchronization purposes as discussed in Section 2.2.2. In order to consider the delay caused by the modified routing protocol and the video transmission, the average end-to-end delay for video packets is shown in Figure 3.11. For most levels of mobility, C2_MP2PDSR has the highest delay, which is about 20 ms higher than the other cases for all levels of mobility, except for the highest speed of 15.0 m/s, where the increase rises to 60 ms. In summary, the proposed framework introduces higher overall delay in the system, but fortunately, the worst case of this additional delay is only about 70 ms (60 ms + 10 ms) that occurs at the maximum speed of 15.0 m/s. We believe the impact of this delay is small considering the size of the playout buffer, which is approximately ten times greater.

Figure 3.10 Simulation I result: Time elapsed for source searching.
As mentioned earlier, the MP2P extension added to the DSR protocol is only used by the video traffic but it is still possible that the extra routing overhead created, as discussed in the previous paragraph, may affect the non-video traffic as well. In order to check this possibility, the throughput of the background traffic is also plotted. As seen in Figure 3.12, the difference in throughput for non-video traffic is very small, less than 0.5%. Generally, the extra overhead created takes only a small portion of the processing power at each node. Therefore, the influence is small. Besides, MP2P transmission tends to distribute the video traffic evenly within the network to avoid local congestion. This directly reduces the loss of non-video packets due to congestion. The negative and positive impacts are offset and hence, it is fair to conclude that the proposed framework has insignificant influence on the non-video traffic.

**Figure 3.11** Simulation I result: Average end-to-end delay.

**Figure 3.12** Simulation I result: Non-video throughput.
Next, the video quality is inspected from a more detailed perspective. Figure 3.13 shows the total number of bad frames received. For C2_MP2PDSR, the number of bad frames is always below 50, which is less than 3% of the total number of frames. For C2_DSR, the number is slightly higher but the performance is still quite stable with respect to mobility. Once again, C1_DSR shows a large variation in performance with the bad frame ratio growing to almost 8% under the highest mobility. Also from Figure 3.13, it is fair to say that C2_MP2PDSR has almost perfect video viewing because the average number of interruptions is always below one. C2_DSR resulted in an average of about one interruption per video session while C1_DSR resulted in more than two interruptions per video session. These observations can be explained by checking on the rate of link changes (RLC) in the topology, as given in Table 3.1. At low mobility, the connection between video source and receiver is stable and using a single node is enough to provide video viewing with good quality. As mobility increases, the connection between the video source and receiver breaks frequently, but with MP2P transmission, the probability of losing connection to both video sources simultaneously is smaller than with point-to-point transmission, and thus a smaller number of interruptions is recorded. Furthermore, the use of disjoint routes in C2_MP2PDSR further increases the probability of receiving at least one video description, and this further reduces the occurrence of bad frames and interruptions.

![Figure 3.13 Simulation I result: Number of bad frames & interruptions.](image-url)
In the second part of the simulation, we inspected the impacts of increasing the number of transmission nodes and video descriptions from two to four; they are represented by C2_MP2PDSR, C3_MP2PDSR and C4_MP2PDSR. In this part, only four parameters are observed. Figure 3.14 shows the average PSNR of the received video. It shows that there is no linear relationship between the video quality and the number of transmission node. C3_MP2PDSR usually has better performance than C2_MP2PDSR, except at maximum speed of 7.5 m/s. However, no significant improvement is achieved in C4_MP2PDSR, except at maximum speed of 10.0 m/s. In summary, it is fair mentioning that using three transmission nodes gives the optimum performance in the MP2P framework. However, the largest difference in improvement is achieved when the number of transmission nodes is increased from one to two, which increases the average PSNR by about 0.9 dB as discussed previously.

![Figure 3.14 Simulation II result: Average PSNR.](image)

Again, the normalized routing load is plotted in Figure 3.15. As expected, the normalized routing load increases as the number of transmission nodes increases, except at maximum speeds of 2.5 m/s and 7.5 m/s. This observation is due to the randomness in the scenarios generated. It is important to note that simulation for ad hoc networks usually involves many random factors that affect the performance, and so 30 scenarios were created for each maximum speed to observe the average performance. Besides, the general assumption that the network performance decreases as the network mobility increases does not always hold true because the impacts of node mobility can be both positive and negative [13]. On the one hand, mobility creates frequent link breakage and thus, leads to an increase in the packet losses. On the other hand, mobility also makes possible the formation of more suitable topology from time to time to provide more reliable transmission. The positive and negative impacts of mobility make ad hoc networking more challenging, yet interesting.
Figure 3.15 Simulation II result: Normalized routing load.

Figure 3.16 shows the number of bad frames and the number of interruptions. Under most levels of mobility, increasing the number of transmission nodes reduces the values of both parameters. However, the improvement is rather small and not consistent as compared to the results for the same comparison in the first part of the simulation study, with respect to Figure 3.13.
3.4 Conclusions

In this chapter, we further enhanced the MP2P framework by taking into consideration issue at the network layer related to the routing protocol. More specifically, we combined the concepts of path and server diversities to develop a complete framework for multipoint-to-point video transmission. The motivation of this work lies on the fact that MD coded video descriptions should be distributed independently to reduce the correlation of their packet losses. As such, we added an extension to the DSR protocol to encourage the discovery of disjoint routes for different video sources, and to diverse the video streaming when more than one video description is sent via the same intermediate node. Generally, this extension is comprised of the preventive and corrective mechanisms. A more comprehensive simulation was carried out in this chapter, and the following conclusions are obtained:

1. The proposed MP2P framework has been proven to outperform the conventional point-to-point transmission in terms of the overall video quality and the number of interruptions during the viewing stage, especially under high mobility.

2. The extension added to the DSR protocol allows MD coded video descriptions to be distributed along disjoint routes. Consequently, lower correlated packet losses of these video descriptions are observed, and hence, the overall video quality is further improved. There is a cost of using the MP2P-DSR protocol, where the control overhead is slightly higher than the original DSR protocol. We believe this minor drawback is acceptable considering the significant improvement in the video quality.

3. A common assumption of using more transmission nodes can increase the performance of MP2P framework does not hold true in wireless networks because the wireless medium is shared among all mobile devices within the network, and so too many concurrent connections to a single point may downgrade the overall network performance. Besides, using too many transmission nodes and video descriptions adds more control overhead to the network. Simulation study shows that using three transmission nodes to distribute two video descriptions is optimum. In spite of this, the largest difference in improvement is achieved when the number of transmission nodes is increased from one to two.
References


  http://www.isi.edu/nsnam/ns/


  http://trace.eas.asu.edu/yuv/index.html

  http://iphome.hhi.de/suehring/tml/download/old_jm/

[10] Multiple Description Coding.
    http://140.116.72.80/~smallko/ns2/MDC.htm

    http://www.tkn.tu-berlin.de/research/evalvid/


CHAPTER 4
MULTIPLE TREE VIDEO MULTICAST

4.1 Motivation

In this chapter, we tackle the issue of video multicast over mobile ad hoc networks by using the concept of diversity and Multiple Description Coding (MDC) scheme. This concept was first introduced to mobile ad hoc networks in [1] but a complete routing protocol was not proposed. This work was further investigated in [2], in which the authors proposed two mesh-based multiple tree multicast routing protocols, namely Serial MDTMR and Parallel MNTMR. The fundamental concept of this protocol is to distribute MD coded video descriptions separately along different multicast trees in order to increase the robustness of the video multicast system. In addition, these multicast trees should be highly disjoint to lower the loss correlation of these video descriptions. As such, both Serial MDTMR and Parallel MNTMR aim to create highly disjoint trees with minimum increase in the control overhead. Figure 4.1 (a) shows video multicast using the protocol in [2].

