國立臺灣大學資訊管理研究所

碩士論文

Graduate Institute of Information Management

National Taiwan University

Master Thesis

多信道無線網狀網路下近似最佳化之分散式具服務品質限制路由演算法

A Near-Optimal Distributed QoS Constrained Routing Algorithm for Multichannel Wireless Mesh Networks

謝友仁

Yu-Jen Hsieh

指導教授：林永松，顏宏旭 博士

Advisor: Frank Yeong-Sung Lin, Hong-Hsu Yen, Ph.D.

中華民國九十八年七月

July, 2009

多信道無線網狀網路下近似最佳化之分散式具服務品質限制路由演算法

A Near-Optimal Distributed QoS Constrained Routing Algorithm for Multichannel Wireless Mesh Networks

本 論 文 係 提 交 國 立 台 灣 大 學
資 訊 管 理 學 研 究 所 作 為 完 成 碩 士
學 位 所 需 條 件 之 一 部 份

研究生：謝友仁 撰

中華民國九十八年七月

論文摘要

論文題目：多信道無線網狀網路下近似最佳化之分散式具服務品質限制路由演算法

作者：謝友仁

指導教授：林永松、顏宏旭 博士

無線網狀網路可視為一個用來提供寬頻存取網際網路之「最後一哩」技術的解決之道。於無線網狀網路中採用多個信道來傳輸，已經被證實是一種用來克服容量因干擾而降低之問題的有效方法。對於每個使用者而言，希望選擇低干擾且傳輸延遲最低的路徑去存取網際網路；然而，這對整個系統而言並不是最佳化的結果。

在這篇論文中，我們提供了一個簡單的信道分配演算法，不但容易實施而且使得每個節點能有局部的最大化平行傳輸。我們也提供一個分散式具有服務品質限制的路由演算法，將系統觀點與使用者觀點都納入考量；為了達到這個目標，我們定義一個路由衡量標準，這個路由衡量標準是由鏈結平均傳輸延遲與佇列長度之導數兩者所組成，並且由拉格蘭日鬆弛法為基礎的問題公式所推導出來。

我們使用鏈路狀態路由協定來作動態路由，並且提供多條最短路徑與多條最快路徑給每一個起終點配對，使這個演算法能藉由流量控制啟發式演算法而適用於更多情況之下。最後，我們透過模擬來評估近似最佳化之分散式具服務品質限制路由演算法。模擬結果顯示我們的演算法在較大的網路下於平均端對端傳輸延遲、延遲時間變化與符合服務品質限制之系統吞吐量上優於其他演算法。

關鍵詞：無線網狀網路、分散式路由演算法、鏈結狀態路由協定、服務品質限制路由演算法、拉格蘭日鬆弛法

THESIS ABSTRACT

GRADUATE INSTITUTE OF INFORMATION MANAGEMENT

NATIONAL TAIWAN UNIVERSITY

NAME: YU-JEN HSIEH **JULY 2009**

ADVISOR: YEONG-SUNG LIN, HONG-HSU YEN

A Near-Optimal Distributed QoS Constrained Routing Algorithm for Multichannel Wireless Mesh Networks

Wireless mesh networks (WMNs) are considered as a solution to providing last-mile broadband Internet access. Employing multiple channels into WMN is shown to be an efficient way to conquer the degradation of capacity due to the interferences. For each user, it is desirable to choose the route with low interference and minimum delay to access the Internet; however, this is suboptimal for the whole system.

In this thesis, we propose a simple channel assignment heuristic algorithm which is easy for implementation and makes each node have locally maximal parallel transmission. We also propose a distributed QoS (Quality-of-Service) constrained routing algorithm which takes "system perspective" and "user perspective" into consideration; to achieve the goal, we define a routing metric which is composed of link mean delay and the derivative of queue length, and is derived from a Lagrangean Relaxation based problem formulation.

We use link-state routing protocol for distributed routing and provide both K shortest paths and K fastest paths for each Origin-Destination pair, so that, this algorithm can be suitable for much more scenarios by the admission control heuristic algorithm we proposed. Finally, we evaluate the performance of Near-Optimal Distributed QoS Constrained (NODQC) routing algorithm via simulations. The simulation results show that our routing algorithm outperforms others in terms of average end-to-end delay, delay jitter and system throughput with QoS satisfaction in large-scale networks.

Keywords: Wireless Mesh Networks, Distributed Routing Algorithm, Link-State Routing Protocol, QoS Constrained Routing Algorithm, Lagrangean Relaxation

Table of Contents

List of Tables

List of Figures

Chapter 1 Introduction

1.1 Background

Wireless mesh networks (WMNs) has emerged as an interesting research in recent years. In general, WMNs are composed of mesh access points (MAPs), mesh routers (MRs), and mesh clients (MCs).The MRs do not generate traffic, instead, they are responsible for relaying the packets for each own MCs and play the roles of both hosts and routers [1]. Some MRs called MAPs that act as gateways with wired connection toward the Internet as well as with wireless connection to communicate with other MRs in the WMNs. The MCs referred to as end terminals, and each of which can only interact with one MR to which it is connected [2]. The architecture of WMNs is shown as **Figure 1-1**.

There are many issues and challenges within this network environment, such as channel assignment [3][4][5], routing metric [6][7], link scheduling [2], and resources allocation [8] and so on. There are still some researches about joint some of these independent problems were addressed in [2][9][10]. Although each of the problems can be seen as distinct and independent challenges, there are some associations among them.

For the routing problem, a path can be calculated by the routing algorithm with its own routing metric. However, any two nodes within the transmission range can communicate with each other only if there is a channel assigned to them. Consequently, the channel assignment determines the network topology and the routing paths are also controlled by it. It is obvious that there exists dependence between the routing algorithm and the channel assignment [1].

Therefore, in order to obtain a good routing performance of the WMNs, it is necessary to design a channel assignment algorithm to well determine the network topology before developing the routing algorithm.

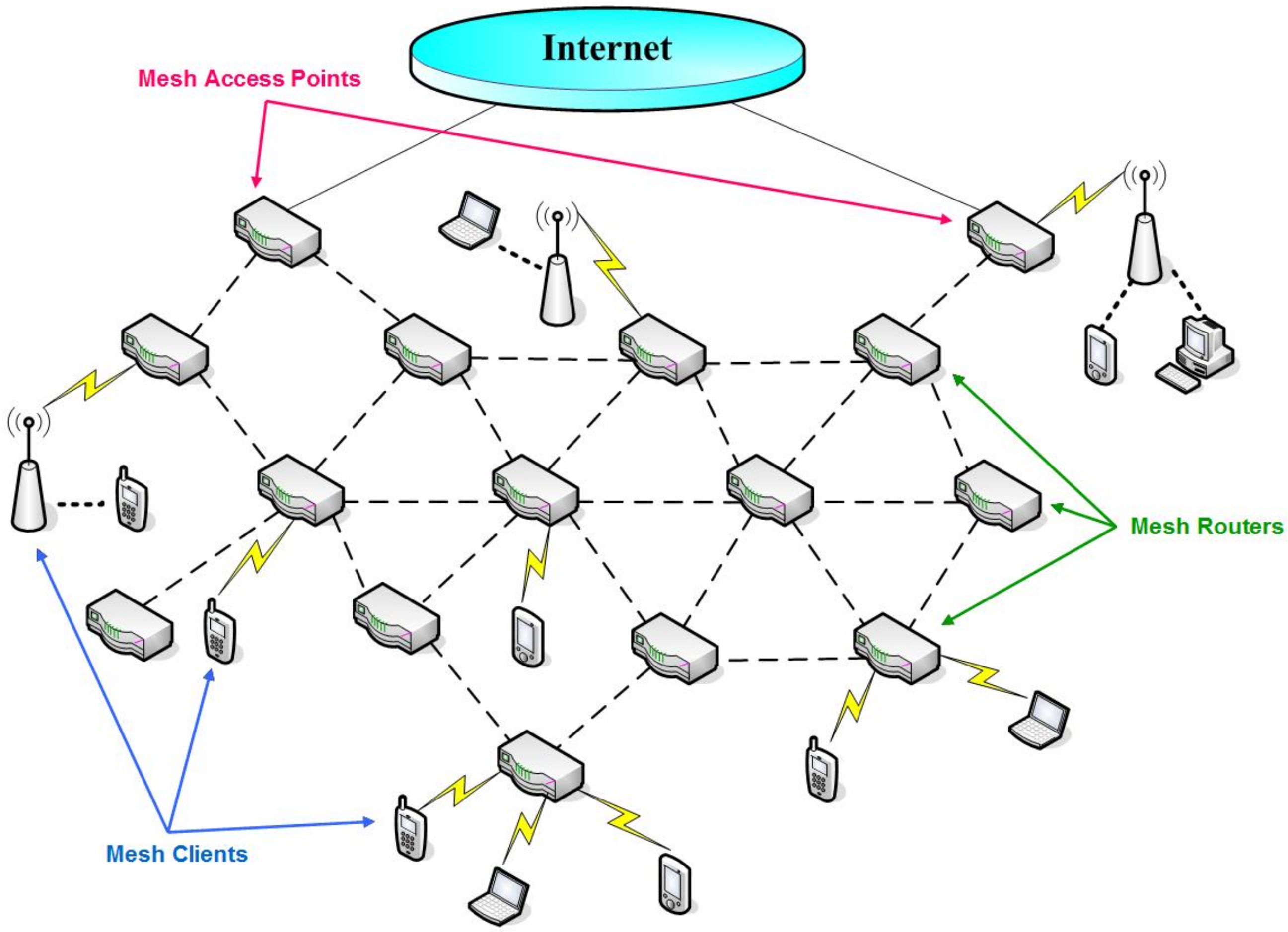

Figure 1-1 Wireless Mesh Networks

1.2 Motivation

The routing problem becomes more complex since the fact that the network topology is determined by the channel assignment [1]. The effectiveness of routing algorithm is dependent on the goodness of channel assignment algorithm. It means that, we need a good enough channel assignment algorithm for the WMNs. In addition to the consideration of the multichannel problem, the more importantly, the assurance of end-to-end Quality-of-Service (QoS) from the “user perspective” should be considered in the routing algorithm.

However, if each user chooses the best route with minimum end-to-end delay in his own way, the so called “optimal path” will be suboptimal due to too many routes on it at the same time. For the system optimality, the congestion of each path is the less the better, even though the path is the fastest one for many users. In other words, the average performance of the whole network is the priority from the “system perspective.”

There are many researches are dedicated to define a new routing metric for the routing algorithm, such as WCETT [6] and BLC [7]. Here, we focus on how the routing metric can account for the average cross-network packet delay and also guarantee the QoS provisioning for each user, so that both the system perspective and the user perspective can be concurrently considered.

1.3 Literature Survey

1.3.1 Multichannel Wireless Mesh Networks

An important goal of designing wireless mesh networks is to avoiding the degradation of link capacity due to the wireless interference. It is evident that employing multiple channels is a common solution for this problem. Essentially, if a node pair can establish communication to each other, both of them must be in the transmission range and using the same channel on their own network interface card (NIC).

By using different and non-overlapping channels, each node can simultaneously communicate with its neighboring nodes (i.e., the nodes within the transmission range) without interferences. However, in single channel wireless networks, it is not allowed to transmitting or receiving data and communicating with two neighbors at the same time because of contention for the shared wireless channel. As noted in [11], using multiple channels instead of single channel in multihop wireless networks has been shown to be able to improve the network throughput dramatically.

The IEEE 802.11b/g standards and IEEE 802.11a standard offer 3 and 12 non-overlapping channels respectively. It has been identified that the fast-channel switching is required by dynamic channel assignment, unfortunately, it is unsuitable for use with commodity hardware. In addition, the channel switching delays itself can be in the order of milliseconds which is an order of magnitude higher than typical packet transmission times (in microseconds). On the other hand, some dynamic channel assignments require specialized MAC protocol or extension of 802.11 MAC layer, and therefore the existing commodity 802.11 hardware will be further unsuitable for use [5].

Therefore, a static channel assignment is desirable to using the multiple channels with commodity hardware efficiently; besides, such channel assignment can be changed infrequently if there are some significant changes in traffic load or the network topology take place [4][5]. Thus, it is so called "quasi-static channel assignment."

1.3.2 Routing Algorithm Issues

1) Performance Measures:

According to [12], there are two main functions performed by a routing algorithm. One is the selection of routes for various origin-destination (O-D) pairs and the delivery of messages to their correct destination once the routes are selected. Another is using a variety of protocols and data structures (i.e., the routing tables). The focus will be on the selection of routes and how it affects network performance. The two main performance measures are throughput and average packet delay and both are affected by the routing algorithm. These performance measures are determined by the interaction of flow control and routing algorithm as shown in **Figure 1-2**.

