國立交通大學

資訊科學與工程研究所

博士論文

一個隨意網路上跨層設計的通用同儕平台

A generic peer-to-peer platform based on cross-layer design in ad-hoc network

研 究 生：郭俊利

指導教授：陳耀宗 教授

中華民國一百零二年六月

A Generic Peer-to-Peer Platform Based on Cross-Layer Design in Ad-Hoc Networks

研 究 生：郭俊利

指導教授：陳耀宗

國立交通大學

資訊科學與工程研究所

博士論文

中華民國一百零二年六月

A Generic Peer-to-Peer Platform Based on Cross-Layer Design in Ad-Hoc Networks

Student: Jun-Li Kuo

Advisor: Yaw-Chung Chen

A Dissertation Submitted to

Institute of Computer Science and Engineering

College of Computer Science

National Chiao Tung University

In partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy

in

Computer Science

Hsinchu, Taiwan

June 2013

一個隨意網路上跨層設計的通用同儕平台

學生: 郭俊利　　　　　　指導教授: 陳耀宗 博士

國立交通大學資訊科學與工程研究所

摘要

「行動隨意網路」不用任何網路基礎建設佈建，即可以發展隨取需求與自成結構的網路，因此行動隨意網路適合用於軍事、緊急救援、警消等艱困任務的環境，隨著行動隨意網路的發展，「同儕計算」已被論證為一種管理行動無線裝置的有效方法，這類的解法或網路架構稱為「行動隨意網路上的同儕管理系統」。然而截至目前為止，整合同儕應用服務與行動環境的網路架構仍然存在一些問題，例如：高度封包遺失率、缺乏跨層修復能力、低弱的服務持續性，這些缺點都會對這類網路架構造成不良的影響。而且目前的系統都只適合少數的客戶需要特定的服務，所以我們設計一個跨層演算法以解決上述的問題，並實作一個同儕計算平台來整合檔案分享與影音串流的應用服務，因為同時改善緩衝機制、排程演算法、跨層結構，這個同儕計算平台可支援高度擴充性與各式各樣的同儕軟體應用。

在這個研究中，跨層設計整合了同儕計算的跨層與路由連線的網路層，並且結合同儕轉傳路線與連點路由途徑，如此一來，我們的方法同時擁有路由表與指向表的資訊，加上使用第六版網際網路協定的行動路由特性，能夠達成跨層的鄰似性，這種鄰似性可以提高封包傳輸的成功率、跨層修復能力、服務的持續性。由於上面所提及的優點，我們稱此方法為「通用同儕行動隨意網路系統」，能夠降低封包傳輸的失敗率與網路流量負擔，進而提升服務品質。

首先，我們研究分析很多行動隨意網路上同儕管理系統的演算法，然後設計出所提

的跨層體制，這是非常適合高度擴增性和多群體的使用者。第二，我們設計緩衝機制、排程演算法、跨層結構等元件，讓各種同儕分享軟體都能適用。第三，我們推導數學模組和理論公式來量化系統的延遲時間和負載量，用理論的分析評估所提方法效能的極限。最後，我們實作所提方法在網路模擬器上，論證模組與公式的正確性，經由模擬結果討論系統效能的改善程度，必且依照模擬結果調校參數，使得系統最佳化。

總而言之，我們提出跨層式的行動隨意網路搭載同儕管理協定，也就是通用同儕行動隨意網路系統，所有同儕分享軟體皆能適用。這套協定不僅能增進封包傳輸的成功率與跨層的修復能力，還能提高服務的持續性，使用很低的成本即能夠發展行動多媒體分享服務。

A generic peer-to-peer platform based on cross-layer design in ad-hoc networks

Student: Jun-Li Kuo Advisor: Dr. Yaw-Chung Chen

Institute of Computer Science and Information Engineering

National Chiao Tung University

Abstract

Mobile ad hoc network (MANET) can be employed to develop on-demand self-organized network without the support of infrastructure. Therefore, MANET is suitable for certain hard task forces, such as military, emergency rescue, and police. With the development of MANET, P2P solution has been demonstrated to be an efficient technique for the management of wireless ad hoc nodes. P2P-MANET (P2P over MANET) has appeared in this kind of network architecture. However, there are several problems in the integration of P2P applications and mobile environment so far. High packet loss rate, low recovery, and low continuity degrade the performance of P2P-MANET. Currently the existing system is only limited in a single requirement for few particular users. Therefore, we design a cross-layer algorithm to solve the above mentioned problems, and implement a P2P platform to integrate file sharing and media streaming. We improve the methods for buffer, scheduling, and overlay to establish a P2P platform for the high scalability and the support of various P2P applications.

In this research, the cross-layer design integrates P2P overlay and routing layer, and combines P2P forwarding path and hop-by-hop routing path, such that there is the information of routing table in finger table. The characteristics of IPv6 mobile routing is used to achieve overlay proximity, which heightens packet delivery rate, recovery ability, and playback continuity. Due to above characteristics, we call the proposed scheme Generic P2P-MANET

System (GPMS). GPMS can reduce both packet failure rate and traffic overhead, which in turn improve quality of service.

First, we survey P2P-MANET algorithms, and design the cross-layer scheme, which is suitable for high scalability and large group of users. Second, we design the buffer, scheduling, and overlay, which is adaptable for all P2P cooperation. Third, we derive the mathematical model or theoretical formula to evaluate latency and overhead. Therefore, we can evaluate the limitation and basis of the performance. Finally, we use a network simulator, OMNet++, to demonstrate the correctness of model and formula. And we configure the parameters and optimize the proposed algorithms depending on the simulation results, which are used to discuss the performance and improvement.

In summary, we propose a cross-layer P2P-MANET protocol, GPMS, which is suitable for all P2P content sharing applications. The proposed scheme can improve packet delivery rate, recovery ability, and playback continuity to develop a media sharing service on mobile environment with affordable cost.

Acknowledgement

First, I thank my parents. They always encourage and support me for study. They give me a good family life and let me concentrate my attention on research. They tolerate my mistakes, forgive me, and believe for my successful graduation of Ph.D. degree. My father's consideration, my mother's love, and my sister's exhortation all help to complete the paper.

Second, I really thank my advisor, Prof. Yaw-Chung Chen. Because he teaches and guides me for the Ph.D. academia, I enjoy and learn much in my NTCU school life. He gave me valuable advice and encouraged me when I had trouble in research, he recommends me for some ways to career now, and he will help me in the future. He is a good teacher in my mind, and I see him as the great model, so I hope that I will be the successful follower of him.

Third, I sincerely thank my partner, Dr. Chen-Hua Shih. Because he always helps and encourages me in my research, I not only increase the network knowledge but also learn the skills of study and teach. Because he studied in Multimedia Communications Laboratory, I didn't feel alone in my student life and talked about everything in all ways. We have studied together for twelve years. He is not only a friend but also a teacher for me. He exists, the paper exists.

Finally, I specially express my thanks to my girl friend, Chia-Ning Lee. She gives me smile and stands me for every useful thing. She understands my ambition of research and is willing to accompany me and wait for me. She always stands by my side to bring me happiness. Her consolation and encouragement surrounds me every time and everywhere. I really treasure her love in my life. Without her love, I cannot finish my Ph.D degree.

Table of Contents

List of Figures

List of Tables

1. Introduction

As technology for high quality multimedia application prevails, everyone has been able to enjoy audiovisual communications and high definition video content through the wireless network. Mobile devices have been gaining popularity along with wireless multimedia services. Mobile access to the Internet via WiFi, WiMax, 3G, or LTE data network from the fixed infrastructure is prevalent. However, generally the traditional cellular-based architecture and infrastructure-based communication are not always adequate to satisfy users' requirements [1]. The limited bandwidth of the wide area cellular networks cannot offer good service quality. In many situations, the communications between mobile hosts neither rely on any fixed infrastructure nor the access to Internet. For example, the high cost or long setup time of a system installation may be unacceptable in the disaster areas and battle fields. It motivates mobile ad hoc network (MANET)[1] which can be expected to operate in the infrastructure-less environment [2].

Wireless local area network (WLAN) provides both Internet accessibility and ad hoc availability for mobile devices. For example, WiFi and WiMax can support network capability without infrastructure in the ad hoc mode [3]. The ad hoc environment features the short delay of an installation without infrastructure cost. The ad hoc solution provides a private network to support on-demand multimedia sharing. The advances in convenient access technologies and high bandwidth availability generate new opportunities to deliver high quality, on-demand, and interactive multimedia applications.

The words *ad hoc* imply that the network is formed in a spontaneous manner to meet an immediate demand. In MANET, two mobile nodes can communicate with each other through other intermediate nodes over wireless infrastructure-less network without the centralized

---

[1] The creation in 1997 of an IETF working group (called MANET WG) triggered an era of research in these networks.

administration.[2] The nodes organize a cooperative network to establish communications with low setup cost and short startup delay. Store-and-forward mechanism is the basic method to transmit data in MANET. The cooperation network uses the decentralized scheme to share data in small scale for the specific purpose.

Users can share resources and files through MANET via application-layer broadcasting or multicasting. However, the inefficient flooding delivery leads to the low data rate in MANET. While the traditional server-client or one-to-many delivery model is not satisfactory to the requirement of multimedia service. A single node is impossible to handle the global load of the multimedia service, hence a decentralized approach like peer-to-peer (P2P) technology could be an appropriate solution adopted in such environment [4]. P2P solution can disperse the centralized actions in traditional network to let the edged terminals connect to each other without central control. P2P solution can alleviate single point of failure and overhead on server.

1.1. Background

Like a peer in P2P application, a mobile node simultaneously plays the roles of server, router, and client. When the mobile node acts as the role of a server, it may be an agent, an anchor, or a repository of WLAN, and becomes the source to provide data. When the role of a mobile node is a router, it may be an intermediate node on communication, and becomes the relay to forward data. When the mobile node acts as a client, it may be an end user or a receiver, and becomes the terminal to query and download data.

Like MANET, P2P network is established as a service-oriented overlay to share data. Peers always collaborate with each other without any centralized server.[3] Therefore, how to

---

[2] A node can communicate with other nodes within its radio range directly or the ones out of its radio range indirectly through multi-hop relay.

[3] P2P solution can establish the many-to-many connections without any central control.

keep the scalability and how to deliver data efficiently are important to both P2P and mobile networks. In order to achieve high scalability and efficiency, more and more P2P applications have been implemented in wireless network in recent years, such resolutions can be called mobile P2P (MP2P), wireless P2P (WP2P), or P2P over MANET (P2P-MANET).

Both P2P and MANET share some fundamental commonalities such as decentralized and self-organized architecture due to the lack of central server, therefore MANET and P2P systems both exhibit a dynamic organization and lack information of user arrival or departure a priori. There are some similarities between MANET and P2P.

(1) A peer should collaborate with each other without a centralized server.
(2) A peer operates as both host and routing node, as well as sender and receiver in both systems.[4]
(3) The overlay of P2P and topology of MANET is changing frequently due to the high churn rate.[5]
(4) In both MANET and P2P, scalability is the basic requirement.
(5) How to deliver or route data efficiently is crucial for both P2P and MANET.

However, P2P and MANET techniques have been developed by different communities and address entirely different requirements. MANET first appeared as DARPA packet radio networks in the early 1970s. P2P systems were initiated in the middle of 1990s. P2P systems are application-oriented overlays and have evolved mainly over the wired Internet so far. By contrast MANET is spontaneous, infrastructure-less wireless networks for mobile terminals. Therefore, P2P systems are unsuitable for MANET because P2P overlay is based on TCP/IP wired network, but the links of MANET change frequently due to the mobility of its wireless nodes. The other differences between P2P and MANET also obstruct the integration of P2P

---

[4] Each peer acts both as a client and a server, or called servant.

[5] The topology means the distribution of nodes in MANET physically, and the overlay means the distribution of peers on P2P logically.

and MANET.

(1) P2P refers to the upper layer; MANET refers to the lower layer.

(2) In P2P, the logical routing path is established on overlay; while in MANET, the physical routing path is established on MAC layer.

(3) Every peer on P2P overlay can receive the incoming streams from multiple sources, but every node in MANET has to deal with the collision of packets from different sources.

A fundamental characteristic of wireless networks is the time-varying and user-dependent fading channel. Mobile devices have the limited resources and bandwidth, and the bottleneck of data traffic causes quality of service (QoS) degradation of multimedia service. QoS requirement of multimedia traffic is strict (little loss, low delay and small jitter) and the nature of MANET is dynamic (fluctuating link quality and changing topology). Therefore, the dynamic multi-hop topology and fluctuating radio quality make it difficult for MANET to support various kinds of P2P traffic with acceptable QoS.

(1) MANET degrades the maintenance of P2P overlay due to the dynamic mobility problem.

(2) P2P overlay is not proximal to MANET topology, because it leads to the inefficient streaming delivery.

(3) The unreliable wireless links lead to the out-of-order congestion of multi-source P2P voice communications.

1.2. Problem

Several problems affect adversely data delivery performance when integrating P2P and wireless networks. The challenges of P2P-MANET are concerning both P2P and MANET, thus P2P-MANET needs to accommodate continuity, scalability, mobility, and proximity to support the multimedia content sharing services.

1.2.1. Continuity

The peers in P2P-MANET leave and join on the go; the continuity means that a peer stays in the system continuously. The word *churn* describes the contrary behavior, or means that peers arrives and departs from time to time. In the application layer, a user can open or close the application anytime. In some cases, a peer can inform its neighbors gracefully when it joins or leaves the system. In other cases, a peer may turn off power or encounter an accidental crash, such that it leaves without any notification. In network layer, three situations regarding the peer churn problem can be categorized:

(1) The P2P application is terminated but the user still stays in the ad hoc network.
(2) User turns off the power and leaves both the P2P application and ad hoc network.
(3) User moves to another coverage area; this case leads to an ambiguous situation in which user leaves neither P2P application nor ad hoc network.

The peer churn is often unexpected a priori, and it destroys delivery overlay and breaks routing path. The churn is the most critical factor that affects the overall performance. Therefore, an integration of P2P-MANET protocol not only defines and addresses clearly the above situations, but also considers the peer churn problem.

1.2.2. Scalability

On *logical overlay*, the size of P2P-MANET varies constantly due to peer churn problem. In addition, the members of P2P-MANET changes dynamically. On *physical topology*, the size of ad-hoc area also varies due to the self-organized infrastructure-less environment. As a result, the basic scalability is necessary for P2P-MANET data sharing service. For P2P-MANET, the main concerns are scalability and robustness. In wired P2P network, the system concern is regarding the scale of million users. However, the scale for hundred users is sufficient in P2P-MANET. The major scalability issue of P2P-MANET system is the sudden burst crowding problem.

1.2.3. Mobility

A major difference between wired and wireless P2P systems is the mobility in P2P-MANET. The traditional P2P system doesn't consider peer mobility because it is developed on the wired network, but the P2P-MANET mobility includes device's movement and hidden terminal problem. Hence, the P2P-MANET application needs a mobility detection mechanism to serve mobile users.

In most of P2P-MANET systems, a mobility detection function is used for monitoring peers' movements and checking peers' departures. The mobility detection includes the *active* notification and the *reactive* notification. In general, the more complicated mobility detection, the better the performance (i.e. mobility prediction), but it costs higher overhead on both computing at mobile terminal and network traffic.

1.2.4. Proximity

Unlike wired P2P application, a self-organized P2P-MANET system cannot rely on any form of *offline* knowledge and resource. Every peer has its own domain, which leads to a challenge to interact and communicate with each other. An inefficient organization results in poor routing.

How to find a mapping from physical topology to logical overlay is a shortcut to realize and solve the proximity problem. The mapping affects significantly how to establish routing path or multicast tree, and influences directly (QoS); users care about the downloading time when using file sharing service, whereas they care about the playback quality when using real-time streaming service.

1.3. Motivation

In all infrastructure-less environments, P2P applications can be implemented in MANET due to high mobility, low underlay cost, and short initiation delay. Especially the environments may be the applications of rescue and academia. In emergency rescue application, every rescuer carries the handheld device and search around in a large area. Every rescuer shares the pictures, gathers the statistics, communicates and shows live video with each other through P2P-MANET. The P2P-MANET technology replaces the satellite telephone to provide the triple-play and low-cost communication. In campus, every teacher and student uses the education multimedia service to share the teaching materials or build a virtual classroom. Everyone uses the handheld device to hold the grouping conference or multimedia multicasting without complex setup procedure.

Therefore, we would like to design a multimedia platform to provide common P2P services including file sharing, voice communication, and video streaming for emergency rescue and academia applications. The development of platform must consider the scalability and high churn rate. Since the P2P-MANET application discussed here tends to be applied to rescue and academia applications, the problems of continuity, scalability, mobility, and proximity must be overcome. Besides the above innate problems, the delivery method should be suitable for all situations, because the rescue application is usually situated in a space of limited users and large area, while the academia application is in a space of numerous users and small area. Therefore, the proposed scheme must consider the generic design of P2P overlay, delivery strategy, and ad hoc route.

In summary, in order to achieve P2P sharing service in wireless network, we want to design a generic P2P platform to support all P2P applications on MANET with affordable cost and short setup delay for high continuity, high scalability, high mobility, and high proximity. The achievement needs the cross-layer information between P2P overlay and ad hoc topology, i.e. the upper tier needs not only logical metrics but also physical metrics, so does for the lower tier. Therefore, we motivate a cross-layer design to enhance the P2P-MANET platform. In our published works, the P2P applications of voice communication [b] and video streaming [c] have been workable via cross-layer scheme.

