

CITY UNIVERSITY OF HONG KONG
香港城市大學

**Distributed Resource Management in Wireless
Ad Hoc Networks: From the Perspective of
Cross-layer Design and Interaction**

**無綫自組網絡中的分佈式資源管理：
從跨層設計和交互的角度**

Submitted to
Department of Electronic Engineering
電子工程學系
in Partial Fulfillment of the Requirements
for the Degree of Master of Philosophy
哲學碩士學位

By

Su Xueyuan
蘇學淵

July 2007
二零零七年七月

Abstract

In recent years, resource management in wireless ad hoc networks has drawn increasing attention. However, due to the interference-limited characteristic of wireless channels and the distributed operation mode of ad hoc networks, before an effective and efficient solution is obtained, several challenges need to be addressed.

Generally speaking, the motivation of resource management is to allocate fair share of bandwidth and support quality of service for different users in the network. Reliable routing protocols could increase the packet delivery ratio and reduce the number of rerouting operations, and thus provide the first step of effectively managing the scarce wireless resource. According to a specific allocation objective, the rate allocation schemes in the network layer calculate the entitled rate for each flow. Then the MAC protocols support these rate allocation schemes by coordinating the packet transmissions and guarantee the calculated rates are achieved in the MAC layer.

As we will discuss in detail later, a well designed reliable routing protocol is an important part of resource management in wireless ad hoc networks, especially when nodes are changing their positions, which essentially leads to mobile ad hoc networks (MANETs). There has been very active research on routing in MANETs and many routing metrics and protocols were proposed in the past few years, for both unicast and multicast transmissions. However, the reliability of transmission remains a problem. In the first part of this thesis, a Robust Link Availability Routing (RLAR) protocol is proposed to address the problem of reliable routing. RLAR combines pro-active routing with on-demand routing. The two modules of RLAR are responsible for pro-active estimation of link reliability and on-demand optimal routing, respectively. The first module relies on the information collected from the physical layer to estimate the normalized link availability (NLA). The second module then takes charge of route discovery and maintenance based on NLA. The optimal routing tree is first constructed and then additional protective links are also added to the tree backbone to form the ultimate mesh structure so to further enhance reliability. Through simulations, RLAR is proved to successfully reduce the number of rerouting and improve the average communication time before a rerouting procedure is triggered. The packet delivery ratio is also remarkably increased.

The study of rate allocation schemes can be divided into several steps. Firstly, a specific fairness criterion needs to be defined. Then an appropriate interference

model is built up to investigate the interference among different flows and their subflows. In the second part of this thesis, a price-based rate allocation scheme is proposed. Max-min fairness is adopted as the fairness criterion to ensure that no user is penalized excessively, and a certain minimum quality of service is guaranteed to all users. To accurately reflect the contention in wireless environment, the flow contention graph is converted from the network topology graph. Based on the flow contention graph, the clique constraint is modeled to generate the clique-based prices, which act as the congestion signal to control the end-to-end rates of multi-hop flows. Through analysis and simulation, the proposed scheme is shown to be able to effectively distribute the bandwidth, according to max-min fairness criterion, to multi-hop flows from the end-to-end perspective. The proposed scheme has many strengths, such as distributed operation, low computational complexity, applicability to both static and dynamic flows, and short convergence time. In order to further accelerate the convergence speed of the algorithm, we propose a fast converging approach which significantly reduces the convergence time. With little modification, we also extend our scheme to support weighted max-min fairness.

In the third part of this thesis, we move to the design of MAC protocol for bandwidth allocation. After the target rates are calculated in the network layer by the rate allocation schemes, a MAC protocol is needed to support these schemes by allowing each flow to obtain its entitled bandwidth through medium access control. Unfortunately, the current MAC protocols, such as IEEE 802.11 DCF and other variants, are usually based on random access or intuitional fairness, and thus cannot satisfy this requirement. Therefore, we propose a novel cross-layer MAC protocol. This protocol has a time-slotted framework. Every N time slots are defined as a transmission frame to estimate the share of the bandwidth within the corresponding maximal clique. All the nodes maintain a clique occupancy table (COT) for each maximal clique to which they are related. Based on this framework, two different MAC algorithms are implemented in this protocol. The first one, Greedy Self-Contention (GSC) algorithm, is a contention-based algorithm. With this algorithm, every node contend for the shared medium at the beginning of each free time slot. It could support arbitrary rate allocation schemes. In order to accelerate the convergence speed, the second algorithm, Cooperative Token Forwarding (CTF) algorithm, is specifically designed for the category of quasi-synchronous rate allocation schemes. It makes use of the quasi-synchronously shared flow contention information within each maximal clique to eliminate the medium access contention, and thus its convergence performance is much better.

Acknowledgements

First and foremost, I would like to express my grateful appreciation and deep thanks to Dr. Sammy Chan for his supervision and suggestions, which helped me to build up my own knowledge system in related area throughout the two years of my study period.

Acknowledgement is due to my qualifying panel members, Prof. Li Ping and Dr. Derek Pao for their helpful suggestions and evaluation of my research work. I am also grateful to Prof. Sumit Roy of the University of Washington, for numerous constructive criticisms of my work during my visiting period.

Finally, my thanks go to all who gave so generously of their time, knowledge and advice in connection with my research work.

Contents

1	Introduction	1
1.1	What is a Wireless Ad Hoc Network	1
1.2	Resource Management in Wireless Ad Hoc Networks	4
1.2.1	The Motivations	4
1.2.2	The Challenges	6
1.3	Outline of the Thesis	8
2	Reliable Routing Protocol	10
2.1	Introduction	10
2.2	Related Work	11
2.3	The Framework of RLAR Protocol	13
2.4	Module I: Link Quality Estimation	14
2.4.1	Physical Layer Information Collection and Position Vector Construction	15
2.4.2	Calculation of the Continuous Available Time T_p	17
2.4.3	Link Reliability Estimation	19
2.5	Module II: Route Discovery and Maintenance	20
2.5.1	Formation of a Robust Route	21
2.5.2	Implementation Issues	23
2.5.3	Multicast Example	25
2.6	Performance Evaluation	26
2.6.1	Simulation Setup	26
2.6.2	Simulation Results	28

2.7	Conclusion	31
3	Rate Allocation in the Network Layer	33
3.1	Introduction	33
3.1.1	Wireless Networks: Characteristics and Constraints	33
3.1.2	End-to-end Rates of Multi-hop Flows	35
3.1.3	Organization of this Chapter	36
3.2	Related Work	37
3.3	Fairness Objective and Price-based Rate Allocation	38
3.4	Implementation Issues	43
3.4.1	Information Maintenance and Exchange	43
3.4.2	Packet Header and Scheme Working Procedure	44
3.5	Performance Evaluation	46
3.5.1	Simulation with Static Flows	46
3.5.2	Simulation with Dynamic Flows	50
3.6	Fast Converging Approach	53
3.6.1	Approach Description	53
3.6.2	Performance Evaluation	55
3.7	Extension to Weighted Max-min Fairness	58
3.8	Contributions of Our Work	59
3.9	Conclusion	60
4	MAC Protocol for Bandwidth Allocation	61
4.1	Introduction	61
4.2	Related Work	62
4.3	The Framework of MAC Protocol for Bandwidth Allocation	63
4.4	Greedy Self-Contention Algorithm	65
4.4.1	Algorithm Description	65
4.4.2	Related Discussions	67
4.4.3	Convergence Time Evaluation	68

4.5	Cooperative Token Forwarding Algorithm	72
4.5.1	Algorithm Description	72
4.5.2	Related Discussions	73
4.5.3	Convergence Time Evaluation	77
4.6	Conclusion	77
5	Conclusions	79
5.1	Summary of the Thesis	79
5.2	Future Research Work	83
A	Proof of Proposition 1: Stability of the Price Feedback System	85
B	Proof of Proposition 2: Convergence of Max-min Fairness	89
C	Proof of Proposition 3: Coverage Area of Overhearing	92
	References	93

List of Figures

1.1	An example of ad hoc networks.	2
1.2	The well-known fairness problem of wireless ad hoc networks.	4
1.3	The framework of two different approaches for interaction between layers.	8
2.1	Cross-layer design framework.	14
2.2	Position vector.	16
2.3	Translation of relative movement vector.	17
2.4	Trajectory of node n relative of node m	18
2.5	Unicast network topology.	21
2.6	Route with minimal number of hops.	22
2.7	Routing result of RLAR.	23
2.8	Multicast example.	25
2.9	Packet delivery ratio.	29
2.10	Average communication time before rerouting.	30
2.11	Number of successful communications.	31
3.1	Contentions in wireless ad hoc networks.	34
3.2	Ring graph of size 5	41
3.3	Topologies for static flow simulation.	46
3.4	Simulation results of static flows.	48
3.5	Topologies for dynamic flow simulation.	51
3.6	Simulation result of dynamic flows.	52
3.7	Outline of fast converging approach.	54

3.8	Topologies for fast converging approach simulation.	56
3.9	Simulation result of fast converging approach.	57
4.1	Network example.	64
4.2	Flow chart of GSC algorithm for subflow $f_{i,j}$	66
4.3	Mismatch of clique occupancy tables	68
4.4	Example of token forwarding	75
4.5	Token deadlock	76

List of Tables

3.1	Theoretical max-min allocation (Mbps) for static flows.	49
3.2	Theoretical max-min allocations (Mbps) for dynamic flows.	52
4.1	Average convergence time of GSC algorithm (frames).	72
4.2	Average convergence time of CTF algorithm (frames).	77

Chapter 1

Introduction

1.1 What is a Wireless Ad Hoc Network

One of the most vibrant and active research topics in computer networking today is wireless ad hoc networks [1]. Under some other names such as packet radio and multi-hop networks, significant research has been ongoing in this area for over 30 years. The research on ad hoc networks can be traced back to the DoD-sponsored packet radio network (PRNET) program in 1972, which later evolved into the survivable adaptive radio networks (SURAN) program in the early 1980s. The initial goal of the PRNET program was to provide packet-switched networking to mobile battlefield elements, e.g., soldiers, tanks, aircraft, etc., in an infrastructureless, hostile environment [2]. The PRNET combined ALOHA [3] and carrier sense multiple access (CSMA) approaches for medium access. It also adopted a kind of distance-vector routing. As an evolved version of PRNET, SURAN had significant improvement of the radios, scalable algorithms and resilience to attacks. The hierarchical link-state routing protocols adopted were highly scalable.

Although originally developed for military applications, advances in portable computing devices and transmission technologies in the 20th century have led to a growing interest in deploying PRNETs in commercial applications, such as cooperative mobile data exchange, virtual classrooms and home networking, etc. The IEEE 802.11 committee adopted the term “ad hoc networks” for PRNETs and

standardized a medium access protocol that was based on collision avoidance and tolerated hidden terminals, which gradually became part of today's well-known 802.11 protocol suite.

Generally, a wireless ad hoc network is a computer network in which the communication links are wireless and thus it does not have a wired infrastructure to support communication among the hosts. Figure 1.1 shows an example of ad hoc networks. As shown, an ad hoc network is a collection of possibly mobile communication nodes. Each node in the network is a router as well as a communication end-point. All communication is over the wireless channel, hopping over several nodes. When node A intends to communicate with node B, the packets are forwarded via some intermediate nodes. Each node forwards data for other nodes, and so the choice of which nodes to forward data is made dynamically based on the network connectivity. This is quite different from the traditional wireline computer networks, where end nodes perform communication via some designated nodes known as routers, switches and hubs. Also, it is different from the wireless access networks in which centralized access points (APs) control the client nodes. When nodes in the wireless ad hoc networks tend to be moving from time to time, it is then usually called mobile ad hoc networks (MANETs).

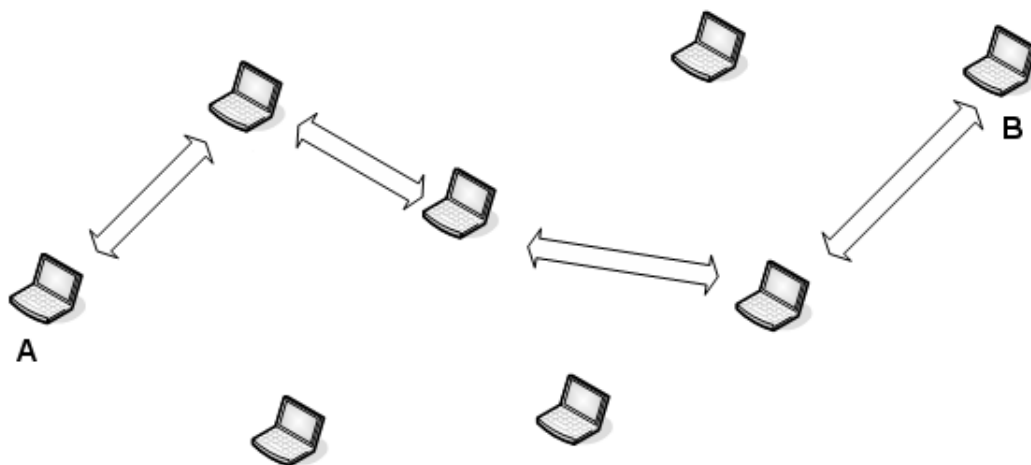


Figure 1.1: An example of ad hoc networks.

Because of the convenience of deployment and usage, ad hoc networks are widely accepted as a means to various applications. Its infrastructureless feature is naturally applicable to the environment where there are no ready central facilities or such facilities are difficult to deploy. The communication capability provided, if not optimal, is fairly sufficient for communications between equipment for emergency rescue, seafloor exploration and Ethernet-type anywhere anytime connectivity in office environment.

Wireless ad hoc networking is a multi-layer problem [1]. From the lowest physical layer to the highest application layer, special designs need to be considered to handle the unique characteristics of wireless ad hoc networks. The physical layer should adapt to rapid changes in wireless signals. The medium access control (MAC) layer needs to include collision avoidance and contention resolution through persistence and/or backoff mechanisms. MAC protocols should allow fair access to the shared medium and reliable data transmission over the single-hop wireless links. Due to its robustness and simple implementation, the widely used MAC protocol in current local area ad hoc networks is the IEEE 802.11 distributed coordination function (DCF) implemented with carrier sense multiple access with collision avoidance (CSMA/CA) mechanism and binary exponential backoff (BEB) algorithm. However, it has many limitations and could not support quality of service (QoS). IEEE 802.11e standard [4] improves the original 802.11 protocol suite by trying to differentiate service types for different traffic flows, however, it still cannot effectively manage the resource and guarantee QoS. The network layer needs to determine suitable paths and distribute the shared bandwidth for different flows. It also needs to cooperate with the MAC layer to connect the single-hop transmissions together. The transport layer must be able to deal with the relatively high packet loss and delay compared to those in the wireline networks. Distinguishing between the packet loss caused by the congestion and the transmission corruption is also needed to be taken into account. Finally, applications need to be designed to cope with the widely varying delay and packet loss, as well as frequent disconnection and

reconnection.

1.2 Resource Management in Wireless Ad Hoc Networks

1.2.1 The Motivations

A significant requirement in building wireless ad hoc networks is to provide fair access to the medium and fair share of the bandwidth, as well as supporting quality of service to meet different requirements of multiple users. However, current protocols are not able to meet this requirement. We illustrate the well-known fairness problem of wireless ad hoc networks in Figure 1.2 as an example.

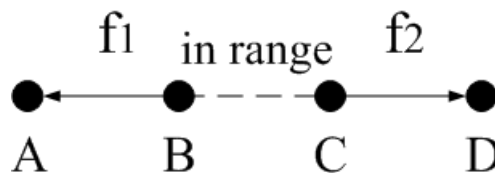


Figure 1.2: The well-known fairness problem of wireless ad hoc networks.

There are four nodes in the network constituting the chain topology. Node B and node C are in the transmission range of each other. Therefore according to the current dominating MAC protocol, IEEE 802.11 DCF, flow f_1 and f_2 , which are initiated from node B and node C respectively, will interfere with each other and contend for the bandwidth. Let us assume that node B is in luck and gets the medium first, then its backoff window will be reset to the minimal value. When node B successfully get the medium and f_1 becomes active, in contrary, the backoff window of node C will be increased repeatedly. Therefore, compared to node C, node B has greater chance to experience a short backoff time and thus could easily obtain the medium for next transmission. Such an algorithm and protocol will lead to the result that f_1 monopolize the channel and f_2 is starved with very limited throughput. This simple example illustrates the fairness problem existing in IEEE

802.11 DCF. Thus, we are motivated to design new resource management algorithms and protocols. However, due to the distributive operation mode of ad hoc networks and the random contention-based medium access control, it is often difficult to coordinate and schedule the transmission of different flows.

On the other side of the story, QoS is sometimes regarded as a joke in the development of computer networks, since after many years of research there is still no QoS even in the Internet, and most of the QoS solutions turn out to be more expensive than the resources they are trying to manage. Therefore, an effective and efficient QoS solution for computer networks, especially in wireless ad hoc networks where the resource is remarkably scarce and interference takes place everywhere in the network span, is in urgent need.

There has been very active research on resource management (also called bandwidth allocation) in wireless ad hoc networks and many schemes using various approaches were proposed in the past few years, for both single-hop and multi-hop flows. As there is no central coordinator in the whole network span, careful cooperation among all the nodes in the neighborhood is often needed for information exchange and negotiation. In addition, due to the distributive operation mode of wireless ad hoc networks, distributed algorithms and protocols are usually in favor to support the bandwidth allocation requirement. However, there is a characteristic of any distributed algorithm that they rely on dissemination of information between nodes, and thus they need a period of time to converge to the steady state. At this standpoint, the convergence time is a very important metric.

The design issues could be organized into three parts. First of all, before flows initiate their transmission, the appropriate routes should be selected. This is called the routing procedure. Due to the vulnerable wireless channel and mobility of nodes, the selected route may not be always reliable. Unreliable route will lead to frequent rerouting operations, which cause large overhead and packet loss, and thus degrade the end-to-end reliability as well as wasting precious wireless resource.

Therefore, a well-designed reliable routing protocol is the first step to efficiently manage the scarce wireless resource. Once the routes are selected and flows become active in the network, they are prone to contending with each other for the shared wireless resources. Therefore, controlling the rates of such flows and regulating their packet transmission are next important steps. In the network layer, the flow rates are calculated according to a specific allocation objective. For the research in this aspect, various algorithms are distinguished from their different approaches in formulating and solving the problem. As we will discuss later in detail, the recent trend for these solutions are often based on optimization theory and game theory. Some researchers also try to look into this problem from the economics perspective and draw a market-based solution. It is quite interesting to adapt the economics theories to solve the engineering problems in wireless networks. In the MAC layer, the design issues concern about the coordination of the medium access of single-hop flows to guarantee that the bandwidth entitlement of each end-to-end flow is indeed achieved. As IEEE 802.11 DCF cannot provide any bandwidth guarantee, a specially designed MAC protocol is needed.

So far, we have discussed the motivations for resource management in wireless ad hoc networks. However, the design of such useful and economical protocols is quite challenging, especially for the distributed algorithms and schemes. Some of the main challenges are highlighted as follows.

1.2.2 The Challenges

Interference Model and Constraint — The wireless channel is interference-limited. Because of the local broadcasting and spatial interference characteristics of wireless network, the constraint for bandwidth allocation is quite different from the counterpart in wireline networks. In order to accurately reflect the channel interferences and flow contentions, interference model and constraint need to be carefully developed.

Reliable Routing — A well-designed reliable routing protocol helps increasing the packet delivery ratio and reducing the number of rerouting operations, thus it is quite useful for efficiently managing the scarce wireless resource. The idea of cross-layer design and interaction is also adoptable in this area of research. How to define the reliability of a chosen path is the first question to be answered. After that, how to implement such design in a practical routing protocol is then put on the desk.

Rate Allocation in the Network Layer — The bandwidth entitlement for each flow is basically calculated in the network layer, in the form of rate allocation. Different rate allocation schemes differ in the approach of formulating and solving the problem, and thus may have different performances, such as application scope, computational complexity and convergence time. Designing a rate allocation scheme that could optimize the performance and draw a balance among the above factors is quite challenging.

Bandwidth Allocation in the MAC Layer — When the entitlement for bandwidth share is obtainable in the network layer, the next emerging problem is how to support these rate allocation schemes, allocate the bandwidth, and guarantee that the target rates are indeed achieved at the MAC layer. The current MAC protocols in wireless ad hoc networks, such as IEEE 802.11 DCF and other variants, are usually based on random access or intuitional fairness. Hence, these MAC protocols cannot satisfy this requirement and a specially designed MAC protocol is needed. However, how to successfully coordinate and schedule the transmissions in the MAC layer in a distributed manner desires more study.

Interaction Between the Two Layers — The interaction between the network layer and the MAC layer is necessary for bandwidth allocation protocol design. Generally speaking, two approaches are currently being explored, as shown in Figure 1.3. The first approach is to try to combine the different functions in two layers into one common algorithm. This method could centralize the problem but often have

limited application scope and high computational complexity. Another approach is to try to solve the problems in different layers individually, and information is exchanged between layers via cross-layer interactions. The two modules together constitute a complete bandwidth allocation framework. No matter what approach is adopted, how to coordinate the interaction between the two layers needs great attention and careful design.

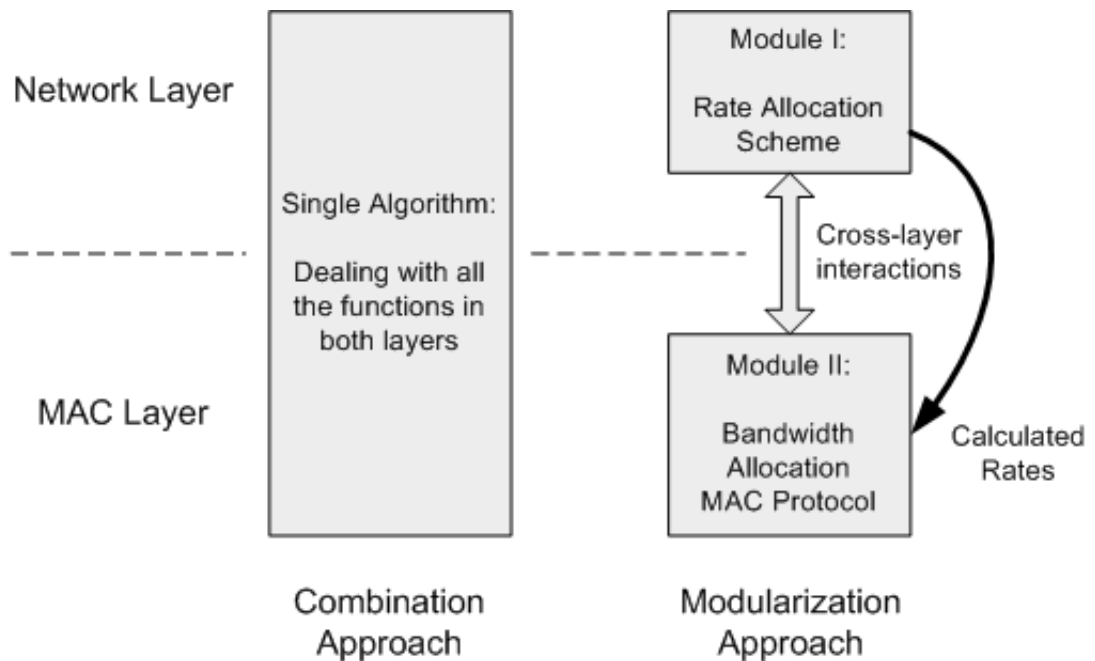


Figure 1.3: The framework of two different approaches for interaction between layers.

1.3 Outline of the Thesis

This thesis consists of five chapters. In this chapter, we have introduced the basic concept of a wireless ad hoc network. Moreover, the main motivations and several challenges for resource management in wireless ad hoc networks have also been addressed.