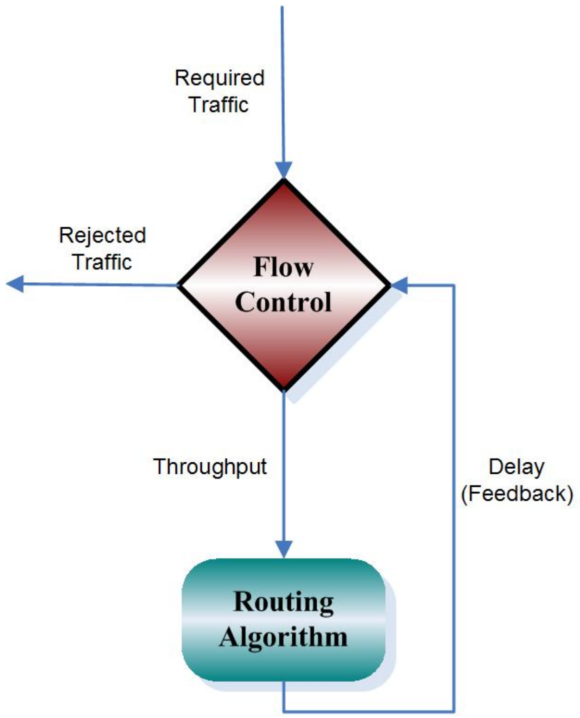

Figure 1-2 Interaction of Flow Control and Routing Algorithm

2) Shortest Path Routing:

If the routing decisions are made only when a new route is being set up and all traffic just follow the previously established path, it is called “session routing”, because a path remains in force for an entire user session [13]. To choose a route for the O-D pair, the routing algorithm finds the shortest path between them. The routing metric (i.e., cost) is used to define the path length and can be measured as number of hops, the mean delay, and even the flow deviation. The mean delay can be used for QoS routing to find the fastest path, and the flow deviation can be applied to system optimization routing where the flow deviation of each route is equal [12].

3) QoS Routing

Quality-of-Service is an internetworking issue that has been discussed more than defined. It can be defined as something a flow seeks to attain including reliability, mean delay, delay jitter, and bandwidth [14].

QoS routing in multihop wireless networks is very challenging due to the wireless interference among different transmission [11]. Even in multichannel wireless networks, the interference still exists; if more than one transmission using the same channel in the interference range, they may interfere each other and make collisions. The bandwidth is used to be a QoS connection request in [11], but for multimedia applications, the delay request is more necessary for QoS requirement.

1.4 Proposed Approach

We propose a simple channel assignment heuristic algorithm based on the principle that locally maximize the parallel transmission for each node. To get the routing metric which represents both the “system perspective” and the “user perspective”, we model the problem as mathematical programming problems and then introduce the Lagrangean Relaxation method to solve them. Thereafter, we take the subgradient method to finding the extreme points to solve the Lagrangean Relaxation problem. Finally, the arc weight form of each link can be derived from the relaxation procedures.

We also propose a distributed routing algorithm based on link-state routing protocol for dynamic routing. The routing metric consisting of the link mean delay and the derivative of queue length is used in this algorithm. To find the shortest path and satisfy the QoS requirement, the routing algorithm constructs K shortest paths as well as K fastest paths in admission control heuristic algorithm.

1.5 Thesis Organization

The remainder of this thesis is organized as follows. A mathematical formulation for the arc weight form used in multichannel wireless mesh networks is shown in Chapter 2. Chapter 3 presents the channel assignment heuristic algorithm and the Lagrangean Relaxation of the problem in Chapter 2. The methods for solving each Lagrangean subproblem and getting primal feasible solutions are also in Chapter 3. Chapter 4 describes how the distributed routing algorithm works and defines the routing metric as well as the routing protocol. The K shortest paths and K fastest paths of the admission control heuristic algorithm are also described in Chapter 4. Chapter 5 is the simulation results and analysis for the performance measures. Chapter 6 includes the summary of this thesis and the suggestions for future works.

Chapter 2 Problem Formulation of Multichannel WMNs

2.1 Problem Description

The problem we addressed is to find out the arc weight form of each link while taking "system perspective" and "user perspective" into account. We use the similar formulation proposed by Yen and Lin [15], which is to minimize the average cross-network packet delay subject to end-to-end delay constraints for users. Further details can be found in the origin reference. This formulation can be extended to use other queue models with monotonically increasing and convex metrics. Thus, we express it as a general form here.

The environment we considered here is a multichannel WMNs which is one case of this generic formulation for illustration purpose. The problem formulation can be applicable to other networks (e.g., virtual circuit networks). The network topology of the WMN is given by the channel assignment heuristic we proposed in this thesis. Once the channel is assigned to the link, it means that any two nodes will use the assigned channel to communicate with each other until the channel assignment is changed (i.e., static channel assignment).

In this thesis, we assume that the buffer of each link is infinity so that there is no loss of packets. Additionally, at the scheduling phase, every link is assumed to be fairly used, namely, the average time of transmitting data over each link is equal. In the multichannel WMNs, the link capacity degrades due to the other links using the same channel in the interference range. Therefore, the link capacity is divided by the number of interference links in the following formulation (i.e., $C_l = \frac{C_l}{I(l)}$) [3].

Note that, the objective of this problem formulation is simply to help us to determine the arc weight form of each link which can be used in the distributed algorithm we focus on this thesis. The summary of problem description is listed as below.

Table 2-1 Problem Description

Problem Assumptions:
1. The channel assignment for each mesh router is fixed for a long period.
2. Each mesh router is stationary.
3. Each mesh router is equipped with multiple 802.11a NICs each of which operates on a particular and non-overlapping channel.
4. Each mesh router can simultaneously communicate with its neighbors in transmission range without interferences by using different channels for each link.
5. A virtual node is added as the destination node to only connect to the mesh access point via wired-line.
6. All flows are transmitted to this virtual node via the mesh access point.
Given:

1. The set of links.
2. The set of mesh routers.
3. The link capacity of each link.
4. The number of interference links of each link.
5. The traffic requirement for each O-D pair.

Objective:

To minimize the average cross-network packet delay of the WMNs.

Subject to:

1. QoS constraints.
2. Path constraints.
3. Capacity constraints.
4. Flow constraints.

To determine:

The arc weight form of each link which takes system perspective and user perspective into consideration for each O-D pair.

2.2 Notation

The notations listed bellow are the given parameters and the decision variables of our formulation shown in **Table 2-2** and **Table 2-3**:

Table 2-2 Notation of Given Parameters

Notation	Description
W	The set of Origin-Destination (O-D) pairs in the WMNs, where $w \in W$.
P_w	The set of directed paths from the origin to the destination of O-D pair w, where $p \in P_w$.
L	The set of communication links in the WMNs, where $l \in L$.
$I(l)$	The number of interference links of link l.
δ_{pl}	The indicator function which is 1 if link l is on path p and 0 otherwise.
C_l	$(packets/s)$ The link capacity of link l.
γ_w	$(packets/s)$ The given traffic input of O-D pair w.
$D_l(g_l)$	The mean delay on link l, which is a monotonically increasing and convex function of aggregate flow g_l.
D_w	The maximum allowable end-to-end QoS for O-D pair w.

Table 2-3 Notation of Decision Variables

Notation	Description
x_p	1 if path p is used to transmit the packets for O-D pair w and 0 otherwise.
y_{wl}	1 if link l is on the path p adopted by O-D pair w and 0 otherwise.
g_l	$(packets/s)$ The estimate of the aggregate flow on link l.

2.3 Problem Formulation

Optimization Problem:

Objective function (IP):

$$\min \sum_{l \in L} D_l(g_l) g_l \qquad \text{(IP)}$$

subject to:

$$\sum_{l \in L} D_l(g_l) y_{wl} \le D_w \qquad \forall w \in W \qquad \text{(IP 1)}$$

$$\sum_{p \in P_w} x_p = 1 \qquad \forall w \in W \qquad \text{(IP 2)}$$

$$\sum_{p \in P_w} x_p \delta_{pl} \le y_{wl} \qquad \forall w \in W, l \in L \qquad \text{(IP 3)}$$

$$0 \le g_l \le \frac{C_l}{I(l)} \qquad \forall l \in L \qquad \text{(IP 4)}$$

$$\sum_{p \in P_w} \sum_{w \in W} x_p \delta_{pl} \gamma_w \le g_l \qquad \forall l \in L \qquad \text{(IP 5)}$$

$$x_p = 0 \text{ or } 1 \qquad \forall p \in P_w, w \in W \qquad \text{(IP 6)}$$

$$y_{wl} = 0 \text{ or } 1 \qquad \forall w \in W, l \in L. \qquad \text{(IP 7)}$$

Explanation of the objective function:

Objective function (IP) is actually the summation of average number of packets on each link, i.e. the queue length, which is obtained by the product of link mean delay and aggregate flow on the link. The expression of the form $D_l(g_l)$ is a monotonically increasing and convex function with respect to g_l, where g_l is the aggregate flow on link l measured in packets per second [12][13][15][16][17].

Note that, the objective function (IP) is to minimize the cross-network packet delay. By Little's Law (i.e., $N = \lambda T$), the objective function is proportional to the average cross-network packet delay. Thus, for a given traffic input (i.e., $\sum_{w \in W} \gamma_w$), minimizing the average number of packets in the network is equivalent to minimizing the average cross-network packet delay [18][19].

Explanation of the constraints:

1) **QoS constraints:**

   Constraint (IP 1) confines that the end-to-end delay should be no larger than maximum allowable end-to-end QoS requirement.

2) **Path constraints:**

   Constraint (IP 2) confines that all the traffic required by each O-D pair are transmitted over exactly one candidate path.

Constraint (IP 3) confines that once path p is selected and link l is on the path, y_{wl} must be equal to 1.

3) Capacity constraints:

Constraint (IP 4) confines the boundaries of aggregate flow on link l.

4) Flow constraints:

Constraint (IP 5) confines the aggregate flow on link l should not exceed the link capacity.

5) Integer constraints:

Constraint (IP 6) and (IP 7) are the integer constraints of decision variables.

Chapter 3 Solution Approach

3.1 Channel Assignment Heuristic

Since the channel assignment in the multichannel WMNs is complicated and therefore is a NP-hard optimization problem, we propose a simple channel assignment heuristic algorithm here to determine the network topology. By the way, this heuristic algorithm is simple such that it can be easily implemented in practical.

The basic idea of this channel assignment heuristic algorithm is to locally avoid using repeating channels and maximize the number of simultaneous communications for each node as possible as we can. We assume the number of NICs of each node is less than or equal to the number of neighbors of each node. This can be regarded as that every node typically communicates with its neighbors by using distinct channels for each one, i.e., each node has only one link to connect each neighbor.

In the WMNs, the MAPs act as gateways toward the Internet with cable connection [1][2]. Since all flows are transmitted to the Internet through the MAPs, we must first assign the channels to the MAP whereby the parallel transmission of the MAP can be as large as possible. At the beginning, we assign different channels to each NIC on the MAP, which has the most number of neighbors. Next, we assign channels to the neighboring nodes of the MAP and start with the neighboring node with the largest channel number on one of its links, and we just assign channels to the links that are not assigned yet. Then, we check the other neighboring nodes of the MAP and also assign channels by means of the previous rule (i.e., start with the node which has the largest channel number on its link), until all neighboring nodes are assigned.

After the 1-hop away nodes were finished, we check the 2-hop away nodes and use the same steps to assign the channels for each one. The channel assignment of all nodes can be done by the "neighbor by neighbor" way. We show the detail procedures in **Table 3-1** and illustrate the heuristic steps from **Figure 3-1** to **Figure 3-7**.

Table 3-1 Channel Assignment Heuristic Algorithm

Step 1: Set the number of NICs to be m_i and the largest channel number n_i for each node i, where m_i is equal to the number of neighbors of node i and n_i is automatically updated whenever a channel is assigned to node i.

Step 2: Choose the MAP b with the largest number m_b among other MAPs, and set n_b to be 0.

Step 3: Assign channels to each link which is not assigned yet of node i. The channel assignment is in increasing order and starts from n_i, i.e., assign channel n_i+1, channel n_i+2….channel $n_i + m_i - 1$ for each link.

Step 3.1: If assigned channel number is more than the number of available channels

Step 3.1.1: Subtract the number of available channels from the channel number.

Step 4: Choose the neighboring nodes in decreasing order of n_i, and assign the channels by Step 3. Until all neighboring nodes are assigned channels.

Step 5: After every neighboring node is finished, check the neighbors of each neighboring node and go to Step 4, until all nodes are assigned channels.

Step 6: Finally, we use a penalty function to check every node. The function removes the links assigned to the same channel of a node and reserves only one of them where the connected node has the least neighbors.