1.4. Goal

We integrate the published works to present a cross-layer approach for P2P forwarding mechanism tailored to MANET. The proposed scheme tends to accommodate the file sharing, voice communication, and video streaming with high data rate and time sensitivity on P2P-MANET. The goal is to achieve the high continuity, high scalability, high mobility, and high proximity to shorten the routing propagation delay, as well as to reduce the signaling overhead.

The proposed scheme builds a service-oriented overlay to share data without any centralized server, and constructs the generic platform, which represents a middleware that combines the context-aware search, the cooperative data management, the anycast-based query, and the mobile P2P delivery. The proposed platform is independent of upper layer for P2P applications and lower layer for MAC protocols, so it is extendable for all kinds of P2P sharing services with high scalability and mobility.

The proposed scheme integrates P2P overlay into the cross-layer scheme to manage the ad hoc topology. Through P2P overlay, a virtual network can be formed in the long-term and large-scale perspective. The cross-layer design can deal with peer arrival and departure quickly and simply to shorten the service interruption time with minimal signaling overhead. The implementation of platform needs neither the complex multimedia mixer nor the connection-oriented delivery.

As above characteristics, we call the proposed scheme Generic P2P-MANET System (GPMS), which establishes an overlay on MANET, then builds a platform on the overlay, and provides multimedia services on the platform. GPMS consists of five algorithms, the first algorithm maintains the application layer overlay which is proximal to physical layer topology[6]; the second algorithm provides an efficient routing via IPv6 to achieve the high data rate[7]; the third algorithm handles the interaction of cross-layer messages to keep an optimized shortest routing path[8]; the fourth algorithm uses anycast-based query to minimize the querying overhead; and the last algorithm schedules the source selection with dynamic buffering mechanism to avoid the performance degradation.[9]

The integration of P2P overlay and MANET routing can update the cross-layer information to reduce signaling overhead (control overhead), speed up recovery time, and improve delivery continuity or service stability. The integration also avoids the packet duplication problem because GPMS adopts neither the flooding algorithm nor the off-the-shelf P2P systems.

---

[6] The algorithm of overlay proximity is published in IEICE Transactions on Communications [b].

[7] The algorithm of IPv6 routing header is published in Ad Hoc Networks [c].

[8] The algorithm of cross-layer scheme is published in IEICE Transactions on Communications [e].

[9] The algorithm of dynamic buffering mechanism is published in International Journal of Intelligent Systems and Applications [a] and The International Symposium on Computer Network and Multimedia Technology [f].

2. Related Works

Because GPMS involves P2P overlay, mobile network, IPv6 protocol, and multimedia delivery, we discuss the P2P-MANET scheme, P2P-MANET taxonomy, P2P-MANET application, middleware design, IPv6 routing header, IPv6 anycast, and delivery strategy one by one.

2.1. P2P-MANET scheme

P2P-MANET application must take care of the P2P overlay and the ad hoc routing protocol. P2P networks can be roughly classified as either unstructured or structured network. Unstructured networks use the flooding algorithms, while structured networks use the distributed indexes. Generally speaking, unstructured protocols are extremely resilient, but they are inefficient for querying and sharing data. On the other hand, structured protocols are efficient to maintain the overlay, but they are unstable in the mobile environments.

- Unstructured P2P networks, such as KaZaA [5] and Gnutella [6], generally use flooding to search peers and files. Gnutella use the *lifespan* to limit the number of visited hops for application-layer flooding. The lifespan is usually set to 1 or 2 hops in wireless network.
- Structured P2P networks use the *finger table* or *distributed hash table* (DHT) to locate peers and files. DHT has proven to be an efficient solution for a variety of scalable and robust distributed applications, for example, CAN [7], Chord [8], and Pastry [9] have been used successfully in wired P2P network.

The ad hoc routing protocols can be classified into *reactive* and *proactive* routing protocols.

- In proactive or *table-driven* routing protocols, such as DSDV (Destination Sequenced Distance Vector) [10] and OLSR (Optimized Link State Routing) [11], every mobile node maintains a *routing table*[10] with the global information of network topology. A node immediately sends packet to the destination without the delay for route discovery. But a lot of computing cost and control overhead is needed to maintain a dynamic routing table.

- The reactive or *on-demand* routing protocols, such as AODV (Ad hoc On demand Distance Vector) [12] and DSR (Dynamic Source Routing) [13], every mobile node finds a routing path only when it is necessary. But the sender must broadcasts the routing request throughout the whole network to select an appropriate path to the receiver. Reactive protocols have larger delivery delay and are not efficient for heavy traffic.

2.1.1. DHT

DHT inherits a hash table, which provides the lookup function inspired by a centralized server. In P2P network, responsibility for maintaining the mapping from keys to contents is distributed among the peers, in such a way that a change causes a minimal amount of disruption. Therefore, DHT is scalable for a large number of peers to handle peer arrivals, departures, and failures.

DHT constructs an overlay, which takes advantage of resources distributed across the Internet, such as bandwidth and hard disk capacity, to provide a multimedia service. As Figure 2.1 illustrated, the *key-based forwarding* scheme guarantees logarithm efficiency for table size, neighboring degree, lookup latency, and delivery overhead.

2.1.2. AODV and OLSR

---

[10] Routing table is a data structure in the form of objects stored in each mobile node. The construction of routing tables is the primary goal of routing protocols for packet forwarding.

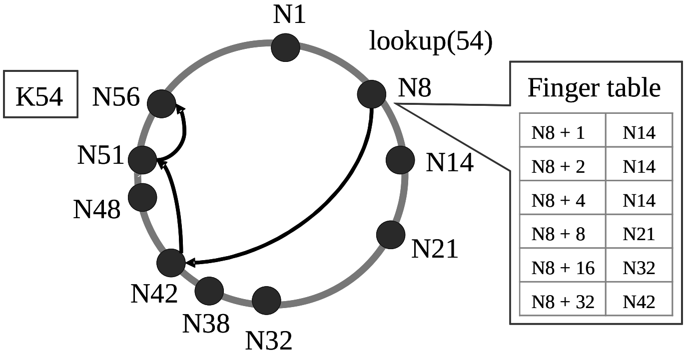
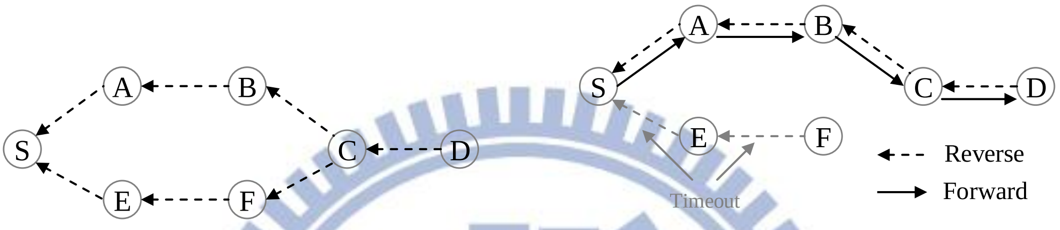


Figure 2.1: The basic key-based forwarding scheme of DHT.

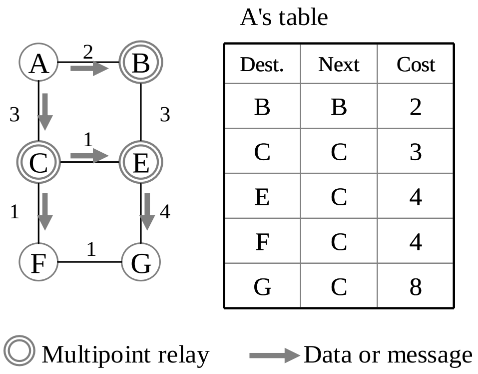


(a) Reverse path (b) Forward path

Figure 2.2: The selection of routing path in AODV.

Although AODV inherits a distance-vector routing protocol as DSDV, AODV is a reactive routing protocol, and it establishes a route between a source and a destination only on demand. AODV uses the concept of *destination sequence number* from DSDV, and it uses a route discovery similar with DSR. Instead of source routing, however, AODV dynamically establishes route entries at intermediate nodes and uses bidirectional routing information; i.e. AODV uses the hop-by-hop routing. As Figure 2.2 (a) illustrated, the destination can set up a *reverse* path via *Route Request Message* and *Route Reply Message*. Then the source and intermediate nodes maintain the *forward* path via an unsolicited *Route Reply Message* with a fresh sequence number. Each node updates its routing information and propagates the message only if a greater destination sequence number than the previous message. The link is deleted after a timeout or a point failure in the forward path as Figure 2.2 (b) illustrated.

OLSR is a proactive link-state routing protocol, which disseminates link state information among individual nodes. OLSR is based on link state algorithm and optimizes the



A's table

Dest.	Next	Cost
B	B	2
C	C	3
E	C	4
F	C	4
G	C	8

Figure 2.3: The maintenance of routing table in OLSR.

efficiency and economical flooding. OLSR uses the *multipoint relay*, which is selected for forwarding message and data. Only multipoint relay can forward data and *Topology Control Message*, which is used to update the information of bidirectional link state. Thus, a routing path is a sequence of hops from the source to destination through multipoint relays as Figure 2.3 illustrated.

2.2. P2P-MANET taxonomy

We classify the taxonomy of P2P-MANET protocols depending on the construction of network stack. Due to consideration of both P2P overlay protocol and ad hoc routing protocol, we can discuss the approach of interaction between the overlay tier and the routing tier, and categorize the existing protocols into three types: *off-the-shelf* scheme, *layer-enhanced* scheme, and *cross-layer* scheme.

First, the off-the-shelf scheme usually puts up the ready-made tools for a specific proposal, such as file sharing or video streaming. A well-known P2P overlay protocol can be combined with an ad hoc routing protocol to form the off-the-shelf scheme, which is easy to implement with affordable cost. The scheme can be divided into two classes: *combination* and *integration*. The combination only considers the conformability of overlay tier and routing tier; on the other hand, the integration fetches the features of P2P service and integrates with MANET.

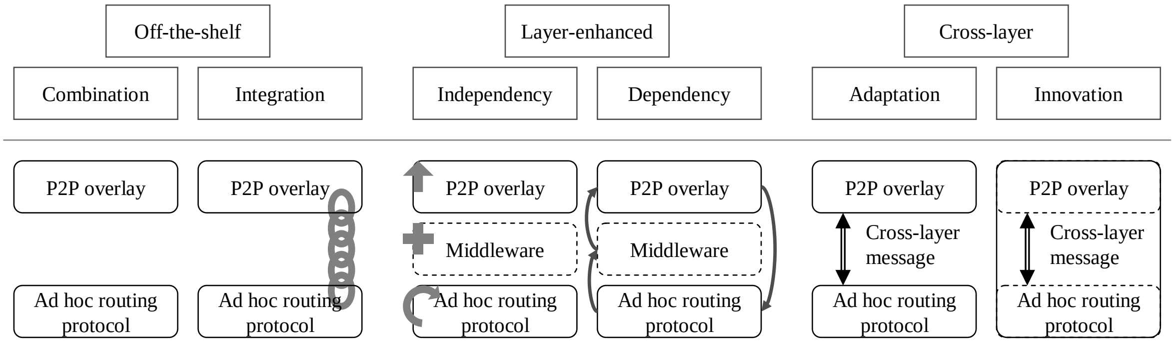


Figure 2.4: The network constructions of P2P-MANET protocols.

Second, the layer-enhanced scheme generally modifies either overlay protocol for mobile environment or routing protocol for P2P service. The modification of overlay usually costs additional recovery overhead; the modification of ad hoc routing usually costs specific point-to-point overlay path. This kind of scheme often depends on the existing protocols to enhance the QoS of P2P service. Although the modification can improve P2P system efficiency, it may change the network protocols and decrease its popularity. The scheme also can be categorized into two classes: *independency* and *dependency*. The independency considers the enhancement of one-sided tier, the overlay is independent of the routing algorithm; on the other hand, the dependency considers the coordination and enhancement of both tiers, the overlay is dependent on the routing algorithm.

Third, the cross-layer scheme mainly breaks the network stack to provide an interaction between the overlay tier and the routing tier via the additional cross-layer messages. The cross-layer messages usually include the information of overlay proximity and routing path. The scheme has a complex cross-layer algorithm and is incompatible with other P2P systems. The scheme can also be divided into two classes: *adaptation* and *innovation*. The adaptation regards adding the cross-layer messages on the original overlay tier or routing tier to adapt the cross-layer design; the innovation regards the novel cross-layer design for P2P-MANET.

As Figure 2.4 illustrated, P2P-MANET protocols can be classified into three categories based on their network constructions. In the Appendix below, we survey and classify the existing protocols via the categorization. We also list these protocols with their proposal date in order, and discuss solutions for the problems of continuity, scalability, mobility, and proximity.

2.3. P2P-MANET application

For the orientation of service, the P2P-MANET application is similar to wired P2P application. The application can be divided into three kinds: file sharing, voice communication, and video streaming.

2.3.1. File sharing

The kind of file sharing is the first application of wired P2P service. The file searching is completed quickly via the P2P overlay, which is constructed by finger table. The mobile P2P application inherits the advantage to develop the file sharing function.

M-CAN (Mobile-CAN) [14] is a modification based on CAN, which provides the P2P functions of registering, grouping, file transmission, caching, as well as the DHT functions. Every node is registered on one or several nearby *super node*s, which are organized by CAN. The goal of M-CAN is to provide a fast lookup for file sharing on ad hoc network. The super node must index all neighborhood available files and handle all lookups, so the overhead of super node is obviously high.

MADPastry (Mobile AD hoc Pastry) [15] combines the DHT overlay, Pastry, and the ad hoc routing protocol, AODV, to reduce the overhead of peer lookup and routing maintenance. MADPastry uses Pastry proximity awareness to reduce the overhead without flooding. MADPastry is implemented at the network layer. MADPastry maintains three routing tables: *Pastry routing table*, *Pastry leaf set*, and *AODV routing table*; those tables complete the *indirect routing*. Pastry routing table indexes and hashes the mobile nodes; Pastry leaf set knows the neighbors on overlay; AODV routing table is extracted from standard AODV.

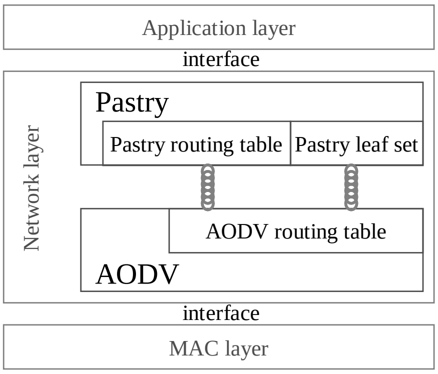


Figure 2.5: The integrated network stack of MADPastry.

These routing tables can be updated via overhearing data packets, and the routing path without data flow is not monitored. The extension of MADPastry integrates three routing tables to provide interfaces to P2P application layer and MAC layer as Figure 2.5 illustrated. Its motivation is to avoid AODV route discovery, minimize overlay maintenance, and maximize available packet information. Because Pastry is suitable for the wired network, MADPastry performs poorly at high speed.

M-Chord (Mobile-Chord) [16] is a modification based on Chord. M-Chord is very similar to M-CAN and is inspired from the hierarchical architecture of M-CAN. M-Chord also uses the *super node* to associate with the ordinary nodes and manage the file indexes. Every super node needs to maintain the *routing finger table* and *sharing files directory* to achieve Chord's lookup and file sharing. The joining/leaving/routing/updating process of peer follows Chord and is similar with M-CAN. M-Chord combines the *real* and *virtual* split strategies to achieve a load balance and low overlay overhead.

2.3.2. Voice communication

The application connection of voice communication or group conference is established via Session Initiation Protocol (SIP) based on flooding-based multicast. A peer always floods the received data to other peers immediately. The hierarchy-based or cluster-based architecture is used to improve the efficiency of flooding delivery.

Skype [17] can hold the voice communication or group conference through Internet.[11] It classifies peers into two types: *super peer*[12] and *ordinary peer*. The super peer is responsible for building P2P overlay and optimizing the routing path. Hence, a super peer is like a leader, which synchronizes voice data then sends it to ordinary peers through multicast.

Audio Conferencing Testbed (ACT) [18] is based on OLSR to set up a one-to-one or many-to-many communication in WiFi MANET. ACT uses the minimum spanning tree[13] to minimize the latency of audio dissemination to the whole network. Every peer must maintain its minimum spanning tree to deliver data by itself. ACT predicts the disconnection and mobility to shorten the service interruption time.

CLAPS (Cross-Layer And P2P based Solution) [19] inherits the tree-based overlay and OLSR extension from MOST[14] for real-time video streaming. CLAPS assumes that the physical routing topology can be provided by OLSR, which sends cross-layer message to optimize the overlay. The source peer maintains a minimum spanning tree as its overlay. The minimum cost is computed via *link distance* packaged in cross-layer message, and the spanning tree is recomputed periodically to keep the overlay proximity. To optimize the overlay proximity and avoid the overlap path in ad hoc routing, the *candidate peer*s are selected as relay nodes on multicast paths according to the equal link distance. The cross-layer architecture of CLAPS is shown in Figure 2.6. In the simulation, CLAPS adopts UDP as transport layer protocol and WiFi as MAC layer protocol.

---

[11] Skype cannot work if MANET cannot access Internet.

[12] Any peer with a public IP address having sufficient CPU, memory, and network bandwidth is a candidate to become a super peer.

[13] *Broadcast tree* is the buzzword in ACT.