In Chapter 2, a cross-layer based reliable routing protocol for mobile ad hoc networks called Robust Link Availability Routing (RLAR) protocol is proposed. RLAR

consists of two modules, which estimates the link reliability using physical layer information and deals with robust optimal routing, respectively. Based on the optimal tree backbone, a mesh structure is formed for reliability enhancement. Through simulations, RLAR is proved to be able to increase the packet delivery ratio and reduce the frequency of reroutings for dynamic network topologies.

The objective of the rate allocation scheme developed in Chapter 3 is to achieve max-min fairness among different flows. To accurately reflect the clique constraint of wireless ad hoc networks, the clique-based price is generalized to act as the congestion signal, which controls the end-to-end rates of multi-hop flows. The distributed algorithms adopted are simple and direct, thus owning low computational complexity. Through analysis and simulation, the proposed scheme is shown to be able to effectively distribute the bandwidth, according to max-min fairness criterion, to multi-hop flows from the end-to-end perspective. In order to accelerate the convergence speed of the algorithm, a fast converging approach which significantly reduces the convergence time is also proposed.

In order to support the rate allocation scheme from the MAC layer, a cross-layer MAC protocol is proposed in Chapter 4 to coordinate transmissions among flows and guarantee their allocated rates. Two different algorithms are implemented in this protocol: the first one is able to support arbitrary rate allocation schemes but has a longer convergence time, while the second one is specifically designed for a category of rate allocation schemes and thus have a much better convergence performance. Both algorithms are evaluated through simulations and proved to be able to achieve the target bandwidth allocation.

Chapter 5 summarizes this thesis and suggests some future directions of cross-layer design in distributed resource management for wireless ad hoc networks.

Chapter 2

Reliable Routing Protocol

2.1 Introduction

There has been very active research on routing in mobile ad hoc networks (MANETs) and many routing metrics and protocols were proposed in the past few years, for both unicast and multicast transmissions. However, the reliability of transmission remains a problem. Due to the mobility of nodes, the topology of a MANET changes dynamically, which poses challenges for reliable transmission. Because of the resource-limited nature of MANETs, too frequent rerouting will cause large overhead and packet loss, which degrade the end-to-end reliability.

As [6] indicates, static and mobile wireless networks can present two very different sets of challenges, and solutions that work well in one setting are not guaranteed to work just as well in another. It also has been shown that many recently proposed routing metrics, such as ETX [21], can only outperform the simple hop count metric in static topologies. When nodes are mobile and network topology changes dynamically, although using hop count metric does not lead to good performance, yet its performance is better than such link-aware metrics. At this standpoint, specially designed link-aware metrics for mobile environment are in urgent need.

In this chapter, we propose a new routing metric called *normalized link availability* (NLA) which measures the reliability of a link. NLA is based on the link availability

defined in [19] that indicates the probability of a link being continuously available during a given period of time. Compared to the previously proposed reliability metrics, such as link lifetime [12][13], NLA is more accurate since besides historical information, it is also based on predictions. Then, we propose a new robust routing protocol called Robust Link Availability Routing (RLAR) protocol for both unicast and multicast. Based on a cross-layer approach, one of the modules of RLAR calculates NLA from estimated physical layer information. The other module of RLAR takes charge of the route discovery and maintenance based on this reliability metric. The objective of our work is to increase the packet delivery ratio and reduce the frequency of reroutings. The performance of RLAR protocol is evaluated with promising results.

The rest of this chapter is organized as follows. Section 2.2 reviews some other reliable multicast protocols for MANETs. Section 2.3 depicts the framework of RLAR. Then Section 2.4 and Section 2.5 describe the two modules of RLAR, respectively. Performance evaluation is conducted in Section 2.6 and Section 2.7 concludes this chapter.

2.2 Related Work

Generally speaking, there are two different approaches to improve the reliability of multicast, one focuses on the MAC level [7][8], i.e., reliable transfer of data across single-hop wireless links; the other one is targeted at the network level. More specifically, the second approach achieves reliable multicast mainly from two aspects, i.e., loss recovery [9][10] and reliable routing [11][12][13]. Loss recovery ensures reliable end-to-end delivery of unreliable multicast packets through acknowledgments and retransmission. Reliable routing aims to build multicast routes with reliable links.

[7] presents an extension to the IEEE 802.11 MAC layer protocol called 802.11MX to provide link level reliability. It adopts NAK/NCTS and dual busy tones to reduce packet collisions and deal with packet loss recovery. Similar to [7], [8] also

uses the busy tones to realize multicast reliability. It proposes RMAC to achieve full reliability which uses positive feedback ACKs. The limitation of RMAC is that it has to deal with multiple feedbacks.

MAODV is proposed in [14], which is an extension of the well-known unicast routing protocol AODV. It is a tree-based routing protocol which finds the multicast route with minimal cost, such as number of hops, based on the distance vector approach. However, it does not specifically consider multicast routing from the reliability perspective. [9] implements anonymous gossip over MAODV to improve the reliability performance. When packet loss occurs in one receiver, a request for a copy of the lost packet is sent to a randomly chosen member of this multicast group. In [10], based on the multicast tree, FAT (family ACK tree) is proposed. When a link of the multicast tree fails, the fragmented parts of the tree will be glued back to the tree using the FAT protocol. Automatic repeat request (ARQ) is used for the recovery of lost packets. [11] proposes a method called Independent-tree ad hoc Multicast Routing which, based on the concept of alternate path routing, simultaneously constructs more than one multicast trees with minimal overlap. Since the trees are independent, if one of them fails, packet may be delivered via an alternative tree without invoking rerouting. However, since the tree construction does not take into account the reliability metric, this method fails to rule out the possibility that all the trees in the set have weak links and therefore are not reliable. This leads to the situation that the alternate paths also fail when they are needed. In addition, with multiple independent trees constructed, the size of the routing table in each node grows significantly. [12] also constructs more than one multicast trees for reliable enhancement, however, different from [11], the multiple multicast trees are of high correlation, since they share some most reliable links as the backbones.

Previous schemes have proposed many routing metrics. The three metrics evaluated in [6], i.e., expected transmission count (ETX), per-hop RTT, and per-hop packet pair, could only have good performance in static networks. Based on historical node staying time, [13] proposes a metric called “link lifetime”. [17] incorporates both

signal stability and location stability to quantify the quality of the wireless link. The link quality is estimated by statistically analyzing the historical information of the link existence and no prediction is involved. Due to the dynamic property of mobile ad hoc networks, these metrics based on historical information may lead to misjudgement of link quality. [18] proposes a probabilistic link availability model to predict the accumulative availability probability during a period of time. In other words, in this estimation, a link is allowed to be broken down during one or more intervals. Therefore, it is not feasible for routing algorithms because usually after a link failure, a rerouting procedure will be performed. Based on the same assumptions as in [18], [19] tries to estimate the link availability by considering both the constant movement and possibly changing movement patterns. Although the link availability is indicated to be predicted from the continuous available time, how to generate this time value is not mentioned in [19].

2.3 The Framework of RLAR Protocol

The RLAR protocol is a hybrid protocol which combines pro-active routing with on-demand routing. Each node in the network maintains two tables. The first one is Link Quality Table (LQT) which stores the quality information of wireless links from the node to its one-hop neighbors. Due to the mobility of nodes in MANETs, the content of LQT should be refreshed from time to time. It is proposed here to use the beaconing mechanism for general update of LQT. With the use of the beacon signal, this part of RLAR is similar to the pro-active schemes. The other table is Routing Table (RT) as mentioned in other routing protocols. In RLAR, the information in RT is updated when a transmission session is initiated. In other words, the routing procedure uses the mechanism of on-demand routing. Therefore, RLAR consists of two modules: (1) link reliability estimation, and (2) route discovery and maintenance. Besides being a hybrid combination of both pro-active routing and on-demand routing, RLAR is also a cross-layer based protocol, since the link

reliability is measured directly from physical layer information and forwarded to higher layers. The framework of cross-layer design is shown in Figure 2.1. In the next two sections, the two modules are discussed in detail respectively.

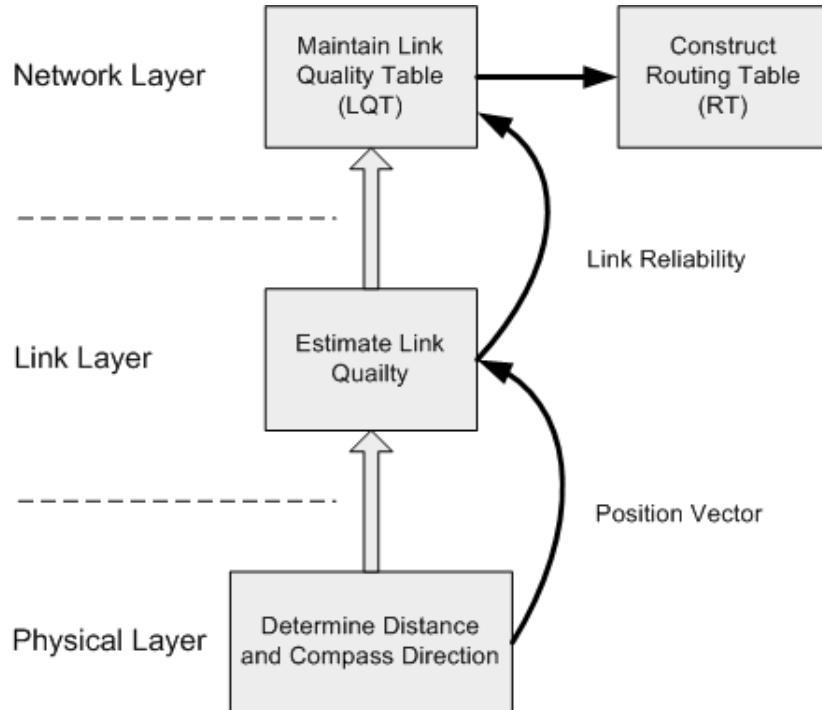


Figure 2.1: Cross-layer design framework.

2.4 Module I: Link Quality Estimation

We discuss the first module of RLAR in this section. For some existing routing algorithms, such as MAODV [14], it is assumed that the best route between two nodes is the one with minimum number of hops. This may be reasonable and convenient in wireline networks, however, in mobile ad hoc networks, due to the dynamic topology and node mobility, links are not always available. Low reliability route will lead to frequent rerouting and cause interruption of communication. This is intolerable in resource-limited wireless networks. Therefore, we give a high priority to route reliability. As it will be shown, the information from physical layer can be used to estimate the link quality between each pair of the neighboring wireless

nodes. RLAR relies on it to update LQT and determine the optimal route for robust routing.

2.4.1 Physical Layer Information Collection and Position Vector Construction

We first describe how to collect information from physical layer and construct the position vector based on such information. Two assumptions are made as follows.

(1) The shadowing model [20] is adopted as the propagation model. Transmission channel could be modeled by the equation

$$P_r = \frac{P_t \times G_t \times G_r \times \lambda_w^2}{16 \times \pi^2 \times D^2 \times L} \quad (2.1)$$

In other words, the receiving power P_r is only related to the transmission power P_t , transmission gain G_t , receiving gain G_r , wavelength λ_w , transmission distance D and system loss L . Usually, P_t , G_t , G_r , λ_w and L are fixed values. Therefore, the only factor that affects the receiving power P_r is transmission distance D . Based on this assumption, a node can determine the distance to its neighbor from the receiving power according to Equation (2.1).

(2) The antenna of one node can determine the compass direction θ from which the other node's signal is arriving. This technology is known as AOA (Angle of Arrival) and has been shown feasible in GSM network positioning. Actually, the wireless channel in wireless ad hoc network is much better than that in GSM network, since the transmission range is much smaller (usually maximal 300m) and the channel suffers less from those problems such as channel fading and multi-path effect. Therefore, in wireless ad hoc networks, it is easier to determine the compass direction from which the signal is arriving. Due to limited space here, we omit the details of AOA. Interested readers are referred to [15][16].

Based on the two assumptions made above, a node could collect information from

physical layer to determine the distance D to another node and the compass direction θ from which the other node's signal is arriving. A *position vector* \vec{L}_{mn} from the receiving node m to the transmitting node n can be derived from these two parameters, with $|\vec{L}_{mn}| = D$ and $\angle \vec{L}_{mn} = \theta$. The construction of position vector is shown in Figure 2.2.

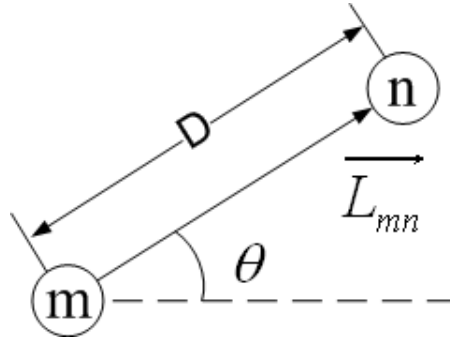


Figure 2.2: Position vector.

In order to enable each node to collect accurate position vectors from one-hop neighbors, we propose a beaconing mechanism for each node to actively broadcast the beacon signal in the neighborhood. The objective of broadcasting beacon signal is only to let one-hop neighbors know the current position of the sender. Therefore, the beacon signal could be carried by a small beacon packet which is transmitted in the same frequency band as data packets, or by a out-of-band signal such as a narrowband tone. In order to avoid contention among beacon signals from many neighbors, the inter-transmission time of two beacon signals of each node is uniformly distributed in the range of $[0.5T, 1.5T]$, where T is the average inter-transmission time of beacon signals. In addition, in order to minimize the overhead caused by beacon signals, if a node has just sent out a data or control (ACK, RTS, and CTS) packet, this packet could be used for determining position vector and the next beacon signal could be skipped. In this way, the number of beacon signal transmissions is reduced. Due to the beaconing mechanism, the link quality information stored in LQT is refreshed actively.

2.4.2 Calculation of the Continuous Available Time T_p

We now show how to calculate the continuous available time T_p via the position vectors constructed from the physical layer information. Without loss of generality, assume that at time $t = 0$, node m and node n are at positions m_0 and n_0 , respectively. Let us consider nodes m and n are both in movement. As shown in Figure 2.3, after a short period of time τ , node m moves to m' , and node n moves to n' , both with constant velocity and direction.

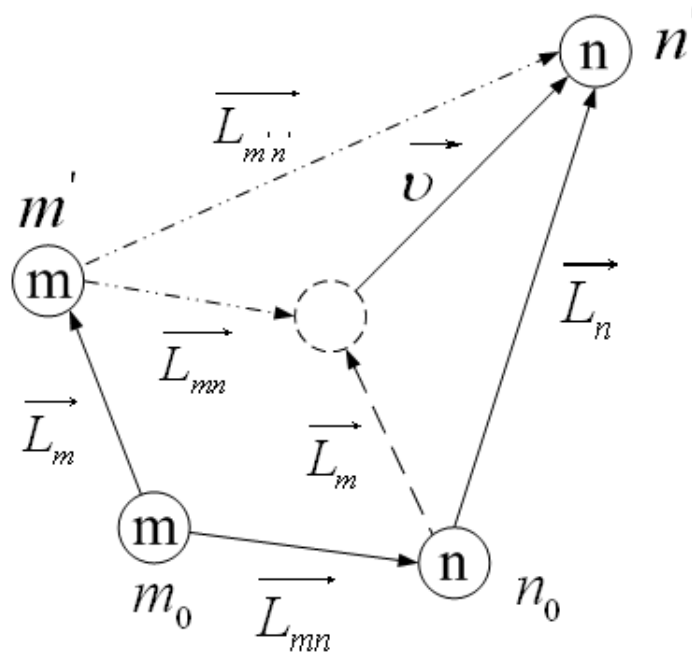


Figure 2.3: Translation of relative movement vector.

The movement vectors are \vec{L}_m and \vec{L}_n , respectively. Now we translate the two nodes' movements into one combined movement by fixing the position of one node, say, node m . From the viewpoint of node m , the relative movement of node n follows the vector \vec{v} , which we call *the relative movement vector*. It is obvious that $\vec{v} = \vec{L}_n - \vec{L}_m$. However, \vec{L}_n and \vec{L}_m are not known due to the distributed structure of MANETs. Instead, we can use another approach, i.e.,

$$\vec{v} = \overrightarrow{L_{m'n'}} - \overrightarrow{L_{mn}} \quad (2.2)$$

As discussed above, the position vectors $\overrightarrow{L_{m'n'}}$ and $\overrightarrow{L_{mn}}$ could be constructed from physical layer information at node m at $t = 0$ and $t = \tau$, respectively. By Equation (2.2), we obtain the relative movement vector from the position vectors. In this sense, the combined movement of both nodes could be interpreted as that while node m is fixed, node n is moving along the vector \vec{v} . Based on \vec{v} and the assumption that both nodes maintain the directions and velocities of their movements, the trajectory of node n seen by node m is shown in Figure 2.4.

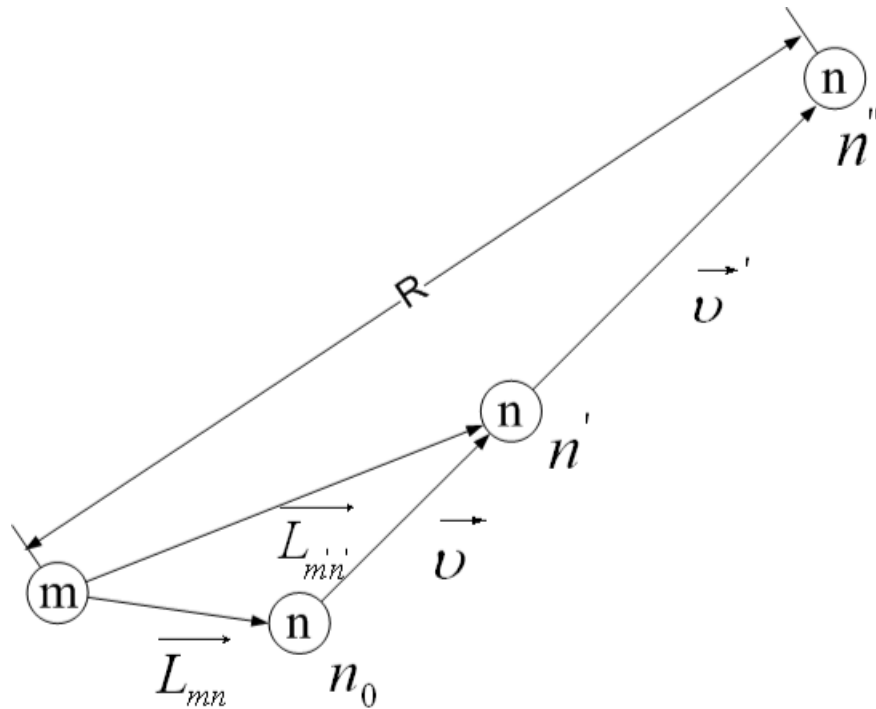


Figure 2.4: Trajectory of node n relative of node m .

Following the trajectory, node n can always communicate with node m until it reaches position n'' , which is at a distance R from m' , where R is the transmission range of each node. Let us denote the time taken by node n to move from n' to n'' as T_p , which is clearly the continuous available time of the wireless link between node m and node n from $t = \tau$. Since $\overrightarrow{L_{mn}}$, $\overrightarrow{L_{m'n'}}$, \vec{v} and τ are all available, and

that $|\vec{v}'|$ can be obtained from R and $|\overrightarrow{L_{m'n'}}|$, T_p can be calculated by

$$T_p = \frac{|\vec{v}'|}{|\vec{v}|} \times \tau \quad (2.3)$$

2.4.3 Link Reliability Estimation

Once T_p is known, we can calculate the probability that a link is available during the time period of T_p using the approach developed in [19]. Assuming a random walk-based mobility model, each node moves in a constant direction at a constant speed during a time interval called mobility epoch. The epoch length of each node is exponentially distributed with mean λ^{-1} . The direction is uniformly distributed over $[0, 2\pi)$ and the speed is also uniformly distributed in a given range. Also assuming that speed, direction and epoch length are uncorrelated, mobility of nodes are uncorrelated and links fail independently. Based on all the assumptions above, a conservative prediction of the link being available in the time period of T_p is given in [19] as

$$L(T_p) = \frac{1 - e^{-2\lambda T_p}}{2\lambda T_p} + \frac{\lambda T_p e^{-2\lambda T_p}}{2} \quad (2.4)$$

To take into account the reliability of a link when selecting routes, we need a metric to reflect this aspect and whose value should lie in the range $[0,1]$. Although $L(T_p)$ satisfies these criteria, it is not suitable for our purpose because it is associated with a time scale of T_p . That is, it provides link availability estimation for the time interval T_p . In general, each link will have a different T_p and thus it is impossible to aggregate the $L(T_p)$ of each link along a path to estimate the path availability. Instead of using $L(T_p)$, we start off with the term $T_p \times L(T_p)$, which is an estimation of average available time of a link. Assuming that estimation will be carried out regularly with period T_r , we can confine our interest of the estimation within T_r . In other words, we are more interested to know the ratio of $T_p \times L(T_p)$ to T_r . When

this ratio is smaller than one, a link with a higher value is more reliable than a link with a lower value. On the other hand, when two links have $\frac{T_p \times L(T_p)}{T_r} \geq 1$, both of them can be regarded as always available in the time period T_r and there is no need to distinguish the availability of two such links. Based on the above reasoning, we propose a new routing metric which is referred to as *normalized link availability* (NLA) and its definition is as follows:

$$NLA = \min\left[\frac{T_p \times L(T_p)}{T_r}, 1\right] \quad (2.5)$$

Once NLA is obtained, it is then forwarded to the network layer and stored in LQTs.

Note that T_r is a tunable system parameter and could be related to R/S , where S is the maximum speed at which nodes can move. As R/S is the minimum time that a node would take to move out of the transmission range of another node, we may set T_r to be a fraction of this value, say, $\frac{R/S}{10}$ or $\frac{R/S}{5}$, depending on how often we want to monitor the availability.

2.5 Module II: Route Discovery and Maintenance

Functionally, three phases are defined in the second module of RLAR: the initialization phase, the operation phase, and the recovery phase. When a unicast or multicast transmission is initiated, RLAR triggers the initialization phase to discover the optimal route based on the link quality information stored in LQTs. After the route is successfully constructed, operation phase begins and data packets are delivered to the receivers. When packet loss occurs, RLAR works in recovery phase and tries to perform local recovery first. If the situation keeps on becoming worse and more packets are lost, RLAR triggers rerouting procedure and enter the initialization phase. Next, we will discuss the route discovery in detail.

2.5.1 Formation of a Robust Route

Assuming that each wireless link fails independently, the reliability of a single-chain route P is then given by

$$R_e(P) = \prod_{l \in P} NLA_l, \quad (2.6)$$

where NLA_l is the normalized link availability of link l .

When a transmission session is initiated, the source node triggers the initialization phase to discover the optimal route. The route discovery phase works in two steps: (1) construct the optimal tree structure (for unicast, there is only one leaf in the tree) with routes of the highest reliability; (2) according to a reliability threshold, make additional protections for some vulnerable links in the tree backbone, thus forming the ultimate mesh structure. Usually this is the partially connected mesh, rather than the fully connected one. By forming the mesh structure, packets may be forwarded through shorter path. In addition, the overall reliability is improved.

In order to simplify the demonstration of RLAR, we take a simple unicast scenario as an example. The network topology is shown in Figure 2.5. The value on each link is the NLA estimated from the first module. The source node S intends to send packets to destination node D. If no route information is available, it will trigger the route discovery.

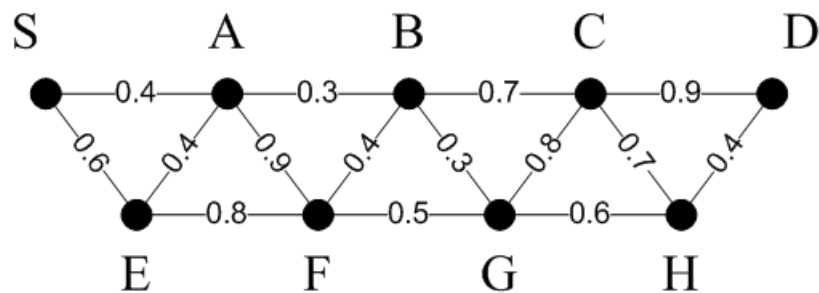


Figure 2.5: Unicast network topology.