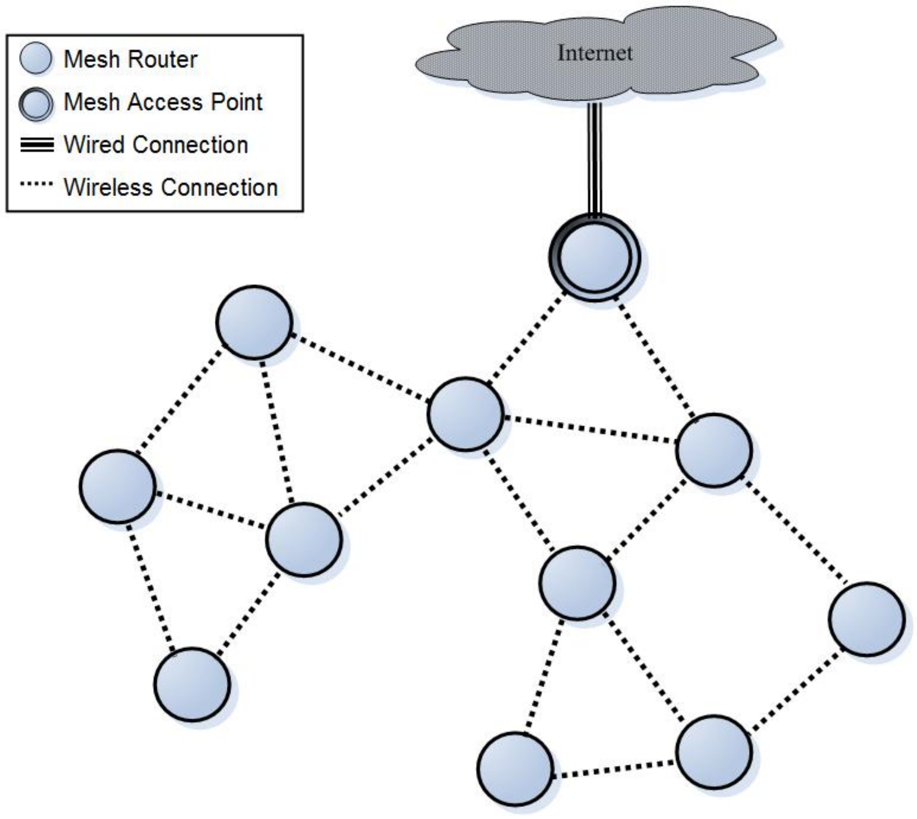

Figure 3-1 Channel Assignment Heuristic Algorithm (Step 1)

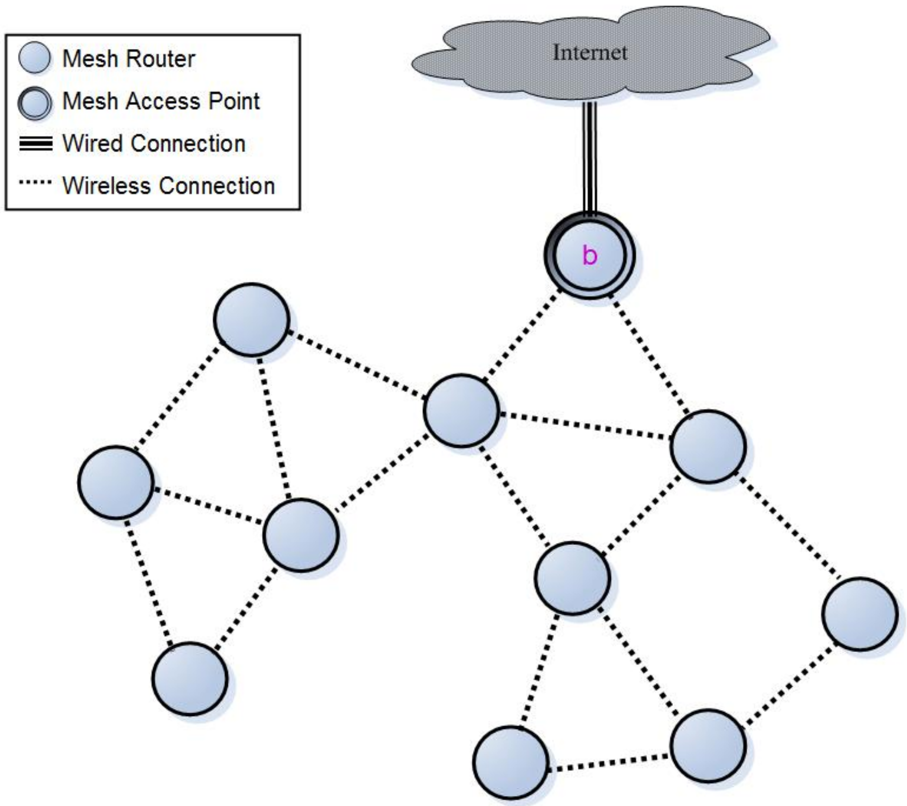

Figure 3-2 Channel Assignment Heuristic Algorithm (Step 2)

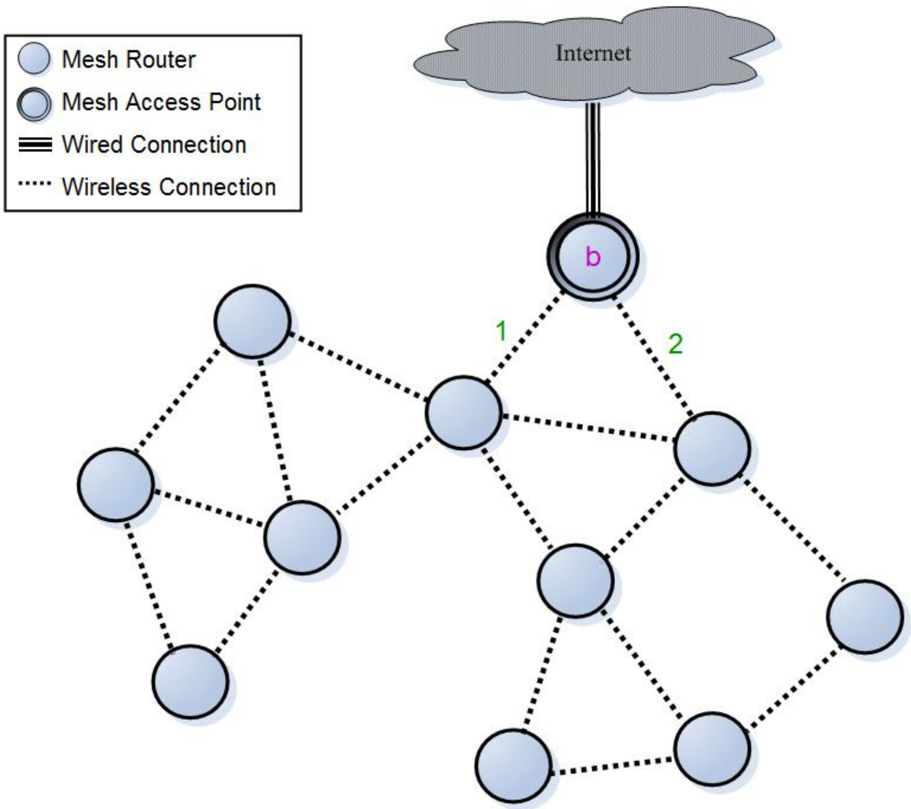

Figure 3-3 Channel Assignment Heuristic Algorithm (Step 3)

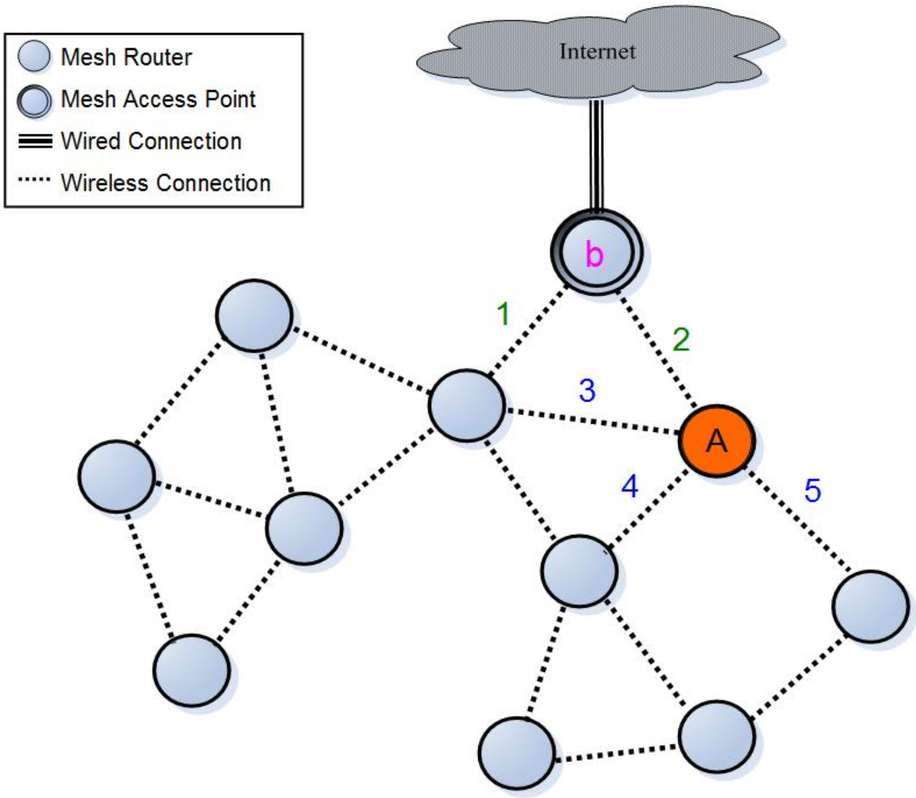

Figure 3-4 Channel Assignment Heuristic Algorithm (Step 4)

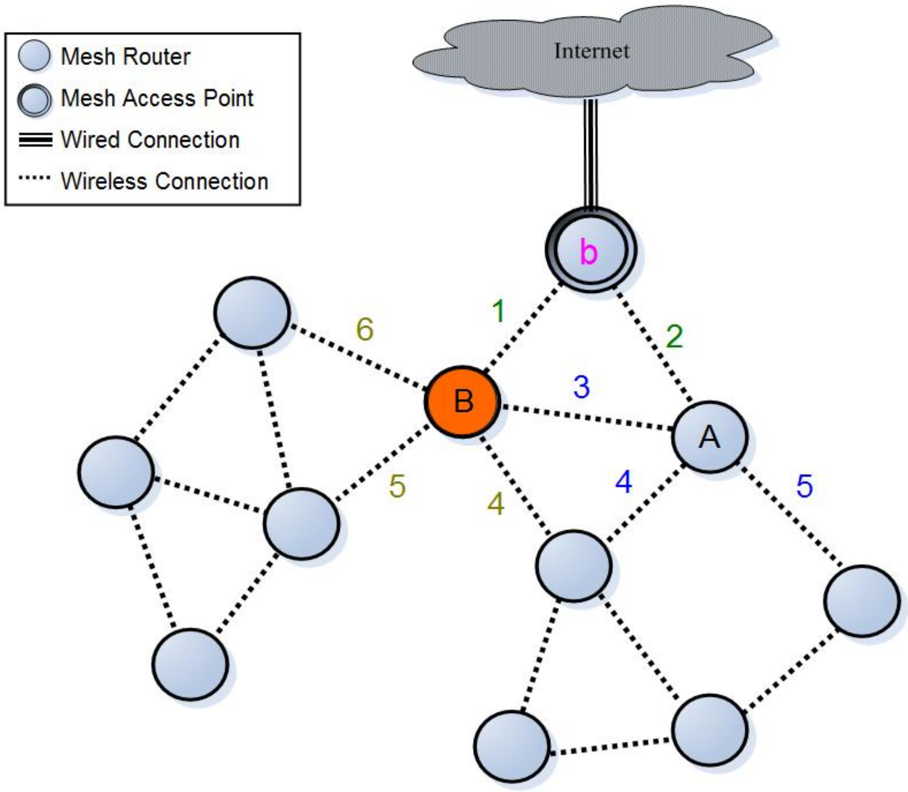

Figure 3-5 Channel Assignment Heuristic Algorithm (Step 5)

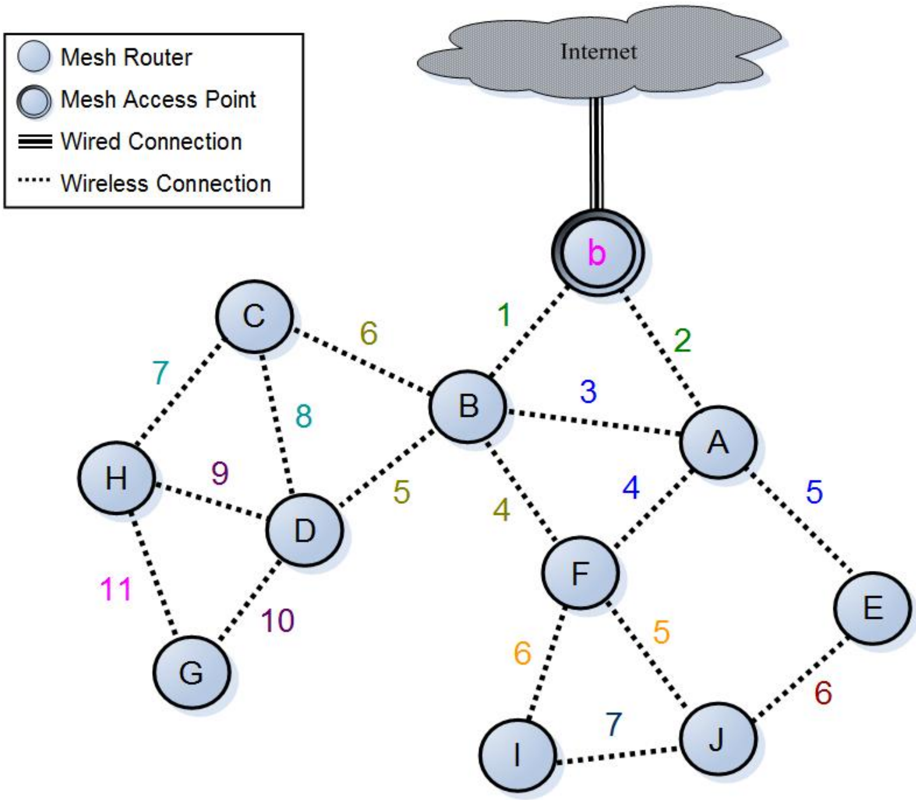

Figure 3-6 Channel Assignment Heuristic Algorithm (Step 6)