[14] MOST (Multicast Overlay Spanning Tree Protocol) extends the OLSR unicast to support multicast routing. It is a multicast routing protocol, not an overlay protocol. Although MOST uses the minimum spanning tree as its overlay, it does not consider P2P issues. However, the overlay of CLAPS is inspired from the spanning tree of MOST due to the multicast purpose.

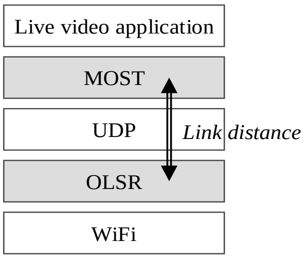


Figure 2.6: The cross-layer architecture of CLAPS.

2.3.3. Video streaming

Both live streaming and video on demand (VoD) are the applications of video distribution. However, supporting live streaming is more difficult than supporting VoD due to the instant factor.[15] The audiences cannot tolerate any sensible lags in such live programs. Therefore, how to continue the stream smoothly and deliver it efficiently among the peers is important to use P2P solution.

Smart Gnutella [20] enhances the original Gnutella for MANET and the real-time application.[16] Smart Gnutella inherits the *ultrapeer* from the wired Gnutella. The ultrapeer must manage the neighborhood *leaf peer*s and support the multicast communication in Smart Gnutella. In order to meet the wireless and mobile environment, Smart Gnutella defines three message types (*ping*/*pong*, *welcome*, and *broadcast*) and four node states (*full*, *stable*, *connecting*, and *idle*). The application-layer QoS can be monitored via *ping*/*pong*; the peer discovery and connection maintenance can be established via *welcome*; the node states can be updated via *broadcast*.

As Figure 2.7 illustrated, MP2PS (Mesh-based P2P Streaming) [21] provides real-time

---

[15] The live streaming service usually provides the live sports games, the first-hand stock information, or the latest news.

[16] Gnutella is designed for file sharing in the wired network. Smart Gnutella is designed for QoS-sensitive *collaborative virtual environment* on MANET.

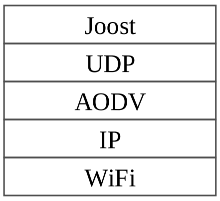


Figure 2.7: The network layer of MP2PS.

streaming with scalability and availability over MANET.[17] MP2PS adopts mesh-based live streaming application, Joost,[18] and no retransmission on wireless network, UDP, and on-demand ad hoc routing protocol, AODV.

2.4. Middleware design

The middleware between application layer and MAC layer is proposed to provide a mobile P2P platform [22]. The middleware is an integrated dependency of overlay tier and routing tier, and it is independent of P2P service and wireless technology [23]. The different kinds of P2P applications and wireless accessibilities are workable and compatible under the middleware. For example, MAPCP (MANET Anonymous P2P Communication Protocol) [24] is a middleware between the P2P applications and ad hoc routing protocols. MAPCP consists of two phases and two tables: a *destination table* records the results of query in the *query phase*, and a *path table* records the information of multiple paths in the *data transmission phase*. P2P-HWMP (P2P over Hybrid Wireless Mesh Protocol) [25] exploits the route discovery functionality and information provided by the underlying routing layer to improve the P2P search performance, and it collates the routing information to get the overlay proximity.

Therefore, a middleware platform should consider the additional adaptability, extensibility, and interoperability besides the issues of continuity, scalability, mobility, and proximity.

---

[17] MP2PS suggests a protocol stack for multimedia streaming, and its mesh overlay doesn't consider the proximity. The ready-made protocols don't consider the streaming continuity.

[18] Joost is a Video-on-Demand P2P system for distributing TV content, created by Skype and KaZaA founders. Its P2P protocol shares the library with Skype.

- Adaptability: P2P overlay is self-organized and independent of routing topology, it is also self-organized without any central administration or control. The wired P2P forwarding protocol leads to the far routing problem[19] due to the lack of overlay proximity, i.e. an inefficient organization leads to a poor P2P routing. Therefore, the mobile P2P resolution must be adaptable for dynamic environment.
- Extensibility: How to integrate with P2P overlay and routing topology to deal with a general purpose system is important in the mobile P2P solution. The extensibility describes the generic integration of P2P application on mobile device, but the mobile P2P scheme are often inherited from wired P2P scheme. Therefore, the mobile P2P solution must improve the extensibility to increase the popularity and flexibility.
- Interoperability: The mobile P2P applications should be service-oriented, but they are usually developed for a specific service in the heterogeneous wireless network. The mobile P2P application usually has an innate hardware limitation and binds the wireless technology, such that the heterogeneity limits the interoperability. Therefore, the mobile P2P solution must provide a platform to improve the interoperability.

2.5. IPv6 routing header

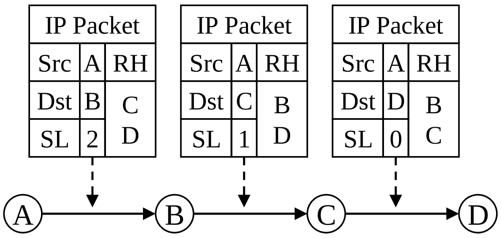
IPv6 packets consist of control information for addressing and routing, and a payload for user data. The control information in IPv6 packets is subdivided into a mandatory fixed header and optional extension headers. Extension headers carry optional network layer information, which forms a chain via *Next Header*. *Routing Header* is one of the extension headers and the type of Next Header is 43. The Routing Header is used to direct a packet to one or more intermediate nodes before being sent to its destination via two attributes, *segments left* and *type-specific data* as illustrated in Figure 2.8. Type-specific data is an array of routing headers,

---

[19] The far routing problem means a routing path through several redundant hops, which leads to a non-optimized end-to-end path.

Bits	0 ~ 7	8 ~ 15	16 ~ 23	24 ~ 31
0	Next header = 17	Extension length	Routing type = 0	Segments left
32	Type-specific Data (variable)			
⋮	⋮			

Figure 2.9: IPv6 routing header format.



Src = source
Dst = destination
SL = segment left
RH = routing header

Figure 2.8: An example of IPv6 routing header.

which is replaced according to segments left. Every mobile node can check the *destination* and the segment left of IPv6 packet to determine the next hop. As Figure 2.9 illustrates, when a logical P2P path A→D is established for the delivery from A to D, the intermediate nodes B and C must forward the packets. The upper layer must provide the path information to decide the next hop in each peer. The path information is filled in the attribute “routing header” of an IPv6 packet, and every mobile node checks the destination so as to forward the packet to the next node.

2.6. Anycast

Besides routing header that is used to speed up and stabilize data delivery, anycast is another feature of IPv6 to reduce the querying overhead. Anycast is an addressing and routing method based on IP network [26]. Unlike *one-to-one unicast*, *one-to-all broadcast*, or *one-to-many multicast*, anycast adopts the *one-to-one-of-many* delivery. A sender queries or sends data to an unspecified receiver, whose forwards such data to other receivers in the anycast group.

Anycast originates from IPv6 for service-orientated applications to decrease network traffic and shorten response delay. An anycast address can be assigned in an anycast group, in which the receivers with the same anycast address should receive the same packets. Although the source should send a query to the nearest destination among an anycast group of multiple receivers, the nearest destination is not constant via different routing principles or arbitrary routing paths. Therefore, anycast is suited to connectionless protocols generally built under UDP.

2.7. Delivery strategy

In MANET, how to route IPv6 packets as quick as possible is important, however, how to deliver the application data for qualities of services is essential in P2P network. In the file sharing perspective, how to quickly fill up the buffer is an important issue. In the voice communication perspective, how to maintain the continuous link and handle incoming/ outgoing queue is the necessary. From live streaming system's viewpoint, nevertheless, how to buffer the arriving data before playback deadlines is more crucial than filling up the buffer, although the latter is essential.

In order to represent a delivery flow, a general buzzword “*chunk*” means a piece of data, such as one file block or one video segment. In other words, a chunk is the smallest unit of application data. Each chunk owns a *sequence number* (SN) to represent the order of playback. The sequence numbers are used to avoid the blank file gap or disordered video frames. Therefore, how to deliver the application data for qualities of services becomes an issue of gathering necessary chunks. The issue is called *chunk scheduling*, which is integrated into the *delivery strategy* in the P2P network in addition to coordination with upload strategy.

A mobile platform for generic P2P applications should consider time sensitive data, time-shifting buffer, and long lasting traffic. In P2P live streaming, peer leaving still causes some chunks not available, also the outmoded chunks become the useless chunks. The video chunks are only alive for a short time which is about 250 second in the whole network. The data in live streaming buffer changes quickly, and the buffer can be regarded as a finite FIFO (first-in-first-out) circular queue. The buffer is shifting regularly as time passing as the contents are modified in accordance with the proceeding of playback. This buffer will be used to temporarily store the chunks for immediate playback. For example, media playback bit rate of PPLive is 380 kbps and peers often reserve 200 seconds for buffer [27]. Therefore, the buffer size is often no greater than 10 MB, which usually could provide 3 minutes IPTV program for playback. The live voice or video data always come in a continuous stream form, i.e. *consecutive segments*. The downloading bit rate should not necessarily be high as long as chunks arrive at a stable rate. In general, the downloading bit rate of a peer with a stable source often equals to the playback bit rate.

Buffer control is indispensable to a P2P network, and it is also one component of P2P system to avoid jitter. Buffer mitigates the possible challenges because of high churn rate. In general, a churn problem often causes an interruption or a break of data delivery; hence, the buffered data can deal with the urgent chunk scarcity problem before the overlay recovery.

3. The Proposed Scheme

In our published works, the P2P applications of video [c] and voice [b] live streaming are supportable in MANET. We use the base to propose a generic platform, including the IPv6 routing [c] and cross-layer design [e]. The proposed scheme, GPMS, is a cross-layer middleware under application layer and above MAC layer. The middleware design allows generic P2P services on WiFi or WiMax network. As Figure 3.1 illustrated, GPMS builds a relationship between the logical overlay and the physical topology. The proposed scheme contributes to the multiple real-time P2P applications on mobile network, the overlay proximity and efficient routing, and the on-demand self-organization.

3.1. System overview

As Figure 3.2 illustrated, GPMS is a cross-layer middleware of overlay layer, transport layer, and network layer.

- IPv6 for network layer: Anycast is used for service querying, and the extended routing header is used for hop-by-hop routing. Therefore, GPMS uses IPv6 as the network layer protocol [c].
- UDP for transport layer: UDP can use a datagram socket to establish end-to-end communication in MANET. Its connectionless characteristics including lightweight and datagram without acknowledgment, retransmission, and timeout, are suitable for wireless P2P real-time services. Importantly, UDP is suitable for IPv6 anycast and mobile hop-by-hop routing.
- P2P for overlay layer: P2P overlay is used to identify, index, and manage mobile nodes [e]. The proposed *finger plus routing table* (*FPRT*) is used to construct the P2P overlay and

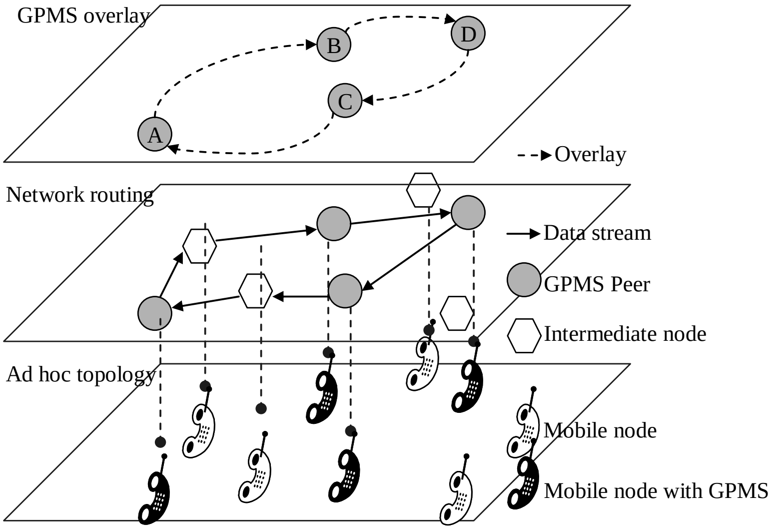


Figure 3.2: Relationship between the logical overlay and the physical topology.

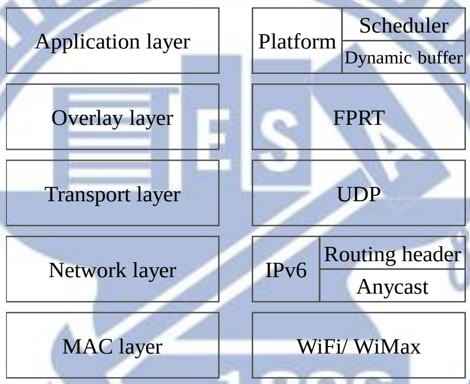


Figure 3.1: The proposed cross-layer scheme (GPMS).

support multiple services.[20]

- Platform for application layer: Generic P2P applications including file sharing, voice communication, and video streaming are applied on the platform. According to the service type of FPRT, various routing paths are derived on the overlay, and the proposed *scheduler* combines the *dynamic buffer* to deliver all kinds of data.

3.2. Finger plus routing table

Because FPRT caches the service type and routing information, GPMS can support multiple

---

[20] FPRT is extended from EDHT of COME-P2P [c] which is published in Ad Hoc Networks.

multimedia services and route data through overlay and ad hoc network simultaneously, and the cached information is called *path information*. Path information is used to maintain FPRT and cross-layer scheme, and provides the information of overlay tier and routing tier simultaneously. Because FPRT must be aware of IPv6 routing principle, the attributes of FPRT include the related information of IPv6 header.

The FPRT is used to construct the P2P overlay. FPRT takes the advantage of path information, and FPRT is shared as the combination of finger table and routing table on overlay tier and routing tier respectively in GPMS. Path information is for UDP and IPv6 routing to derive a local path between two peers through multiple intermediate nodes. An intermediate node without GPMS still uses its routing table and ignores FPRT. Every mobile peer maintains its FPRT to know its members via the reactive notification of arrival or departure and the proactive detection of movement, so the path information is updated to accommodate the peers' mobility and churn. As a result, GPMS achieves the high overlay proximity.

The high overlay proximity improves the high routing scalability, so the path information can be used in a routing protocol in the proposed cross-layer scheme. GPMS doesn't need the flooding scheme of routing discovery. It uses the extended routing header in IPv6 to forward packets hop by hop, so the ad hoc routing protocol is replaced by the membership of overlay, such that the overhead is little. Because the next hop field[21] of IPv6 is based on the path information of FPRT, the overlay proximity leads to high routing efficiency.

3.3. Cross-layer scheme

GPMS uses the cross-layer messages to implement the proposed middleware. When overlay tier or routing tier has actions or events, FPRT is informed to update for overlay proximity and routing efficiency. Because FPRT is shared in overlay tier and routing tier, the actions must be considered as Figure 3.3 illustrated. For example, the arrival or departure of peers may lead to

---

[21] Next hop is a field or attribute of IPv6 header.

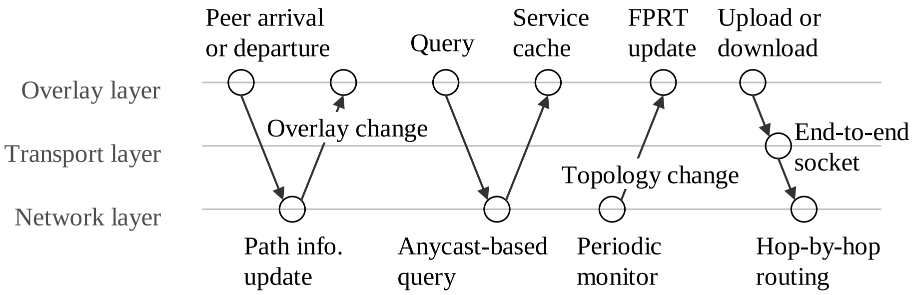


Figure 3.3: The FPRT update between overlay tier and routing tier.

the redirection of routing path, which must change the overlay; therefore, the FPRT must be updated for the peer churn to achieve the overlay stability and scalability.

In GPMS, FPRT must cache P2P service type and file query to support multiple P2P applications. When a peer queries a P2P service, an anycast resolution is initiated to search on MANET, and the results of query will be cached in the intermediate peers. Since each mobile peer is similar to the router on a local wireless network, the service priority and anycast addressing are enabled to route packets possibly. A group of peers sharing the same content, it is like a *swarm*[22] that shares the same IPv6 anycast address.

Besides the reactive notifications of arrival and departure, each peer periodically monitors the neighbors to detect the movement or failure proactively. The network layer essentially broadcasts *Hello Message* periodically to keep alive, so the accumulation of *Hello Message* is used as a metric of peer mobility to update FPRT for overlay and routing optimization. Since *Hello Message* is provided essentially in MANET, the cross-layer approach doesn't cause additional overhead.

When delivering content that includes file or multimedia stream, an end-to-end UDP socket is created to support the hop-by-hop routing path. In our proposed scheme, the attribute “next hop” of IPv6 is used to forward packets to the destination. The extended *routing header* in

---

[22] Swarm originates from BitTorrent, and it means a kind of mesh overlay managing network of peers. The main advantage of the swarming content delivery is its ability to effectively utilize the outgoing bandwidth of participating peers as the group size grows.