If the number of hops is used as the cost metric, the best route is the one with

minimal number of hops, which is $S \rightarrow A \rightarrow B \rightarrow C \rightarrow D$, as shown in Figure 2.6. However, from the viewpoint of reliable transmission, this route is not the best one. According to Equation (2.6), the reliability of this route is $R_e = 0.4 \times 0.3 \times 0.7 \times 0.9 = 0.0756$.

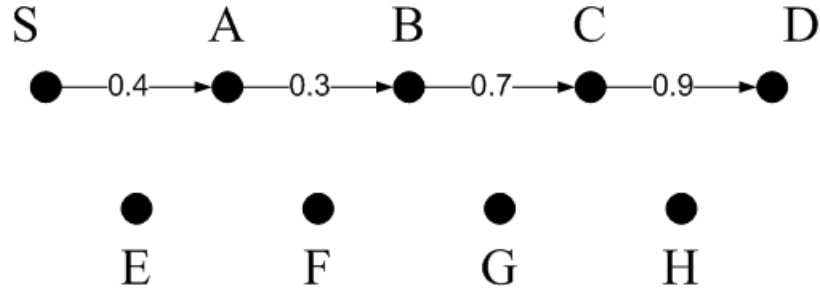
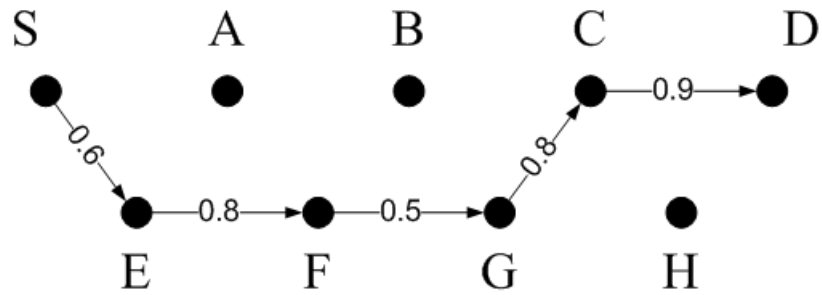


Figure 2.6: Route with minimal number of hops.

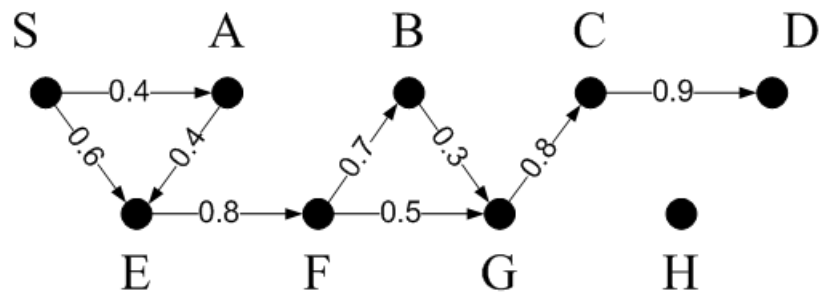
RLAR algorithm regards NLA as the primary metric, so the best route is the one with the highest reliability, but not necessary the one with minimal number of hops. For this network topology, RLAR algorithm first constructs the optimal tree structure $S \rightarrow E \rightarrow F \rightarrow G \rightarrow C \rightarrow D$, which is shown in Figure 2.7(a). The reliability of this route is $R_e = 0.6 \times 0.8 \times 0.5 \times 0.8 \times 0.9 = 0.1728$, which is much better than that of the minimal-hop route.

Then, in the second step, RLAR examines each link in the tree structure. For some links, if the NLA is smaller than a given threshold L_r , e.g. $L_r = 0.8$, protections for such links are made. Redundant paths are set up to protect these relatively weaker links. Note that there is no resource reservation in the redundant paths, only the route information is stored in RTs of corresponding nodes. When delivering a packet, if the weak links in the tree backbone fail, the packet can be delivered through the protection path. After this step, the tree structure becomes a mesh structure, which is shown in Figure 2.7(b).

As a result, the reliability of this route is $R_e = [1 - (1 - 0.6)(1 - 0.4 \times 0.4)] \times 0.8 \times [1 - (1 - 0.5)(1 - 0.7 \times 0.3)] \times 0.8 \times 0.9 = 0.2314$, which is better than the one we obtain in the first step.



(a) RLAR 1st step, tree structure.



(b) RLAR 2nd step, mesh structure.

Figure 2.7: Routing result of RLAR.

2.5.2 Implementation Issues

In the previous section, we have demonstrated the procedure of route discovery. Here we discuss some related implementation issues.

1) *How to construct the optimal tree structure?*

This can be achieved by simply adopting an flooding approach similar to that of DSR. Let us consider the network depicted in Figure 2.5. Node S broadcasts RREQ messages to A and E. Instead of the number of hops, the information included in each RREQ message is called *cumulative path reliability*, which is the product of the traversed link reliabilities. Node A then inserts (S-A 0.4) into the RREQ message and broadcast it, while Node E inserts (S-E 0.6) into the RREQ message. When the RREQ message from node A arrives at node E, E inserts (S-A-E 0.16) and compares the value with the previous one. It is obvious that $0.16 < 0.6$, so node E knows it is a worse path and discards the RREQ message from node A. All the intermediate

nodes repeat the same procedure. When all the RREQ messages arrive at node D, it compares the values in each packet and chooses the maximal one. The path included in this packet is therefore part of the optimal tree structure which has the leaf as node D. Obviously, this approach does not introduce additional overhead compared to DSR and AODV.

2) How to construct and maintain the mesh structure?

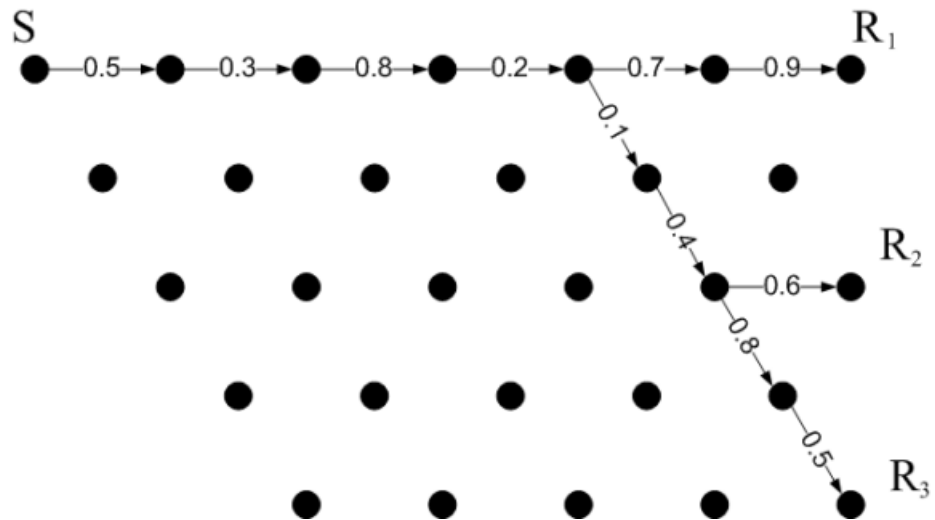
In addition to the optimal tree structure, additional protections are made for some vulnerable links, which leads to the mesh structure. First, a reliability threshold L_r is used to determine whether a link is reliable enough. If a certain link with reliability smaller than L_r , e.g., in Figure 2.5 link S-E with $NLA = 0.6 < L_r = 0.8$, the redundant path S-A-E is included to protect it. This is done by including corresponding route information in the RTs of node S and node A. Similarly, by including route information in the RTs of node F and node B, protection path F-B-G is provided for link F-G. The value of L_r determines the number of protected links and therefore affects the overall reliability as well as the requirement of memory storage for RTs. It can be tuned for different network topologies and node mobilities.

After the initialization phase, RLAR enters operation phase and data packets are delivered to the receivers. When packet loss occurs during the operation phase, RLAR will first try to perform local recovery. This is done by adjusting the protection links according to the up-to-date content of LQTs, while the backbone of the original tree structure will not be changed. When the situation becomes worse and more packets are lost, a rerouting procedure is performed and a new tree backbone will be constructed.

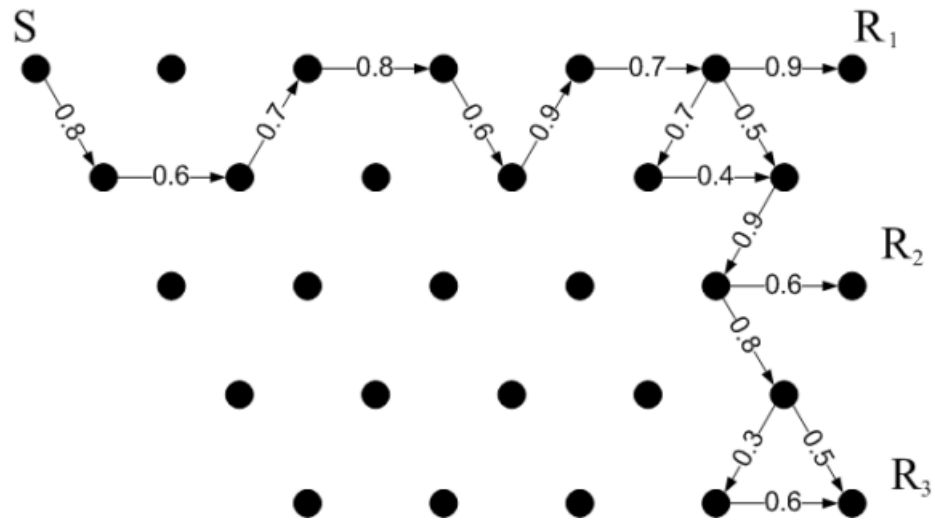
It should be noted that there are many different approaches to make the protections. In the above discussion, we adopt the most direct one - link protection. Some other approaches such as end-to-end protection and portion protection are still under study. The objective of these protection approaches is to draw a balance between the reliability enhancement and the increased requirement for RT memory storage.

2.5.3 Multicast Example

In this section, a multicast scenario is shown as an example. Based on the description in the previous section, the route obtained by MAODV is shown in Figure 2.8(a), while the one obtained by RLAR is shown in Figure 2.8(b). It is obvious that the latter one has higher reliability.



(a) Resultant route of MAODV.



(b) Resultant route of RLAR.

Figure 2.8: Multicast example.

2.6 Performance Evaluation

This section describes the simulations we have done to evaluate the performance of RLAR. We compare the performance of this protocol with MAODV and two other reliable routing protocols, i.e., ITAMAR [11] and RoMR [12]. How the accuracy of link reliability estimation will affect the performance of RLAR is also investigated.

2.6.1 Simulation Setup

First the simulation setup for our experiments is described. In order to estimate the reliability performance of the algorithms, it is assumed that no congestion and collision is involved and all the packet losses are caused by link failures due to node mobility.

Firstly, there is a requirement for our proposed protocol that the node density is high enough in the neighborhood so that redundant protective links could be constructed. Note that this is also a common requirement for ITAMAR and RoMR. One multicast session with single source and multiple destinations is simulated. We use the shadowing model [20] as the propagation model in physical layer. There are 50 uniformly placed nodes in the wireless network, with distance between two adjacent nodes about 250 m. The transmission range is 300 m. The source sends data at the rate of 100 packets/second. The random walk-based mobility model is used to determine the movement of each node. $\lambda^{-1} = 10$ and the range from which the random speed is generated is tuned for different scenarios. The reliability threshold is $L_r = 0.7$. For each scenario, at first all the nodes have a velocity of 10 m/s. In the next epoch, the velocity is uniformly generated from a specific range. The moving direction is uniformly generated from $[0, 2\pi)$. The link reliability estimation and route discovery are performed at the beginning of the simulation. MAODV uses hop-count as the routing metric, while ITAMAR builds up two independent trees for each multicast session and RoMR uses link lifetime as the routing metric. After the source node sends out 1100 packets, the simulation is finished. In order to better

estimate the reliability performance, during the simulation, no rerouting is allowed. Every scenario is simulated for 10000 times and the average result is adopted.

The performance metrics used in evaluation are:

- **Packet delivery ratio:** The number of packets successfully delivered to *all* the multicast receivers over the total number of data packets needed to be delivered. This metric directly measures how reliable a route is because no retransmission is assumed.
- **Average communication time before rerouting:** The average communication time before a rerouting procedure should be triggered, i.e., the continuous reliable communication time before route failure. Route failure followed by rerouting procedures would interrupt the data transmission and cause packet loss during the transient period. Therefore reliability and throughput would benefit from longer communication time before rerouting.
- **The number of successful communications:** The number of communications during which all the packets are successfully delivered to the destinations, i.e, no route failure occurs during the communication. This is another metric to reflect the reliability of a selected route.

The performance comparison between RLAR and other schemes are evaluated in each scenario. With any estimation method, certainly there is estimation error. In order to evaluate how the estimation error in T_p will affect the performance, we also perform some simulations of RLAR for the case with precise estimation of T_p . This is done by pre-defining the movement trajectory of each mobile node. In this way, an exact T_p is obtained directly. In the following plots, those results with exact T_p are represented by the curve labeled as "RLAR (no error)". Note that although an exact T_p is used in Equation (2.4) to estimate the link availability, the estimated availability still has error incurred by the model used in [19]. Also note that it is not possible in actual implementations to have precise T_p , however, some additional

technologies such as GPS might help minimizing the estimation errors.

2.6.2 Simulation Results

In this section, the simulation results are shown and analyzed. The performances of different protocols are drawn in the same figure for comparison. The horizontal axis of each figure is nodes' mobility speed, which represents the range from which the random speed is generated. For example, the value 5 means the range is $[0, 5]$, 10 represents $[5, 10]$, and so on.

Figure 2.9 shows the packet delivery ratio. Note that during the simulation, no rerouting is allowed, therefore it is reasonable that the packet delivery ratios of other protocols may seem to be smaller than those reported in the literature. It is obvious that MAODV has the worst performance since it does not consider the reliable delivery of packets. Also, due to the less accurate estimation of link reliability, RoMR cannot choose the optimal route and thus has lower packet delivery ratio than RLAR. When nodes are static or quasi-static, ITAMAR has higher delivery ratio because of two independent alternative paths. However, as the mobility speed increases, the performance of ITAMAR drops dramatically. This is mainly due to the fact that ITAMAR does not estimate link reliability for routing metric. When nodes are moving fast, the probability of both independent trees having weak links is high and the performance of the algorithm is degraded. Although ITAMAR can reduce this probability by building up more independent trees, the requirement for memory storage puts too much overhead to the nodes. As can be seen from the curve labeled as "RLAR (no error)", if we can eliminate the estimation error, the performance of RLAR will be improved and the performance degradation due to speed increase will also be reduced. We have also plotted the 95% confidence interval of the simulation results of RLAR and they are reasonably small.

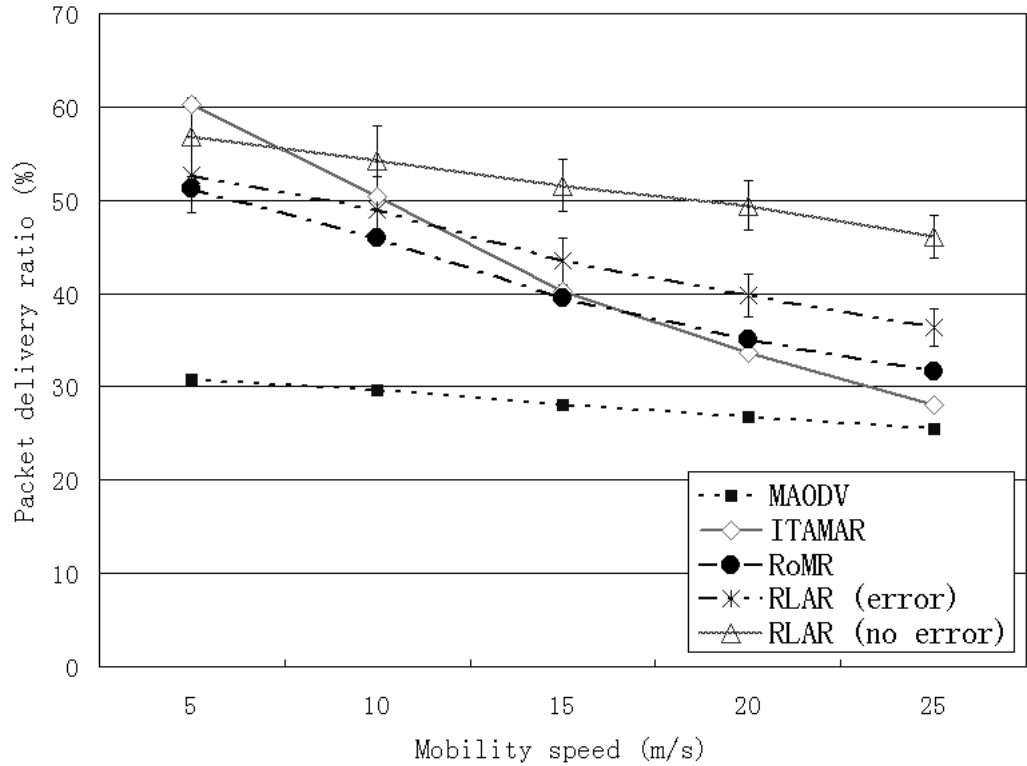


Figure 2.9: Packet delivery ratio.

Figure 2.10 shows the average communication time before rerouting. This metric is obtained by recording the simulation time before any route failure. Once a route fails, the packet cannot be delivered to the destination via the original route, thus a rerouting should be performed. Due to the resource-limited characteristic of wireless networks, frequent rerouting will cause large overhead and degrade the performance of the network. In addition, the packet in delivery during the rerouting procedure may be lost. Therefore, if the communication time before rerouting is longer, the original route is more reliable and the frequency of rerouting will be decreased. From Figure 2.10, we can see that RLAR always has longer average communication time than MAODV and RoMR. ITAMAR can achieve longer time with low mobility speed at the cost of more memory storage of RTs. When mobility speed increases, the average communication time achieved by ITAMAR has the similar decreasing trend as the packet delivery ratio.

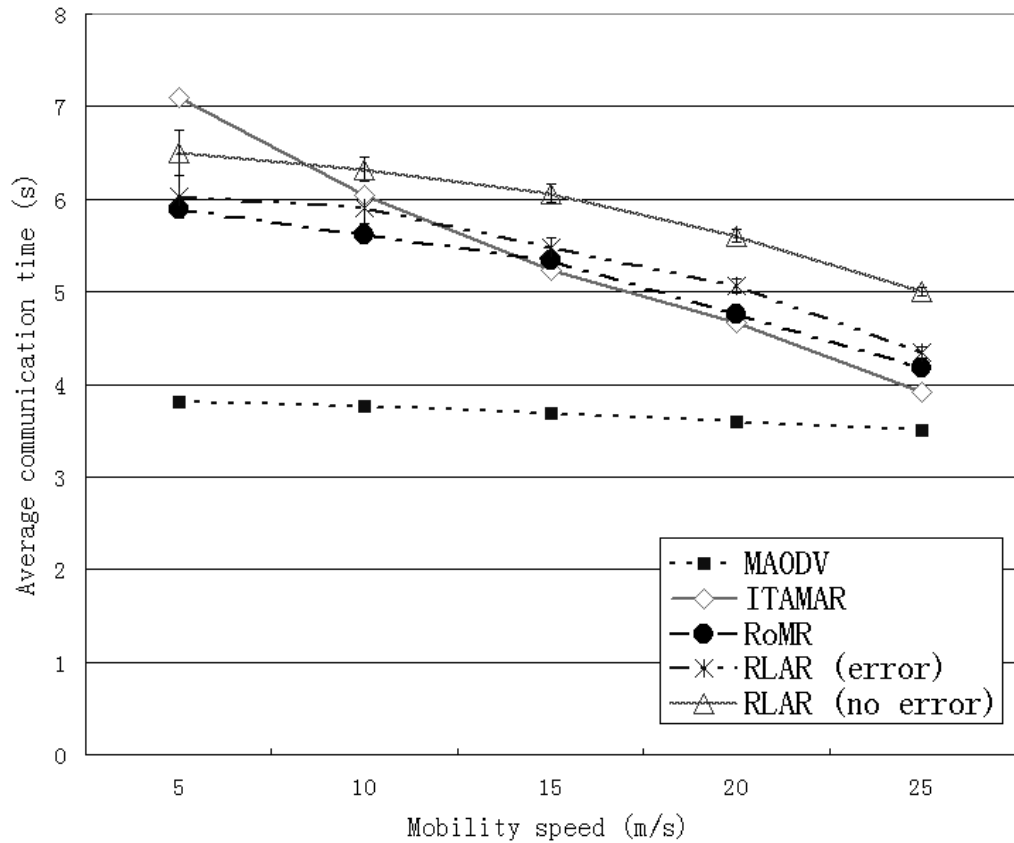


Figure 2.10: Average communication time before rerouting.

Figure 2.11 shows the number of successful communications. A communication is regarded as successful if no route failure happens during the whole simulation period, i.e, no rerouting procedure is needed in the simulation time. This metric is related to the reliability of the original route and the nodes' mobility speed. As shown in Figure 2.11, even when speed is low, MAODV has very few successful communications, while the other three reliable routing protocols have much more. It is also noted that as speed increases, the number of successful communications drops dramatically for all protocols. This implies that in highly dynamic scenarios, rerouting is necessary.

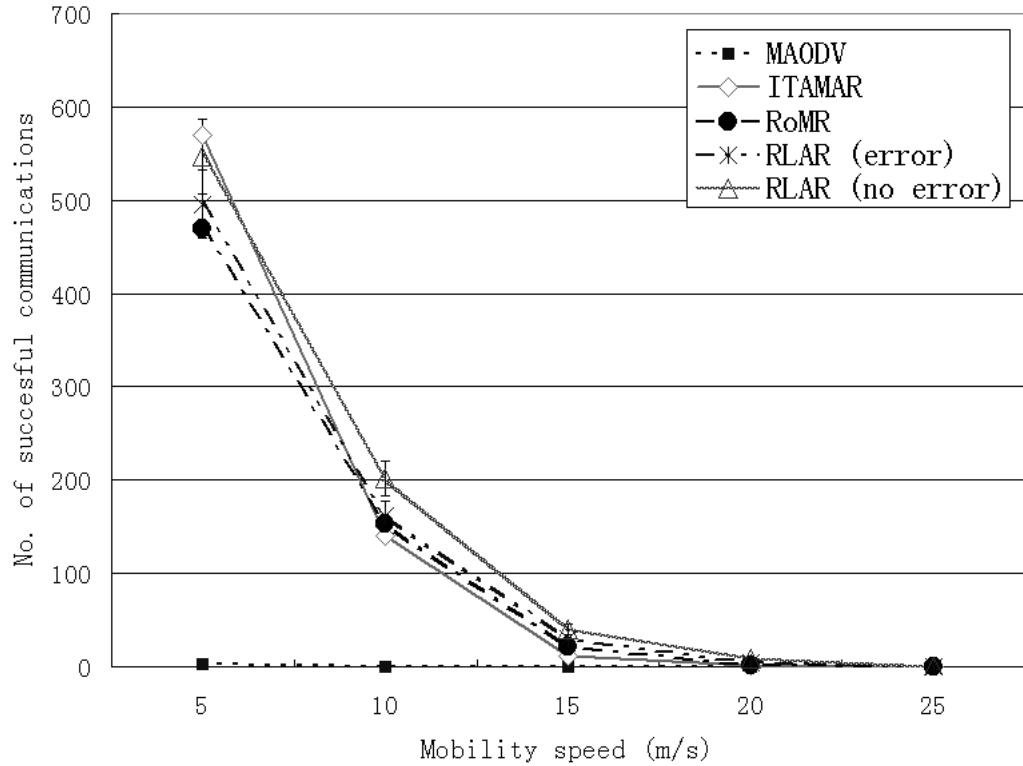


Figure 2.11: Number of successful communications.