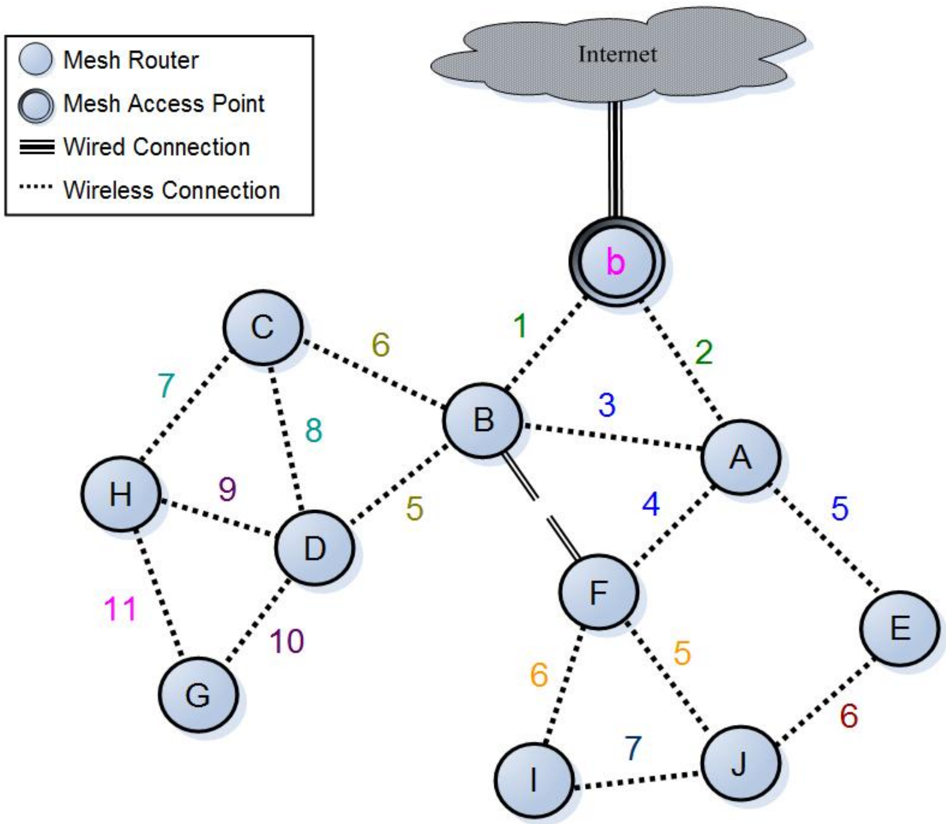

Figure 3-7 Channel Assignment Heuristic Algorithm (Step 7)

3.2 Solution Approach for Multichannel WMNs Formulation

3.2.1 Introduction of Lagrangean Relaxation Method

In the 1970s, the Lagrangean Relaxation method was introduced to solve large-scale mathematical programming problems and there were many researches of it after that [20][21]. It is provides good solutions to those problems and has become a widely used tool for dealing with optimization problems, such as integer programming problems and even nonlinear programming problems.

By adopting Lagrangean Relaxation, a complicated programming problem can be viewed as a small set of relatively easily-solved problems with side constraints. This method reduces the complexity by decomposing the original problem into several independent subproblems with their own constraints, and each of which can be further solved by some well-known algorithms.

The basic idea of Lagrangean Relaxation method and the detail procedures are shown in **Figure 3-8** and **Figure 3-9**, respectively. The Lagrangean Relaxation of the primal problem is developed to be a lower bound of the optimal value for the original minimization problem because some constraints of the original problem are relaxed. Therefore, we can use the boundary to design heuristic algorithms to get the primal feasible solution. To minimize the gap between the primal problem and the Lagrangean Relaxation problem, the subgradient optimization method can be used to derive the tightest lower bound by adjusting the multipliers at each iterations and updating them to improve the results.

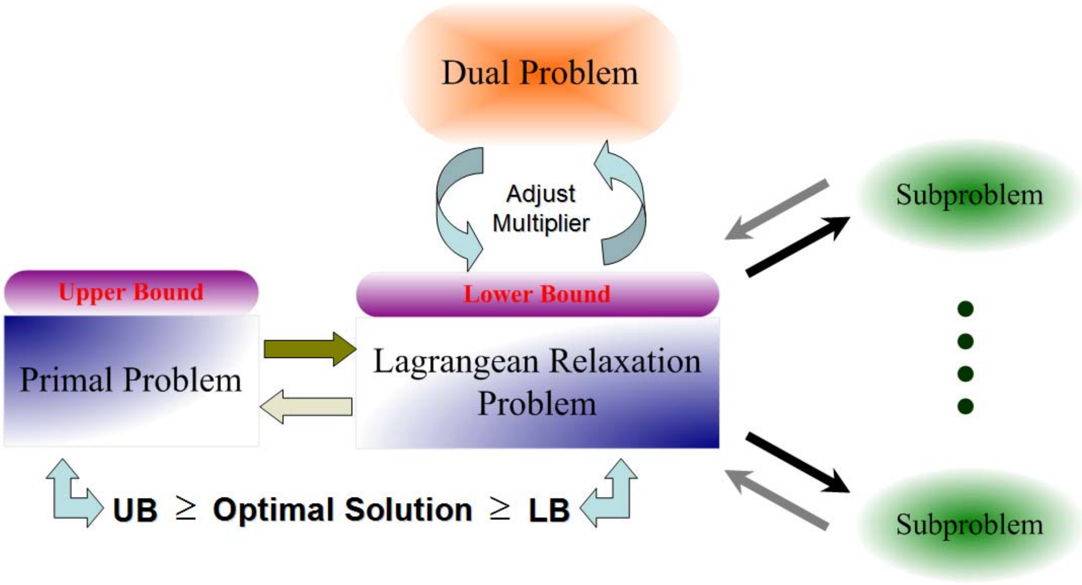

Figure 3-8 Concept of Lagrangean Relaxation Method

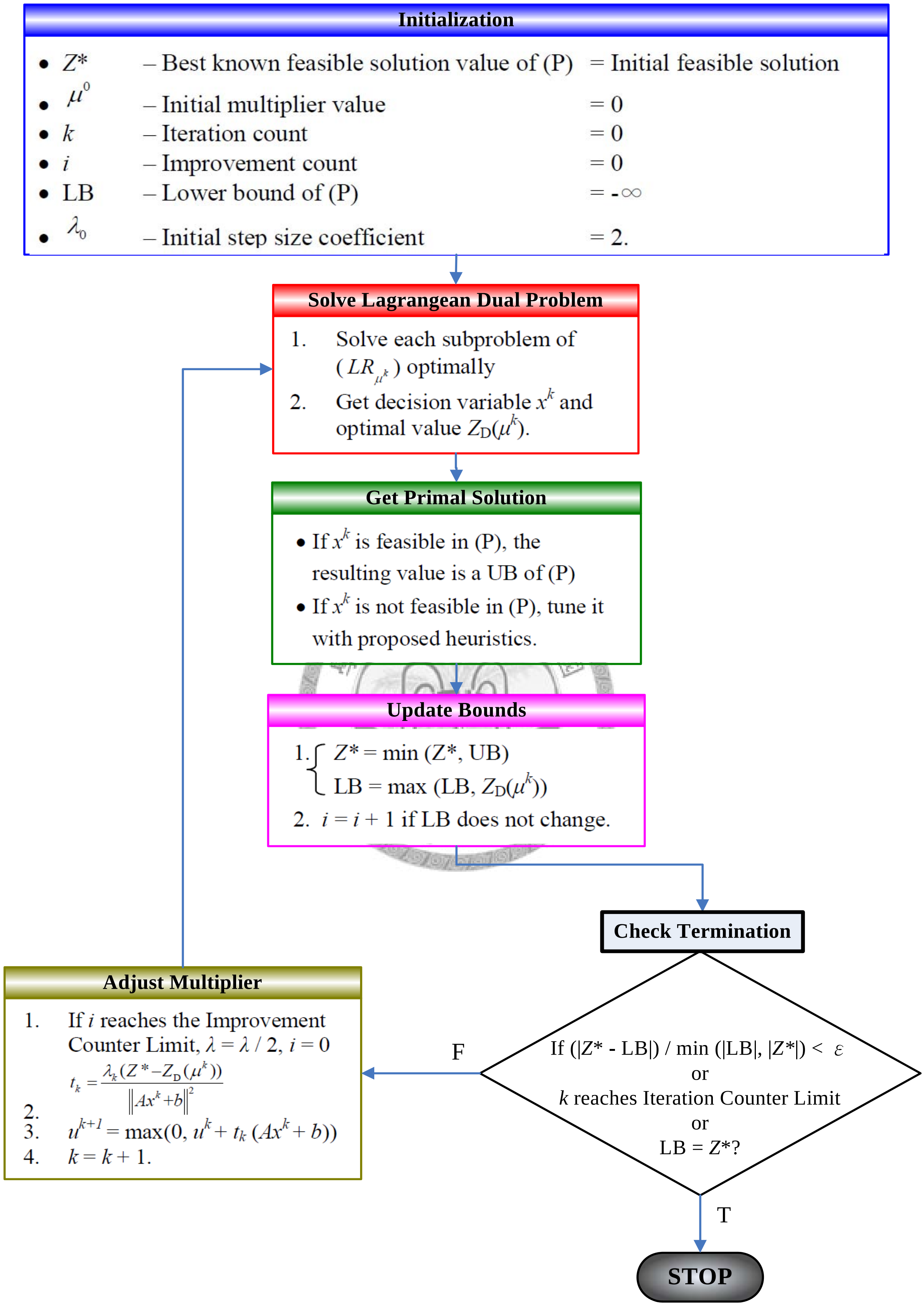

Figure 3-9 Lagrangean Relaxation Procedures

3.2.2 Lagrangean Relaxation

The solution approach of the problem formulation is based on Lagrangean Relaxation. We relax Constraints (IP 1), (IP 3) and (IP 5) and multiply them by nonnegative Lagrangean multipliers, respectively, which add to the objective functions as follows:

Optimization Problem (LR):

$$Z_{LR}(\mu_w^1, \mu_l^2, \mu_{wl}^3) =$$

$$\min \sum_{l \in L} D_l(g_l) g_l$$

$$+ \sum_{w \in W} \mu_w^1 \left(\sum_{l \in L} D_l(g_l) y_{wl} - D_w\right)$$

$$+ \sum_{l \in L} \mu_l^2 \left(\sum_{p \in P_w} \sum_{w \in W} x_p \delta_{pl} \gamma_w - g_l\right)$$

$$+ \sum_{w \in W} \sum_{l \in L} \mu_{wl}^3 \left(\sum_{p \in P_w} x_p \delta_{pl} - y_{wl}\right)$$

(LR)

subject to:

$$\sum_{p \in P_w} x_p = 1 \qquad \forall w \in W \qquad \text{(IP 2)}$$

$$0 \le g_l \le \frac{C_l}{I(l)} \qquad \forall l \in L \qquad \text{(IP 4)}$$

$$x_p = 0 \text{ or } 1 \qquad \forall p \in P_w, w \in W \qquad \text{(IP 6)}$$

$$y_{wl} = 0 \text{ or } 1 \qquad \forall w \in W, l \in L. \qquad \text{(IP 7)}$$

Note that the constraints are relaxed in such a way that the corresponding Lagrangean multipliers μ_w^1, μ_l^2, and μ_{wl}^3 are nonnegative. To solve this problem, we can decompose (LR) into the following two independent and easily solvable optimization subproblems. Besides, it is worth to remind that this solution approach is not guaranteed to have a feasible solution due to accessing a big value of γ_w into the network which will result in unsatisfactory QoS requirement to the O-D pair w.

3.2.2.1 Subproblem 1 (related to decision variable x_p)

Objective function:

$$Z_{Sub1}(\mu_l^2, \mu_{wl}^3)$$
$$= \min \sum_{w \in W} \sum_{p \in P_w} \sum_{l \in L} (\mu_{wl}^3 + \mu_l^2 \gamma_w) x_p \delta_{pl} \qquad \text{(Sub 1)}$$

subject to:

$$\sum_{p \in P_w} x_p = 1 \qquad \forall w \in W \qquad \text{(IP 2)}$$

$$x_p = 0 \text{ or } 1 \qquad \forall p \in P_w, w \in W. \qquad \text{(IP 6)}$$

This problem can be further decomposed into $|W|$ independent shortest path problems with nonnegative arc weights. Each shortest path problem can be easily solved by the Dijkstra's algorithm. Note that $(\mu_{wl}^3 + \mu_l^2 \gamma_w)$ is exactly the arc weight form we want to determine for the distributed routing protocol. The more details will be described in Chapter 4.