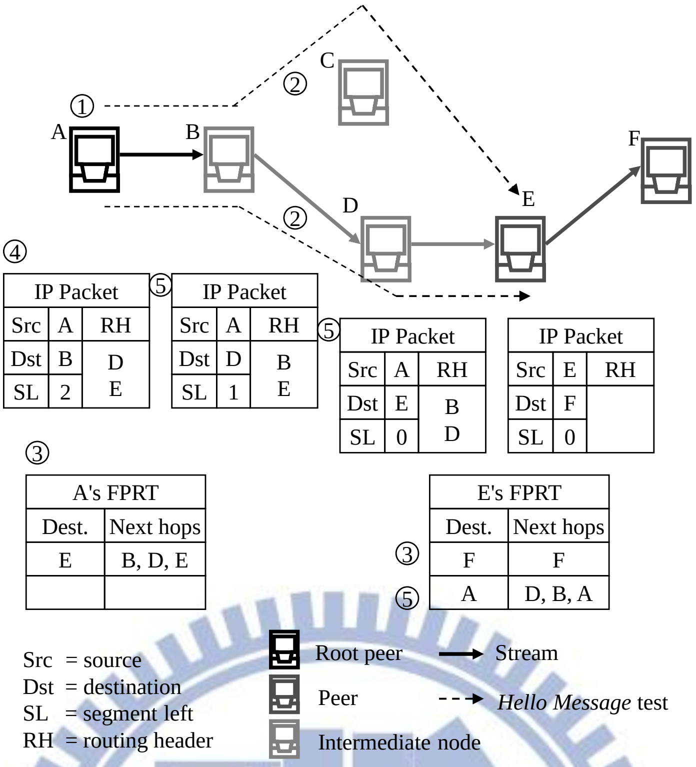


Figure 3.4: The proposed FPRT over IPv6 routing.

IPv6 helps forward packets hop by hop in pre-computed order. Every mobile node can check the *destination* and the *segment left* of IPv6 to determine the next hop. In addition, our proposed scheme is implemented on the simplest “Hot Potato” routing protocol, and it uses less overhead on network traffic and less computation on mobile node than other routing protocols. Hot potato scheme checks the next hop of IPv6 packet and forwards it to the destination immediately. As a result, FPRT provides path information of next hop to route the incoming packets in our proposed scheme. Although the limitation of GPMS is that only IPv6-enabled nodes can work, nowadays most of the modern routers provide dual stack for routing both IPv4 and IPv6 packets.

3.4. Routing algorithm

Hot Potato routing protocol can forward packet quickly without complicated and lengthy computation, and produce less overhead during routing packet. The routing path will not be changed until the path information is changed. Figure 3.4 illustrates how to forward the stream via the proposed FPRT and IPv6 routing. In the topology, node A is a peer which multicasts a streaming data; node B, C, D are the intermediate nodes; node E and F are the receivers.

(1) A logical P2P path $A \rightarrow E \rightarrow F$ is established in the P2P overlay via the FPRT lookup.

(2) After the latency estimation, a physical routing path $A \rightarrow B \rightarrow D \rightarrow E$ is selected because path $A \rightarrow B \rightarrow D \rightarrow E$ is shorter than path $A \rightarrow B \rightarrow C \rightarrow E$, path $A \rightarrow B \rightarrow C \rightarrow D \rightarrow E$ and so on.

(3) The path information is updated in the FPRT and used to decide the next hop in each peer.

(4) The path information is filled in the attribute “routing header” of IPv6 packet to speed up and stabilize data delivery.

(5) Every mobile node checks the destination of IPv6 packet and forwards it to the next node.

Because the FPRT provides the overlay proximity, the path information can be used for the hop-by-hop wireless IPv6 routing. Every mobile node is considered as a P2P router and follows the routing header of IPv6 packet. The optimal routing path in FPRT can be derived in advance to avoid the non-optimized end-to-end path via the cross-layer design. Instead of an on-demand routing in the intermediate node, the cross-layer IPv6 routing not only avoids the non-optimized end-to-end path, but also hastens and stabilizes the large-size video stream. The integration with Hot Potato routing protocol can forward packet quickly without complicated and lengthy computation, and produce less overhead to facilitate efficient routing over MANET.

The inter-layer packet is replaced with the proposed combination of packet headers as Figure 3.5 illustrated. Because the packet format is based on IP under UDP,[23] and the cross-layer approach is workable without any modification.

---

[23] As Figure 3.5 illustrated, the pseudo packet structure of GPMS can be seen as the combination of UDP and IP headers.

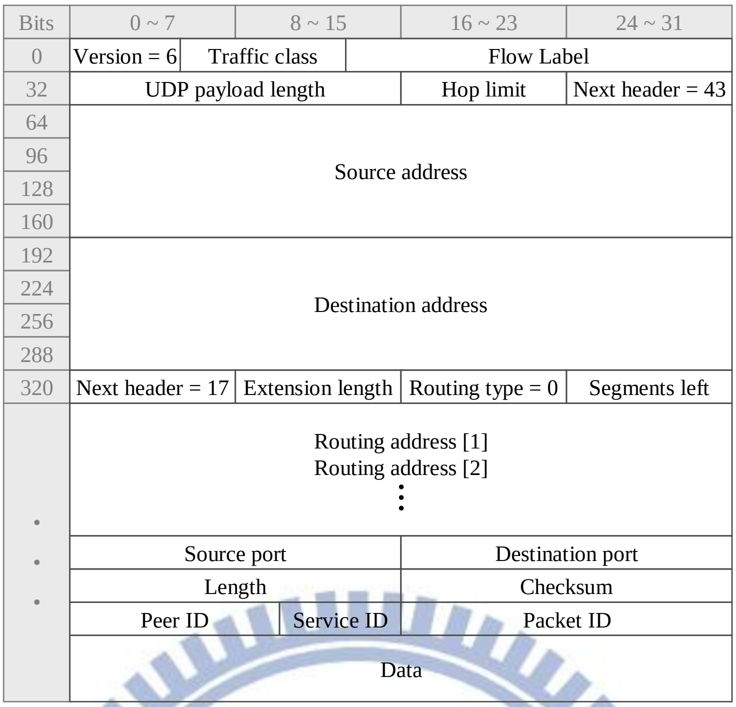


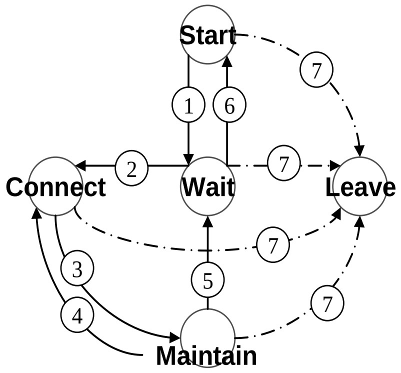
Figure 3.5: The pseudo packet structure of GPMS.

3.5. GPMS design

At the beginning, a peer joins the P2P network, and it maintains its neighborhood overlay via exchanging signaling messages with other peers until leaving P2P network. The lifecycle of peer represents the self-organization of overlay, and the finite state machine is driven via such messages to design for the lifecycle and behaviors. As Figure 3.6 illustrated, every peer should go through the states: start, wait, connect, maintain, and leave. Every current state should deal with the exact messages to avoid errors, which may crash the overlay.

Based on Figure 3.6, we give a normal example as in Figure 3.7, in which peer A plays the role in such example. Peer A is a newly joined peer in the P2P network, the joining process and querying process are described as the follows:

(1) First, because peer A knows neither any peer nor the overlay, it broadcasts *Arrival Message* to search the neighbors. The *Arrival Message* has the source address and routing trace.



① send *Join Message*

② receive *Join Accept Message*

③ receive *Join Message*
receive *Rejoin Message*
receive *Leave Message*
receive *Maintain Message*
timeout for *Hello Message*

④ send *Join Accept Message*
send *Join Reject Message*
send *Remain Message*

⑤ send *Rejoin Message*

⑥ receive *Join Reject Message*
timeout for *Join Message*
timeout for *Rejoin Message*

⑦ send *Leave Message*

Figure 3.7: The finite state machine or lifecycle of a peer.

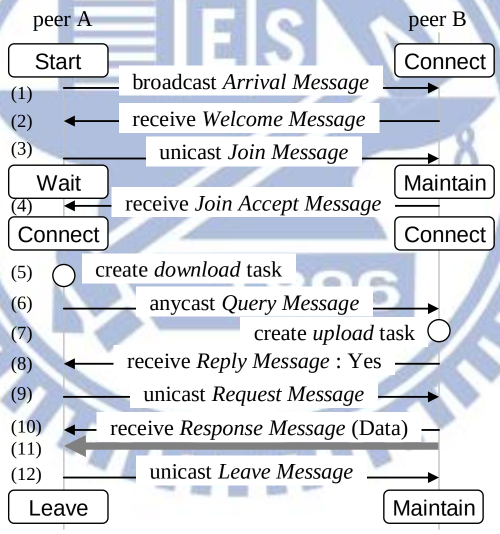


Figure 3.6: The normal message flow of GPMS.

(2) If an intermediate node receives an *Arrival Message*, it doesn't know such message and rebroadcasts to search some peers. When a peer receives an *Arrival Message*, it sends *Welcome Message* to peer A via the source address and routing trace, and it stops rebroadcasting. The *Welcome Message* has the destination address and routing trace.

(3) After receiving the *Welcome Message*, peer A unicasts a *Join Message* to select its member.

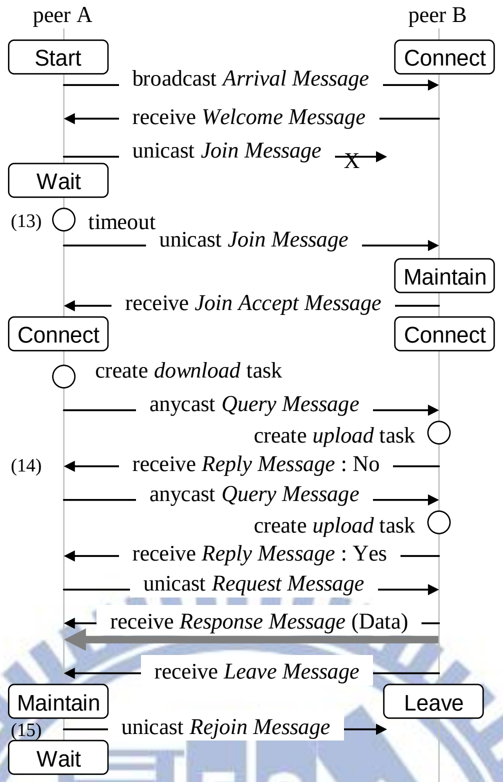


Figure 3.8: The abnormal message flow of GPMS.

(4) After receiving a *Join Accept Message*, peer A becomes one member in P2P network and can use P2P service.

(5) Peer A creates a *download task* when querying a file sharing or live streaming. If Peer A needs multiple services, it creates multiple download tasks.

(6) For example, peer A may anycast a *Query Message* for a specific document.

(7) When receiving the *Query Message*, the peer creates an *upload task* for a query.

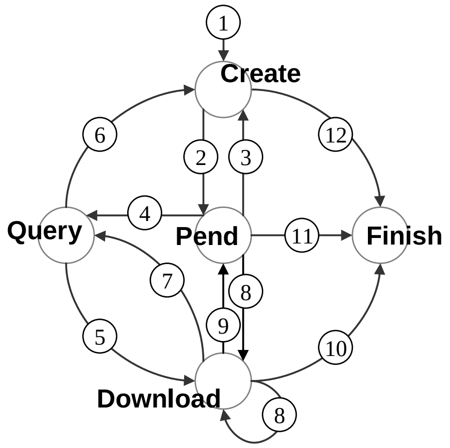
(8) *Reply Message* with the result of the query (i.e. yes or no) is sent to peer A. The query result is sent to user.

(9) Peer A unicasts *Request Message* for the available document depending on the query result.

(10) After receiving *Response Message*, peer A can download the queried document.

(11) The data is delivered upon the download task finished.

(12) Peer A leaves the P2P network via *Leave Message*. However, GPMS process isn't always



① demand query
② send *Query Message*
③ timeout for *Query Message*
④ receive *Reply Message*
: Yes or No
⑤ *Reply Message*: Yes
send *Request Message*
⑥ *Reply Message*: No
⑦ timeout for *Request Message*
⑧ receive *Response Message*
download data
⑨ timeout for download
peer adaptation
⑩ complete
⑪ abort
⑫ cancel

Figure 3.9: The finite state machine or lifecycle of download task.

normal during the overlay construction or data distribution. Therefore, we give an abnormal example based on Figure 3.7, it is shown in Figure 3.8.

(13) When peer A chooses the neighbor, peer B, as its member, it sends a *Join Message* to peer B and expects to receive *Join Accept Message*. If *Join Message* or *Join Accept Message* is lost, the timeout of new joining process avoids the endless waiting and *Join Message* will be sent again.

(14) When receiving the *Reply Message* with NO as the query result, peer A gives up the query.

(15) After peer B leaves, peer A searches other uploaders if peer B is the only one uploader. Therefore, the download and upload task must have the finite state machine to proceed the process.

As Figure 3.9 illustrated, GPMS creates a download task when a demand query arrives. The download task starts a timer when sending *Query Message*, and stops it upon receiving *Reply Message*. The file or stream is shared via *Request Message* depending on *Reply Message*, and a

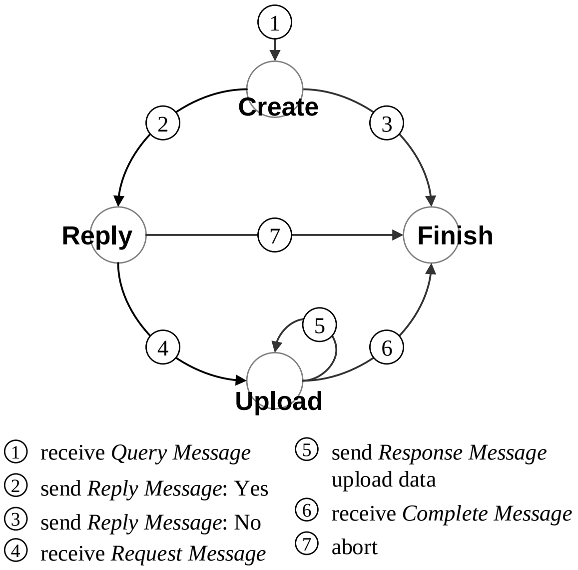


Figure 3.10: The finite state machine or lifecycle of upload task.

timer is switched to perform the download process. When the download efficiency is low or the up-loader is gone, GPMS executes the peer reselection to search other appropriate up-loaders and waits for new a *Reply Message*. The download task is completed in a normal case, but it will be aborted when pending the reply, or cancelled when creating the query.

The download task should go through the states: create, pending, query, download, and finish. The upload task is operating in coordination with the download task. Therefore, the upload task is created when receiving a *Reply Message* and finished when receiving a *Complete Message*. As Figure 3.10 illustrated, the upload task should go through the states: create, reply, upload, and finish. Since the upload process is coordinated with the download process, we don't describe the superfluous details here.

The relation between peers is many-to-many, and the relation between upload and download tasks is many-to-many, too. Therefore, the message flow of GPMS should combine the finite state machine of peer behaviors and upload/ download operations as Figure 3.11 illustrated. In the scenario, peer A and D are new joining peers, and they download the same video stream from peer B and C individually. During the period of live streaming delivery, peer C leaves, and peer D requests another member, i.e. peer B, for continuous streaming recovery.

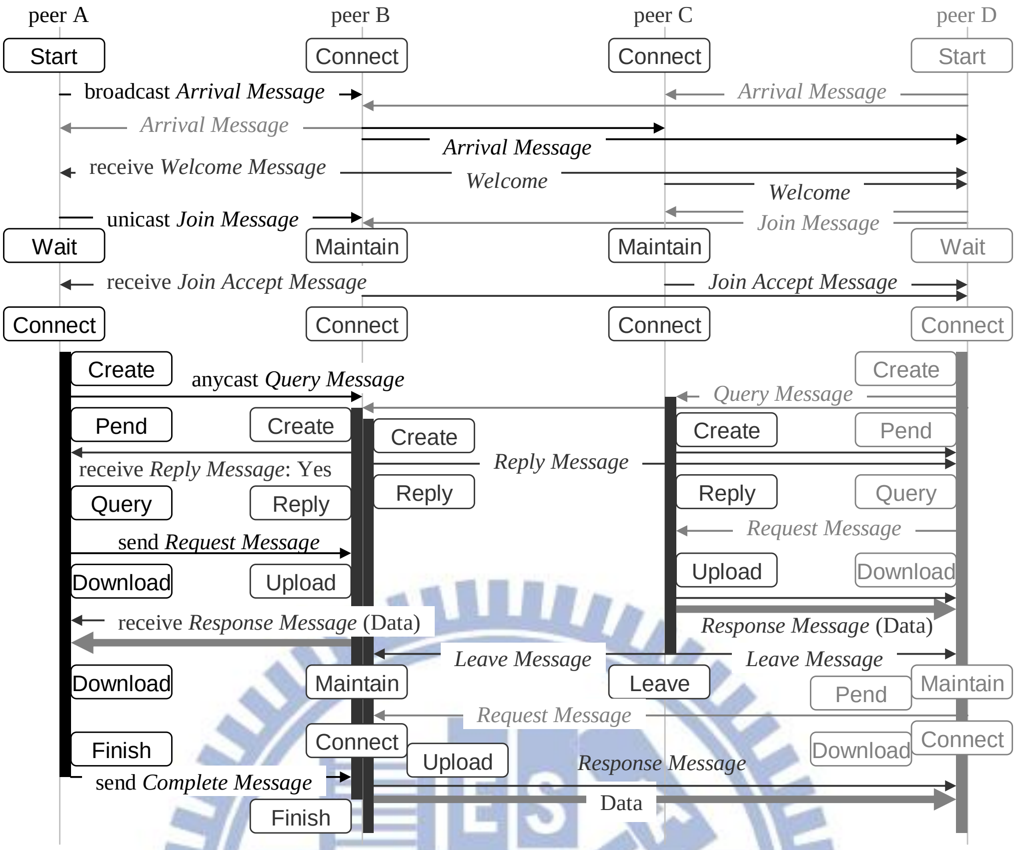


Figure 3.11: The message flow of GPMS.