Multicast routing protocols can be generally divided into two categories: mesh-based and tree-based protocols. It is generally understood that mesh-based protocol is more robust due to the availability of alternate route. However, the control overhead is generally higher and the forwarding efficiency is lower than tree-based protocol. The Serial MDTMR and Parallel MNTMR protocols above are based on the mesh-based ODMRP protocol. These protocols are source-initiated and the multicast receivers do not participate in data forwarding. As such, a considerable level of node density is desired to ensure the good functionality of these protocols. Moreover, the Parallel MNTMR requires the network to be divided into two virtual topologies in order to execute the tree construction simultaneously at both topologies to construct two multicast trees. Apparently, this makes the requirement on node density more rigid. Besides, the source-initiated protocols require each source to start its own tree construction; therefore, the control overhead will increase rapidly as the number of sources increases. In light of this, there is a need to research on tree-based multiple tree multicast routing protocol because it may be a better alternative in some topologies and different sources can share the same multicast tree.
In this chapter, we provide an alternative to the mesh-based protocols proposed in [2] by introducing tree-based multiple tree multicast routing protocol. For this purpose, we provide the Multicast Ad Hoc On-demand Distance Vector (MAODV) routing protocol with the ability to construct multiple trees in a single routine [3]. MAODV is selected because it is a complete routing protocol that can support both unicast and multicast data. More importantly, each join request usually returns more than one reply, and so multiple tree construction can be carried out in a single routine. In addition, it is a receiver-initiated protocol that allows a receiver to join a multicast group at any time. The new routing protocol is called Multiple Tree Multicast Ad Hoc On-demand Distance Vector (MT-MAODV) routing protocol and Figure 4.1 (b) demonstrates video multicast using MT-MAODV.

![Diagram of multiple tree video multicast over mobile ad hoc networks using: (a) Mesh-based protocol; (b) Tree-based protocol.](image)

**Figure 4.1** Multiple tree video multicast over mobile ad hoc networks using: (a) Mesh-based protocol; (b) Tree-based protocol.
Chapter 4. Multiple Tree Video Multicast  
Section 4.2

4.2 Multicast Ad Hoc On-Demand Distance-Vector (MAODV) Routing Protocol

MAODV is an extension of AODV to support multicast in mobile ad hoc networks. It is a shared-tree-based protocol relying on the first node that joins the multicast group and serves as the group leader (GL) of this multicast group. The main responsibilities of a GL are to maintain a sequence number for the multicast group, and to ensure the multicast tree connectivity by sending out group hello message (GRPH) periodically. Besides, this protocol also requires each node to maintain three tables:

i. **Unicast Route (UR) table** that records the next hop for route to other destinations for unicast traffic.

ii. **Multicast Route (MR) table** that contains a list of next hops for the tree structure of each multicast group. Each next hop is associated with a link direction, upstream if the next hop is closer to the GL and downstream if otherwise. Every tree member must have one and only one upstream node, except the GL that has only downstream node. The node status, weather it is a member of the multicast group or just a forwarding node for the multicast tree, is also recorded.

iii. **Group Leader (GL) table** records the information obtained from the periodically broadcasted GRPH. It consists of all currently known multicast groups and the next hop toward the GL of each group.

Multicast tree construction is initiated by a node wishing to join a multicast group. If this node is currently a tree member of the multicast group, it only needs to change its status to group member. Otherwise, it broadcasts a route request packet with join flag (RREQ_J) to the network and set a waiting timer that expires after RREP_WAIT_TIME. However, if this node knows the next hop to the GL of this multicast group by checking its GL table, the RREQ_J packet may be unicast to the GL in its first trial. Each intermediate node is allowed to forward only one RREQ_J and the reverse route back to the requestor must be recorded in its UR table. When a non-duplicated RREQ_J arrives at a member of the multicast tree, a route reply packet with join flag (RREP_J) is created and unicast back to the requestor following the reverse route. Node forwarding RREP_J must cache a potential upstream node to the GL. Upon the timeout of the waiting timer, the requestor sends a multicast route activation packet with join flag (MACT_J) to activate the branch connecting it to the tree.

Figure 4.2 illustrates an example of multicast tree construction. The initial multicast tree is comprised of three nodes: node GL is the group leader, node X is a group member, and node A is a tree member. Node Y wishes to join the multicast group and does not have any information in its GL table, it broadcasts a RREQ_J to the network. This RREQ_J is replied by nodes GL and X, and so two RREP_J are returned to node Y, both with the same number of hop count to the
nearest tree member, *i.e.* 2 hops. However, the reply from X is four hops away from GL, whereas the reply from GL is only two hops away, therefore, node Y is connected to the multicast tree through nodes B and C. Figure 4.2(b) shows the resultant multicast tree.

**Figure 4.2** Multicast tree construction:
(a) Exchange of control packets; (b) Resultant multicast tree.

As mentioned briefly, the tree maintenance mechanism involves the broadcast of the GRPH message periodically by the GL. Upon receiving a fresh copy of GRPH, each node updates its GL table accordingly. Thereafter, a fresh GRPH is processed as follows, according to the node status.

i. A non-tree member sets the M-flag to indicate that this GRPH has been traveled off the multicast tree, and rebroadcasts the message.

ii. A tree or group member of the same GL updates the group information accordingly if and only if the GRPH is received from its upstream. Then, the GRPH is rebroadcast.

iii. A tree or group member with different GL discards the GRPH.

iv. GL which is not the originator of the GPRH also discards the GRPH.

Steps iii and iv above indicate that a tree partition is detected. Tree partition may occur if a node fails to reconnect itself to the multicast tree after a link breaks, and subsequently declares itself as the new GL. Since each multicast group is supposed to have only one GL, tree merge must be carried out to reconnect the parted trees and to select only one GL. Assuming GL₁ is the currently known GL of a tree node receiving GRPH originated by GL₂ and the address of GL₁ is smaller than the address of GL₂, the node that detects the tree partition sends a route request packet with repair flag (RREQ_R) to get permission from GL₁ to initiate a tree merge. If GL₁ has not given any permission to other node to execute tree merge, it returns a route reply packet with repair flag.
(RREP_R) to the requesting node. Upon receiving this packet, the requesting node unicasts a route request packet with both join and repair flags (RREQ_JR) to GL2. If GL1 is the node that detects the tree partition and has not given permission to any other node to execute tree merge, it can perform tree merge by sending a RREQ_JR directly to GL2. Upon receiving RREQ_JR, GL2 replies with route reply packet with both join and repair flags (RREP_JR) and it remains as the GL. In other words, GL1 surrenders its group leadership upon receiving RREP_JR.

Any node that obtains the latest information about the multicast tree must inform its downstream nodes using group hello packet with U flag (GRPH_U). A group member can revoke its membership when it wishes to leave the multicast group. This is done by sending an activation packet with prune flag (MACT_P) to its upstream. Furthermore, if this node is the GL, it must also send an activation packet with group leader flag (MACT_GL) to one of its downstream to instruct the selection of new GL. Besides, a tree member who is not a group member and has no downstream node can also prune itself by sending MACT_P to its upstream. Table 4.1 summarizes the functions of the control messages used in the MAODV protocol.