3.2.2.2 Subproblem 2 (related to decision variable y_{wl} and g_l)

Objective function:

$$Z_{Sub2}(\mu_w^1, \mu_l^2, \mu_{wl}^3)$$
$$= \min[\sum_{l \in L}(D_l(g_l)g_l + \sum_{w \in W}\mu_w^1 y_{wl} D_l(g_l) - \mu_l^2 g_l - \sum_{w \in W}\mu_{wl}^3 y_{wl})$$
$$- D_w \sum_{w \in W}\mu_w^1] \quad \text{(Sub 2)}$$

subject to:

$$0 \le g_l \le \frac{C_l}{I(l)} \quad \forall l \in L \quad \text{(IP 4)}$$

$$y_{wl} = 0 \text{ or } 1 \quad \forall w \in W, l \in L. \quad \text{(IP 7)}$$

Because the last term $-D_w \sum_{w \in W}\mu_w^1$ in the objective function (Sub 2) will not affect the optimal solution, it can be disregarded first and added back to the objective value. Therefore, (Sub 2) can be reformulated and can be further decomposed into $|L|$ independent subproblems. For each link l,

$$\min[D_l(g_l)g_l + \sum_{w \in W}\mu_w^1 y_{wl} D_l(g_l) - \mu_l^2 g_l - \sum_{w \in W}\mu_{wl}^3 y_{wl}] \quad \text{(Sub 2.1)}$$

subject to:

$$0 \le g_l \le \frac{C_l}{I(l)} \quad \forall l \in L \quad \text{(IP 4)}$$

$$y_{wl} = 0 \text{ or } 1 \quad \forall w \in W, l \in L. \quad \text{(IP 7)}$$

In problem (Sub 2.1), the first term $D_l(g_l)g_l$ is a monotonically increasing and nonnegative function, and it will not affect the optimal value of the other terms in (Sub

2.1); for this reason, we can express (Sub 2.1) as (Sub 2.1’) for each link l,

$$\min[\sum_{w\in W}\mu^1_w y_{wl} D_l(g_l) - \mu^2_l g_l - \sum_{w\in W}\mu^3_{wl} y_{wl}] \qquad \text{(Sub 2.1')}$$

subject to:

$$0 \le g_l \le \frac{C_l}{I(l)} \qquad \forall l \in L \qquad \text{(IP 4)}$$

$$y_{wl} = 0 \text{ or } 1 \qquad \forall w \in W, l \in L. \qquad \text{(IP 7)}$$

This problem can be solved by the algorithm developed by Cheng and Lin in [22]. The solving steps of the algorithm are briefly described as bellow.

Table 3-2 The Algorithm for Solving Problem (Sub 2.1’)

Step 1:	Solve $y^*_{wl}(g_l) = \mu^1_w D_l(g_l) - \mu^3_{wl} = 0$ for each O-D pair w, and define the results as a set of break points of g_l.
Step 2:	Sort these break points and denote as $g^1_l, g^2_l, ..., g^n_l$. There are at most $\|W\|$ break points.
Step 3:	At each interval $g^i_l \le g_l \le g^{i+1}_l$, $y^*_{wl}(g_l)$ is 1 if $\mu^1_w D_l(g_l) - \mu^3_{wl} \le 0$ and is 0 otherwise.
Step 4:	Denote $\sum_{w\in W}\mu^1_w y^*_{wl}(g^i_l)$ as a_l and $\sum_{w\in W}\mu^3_{wl} y^*_{wl}(g^i_l)$ as b_l, within the interval $g^i_l \le g_l \le g^{i+1}_l$, the problem (Sub 2.1’) can be expressed as: $a_l D_l(g_l) - \mu^2_l g_l - b_l$. Thus, the local minimum is either the boundary point, g^i_l or g^{i+1}_l, or at point g^*_l, where g^*_l is the solution to $a_l \frac{\partial D_l(g_l)}{\partial g_l} - \mu^2_l = 0$.

Step 5: By examine the $|W|+1$ intervals, the global minimum point can be found by comparing these local minimum points.

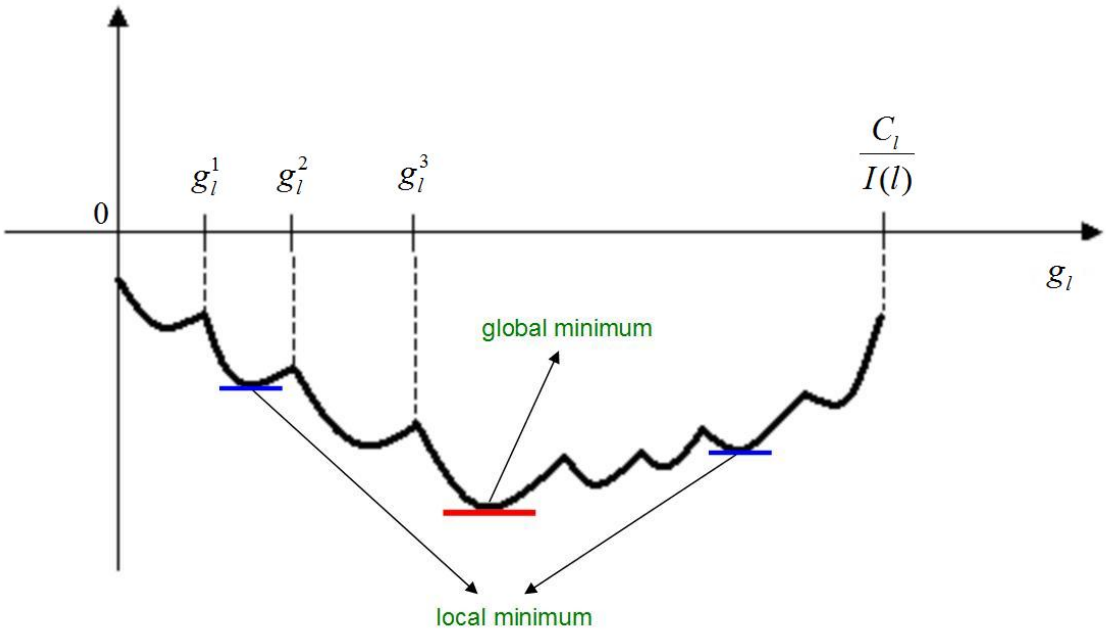

Figure 3-10 A Typical Graph of Problem (Sub 2.1')

3.2.3 The Dual Problem and the Subgradient Method

According to the weak Lagrangean duality theorem [23], for any $\mu_w^1, \mu_l^2, \mu_{wl}^3 \geq 0$, the objective value of Lagrangean Relaxation problem $Z_{LR}(\mu_w^1, \mu_l^2, \mu_{wl}^3)$ is a lower bound of the primal problem Z_{IP}. Based in problem (LR), the following dual problem Z_D is constructed to calculate the tightest lower bound.

Dual problem (D):

$$Z_D = \max Z_D(\mu_w^1, \mu_l^2, \mu_{wl}^3) \quad \text{(D)}$$

subject to:

$$\mu_w^1, \mu_l^2, \mu_{wl}^3 \geq 0.$$

We use a popular method, the subgradient method [20][21], for solving the dual problem (D). Let the vector S be a subgradient of $Z_D(\mu_w^1, \mu_l^2, \mu_{wl}^3)$. In iteration k of the subgradient optimization procedure, the multiplier vector $\pi^k = (\mu_w^1, \mu_l^2, \mu_{wl}^3)$ is updated by $\pi^{k+1} = \pi^k + t^k S^k$. The step size t^k is determined by $t^k = \lambda \frac{(Z_{IP}^h - Z_D(\pi^k))}{\left\| S^k \right\|^2}$. Z_{IP}^h is the primal objective function value (an upper bound on Z_{IP}) and λ is constant where $0 \leq \lambda \leq 2$.

3.2.4 Getting Primal Feasible Solutions

If there is a primal feasible solution to (IP), it must be a solution to (LR) and satisfy all constraints as well. Otherwise, it has to be modified to be feasible to (IP) by a getting primal feasible solutions heuristic. According to the computational experiments in [15], we can get a better solution by the heuristic which considers the end-to-end delay constraints first. That is, if the end-to-end delay constraints are not satisfied, the arc weights along those paths that violate the end-to-end delay constraints are increased; therefore, the routing assignments have to be recalculated.

Note that, this solution approach needs few minutes to get the feasible solution or even the near-optimal solution. However, in this thesis, we focus on the dynamic network environment and derive the arc weight from the solving process of this approach. Therefore, we take this arc weight to be our routing metric which is used by the distributed routing protocol for dynamic routing.

Chapter 4 Distributed Routing Algorithm

4.1 Routing Metric

A metric can be seen as a cost assigned for passing through the network; moreover, by the routing algorithm, each router chooses the path with the smallest (shortest) sum of the metrics. Distinct routing protocols define the metric totally differently. The metric can be distance, number of hops, delay, and so on. Here, we define the metric as a combination of the link mean delay and the derivative of queue length for each link.

In Section 3.2, we use Lagrangean Relaxation method to relax some constraints in the formulation and decompose (LR) into two independent subproblems. In the first subproblem (Sub 1), the nonnegative weight is calculated as $(\mu_{wl}^3 + \mu_l^2 \gamma_w)$ on each link. The actual meaning of the multipliers μ_{wl}^3 and μ_l^2 are the link mean delay and the derivative of queue length, respectively, and both can be derived from the solving process of the problem (LR).

By the problem (LR), the multiplier μ_{wl}^3 is related to the relaxation of link selection for each O-D pair. When we add one unit on the decision variable of link selection (i.e., y_{wl}), it will generate one unit of the corresponding mean delay as formulated in QoS constraint. But when the problem (LR) is solved, the decision variable y_{wl} is either 1 or 0 not only one unit, thus, the multiplier μ_{wl}^3 is equivalent to the mean delay on the chosen link.

Besides, the link selection is based on the sum of link mean delay $D_l(g_l)$ of each O-D pair weighted by μ_w^1, that is, how much the end-to-end delay will impact on the objective function (IP). Moreover, in the problem (Sub 2.1'), the first step for solving it is also related to the multiplier μ_{wl}^3 and chooses the links for each O-D pair according to $y_{wl}^*(g_l) = \mu_w^1 D_l(g_l) - \mu_{wl}^3 = 0$. It means that whenever we select a link for an O-D pair, the link will produce corresponding link mean delay μ_{wl}^3 and affect the end-to-end delay constraint which is weighted by multiplier μ_w^1 in the problem (LR).

On the other hand, the multiplier μ_l^2 is related to the relaxation of aggregate flow over each link. Whenever we relax one unit of the link flow, how much it affects the objective function (IP), i.e., the sensitivity of the average number of packets over the link with respect to the aggregate flow.

Briefly speaking, μ_{wl}^3 implies the link mean delay of each link, and μ_l^2 indicates the derivative of queue length. The term γ_w in the arc weight form represents the weighting factor between μ_{wl}^3 and μ_l^2. Thus, we use this metric which consists of the link mean delay and the derivative of queue length for considering both system perspective and user perspective for our distributed routing protocol, and use the traffic requirement of each O-D pair as the weighting factor to combine these two parameters.

Table 4-1 The Arc Weight of Each Link

$$
\begin{aligned}
\textit{arc weight} &= (\mu_{wl}^{3} + \mu_{l}^{2}\gamma_{w}) \\
&= \textit{link mean delay} + \textit{derivative of queue length} \times \textit{required traffic} \\
&= D_l(g_l) + \gamma_w \times \frac{\partial\left(D_l(g_l)g_l\right)}{\partial g_l} \\
&= D_l(g_l) + \gamma_w \times \left(D_l(g_l) + g_l \times \frac{\partial D_l(g_l)}{\partial g_l} \right)
\end{aligned}
$$

4.2 Estimation of Routing Metric Parameters

1) Perturbation Analysis:

In [19], Towsley *et al*. proposed an useful technique, "Perturbation Analysis" (PA), to effectively estimate the so called "marginal delay" which is defined in [18]. In precisely speaking, this method is used to estimate the derivative of queue length. As mentioned in previous section, the derivative of queue length is composed of link mean delay, aggregate flow, and the partial derivative of link mean delay shown as **Table 4-1**. The PA approach can also estimate the value of link mean delay which is in advance of calculating the derivative of queue length. By using the PA technique, we can obtain the link mean delay and the derivative of queue length; both of them are the major information of the distributed routing protocol.

2) Time Stamping for Packets:

Besides, we can use other approaches to estimate the essential information of each link in a conceptually straightforward way. While receiving a packet, the packet delay time can be calculated by the difference between the sending time and the receiving time. For each adjacent link mean delay, we can divide the sum of each packet delay by the number of received packets corresponding to each neighbor node.

3) Approximation Function:

For the derivative of queue length, we can record the queue length and the corresponding aggregate flow over the link at each time interval. The newer the record is, the more weight the record is assigned. We can use a monotonically increasing and convex function, such as power regression function (i.e., $y = ax^b$, where $a > 0$ and $b > 1$) to approximate the recording results, and the newest record has the most impact to this approximation. Therefore, we can obtain the derivative of queue length by calculating the derivative of the function with respect to the corresponding aggregate flow on the link.

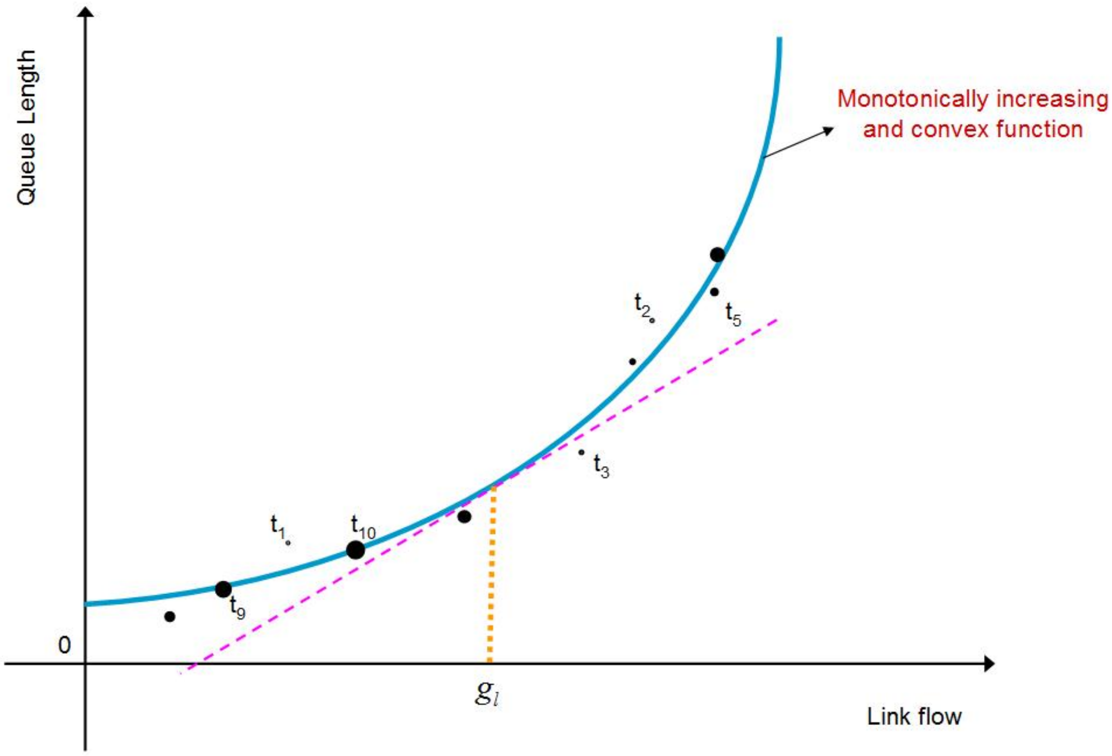

Figure 4-1 Estimation of the Derivative of Queue Length

4.3 Distributed Routing Protocol

The routing protocol we used here is based on the well-known link-state routing protocol [13][14]. Link-state routing protocol is allowed to assign a cost (i.e., metric) to each route. The link mean delay is provided to be a metric by the present routing protocol. We can add the value of the derivative of queue length to the information exchanged by each node.