When peer A stops the streaming delivery, it sends a *Complete Message* to peer B to terminate the upload task. The upload/ download operations of file sharing application are the same as the ones of live streaming application.

When a peer moves or leaves, the overlay must be adjusted to avoid non-optimized end-to-end path and service interruption. We give an example to explain the overlay maintenance and recovery of GPMS as Figure 3.12 illustrated.

(1) As Figure 3.12 (a) illustrated, the path A→C→E is connected, and all nodes (A, B, C, D, E) broadcast a *Hello Message* to keep alive periodically on ad hoc protocols.

(2) As Figure 3.12 (b) illustrated, after peer C moves, the number of received *Hello Messages* from peer C to A decreases.

(3) When peer A considers that peer C moves far away, it sends a *Maintain Message* to all peers on the path (only peer E in the example).

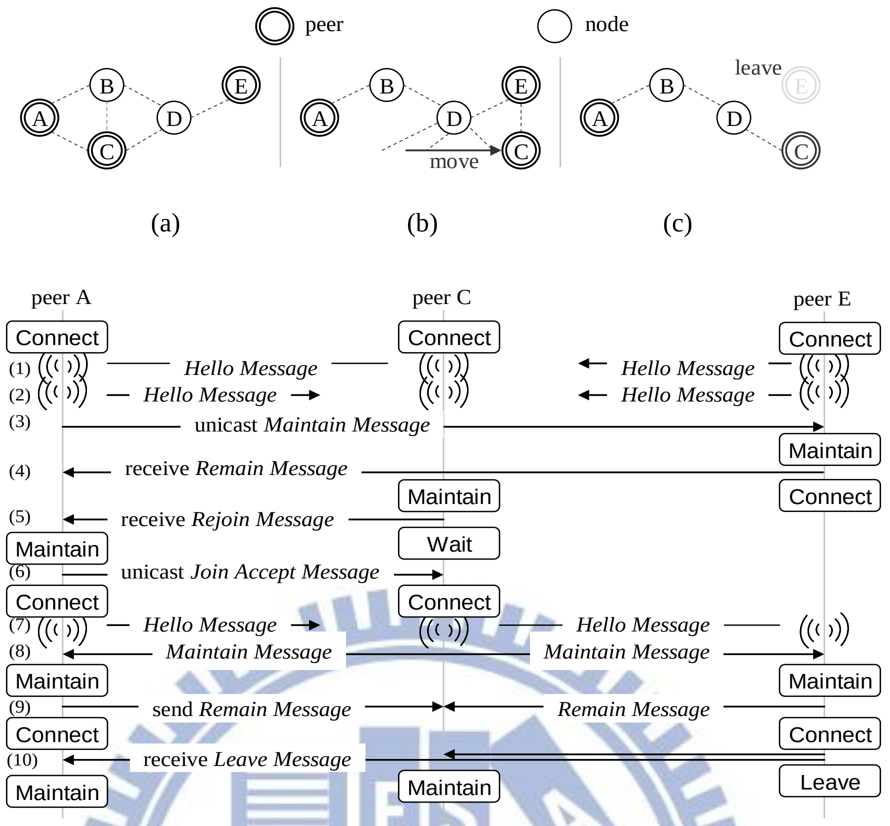


Figure 3.12: The overlay maintenance and recovery of GPMS.

(4) Peer C is still routable via node D, so both peers A and E update the path information. One path becomes two paths A→C and A→E. Peer E returns a *Remain Message* to avoid the interruption.

(5) If peer C cannot receive the *Hello Message* from peers A and E, it sends a *Rejoin Message* to peer A. If peer C receive data from peer A (via node D) before sending the *Rejoin Message*, the step will be ignored.

(6) Because peer A can route to peer C, it returns a *Join Accept Message*, and then peer C recovers the overlay. If step (5) is ignored, step (6) is ignored, too.

(7) When peer C receives *Hello Message* from peers E, it updates the path information because peer E has been known.

(8) Peer C sends a *Maintain Message* to peer A for the modification of path A→C, and so does peer C to peer E for path A→C→E.

(9) Both peers A and E update the path information for path A→E→C and return a *Remain Message* to peer C for the modification.

(10) As Figure 3.12 (c) illustrated, peer E sends a *Leave Message* to all peers on the path before leaving the P2P network. Peers A and C update the path information for path A→C as that illustrated in Figures 3.7 and 3.8.

GPMS can manage P2P overlay on MANET to provide file sharing or live streaming service due to the cross-layer scheme and FPRT design. Recall that in Figure 3.3, even if the peers move or leave, FPRT is updated via the path information and service cache to keep the continuous data. The algorithms of GPMS extend the P2P service, reduce the signaling overhead, and improve the recovery efficiency.

3.6. Peer departure and mobility

When a peer leaves the P2P network, it voluntarily informs its neighbors via a *Leave Message*, so that its neighbors can recover the overlay. However, in wireless mobile ad hoc network, the message may get lost. Therefore, in our proposed scheme, all peers rely on *Hello Message*s to maintain indirectly P2P overlay and monitor the neighbors' movements. If the intermediate node leaves, the number of received *Hello Message*s would become smaller and smaller. The degradation informs the neighboring peers to update their path information, and the neighboring peers find another intermediate node to maintain the end-to-end connections. If no intermediate node can support, the link breaks down and the receiver will redetect another sender. Ditto, if the sender disappears, the receiver will redetect another sender because of no incoming data.

If the receiver moves, the sender updates its FPRT and path information according to the proactive *Maintain Message* or reactive *Hello Message*. The sender may reselect the next hop. Similarly, if the receiver disappears, the delivery interruption breaks the socket and the sender will update its FPRT to recover the P2P overlay.

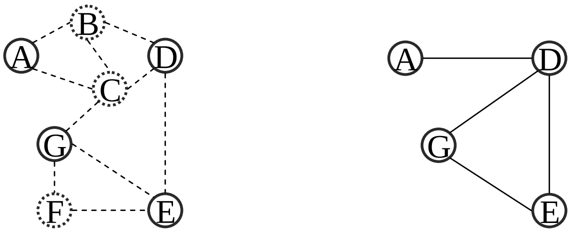
If there are more than two intermediate nodes between two peers, and one of the intermediate nodes is not an immediate neighbor of peers, when this intermediate node leaves the network, peers cannot detect actively this departure via the number of received *Hello Message*. Therefore, peers cannot update actively their path information in advance. Peers reselect a routing path reactively when the stream breaks.

In summary, two kinds of approach can perceive peer leaving or moving: (1) every peer voluntarily informs its neighbors upon leaving. The major purpose of this information is for the neighbors re-deriving the routing paths; (2) every peer also collects the *Hello Message* to monitor the neighbors' movements. The changed QoS represents the neighbors' movement or the variation of bandwidth, thus the notification is unicasted to all peers on the routing path to re-derive the routing.

3.7. Peer adaptation

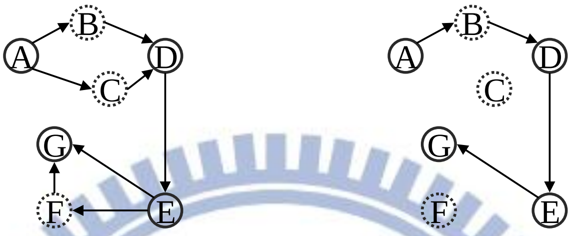
A *peer adaptation* is a key point to guarantee QoS for peers, maintain an optimal overlay, and speed up data delivery in P2P network [28]. Generally speaking, peer adaptation is executed to keep smooth quality when the incoming bit rate decreases or is unstable in real-time service. In wireless P2P, we consider that peer adaptation should be executed when response time becomes longer or SNR becomes lower, because the neighbor may move away. In our proposed scheme, any peer can execute peer adaptation to replace its partner in the upstream or downstream by another peer with larger number of received *Hello Message*. This peer adaptation guarantees that the P2P overlay is proximal to the physical topology, and more advance to detect peer departure.

An ad hoc network topology is shown in Figure 3.13 (a). Node *A*, *D*, *E*, and *G*, form a P2P network, and their overlay is shown in Figure 3.13 (b). Every peer knows its members via its DHT from the overlay. For example, *A* knows *D* but not *E*. If *A* is going to send data to *E*, it will ask its members where *E* is, then it knows that packets can be forwarded through *D* to *E* when *D*



(a) (b)

Figure 3.14: (a) The physical topology of ad hoc network; (b) The P2P overlay



(a) (b)

Figure 3.13: (a) The stream flow via DHT over AODV.
(b) The stream flow via GPMS over cross-layer routing.

answers. By the way, *A* does not know the path *A*→*D*→*E* or *A*→*D*→*G*→*E* for sending packets to *E*.

In the general P2P-MANET scheme, the source decides the logical path to the destination via DHT in the application layer, and AODV decides the routing path on demand. In application layer, both paths *A*→*D*→*E*→*G* and *A*→*D*→*G*→*E* can be implied as a stream path. As illustrated in Figure 3.14 (a) for the path *A*→*D*→*E*→*G*, *A* selects the shortest path via AODV to *D* through *B* or *C* on demand. Similarly, both *D* and *E* also perform AODV to find the shortest path to next hops. This scheme can use AODV to ensure the local optimization, but the stream may be unstable because the path is changed and is selected on demand due to peers' movement.

Our proposed GPMS uses FPRT to form P2P overlay and select routing path based on the real world estimation. The path information in FPRT decides the IPv6 hop-by-hop routing path via the cross-layer design. As a result, GPMS selects the path *A*→*D*→*E*→*G* and every peer can select its next hop in the application layer as illustrated in Figure 3.14 (b).

The traditional DHT can be altered due to peer churn, instead of peer mobility, because

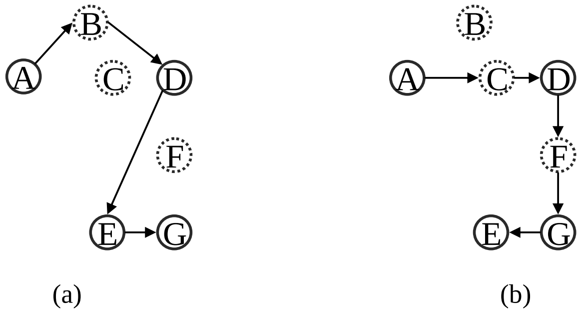
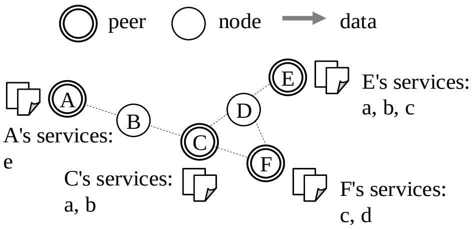


Figure 3.15: (a) The stream flow via DHT over AODV;
(b) The stream flow via GPMS over cross-layer routing.

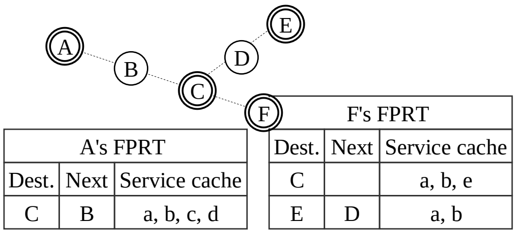
P2P technology is developed from the wired network. Therefore, the general P2P-MANET scheme can find the end-to-end shortest path via AODV, but it is unable to inform DHT to look up a new logical path. As Figure 3.15 (a) shows, several peers move and ad hoc topology changes. In the general P2P-MANET scheme, *A* can find the shortest routing path through *C* to *D* via AODV, but *D* still forwards chunks to *E* via DHT. In our proposed GPMS, FPRT with routing information is altered dynamically and periodically and interacts with network layer, hence *D* forwards chunks to *G* as shown in Figure 3.15 (b). As a result, FPRT can establish an overlay with overlay proximity to select the shortest routing path.

3.8. Anycast algorithm

FPRT caches the service type, so GPMS can accomplish the anycast-based query to extend all P2P services, such as file sharing, voice communication, and video streaming. As Figure 3.16 illustrated, there are peers A, C, E, and F in overlay like a local anycast group. As Figure 3.16 (a) illustrated, every peer owns the local services, e.g. peer A owns the service e, which may be a file, or peers C and E own the services a and b, which may be a voice conference. As Figure 3.16 (b) illustrated, every peer updates its FPRT via querying on demand. If peers cannot communicate within one hop, the intermediate nodes as B and D relay to establish a P2P connection. The connection maintenance is updated in the path information, which is combined with the service cache in FPRT. Although peer A owns only service e, it makes services a, b, c,



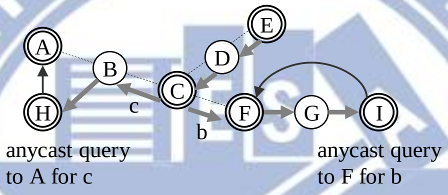
(a) Content availability



A's FPRT		
Dest.	Next	Service cache
C	B	a, b, c, d

F's FPRT		
Dest.	Next	Service cache
C		a, b, e
E	D	a, b

(b) FPRT of GPMS



(c) Anycast-based query

Figure 3.16: FPRT process to service cache and query.

and d available via peer C on P2P overlay.

As Figure 3.16 (c) illustrated, peers H and I start GPMS to join such P2P network. Peer H creates a download task to query for service c, and if peer A receives such *Query Message*, peer A returns a *Reply Message* with the queried result (YES). Because peer H is routable to peer C, the routing path E→D→C→B→H can be derived. On the other hand, peer I creates a download task to query for service b, if peer F receives such *Query Message*, peer F returns a *Reply Message* with the queried result (YES). Because peer I is unroutable to peers C and E, the routing path C→F→G→I can be derived. Of course, the intermediate nodes don't know GPMS, so they just forward the messages via the ad hoc routing protocol. The reactions of *Query Message* and *Reply Message* are described as Figure 3.17 illustrated.

```
when a Query Message is received do
    let QM be such message;
    let RM be a Reply Message;
    if QM.getPeerId( ) $\in$ this.members{ } then
        new a UT to create a upload task;
        UT.setState(Create);
        UT.setPeerId(QM.getPeerId( )) ;
        if QM.getQueryService( ) $\in$ this.FPRT.cache{ } then
            RM.setResult(true);
            RM.setCache(this.FPRT.cache);
            UT.setState(Reply);
        else
            RM.setResult(false);
            UT.setState(Finish);
        RM.setQueryService(QM.getQueryService());
        sends RM to QM.getSource( );
    else
        drop such message;
end when


when a Reply Message is received do
    let RM be such message;
    if RM.getQueryService() $\in$ this.download{ } then
        this.download();
    else
        drop such message;
end when


when download task receives a Reply Message do
    let RM be such message;
    if getState( ) is Pend then
        setState(Query);
        .
        .
        .
        if RM.getResult( ) is true
            setState(Download);
            .
            .
            .
        else
            setState(Create);
            .
            .
            .
    else
        drop such message;

end when
```

Figure 3.17: The proposed algorithms of query process.

In the on-going progress of implementation, we use the anycast concept, but the characteristics of anycast are not all completed. Therefore, the *content-aware* query or *cache-based* query may be suitable for the proposed scheme.

3.9. Dynamic buffer

In order to deliver real-time streaming, we integrate dynamic buffer [a] and delivery strategy [g]. The partner departure will break a delivery path, and a temporary lack of data resource leads to a reboot. Once the reboot happens, the peer will be compelled to execute adaptation or selection again and waits for a long time to acquire the service. In GPMS, the reboot can be avoided via the method in that peer monitors the variation of its data rate, and adjusts the buffer size based on the data rate.

The decrease of data rate often indicates a warning of partner departure or wireless degradation. When the data rate decreases, the size of buffer is enlarged. Thus the buffer early reserves some un-continuous data. In this duration of overlay recovery, collecting sequential data is not the top priority in the schedule of our proposed algorithm. A lack of data source must result in bad QoS during recovery time. Although users experience the un-continuous quality, the matter avoids the reboot. In summary, we have to sacrifice real-time quality of content distribution to minimize the probability or frequency of reboot.

The decrease of data rate also indicates a sudden jitter or bottleneck of wireless link. Peers still keep and maintain the delivery paths or multicast streams to share data chunks between each others. Peers should choose to wait for a quality restoration passively or handle peer adaptation actively. The size of buffer is enlarged to handle the situation of unstable network bandwidth. After peer churn, peer adaptation, overlay recovery, or etc., the buffer shortens to the original size when coming back in the stable quality.

In order to implement the proposed dynamic buffer, we must design a special delivery strategy to coordinate the dynamic buffering mechanism. The traditional delivery strategies, such as Sequential First, Rarest First, and Urgent Region First, are stable for the fixed buffer. We propose Stream Block First delivery strategy, which is based on two principles:

(1) *Oldest first:* Data with smaller sequence number, i.e. non-freshness, is prior to those with

larger sequence number in delivery. This principle can avoid absence of desired data in a swarm.