<table>
<thead>
<tr>
<th>Message</th>
<th>Flag</th>
<th>Direction</th>
<th>Functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>RREQ / RREP</td>
<td>-</td>
<td>Down ➔ Up</td>
<td>Unicast</td>
</tr>
<tr>
<td></td>
<td>J</td>
<td></td>
<td>Tree construction/reconstruction</td>
</tr>
<tr>
<td></td>
<td>R</td>
<td></td>
<td>Request tree merge approval</td>
</tr>
<tr>
<td></td>
<td>JR</td>
<td></td>
<td>Tree merge</td>
</tr>
<tr>
<td>MACT</td>
<td>J</td>
<td>Down ➔ Up</td>
<td>Tree construction/reconstruction</td>
</tr>
<tr>
<td></td>
<td>P</td>
<td></td>
<td>Self-pruning</td>
</tr>
<tr>
<td></td>
<td>GL</td>
<td>Up ➔ Down</td>
<td>Selection of new GL</td>
</tr>
<tr>
<td></td>
<td>P</td>
<td></td>
<td>Self-pruning &amp; selection of new GL</td>
</tr>
<tr>
<td>GRPH</td>
<td>-</td>
<td>Up ➔ Down</td>
<td>Periodical tree maintenance</td>
</tr>
<tr>
<td></td>
<td>U</td>
<td></td>
<td>Updating latest tree information</td>
</tr>
<tr>
<td>RERR</td>
<td>-</td>
<td>Down ➔ Up</td>
<td>Route/Tree repair</td>
</tr>
</tbody>
</table>

Down ➔ Up: received from downstream node  
Up ➔ Down: received from upstream node
4.3 Multiple Tree Multicast Ad Hoc On-demand Distance Vector Routing Protocol

4.3.1 The Core of MT-MAODV

As explained in the previous section, multicast tree construction in MAODV involves a three-way handshaking using three types of packets: RREQ_J, RREP_J and MACT_J. The RREQ_J is usually broadcast to the network or unicast to the GL depending on the availability of information about this multicast group in its GL table. For the case of broadcast, multiple replies may obtain because every tree member that receives the RREQ_J must reply with a RREP_J. Obviously, the potential of constructing multiple multicast tree is readily available. Therefore, the issue here is on how to construct multiple disjoint trees with minimum increase in the control overhead. In order to assist in simultaneous multiple tree construction, an 8-bit field at the control packets called tree field is introduced. More precisely, this chapter concentrates on constructing two disjoint trees only, namely Tree-1 and Tree-2. Therefore, the tree field can take three discrete values: 0 if the control packet is related to both multicast trees; 1 if it is related to Tree-1; and 2 if it is related to Tree-2. The abbreviations below are used in the algorithm to indicate the status of a mobile node (node_status):

- ON_GROUP : multicast group member;
- ON_TREE_0 : forwarding node of both Tree-1 and Tree-2;
- ON_TREE_1 : forwarding node of Tree-1;
- ON_TREE_2 : forwarding node of Tree-2;
- NOT_ON_TREE : not a tree member.

i. Multiple Tree Construction

The algorithm of multiple tree construction is given in Figure 4.3 [4, 5]. A node initiates a RREQ_J following step 1 when it wants to join a multicast group, unless if it is currently a tree member for both multicast trees (node_status == ON_TREE_0), where it only needs to change its status to ON_GROUP. First, it is important to set the value of join_tree field that indicates which tree or trees are to be connected. If the node has not connected to any tree yet (node_status == NOT_ON_TREE), join_tree is set to 0. It the node is currently a tree member of one of the trees, it only needs to connect to the other tree, for example an ON_TREE_1 node sets join_tree equals to 2, and vice versa. The value of join_tree is carried in the tree field of RREQ_J. There are two ways to send a RREQ_J. A node, in its first trial, can unicast the RREQ_J to the GL if the related information can be obtained from its GL table. Otherwise, the request is broadcast. Finally, a waiting timer is set to expire after RREP_WAIT_TIME.
Upon receiving a RREQ_J, a node forwards or replies to this request following the instruction in step 2. An intermediate node, which node_status is NOT_ON_TREE, is allowed to forward only the first RREQ_J received, and the immediate neighbor of the requestor must record its ID in the first_hop field. Besides, the reverse route back to the requestor is recorded in its UR table. The RREQ_J is then unicast or broadcast accordingly. Duplicate RREQ_J is not forwarded, but if the route it traversed is shorter than the previously forwarded one, the record in the UR table is overwritten.

If the node receiving RREQ_J is a multicast tree member, this request is replied. A multicast group member is allowed to reply to at most two RREQ_Js that arrive to it, as given in step 2.2. A fresh RREQ_J is replied with a RREP_J. The tree field is set to 0 because this reply is suitable for both Tree-1 and Tree-2. Besides, a req_tree that copies the value of tree in the RREQ_J is also introduced. The RREP_J is unicast to the prev_hop from which the RREQ_J is received. As mentioned earlier, two replies can be generated by a multicast group member; however, the second reply must have completely disjoint route with the first one. For this purpose, the first_hop of this RREQ_J is compared with the previously replied one. Since every intermediate node is allowed to forward only one RREQ_J, having different values of first_hop is the necessary and sufficient condition to determine two completely disjoint routes, as demonstrated briefly in Figure 4.4. Obviously, Route-1 and Route-2 or Route-1 and Route-3 are fully disjoint because they have different first_hop, but Route-2 and Route-3 are not disjoint because they have the same first_hop.

On the other hand, a tree member that is not a multicast group member replies to the first RREQ_J received only, as given in step 2.3. The same procedure above is followed, except the tree field is set accordingly to indicate which tree or tree this node belongs to. For example, an ON_TREE_1 node sets tree field to 1. The duplex (tree, req_tree) indicates the priority of RREP_J. For example, a RREP_J with (tree = 1, req_tree = 2) is low priority because the requestor wants to join Tree-1 but the reply provides a route to connect to Tree-2. This type of RREP_J is delayed by a small interval (SD) in order to allow more suitable RREP_J to be sent first. This type of reply is allowed because it may be the most suitable reply can be obtained in certain network topologies. Obviously, MT-MAODV sacrifices tree disjointedness in order to achieve full connectivity.

Step 3 gives instruction to handle a RREP_J received. For this purpose, each node is allowed to keep two possible upstream nodes (upstream_1, upstream_2), one for each multicast tree. An intermediate node, upon receiving a RREP_J, stores or updates one of the possible upstream nodes, based on the tree field. RREP_J (tree = 0) can be used to add or update either one of the two locations, whereas RREP_J (tree = 1) and RREP_J (tree = 2) can only be used to update their respective location, that is upstream_1 and upstream_2, respectively. Besides, the selection of best possible upstream node is based on the following parameters in the sequence of decreasing priority: hops_to_tree and hops_to_GL. Another rule of storing upstream nodes is that both locations should avoid
connecting to the same upstream nodes in order to achieve optimum disjointedness. When a RREP_J arrives at the destination, the same storing procedure as above is used. In addition to the above, the requestor also checks the originator of the RREP_J in order to encourage the divergence of these multicast trees by connecting to different multicast members.

Upon the timeout of the RREP waiting timer, the requestor executes the branch grafting to connect to the multicast trees, as given in step 4. This procedure depends on how many trees are to be connected and how many possible upstream nodes are available. If it is desired to join both trees (join_tree = 0) and two possible upstream nodes are available, two MACT_Js, each carrying different values in its tree field, are used to connect to these multicast trees. If there is only one possible upstream node is available, this node is used as upstream node to both multicast trees, in this case, MACT_J (tree = 0) is sent. Obviously, tree disjointedness is sacrificed to achieve full connectivity. If no possible upstream is obtained from this join request, the requestor restarts the join mechanism if the number of trials is below the maximum number of allowed trials. Otherwise, it can declare itself as the group leader (GL) of the multicast group. On the other hand, if the node currently has a connection to one multicast tree, only one possible upstream is needed to connect to the other tree. If a possible upstream exists, MACT_J (tree = join_tree) is sent to the potential upstream to activate the tree branch. The tree construction can be restarted if there is no potential upstream node recorded and the maximum number of allowed trials is not exceeded. Otherwise, this node copies the upstream from the currently connected tree to the other tree and sends a MACT_J (tree = join_tree) to this upstream to activate the branch. Again, it is important to emphasize that a node cannot declare itself as the group leader as long as it still has one connection to the group leader.