The main idea of this kind of routing is that each node shares the state about its neighborhood to every other node in the network. By exchanging local information between all nodes, each node will have all link states in its own database [14], namely, every node has the whole network topology including the weight information of each link. The link state packets for all nodes in **Figure 4-2** are shown in **Figure 4-3**.

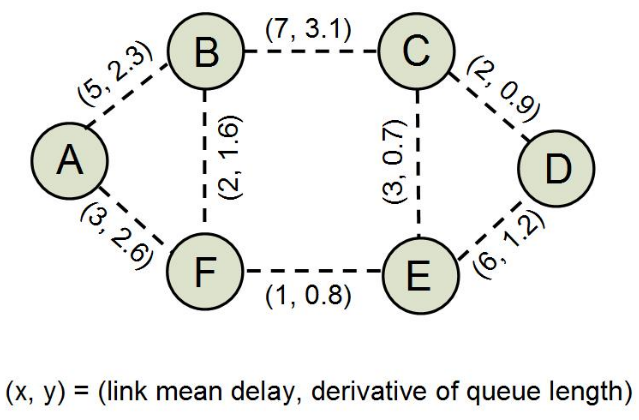

Figure 4-2 Simple Networks with Weight Information of Each Link

A		
Sequence Number		
Age		
B	5	2.3
F	3	2.6

B		
Sequence Number		
Age		
A	5	2.3
C	7	3.1
F	2	1.6

C		
Sequence Number		
Age		
B	7	3.1
D	2	0.9
E	3	0.7

D		
Sequence Number		
Age		
C	2	0.9
E	6	1.2

E		
Sequence Number		
Age		
C	3	0.7
D	6	1.2
F	1	0.8

F		
Sequence Number		
Age		
A	3	2.6
B	2	1.6
E	1	0.8

Figure 4-3 Link State Packets for All Nodes in Simple Networks

The idea behind this routing protocol is simple, and the most important is that we can only make a few changes in present link-state routing protocol to apply to our routing algorithm. Each node must do the following steps in **Table 4-2**.

Table 4-2 Distributed Routing Protocol for Each Node

Step 1:	Discover its neighbors and learn their network address.
Step 2:	Measure the link mean delay and the derivative of queue length to each of its neighbors.
Step 3:	Build a packet with the above information at regular intervals.
Step 4:	Flood this packet to all other node in the network.
Step 5:	Compute the shortest path based on the arc weight form to every other node.
Step 6:	Flood the path results to all other nodes included in the path.

Whenever a route is required from a source to any destination in the network, the source node can computes the arc weight for each link by the required traffic and the essential information of each link in the link state database.

4.4 Admission Control Heuristic Algorithm

By the link-state routing protocol, we can apply the Dijkstra algorithm to each node to calculate the routing table. The Dijkstra algorithm can compute the shortest path between any two nodes in the network. The algorithm denotes two states for the nodes, tentative and permanent. We briefly described the steps of the Dijkstra algorithm as below.

Table 4-3 The Dijkstra Routing Algorithm [14]

Step 1: Start with the local node (i.e., the root of the tree).
Step 2: Assign a cost of 0 to this node and make it the first permanent node.
Step 3: Examine each neighboring node of the last permanent node.
Step 4: Assign a cumulative cost to each node and make it tentative.
Step 5: Among the list of tentative nodes
Step 5.1: Find the node with the smallest cumulative cost and make it permanent.
Step 5.2: If a node can be reached from more than one direction
Step 5.2.1: Select the direction with the shortest cumulative cost.
Step 6: Repeat steps 3 to Step 5 until every node becomes permanent.

The Dijkstra algorithm computes the only one shortest path for each node pair, to make our routing algorithm be more flexible, we want to construct the second shortest path, the third shortest path, and so on; that is so called “K shortest paths.” There are many researches about the K shortest paths problem as mentioned in [24] and [25], and here we adopt the K shortest paths (KSP) algorithm proposed by Katoh *et al*. in [24]. The algorithm is suitable for an undirected graph with nonnegative arc weight and can compute K simple paths (i.e., without cycles between the paths) in $O(Kc(n,m))$ time. where $c(n,m)$ is the time to compute shortest paths from one node to all the other nodes.

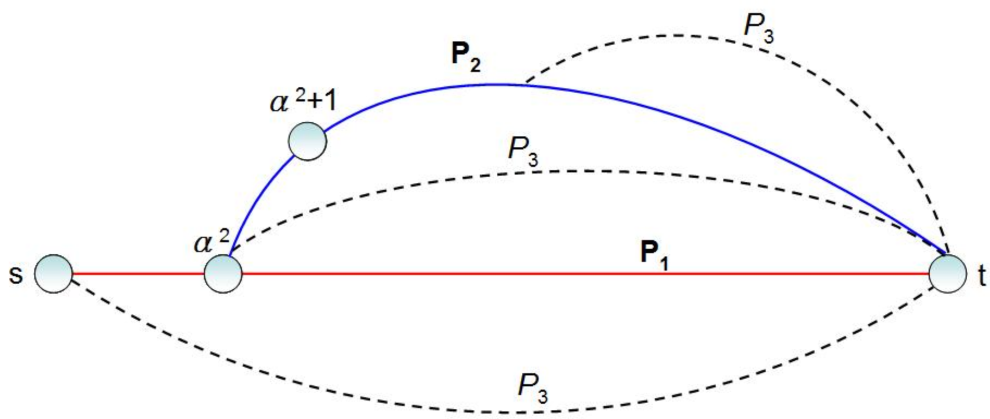

Figure 4-4 Illustration of K Shortest Paths

It means that, if we use the Dijkstra algorithm to be the shortest path algorithm, the complexity of the K shortest paths algorithm will be $O(Kn^2)$. Furthermore, the metric for calculating the candidate shortest paths is not only the arc weight, but also the predictive link mean delay. In other words, two types of the K shortest paths. If all K shortest paths calculated by the original arc weight cannot satisfy the end-to-end delay requirement, we try to use the predictive link mean delay to be the metric for each link and calculate the corresponding K shortest paths, i.e., the K fastest paths, to satisfy the maximum allowable QoS requirement as possibly as we can.

By the K shortest paths and the K fastest paths, we can construct our admission control heuristic algorithm. The idea behind this admission control heuristic algorithm is simple, that is, we give consideration to both "system perspective" and "user perspective," but the K shortest paths may not satisfy the basic assumption, the QoS provisioning. Thus, we must provide other K fastest paths to meet the QoS requirement, and the original arc weight corresponding to each of which cannot exceed a threshold β which confines the effect to the whole system. The detail steps are shown in **Table 4-4**.

Table 4-4 The Admission Control Heuristic Algorithm

Step 1: Calculate the fastest path by the predictive link mean delay as the arc weight.
Step 1.1: If the fastest path can not satisfy the QoS requirement.
Step 1.1.1: Reject this traffic.
Step 2: Calculate K shortest paths by the arc weight we proposed, and denote θ as the cost of first shortest path.
Step 3: Compare the QoS requirement to the predictive end-to-end delay of each K shortest paths.
Step 3.1: If one of the K shortest path can satisfy the QoS requirement
Step 3.1.1: Choose the shortest one for routing this traffic.
Step 4: Calculate K fastest paths by the predictive link mean delay as the weight of each link, and set α_K to be the original length corresponding to the K-th fastest path.
Step 4.1: If the K-th fastest path can satisfy the QoS requirement and $\frac{\alpha_K}{\theta} \le \beta$
Step 4.1.1: Choose the fastest one (i.e., smallest K), and route this traffic.
Step 5: Reject this traffic.

We can use the power regression function to predict the link mean delay of each link. By admission control heuristic algorithm, we can reject the required traffic which can not satisfy the QoS requirement with the predictive end-to-end delay. Therefore, we can reduce the impact of whole system and other flows in advance and achieve a better performance by this mechanism.

Chapter 5 Simulation

5.1 Simulation Environment

We construct four grid topologies of 3×3, 5×5, 7×7 and 9×9 squares, and place one node in each intersection point as shown in **Figure 5-1**. Each mesh router has a radio transmission range of 250m and a radio interference range of 550m. The only one mesh access point is located on the center of the first row in each square, and the sessions are randomly generated by the other nodes in the topology. We conduct simulations with ns-2 simulator to evaluate the performance of Near-Optimal Distributed QoS Constrained (NODQC) routing algorithm with different session types and compare with other routing algorithms.

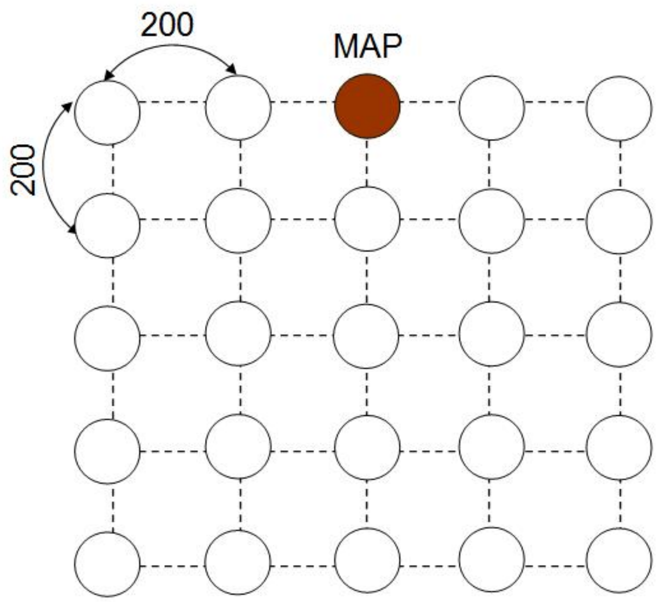

Figure 5-1 Simulation Environment (5×5)

Each session will be disposed one path for routing its required traffic and not allowed to change the routing path in the holding time. The session arrivals are followed Poisson process, and the holding time of each session is set to be an exponential distribution with average 10 seconds. We experiment with average session arrival rates of 0.25, 0.5, 1 and 2, and the corresponding packet arrival rates are 40, 20, 10 and 5. The QoS requirements set in 3×3, 5×5, 7×7 and 9×9 squares are 3ms, 4ms, 5ms and 6ms, respectively.

Table 5-1 Experimental Session Types

Parameters	Value			
Average Holding Time	10			
Average Session Arrival Rate	0.25	0.5	1	2
Average Number of Active Sessions	2.5	5	10	20
Packet Arrival Rate	40	20	10	5
Average Traffic Input	100($packets/s$)			

As we can see in **Table 5-1**, there are average 100 packets per second in the network of each topology. The size of each transmitting UDP packet is 1000 bytes at each packet arrival rate. Besides, the control messages of distribution routing protocol, such as HELLO message for sensing the neighbors and TC message for broadcasting topology information are sent at a fixed period of 5 seconds in the simulation stage.

To estimate the parameters of our routing metric, we record each packet delay and packet inter arrival time, and then calculate the mean delay and aggregate flow with corresponding queue length (i.e., average number of packets) of each link every 1 second. We set the number of fitting data for power regression to be 10, that is, the data used to compute the regression function are the newest 10 records and the interval of each is 1 second. In addition, the value of K in the routing algorithm is set as 5 (i.e., at most 5 shortest or fastest paths for each O-D pair) and the threshold β for admission control heuristic algorithm is set to be 1.3.

Table 5-2 Simulation Parameters

Parameters	Value
Transmit and Receive Antenna Gain	1.0
Transmit and Receive Antenna Height	1.5(m)
Reception Threshold	$3.625e-10$
Carrier Sensing Threshold	$1.559e-11$
Transmission Range	250(m)
Interference Range	550(m)
Distance between Each Node	200(m)
UDP Packet Size	1000(*bytes*)
Sending Interval of HELLO Message	2(s)
Sending Interval of TC Message	5(s)
Recording Interval of Fitting Data	1(s)
Number of Fitting Data	10
K	5
β	1.3

If the received signal strength is greater than the reception threshold, the packet can be successfully received. If received signal strength is greater than the carrier sensing threshold, the packet transmission can be sensed. However, the packet cannot be decoded unless signal strength is greater than the reception threshold. Both the transmission range and interference range are calculated by two-ray ground reflection model according to the reception threshold and the carrier sensing threshold, respectively.