(2) *Consecution first:* Data with consecutive sequence numbers, i.e. in-order sequence, is prior to those data with non-consecutive sequence number in delivery. This principle can preserve the cluster of wireless peers.

Because of oldest first principle and consecution first principle, Stream Block First needs a priority queue to support the downloading schedule. We design a scoring function F_{score} to calculate and evaluate priorities of all chunks. First, to obey oldest first principle for the first loop:

$$F_{score}(i) = isAccessible \times Age(i) \times W_{old} \tag{1}$$

In the above equation, isAccessible which is either 0 or 1, represents whether the chunk is accessible or not. Age(i) is used to compute the queued time of chunk and outputs a scalar. W_{old} denotes the weighted score for aging.

Second, if someone can provide continuous, sequential and accessible data, the scoring function adds more scores. To obey consecution first principle for the second loop:

$$F_{score}(i) = F_{score}(i + 1) = (Age\ (i) + Age\ (i + 1)) \times W_{block} \tag{2}$$

W_{block} denotes the weighted score for continuous sequence. If $W_{old} > W_{block}$, we consider that the in-time streaming is important than continuous streaming.

It is anticipated that some chunks with small SN or continuous SN can get the high scores and have high priorities to be requested. However, the selected chunks with high score possibly get high scores for all time if they cannot be available. Trial and error of the same chunks hence leads to starvation of other chunks. As a result, we add a *score subtracting mechanism* on Stream Block First strategy. Score subtracting mechanism subtracts a penalty from F_{score} (i) when requesting i misses. The dynamic score is able to resolve the starvation efficiently.

4. Analysis

Via the mathematical equations and theoretical computation, the *routing complexity* and *maintenance load* are used to analyze the efficiency of compared schemes. We analyze the complexity of average cases rather than initial cases or worst cases, because the average performance is mapping to the results of repeating simulated experiments.

4.1. Routing complexity

One P2P scheme combined with one ad hoc routing protocol provides the P2P service on MANET. We use AODV as an example. AODV is a popular hop-by-hop routing protocol, and it always establishes the P2P communication on demand. P2P solution can search data via flooding or DHT query. The flooding query is suitable for dynamic topology, but the DHT query is efficient. We compare GPMS with other schemes including the flood-query and DHT-query over AODV.

In flood-query over AODV, a peer searches a source via flooding the query, whose complexity O(n) is well-known. Every query is forwarded on demand by intermediate node via AODV, whose complexity O(log n) is well-known. Therefore, the routing complexity of flooding over AODV is O(n log n).

In DHT-querying over AODV, a peer searches a source via DHT-based approach, whose complexity is well-known O(log n). The complexity of AODV is well-known O(log n). Therefore, the routing complexity of DHT-querying over AODV is O((log n)^2).

In GPMS, a peer searches a source via FPRT, which inherits DHT, thus its complexity is O(log n). The routing path is derived via FPRT to avoid the choice of on-demand intermediate node. The complexity of IPv6 routing via hop-by-hop routing header is O(1). Therefore, the routing complexity of proposed scheme is O(log n).

4.2. Maintenance load

Maintenance load means that each peer requires the maintenance load of P2P overlay when overlay is changed.

In flood-query over AODV, a peer usually discovers its members for reactive query, so the proactive overlay maintenance is unnecessary. Each peer just checks the arrival or departure of its member during the P2P communications, and the load is O(1) trivially. In addition, the maintenance load of AODV is well-known O(n) due to rediscovery. Therefore, the maintenance load of flooding over AODV is O(n).

In DHT-query over AODV, a peer maintains its DHT for indexing peers and files, and the maintenance load of DHT is well-known O(log n). AODV reestablishes a P2P communication without rediscovery due to the proactive maintenance of DHT, and the reestablishment of hop-by-hop path is well-known O(log n). Therefore, the maintenance load of DHT-querying over AODV is O((log $n)^2$).

In GPMS, a peer maintains its FPRT for path information and service cache. Both updates of path information and service cache cost O(log n). Because FPRT shares the finger table and routing table, the maintenance of ad hoc routing is O(1). Therefore, the maintenance load of GPMS is O(log n). In summary, the routing complexity and maintenance load are summarized as Table 4.1 illustrated.

4.3. Routing overhead

Table 4.1: A summary of P2P over MANET schemes.

	Routing complexity	Maintenance load
Flood-querying over AODV	O(n log n)	O(n)
DHT-querying over AODV	O((log $n)^2$)	O((log $n)^2$)
The proposed cross-layer	O(log n)	O(log n)

In our proposed forwarding scheme, we can assume a probability that the i^{th} peer creates a multimedia packet in a given time t is p. The event b_i represents the process among n peers, and a sequence $b_1, b_2, b_3, \ldots$ can be defined via Bernoulli formula:

$$\begin{aligned} &\mathrm{P}(b_i = 1) = \mathrm{P}(i^{\text{th}} \text{ peer speaks}) = p, \forall\ i \text{ and } i \in n \\ &\mathrm{P}(b_i = 0) = \mathrm{P}(i^{\text{th}} \text{ peer calms}) = 1 - p \end{aligned} \tag{3}$$

At some time t, the probability that k peers create multimedia packets simultaneously is $\mathrm{P_K}$.

$$\mathrm{P_K}(k) = \boldsymbol{C}_k{}^n\, p^k\, (1 - p)^{n - k} \tag{4}$$

Therefore, the probability that a peer creates h packets in m time slots is $\mathrm{P_H}$ is similar to Equation 4 theoretically.

$$\mathrm{P_H}(h) = \boldsymbol{C}_h{}^m\, p^h\, (1 - p)^{m - h} \tag{5}$$

For one peer, t_j can be defined that j^{th} packet is created in an inter-arrival time. Then the probability that a peer creates h packets in a given time t is $\mathrm{P_T}$.

$$\begin{aligned} \mathrm{P_T}(t) &= \mathrm{P}(b_i \text{ is true AND } h - 1 \text{ packets are created} \\ &\qquad \text{in the first } t - 1), \text{ for each } i \\ &= \mathrm{P}(b_i = 1) \cdot \mathrm{P_H}(h - 1), \text{ while based on } t - 1 \text{ for } t \geq h \\ &= p \cdot \boldsymbol{C}_{h-1}{}^{t-1}\, p^{h-1}\, (1 - p)^{t-h} \\ &= \boldsymbol{C}_{h-1}{}^{t-1}\, p^{h}\, (1 - p)^{t-h} \end{aligned} \tag{6}$$

And the expected time of h packets is $\mathrm{E_H}(h)$.

$$\begin{aligned} \mathrm{E_H}(h) &= \mathrm{E}(t_1) + \mathrm{E}(t_2) + \ldots + \mathrm{E}(t_h), \text{ E is expected function} \\ &= 1/p + 1/p + \ldots + 1/p = h/p \end{aligned} \tag{7}$$

The Bernoulli formal can be applied to an analog of Poisson process in a continuous time. We can define that the rate of packet created is λ in a given time t. The probability that k peers create packets in t is $\mathrm{P_N}(k, t)$.

$$\mathrm{P_N}(k, t) = \mathrm{e}^{-\lambda t}\, (\lambda t)^k / k!, \text{ while } \lambda t = h = \mathrm{E_H}(h) \times p \tag{8}$$

Therefore, the expected value of duplicated packets is simplified as $\mathrm{E_N}$.

$$\mathrm{E_N}(n) = \Sigma_k\, \mathrm{P_N}(k, t) \times k \times n \tag{9}$$

The fixed λ, t, n can derive the $E_N(n)$ for the low bound of packet duplications in the proposed hop-by-hop forwarding scheme.

4.4. Compared schemes

File sharing, voice communication, and video streaming are the live applications of P2P over MANET. According to the analysis of routing complexity, maintenance load, and routing overhead, we summarize the compared schemes as Table 4.2 illustrated, and list the descriptions.

- ORION (Optimized Routing Independent Overlay Network) [29] is based on an application layer overlay combined with the reactive ad hoc routing protocol for file sharing to guarantees high data rates and low transmission overhead over MANET. ORION integrates Gnutella [6] query processing and overlay network construction with AODV routing discovery. Gnutella is a flooding base, so ORION belongs to the flood-querying over AODV.
- MADPastry (Mobile AD hoc Pastry) [15] is a file sharing application, and it belongs to the DHT-querying over AODV. MADPastry uses Pastry [9] proximity awareness to reduce the overhead without flooding. Pastry routing table indexes and hashes the mobile nodes.
- ACT (Audio Conferencing Testbed) [18] is based on OLSR to set up a meeting for group of people in WiFi MANET. ACT uses the minimum spanning tree to minimize the latency of audio dissemination to the whole network. Every peer must maintain its minimum spanning tree to deliver data by itself. ACT predicts the disconnection and mobility to

Table 4.2: The comparison of schemes.

	File sharing	Voice communication	Video streaming
Flood-querying over AODV/ OLSR	ORION	ACT	Smart Gnutella
DHT-querying over AODV/ OLSR	MADPastry	CLAPS	MP2PS

shorten the service interruption time.

- CLAPS (Cross-Layer And P2P based Solution) [19] is based on the combination of tree-based overlay and OLSR extension. CLAPS assumes that the physical routing topology can be provided by OLSR, which sends cross-layer message to optimize overlay. The source peer maintains a minimum spanning tree as its overlay. The minimum cost is computed via *link distance* packaged in cross-layer message, and the spanning tree is recomputed periodically to keep the overlay proximity.
- Smart Gnutella [20] enhances the original Gnutella to suit for MANET and for the real-time application. Because the periodic *ping*/ *pong* and *broadcast* can assist in routing maintenance, Smart Gnutella adopts AODV as the ad hoc routing protocol. Therefore, Smart Gnutella belongs to the flood-querying over AODV.
- MP2PS (Mesh-based P2P Streaming) [21] is a live streaming application, and it belongs to the DHT-querying over AODV. MP2PS provides real-time streaming with scalability and availability over MANET. MP2PS adopts mesh-based live streaming application, Joost [30], and no retransmission on wireless network, UDP, and on-demand ad hoc routing protocol, AODV.

5. Results

We use OMNet++ 4.0 [31] to simulate the P2P service over WiFi, and use INET [32] to simulate the network behavior including organizing an ad hoc topology and handling messages. We use MF (Mobility Framework) module [33] to simulate wireless MAC layer and use OverSim module [34] to simulate P2P overlay layer. The simulation adopts RTS/CTS mechanism and random waypoint model. 120 mobile nodes move inside a bounded area of 1000 m × 1000 m, and the radio transmission range of a node is 125 ± 25 m. Although the ideal bandwidth of WiFi standard is 11 Mbps, the estimated value of our measurement is less than 2 Mbps under ad hoc model in reality. As a result, we set the bandwidth 1000 ± 200 kbps without any fading interference, but we use the degradation function[24] provided by INET module. Every statistic value of simulation result is the average of 30 repeated experiments.

In random waypoint model, we define every wireless node with parameters (x, y, π, v, s) as Figure 5.1 shown. The parameters (x, y) represent the coordinate of a node in the simulation square. π represents the moving angle of each node. If a node crosses the simulated boundary, π becomes a reflective angle as node *b* to ensure that every node moves in the simulation square. v represents the moving speed of each node. When $v = 0$, it means that the node is static as node *c*. If s = true, the node is a member of P2P overlay as node *a*; otherwise, the node is just a member of MANET topology as node *c*. We list the parameters in Table 5.1.

5.1. File sharing

We characterize the download strategy and file factors on BitTorrent [35] to build a simulated benchmark. The size of a file piece is 256 kbytes and the packet size is 512 bytes, and the data delivery mechanism is best effort. The *download completion time* is a major metric of file sharing, and it means how much time a file is finished downloading. We evaluate it via the

---

[24] The packet loss rate is influenced by the degradation function.

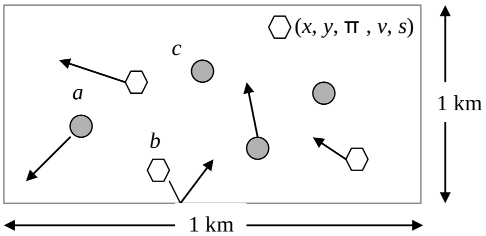


Figure 5.1: The simulation environment.

network size, moving speed, and churn rate in a many-to-many delivery scenario.

As Figure 5.2 illustrated, the proposed GPMS has the shortest completion time among the compared schemes. Because GPMS and MADPastry inherit the DHT query, the querying latency of GPMS and MADPastry is shorter than that of ORION. GPMS uses not only DHT-based index but also anycast-based query, so the querying latency of GPMS is shorter than that of MADPastry and ORION. After finishing query or constructing overlay, the AODV routing discovery or connection must proceed in MADPastry and ORION. However, the routing path can be derived from FPRT due to the integration of finger and routing table in GPMS, so GPMS shortens the completion time. The completion time is lengthened gracefully with the increasing network size, so the proposed GPMS is demonstrated with scalability.

As Figure 5.3 illustrated, although the completion time is lengthened with the increasing moving speed, the service is still workable, so the proposed GPMS is demonstrated with mobility. GPMS can detect the peer movement via the accumulation of *Hello Message* to

Table 5.1: The parameters of simulation

Parameter	Type	Range	Default	Description
x	natural number	0 ~ 1000 m	random	x-axis
y	natural number	0 ~ 1000 m	random	y-axis
π	decimal	0 ~ 360^o	random	moving angle
v	natural number	0 ~ 36 m/s	0	moving speed
s	binary number	true of false	false	turn on or not
n	natural number	10 ~ 120	100	number of peers
f	real number	0 ~ 1	0.02	failed probability

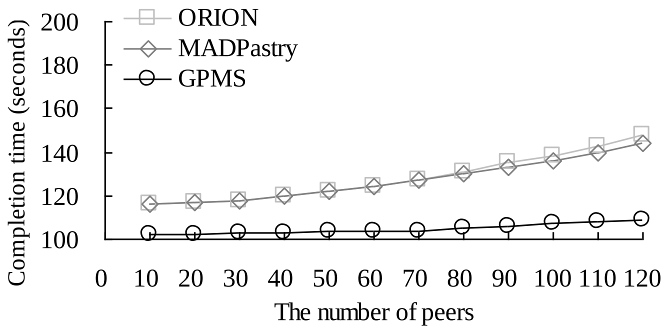


Figure 5.2: Download completion time vs. the number of nodes.

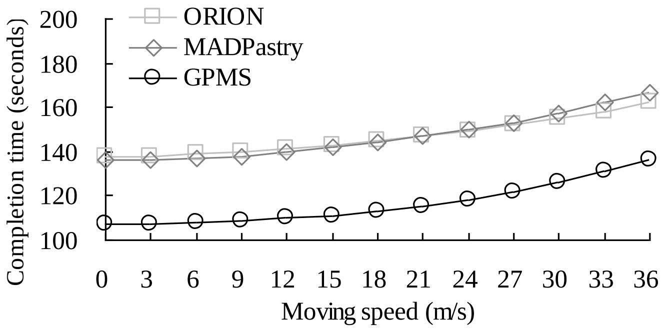


Figure 5.3: Download completion time vs. the moving speed.

update path information in FPRT due to the cross-layer scheme. The update of path information not only recovers the routing path but also improves the overlay proximity, which avoids the far routing problem. Although the peers periodically probe their neighbors in MADPastry and ORION, the non-optimized end-to-end path cannot be avoided. The routing path may be disconnected, and AODV reestablishes the routing discovery, which lengthens the service interruption latency.

The peer arrival/ departure may change the P2P overlay structure, and we define that the *churn rate* is the probability of peer arrival/departure. Given a random interval, every peer stays or leaves depending on the churn rate. As Figure 5.4 illustrated, the proposed GPMS has the shortest completion time among the compared schemes. Because GPMS and MADPastry inherit the DHT maintenance, every peer can maintain its members to recover a broken overlay. Every peer builds the local overlay on demand in ORION, so the overlay is suitable for a

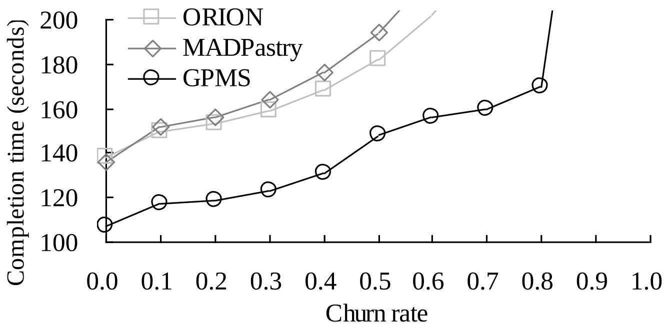


Figure 5.4: Download completion time vs. the churn rate.

dynamic environment. Of course, the parts of file become unavailable because the source leaves, so the completion time become much longer or the file cannot be downloaded completely when the churn rate is high. When the churn rate is 0.5, the completion time is 148 seconds in GPMS, so the proposed GPMS is demonstrated with recoverability.

5.2. Voice Communications

We add up the number of voice packets which are transmitted through the ad hoc network as Figure 5.5 illustrated. We can find out that GPMS, ACT, and CLAPS all have the similar performance, thus P2P solution can be demonstrated in an efficient delivery with scalability. Especially, GPMS has the performance approximated to an ideal $E_N(n)$ as in Equation (9), because every peer only copies once for its rear peer and every intermediate node only copies once for the destination.