In summary, RLAR is more efficient and effective when compared to some previous multicast routing protocols. The reliability metric estimated by RLAR from physical layer information is more accurate than the historical data used by RoMR, and it requires much less memory storage than ITAMAR with comparable performance at low mobility speed, while much better performance at high mobility speed.

2.7 Conclusion

RLAR is a new robust routing protocol for mobile ad hoc networks. It combines pro-active routing with on-demand routing. The two modules of RLAR are responsible for pro-active estimation of link reliability and on-demand optimal routing, respectively. One strength of RLAR is that it accurately estimates link reliability from physical layer information, therefore it works well with dynamic network

topologies. Based on estimated link reliability, the optimal routing tree is constructed. Additional protective links are also added to the tree backbone to form the ultimate mesh structure so as to further enhance reliability. From simulations, RLAR is proved to successfully reduce the number of reroutings and improve the average communication time before a rerouting procedure is triggered. The packet delivery ratio is also remarkably increased. In addition, it has been shown that the performance of RLAR can be further improved by more precise estimation of link reliability.

Chapter 3

Rate Allocation in the Network Layer

3.1 Introduction

3.1.1 Wireless Networks: Characteristics and Constraints

After the reliable routing protocol determines the traffic pattern in the network, the following question is then motivated: can we make intelligent rate allocation decisions that maximize the resource utility for all users? Because of the local broadcasting and spatial interference characteristics of wireless network, the constraints for the optimal solution are quite different from that in wireline networks. Let us first discuss such constraints in detail. In wireline networks, different flows compete with each other only when they share the same link. So the bandwidth of individual links is the resource to be allocated. However, since the wireless channel is a shared medium and interference-limited, flows within the interference ranges of each other compete for the medium, the shared resource in wireless networks is thus no longer the link-based bandwidth. Let's consider the network topology in Figure 3.1(a).

First we assume the transmission range equals the interference range. This is a reasonable assumption adopted in many previous works, such as [22][23][24][25]. Actually, the interference range could be adjusted by tuning the carrier sensing

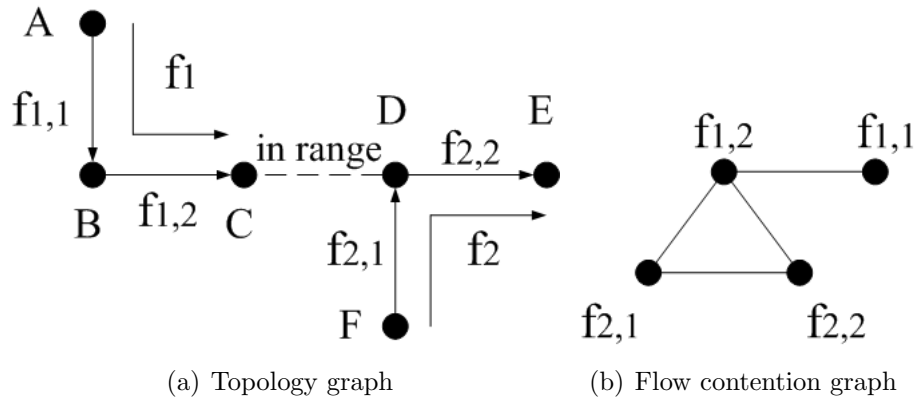


Figure 3.1: Contentions in wireless ad hoc networks.

threshold and is not necessarily equal to the transmission range. The choice of interference range is a trade-off between interference avoidance and spatial reuse. The larger the interference range is, the larger the interfering neighborhood is and thus, nodes are more sensitive to other active flows and spatial reuse is limited. However, such trade-off will not affect the discussion and analysis of our problem. We will keep the above assumption unless specially indicated. In Figure 3.1, $f_{1,1}$ and $f_{1,2}$ are two subflows of the multi-hop flow f_1 . Similarly, $f_{2,1}$ and $f_{2,2}$ are two subflows of the multi-hop flow f_2 . In wireless networks, nodes are usually regarded as half-duplex. It means that a node cannot transmit and receive data simultaneously. This implies that $f_{1,1}$ and $f_{1,2}$ cannot be active at the same time, due to the constraint of node B. This constraint is called *MAC constraint*. Also, $f_{1,2}$ and $f_{2,2}$ cannot be active simultaneously. This is because that node C and node D are within the interference ranges of each other. In wireless networks, nodes usually use omnidirectional antennas and the transmission is performed as a local broadcast. When node D is transmitting to E, the signal can also reach node C. This signal interferes with that from node B, which prevents node C from correctly receiving the data. In addition, assuming that the basic two-way handshake mechanism of IEEE 802.11 MAC protocol is used, an ACK packet will be sent back to the source node from the destination node after the data packet is correctly received. Therefore, when $f_{2,1}$ is active, the subsequent ACKs transmitted in the reverse direction could

possibly collide with the data packets of $f_{1,2}$ at node C, thus they cannot be active simultaneously either. In these cases, flows that traverse different paths interfere with each other. This is the interference-limited characteristic of wireless media.

Based on the topology graph, we define the *flow contention graph* [22], $G = (V, E)$ whose vertices, V , are the one-hop flows in the network. If two vertices have a common edge $e \in E$, it means that the two flows cannot transmit simultaneously. Figure 3.1(b) is the corresponding flow contention graph of the topology graph in Figure 3.1(a). Note that a flow contention graph is different from a wireless link contention graph mentioned in [23], in which each vertex represents a wireless link. There might be more than one flow in each wireless link. Under this circumstance, these flows traverse the same wireless link will contend for the shared capacity. In order to show the contention relations between flows clearly, different flows in the same wireless link are represented by different vertices in the flow contention graph, rather than a single vertex as in the wireless link contention graph.

As we can see, there are some fully meshed subgraphs in the flow contention graph. We call these subgraphs as *cliques*. A maximal clique is a clique which is not a subgraph of any other clique. We use Ω to denote a maximal clique. For example, there are two maximal cliques in Figure 3.1(b), i.e., $\Omega(f_{1,1}, f_{1,2})$ and $\Omega(f_{1,2}, f_{2,1}, f_{2,2})$. Only one flow can be active in a maximal clique at any time. This constraint is called *clique constraint*. It is obvious that clique constraint is more restrictive than MAC constraint. With clique constraint, flows in a maximal clique compete for the shared medium. So in wireless ad hoc networks, the shared resource is the bandwidth of each maximal clique rather than individual wireless links.

3.1.2 End-to-end Rates of Multi-hop Flows

There are many previously proposed algorithms which have been shown to perform well in providing fair rate allocation among single-hop wireless flows. Regarding the multi-hop flows, these algorithms attempt to break each of them into multiple

single-hop flows, called subflows. Unfortunately, this approach might cause some serious problems. This is illustrated by an example using the topology in Figure 3.1(a) again. We assume that each wireless link has an effective bandwidth of 1 Mbps. Let f_1 and f_2 be split into subflows $f_{1,1}$, $f_{1,2}$ and $f_{2,1}$, $f_{2,2}$ respectively. When we consider the rates of subflows, according to the max-min fairness criterion, $f_{1,2}$, $f_{2,1}$ and $f_{2,2}$ share the same clique and each has an equal rate of $1/3$. Thus $f_{1,1}$, which shares the other clique with $f_{1,2}$, will be allocated with the remaining bandwidth and have a rate of $2/3$. However, since subflows $f_{1,1}$ and $f_{1,2}$ are both part of flow f_1 , if the rate of $f_{1,1}$ is larger than $f_{1,2}$, packets from $f_{1,1}$ will accumulate and cause congestion at node B. With the use of an end-to-end transport protocol such as TCP, the rates of $f_{1,1}$ and $f_{1,2}$ will stabilize to the same value, i.e., the end-to-end rate of the multi-hop flow f_1 . Thus, there is no point to allocate a higher rate to $f_{1,1}$. In addition, a flow generally spans multiple hops across the network and it is the end-to-end throughput which determines the quality of service perceived by users. At this standpoint, it is more appropriate to consider the rate allocation from the end-to-end perspective, rather than splitting each multi-hop flow into multiple subflows and allocating the rates of subflows separately. Therefore, in the following discussions, the end-to-end rates of multi-hop flows is considered.

3.1.3 Organization of this Chapter

The rest of this chapter is organized as follows. Section 3.2 presents some related work. Section 3.3 describes the fairness objective and the price-based rate allocation scheme. According to the clique constraint of wireless networks, the link-based price is generalized into clique-based price in this section. Based on the clique-based price, a new fair rate allocation algorithm is proposed to achieve max-min fairness among different multi-hop flows. In Section 3.4, we discuss some implementation issues. From the approach of implementing the proposed rate allocation algorithm, it could be noted that this algorithm works in a completely distributed manner. The performance of the proposed scheme is evaluated in Section 3.5, with both static

and dynamic flows. In Section 3.6, we further improve the convergence time of our scheme by skipping some unnecessary gradual adjustments. The performance of this fast converging approach is evaluated through simulation and the improvement is significant. The proposed scheme is extended to weighted max-min fairness in Section 3.7. Section 3.8 highlights the contributions of our work in this chapter and Section 3.9 concludes this chapter.

3.2 Related Work

In wireline networks, the price-based resource allocation approach has been well studied [26]-[28]. [26] has proved that max-min fairness can be achieved for a specific utility function. [27] shows that max-min fairness can also be achieved through the appropriate choice of utility functions which, however, needs global information of the network. In addition, as shown in [28], defining a sequence of utility functions such that the limit of the maximum utility rate allocation is max-min fair will approach max-min fair share of the network resource. [29][30] propose a simple and direct solution - MaxNet, to this problem, i.e., to charge users the maximum price of any of the links they use. If all users have the same utility functions, it achieves max-min fairness.

In recent years, the rate allocation for wireless networks receive more and more attention. Due to the unique characteristics of wireless networks, different schemes try to look into this problem from various approaches. [22] focuses on providing max-min fair allocation and attempts to compute the appropriate bandwidth share for each flow based on its surroundings. However, it only considers single-hop independent flows and splits the multi-hop flows into multiple subflows. [31] designs a general analytical framework and mechanism for arbitrarily specified fairness model via a utility function, and concentrates on achieving such a given fairness model by an appropriate MAC layer design. According to the utility function, it adjusts the persistence probability of each flow to contend for the shared medium. Again, it

only considers one-hop flows rather than the end-to-end rates of multi-hop flows. A scheduling algorithm is proposed in [32]-[34] to provide max-min fair end-to-end rate allocations in wireless ad hoc networks with MAC constraint, which assumes that nodes in a neighborhood have different frequencies and only requires that no node simultaneously transmit and receive packets. [35] puts forward the problem of clique constraint, however, when the problem is solved, the constraint is reduced to the MAC constraint. The rate control mechanism studied in [23][38][39] can be categorized as the dual congestion controller since it can be interpreted as a gradient algorithm for the dual of an optimization problem. The limitation is that the iteration of the solution to the optimization problem has highly computational complexity. In addition, when applied to the dynamic flows, the whole iteration process might have to restart from the very beginning. [40] proposes and studies the primal-dual congestion controller, which updates the data rates to mimic the gradual response to congestion feedback of TCP. In this work, even though the congestion control is distributed, the scheduling is still assumed to be centralized. [25] also studies rate allocation from an end-to-end perspective, but its objective is to maximize the spatial reuse of spectrum, while guaranteeing a basic fair share to each flow. Also, its distributed algorithm could only lead to approximate solutions. Although [41] realizes many limitations of the previous papers, when trying to eliminate such limitations, the study is based on the node-exclusive model, which essentially only reflects the MAC constraint.

3.3 Fairness Objective and Price-based Rate Allocation

Our scheme employs max-min fairness as the criterion. The basic idea behind max-min fairness is to first allocate equal bandwidth to all contending users. If a user cannot utilize its bandwidth, because of constraint elsewhere, then the residual bandwidth is distributed among others. Thus, no user is penalized excessively, and a

certain minimum quality of service is guaranteed to all users [32]. Max-min fairness guarantees equal bandwidth to flows that traverse paths of similar congestion level and transmit packets at equal rates. The definition of max-min fairness is as follows.

Definition 1. *Assume there are n users in the network. If for a rate allocation, the aggregate of allocated rates does not exceed the bandwidth bound, we say that this is a feasible rate allocation. For a feasible rate allocation X (consisting of n rates, i.e., x_1, x_2, \dots, x_n for n users), comparing to any other feasible rate allocation S , if there is a i such that $x_i < s_i$, and there is a j such that $x_j \leq x_i$ and $x_j > s_j$, then X is said to be max-min fair allocation.*

More generally, a feasible rate allocation is said to be max-min fair if no flow rate can be increased while maintaining feasibility without decreasing that of any other flow having equal or less rate. Note that the max-min fair rate allocation is unique since the feasible set is compact and convex [33].

Consider a network with a set of users S and a set of links L . In the price-based rate allocation framework, in general, user and link processes are described by:

$$x_s(t+1) = D_s(q_s(t)) \quad (3.1)$$

$$p_l(t+1) = G(y_l(t), p_l(t)), \quad (3.2)$$

where $x_s(t)$ is the transmission rate of user s at time t , $q_s(t)$ is the network's congestion signal communicated to user s , $p_l(t)$ is the congestion signal link l generates (also called the price of link l), and $y_l(t)$ is the aggregate traffic arriving at l . Function $D(\cdot)$ is called demand function, which is often assumed to be a continuous, positive and decreasing one. The inverse of the demand function is called utility function, which indicates user's goal-attainment or want-satisfaction. Function $G(\cdot)$ is called price update rule. In wireline networks, different flows compete with each other only when they share the same link, so the price is generated from each link

to reflect the congestion within it. The basic form of $G(\cdot)$ is described as:

$$G(\cdot) : p_l(t+1) = [p_l(t) + \gamma(y_l(t) - b_l)]^+, \quad (3.3)$$

where b_l is the bound of link l , i.e., the link capacity, γ is a step size parameter and $[z]^+ = \max(z, 0)$.

In general Internet-like networks (called SumNet in [29]), a user is charged the sum of the prices of all the links that it uses, i.e.,

$$SumNet : q_s(t) = \sum_{l \in L_s} p_l(t), \quad (3.4)$$

where L_s denotes all the links that user s uses. Some previous works ([26][27][28]) have proved that by charging the sum of the prices at all the links that a flow traverses, max-min fairness can be achieved under certain conditions, for instance, the appropriate choice of utility functions.

As shown in [29], MaxNet is a simple and direct approach to achieve max-min fairness in wireline networks, in which users are charged by the maximum price of any link they use, rather than the sum of the prices. That is,

$$MaxNet : q_s(t) = \max_{l \in L_s} p_l(t) \quad (3.5)$$

If all users have the same utility functions, MaxNet achieves max-min fairness. The Internet naturally supports this assumption of the same source behavior since most of the flow-controlled traffics are generated by the dominant transport protocol TCP. If all traffic sources have the same utility functions, simply by charging each user the maximum price of any link that its flow traverse, the max-min fairness of bandwidth allocation is achieved.

Since wireless transmissions are locally broadcasting in the shared media, wireless links are always interference-limited and the flows within the interference ranges of

each other compete for the medium even though they do not share the same wireless link. Due to the clique constraint, the shared resource is no longer the bandwidth of individual wireless links, but should be based on the maximal cliques. At this standpoint, we consider to regard each maximal clique as an entity and analyze the contention within it. In our scheme, the clique-based price is generated as:

$$p_{\Omega}(t+1) = [p_{\Omega}(t) + \gamma(\sum_{l \in C_{\Omega}} \sum_{s \in S(l)} \frac{x_s(t)}{c_l} - b_{\Omega})]^+, \quad (3.6)$$

where C_{Ω} denotes the wireless links traversed by all subflows in maximal clique Ω and $S(l)$ denotes the set of flows using wireless link l with capacity c_l . Since for a feasible rate allocation, the aggregate of all the time fractions x_s/c_l never exceeds 1, the normalized bound of the resource b_{Ω} can be set to 1.

It should be noted that, the condition that the bound $b_{\Omega} = 1$ is only applicable to perfect graph¹. If a contention graph is not a perfect graph, the bound should be reduced accordingly. For example, let us consider the flow contention graph in Figure 3.2.

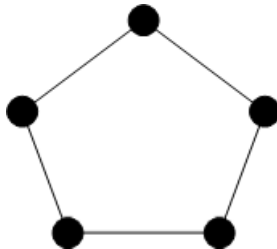


Figure 3.2: Ring graph of size 5

If we apply $b_r = 1$ to this flow contention graph, according to max-min fairness, each flow would get a rate of $1/2$. However, it is impossible to get such rates since at any time at most two flows can transmit simultaneously. In fact, this is not a perfect graph and thus the bound should be reduced accordingly. Generally speaking, according to the result on edge colorability in graphs [43], if we reduce the

¹A perfect graph [42] is a graph in which the chromatic number of every induced subgraph equals the clique number of that subgraph. A perfect graph has no induced subgraph that is isomorphic to an odd hole, which contains an odd number of vertices, or its complement.

bound b_Ω to $2/3$, this condition will be sufficient for all the graphs, either perfect ones or non-perfect ones. In this chapter, we regard all the graphs as perfect ones and keep $b_\Omega = 1$. Take the maximal clique $\Omega(f_{1,2}, f_{2,1}, f_{2,2})$ in Figure 3.1 for example. The price is updated through the following calculation: $p_\Omega(t+1) = [p_\Omega(t) + \gamma(\frac{x_1}{c_{BC}} + \frac{x_2}{c_{FD}} + \frac{x_2}{c_{DE}} - 1)]^+$.

Based on the clique-based price p_Ω in Equation (3.6), the algorithm adopted in our scheme is described as:

$$q_s(t) = \max_{\Omega \in R_s} p_\Omega(t), \quad (3.7)$$

where R_s denotes all the maximal cliques that multi-hop flow s traverses. This equation indicates that each multi-hop flow is charged with the maximum price of all maximal cliques that it traverses; in other words, this flow is controlled by the maximum price. The maximal clique with the maximum price is called the flow's *controlling clique*. According to Equation (3.6) and Equation (3.7), we put forward Proposition 1 and Proposition 2 as follows. The detailed theoretical proofs of Proposition 1 and 2 are given in Appendix A and B respectively.

Proposition 1. *In wireless ad hoc networks, congestion prices are generated at each maximal clique. By charging each multi-hop flow with the maximum price of all maximal cliques that it traverses, the rate allocation will converge to a stable equilibrium.*

Proposition 2. *When the rates of all the multi-hop flows are determined by the same demand functions, by charging each multi-hop flow with the maximum price of all maximal cliques that it traverses, the steady state of rate allocation is max-min fair.*

3.4 Implementation Issues

In this section, we discuss some implementation issues on how to apply the proposed rate allocation scheme to current wireless ad hoc networks.

3.4.1 Information Maintenance and Exchange

In the above discussion, a maximal clique is regarded as a network entity that can carry out the price update function. However, a maximal clique is only a logical entity based on flow contention graph. To deploy the scheme in actual ad hoc networks, the information maintenance and price update need to be carried out by the nodes traversed by all subflows in the maximal clique.

To implement our scheme, each node in the network needs to know neither the entire network topology, nor the complete flow contention graph. Instead, each node maintains the contention information of local portion of the network, which includes the local flow contention graph, rate of each local subflow, and price of each local maximal clique. Take Figure 3.1(a) for example. Node A is only related to subflow $f_{1,1}$, which is contending with $f_{1,2}$. Therefore the information maintained by node A includes the flow contention graph of maximal clique $\Omega(f_{1,1}, f_{1,2})$, the rates of $f_{1,1}$ and $f_{1,2}$, and the price of maximal clique $\Omega(f_{1,1}, f_{1,2})$. For node B, it is related to subflow $f_{1,2}$, which is contending with subflows $f_{1,1}$, $f_{2,1}$ and $f_{2,2}$. Hence, the information maintained by node B includes the flow contention graph of maximal cliques $\Omega(f_{1,1}, f_{1,2})$ and $\Omega(f_{1,2}, f_{2,1}, f_{2,2})$, the rates of $f_{1,1}$, $f_{1,2}$, $f_{2,1}$ and $f_{2,2}$, and the prices of maximal cliques $\Omega(f_{1,1}, f_{1,2})$ and $\Omega(f_{1,2}, f_{2,1}, f_{2,2})$.

Here rises another problem, i.e., how to obtain the information mentioned above. For node B, it is unaware of subflows $f_{2,1}$ and $f_{2,2}$ itself, since both of them are out of its transmission range. Therefore, in order to obtain the necessary local information, node B needs to exchange information with the neighboring nodes. In this example, node C could provide node B with necessary information about

subflows $f_{2,1}$ and $f_{2,2}$.

In order to generalize the situation, Proposition 3 is proposed as follows.

Proposition 3. *If we call the two communicating nodes of each subflow as its transmission pair, for each subflow in a maximal clique, at least one end of its transmission pair can overhear the ongoing transmission (either data packets or ACKs) of other subflows in the maximal clique.*

Proposition 3 is proved in Appendix C. Based on Proposition 3, the two ends of each transmission pair together could cover all the necessary information in local cliques. By overhearing the subflows (either data packets or ACKs) within its transmission range and exchanging information with immediate neighbors, each node could construct the local flow contention graph.

It should be noted that some nodes, such as node A and node B, are involved in multiple maximal cliques. In order to perform price update in each clique, they should be able to decompose their local flow contention graph into separate maximal cliques. The feasibility of decomposition has been proved in [22] based on the definition of *contention tree* and we omit the relevant discussion here.

3.4.2 Packet Header and Scheme Working Procedure

In order to apply our scheme, two fields are included in the packet header. One is the *price field* which contains p_Ω and the other one is the *rate field* which contains x_s . The two additional header fields should be applied to both data packets and control packets, such as ACK, RTS and CTS.

The value in the rate field provides information about the calculated rate of the ongoing subflow. Each node in the network regularly listens to the media to get the rate information and exchanges it with other ends of its transmission pairs.

At the same time, each node uses Equation (3.6) to update the local contention information it maintains independently. Therefore, there is no need to exchange price information.

According to the feedback of the congestion price p_Ω and demand function, the source node of each multi-hop flow calculates the transmission rate x_s and fills this value into the rate field of packet header. The price field is also initiated as zero here. Then this packet is sent out to the first relaying node. When a new packet which needs to be relayed arrives, the intermediate node compares the price stored locally with that in the price field of the packet header. If the first one is greater, this node updates the price field. At the same time, it reads the rate field and schedules the transmission accordingly. In this way, when the packet arrives at the destination node, the value of price field should be the maximum one. Then this maximum price will be sent back to the source node in acknowledgement packet.

It is noted that in our proposed implementation, price updates in each maximal clique are based on the calculated flow rates stored in the rate field, rather than the actual transmission rates. Since the calculated rates are always changing faster than the actual rates toward the equilibrium, this scheme has a shorter convergence time than those previous price-based schemes which update price based on actual rates. After equilibrium is reached, the rate allocation converges to feasible max-min fairness, and thus calculated rates equal actual rates.

From the discussion above, it can be seen that this fair allocation scheme works in a completely distributed manner. Each node maintains local contention information and performs the price update and packet transmission independently. The only information exchanges occur within each transmission pair.

3.5 Performance Evaluation

Two sets of simulations are conducted, one is with static flows and the other one is with dynamic flows. Static flow refers to a flow that is active for such a long time period that it can be regarded as if it will never leave the network after it joined in. In contrast, dynamic flows join and leave the network dynamically and often change the flow contention graphs accordingly. The effective bandwidth of each wireless link is 1 Mbps and the wireless channel is error-free. All the sources have the same demand function $x_s(t+1) = x_{max}e^{-k \cdot q_s(t)}$ and $\gamma = 0.1$.