Table 5-3 Two-Ray Ground Reflection Model

$$\mathit{Transmission\ Range}\ (\mathit{or\ Interference\ Range}) =$$

$$\sqrt[4]{\frac{TransmitPower \times TransmitAntennaGain \times \mathrm{Re}ceiveAntennaGain \times (TransmitAntennaHeight)^2 \times (\mathrm{Re}ceiveAntennaHeight)^2}{Threshold}}$$

In this thesis, we focus on the effectiveness of our routing metric and related parameters at the on-line stage, and by which, we can reduce the impact of the whole system between each coming session with QoS provisioning. Thus, we simply use single channel in the simulation environment to decouple the effect of the channel assignment algorithm and evaluate the performance of our routing algorithm.

5.2 Simulation Results and Discussion

5.2.1 Simulation Results

In this section, we present the results from simulation to demonstrate the effectiveness of our routing algorithm, NODQC. We evaluate NODQC with different session types under the same average traffic loading in the network, and compare the performance with other routing algorithms in terms of average end-to-end delay, delay jitter and system throughput with QoS satisfaction. Note that, the delay jitter is defined as the variance in the following tables and figures. We show the simulation results in the following tables and figures.

To evaluate our routing algorithm with different session types as defined in **Table 5-1**, we take 7×7 square as the network topology to simulate with 660 seconds and measure the packets at last 600 seconds. For comparison with other routing algorithms, we choose the session type whose average session arrival rate and packet arrival rate are equal to 0.25 and 40, respectively, and use different network sizes to show the performance of Near-Optimal Distributed QoS Constrained (NODQC), Optimized Link State Routing (OLSR), Ad Hoc On-Demand Distance Vector (AODV) and Destination-Sequenced Distance Vector (DSDV) routing algorithms. The simulation time is set as 660 seconds and the measurement time is the last 600 seconds.

Table 5-4 Evaluation with Different Session Rates (Average End-to-End Delay)

Average End-to-End Delay (*ms*)				
Average Session Rate	0.25	0.5	1	2
Packet Arrival Rate	40	20	10	5
Average End-to-End Delay	3.325215	3.730487	3.88681	3.942485

Table 5-5 Evaluation with Different Session Rates (System Throughput with QoS)

System Throughput with QoS (*Kbps*)				
Average Session Rate	0.25	0.5	1	2
Packet Arrival Rate	40	20	10	5
System Throughput with QoS	632.062518	521.353393	456.291533	376.553628

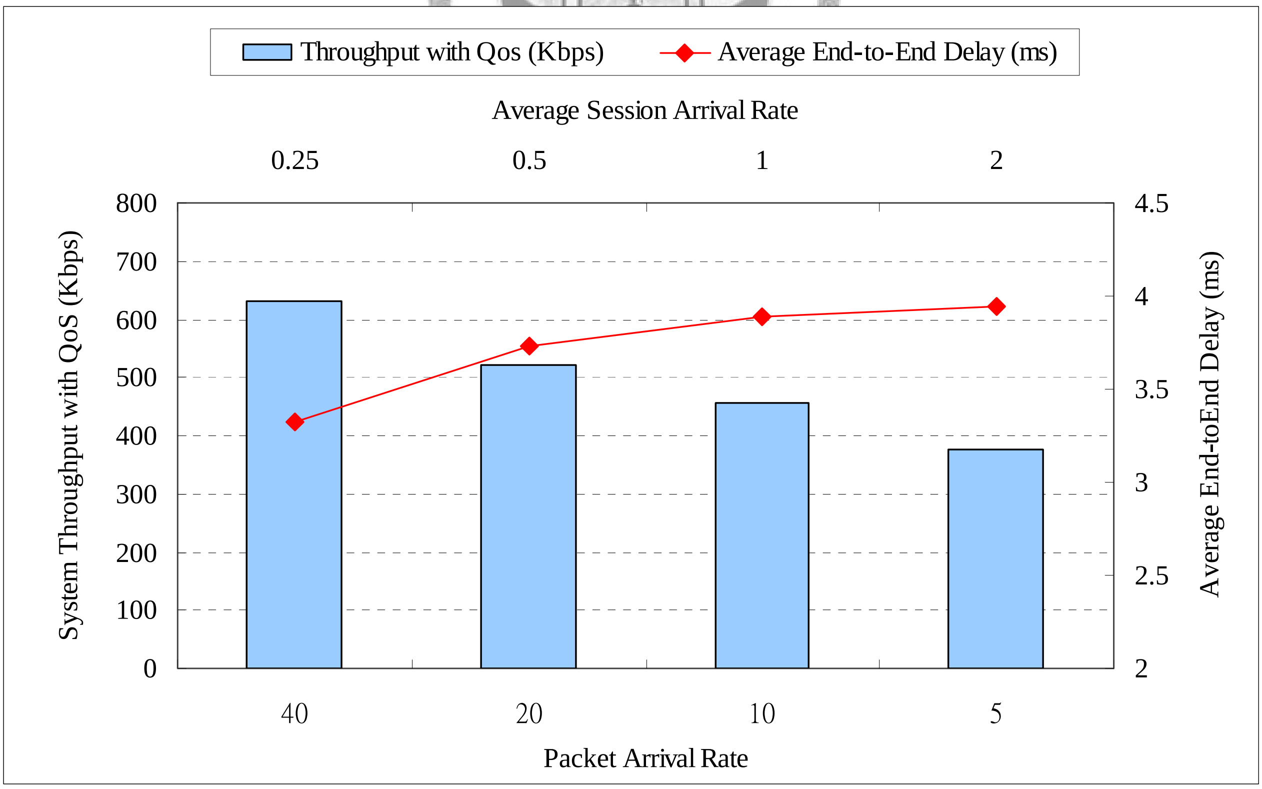

Figure 5-2 Evaluation with Different Session Arrival Rates

Table 5-6 Simulation Results of Routing Algorithms (Average End-to-End Delay)

Average End-to-End Delay (*ms*)				
Routing Algorithms	3×3	5×5	7×7	9×9
ODQC	1.54599	2.550296	3.325215	4.176691
OLSR	1.28128	2.522533	3.620345	4.792818
AODV	1.248075	2.339906	3.848888	5.456055
DSDV	1.220291	2.495117	3.873495	9.171553

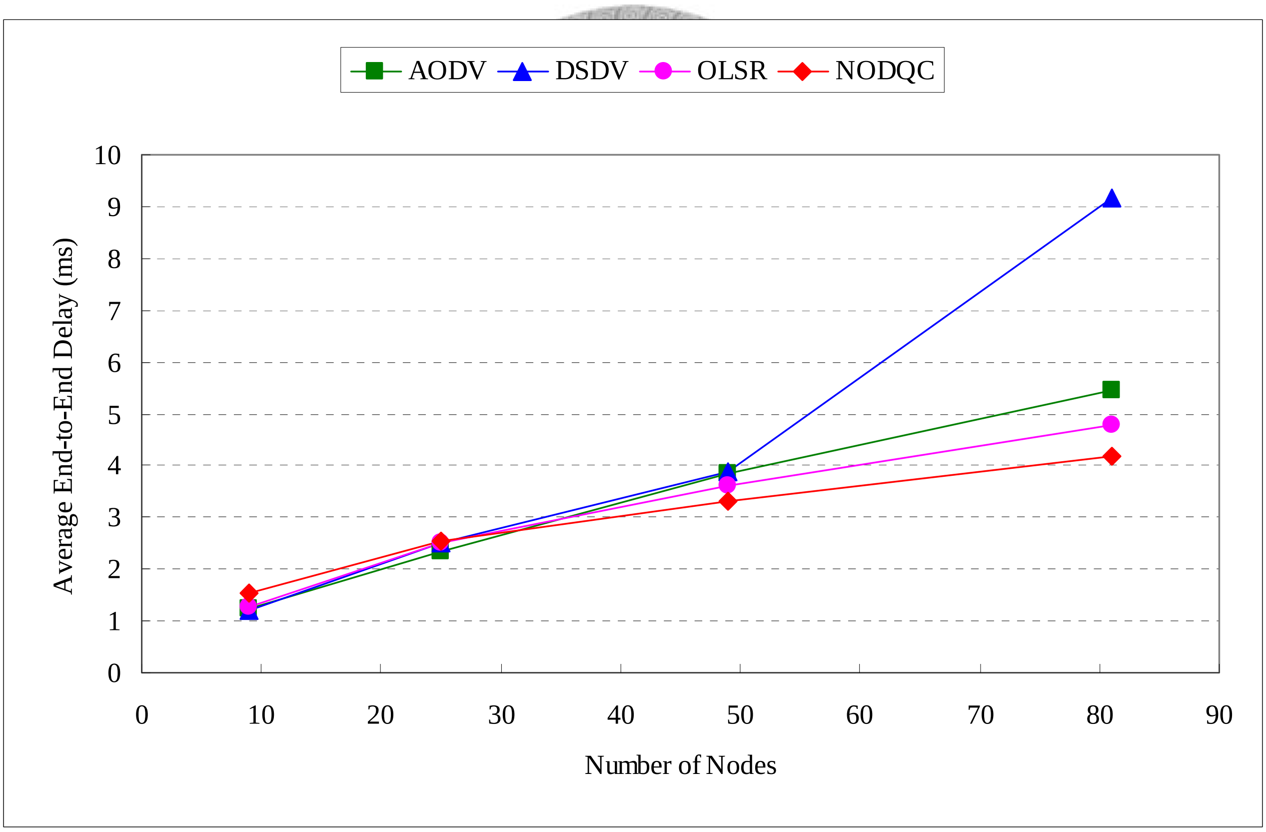

Figure 5-3 Simulation Results of Routing Algorithms (Average End-to-End Delay)

Table 5-7 Simulation Results of Routing Algorithms (Delay Jitter)

Delay Jitter (ms^2)				
Routing Algorithms	3×3	5×5	7×7	9×9
NODQC	0.623741	1.529524	2.789917	4.239667
OLSR	0.536928	1.463894	3.332403	5.098757
AODV	0.556566	1.720252	5.129518	13.445979
DSDV	0.330232	1.418664	3.424205	11.586675

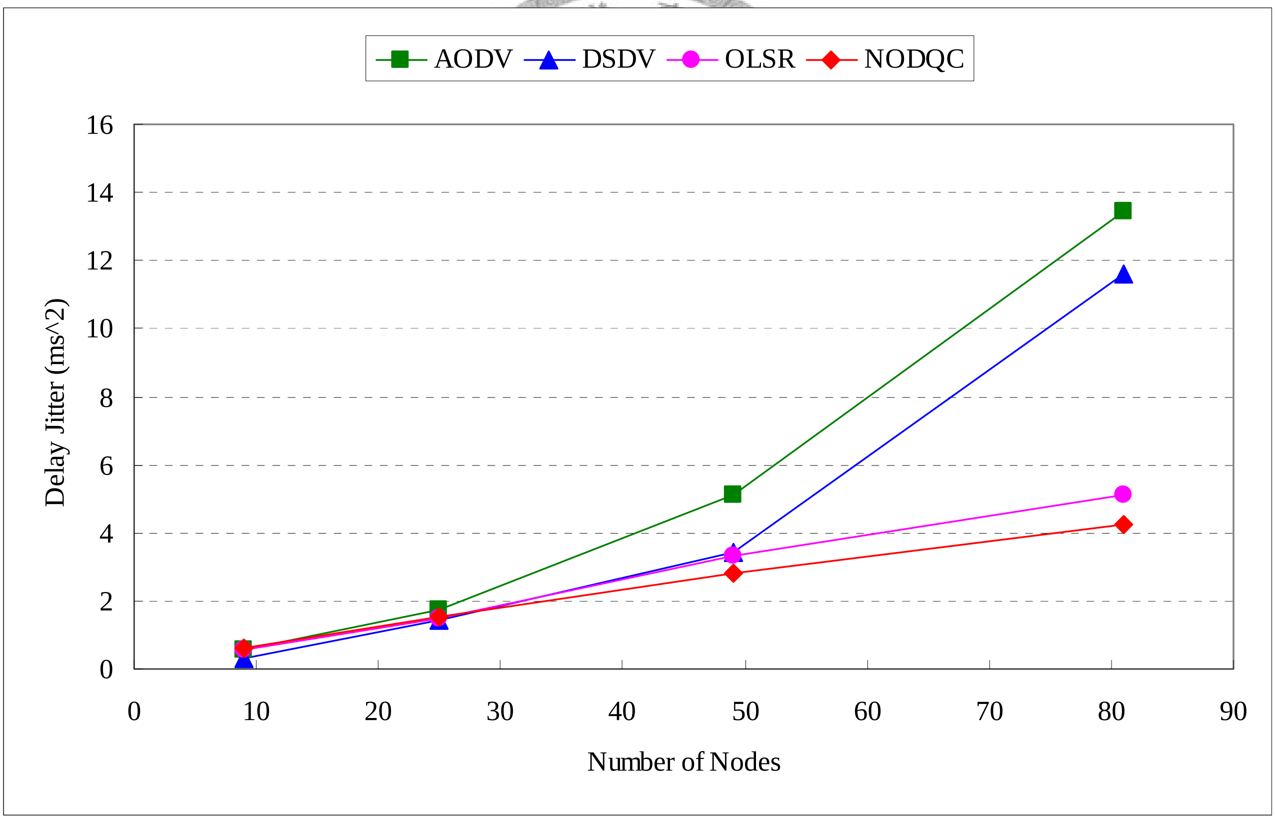

Figure 5-4 Simulation Results of Routing Algorithms (Delay Jitter)