Step 5 shows the mechanism to handle a MACT_J. Upon receiving a MACT_J, a NOT_ON_TREE node adds an entry to its MR table and becomes a tree member by changing its status according to the tree field. The node_status is ON_TREE_0 if MACT_J (tree = 0) is received, ON_TREE_1 if MACT (tree = 1) is received, or ON_TREE_2 if MACT (tree = 2) is received. The prev_hop from which the MACT_J is received becomes its downstream node, and the possible upstream recorded earlier (upstream_1 or upstream_2) becomes its upstream node and the next destination of this MACT_J. A tree member, upon receiving a MACT_J, adds the prev_hop in its downstream list if this node is a member of the right tree, and the MACT_J will not be forwarded further. A special case is when the tree belongs to one tree, and the MACT_J received requires it to connect to the other tree. For example, in the case of an ON_TREE_1 node receiving MACT_J (tree = 2). When this happens, the tree node copies the branch from Tree-1 to Tree-2. More specifically, this node adds the prev_hop in its downstream list of Tree-2, copies the upstream node of Tree-1 to upstream node of Tree-2, forwards the MACT_J (tree = 2) to this upstream node, and change its status to ON_TREE_0. This mechanism copies the branch from one tree to the other tree.
1. **Send RREQ_J**(tree)
   1.1. Set join_tree to indicate which tree(s) to be joined.
   1.2. If (no information in GL table || not first trial), broadcast RREQ_J(tree = join_tree).
   1.3. Else if (join_tree != 0), unicast RREQ_J(tree = join_tree) to GL.
   1.4. Else,
      1.4.1. Unicast RREQ_J(tree = 1) to GL.
      1.4.2. Unicast RREQ_J(tree = 2) to GL after ARP_TIMEOUT.
   1.5. Set WAIT_RREP_timer.

2. **Receive RREQ_J**(tree, first_hop)
   2.1. If (node_status == NOT_ON_TREE),
      2.1.1. If (fresh RREQ_J),
         2.1.1.1. Record reverse route in UR table.
         2.1.1.2. If (first_hop == null), first_hop = node’s ID.
         2.1.1.3. Unicast/broadcast RREQ_J according.
      2.1.2. Else,
         2.1.2.1. If (shorter reverse route), replace the reverse route in UR table.
         2.1.2.2. Discard the RREQ_J.
   2.2. Else if (node_status == ON_GROUP),
      2.2.1. If (replied_RREQ < 1 || (replied_RREQ < 2 && replied_first_hop != first_hop)),
         2.2.1.1. req_tree = tree of RREQ_J.
         2.2.1.2. replied_first_hop = first_hop of RREQ_J.
         2.2.1.3. replied_RREQ++.
         2.2.1.4. Unicast RREP_J(tree = 0, req_tree) to prev_hop.
      2.2.2. Else, discard the RREQ_J.
   2.3 Else,
      2.3.1. If (fresh RREQ_J),
         2.3.1.1. req_tree = tree of RREQ_J.
         2.3.1.2. Set tree field according to node_status.
         2.3.1.3. If (tree != req_tree && tree != 0 && req_tree != 0), delay = SD.
         2.3.1.4. Unicast RREP_J(tree, req_tree, delay) to prev_hop.
      2.3.2. Else, discard the RREQ_J.

3. **Receive RREP_J**(tree)
   3.1. If (tree == 0),
      3.1.1. If (upstream_1 == null), upstream_1 = prev_hop and forward = 1.
      3.1.2. Else if (upstream_2 == null), upstream_2 = prev_hop and forward = 1.
      3.1.3. Else if (better upstream_1), upstream_1 = prev_hop and forward = 1.
      3.1.4. Else if (better upstream_2), upstream_2 = prev_hop and forward = 1.
      3.1.5. Else, forward = 0.
   3.2. Else,
      3.2.1. If (tree == 1 && (upstream_1 == null || better upstream_1),
         upstream_1 = prev_hop and forward = 1.
      3.2.2. Else if (tree == 2 && (upstream_2 == null || better upstream_2),
         upstream_2 = prev_hop and forward = 1.
      3.2.3 Else, forward = 0.
   3.3. If (forward == 1), unicast to next_hop according to UR table.
   3.4. Else, discard the RREP_J.

4. **Wait_RREP_timerExpired**
   4.1. If (join_tree == 0),
      4.1.1. If (upstream_1 != null && upstream_2 != null),
         4.1.1.1. Add upstream_1 as upstream in Tree-1.
         4.1.1.2. Unicast MACT_J(tree=1) to upstream_1.
4.1.1.3. Add upstream_2 as upstream in Tree-2.
4.1.1.4. Unicast MACT_J(tree=2) to upstream_2 after ARP_TIMEOUT.

4.1.2. Else if (upstream_1 != null || upstream_2 != null),
4.1.2.1. next_hop = upstream_1 or upstream_2, whichever is not null.
4.1.2.2. Add next_hop as upstream in both trees.
4.1.2.3. Unicast MACT_J(tree=0) to next_hop.

4.1.3. Else,
4.1.3.1. If (num_retries < max_allow_retries), go to step 1.
4.1.3.2. Else, declare itself as group leader.

4.2. Else,
4.2.1. If (upstream_1 != null || upstream_2 != null),
4.2.1.1. next_hop = upstream_1 or upstream_2, whichever is not null.
4.2.1.2. Add next_hop as upstream in the right tree.
4.2.1.3. Unicast MACT_J(tree=join_tree) to next_hop.

4.2.2. Else,
4.2.2.1. If (num_retries < max_allow_retries), go to step 1.
4.2.2.2. Else,
4.2.2.2.1. next_hop = upstream of currently connected tree.
4.2.2.2.2. Add next_hop as upstream in the other tree.
4.2.2.2.3. Unicast MACT_J(tree=join_tree) to next_hop.

5. Receive MACT_J(tree)
5.1. If (node_status == NOT_ON_TREE),
5.1.1. Change node_status according to tree.
5.1.2. Find next_hop from the recorded potential upstream (upstream_1 or upstream_2).
5.1.3. Add prev_hop as downstream and next_hop as upstream in the right tree(s).
5.1.4. Unicast MACT_J to next_hop.
5.2. Else if (node_status != NOT_ON_TREE && the right tree),
5.2.1. Add prev_hop as downstream in the right tree.
5.2.2. Discard MACT_J.
5.3. Else,
5.3.1. node_status = ON_TREE_0.
5.3.2. Add prev_hop as downstream in the right tree(s).
5.3.3. Copy upstream from Tree-1 to Tree-2, or otherwise, and next_hop = upstream.
5.3.4. Unicast MACT_J to next_hop.

Figure 4.3 MT-MAODV: multiple tree construction.

Figure 4.4 Route disjointedness.
An example of multiple tree construction using MT-MAODV is given here based on the topology shown in Figure 4.5. Assuming GL is the first node that joins a multicast group and therefore becomes the group leader, nodes A, B, C, D and E subsequently join this multicast group one after one. Also, it is assumed no information is recorded in the GL table of each node.