3.5.1 Simulation with Static Flows

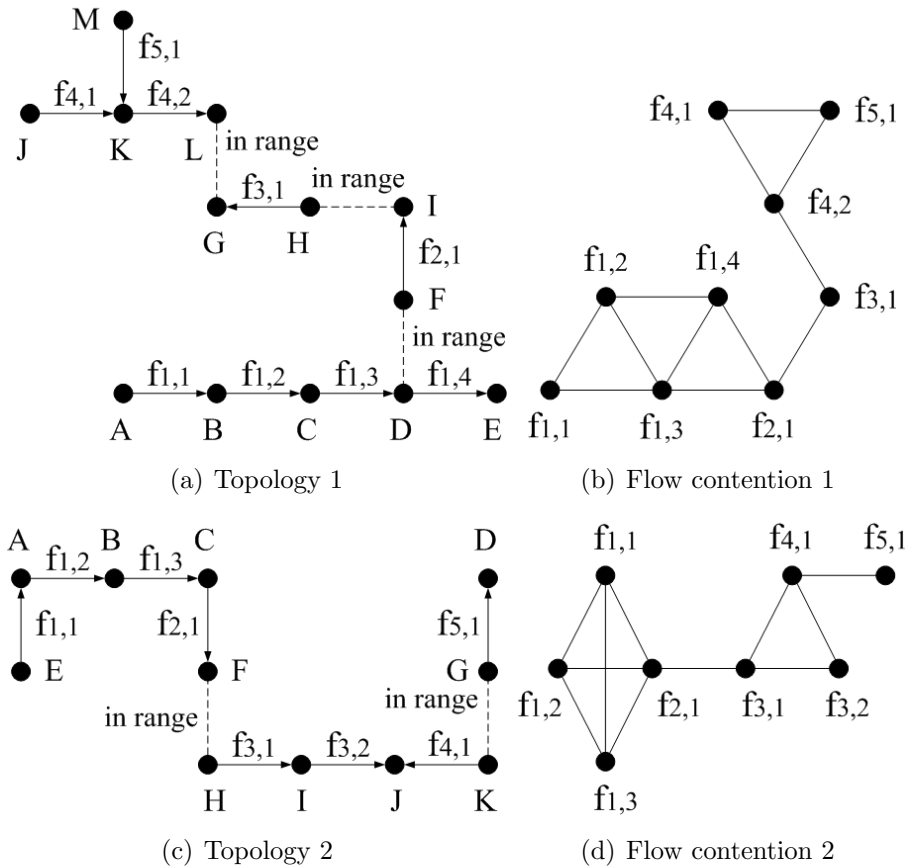
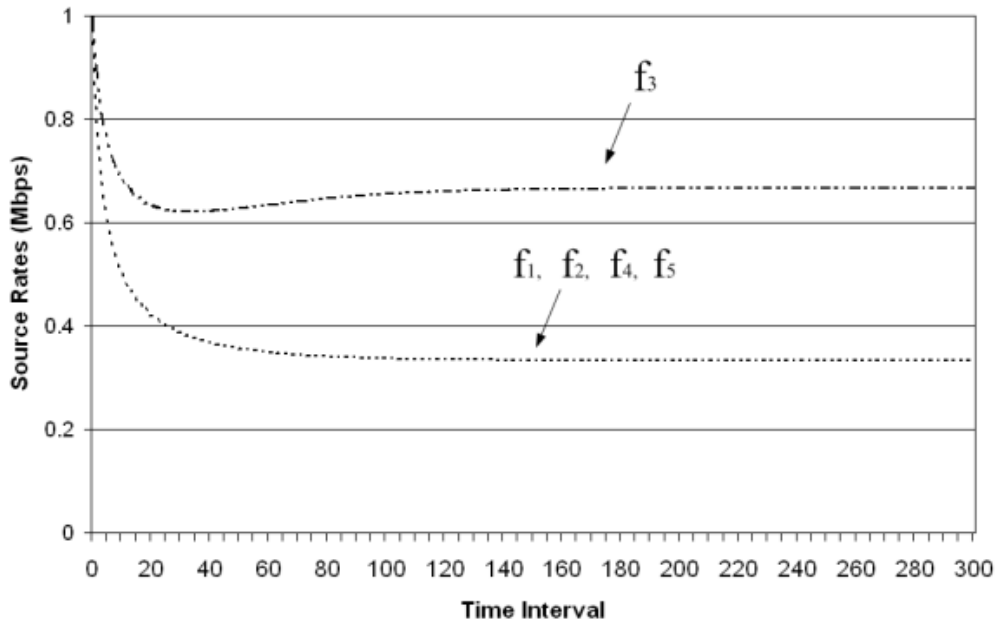


Figure 3.3: Topologies for static flow simulation.

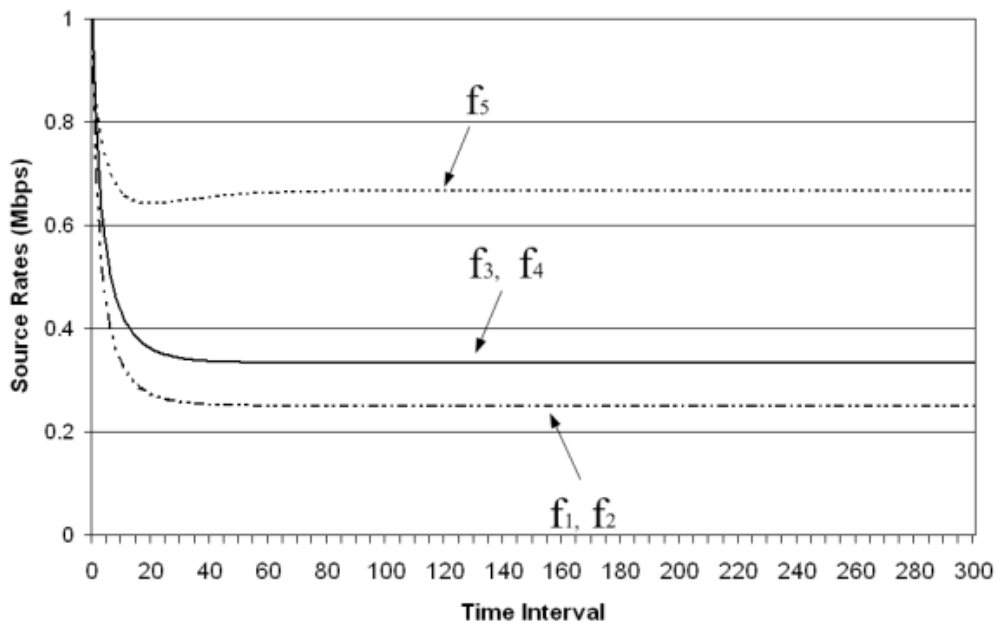
The proposed fair rate allocation scheme is simulated in two topologies, shown in

Figure 3.3 along with their corresponding flow contention graphs. Note that the simulation for the first topology is also conducted in [25]. We make the comparison in order to show that different objectives lead to different results. Our objective is different from that in [25], since it focuses on the overall spatial reuse with basic fairness while we focus on the strict max-min fairness.

The calculated source rates of the flows are shown in Figure 3.4. Initially, all the maximal cliques have price $p_\Omega = 0$, so all the sources have $q_s = 0$ and get the calculated price as x_{max} . Then the prices are updated at each maximal clique, and the maximum one is regarded as the feedback network congestion signal communicated to each source. Every source adjusts its end-to-end flow rate according to the congestion signal. Note that in a maximal clique, the aggregate of all the time fractions x_s/c_l should not exceed 1. At this standpoint, before steady state is reached, those rates shown in Figure 3.4 are not the feasible rates of the multi-hop flows, but are the calculated rates obtained from the prices and demand function.



(a) Simulation result 1



(b) Simulation result 2

Figure 3.4: Simulation results of static flows.

In steady state, the calculated rates satisfy the clique constraint as well as max-min fairness, so in this period, the actual rates are equal to the calculated rates, which are already feasible. In Figure 3.4(a), after about 120 time intervals (round-trip-time), flow rates converge to the max-min allocation as listed in Table 3.1. In Figure 3.4(b), it takes about 60 time intervals to get into the steady state and achieve max-min fairness. Note that the convergence speed is comparable to and even better than other price-based schemes in ad hoc networks, for instance [23], whose system converges to an equilibrium rate allocation and an equilibrium price vector within an average of 800 iterations.

We can see that there are only two flow rate curves in the first topology. This is because that there are only two kinds of maximal cliques in the flow contention graph, i.e., with two vertices and three vertices. All the sources are either controlled by the first kind or the second kind, leading to two different rate curves in Figure 3.4(a). In the second topology, sources are controlled by three different kinds of maximal cliques. Therefore there are three rate curves in Figure 3.4(b).

Table 3.1: Theoretical max-min allocation (Mbps) for static flows.

	f_1	f_2	f_3	f_4	f_5
Topology 1	1/3	1/3	2/3	1/3	1/3
Topology 2	1/4	1/4	1/3	1/3	2/3

3.5.2 Simulation with Dynamic Flows

In general, flows in the network change dynamically. In this section, we evaluate the performance of the proposed rate allocation scheme with dynamic flows. The topology is shown in Figure 3.5. Initially, there are five active flows, f_1 , f_2 , f_3 , f_4 and f_5 , as shown in Figure 3.5(a). The corresponding flow contention graph is shown in Figure 3.5(b). After 150 time intervals, f_1 leaves the network, which is shown in Figure 3.5(c). The flow contention graph becomes Figure 3.5(d). Then, at the 300th time interval, two other flows, f_6 and f_7 begin to be active. The corresponding flow contention graph is shown in Figure 3.5(f).

The simulation result is shown in Figure 3.6. In order to clarify the behavior of the scheme when some flows are joining or leaving the network, the rates of the three most representative flows are described in the figure. As is shown, before f_1 leaves the network, the rate allocation converges to $x_2 = 1/4$, $x_3 = 1/3$, $x_5 = 2/3$. Up until the 150th time interval, since no change in flow contention graph has occurred, these curves could be observed as the algorithm performance with static flows. When f_1 becomes idle, after a short transient period, the rate allocation again converges to $x_2 = 2/3$, $x_3 = 1/3$, $x_5 = 2/3$. Then f_6 and f_7 begin to be active and the rate allocation finally converges to $x_2 = 3/8$, $x_3 = 1/4$, $x_5 = 3/4$. For comparison, the theoretical allocation of each topology is listed in Table 3.2. Obviously, the proposed rate allocation scheme also works well with dynamic flows.

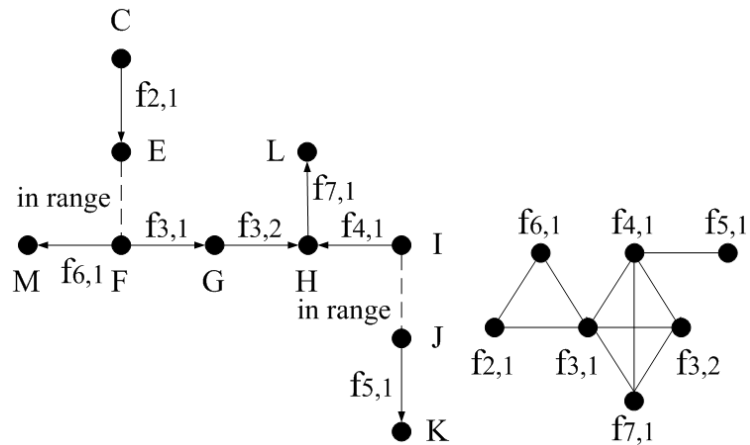
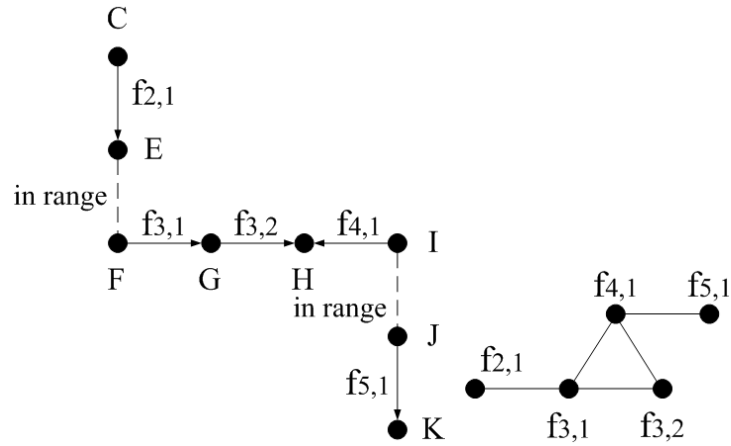
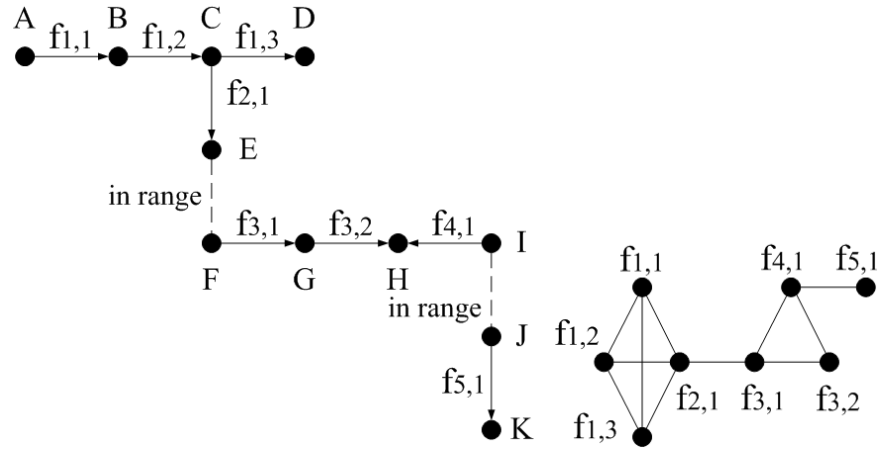


Figure 3.5: Topologies for dynamic flow simulation.

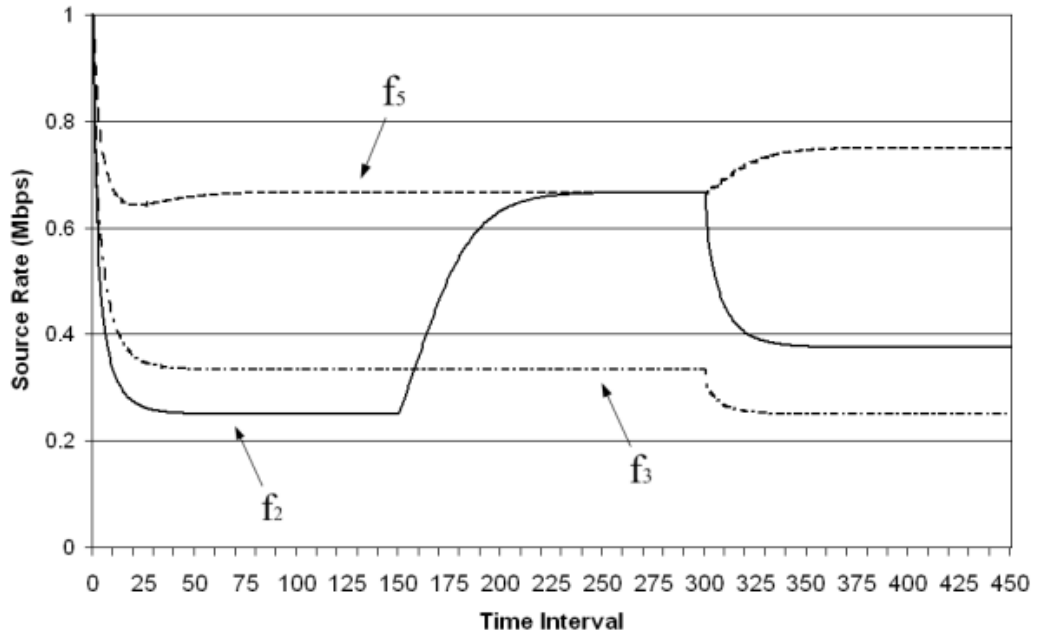


Figure 3.6: Simulation result of dynamic flows.

Table 3.2: Theoretical max-min allocations (Mbps) for dynamic flows.

	f_1	f_2	f_3	f_4	f_5	f_6	f_7
allocation 1	1/4	1/4	1/3	1/3	2/3	-	-
allocation 2	-	2/3	1/3	1/3	2/3	-	-
allocation 3	-	3/8	1/4	1/4	3/4	3/8	1/4

3.6 Fast Converging Approach

3.6.1 Approach Description

In order to accelerate the convergence speed, we enhance the rate allocation scheme as follows. From the simulation results, it is noted that there are two different behaviors of changing flow rates before entering the steady state. *Type I flow* is of the monotonously decreasing trend, for example, flow f_1 in Figure 3.4(a). If in a maximal clique, all the flows are controlled by this maximal clique, which means none of them are bottlenecked elsewhere, we call the flows in this maximal clique are Type I flow. It means that such flows all have the same price of this maximal clique and thus equally share the capacity of this maximal clique. The trend of *Type II flow* is first monotonously decreasing, and then after a turning point monotonously increasing. This characteristic arises from the fact that, in a maximal clique, one or more Type I flows are not controlled here and thus have greater price than this clique. The rates of such flows are determined by other more congested cliques elsewhere. Thus the remaining flows in this maximal clique, which are called Type II flow, can have better rates with the residuary capacity. For example, flow f_3 in Figure 3.4(a) is a Type II flow.

As we can see, before the flow rate allocation converges to steady state, the price-based scheme adjusts the rates gradually. Actually in steady state, for Type I flow, the capacity of the maximal clique is evenly distributed. Even for Type II flow, if we can make the turning point appear earlier, the rate might also converge faster. This characteristic inspires an approach to speed up the convergence speed. By carefully skipping some gradual adjustments, the overall convergence time should be reduced greatly. Incorporating the mechanism of event-driven and cross-layer concept, this approach can be realized. It is outlined in Figure 3.7.

Again, there is an assumption as we addressed earlier, i.e., all users have the same utility functions. In a wireless ad hoc network, all nodes are potential users that

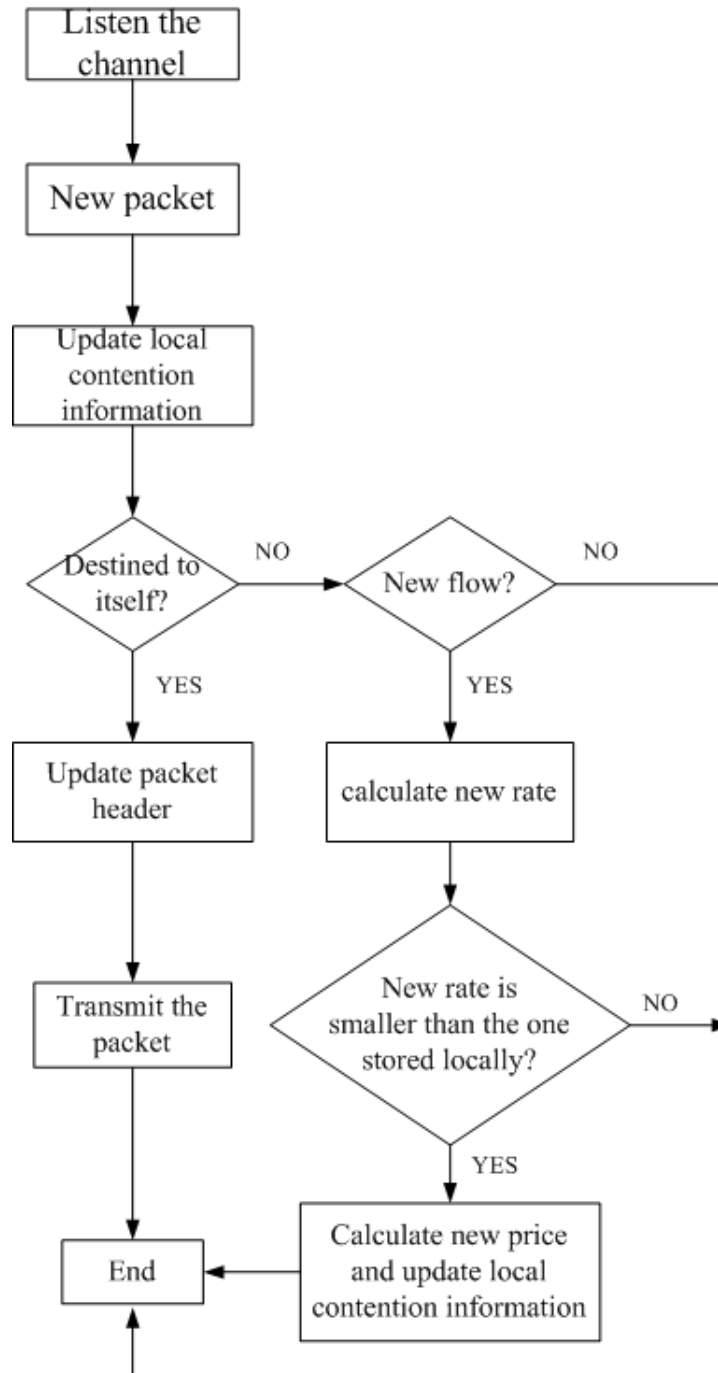


Figure 3.7: Outline of fast converging approach.

may initiate a flow. According to the assumption made above, all nodes in the network have the same utility function and demand function, which is the inverse of the utility function. When a flow is initiated in a node, it may not be aware of the contention situation around. The price of this flow is 0 and thus it tries to

send packets with rate equal to the capacity of the wireless link. After that, the source node adjusts the rate of the flow gradually according to the feedback prices. When one of the nodes which are traversed by this flow senses another active flow in the transmission range from the physical layer, this information is informed to higher layers. On receiving the information, this node determines whether this flow is in the local flow contention graph stored in itself. If this flow is not in the graph, then the node adds it into the graph and calculates the rate by dividing the clique capacity with the number of subflows in the maximal clique. It then compares this calculated rate with the one stored locally. If this rate is smaller, an event called *rate dividing* is triggered. Upon receiving this event, the node calculates the new price according to this rate and the utility function. Then the new price is locally stored. In this way, some of the gradual adjustments are skipped. When the price feedback returns to the source node, it calculates the rate as usual. It is noted that this approach is transparent for source nodes, since they do not know how the prices are determined. The only task they have to perform is calculating the rates according to the demand function and prices. It should be pointed out that in reality, a node can sense other active flows within its transmission range in a very short period. Also, as discussed before, to deploy the scheme in actual ad hoc networks, the information maintenance and price update need to be carried out by the nodes traversed by all subflows in the maximal clique. This approach is applicable to both static and dynamic flows.

3.6.2 Performance Evaluation

This section describes the simulation we have done to compare the performances of the original price-based approach and the fast converging approach. The topology graph and the corresponding flow contention graph are shown in Figure 3.8. There are initially four active flows in the network. After 100 time intervals, two flows join the network and change the pattern of flow contention graph. In order to clarify the detailed procedure of the fast converging approach, we assume that a node can

sense other contending flows in its transmission range within one time intervals. It should be noted that in reality, this period might be much shorter. However, the price feedback could only reach the source node after at least one time interval, then the rate adjustment will be carried out.