Table 5-8 Simulation Results of Routing Algorithms (System Throughput with QoS)

System Throughput with QoS (*Kbps*)				
Routing Algorithms	3×3	5×5	7×7	9×9
NODQC	716.849085	673.698465	632.062518	612.322159
OLSR	753.008325	691.484223	603.130035	556.022380
AODV	757.683145	700.203701	575.904857	516.320711
DSDV	773.355103	701.779457	602.953338	509.221205

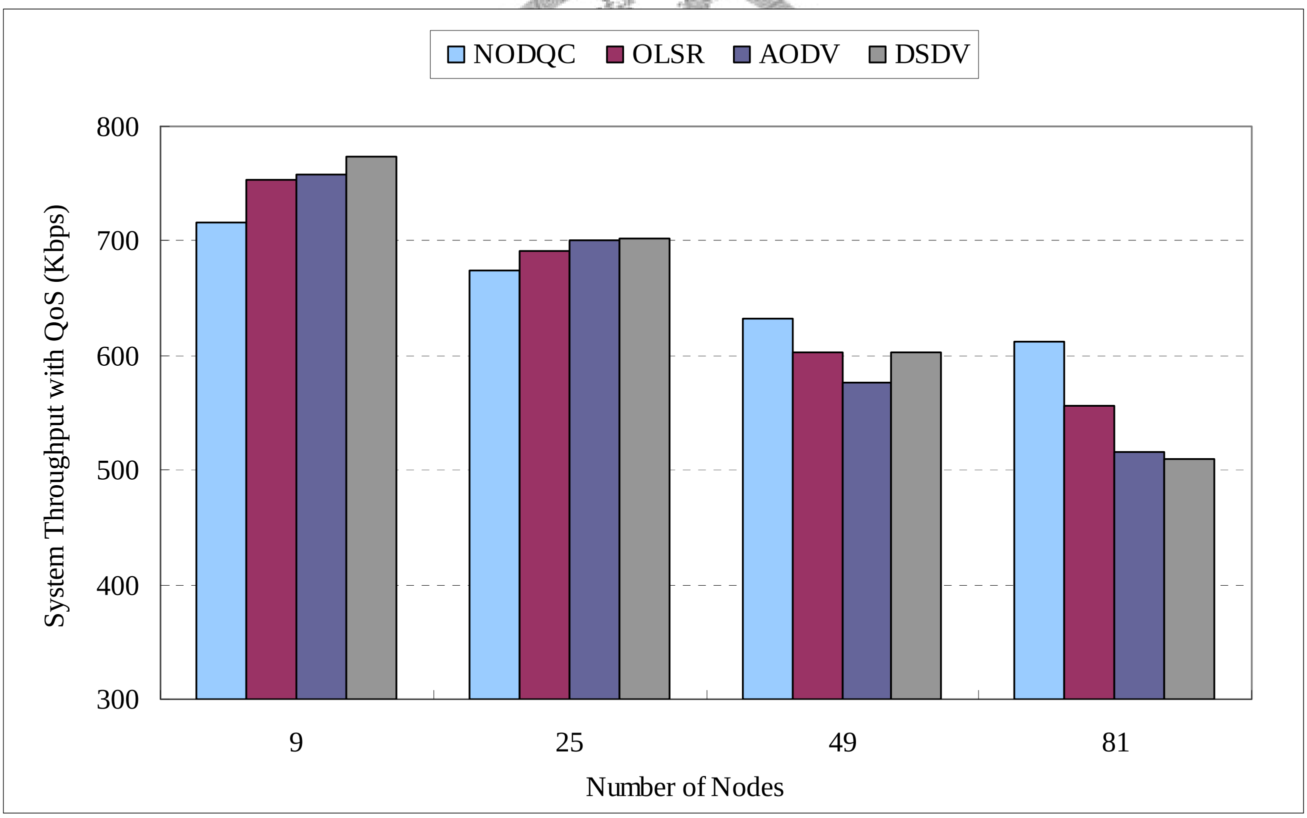

Figure 5-5 Simulation Results of Routing Algorithms (System Throughput with QoS)

5.2.2 Discussion of Simulation Results

From **Table 5-4**, **Table 5-5** and **Figure 5-2**, obviously, the value of average end-to-end delay increases and the system throughput with QoS decreases with the average session arrival rate. This is because that the nodes exchange information every 5 seconds, but the sessions enter the network at a short time interval. As a result, most of the sessions are routed to a suboptimal path due to the lack of updated information for routing metric. Therefore, by the evaluation results, we can find out that our routing algorithm performs better in the lower average session arrival rate but higher packet arrival rate under the same system loading of average 100 packets per second.

Table 5-6 to **Table 5-8** and **Figure 5-3** to **Figure 5-5** are the results of performances between different routing algorithms in each network size. In the small-scale network (e.g., 3×3 and 5×5 squares), the broadcasting of routing protocol control messages for exchanging information will cause quite large delay to the sending packets especially for the destination node. Besides, the average path length of each session is shorter in small network size, in consequence, the superiority of our routing metric is not obvious in the small-scale network.

When the network size gets large, the average path length of each session will become longer and the path selection at this scenario will also be more important. The

NODQC can choose the less congested paths for the sessions and balance the traffic loading for the whole network; moreover, the NODQC also takes the QoS provisioning into account for the coming sessions, consequently, the system throughput with QoS satisfaction is more than others in larger network size.

Evidently, the NODQC has lower average end-to-end delay and delay jitter and higher system throughput with QoS satisfaction than other routing algorithms in large-scale network because both the system perspective and user perspective are concurrently considered in NODQC.

Chapter 6 Conclusion

6.1 Summary

In wireless mesh networks, the mesh routers forward the packets to the Internet via the mesh access points with wired line. It is expectable that each router wants to route the required traffic on the optimal path in its own way, but this kind of routing decision will cause local congestion of the network.

In this thesis, we develop a simple channel assignment heuristic algorithm to decide the topology of WMN and derive the arc weight from Lagrangean Relaxation based problem formulation, which is composed of the link mean delay and the derivative of queue length (i.e., average number of packets on the link) to take "system perspective" and "user perspective" into account. After that, we use link-state routing protocol to employ our routing metric and construct K shortest paths and K fastest paths for admission control heuristic algorithm.

The proposed routing metric considers the average conditions for each link, so it requires a short time interval to gather the essential information and well define the situation of whole network. In addition, in the larger network size, the superiority of the Near-Optimal Distributed QoS Constrained (NODQC) routing algorithm is more obvious, since the path selection is more important at this scenario. The goal of NODQC is to select the path which has less variance of congestion while considering the QoS provisioning for the users. From the simulation results, especially in large-scale network, the NODQC has lower average end-to-end delay and delay jitter than Optimized Link State Routing (OLSR), Ad Hoc On-Demand Distance Vector (AODV) and Destination-Sequenced Distance Vector (DSDV) and outperforms them in terms of system throughput with QoS satisfaction.

6.2 Future Work

1) Lagrangean Multipliers and Karush-Kuhn-Tucker Conditions:

One of the most important parts of this thesis is the routing metric and the related parameters. We use Lagrangean Relaxation formulation to present the arc weight and infer the actual meaning of the corresponding multipliers. In addition to Lagrangean Relaxation, we can also use Lagrangean multipliers and Karush-Kuhn-Tucker Conditions to well describe the arc weight form. It could simplify the process of inference and have the same consequence with Lagrangean Relaxation.

2) QoS Metric:

By the solving process of the Lagrangean Relaxation formulation, we can implement other QoS requirements on different purposes of services. The end-to-end delay is presented as a function in our formulation to be the QoS consideration. We can apply distinct QoS metrics (e.g., delay jitter or packet loss rate) into the Lagrangean Relaxation formulation, and the corresponding multiplier of the routing metric will be replaced as the new defined QoS metric instead of the link mean delay.

References

[1] T. Liu and W. Liao, “On Routing in Multichannel Wireless Mesh Networks: Challenges and Solutions,” *IEEE Network*, Vol. 22, NO. 1, pp. 13-18, January/February 2008.

[2] L. Badia, A. Erta, L. Lenzini, and M. Zorzi, “A General Interference-Aware Framework for Joint Routing and Link Scheduling in Wireless Mesh Networks,” *IEEE Network*, Vol. 22, NO. 1, pp. 32-38, January/February 2008.

[3] A. Raniwala, K. Gopalan, and T.-C. Chiueh, "Centralized Channel Assignment and Routing Algorithms for Multi-channel Wireless Mesh Networks," *ACM Mobile Computing and Communications Review (MC2R)*, Vol. 8, NO. 2, pp. 50–65, April 2004.

[4] P. Kyasanur and N. H. Vaidya, “Routing and Interface Assignment in Multi-Channel Multi-Interface Wireless Networks,” *Proc. IEEE WCNC*, Vol. 4, pp. 2051- 2056, 2005.

[5] A.P. Subramanian, H. Gupta, S.R. Das, “Minimum Interference Channel Assignment in Multi-Radio Wireless Mesh Networks,” *Proc. IEEE SECON*, pp. 481-490, 2007.

[6] R. Draves, J. Padhye, and B. Zill, “Routing in Multi-Radio, Multi-Hop Wireless Mesh Networks,” *Proc. ACM MOBICOM*, pp. 114-128, 2004.

[7] T. Liu and W. Liao, "Capacity-Aware Routing in Multi-Channel Multi-Rate Wireless Mesh Networks," *Proc. IEEE ICC*, Vol. 5, pp. 1971-1976, 2006.

[8] Y.C. Tzeng, "Backhaul Assignment and Routing Algorithm with End-to-End QoS Constraints in Wireless Mesh Networks," *Department of Information Management, National Taiwan University*, July 2006.

[9] H.T. Cheng and W. Zhuang, "Joint Power-Frequency-Time Resource Allocation in Clustered Wireless Mesh Networks," *IEEE Network*, Vol. 22, NO. 1, pp. 45-51, January/February 2008.

[10] M. Alicherry, R. Bhatia, and L. Li, "Joint Channel Assignment and Routing for Throughput Optimization in Multi-Radio Wireless Mesh Networks," *Proc. ACM MOBICOM*, pp. 58-72, 2005.

[11] J. Tang, G. Xue and W. Zhang, "Interference-Aware Topology Control and QoS Routing in Multi-Channel Wireless Mesh Networks," *Proc. ACM MOBIHOC*, pp. 68-77, 2005.

[12] D. Bertsekas and R.G. Gallager, *Data Networks*, 2nd ed., Upper Saddle River, NJ, Prentice Hall, 1992.

[13] A.S. Tanenbaum, *Computer Networks*, 4th ed., Upper Saddle River, NJ, Prentice Hall, 2003.

[14] B.A. Forouzan, *Data Communication and Networking*, 3rd ed., New York,

McGraw Hill, 2003.

[15] H.H. Yen and F.Y.S. Lin, “Near-Optimal Delay Constrained Routing in Virtual Circuit Networks,” *Proc. IEEE INFOCOM*, Vol. 2, pp. 750-756, 2001.

[16] L. Kleinrock, *Queueing Systems*, Vol. 2, New York, Wiley-Interscience, 1976.

[17] Y.F. Wen, “Performance Optimization Algorithm for Wireless Networks,” *Department of Information Management, National Taiwan University*, July 2007.

[18] R.G. Gallager, “A Minimum Delay Routing Algorithm Using Distributed Computation,” *IEEE Transactions on Communications*, Vol. 25, NO. 1, pp. 73-85, January 1977.

[19] C.G. Cassandras, M.V. Abidi, and D. Towsley, “Distributed Routing with On-Line Marginal Delay Estimation,” *IEEE Transactions on Communications*, Vol. 38, NO. 3, pp. 348-359, March 1990.

[20] A.M. Geoffrion, “Lagrangean Relaxation and its Use in Integer Programming,” *Mathematical Programming Study 2*, Vol. 2, pp. 82-114, January 1974.

[21] M.L. Fisher, “An Application Oriented Guide to Lagrangean Relaxation,” *Interfaces*, Vol. 15, NO. 2, pp. 10-21, April 1985.

[22] K.T. Cheng and F.Y.S. Lin, "Minimax End-to-end Delay Routing and Capacity Assignment for Virtual Circuit Networks," *Proc. IEEE GLOBECOM’95*, pp. 2134-2138, 1995.

[23] M.L. Fisher, “The Lagrangean Relaxation Method for Solving Integer Programming Problems,” *Management Science*, Vol. 27, NO. 1, pp. 10-21, January 1981.

[24] N. Katoh, T. Ibaraki, and H. Mine, "An Efficient Algorithm for K Shortest Simple Paths," *Networks*, Vol. 12, NO. 4, pp. 411-427, October 1982.

[25] E. Hadjiconstantinou and N. Christofides, “An Efficient Implementation of an Algorithm for Finding K Shortest Simple Paths,” *Networks*, Vol. 34, NO. 2, pp. 88-101, August 1999.

簡 歷

姓　名：謝 友 仁

出 生 地：台 灣　台 北 市

生　日：中 華 民 國 七 十 三 年 十 二 月 十 七 日

學　歷：九 十 二 九 月 至 九 十 六 年 六 月

國 立 中 央 大 學　資 訊 管 理 學 系 學 士

九 十 六 年 九月 至 九 十 八 年 七 月

國 立 台 灣 大 學　資 訊 管 理 學 研 究 所 碩 士