1. When node A initiates a join request, two replies, both with \((tree = 0)\), are obtained from GL through nodes y and z. Thus, two possible upstream nodes are available. Node A can arbitrarily select one of them to be the upstream node to Tree-1 (node y) and the other one to Tree-2 (node z). Two MACT_J, one with \((tree = 1)\) and the other with \((tree = 2)\), are used for branch grafting.

2. Next, node B initiates its join request and again, two replies are obtained: one from node A \((tree = 0)\) and the other one from node z \((tree = 2)\). Obviously, B must select node z as its upstream node to Tree-2, and so node A to Tree-1. Again, two MACT_J with different value in \(tree\) are involved.

3. The join request initiated by node C returns three replies: two from node A \((tree = 0)\) via nodes x and w, and one from node B \((tree = 0)\) via node v. Here, priority is given to connect to different tree or group members; therefore, either x or w is selected for one tree (Tree-1), and node v is used for the other tree (Tree-2). The same branch grafting as above is executed.

4. Node D has only one choice from its join request, which is node u. Since the priority is connectivity instead of tree disjointedness, node u must become the upstream node to both Tree-1 and Tree-2. In this case, only one MACT_J \((tree = 0)\) is used. When node u receives this MACT_J, it adds node D as its downstream list for both trees, and uses node B as its upstream node to both multicast trees as well. When MACT_J arrives at node B, node B adds node u as its downstream node of both trees.

5. Similar to node D, node E has only one option from its join request, \(i.e.\) via node z. To activate this branch, a MACT_J \((tree = 0)\) is sent to node z. Upon receiving this MACT_J, node z adds node D as its downstream list in both multicast tree, copies its upstream node from Tree-1 to Tree-2, and changes its status from ON_TREE_2 to ON_TREE_0. Then, it sends a MACT_J \((tree = 1)\) to node GL to activate the branch for second tree as well. When node GL receives this MACT_J, it adds z to its downstream list of Tree-2.

Table 4.2 summarizes the status of every node together with its next hops in each tree.
### Table 4.2 Multicast table for topology in Figure 4.5.

<table>
<thead>
<tr>
<th>Node</th>
<th>Status</th>
<th>Tree-1</th>
<th>Tree-2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Upstream</td>
<td>Downstream</td>
</tr>
<tr>
<td>GL</td>
<td>ON_GROUP</td>
<td>-</td>
<td>y, z</td>
</tr>
<tr>
<td>A</td>
<td>ON_GROUP</td>
<td>y</td>
<td>x, B</td>
</tr>
<tr>
<td>B</td>
<td>ON_GROUP</td>
<td>A</td>
<td>u</td>
</tr>
<tr>
<td>C</td>
<td>ON_GROUP</td>
<td>x</td>
<td>-</td>
</tr>
<tr>
<td>D</td>
<td>ON_GROUP</td>
<td>u</td>
<td>-</td>
</tr>
<tr>
<td>E</td>
<td>ON_GROUP</td>
<td>z</td>
<td>-</td>
</tr>
<tr>
<td>u</td>
<td>ON_TREE_0</td>
<td>B</td>
<td>D</td>
</tr>
<tr>
<td>v</td>
<td>ON_TREE_2</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>w</td>
<td>NOT_ON_TREE</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>x</td>
<td>ON_TREE_1</td>
<td>A</td>
<td>C</td>
</tr>
<tr>
<td>y</td>
<td>ON_TREE_1</td>
<td>GL</td>
<td>A</td>
</tr>
<tr>
<td>z</td>
<td>ON_TREE_0</td>
<td>GL</td>
<td>E</td>
</tr>
</tbody>
</table>

**Figure 4.5** Multiple tree construction.
ii. Multiple Tree Maintenance

The tree maintenance mechanism of MAODV is given in Section 4.2. In this part, only the modifications introduced in MT-MAODV are highlighted and the same definitions are used in this section. Specifically, GL1 is used to represent GL with smaller address whereas GL2 is the GL with larger address.

In the proposed MT-MAODV protocol, tree partition only occurs when both multicast trees are parted because a node can declare itself as GL only when it is not connected to both multicast trees. Partition of only one tree does not occur in MT-MAODV because if a node fails to reconnect to one tree but still connecting to the other tree, it can then send a MACT_J to make this branch to connect to both trees, as given by step 4.2.2.2 in Figure 4.3. Besides, MAODV allows any tree member to initiate a tree merge after getting permission from GL1. However, in MT-MAODV, only group members are allowed to initiate a tree merge because they are connected to both multicast trees. This step is important to ensure tree merge is started simultaneously for both multicast trees. A tree member that is not group member can inform the tree partition to its nearest group member in the upstream direction toward GL1 by using a GRPH message with repair flag (GRPH_R).

Similar to MAODV, a tree merge is executed using RREQ_JR, but in MT-MAODV, this RREQ_JR is broadcast in a control manner to GL2. When an intermediate node receives a RREQ_JR, it checks its GL table for the recorded GL for this multicast group. The RREQ_JR is re-broadcast if the recorded GL is GL2; otherwise, it is discarded quietly. Besides, if this node is an immediate neighbor of the requesting node, its ID is recorded in the first_hop field of the RREQ_JR. Again, this is done in order to allow GL2 to select highly disjoint branches to reconnect these multicast trees. Upon receiving a RREQ_JR by GL2, its information is recorded and a waiting timer is initiated. Upon the timeout of this waiting timer, two next hops are selected and two RREP_JRs are sent back to the requesting node. Again, these RREP_JRs carry different values in tree field.

Figure 4.6 demonstrates the merge mechanism. Figure 4.6(a) shows two parted trees, with GL1 and GL2 being the GL for these trees, respectively, where the address of GL1 is smaller than GL2. Assume node A notices a tree partition and initiates a tree merge after getting the permission from GL1, by exchanging RREQ_R and RREP_R via node y, as shown in Figure 4.6(b). Then, node A broadcasts a RREQ_JR to the network. Only nodes j and f rebroadcast the RREQ_JR, assuming they have the necessary information in its GL table, obtained from GRPH broadcasted by GL2. Nodes y and z discard the RREQ_JR received. Also, each intermediate node is allowed to rebroadcast the request once, and as a result, node f discards the second RREQ_JR received from node j. At GL2, three RREQ_JR are received: from node C (tree = 2), node u (tree = 1), and node v (tree = 2). Since the second and third options have the same first_hop,
which is node $j$. GL$_2$ selects the first and the second options and replies with two RREP_JRs with the right value for `tree` field. These RREP_JRs are unicast along the backward route toward node A. Upon receiving a RREP_JR, node A informs the changes by using a GRPH_U packet. Figure 4.6(c) shows the repaired topology for the multicast trees.

**Figure 4.6** Multiple tree merge:
(a) Parted tree; (b) Exchange of control packets; (c) Merged tree.
A tree branch becomes invalid when the link connecting two nodes is broken. When this happens, both upstream and downstream nodes delete this link from their list of next hops and the downstream node will initiate a tree repair to reconnect to the multicast tree. This process is the same as the tree construction, which involves the exchange of RREQ_J, RREP_J and MACT_J. Specifically, the downstream node broadcasts a RREQ_J with destination address set to the address of the multicast group. Again, the tree field is set accordingly to indicate which tree is to be repaired. Besides, in order to avoid nodes with outdated information to reply to this join request, the previous hops count to the GL is recorded in the RREQ_J, and only nodes with hops count less than or equal to the indicate value can reply to RREQ_J.