The *packet failure rate* is a major metric of live streaming, and it is defined as the number of multimedia chunks that lose or arrive before playback deadlines over the total number of multimedia chunks. As Figure 5.6 illustrated, GPMS has the lowest packet failure rate among all compared models. However, the large network leads to the long routing path, which leads to high packet failure rate. ACT and CLAPS have the high packet failure rate, because the collision happens in the overlap of spanning trees when many peers simultaneously speak. Especially CLAPS overlay tends to share the overlapping routing path, thus its packet failure

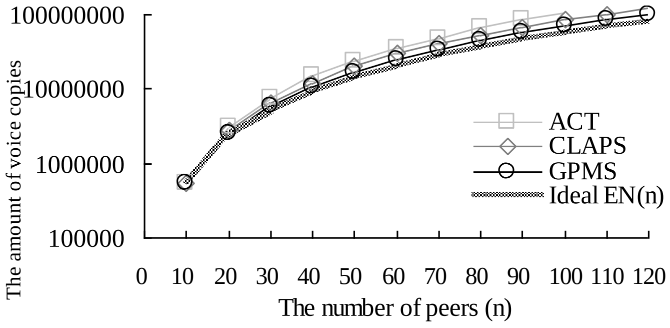


Figure 5.5: The amount of voice copies increases with the number of peers.

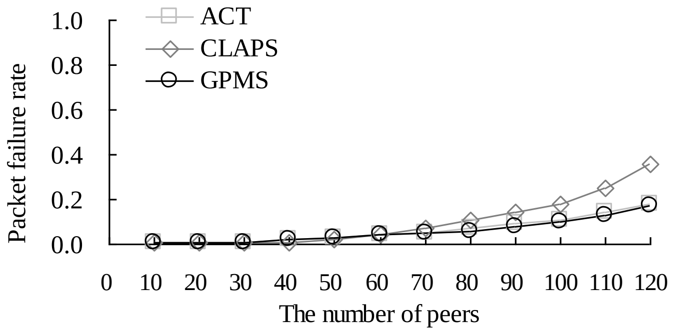


Figure 5.6: Packet failure rate vs. the number of nodes.

rate raises seriously with scalability.

The simulation result about mobility is curved in Figure 5.7, and we can discover that GPMS has the lowest packet failure rate among the compared models. The cross-layer scheme detects peer movement to avoid the far routing path. The cross-layer scheme always performs time sensitivity of neighborhood to keep proximity.

As Figure 5.8 illustrated, the proposed GPMS has the lowest packet failure rate among the compared schemes. Because the cross-layer design speeds up recovery latency and improves streaming stability, voice packets can be redirected through multiple intermediate nodes. FPRT is always updated to recover backup routing path, but ad hoc routing disconnection or rediscovery on OLSR may be conflicted with overlay recovery in ACT and CLAPS. When the churn rate is higher than 0.8, three models work impossibly, so the broken delivery path hurts VoIP service seriously.

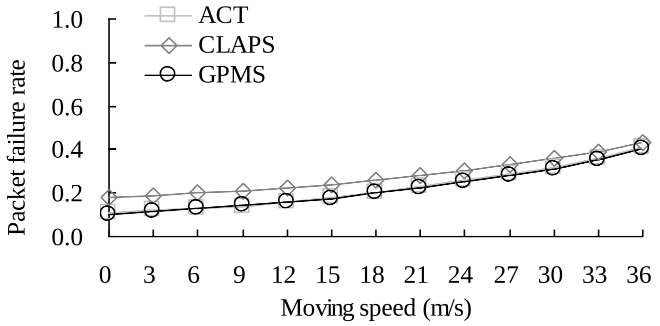


Figure 5.7: Packet failure rate vs. the moving speed.

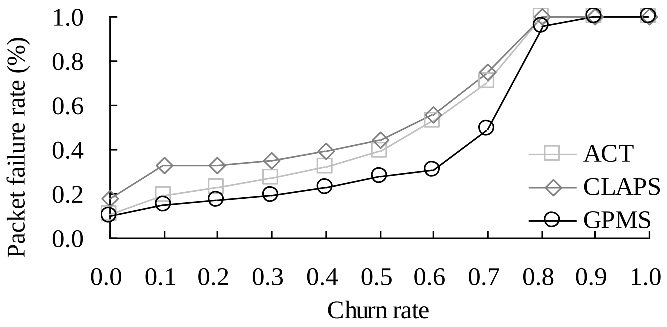


Figure 5.8: Packet failure rate vs. the churn rate.

As Figure 5.9 illustrated, in GPMS, even if the overlay changes, the moving peer always influences only neighboring peers. Therefore, the low speed does not cause the high packet loss. However, the high speed changes the overlay drastically, which leads to the continuous and frequent packet loss. Although packet failure rate is rose with moving speed, GPMS still keeps the sequential receipt. Because the hop-by-hop routing path inherits the unidirectional stream, the voice arrives in order even if multiple peers speak simultaneously.

5.3. Video streaming

In order to build a simulated benchmark, we characterize the delivery principle, the overlay construction, and the streaming factors on Joost [36] referred to MP2PS. The video packet size is 1024 bytes, and the data rate of streaming is 450 kbps (constant bit rate). As Figure 5.10 illustrated, the network stacks of compared schemes are described to differentiate between

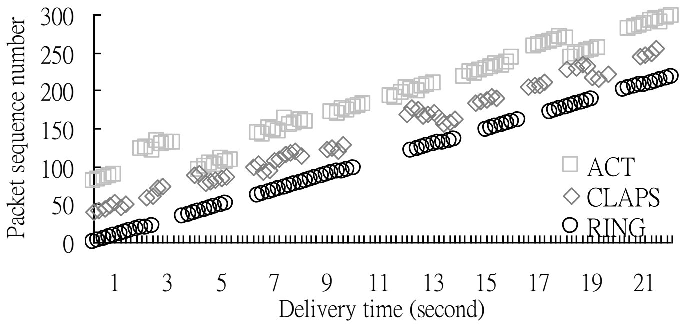


Figure 5.9: The packet loss at v = 36 m/s.

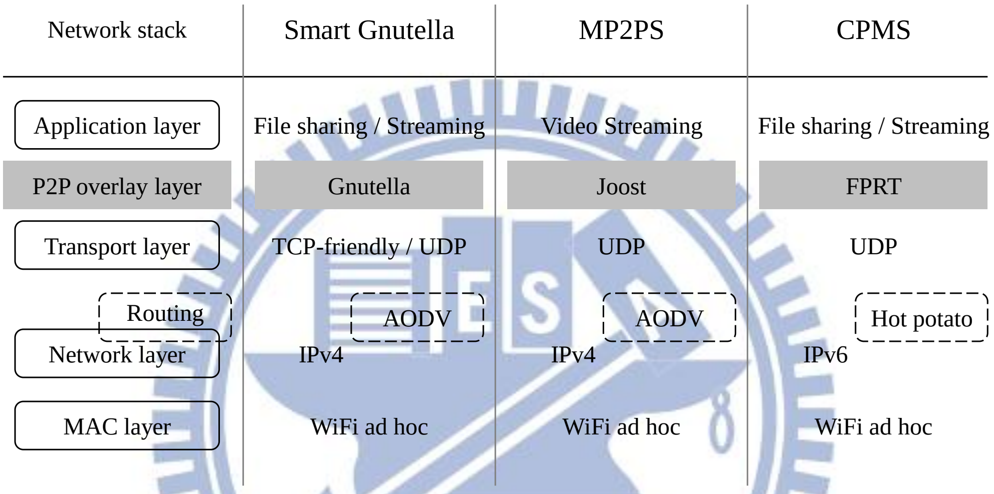


Figure 5.10: The compared network stacks.

network features.

As Figure 5.11 illustrated, the proposed GPMS has the lowest packet failure rate among the compared schemes. When the number of nodes increases to 120, the failure rate is 0.07, so the proposed GPMS is demonstrated with scalability. However, MP2PS cannot support high scalability because the overlay forwarding is not mapped into the ad hoc routing. The larger number of peers is relative to the more routing paths, so the flooding query degrades the streaming continuity in Smart Gnutella.

In wireless network, the mobility problem leads to a difficulty of keeping a continuous playback. As Figure 5.12 illustrated, the cross-layer design can be appropriate for the mobility. Due to the overlay maintenance and overlay proximity of GPMS, it can maintain a low failure

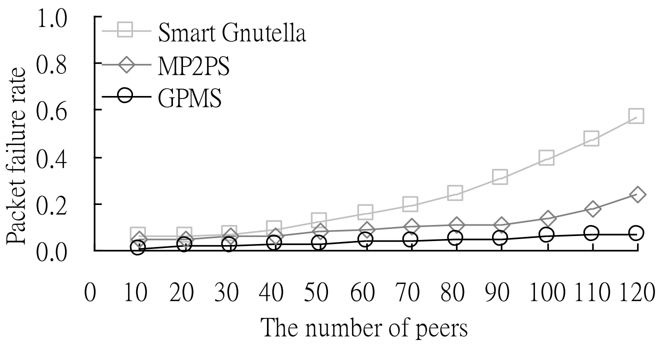


Figure 5.11: Packet failure rate vs. the number of peers.

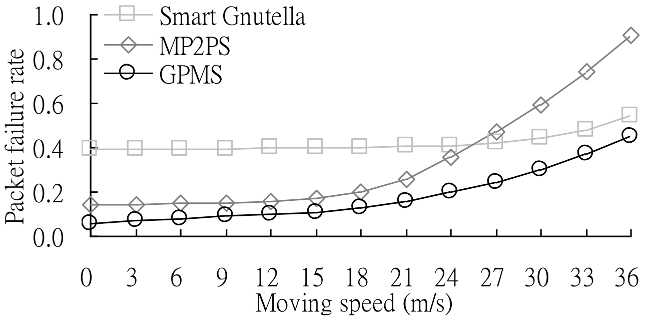


Figure 5.12: Packet failure rate vs. the moving speed.

rate and low overhead when mobile peers move. Smart Gnutella also has stable performance against the moving speed, because the moving speed affects the flooding scheme slightly. However, DHT originally is designed for wired network, so MP2PS is unable to detect peer movement, such that the peer cannot find its nearest neighbor to forward stream. Under the high moving speed, GPMS highlights the advantage for mobility.

For live streaming, the peer churn leads to a difficulty of keeping a stable overlay. As Figure 5.13 illustrated, the proposed GPMS is suitable for a dynamic MANET, because the cross-layer design speeds up recovery latency and improves streaming stability. FPRT is always updated to recover backup routing path, but ad hoc routing disconnection or rediscovery on AODV may be conflicted with overlay recovery in Smart Gnutella and MP2PS. Therefore, the packet failure rate is only 0.25 in GPMS when the churn rate is 0.5, however, that is 0.44 and 0.78 in Smart Gnutella and MP2PS respectively. GPMS is suitable for dynamic network

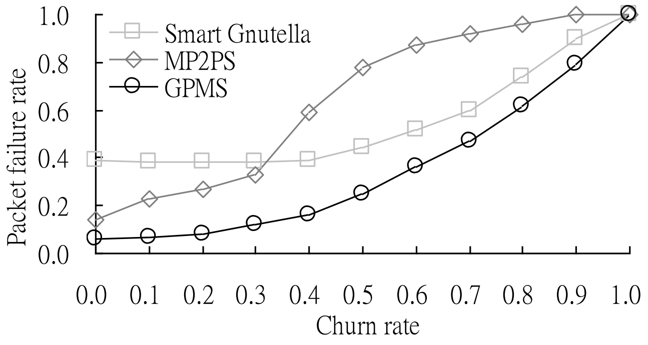


Figure 5.13: Packet failure rate vs. the churn rate.

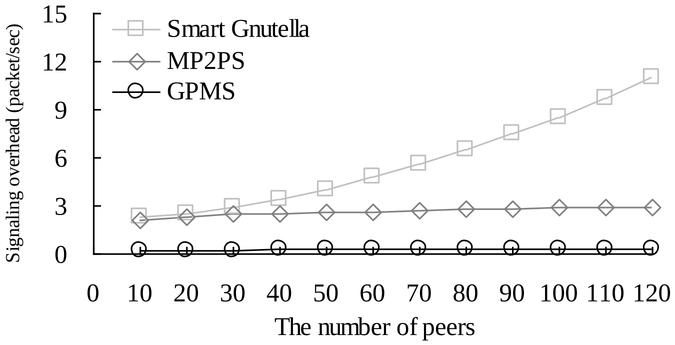


Figure 5.14: Signaling overhead vs. the number of peers.

with high churn rate.

5.4. Signaling overhead

The *signaling overhead* includes routing overhead and overlay overhead, so we define the signaling overhead as the number of non-data packets sent per peer per second. As Figure 5.14 illustrated, the proposed GPMS has the lowest signaling overhead among the compared schemes, and we discovery several observations.

- In Smart Gnutella, the major overhead is routing (especially querying) overhead, because the on-demand flooding query leads to high overhead. The signaling overhead increases linearly with network size, and it lacks scalability. On the other hand, the signaling overhead of DHT-based schemes (such as MP2PS and GPMS) is stable with scalability.
- In MP2PS, the major overhead is routing overhead, because the routing discovery needs

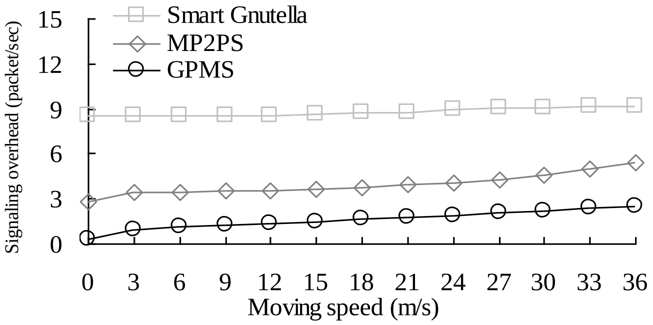


Figure 5.15: Signaling overhead vs. the moving speed.

an additional overhead to plot the DHT-based forwarding.

- GPMS reduces much signaling overhead, because GPMS detects peer churn and movement via the accumulation of received *Hello Message*, which needn't any modification or overhead. The proposed anycast-based query also reduces the signaling overhead, because anycast minimizes the number of querying hops.

As Figure 5.15 illustrated, GPMS has the lowest signaling overhead among the compared schemes, because FPRT is updated without additional measurement. However, GPMS sometimes mistakes the peer movement for peer churn. On the other hand, in Smart Gnutella, because a peer searches its members when querying real-time data on demand, the moving speed affects the signaling overhead slightly.

As Figure 5.16 illustrated, the signaling overhead of GPMS linearly increases when churn rate rises. In the proposed scheme, when a peer joins, its neighbors inform each other to update their FPRTs and derive the routing path; and when a peer leaves, it proactively informs its neighbors. As a result, GPMS generates the high signaling overhead at high churn rate. Similarly, MP2PS faces the difficulty of peer churn, but it is unable to derive routing path in advance to economize signaling overhead.

There are three disadvantages of Smart Gnutella and MP2PS: inefficient routing for live high-bit-rate streaming, high on-demand routing overhead, low mobility and far routing problem. The simulation results demonstrate that the cross-layer integration of FPRT and IPv6

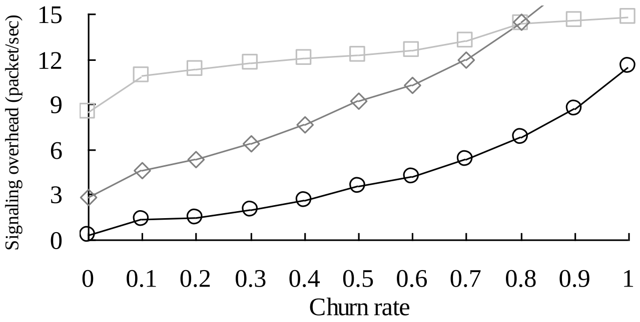


Figure 5.16: Signaling overhead vs. the churn rate.

can avoid the disadvantages. Both Smart Gnutella and MP2PS cannot change the logical link on overlay, so the schemes perform AODV to visit many unnecessary intermediate nodes.

Fast moving speed and high churn rate cannot produce much overhead to crash GPMS. The proposed FPRT really maintains neighborhood and interacts cross layers for deriving routing path. Via integrated with anycast and routing header in IPv6 routing, the signaling overhead is less than 0.5% over total streaming traffic. The overlay proximity is suitable for dynamic MANET and improves routing efficiency.

6. Discussion

In this paper, we investigate how to integrate generic P2P applications and improve the performance of streaming service on wireless ad hoc network, thus we present a cross-layer design for P2P application in MANET. The proposed scheme, GPMS (Generic P2P-MANET System) integrates the routing protocol with overlay protocol for all P2P services on the dynamic wireless network.

6.1. Conclusion

A combination of the proposed FPRT (Finger Plus Routing Table), path information, and IPv6 routing protocol can manage neighboring peers and derive the routing path for reactive query or proactive delivery due to logical overlay proximal to physical topology. Therefore, the proposed scheme can provide the stable routing path to improve download delay, playback continuity, and low signaling overhead for high data rate and low latency in P2P service when facing scalability, mobility, or churn with the reasonable overhead.

GPMS uses a cross-layer middleware to integrate DHT-based lookup, anycast-based query, and P2P delivery via the IPv6 routing header. We demonstrate that GPMS has outstanding performance on routing efficiency and traffic overhead for high bit-rate delivery via both the experimental discussions. Depending on the mathematical analysis, GPMS has the logarithmical increase of complexity and overhead with scalability; depending on the simulation results, GPMS has the best performance among the compared schemes.