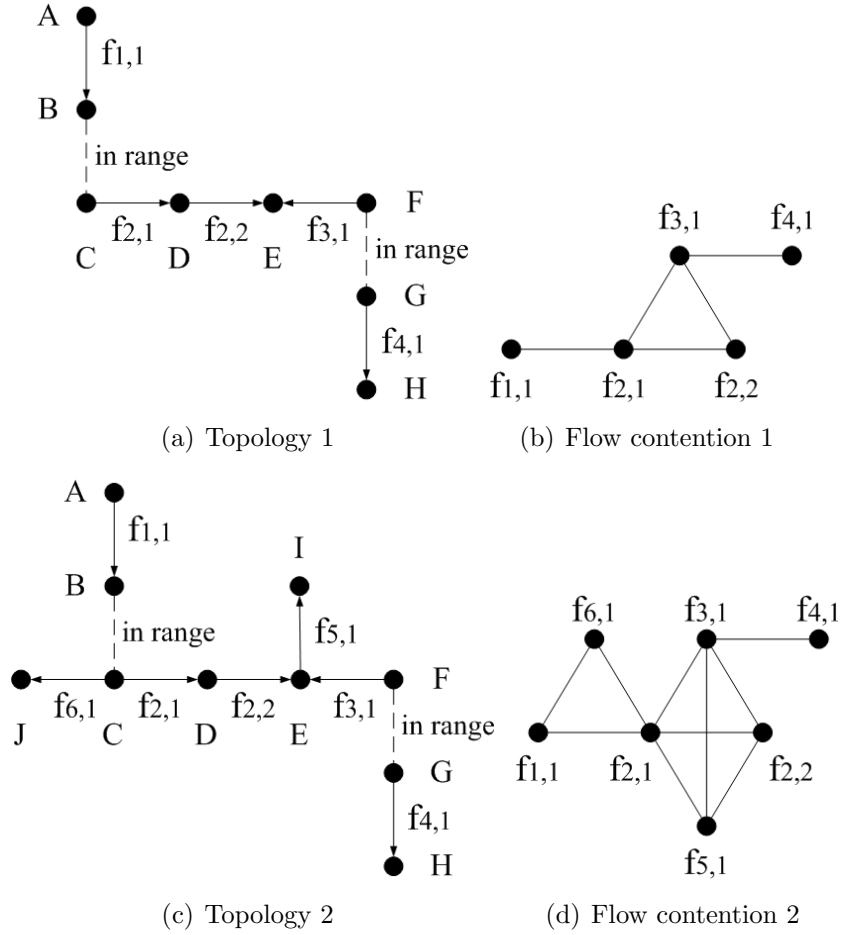
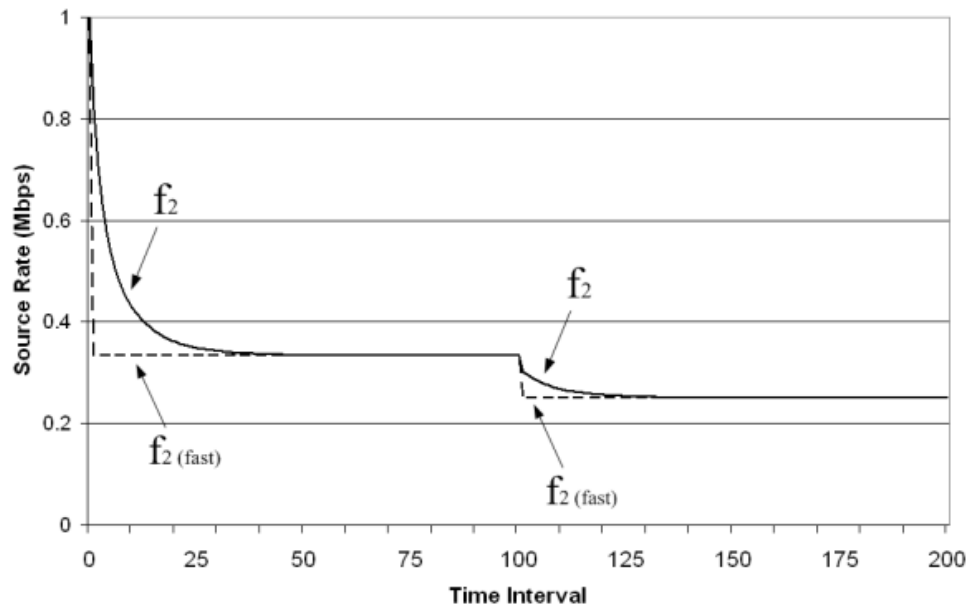


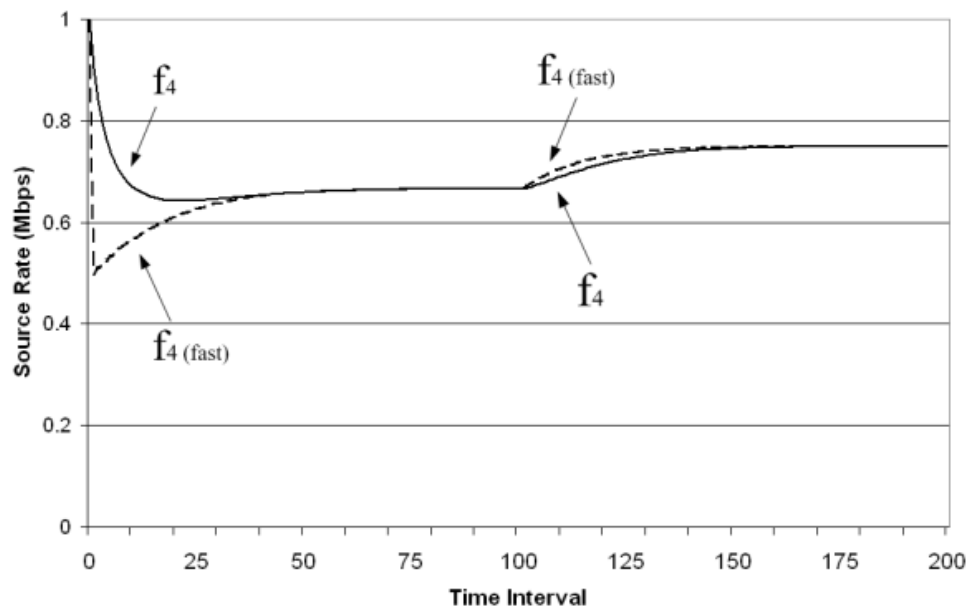
Figure 3.8: Topologies for fast converging approach simulation.

The simulation result is shown in Figure 3.9. It is noted that for different types of flows, the improvement varies. In Figure 3.9(a), the convergence time for f_2 , a Type I flow, is dramatically reduced. As soon as the node updates its local flow contention graph according to the sensing of new contending subflows, the rate can be adjusted to the appropriate value within one time interval. Regarding the convergence time for Type II flows, such as f_4 in Figure 3.9(b), it is also improved; however, the improvement is not as significant as that of Type I flows. This is mainly caused by

the fact that the convergence time of a Type II flow is also affected by the state of the other Type I flows in its controlling maximal clique.



(a) Flow 2, a Type I flow



(b) Flow 4, a Type II flow

Figure 3.9: Simulation result of fast converging approach.

3.7 Extension to Weighted Max-min Fairness

Sometimes, in order to reflect different QoS demand, each user s is pre-assigned with a weight w_s . Higher weight leads to more allocated bandwidth. So in this part, the proposed scheme is modified to support *weighted max-min fairness*, which is defined as follows.

Definition 2. For a feasible rate allocation X , comparing to any other feasible rate allocation S , if there is a i such that $\frac{x_i}{w_i} < \frac{s_i}{w_i}$, and there is a j such that $\frac{x_j}{w_j} \leq \frac{x_i}{w_i}$ and $\frac{x_j}{w_j} > \frac{s_j}{w_j}$, then X is said to be *weighted max-min fair allocation*.

If we call $\frac{x_i}{w_i}$ as the *normalized rate*, similarly, a feasible rate allocation is said to be weighted max-min fair if no normalized rate can be increased while maintaining feasibility without decreasing that of any other flow having equal or less normalized rate. Take Figure 3.1 for example, the max-min fair allocation is $x_1 = x_2 = 1/3$. If we set the weight of each flow as $w_1 = 1$, $w_2 = 2$, then the weighted max-min fair rate allocation should be $x_1 = 1/5$, $x_2 = 2/5$.

Weighted max-min fairness can also be achieved via our algorithm. The only modification is the demand function of users with different weights. The demand function we used in previous sections is $x_s(t+1) = x_{max}e^{-k \cdot q_s(t)}$. To support weighted max-min fairness, this function should be modified to $x_s(t+1) = w_s \cdot x_{max}e^{-k \cdot q_s(t)}$.

Flows with different weights are also simulated, for example, the topology in Figure 3.3(c). If we assign the weights of flows as $w_1 : w_2 : w_3 : w_4 : w_5 = 1 : 3 : 1 : 2 : 2$. All the simulation parameters are the same as in Section 3.5, except for the demand functions. As the result, the flow rates converge to $x_1 = 1/6$, $x_2 = x_4 = x_5 = 1/2$, $x_3 = 1/4$, which is a weighted max-min fair allocation.

3.8 Contributions of Our Work

A price-based max-min fair rate allocation scheme for wireless ad hoc networks is proposed in this chapter. The main differences between this scheme and some previous ones are summarized as follows:

1) *Clique-based price is employed, which more accurately reflects the interference-limited characteristic.*

Some previous works, such as [34][35][41], assume that nodes in a neighborhood have different frequencies and only require that no node simultaneously transmit and receive packets, which is essentially the MAC constraint. Instead of using this restrictive assumption, our scheme generates prices at each maximal clique, which could more accurately reflect the interference-limited characteristic of ad hoc networks.

2) *End-to-end flow rate is considered.*

[22] and [31] consider the rate allocation from the MAC layer perspective. In their approaches, each multi-hop flow is broken into multiple independent subflows. Each subflow of the same multi-hop flow might be assigned with different rates. This approach has some significant limitations as we discussed earlier. Therefore, the scheme proposed here considers the end-to-end rates of multi-hop flows instead of the one-hop rates of subflows.

3) *Both static and dynamic flows could be easily handled.*

Many previously proposed schemes [22]-[25] assume that the network configuration is composed of static or quasi-static nodes, in other words, changes in the topology only occur in a much larger time scale than the time required for the algorithm to converge to a fair allocation. Also, for some other schemes, when applied to dynamic flows, the iteration process might have to restart from the very beginning. These schemes might have shown good performances with static flows; however, the

performances with dynamic flows are often not studied and remain unknown. In this chapter, our scheme is evaluated in multiple scenarios. The results show that the proposed scheme works well with both static and dynamic flows.

4) The implementation is totally distributed.

Only the centralized algorithm in [25] could lead to optimal solutions to the rate allocation optimization problem. Moreover, as [40] indicates, almost all the previous papers make an assumption that congestion price information can be instantaneously exchanged between all the nodes. On the contrary, as we have discussed, the proposed scheme is totally distributed. In addition, by including two appropriate fields in the packet header, all the nodes in the same maximal clique compute the clique-associated price individually, which means that there is no need to exchange price information between nodes.

3.9 Conclusion

The interference-limited characteristic of wireless ad hoc networks makes the rate allocation for multi-hop flows very challenging. In this chapter, a price-based rate allocation scheme for multi-hop wireless ad hoc networks has been proposed. The objective of this scheme is to achieve max-min fairness among different flows. To accurately reflect the clique constraint of wireless ad hoc networks, the clique-based price is generalized to act as the congestion signal, which controls the end-to-end rates of multi-hop flows. The distributed algorithms adopted are simple and direct, thus owning low computational complexity. Through analysis and simulation, the proposed scheme is shown to be able to effectively distribute the bandwidth, according to max-min fairness criterion, to multi-hop flows from the end-to-end perspective. In order to accelerate the convergence speed of the algorithm, we have also proposed a fast converging approach which significantly reduces the convergence time. Finally, the proposed scheme has been extended to support weighted max-min fairness criterion.

Chapter 4

MAC Protocol for Bandwidth Allocation

4.1 Introduction

Based on the rate allocation scheme proposed in Chapter 3, the bandwidth is distributed to the contending flows and target flow rates are calculated, according to a certain fairness criterion. After the target rates are calculated in higher layers by the rate allocation schemes, a MAC protocol should support these schemes by allowing each flow to obtain its entitled bandwidth through medium access control. Unfortunately, the current MAC protocols, such as IEEE 802.11 DCF and other variants, are usually based on random access or intuitional fairness, and thus cannot satisfy this requirement. Therefore, a specially designed MAC protocol is needed.

In order to support the rate allocation schemes at the MAC layer, we propose a new MAC protocol in this chapter. The objective of this MAC protocol is, given the rates calculated by a rate allocation scheme at the network layer, to guarantee the allocated flow rates to be achieved. As a result, it is a cross-layer MAC protocol. It is noted that the MAC design proposed in this chapter is not trying to differentiate service types for different traffic flows, which is being discussed by IEEE 802.11e standard [4]. Instead, the aim of our work is to support the rate allocation schemes existing in higher layers.

The rest of this chapter is organized as follows. Some related work are discussed in Section 4.2. Section 4.3 discusses the framework of this MAC protocol, and in Section 4.4 and 4.5, two algorithms are described and analyzed, respectively. Section 4.6 concludes the whole chapter.

4.2 Related Work

Through adjusting the persistence probability, [31] tries to achieve MAC layer fairness. A mechanism for translating a given fairness model into a corresponding contention resolution algorithm is proposed. According to this translation procedure, by carefully choosing the adjustment function, some specific fairness criteria are proved to be achieved. This work provides the path for achieving fairness objectives over shared wireless channels; however, as the authors pointed out, as wireless networks move from the academic domain to the commercial domain, these mechanisms must support network level quality of service, which needs to be translated to MAC level fairness objective. In [22], max-min fairness is achieved by scheduling the access of single-hop flows to the medium, according to the calculated fair share. However, the achieved max-min fairness is for MAC layer subflows, not for the end-to-end multi-hop sessions. A two-tier distributed and iterative algorithm is proposed in [23]. The first tier algorithm completes the per-clique price calculation, and then the second tier algorithm performs per-node price calculation. At this point, the transmission of each node is scheduled according to per-node price. After careful choice of utility functions, proportional fairness and max-min fairness could be achieved. However, the iteration process of this algorithm has a relatively long convergence time, which might limit the application scope of this algorithm. The scheduling algorithm for arbitrary rate allocation adopted in [25] is based on tag marking and virtual clock. This algorithm needs to select the next packet to transmit from the head of the queues according to the allocated share. Each node maintains a virtual clock and uses the tag to estimate an approximate backoff value

for each packet queued in its buffer. This algorithm only guarantees that subflows from the same flow receive approximately the same channel share. [34] combines a link scheduling that avoids collisions, a fair session service discipline per link, and a hop-by-hop window flow control into one algorithm. The scheduling module relies on the existing maximum difference backlog scheduling approach to schedule the transmission of the released packets so as to attain the fairness estimates. The max-min fair scheduling is achieved by generating tokens at source nodes and max-min packets releasing process. This scheduling module is not built for arbitrary rate allocation schemes. IEEE 802.11e [4] has been approved in late 2005 as a standard that defines a set of QoS enhancements for 802.11 Wi-Fi standard. Enhanced Distributed Channel Access (EDCA) is proposed as an improvement of DCF access method by defining priority for different type of traffic flows. However, it is not capable of ensuring the bandwidth entitlement for each multi-hop flow.

4.3 The Framework of MAC Protocol for Bandwidth Allocation

This section discusses the framework of our proposed MAC protocol for bandwidth allocation in wireless ad hoc networks. Recalling that flows from different maximal cliques could be active simultaneously, while flows within the same clique contend for the shared bandwidth, the MAC protocol would thus basically coordinate the data transmission of different subflows within each maximal clique. In this framework, the transmission time is slotted and packets are sent out at the beginning of each time slot. In each maximal clique, every N time slots are defined as a *transmission frame*. Each transmission frame is regarded as a unit to estimate the share of the bandwidth within the corresponding maximal clique. The value of N is related to the precision of the calculated flow rates provided by the rate allocation schemes. Let $x_{i,j}$ and c_l represent the flow rate of $f_{i,j}$ and effective bandwidth of link l respectively, without losing generality, in the following discussion, we assume $N =$

100. Therefore, for a flow to achieve a rate of $1/3$, it has to acquire 33 slots in each frame.

In each node, packets from different subflows are queued in different buffers. All the nodes maintain a *clique occupancy table* (COT) for each maximal clique to which they are related. Therefore, for some nodes in the network, they will maintain multiple COTs. For example, in Figure 4.1, node A maintains a COT for maximal clique $\Omega(f_{1,1}, f_{1,2})$, while node C maintains two COTs respectively for maximal cliques $\Omega(f_{1,1}, f_{1,2})$ and $\Omega(f_{1,2}, f_{2,1}, f_{2,2})$. The length of each COT equals N , according to the length of the transmission frame. The content of the COT describes the occupancy result within the corresponding maximal clique. When a subflow $f_{i,j}$ successfully occupies a free time slot in a frame, all the nodes in the same maximal clique will update their COTs to mark the time slot with $f_{i,j}$.

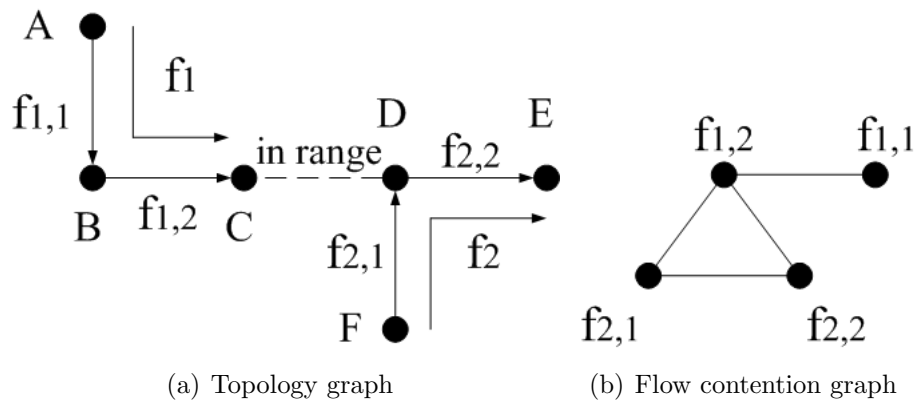


Figure 4.1: Network example.

Also, in subsequent frames, each subflow will send packets at the same time slots that it has already occupied, i.e., marked with its ID in COTs. If the number of occupied time slots is not enough to satisfy the calculated rate (this situation happens when the allocated rate is not yet achieved or some flows have left the maximal clique), more time slots will be occupied by this subflow. Similarly, if the number of occupied time slots exceeds its entitlement (this situation happens when some new flows have joined the maximal clique), the subflow would release some occupied time slots accordingly. In this sense, the proposed MAC protocol is

compatible with both static and dynamic flows.

Based on the above framework of the MAC protocol, two different MAC algorithms will be discussed in the next two sections, respectively.

4.4 Greedy Self-Contention Algorithm

4.4.1 Algorithm Description

Greedy Self-Contention (GSC) algorithm is a contention-based algorithm. In multiple access protocols, contention resolution has typically been achieved through two mechanisms: persistence and backoff [31]. The persistence probability and backoff window are functions of the estimated contention, and different contention resolution algorithms differ in terms of how they adjust these parameters in response to collisions and successful transmissions.

In each maximal clique, the normalized bandwidth bound never exceeds 1 and for each subflow $f_{i,j}$, it occupies the medium with a time fraction of $x_{i,j}/c_l$. This time fraction has a natural interpretation as a probability. Therefore, in GSC, each subflow uses $p_{i,j} = x_{i,j}/c_l$ as the persistence probability to contend for the free time slot. Initially, all COTs are empty, which means that all time slots in a transmission frame are free. At the beginning of a free time slot, each subflow transmits a packet with probability $p_{i,j}$. If subflow $f_{i,j}$ is the only subflow transmitting a packet, it would successfully obtain this time slot. All the other nodes in the same maximal clique that overhear this transmission mark the corresponding time slot in their own COTs with $f_{i,j}$. If collision happens, no subflow successfully obtains this time slot and thus no mark is made. Since the acquired bandwidth is estimated within each transmission frame, if some time slots are not occupied in the current frame, they could only be occupied in next frame. At this standpoint, in order not to affect the convergence time and to have a simple implementation approach, GSC algorithm does not adopt any backoff mechanism.

When the number of occupied time slots $n_{i,j}$ for subflow $f_{i,j}$ satisfies the equation:

$$\frac{n_{i,j}}{N} = \frac{x_{i,j}}{c_l}, \quad (4.1)$$

it adjusts the persistence probability to zero. In other words, it would not contend for free time slots any more. This equation implicitly shows that the bandwidth acquired by subflow $f_{i,j}$ is already equal to its entitlement.

In subsequent frames, each subflow will transmit in the time slots marked with its ID in the COT. If the time slot is marked with other subflows, it will not contend. In this way, in those marked time slots, the transmission is contention-free. The flow chart of this algorithm is shown in Figure 4.2.

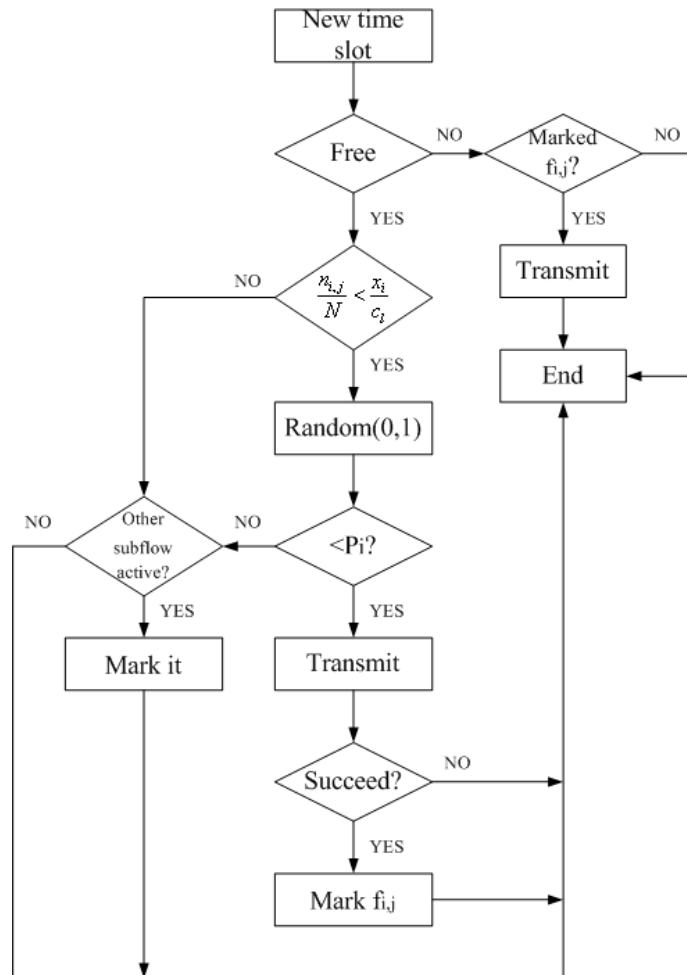


Figure 4.2: Flow chart of GSC algorithm for subflow $f_{i,j}$

Finally, this contending process converges to steady state, in which all the transmission frames have the same content, i.e., the active subflows in corresponding time slots are the same. In steady state, every subflow occupies a number of slots according to Equation (4.1) in order to obtain its entitled bandwidth.

4.4.2 Related Discussions

From the process described above, it can be seen that given $x_{i,j}/c_l$, each subflow tries to occupy $n_{i,j}$ out of N time slots in a frame, where $\frac{n_{i,j}}{N} = \frac{x_{i,j}}{c_l}$. After it successfully occupies $n_{i,j}$ time slots, its persistence probability is set to zero and it will not contend for free time slot anymore. Therefore, no subflow could occupy more time slots than it is entitled to.

In order to simplify the notations, in the following discussion, we assume there are n subflows in the maximal clique Ω and use f_i to represent the subflow. We define x_i as the rate of subflow f_i and n_i as the number of time slots it should obtain according to the allocated rate.

1) *Clique occupancy table exchange*

Based on Proposition 3 in Chapter 3, if COTs are exchanged between the two ends in each transmission pair, all the nodes in a maximal clique can maintain the COTs with the same content. Note that the COT exchange only occurs within a transmission pair, so the control information exchange is bounded locally by each maximal clique. In addition, COTs can be piggybacked in data/ACK packets to be exchanged between a transmission pair, thus no new packet format needs to be introduced.