**ii. Tree Optimization**

In order to optimize the multicast trees constructed, a tree optimization mechanism is introduced in MT-MAODV to ensure these trees are optimally connected. This mechanism is especially important after the tree merge explained above because the trees merge mechanism do not always return the most suitable options for the reconnection. This mechanism is associated with the periodic GRPH message. When a tree node receives a fresh GRPH not from its upstream, it does not discard it immediately. Instead, it can replace its current upstream node to the node from which the GRPH is received if all the following conditions are fulfilled:

i. The GRPH is originated by its known GL.

ii. The potential upstream is a tree member of the right tree, but not the GL.

iii. The hops count to GL is shorter than its current hops count to GL.

In order to replace the upstream node, the node must send a MACT_P to the old upstream node to remove itself from its downstream list. After ARP_TIMEOUT, a MACT_J packet is sent to its new upstream node to inform this node to add it in its downstream node.

An example of this mechanism is given below, by referring to the example in the previous section, with respect to Figure 4.6(b). If node B has a virtual connection with node f and upon receiving a GRPH \((tree = 2)\) forwarded by node f, which is not its upstream node in Tree-2. Obviously, all the three conditions above are fulfilled, and as a result, node B should choose node f as its new upstream node to Tree-2. For this purpose, node B sends a MACT_P \((tree = 2)\) to node x to break its connection with node x, and then sends a MACT_J \((tree = 2)\) to node f to activate the new branch. Besides, node x, which is not a group member, becomes a tree node without a downstream node; consequently, it sends another MACT_P \((tree = 2)\) to GL1 to prune itself and changes its status to NOT_ON_TREE. The resultant tree topology is given in Figure 4.7.
4.3.2 Video Multicast with MT-MAODV

Similar with many implementations of MD video multicast, a video source must have a connection to both multicast trees in order to use this framework. For this purpose, the video source is assumed to join the multicast group following the mechanism given in Section 4.3.1 prior to the video transmission. Besides, the video is coded using the MDC scheme and a traffic allocator is used to split the traffic accordingly, one video description to each multicast tree. This is done by allowing the video application to set a tree field in the packet header to indicate through which tree this packet is to be transmitted. Upon receiving a video packet, a node processes the packet according to its node status, for example if video description-1 is sent along Tree-1, a member with at least one downstream node along Tree-1 must re-broadcast the data. Otherwise, this video packet is not forwarded. By using this concept, the received video quality can maintain at an acceptable quality as long as packet loss does not occur simultaneously in both multicast trees, this is assisted by the use of optimally disjoint trees. Figure 4.8 summarizes the actions of video packet forwarding according to the status of the node.
Figure 4.8 Video packet forwarding algorithm.
4.4 Performance Evaluation

4.4.1 Simulation Settings

For the evaluation purpose, the multiple tree video multicast framework and the MT-MAODV routing protocol were added in NS-2 [6, 7, 8]. Subsequently, simulation study was carried out to compare the following three cases: (C1) Video Multicast with Single Description Coding (SDC) scheme and MAODV, (C2) Video Multicast with MDC and MAODV, and (C3) Video Multicast with MDC and MT-MAODV. We used homogeneous network with channel bandwidth of 5.5 Mbps and transmission range of 250 meters. UDP was used in the transport layer. The simulation topology contained 60 nodes within an area of 1500 by 600 square meters. The node movement was modeled using the enhanced Random Waypoint Model [9]. The minimum traveling speed was set to 0.1 m/s and the maximum speed was varied between 2.5 m/s and 15.0 m/s to model different levels of node mobility.

One video source and ten multicast receivers were selected randomly and the multicast group was created at the beginning of the simulation. The identity of these video source and receivers were recorded so that the same nodes were used for each case to maintain the fairness of the comparison study. The frame rate of the video application used was 8 frames per second, and the data rates were set to 48 kbps and 64 kbps for SDC and MDC, respectively [10]. For SDC, standard codec was used and the MDC codec used in this chapter is based on a simplified model of matching pursuits multiple description coding [11]. The data rate for MDC was about 30% higher than SDC due to the additional overhead created in MDC [12]. The EvalVid toolset was used to determine the frame size [13]. For SDC, each frame was transmitted in a single packet; whereas for MDC, each video frame was divided into two descriptions and each description was contained in a single packet for transmission. At each receiver, a 0.5 seconds play-out buffer was used to absorb the jitter and delay in frames re-sequencing. A frame received after the play-out deadline was discarded. Each simulation ran for 200 s and the results were average over 30 scenarios. For these scenarios, the average node degree (AND), as defined previously in Section 3.3.1, is between 9 and 10. This means each node has 9 to 10 neighboring nodes at most the time.

The simulation study aims to compare the performance of the proposed MT-MAODV and the original MAODV protocols in terms of providing better quality best-effort video. One key factor that determines the success of MT-MAODV is the disjointedness of the multicast trees constructed for the same multicast group. In light of this, the tree similarity is presented in Figure 4.9 for reference in the discussion section. Two parameters are dedicated to measure tree similarity:
i) $S_{t1}$ is the ratio of the number of common forwarding nodes of both multicast trees to the number of forwarding nodes of tree with smaller number of forwarding nodes.

ii) $S_{t2}$ is the same as $S_{t1}$ but multicast group members are excluded in the calculation.

For example, by referring to the topology in Figure 4.5 and assume the data is send by GL to other multicast group members,

i) Calculation of $S_{t1}$,

The forwarding nodes of Tree-1 are: A, B, u, x, y and z. (6 nodes)
The forwarding nodes of Tree-2 are: B, u, v and z. (4 nodes)
Common forwarding nodes are: B, u and z. (3 nodes)
\[ S_{t1} = \frac{3}{4} = 0.75 \]

i) Calculation of $S_{t2}$, (without considering the multicast group members)

The forwarding nodes of Tree-1 are: u, x, y and z. (4 nodes)
The forwarding nodes of Tree-2 are: u, v and z. (3 nodes)
Common forwarding nodes are: u and x. (2 nodes)
\[ S_{t2} = \frac{2}{3} = 0.67 \]

From Figure 4.9, the tree similarity $S_{t1}$ is approximately 0.1 and $S_{t2}$ is slightly lower. From this result, it is fair to conclude that the proposed MT-MAODV successfully creates two highly disjoint multicast trees connecting the same multicast group members. Generally, lower tree similarity indicates lower correlated packet loss across these multicast trees and consequently, higher video quality can be guaranteed. The importance of highly disjoint multicast trees is demonstrated in the next section.

![Figure 4.9 Tree similarities in the MT-MAODV protocol.](image-url)
4.4.2 Performance Parameters

The following metrics are used for performance evaluation.

(a) **Percentage of Bad Frames** indicates the quantity of video frames with bad quality. A frame is undecodable when the packet containing this frame is lost in transmission or its preceding frame is undecodable if this frame is not prediction independently during the video encoding stage. In SDC a frame is considered as bad quality if it is undecodable whereas in MDC a frame is considered as bad quality when both video descriptions are undecodable. If either one of the video descriptions is received, a frame with acceptable quality is produced.

(b) **Number of Interruptions** measures how often distortions occur during the video viewing at the multicast receivers. An interruption is observed when a number of bad frames are received consecutively.

(c) **Average Hops Count** is the average hops traversed by a video packet. This parameter indicates the influence of MT-MAODV in terms of additional hop distance traversed. Considering that longer distance traversed is equivalent to longer time elapsed in transmission, this parameter also indicates additional delay imposed.

(d) **Average End-to-end Delay** is the average time traversed by a video packet from the sender to the receiver. It is measured in milliseconds.