As the discussion of simulation shown, GPMS accommodates the large number of users for scalability, handles the high churn rate for recoverability, detects peer movement for mobility, integrates P2P with MANET for adaptability, supports the all kinds of P2P services for extensibility, and supplies different resources for interoperability. Although this proposed scheme cannot be suitable for IPv4 network nowadays, an integration with mobile IPv6 will be an extension in the future.

6.2. Future work

In the future, we consider a *content-aware* overlay and an admission control to save energy and shorten packet delay. The extension of GPMS should be suitable for the wireless half-duplex connections to alleviate the multi-source problem. The performance of QoS metrics is satisfiable, for examples, low packet duplication and short source-to-end delay is still improved, however, the main challenge is to provide mobile users moving in large scale or high speed with satisfactory QoE for a variety of applications. Moreover, because the network metrics are unsupportable for the proposed delivery strategy, the combination of chunk scheduling and source selecting should be considered to avoid the continuous interruption.

We also discuss the improvement of GPMS. When a good query leads to searching a powerful source, the transmission is efficient and the propagation delay is short. When a good transmission leads to deriving an efficient routing path, the routing path is suitable and satisfiable for the advantage of overlay proximity. We will discuss the relation between query and transmission, and we will improve the overlay organization of GPMS.

7. Appendix

According to above P2P-MANET taxonomy, we survey and classify the existing protocols via the categorization as Figure 7.1 illustrated. In Table 7.1, we list these protocols with their proposed date in order, and discuss to solve the problems of continuity, scalability, mobility, and proximity.

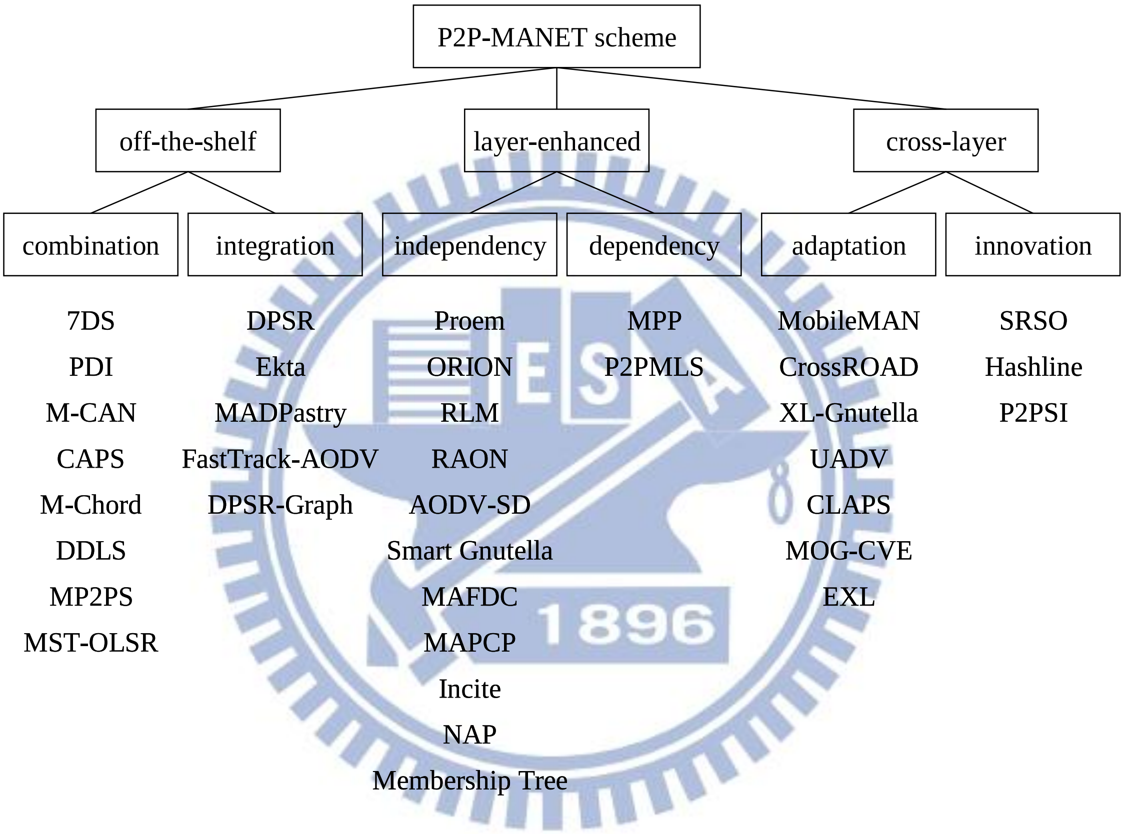


Figure 7.1: The categories of P2P-MANET protocols.

Table 7.1: The P2P-MANET protocols in the survey.

C = Continuity, S = Scalability, M = Mobility, P = Proximity

Name	Year	Application	Taxonomy	C	S	M	P
7DS	Apr. 2001	Web sharing	Off-the-shelf combination		✓	✓	
Proem	Aug. 2001	File sharing	Layer-enhanced independency	✓			
PDI	Jul. 2002	File searching	Off-the-shelf combination		✓	✓	

Name	Year	Application	Taxonomy	C	S	M	P
DPSR	Jun. 2003	Ad hoc routing	Off-the-shelf integration	✓	✓		
ORION	Oct. 2003	File sharing	Layer-enhanced independency	✓	✓	✓	
MPP	Oct. 2003	File sharing	Layer-enhanced dependency	✓	✓	✓	✓
MobileMAN	Mar. 2004	File searching	Cross-layer adaptation	✓			✓
M-CAN	May 2004	File sharing	Off-the-shelf combination	✓	✓		
RLM	Nov. 2004	File searching	Layer-enhanced independency	✓	✓		✓
Ekta	Dec. 2004	File searching	Off-the-shelf integration	✓	✓	✓	✓
CAPS	Feb. 2005	File sharing	Off-the-shelf combination		✓	✓	
CrossROAD	Mar. 2005	Service discovery	Cross-layer adaptation				✓
RAON	Mar. 2005	File searching	Layer-enhanced independency		✓	✓	
XL-Gnutella	Mar. 2005	File sharing	Cross-layer adaptation	✓	✓	✓	✓
AODV-SD	Jun. 2005	File searching	Layer-enhanced independency		✓		
MADPastry	Jul. 2005	Peer lookup	Off-the-shelf integration	✓	✓	✓	✓
FastTrack-AODV	Aug. 2005	File sharing	Off-the-shelf integration		✓	✓	
M-Chord	Jan. 2006	File sharing	Off-the-shelf combination	✓	✓		
Smart Gnutella	Oct. 2006	Overlay managing	Layer-enhanced independency	✓	✓	✓	
DPSR-Graph	Dec. 2006	Ad hoc routing	Off-the-shelf integration	✓	✓	✓	
MAFDC	Jan. 2007	File sharing	Layer-enhanced independency		✓		
MAPCP	Feb. 2007	File sharing	Layer-enhanced independency		✓	✓	
DDLS	May 2007	Web sharing	Off-the-shelf combination		✓		
P2PMLS	May 2007	Live streaming	Layer-enhanced dependency	✓		✓	
UADV	Aug. 2007	File searching	Cross-layer adaptation			✓	
SRSO	Oct. 2007	File sharing	Cross-layer innovation	✓	✓		
MP2PS	Nov. 2008	Live streaming	Off-the-shelf combination		✓	✓	
Hashline	Jan. 2009	File sharing	Cross-layer innovation	✓	✓		✓
Incite	Jan. 2009	File sharing	Layer-enhanced independency		✓		

Name	Year	Application	Taxonomy	C	S	M	P
CLAPS	Jan. 2009	Live streaming	Cross-layer adaptation	✓	✓	✓	✓
NAP	May 2009	File sharing	Layer-enhanced independency				
P2PSI	Jun. 2009	File sharing	Cross-layer innovation		✓	✓	
Membership Tree	Sep. 2009	Overlay managing	Layer-enhanced independency	✓	✓	✓	
MST-OLSR	Jun. 2010	File sharing	Off-the-shelf combination	✓	✓	✓	✓
MOG-CVE	Jul. 2010	Overlay managing	Cross-layer adaptation	✓	✓	✓	
EXL	Dec. 2010	File sharing	Cross-layer adaptation	✓	✓	✓	

C = Continuity, S = Scalability, M = Mobility, P = Proximity

Reference

[1] T. Biermann, L. Scalia, C. Choi, H. Karl, and W. Kellerer. *CoMP clustering and backhaul limitations in cooperative cellular mobile access networks*. Pervasive and Mobile Computing, volume 8, issue 5, 2012, pages 662–681.

[2] S. Sesay, Z. Yang, and J. He. *A survey on mobile ad hoc wireless network*. Information Technology Journal, volume 3, issue 2, 2004, pages 168–175.

[3] G. Anastasi, E. Borgia, M. Conti, E. Gregori, and A. Passarella. *Understanding the real behavior of Mote and 802.11 ad hoc networks: an experimental approach*. Pervasive and Mobile Computing, volume 1, issue 2, 2005, pages 237–256.

[4] A. Mawji, H. Hassanein, and X. Zhang. *Peer-to-peer overlay topology control for mobile ad hoc networks*. Pervasive and Mobile Computing, volume 7, issue 4, 2011, pages 467–478.

[5] N. Leibowitz, M. Ripeanu, and A. Wierzbicki. *Deconstructing the Kazaa network*. IEEE Workshop on Internet Applications, 2003, pages 112–120.

[6] M. Ripeanu. *Peer-to-Peer Architecture Case Study: Gnutella Network*. International Conference on Peer-to-Peer Computing, 2001, pages 99–100.

[7] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker. *A scalable Content-Addressable Network*. ACM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, 2001.

[8] I. Stoica, R. Morris, D. Karger, M.F. Kaashoek, and H. Balakrishnan. *Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications*. ACM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, 2001, pages 149–160.

[9] A. Rowstron and P. Druschel. *Pastry: Scalable, Distributed Object Location and Routing for Large-Scale Peer-to-Peer systems*. IFIP/ACM International Conference on Distributed Systems Platforms, 2001.

[10] C. Perkins and P. Bhagwat. *Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers*. ACM Conference on Communications Architectures, Protocols and Applications, 1994, pages 234–244.

[11] T. Clausen, P. Jacquet, A. Laouiti, P. Muhlethaler, A. Qayyum, and L. Viennot. *Optimized Link State Routing Protocol for Ad Hoc Networks*. IEEE International Multitopic Conference, 2001, pages 62–68.

[12] C. Perkins and E. Royer. *Ad hoc On-demand Distance Vector Routing*. IEEE Workshop on Mobile Computing Systems and Applications, 1999, pages 90–100.

[13] D.B. Johnson, D.A. Maltz and J. Broch. *DSR: the dynamic source routing protocol for multi-hop wireless ad hoc networks*. Ad Hoc Networking, 2001.

[14] G. Peng, S. Li, H. Jin, and T. Ma. *M-CAN: a Lookup Protocol for Mobile Peer-to-Peer Environment*.

International Symposium on Parallel Architectures, Algorithms and Networks, 2004, pages 544–549.

[15] T. Zahn and J. Schiller. *MADPastry: A DHT Substrate for Practicably Sized MANETs*. Workshop on Applications and Services in Wireless Networks, 2005.

[16] M. Li, E.Chen, and P.C. Sheu. *A Chord-Based Novel Mobile Peer-to-Peer File Sharing Protocol*. Asia-Pacific Web Conference on Frontiers of WWW Research and Development, 2006, pages 806–811.

[17] S.A. Baset and H. Schulzrinne. *An analysis of the Skype peer-to-peer Internet telephony protocol.* IEEE International Conference on Computer Communications, 2006, pages 23–33.

[18] W. Wang, B. Soong, and Y. Chan. *Design and Implementation of Audio Conferencing Testbed based on Wi-Fi MANET*. IEEE Wireless Communications and Networking Conference, 2009, pages 1–6.

[19] N.B. Niraula, K. Kanchanasut, and A. Laouiti. *Peer-to-Peer Live Video Streaming over Mobile Ad Hoc Network*. International Conference on Wireless Communications and Mobile Computing: Connecting the World Wirelessly, 2009, pages 1045–1050.

[20] A. Boukerche, A. Zarrad, and R.B. Araujo. *A Smart Gnutella Overlay Formation for Collaborative Virtual Environments over Mobile Ad-Hoc Networks*. IEEE International Symposium on Distributed Simulation and Real-Time Applications, 2006, pages 143–156.

[21] N.N. Qadri, M. Alhaisoni, and A. Liotta. *Mesh Based P2P Streaming Over MANETs*. International Conference on Advances in Mobile Computing and Multimedia, Nov. 2008.

[22] T. Zahn and J. Schiller. *Designing structured peer-to-peer overlays as a platform for distributed network applications in mobile ad hoc networks*. Computer Communications, volume 31, issue 3, 2008, pages 643–654.

[23] D. Bottazzi, R. Montanari, and G. Rossi. *A self-organizing group management middleware for mobile ad-hoc networks*. Computer Communications, volume 31, issue 13, 2008, pages 3040–3048.

[24] C.C. Chou, D.S.L. Wei, C.C.J. Kuo, and K. Naik. *An efficient anonymous communication protocol for peer-to-peer applications over mobile ad-hoc networks.* IEEE Journal on Selected Areas in Communications, volume 25, issue 1, 2007, pages 192–203.

[25] A. Marques, F.M. Silva, and R. Rocha. *P2P over mobile ad-hoc networks*. Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks Workshops, 2009, pages 1–3.

[26] S. Weber and L. Cheng. *A survey of anycast in IPv6 networks*. IEEE Communications Magazine, volume 42, issue 1, 2004, pages 127–132.

[27] X. Hei, Y. Liu, and K.W. Ross. *Inferring Network-Wide Quality in P2P Live Streaming Systems.* IEEE Journal on Selected Areas in Communications, volume 25, issue 9, 2007, pages 1640–1654.

[28] S. Xie, B. Li, G.Y. Keung, X. Zhang. *Coolstreaming: Design, theory, and practice.* IEEE Transactions on Multimedia, volume 9, issue 8, 2007, pages 1661–1671.

[29] A. Klemm, C. Lindemann, and O. P. Waldhorst. *A Special-Purpose Peer-to-Peer File Sharing System for*

Mobile Ad Hoc Networks. IEEE Vehicular Technology Conference, 2003, pages 2758–2763.

[30] M. Alhaisoni, and A. Liotta. *Characterization of signaling and traffic in Joost*. Peer-to-Peer Networking and Applications, volume 2, issue 1, 2009, pages 75–83.

[31] *OMNET++ 4.0*. http://www.omnetpp.org.

[32] *INET Framework*. http://inet.omnetpp.org.

[33] *Mobility Framework for OMNeT++*. http://mobility-fw.sourceforge.net/ hp/index.html.

[34] *OverSim for OMNeT++*. http://www.oversim.org.

[35] J.A. Johnsen, L.E. Karlsen, and S.S. Birkeland. *Peer-to-peer networking with BitTorrent*. Department of Telematics, NTNU, 2005.

[36] J. Lei, L. Shi, and X. Fu. *An experimental analysis of Joost peer-to-peer VoD service*. Peer-to-Peer Networking and Applications, volume 3, issue 4, 2010, pages 351–362.

Publication

[a] J.L. Kuo, C.H. Shih, and Y.C. Chen. *A dynamic self-adjusted buffering mechanism for peer-to-peer real-time streaming*. International Journal of Intelligent Systems and Applications, volume 3, number 3, 2011, pages 1–10.

[b] J.L. Kuo, C.H. Shih, C.Y. Ho, M.C. Wang, and Y.C. Chen. *RING: A cross-Layer P2P group conferencing mechanism over mobile ad-hoc networks*. IEICE Transactions on Communications, volume E95-B, number 9, 2012, pages 2759–2768.

[c] J.L. Kuo, C.H. Shih, C.Y. Ho, and Y.C. Chen. *A Cross-Layer Approach for Real-Time Multimedia Streaming on Wireless Peer-to-Peer Ad-Hoc Network*. Ad Hoc Networks, volume 11, issue 1, 2012, pages 339–354.

[d] C.H. Shih, J.L. Kuo, C.Y. Ho, and Y.C. Chen. *Media-Independent Handover Design for Seamless Mobility in Heterogeneous WiMAX/WiFi Networks*. IEICE Transactions on Communications, volume E95-B, number 10, 2012, pages 3168–3178.

[e] C.H. Shih, J.L. Kuo, and Y.C. Chen. *A Cross-layer Design for Wireless Ad-Hoc Peer-to-Peer Live Multimedia Streaming*. IEICE Transactions on Communications, volume E95-B, number 10, 2012, pages 3316–3319.

[f] J.L. Kuo, C.H. Shih, and Y.C. Chen. *A self-adjusted buffer control for P2P live streaming*. The International Symposium on Computer Network and Multimedia Technology, 2010.

[g] J.L. Kuo, C.H. Shih, and Y.C. Chen. *A comprehensive study of delivery strategies with chunk scheduling for mesh P2P live streaming*. Proceedings of the International Conference on Ubiquitous Information Management and Communication, 2011.

[h] C.H. Shih, J.L. Kuo, C.H. Huang, and Y.C. Chen. *A proxy-based fast handover scheme for hierarchical mobile IPv6*. Proceedings of the International Conference on Ubiquitous Information Management and Communication, 2011.

[i] J.L. Kuo, C.H. Shih, and Y.C. Chen. *A Host-based Fast Mobility Scheme (HFMS) in 802.16j Mobile RS Mode*. International Conference on Network and Computing Technology, 2013.