2) *Mismatch of COTs between a transmission pair*

The COTs between a transmission pair should match each other. But sometimes mismatch may happen. Consider the situation in Figure 4.3, where there is a maximal clique $\Omega(f_1, f_2, f_3)$. Consider subflows f_1 and f_2 are both trying to transmit

a packet in the same free time slot. Of course collision happens and no one should mark the slot. However, node E is out of the range of subflow f_2 and just overhears the transmission of subflow f_1 . It does not know collision has happened and according to the algorithm, it will mark the corresponding time slot in its own COT with f_1 . Same thing happens to node F and it will mark the corresponding time slot with f_2 . So mismatch between node E and F happens.

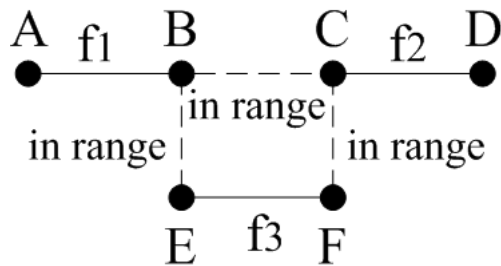


Figure 4.3: Mismatch of clique occupancy tables

To deal with this problem, the following rule is used to regulate the COT exchange between a transmission pair. When the COT from the other end arrives, this end first compares it with the one of its own. If it finds that for a corresponding time slot, two COTs are marked with two different subflows, it concludes collision has happened. It then deletes the subflow marked in this time slot and inform the other end this situation. Only if its own COT is empty in a time slot, it copies the corresponding entry in the COT of the other end.

4.4.3 Convergence Time Evaluation

The upper bound of the average convergence time in each maximal clique is derived in this subsection. We assume in a maximal clique Ω , there are n subflows. Let ω be the set of these n subflows. Before obtaining the entitled bandwidth, each subflow f_i will contend for every free time slot with probability $p_i = x_i/c_l$. The probability to obtain a free time slot for this subflow is $(p_i \prod_{\substack{j \in \omega \\ j \neq i}} (1 - p_j))$. On average, after contending for $(1/[p_i \prod_{\substack{j \in \omega \\ j \neq i}} (1 - p_j)])$ free time slots, subflow f_i can successfully

transmit once. Let N_i denotes the number of free time slots that subflow f_i has contended for before obtaining n_i time slots.

Assume subflow f_a is the first one to obtain its entitled bandwidth, i.e., obtaining n_a time slots it needs. Then the average number of free time slots it has contended for should be

$$N_a = \frac{n_a}{p_a \prod_{\substack{i \in \omega \\ i \neq a}} (1 - p_i)} \quad (4.2)$$

After subflow f_a gets n_a time slots, its persistence probability is set to zero. If subflow f_b is the second one to obtain its entitled bandwidth, then on average the number of free time slots it has contended for should be

$$N_b = \frac{n_{ba}}{p_b \prod_{\substack{i \in \omega \\ i \neq b}} (1 - p_i)} + \frac{n_{bb}}{p_b \prod_{\substack{i \in \omega \\ i \neq a, i \neq b}} (1 - p_i)}, \quad (4.3)$$

where n_{ba} is the number of time slots f_b obtains before subflow f_a obtains its entitled bandwidth, n_{bb} is the number of time slots f_b obtains between the time f_a obtains its entitled bandwidth and the time it obtains its entitled bandwidth. It is obvious that $n_{ba} + n_{bb} = n_b$.

Then subflow f_b also stops contending for more time slots. Following this way, for the last subflow f_k , the average number of free time slots it has contended for is

$$N_k = \frac{n_{ka}}{p_k \prod_{\substack{i \in \omega \\ i \neq k}} (1 - p_i)} + \frac{n_{kb}}{p_k \prod_{\substack{i \in \omega \\ i \neq a, i \neq k}} (1 - p_i)} + \dots + \frac{n_{kk}}{p_k}, \quad (4.4)$$

where $n_{ka} + n_{kb} + \dots + n_{kk} = n_k$.

Let us consider the worst situation: all the subflows occupy the required time slots strictly one by one, i.e., before subflow f_a obtains n_a time slots, the other subflows do not obtain any time slot, then subflow f_b obtains n_b time slots followed by f_c obtaining n_c time slots and so on. In this situation, the convergence time is the longest one. Hence, (4.3) becomes

$$N_b = \frac{n_b}{p_b \prod_{\substack{i \in \omega \\ i \neq a, i \neq b}} (1 - p_i)}, \quad (4.5)$$

which means that before subflow f_a obtains all the n_a time slots, f_b does not obtain any time slot. And (4.4) becomes

$$N_k = \frac{n_k}{p_k} \quad (4.6)$$

Let subflow f_a be the first one to obtain its entitled bandwidth. In the first frame, since all the time slots are free, the number of time slots it can obtain in this frame is

$$n_a(1) = N \cdot p_a \prod_{\substack{i \in \omega \\ i \neq a}} (1 - p_i) \quad (4.7)$$

Note that it is a statistical analysis here, so $n_a(1)$ may not be an integer.

Then in the second frame, $n_a(1)$ time slots are already occupied by f_a and the number of free time slots is $(N - n_a(1))$. So it can obtain

$$n_a(2) = (N - n_a(1)) \cdot p_a \prod_{\substack{i \in \omega \\ i \neq a}} (1 - p_i) \quad (4.8)$$

more time slots in the second frame.

Repeat the calculation in this way, finally in the ξ_a frame, after obtaining

$$n_a(\xi_a) = (N - \sum_{j=1}^{\xi_a-1} n_a(j)) \cdot p_a \prod_{\substack{i \in \omega \\ i \neq a}} (1 - p_i) \quad (4.9)$$

more time slots, subflow f_a obtains $n_a = \sum_{j=1}^{\xi_a} n_a(j)$ time slots in total and sets its persistence probability to zero.

As we assumed above, in the next frame after ξ_a , subflow f_b begins to obtain time slots. In this frame, since subflow f_a has already occupied n_a time slots, the number of free time slots is $(N - n_a)$. So in its first frame, f_b can get

$$n_b(1) = (N - n_a) \cdot p_b \prod_{\substack{i \in \omega \\ i \neq a, i \neq b}} (1 - p_i) \quad (4.10)$$

In the same way, for its second frame, f_b gets $n_b(2) = (N - n_a - n_b(1)) \cdot p_b \prod_{\substack{i \in \omega \\ i \neq a, i \neq b}} (1 - p_i)$ time slots, and in its last frame, $n_b(\xi_b) = (N - n_a - \sum_{j=1}^{\xi_b-1} n_b(j)) \cdot p_b \prod_{\substack{i \in \omega \\ i \neq a, i \neq b}} (1 - p_i)$. At

this moment, f_b obtains its entitled bandwidth with $n_b = \sum_{j=1}^{\xi_b} n_b(j)$ and sets its persistence probability to zero.

If we repeat the calculation on the last subflow f_k in the same way, until after a specific frame ξ_k , the remaining time slot for f_k is one.

$$n_k(\xi_k) = (N - \sum_{\substack{i \in \omega \\ i \neq k}} n_i - \sum_{j=1}^{\xi_k-1} n_k(j)) \cdot p_k \quad (4.11)$$

Then f_k will spend $(1/p_k)$ more frames to obtain the last time slot and obtain its entitlement with $n_k = \sum_{j=1}^{\xi_k} n_k(j) + \frac{1}{p_k}$.

After f_k obtains n_k time slots, all the subflows have obtained their entitled bandwidth and bandwidth allocation is completed. As we assumed that subflows obtain the time slot one by one, so the overall convergence time is

$$\xi = \sum_{i \in \omega} \xi_i + \frac{1}{p_k} \quad (4.12)$$

This means that after ξ frames, each subflow f_i have obtained required number of time slots n_i and the allocated entitlement is achieved. Note that Equation (4.12) provides an upper bound of the average convergence time. The average convergence time should be less than this value since subflows are not constrained to obtain the time slots strictly one by one.

We calculate the upper bound of the average convergence time following the approach described above in three different cases and carry out 1000 simulations for each case to obtain the average convergence times. The results are shown in Table 4.1. As mentioned before, the convergence time is usually less than that in the worst situation.

Table 4.1: Average convergence time of GSC algorithm (frames).

	Upper Bound (analysis)	Simulation
$N = 10, f_i = 1/3$	10	8.6
$N = 100, f_i = 1/3$	12	8.6
$N = 100, f_1 = 1/2, f_2 = f_3 = 1/4$	18	13.4

4.5 Cooperative Token Forwarding Algorithm

4.5.1 Algorithm Description

The deficiency of GSC algorithm is the relatively long convergence time due to its contention operation. Some recently proposed rate allocation schemes, for example, the scheme we proposed in Chapter 3 and the ones in [23][25][36], have an inherent characteristic that the local flow contention graph and the local contention information are available among all nodes in the same maximal clique. In other words, nodes in the maximal clique could be regarded as quasi-synchronized with the same flow contention information.

Cooperative Token Forwarding (CTF) algorithm takes advantage of this inherent characteristic, and thus eliminates the contention among nodes in a maximal clique. In CTF, each node is assigned with a priority determined by its MAC address. Since all the nodes in the same maximal clique are aware of the flow information within the clique according to the rate allocation scheme, the assigned priorities are also available to all of them. Nodes with smaller MAC addresses are assigned with higher priorities (the reverse ordering is also applicable). In other words, if the sending node of a subflow has the smallest MAC address, this subflow has the highest priority to acquire the time slots. Some subflows may share the same sending node. In this case, there is no need to distinguish these subflows from different priorities. When such a sending node has acquired the right for transmission, it will internally schedule the transmission of its subflows. From the discussion above, it is noted that the priority assignment is naturally based on the quasi-synchronous characteristic

of the rate allocation scheme, and thus no central coordinator is needed.

When a node joins the maximal clique, it generates a token. The token is a logical symbol with which a node could transmit data packets. After this node overhears the medium and exchanges information with one-hop neighbors, if it finds another node with higher priority, it would destroy the token generated by itself. In this way, at the beginning of the algorithm, all the nodes generate a token themselves, and then only one token which is generated by the node with the highest priority is maintained in each maximal clique.

Only the node with the token is eligible to transmit data packets, other nodes would just listen to the medium and update their COTs. When a node obtains the token, it should first check whether it still needs more time slots in this transmission frame, and whether it is ready for transmission. If both conditions are satisfied, it will begin transmission; otherwise, it would forward the token to the next node. Initially, the node with the highest priority obtains the token and keeps on transmitting data packets and occupies continuous time slots. When the number of occupied time slots satisfies Equation (4.1) for all subflows, the node forwards the token to another node with the second highest priority. Then this node repeats the procedure and forwards the token to the next one. When the node with the lowest priority satisfies its entitlement, it returns the token to the node with the highest priority so as to launch a new round of transmission. When the token is lost due to unexpected node failure or link failure, the node with the highest priority is eligible to generate a new token and continue the transmission. This operation continues until all the subflows satisfy their bandwidth entitlement.

4.5.2 Related Discussions

1) *Token forwarding strategy*

After occupying enough time slots to obtain its entitled bandwidth, the node which holds the token will forward it to the next node. However, this strategy might cause

a problem. Before being eligible to transmit, nodes with lower priorities have to wait for the nodes with higher priorities to complete their transmissions in a frame. Therefore, nodes with lower priorities are likely to occupy the later portion of a transmission frame. This may lead to the situation that packets accumulate and cause congestion at those nodes with lower priorities.

In order to solve this potential problem, the token forwarding strategy could be modified as follows: when a subflow successfully occupies $n'_{i,j}$ time slots so that $\frac{n'_{i,j}}{N} = p\% \times \frac{x_{i,j}}{c_i}$, it would forward the token to the next subflow. Here $p\%$ is a tunable parameter. For example, if $p\% = 20\%$, the subflow with the token will forward it to the next subflow after it satisfies 20% of its bandwidth entitlement. This strategy could give opportunity to subflows with lower priorities to transmit data packets at earlier time and thus help solve the potential problem. In addition, this strategy would not affect the ultimate bandwidth allocation since the rate of each subflow is still bounded by its entitlement.

The operation of token forwarding is implemented by adding a new header bit in each packet, called *transmission completion*. When the host who currently holds the token decides that it will continue transmitting packets in the next time slot, the transmission completion bit is set to 0 for current packet. During the last time slot when the host is ready to forward the token to the next host, it sets this bit as 1 in the last data packet. Then the host with next priority will notice the completion of the last host's transmission and get ready to transmit its own data packets in the next time slot. In this way, the operation of token forwarding is completed and no additional operation time is needed.

2) Subflows related to multiple cliques

An example of token forwarding is shown in Figure 4.4, in which there are 3 maximal cliques in the flow contention graph. It should be noted that there is one token for each maximal clique. For subflows belonging to multiple cliques, it is required that they obtain all the tokens in the related cliques before they can transmit their data

packets. For example, in Figure 4.4, f_1 belongs to two maximal cliques and it must obtain both Token 2 and Token 3 to become active.

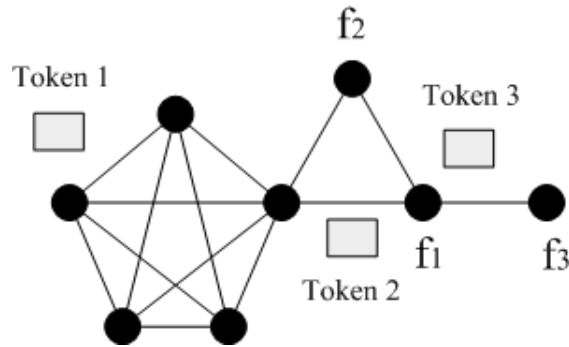


Figure 4.4: Example of token forwarding

It is crucial that f_1 could occupy appropriate time slots at the respective transmission frames of the two maximal cliques. We impose the restriction that if it obtains one of the two tokens, the sending node will hold it and wait for the other token. This restriction tries to guarantee that flows belonging to multiple cliques are able to occupy appropriate time slots in different cliques, at the cost of convergence time.

It is noted that if the subflows belonging to multiple cliques could get the token in each clique at a earlier time, the convergence time would be improved. Therefore, we introduce the following strategy.

Definition 3. *The degree of a subflow refers to the number of maximal cliques it is related to.*

For example, in Figure 4.4, f_1 belongs to two maximal cliques, so its degree is two. A subflow must get the number of tokens that equals its degree in order to be active for transmission. When allocating the priority, higher priorities are allocated to subflows with higher degrees. If more than one subflows in the same maximal clique have the same degree, the one with smaller MAC address of sending node would get higher priority. In this way, priority is related to the degree of each

subflow thus guaranteeing subflows with higher degrees would be in advantageous circumstance to get the token.

The above strategy could give higher priority to subflows with higher degree. However, since subflows belonging to multiple cliques would hold the arrived tokens before it obtains all the tokens, there is possibility that a special case of token deadlock would occur. See the ring flow contention graph in Figure 4.5.

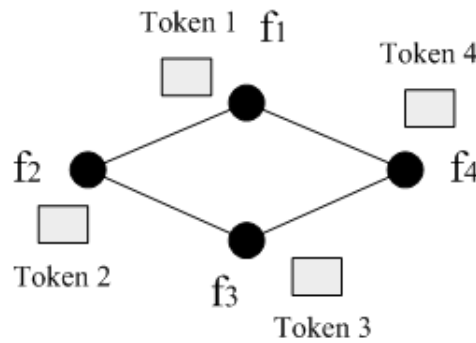


Figure 4.5: Token deadlock

Four subflows constitute four cliques. All the subflows have the same degree of 2. If all of them could only obtain one token at the same time and hold it so as to wait for the other token, none of them would obtain two tokens and thus no subflow could transmit data packets. Actually, the priority allocation strategy mentioned above guarantees that this token deadlock case would normally not occur. In clique $\Omega(f_1, f_2)$, f_1 obtains Token 1 first, which means it has a smaller MAC address than f_2 , i.e., $Addr_1 < Addr_2$. Similarly, $Addr_2 < Addr_3$ in clique $\Omega(f_2, f_3)$, and $Addr_3 < Addr_4$ in clique $\Omega(f_3, f_4)$. Add all the inequations together and it can be seen that $Addr_1 < Addr_2 < Addr_3 < Addr_4$. In this sense, f_1 should have higher priority than f_4 in clique $\Omega(f_1, f_4)$ and thus it could obtain Token 4 at the same time, if the whole algorithm works correctly. Even if some errors occur and the situation in Figure 4.5 happens, the problem could be solved by imposing a timeout threshold for each subflow to hold the tokens. This threshold is randomly generated from a normal distribution and when the time expires, the subflow should forward the held tokens to others.

4.5.3 Convergence Time Evaluation

Due to the cooperative operation within each maximal clique, CTF algorithm has reduced the convergence time remarkably. If we consider the operation in a separate maximal clique, the bandwidth allocation would be completed within one transmission frame. Therefore, there is no need to calculate the upper bound of convergence times in each maximal clique as we did for GSC algorithm.

In order to evaluate the performances of CTF algorithm in multiple cliques, we carry out 1000 simulations for the flow contention graph shown in Figure 4.4. In addition, the effect of different values of $p\%$ is also investigated. The average convergence times obtained are shown in Table 4.2.

As expected, the convergence time achieved by CTF algorithm is much shorter than GSC algorithm. Moreover, smaller value of $p\%$ would more effectively solve the potential congestion at nodes with lower priorities, but it would prolong the convergence time.

Table 4.2: Average convergence time of CTF algorithm (frames).

$p\%$	20%	50%	100%
time	7.4	3.2	1.9

4.6 Conclusion

A novel MAC protocol for bandwidth allocation in wireless ad hoc networks is proposed in this chapter. Given the flow rates allocated at the network layer, this MAC protocol ensures flows to achieve the target rates. Two different algorithms are implemented in this protocol. The first one is a contention-based algorithm and could support arbitrary rate allocation schemes. In order to accelerate the convergence speed, the second algorithm is specifically designed for the category of quasi-synchronous rate allocation schemes and thus its convergence performance is

much better. Both of the MAC algorithms are distributed and easy to implement.

Chapter 5

Conclusions

5.1 Summary of the Thesis

This thesis consists of three main chapters. In the first main contribution of this thesis, we improve the first step of resource management in mobile ad hoc networks by proposing a cross-layer based reliable routing protocol, called Robust Link Availability Routing (RLAR) protocol [44]. RLAR combines pro-active routing with on-demand routing. Each node in the network maintains two tables. The first one is Link Quality Table (LQT) which stores the quality information of wireless links from the node to its one-hop neighbors. Due to the mobility of nodes in MANETs, the content of LQT should be refreshed from time to time. It is proposed to use the beaconing mechanism for general update of LQT. With the use of the beacon signal, this part of RLAR is similar to the pro-active schemes. The other table is Routing Table (RT) as mentioned in other routing protocols. In RLAR, the information in RT is updated when a transmission session is initiated. In other words, the routing procedure uses the mechanism of on-demand routing.

The two modules of RLAR are responsible for pro-active estimation of link reliability and on-demand optimal routing, respectively. We propose a new reliability estimation metric, called normalized link availability (NLA), to act as the main routing metric. The first module estimate NLA from the physical layer information, and then the second module takes charge of route discovery and maintenance based

on NLA. One strength of RLAR is that it accurately estimates link reliability from physical layer information, therefore it works well with dynamic network topologies. Based on estimated link reliability, the optimal routing tree is constructed. According to a reliability threshold, additional protective links are also added to the tree backbone to form the ultimate mesh structure so as to further enhance reliability. From simulations, RLAR is proved to successfully reduce the number of reroutings and improve the average communication time before a rerouting procedure is triggered. The packet delivery ratio is also remarkably increased. In addition, it has been shown that the performance of RLAR can be further improved by more precise estimation of link reliability.

In the second main contribution, a rate allocation scheme in the network layer is proposed to achieve max-min fairness among different flows [36][46]. Max-min fairness guarantees equal bandwidth is assigned to flows that traverse paths of similar congestion level, and transmit packets at equal rates. The basic idea behind max-min fairness is to first allocate equal bandwidth to all contending users. If a user cannot utilize its bandwidth, because of constraint elsewhere, then the residual bandwidth is distributed among others. Thus, no user is penalized excessively, and a certain minimum quality of service is guaranteed to all users. In order to clearly investigate the contention among different flows and their subflows, the network topology graph is converted into flow contention graph. Based on the flow contention graph, each fully meshed subgraphs is distinguished as a clique. A maximal clique is a clique which is not a subgraph of any other clique. Only one subflow can be active in a maximal clique at any time. This constraint is called clique constraint. To accurately reflect the clique constraint of wireless ad hoc networks, the price is generalized in each maximal clique to act as the congestion signal. Also, we have argued that a flow generally spans multiple hops across the network and it is the end-to-end throughput which determines the quality of service perceived by users. Therefore, we allocate the end-to-end rate to each multi-hop flow, rather than splitting it into multiple subflows and consider their rates

separately.

By charging each flow with the maximum clique-based price, we have proved that, through analysis and simulation, the proposed scheme is able to effectively distribute the bandwidth, according to max-min fairness criterion, to end-to-end multi-hop flows. We have also proved the stability of the feedback system. The distributed algorithms adopted are simple and direct, thus owning low computational complexity. In order to accelerate the convergence speed of the algorithm, we also propose a fast converging approach which significantly reduces the convergence time. Finally, the proposed scheme has been extended to support weighted max-min fairness criterion.

Based on the rate allocation scheme, the bandwidth is distributed to the contending flows and target flow rates are calculated, according to a certain objective criterion. After the target rates are calculated in the network layer by the rate allocation schemes, a MAC protocol should support these schemes by allowing each flow to obtain its entitled bandwidth through medium access control. The third main contribution of this thesis proposes a MAC protocol for bandwidth allocation [45][46]. The objective of this MAC protocol is, given the rates calculated by a rate allocation scheme at the network layer, to guarantee the allocated flow rates. As a result, it is a cross-layer MAC protocol.

A time-slotted framework is firstly introduced. In each maximal clique, every N time slots are defined as a transmission frame. Each transmission frame is regarded as a unit to estimate the share of the bandwidth within the corresponding maximal clique. In each node, packets from different subflows are queued in different buffers. All the nodes maintain a clique occupancy table (COT) for each maximal clique to which they are related. The content of the COT describes the occupancy result within the corresponding maximal clique. When a subflow $f_{i,j}$ successfully occupies a free time slot in a frame, all the nodes in the same maximal clique will update their COTs to mark the time slot with $f_{i,j}$. In subsequent frames, each subflow sends packets at the same time slots that it has already occupied, i.e., marked with

its ID in COTs.

Based on this framework, two different MAC algorithms are implemented in this protocol. Greedy Self-Contention (GSC) algorithm is a contention-based algorithm. At the beginning of a free time slot, each subflow transmits a packet with probability $p_{i,j} = x_{i,j}/c_l$ as the persistence probability to contend for the free time slot. In order not to affect the convergence time and to have a simple implementation approach, GSC algorithm does not adopt any backoff mechanism. GSC could support arbitrary rate allocation schemes but has a slower convergence speed due to the contention-based nature. In order to accelerate the convergence speed, the second algorithm, called Cooperative Token Forwarding (CTF) algorithm, is specifically designed for the category of quasi-synchronous rate allocation schemes. Many recently proposed rate allocation schemes, such as the one we have proposed in Chapter 3, have an inherent characteristic that the local flow contention graph and the local contention information are available among all nodes in the same maximal clique. In other words, nodes in the maximal clique could be regarded as quasi-synchronized with the same flow contention information. CTF takes advantage of this inherent characteristic, and thus eliminates the contention among nodes in a maximal clique. In CTF, each node is assigned with a priority determined by its MAC address and degree. A logical symbol called token, with which a node could transmit data packets, is used to control the media access and thus allocate the bandwidth. Because CTF eliminates the contentions, its convergence performance is much better. Both of the MAC algorithms are distributed and easy to implement.