(e) **Normalized Routing Load** is the number of control packets forwarded at each node divided by the number of video frames received at all receivers. In this calculation, the denominator used is the number of video frames instead of the number of video packets in order to maintain the fairness because both SDC and MDC have the same number of video frames but the number of video packets is doubled in MDC. This parameter evaluates the effectiveness of the multicast routing protocol.

(f) **Forwarding Efficiency** is calculated by dividing the number of video packet forwarded at each node to the total number of video packets received at the receivers. A large value of this parameter indicates low forwarding efficiency. Generally, this parameter is more important than normalized routing load because the total number of video packets is much larger than the control packets. A large value of this parameter indicates low forwarding efficiency.
4.4.3 Results and Discussion

From Figure 4.10, an improvement of about 2% is observed when MDC is used, by comparing cases C1 and C2. Further improvement of 2% is observed in case C3 with MT-MAODV. As expected, the MDC scheme makes the video less subtle to the impact of packet losses because a bad frame is created only when both video descriptions are undecodable. Furthermore, the use of disjoint trees guarantees lower correlated packet losses of these video descriptions. The disjointedness of the multicast trees is presented in the previous section. Besides, the use of disjoint trees also prevents connection from a multicast receiver to both multicast trees breaks at the same time, and so resulting in fewer bad frames.

A more insightful observation is presented in Figure 4.11 that shows the number of interruptions observed during video viewing. Clearly, case C3 has the lowest occurrence of distortion, which is always below 20 interruptions per video session. Distortions are observed frequently in C1, which is about 50 interruptions per video session. C2 performs moderately with 30 – 50 interruptions per video session. Up to this point, it is fair mentioning that multiple tree video multicast, especially with the use of MT-MAODV protocol, contributes to improvement in video quality over mobile ad hoc networks.

![Figure 4.10 Simulation result: Percentage of bad frames](image-url)
Figure 4.11 Simulation result: Number of interruptions

Figure 4.12 gives the distribution of video frames according to their quality. Since MDC introduces frames with acceptable quality which does not occur in SDC and the quantity of bad frames is explained in the previous paragraph, the focus here is to compare C2 and C3. The ultimate objective of video transmission is to obtain video at best possible quality, but under unavoidable circumstance, this cannot be achieved, especially in lossy networks. However, it is still crucial to maintain the quantity of good quality frames at highest possible level. Obviously, C3 has better performance than C2 in terms of the amount of good frames. Considering that C3 secures more good frames and less bad frames than C2, it is fair to mention that MT-MAODV is a good option to be used in the multiple tree video multicast framework.

Figure 4.12 Simulation result: Percentage of each type of frames
Next, the performance of the proposed protocol is evaluated from the network perspective, in terms of average hops traversed by each packet, normalized routing load, and forwarding efficiency. Figure 4.13 shows the average number of hops traversed by each video packet. As expected, the results show that case C3 has a small increase in this parameter because in the original MAODV protocol, only the shortest route is selected to connect to the tree, and so the video packets always traverse the shortest route. On the other hand, MT-MOADV selects two shortest routes to connect to different trees, which is averagely longer than the shortest route in the original MAODV. Fortunately, the increase is small, which is less than one hop. This also indicates that most of the time there are more than one shortest route is available for a node to connect to the multicast group. This observation further supports the feasibility of using more than one multicast tree for video distribution. A conservative estimation on the one-hop traversal time for a packet including queuing delays, interrupt processing time and transfer time is about 10 – 30 ms. In order to verify this, the average end-to-end delay of video packets is given in Figure 4.14. As seen, MT-MAODV has the highest end-to-end delay, which is about 60ms. As compared to MAODV in case C2, an increase of about 20 ms or 50% is observed. However, we believe the impact of this increase is small because it can be easily absorbed by the play-out buffer at the receivers, which is 500 ms in this simulation.

![Figure 4.13 Simulation result: Average hops count](image-url)
Figure 4.14 Simulation result: Average end-to-end delay

Figure 4.15 shows the results of normalized routing load. Generally, C1 and C2 have almost the same amount of routing control packets because the same routing protocol is used in both cases. However, C2 has slightly larger normalized routing load because this parameter is evaluated based on the number of frames successfully received. For the MDC scheme, a frame is divided into two descriptions and when only one description is received, an acceptable frame is only half-weighted. As expected, C3 has the highest normalized routing load because more control packets are needed to construct and to maintain two multicast trees. However, the difference is relatively small because the number of successfully decoded frames also increases; therefore, the positive and negative impacts are offset and the resultant output is only a small increase.

Figure 4.15 Simulation result: Normalized routing load
Finally, the forwarding efficiency of the multicast trees is given in Figure 4.16. It is important to note that cases C2 and C3 have twice as many video packets as in C1. Due to this reason, C1 has the worst forwarding efficiency because with smaller value of both numerator and denominator, the effect of the denominator becomes more significant and greater value is obtained. With this in mind, the focus here is to compare C2 and C3. From the simulation results, both MAODV and MT-MAODV have almost the same forwarding efficiency with MT-MAODV forwards slightly more video packets at intermediate nodes per video packet arrived at the receiver. Considering the fact that MT-MAODV has slightly larger hops count as presented in Figure 4.13, this observation is reasonable although the number of video packets received at the receivers is higher.

Figure 4.16 Simulation result: Forwarding efficiency
4.5 Conclusions

In this chapter, we studied the concept of multiple tree video multicast over mobile ad hoc networks. For this purpose, we introduced Multiple Tree Multicast Ad Hoc On-demand Distance Vector (MT-MAODV) routing protocol. Particularly, the tree construction mechanism was modified accordingly to allow a node wishing to join a multicast group to connect to the multicast group via two disjoint routes, and preferably to two different tree members. The final structure can be viewed as two highly disjoint multicast trees connecting members of the same multicast group. Similar to other frameworks that utilize the concept of diversities, the MDC scheme was used for video coding in order to create independent and discrete video descriptions. We implemented the MT-MAODV protocol in NS-2 and verified its performance via extensive simulation. From the simulation study, multiple tree video multicast was proven to outperform the conventional single tree video multicast. Furthermore, the use of disjoint trees in MT-MAODV ensures the packet losses across the multicast tree are incoherent in order to fully utilize the strengths of the MDC scheme.

A comparison between MT-MAODV and those previously proposed mesh-based multiple tree multicast routing protocols, with respect to [2], is not included in this chapter because a comparison between them cannot be carried out without making some assumptions that are unfair to either one party. A mesh-based protocol provides higher robustness with the availability of alternates routes, whereas a tree-based protocol such as MT-MAODV has less stringent demand on node density and more flexible in allowing the receiver to join the multicast group at any time. A comprehensive comparison study on the pros and cons of tree-based and mesh-based protocols can be obtained in [14]. Generally, the selection of a suitable multicast routing protocol depends on many factors. In light with this, the main contribution of this chapter is to provide a new tree-based multiple tree multicast routing protocol suitable for the multiple tree video multicast framework.
Chapter 4. Multiple Tree Video Multicast

References


CHAPTER 5
CONCLUSIONS AND FUTURE WORK

5.1 Conclusions

In this dissertation, the concept of diversities has been implemented in two separate frameworks to handle video transmission over mobile ad hoc networks. In the first approach, multipoint-to-point transmission that combines both path and server diversities has been introduced to provide a more fault-tolerant video transmission system to deliver video to a receiver. The main objective is to provide best-effort uninterrupted video to a receiver as long as this receiver is still connected to any one of the video servers. This feature is important considering the rapid changes in the topology of the mobile ad hoc networks. This framework is comprised of a source searching mechanism to locate the potential video sources, a routing protocol to discover highly-disjoint routes for these video sources, and a playout buffer at the receiver to absorb the difference in the packets arriving time. In the second approach, the multiple tree video multicast concept has been further accomplished with a tree-based multiple tree routing protocol called Multiple Tree Multicast Ad Hoc On-demand Distance Vector (MT-MAODV) routing protocol, which aims to construct two multicast trees connecting the same multicast group in a single routine. The core of this approach is to distribute independent coded video along different trees to ensure minimum video quality at the multicast receivers at most of the time.