In summary, our work has demonstrated that an effective and efficient distributed resource management framework spanning over the network and the MAC layers is practical and useful for wireless ad hoc networks.

5.2 Future Research Work

RLAR protocol proposed in this thesis provides direction for considering physical link reliability when dealing with the routing problem in MANETs, which is usually ignored in previous works. However, this work is just the first step towards a comprehensive solution for reliable routing in MANETs and many open issues need to be addressed in the future. The first open problem is the estimation precision of the first module. In this work, we adopt the simple shadowing model as the propagation model in physical layer and assume the channel is perfect, i.e., no other channel impairments and collisions are involved. Based on this simplification, we use received signal strength indication (RSSI) and compass direction to estimate the link availability. We notice that the shadowing model over-simplifies the actual physical layer characteristics. New approaches of estimating the link reliability based on more accurate physical layer models are being studied.

Another open problem is that a robust routing metric needs to consider not only the physical layer aspect but also the MAC layer aspect. In this work, in order to highlight the importance of considering the physical link reliability, we directly use the link reliability as the routing metric. However, as we have indicated earlier, a good routing metric should also reflect the MAC layer conditions, for example, data rate, interference and congestion. We have already done some preliminary work in developing a better model to capture the MAC layer interference in wireless mesh networks [47]. Our next step is to combine the physical layer and the MAC layer aspects together, and thus developing a comprehensive solution to the robust routing problem.

The future work about rate allocation schemes in the network layer should focus on low computational complexity and fast convergence speed. Both of the two features could simplify the implementation of such schemes and benefit their performance, as well as broadening the application scope. This research combines theories from economics and mathematics to design new schemes and algorithms. Such theories

may involve game theory, optimization theory, control theory and graph theory.

When further designing the QoS-aware MAC protocols for bandwidth allocation in wireless ad hoc networks, the main concern also relies on implementation complexity and convergence performance. When contention-based MAC algorithms are being considered, better contention resolution techniques need to be designed. In contrary, the cooperative algorithms are free of contention, thus they need not to deal with contention resolution. The main design focus of such algorithms is how to efficiently negotiate and coordinate transmissions among flows in the neighborhood. The management of local information exchange also needs to be taken into consideration.

Appendix A

Proof of Proposition 1: Stability of the Price Feedback System

Our proposed price-based rate allocation scheme is a kind of negative feedback system. Equation (3.6) indicates that the increment of price Δp_Ω is proportional to the difference of aggregate traffic time fraction and normalized capacity bound $(\sum_{l \in C_\Omega} \sum_{s \in S(l)} \frac{x_s(t)}{c_l} - b_\Omega)$. According to Equation (3.7) and the property of demand function $D(\cdot)$ as a continuous, positive and decreasing one, the source nodes will adjust the flow rates in an opposite direction to the prices generated in the controlling cliques, i.e., the larger the prices are, the smaller the flow rates will be.

Generally speaking, the stability of the closed-loop control system may be determined directly by computing the poles of the closed-loop transfer function. If all the poles are on the left side of the S -plane, in other words, if all the eigenroots have negative real parts, the system is stable and will converge to the equilibrium. [37] provides an alternative way of judging the stability of a closed-loop system by calculating its open-loop transfer function. The stability proof in wireline networks has been done in [30] and for the convenience of interested readers, we provide the similar counterpart in wireless networks for our proposed system as follows.

Consider a standard unity feedback loop, with the open-loop transfer function $L(s) = \frac{\gamma}{s} F(s)$. The closed loop is stable for all $\gamma \in (0, 1]$, if the following three conditions are satisfied:

(1) $F(s)$ is analytic in $Re(s) > 0$, and $\|F(s)\| \leq \beta$ in $Re(s) \leq 0$.

(2) $F(0)$ has strictly positive eigenvalues.

(3) For all $\gamma \in (0, 1]$, -1 is not an eigenvalue of $L(j\omega)$, $\omega \neq 0$.

It is also shown in [37] that condition (1) is automatically satisfied. Here we will prove that the proposed price-based scheme satisfies the remaining two conditions.

The network is modeled in the Laplace domain as:

$$y(s) = R_f(s)x(s) \tag{A.1}$$

$$q(s) = R_b(s)^T p(s) \tag{A.2}$$

R_f and R_b represent the delayed forward and backward routing matrices, defined by

$$[R_f(s)]_{\Omega,i} = \begin{cases} e^{-\tau_{i,\Omega}^f s} & , \text{ if source } i \text{ uses clique } \Omega \\ 0 & , \text{ otherwise} \end{cases} \tag{A.3}$$

$$[R_b(s)]_{\Omega,i} = \begin{cases} e^{-\tau_{i,\Omega}^b s} & , \text{ if source } i \text{ is controlled by clique } \Omega \\ 0 & , \text{ otherwise} \end{cases} \tag{A.4}$$

Following the small signal model in [37], we consider a small perturbation around equilibrium; $x = x_0 + \delta x$, $y = y_0 + \delta y$, $p = p_0 + \delta p$, $q = q_0 + \delta q$. If the perturbation is small enough, it will not change the set of controlling cliques. In this case, by eliminating the rows corresponding to non-controlling cliques, the model can be reduced to

$$\delta y(s) = R_f(s)\delta x(s) \tag{A.5}$$

$$\delta q(s) = R_b(s)^T \delta p(s) \tag{A.6}$$

Since for each source, there is only one controlling clique, the static gain of the source is therefore

$$\kappa_i = \frac{\alpha_i x_{0i}}{\tau_i} \quad (\text{A.7})$$

The overall multi-variable feedback loop is then described as follows with open loop transfer function

$$L(s) = R_f(s) \mathcal{K} R_b^T(s) \mathcal{C} \frac{1}{s}, \quad (\text{A.8})$$

where $\mathcal{K} = \text{diag}(\kappa_i)$, $\mathcal{C} = \text{diag}(\frac{1}{b_\Omega})$.

For each source, it is only controlled by one controlling clique whose price is maximal among all the cliques the source flow traverses, while each controlling clique Ω may control a set of sources m_Ω . Therefore $R_f(s)$ is block lower triangular with the Ω th diagonal block, $D_{\Omega\Omega}$, having size $1 \times |m_\Omega|$, where $|m_\Omega|$ is the cardinality of m_Ω . Since the only nonzero element of column i of $R_b(s)$ is in the n_i th row, $R_b(s)$ is a block diagonal with the Ω th diagonal block having size $(1 \times |m_\Omega|)$. Because both of \mathcal{K} and \mathcal{C} are diagonal, $\mathcal{K} R_b(s)^T \mathcal{C}$ is also block diagonal and the size of its Ω th diagonal block is $(|m_\Omega| \times 1)$. Here we use $B_{\Omega\Omega}^T$ to represent the Ω th diagonal block of $\mathcal{K} R_b(s)^T \mathcal{C}$. Thus $F(s) = R_f(s) \mathcal{K} R_b(s)^T \mathcal{C}$ is block lower triangular, having the Ω th diagonal block as

$$[F(s)]_{\Omega,\Omega} = \frac{1}{b_\Omega} \sum_{k \in m_\Omega} \frac{\alpha_k x_{0k}}{\tau_k} e^{-\tau_k s} \quad (\text{A.9})$$

Therefore the eigenvalues of $F(0)$ are the diagonal elements of the triangular matrix as

$$\text{eig}(F(0)) = [F(0)]_{\Omega,\Omega} = \frac{1}{b_\Omega} \sum_{k \in m_\Omega} \frac{\alpha_k x_{0k}}{\tau_k}, \quad (\text{A.10})$$

which are nonempty sums of strictly positive real numbers. Thus condition 2 is satisfied.

In addition, because $F(s)$ is lower triangular, $L(s)$ is also lower triangular with eigenvalues

$$[L(s)]_{\Omega,\Omega} = \frac{\gamma}{b_{\Omega}} \sum_{k \in m_{\Omega}} \frac{\alpha_k x_{0k}}{\tau_k s} e^{-\tau_k s} \quad (\text{A.11})$$

It is already known that if $0 < \alpha_i < 1$ then $\text{Re}\left[\frac{\alpha_i e^{-j\tau_i \omega}}{j\tau_i \omega}\right] > -1$ for $\omega \neq 0$. Thus, as their positive weighted sum, $\text{Re}[\text{eig}(L(j\omega))] > -1$. Condition 3 is satisfied.

In summary, since all the three conditions are satisfied, the proposed clique-based price feedback system is stable.

Appendix B

Proof of Proposition 2: Convergence of Max-min Fairness

When allocating the bandwidth, we consider all the flow rates from the end-to-end perspective, rather than dividing each flow into multiple subflows and considering the rates of subflows. Let array X contain all the source flow rates arranged into subsets which contain flows of equal rates. Subsets are arranged in X in an ascending order. X is arranged as follows:

$$X = \{(x_{0a}, x_{0b}, \dots), (x_{1a}, x_{1b}, \dots), \dots, (x_{Na}, x_{Nb}, \dots)\} \quad (\text{B.1})$$

Let x_z be the rate $x_z = x_{za} = x_{zb} = \dots$, we can see that $x_0 < x_1 < \dots < x_N$. We assume that all sources have the same demand function $x_s = D(q_s)$, where x_s is the transmission rate of source s and q_s is the price assigned to this source. The function $D(q_s)$ for $q_s \geq 0$ is assumed to be continuous, positive and decreasing. Let q_z be the price $q_z = D^{-1}(x_z)$. We get $q_0 > q_1 > \dots > q_N$. Let W_z denotes the set of maximal cliques with price $p_z = q_z$. With all the assumptions and definitions above, we now demonstrate the feasibility of the proposed scheme in wireless ad hoc networks through mathematical induction as follows:

a) Let us first consider the maximal cliques in W_0 . All the maximal clique(s) in W_0 have the price of p_0 . Given that the price update algorithm in (3.6) is employed at

every maximal clique, we can see that in the maximal clique C_r with price p_0 :

$$p_0(t+1) = [p_0(t) + \gamma(\sum_{l \in C_r} \sum_{s \in S(l)} \frac{x_s(t)}{c_l} - 1)]^+ \quad (\text{B.2})$$

Since p_0 is the maximum price in the network, all the flows through this maximal clique will be assigned the price $q = p_0$. With the assumption of same utility function, it leads to all the flows in C_r having a equal rate $x = D(p_0)$. In steady state, $p_0(t+1) = p_0(t)$, thus:

$$x_s(t) = x_0(t) = \frac{1}{\sum_{l \in C_r} \sum_{s \in S(l)} \frac{1}{c_l}} \quad (\text{B.3})$$

Note that the flows with the price p_0 share the clique bandwidth equally. They are feasible and no rate can be increased without decreasing another rate that has the maximum price p_0 . So for these flows with price p_0 , they are max-min fair.

b) The W_1 set of cliques are the maximal cliques with the second highest prices p_1 . All flows through maximal clique(s) in W_1 are either controlled by cliques with price p_0 elsewhere, or the cliques with price p_1 , which belong to W_1 . We can see the price update rule for cliques with price p_1 is:

$$p_1(t+1) = \{p_1(t) + \gamma[\sum_{l \in C_r} (\sum_{\substack{s \in S(l) \\ q_s = p_0}} \frac{x_s(t)}{c_l} + \sum_{\substack{s \in S(l) \\ q_s = p_1}} \frac{x_s(t)}{c_l}) - 1]\}^+ \quad (\text{B.4})$$

In steady state:

$$x_s(t) = x_1(t) = \frac{1 - \sum_{l \in C_r} \sum_{\substack{s \in S(l) \\ q_s = p_0}} \frac{x_0(t)}{c_l}}{\sum_{l \in C_r} \sum_{\substack{s \in S(l) \\ q_s = p_1}} \frac{1}{c_l}} \quad (\text{B.5})$$

Note that those flows with price p_0 , if exist, are not controlled by this maximal clique, and we have already shown that their rates are maximized in (B.3). All other

flows in this clique will be assigned a price of p_1 and equally share the remaining bandwidth by (B.5), which is also a max-min fair allocation.

c) Let us assume that resource allocation to flows with prices $p_{k-1}, p_{k-2}, \dots, p_0$ is max-min fair. Now we consider the flows with prices p_k . In cliques with price p_k , there may be flows which are controlled by other cliques with prices $p_{k-1}, p_{k-2}, \dots, p_0$, and a set of flows which are controlled by themselves with the price p_k . We follow the same way of (B.4) and generate the corresponding form of (B.5) in maximal cliques with price p_k as:

$$x_s(t) = x_k(t) = \frac{1 - \sum_{\substack{l \in C_{r,s} \in S(l) \\ q_s = p_0}} \sum_{c_l} \frac{x_0(t)}{c_l} - \dots - \sum_{\substack{l \in C_{r,s} \in S(l) \\ q_s = p_{k-1}}} \sum_{c_l} \frac{x_{k-1}(t)}{c_l}}{\sum_{\substack{l \in C_{r,s} \in S(l) \\ q_s = p_k}} \sum_{c_l} \frac{1}{c_l}} \quad (\text{B.6})$$

Since flows with prices larger than p_k are not controlled by this clique and we have assumed that they already have max-min allocation at other cliques. The remaining capacity is allocated equally to the flows with price p_k in this maximal clique, thus satisfying the max-min condition for such flows. Therefore the max-min condition is satisfied by all flows in the whole network and global fairness is achieved.

Appendix C

Proof of Proposition 3: Coverage Area of Overhearing

The transmission in our discussion refers to the bi-directional transmission including data packets and ACKs. If any of these packets are lost due to interference with transmissions of other nodes in the neighborhood, the transmission is regarded as failed. Let us assume that neither of the transmission pair can overhear the ongoing transmission within the maximal clique. It means that both of them are out of the transmission range of the ongoing active subflow. So they can perform the communication without causing interference in the maximal clique. This contradicts with the fact that no two subflows in a maximal clique can be active at the same time. Hence, at least one node of a transmission pair could overhear the active subflow in the maximal clique.

References

- [1] R. Ramanathan and J. Redi, “A Brief Overview of Ad Hoc Networks: Challenges and Directions,” *IEEE Communication Magazine*, pp. 20–22, May 2002.
- [2] J. Freebersyser and B. Leiner, “A DoD Perspective on Mobile Ad Hoc Networks,” *Ad Hoc Networking*, ed. C. E. Perkins, Addison-Wesley, pp. 29–51, 2001.
- [3] N. Abramson, “The Aloha System - Another Alternative for Computer Communications,” *Proceedings of Fall Joint Computer Conference, AFIPS Conference*, 1970.
- [4] IEEE 802.11 WG, *Draft Supplement to STANDARD FOR Telecommunications and Information Exchange Between Systems - LAN/MAN Specific Requirements - Part 11: Wireless Medium Access Control (MAC) and physical layer (PHY) specifications: Medium Access Control (MAC) Enhancements for Quality of Service (QoS)*, IEEE 802.11e/D2.0, Nov. 2001.
- [5] M. S. Corson, J. P. Macker and G. H. Cirincione, “Internet-based Mobile Ad hoc Networking,” *IEEE Internet Computing*, vol. 3, issue 4, pp. 63–70, 1999
- [6] R. Draves, J. Padhye, and B. Zill, “Comparisons of Routing Metrics for Static Multi-Hop Wireless Networks,” *Proceedings of SigComm 2004*, pp. 133–144, 2004
- [7] S. Gupta, V. Shankar, and S. Lalwani, “Reliable Multicast MAC Protocol for Wireless LANs,” *Proceedings of ICC 2003*, pp. 93–97, 2003

- [8] W. Si and C. Li, "RMAC: A Reliable Multicast MAC Protocol for Wireless Ad Hoc Networks," *Proceedings of ICPP 2004*, pp. 494–501, 2004
- [9] R. Chandra, V. Ramasubramanian and K. Birman, "Anonymous Gossip: Improving Multicast Reliability in Mobile Ad-hoc Networks," *Proceedings of ICDCS 2001*, pp. 275–283, 2001
- [10] M. Jiang and W. Liao, "Family ACK Tree (FAT): A New Reliable Multicast Protocol for Mobile Ad Hoc Networks," *Proceedings of ICC 2002*, vol. 5, pp. 3393–3397, 2002
- [11] S. Sajama and Z.J. Hass, "Independent-tree Ad Hoc Multicast Routing (ITAMAR)," *Proceedings of VTC 2001*, vol. 2, pp. 600–604, 2001
- [12] G. H. Lynn and T. F. Znati, "RoMR: A Robust Multicast Routing Protocol for Ad-Hoc Networks," *Proceedings of IEEE Conference on Local Computer Networks*, pp. 260–268, 2001.
- [13] T. Gopalsamy, M. Singhal, D. Panda and P. Sadayappan, "A Reliable Multicast Algorithm for Mobile Ad hoc Networks," *Proceedings of ICDCS 2002*, pp. 563–571, 2002
- [14] E. M. Royer and C. E. Perkins, "Multicast Operation of the Ad-hoc On-Demand Distance Vector Routing Protocol," *Proceedings of MobiCom 1999*, pp. 207–218, 1999
- [15] J. Yli-Hietanen, K. Koppinen, J. Astola, "Low-complexity Angle of Arrival Estimation of Wideband signals using Small Arrays," *Proceedings of the 8th IEEE Signal Processing Workshop on Statistical Signal and Array Signal Processing*, pp. 109–112, 1996
- [16] J. Yli-Hietanen, K. Koppinen, J. Astola, "Time-Delay Selection for Robust Angle of Arrival Estimation," *SIP 1999*, pp. 81–83, 1998

- [17] R. Dube, C. Raia, K-Y Wang and S. Tripathi, "Signal Stability Based Adaptive Routing (SSA) for Ad Hoc Networks," *IEEE Personal Communications*, pp. 36–45, 1997
- [18] A. B. McDonald and T. F. Znabi, "A Path Availability Model for Wireless Ad hoc Networks," *Proceedings of WCNC 1999*, pp. 35–40, 1999.
- [19] S. Jiang, D. He and J. Rao, "A Prediction-based Link Availability Estimation for Mobile Ad Hoc Networks," *Proceedings of INFOCOM 2001*, vol. 3, pp. 22–26, 2001.
- [20] T. S. Rappaport, "Wireless Communications, Principles and Practice," Prentice Hall, 1996.
- [21] D. De Couto, D. Aguayo, J. Bicket and R. Morris, "High-throughput Path Metric for Multi-hop Wireless Routing," *Proceedings of MobiCom 2003*, 2003.
- [22] X. L. Huang and B. Bensaou, "On Max-min Fairness and Scheduling in Wireless Ad-hoc Networks: Analytical Framework and Implementation," *Proceedings of ACM Mobihoc 2001*, 2001.
- [23] Y. Xue, B. Li, and K. Nahrstedt, "Price-based Resource Allocation in Wireless Ad Hoc Networks," *Proceedings of Int. Workshop on Quality of Service (IWQoS)*, pp. 79–96, Jun. 2003.
- [24] L. Chen, S. H. Low, J. C. Doyle, "Joint Congestion Control and Media Access Control Design for Ad Hoc Wireless Networks," *Proceedings of INFOCOM 2005*, vol. 3, pp. 2212–2222, March 2005
- [25] B. Li, "End-to-End Fair Bandwidth Allocation in Multi-Hop Wireless Ad Hoc Networks," *Proceedings of ICDCS 2005*, pp. 471–480, 2005.
- [26] F. P. Kelly, "Charging and Rate Control for Elastic Traffic," *European Transactions on Telecommunication*, vol. 8, pp. 33–37, 1997.

- [27] S. H. Low and D. E. Lapsley, "Optimization Flow Control: Basic Algorithm and Convergence," *IEEE/ACM Trans. on Networking*, vol. 7, no. 6, pp. 861–874, 1999.
- [28] J. Mo and J. Walrand, "Fair End-to-end Window-based Congestion Control," *IEEE/ACM Trans. on Networking*, vol. 8, pp. 556–567, Oct. 2000.
- [29] B. Wydrowski and M. Zukerman, "MaxNet: A Congestion Control Architecture for Maxmin Fairness," *IEEE Commun. Lett.*, vol. 6, pp. 512–514, Nov. 2002.
- [30] B. Wydrowski, L. L. H. Andrew, and M. Zukerman, "MaxNet: A Congestion Control Architecture for Scalable Networks," *IEEE Commun. Lett.*, vol. 7, pp. 511–513, Oct. 2003.
- [31] T. Nandagopal, T. E. Kim, X. Gao and V. Bharghavan, "Achieving MAC Layer fairness in wireless packet networks," *Proceedings of MobiCom 2000*, 2000.
- [32] L. Tassiulas and S. Sarkar, "Maxmin Fair Scheduling in Wireless Networks," *Proceedings of INFOCOM 2002*, vol. 2, pp. 763–772, 2002.
- [33] S. Sarkar and K. Sivarajan, "Fairness in Wireless Mobile Networks," *IEEE Trans. on Information Theory*, vol. 48, no. 8, pp. 2412–2426, 2002.
- [34] S. Sarkar and L. Tassiulas, "End-to-end Bandwidth Guarantees Through Fair Local Spectrum Share in Wireless Ad-hoc Networks," *IEEE Trans. on Automatic Control*, vol. 50, no. 9, pp. 1246–1259, September, 2005.
- [35] L. Tan, X. Zhang, L. L. H. Andrew, S. Chan and M. Zukerman, "Price-based Max-min Fair Rate Allocation in Wireless Multi-hop Networks," *IEEE Commun. Lett.*, vol. 10, issue 1, pp. 31–33, 2006
- [36] X. Su and S. Chan, "Max-min Fair Rate Allocation in Multi-hop Wireless Ad Hoc Networks," *Proceedings of The Third IEEE International Conference on Mobile Ad-hoc and Sensor Systems*, pp. 513–516, 2006.

- [37] F. Paganini, J. C. Doyle, and S. H. Low, “Scalable Laws for Stable Network Congestion Control,” *Proceedings of IEEE CDC*, pp. 185–190, 2001.
- [38] X. Lin and N. B. Shroff, “Joint Rate Control and Scheduling in Multihop Wireless Networks,” *Proceedings of IEEE CDC*, pp. 1484–1489, 2004.
- [39] M. J. Neely, E. Modiano and C. Li, “Fairness and Optimal Stochastic Control for Heterogeneous Networks,” *Proceedings of IEEE INFOCOM*, pp. 1723–1734, 2005.
- [40] A. Eryilmaz and R. Srikant, “Joint Congestion Control, Routing and MAC for Stability and Fairness in Wireless Networks,” *Proceedings of International Zurich Seminar on Communications*, 2006.
- [41] L. X. Bui, A. Eryilmaz, R. Srikant, and X. Wu, “Joint Asynchronous Congestion Control and Distributed Scheduling for Multi-Hop Wireless Networks,” *Proceedings of IEEE INFOCOM*, 2006.
- [42] R. Diestel, “Graph Theory”, Springer-verlag, 1997.
- [43] C. E. Shannon, “A theorem on coloring the lines of a network,” *Journal of Mathematical Physics*, vol. 28, pp. 148–151, 1949.
- [44] X. Su, S. Chan and K. S. Chan, “RLAR: Robust Link Availability Routing Protocol for Mobile Ad Hoc Networks,” *Proceedings of the IEEE International Conference on Communications (ICC), 2007*.
- [45] X. Su and S. Chan, “Cross-layer MAC Design for Bandwidth Allocation in Wireless Ad Hoc Networks,” *submitted to submitted to IEEE ICC 2008*.
- [46] X. Su and S. Chan, “Distributed Max-min Fair Rate Allocation and Scheduling in Multi-hop Wireless Ad Hoc Networks,” *submitted to IEEE Transactions on Mobile Computing*.
- [47] X. Su and S. Chan, “Co-channel Interference Evaluation in Multi-channel Multi-hop Wireless Mesh Networks,” *submitted to IET Electronics Letters*.