The above frameworks have been evaluated extensively by simulation. In both frameworks, the simulation results show that the implementation of path and server diversities not only significantly improves the overall video quality, but also reduces and shortens the distortions during video viewing. This improvement is not only contributed by the use of the MDC scheme alone because using MDC increases the overall workload and this may lead to a decrease in the overall performance. The strengths of MDC are fully utilized by the use of path and server diversities so that the MD coded video descriptions are delivered separately along independent routes to minimize the loss correlation of these video descriptions. Considering the rapid changes in the topology of mobile ad hoc networks, implementing path and server diversities also provides better fault tolerance for the video transmission system and so leads to a better video quality. There are two main drawbacks in the proposed frameworks. First, the use of MDC introduces additional workload to the network due to the additional overhead
added to the coded video for better error-resilient. Second, the concept of diversities imposes the need to discover alternative route or additional video servers, which is usually associated with the exchange of more control packets.

There are other considerations in designing the proposed frameworks due to the nature of the mobile ad hoc networks. As mentioned earlier, the use of MDC leads to lower compression efficiency and so a larger bandwidth is required to host the resultant video descriptions. In addition, the amount of the additional overhead is directly proportional to the number of video descriptions created. Consequently, it is generally not recommended to create too many video descriptions for economical reason. This consideration is especially important in mobile ad hoc networks due to the need to share the wireless transmission medium because an excessive workload may degrade the network performance. Besides, having too many connections to a receiver is not a good approach because the wireless medium is shared among all the mobile devices in the mobile ad hoc networks. Assuming N is the number of video descriptions used as well as the number of transmission nodes in multipoint-to-point framework or the number of multicast trees in the multiple tree video multicast framework, increasing N from one to two gives the maximum increase in the performance. Further increase in N may give only a small additional improvement or degrade the performance due to excessive overhead.

In summary, the use of path and server diversities can give a remarkable improvement on the video quality due to the existence of alternative path and/or server to ensure uninterrupted service at most of the time. This improvement is associated with a small increase in the utilization of the network resources. With this in mind, it is important to design the system carefully so that the negative impacts are minimized. In other words, the tradeoffs between performance and overhead should be taken into consideration when designing the system.
5.2 Future Work

In this section we propose some open issues for further investigation. In Chapter 2, we have introduced a source searching mechanism that enables the receiver to locate multiple video senders, and selects the most suitable ones to obtain the video signal. Considering that the network topology changes regularly and the suitability of these video senders may vary over time, a more robust source searching mechanism may provide video transmission with better quality. This mechanism should allow the receiver to detect more suitable video senders from time to time, and switch the video transmission from the current video senders to the new ones. Besides, the MP2P-DSR protocol can be further enhanced to reduce the overall control overhead.

Generally, there are four types of model for video transmission: point-to-point, point-to-multipoint, multipoint-to-point and multipoint-to-multipoint. The first model was well-studied in many literatures while model two and three have been studied in this dissertation. From the first part of this dissertation, we observe the potential of implementing server and path diversities to provide reliable video transmission. In light of this, using multiple video senders with multiple trees seems to be an efficient means of providing good quality video multicast. One major issue to be tackled is the synchronization between the video senders. This issue is more complicated than the unicast case because it involves video transmission to more than one receiver. With MT-MAODV proposed in Chapter 4, this synchronization job can be dedicated to the group leader because it is the only node whose identity is known by all the group and tree members of the multicast group.

Figure 5.1 Multiserver multiple tree video multicast over mobile ad hoc.
Another issue worth further investigation is the security issue. The security of communication in ad hoc wireless networks is very important, but very complicated due to the lack of centralized coordination and shared wireless medium. These make the networks highly vulnerable to security attacks. It is important to propose an efficiency mechanism to identify various types of attacks and to isolate the problematic node or nodes. Although some secure routing protocols have been proposed to date to combat the security attacks, none of these takes into consideration the stringent requirements of video applications.

With the advances in wireless technologies, it would be interesting to explore how these technologies can help in improving the video quality. The newly introduced IEEE802.11n standard that utilizes the Multi-Input Multi-Out (MIMO) technology is estimated to achieve a theoretical data rate of 540 Mbps. With this data rate, transmission of high-resolution video over wireless ad hoc networks can be implemented provided if a reliable packet delivery scheme is available. Another open issue is video transmission over heterogeneous networks. Considering that there are a few standards for wireless communication are available, such as IEEE802.16 (WiMax), IEEE 802.11 (Wi-Fi), IEEE802.20 (MBWA) and Ultra-Wide-Band Network (UWB), it is a timely research topic to propose a feasible framework for video transmission over hybrid and heterogeneous wireless networks.
LIST OF PUBLICATIONS

A. JOURNALS


B. CONFERENCES


ACKNOWLEDGMENTS

I would like to take this opportunity to express my heartfelt appreciation to many people who made this dissertation possible.

I am deeply indebted to my advisor, Professor Hiroshi Ishii, who provided me with invaluable and unlimited support to complete my studies and gave me the freedom to determine the direction of my research. I will never forget his continuous encouragement and extensive guidance throughout my years at Tokai University, especially when I faced difficulties in my work. I am glad to have been given this opportunity to work in the Ishii Laboratory for the past three years and a half.

Of equally importance is the financial support provided by the Hitachi Scholarship Foundation. I thank the foundation for awarding me a scholarship to pursue my studies in Japan and funding my research activities. I would also like to dedicate my special gratitude to Mr. Tsutomu Gomibuchi, Mr. Kazuyuki Miyanaga, Mr. Mikio Homma, Mr. Michio Sasamori, Mr. Hisao Miyake and Ms. Maki Nunokami from the Hitachi Scholarship Foundation. Besides, I also acknowledge the financial support on this project by the Japan’s Ministry of Internal Affairs and Communications under the Strategic Information and Communications R&D Promotion Program (SCOPE), and the Japan Society for the Promotion of Science (JSPS) under the Grant-In-Aid for Scientific Research (C) (no: 19500064).

Next, thanks are due to the judgment panel of this dissertation: Professor Hiroaki Kikuchi, Professor Hidekazu Tsuji, Professor Osamu Mikami, and Professor Kiyoshi Nosu. Their constructive comments and recommendations have made this dissertation more tangible and readable.

It was a pleasant experience working in the Ishii Laboratory. I thank all the members of Ishii Laboratory for being helpful and kind to me, especially to Mr. Yoshimasa Sajima, Mr. Masahiro Yamamoto, Mr. Masahiro Nakaiwa, Mr. Hideaki Sato, Mr. Naoyuki Hirata and Mr. Keisuke Utsu. Besides, there are other foreign students in Tokai that have helped me from time to time; they are Ms. Tomoko Yamaki from Dominican Republic, Dr. Marizan Mubin from Malaysia, Mr. Khamphong Khongsomboon from Laos, and Ms. Sutheera Puntheeranurak from Thailand.

To all my dearest friends back in Malaysia, thanks for being with me, sharing my emotion patiently, and giving me your sincerest advices. Without your supports and encouragements, I would not have gone this far. Last but most importantly, I would like to dedicate my achievement to my beloved family, especially my parents, for their constant love, understanding and support, and making my life more meaningful.

Chee Onn
November 2007