國立臺灣大學資訊管理研究所

碩士論文

指導教授：林永松　博士

考慮群播服務與路由可變下之允入控制及具服務品質限制條件路由演算法

Admission Control and QoS-constrained Routing Algorithms Considering Multicast Service and Traffic Rerouting

研究生：胡晉華　撰

中華民國九十二年七月

考慮群播服務與路由可變下之允入控制及具服務品質限制條件路由演算法

Admission Control and QoS-constrained Routing Algorithms Considering Multicast Service and Traffic Rerouting

本論文係提交國立台灣大學
資訊管理研究所作為完成碩士
學位所需條件之一部分

研究生：胡晉華　撰

中華民國九十二年七月

謝　　　誌

兩年前，剛退伍的我回到台大資管繼續我未完成的求學生涯，選擇了大學時修習專題時惠我良多的林永松老師當論文指導老師，追隨著他進入最佳化演算法的學習殿堂，兩年來在林老師相當具親和力的指導之下，終於能有所小成，完成這篇論文。尤其是在論文撰寫期間所遇到的各種困難，林老師總是不辭辛勞、不厭其煩地給予提示、解答，並在關鍵處指引我，這一路走來，雖然辛苦，但是靠著這盞明燈的指引，終於走向最後完成之路。所以首先應該要說感謝的人是我的恩師－林永松教授。

另外感謝各位口試委員：台科大電子系陳郁堂教授、台大資管系莊裕澤教授、世新大學資管系顏宏旭教授寶貴的指導與建議，讓本論文更加的完備。另外感謝同門佳宏、茂原、至鈞、麒安與克行，兩年來在學業上彼此互相切磋、討論，在生活也帶給我許多歡樂。尤其是在我外務纏身之時，給予我即時的協助，這份情誼我會銘記於心，同學的關係是短暫的，但朋友是一輩子的。另外也要感謝實驗室的學弟妹們，在我們忙於論文的時候分擔了實驗室許多例行的工作，尤其在口試時的幫忙，讓我們能全心於口試的準備。

另外，生育我、養育我的父母還有弟弟更是在背後永遠地支持我、給我鼓勵，讓我在遇到挫折時感受到親情的溫暖，能有勇氣再站起來，重新面對挑戰，才能突破所有的困難，完成這篇論文。

最後，謹以本論文向所有曾經幫助我的人致上最高的謝意，包括我的前後幾位室友兼國中同學們在我忙於論文時給我的鼓勵與生活上的幫助。也包括一位很好的朋友在平淡的論文寫作中為我帶來的變化與歡樂。

胡晉華　謹識

2003.7.23 於台北內湖

論文摘要

由於等效頻寬將各流量之間彼此影響的非線性關係轉換成一線性表示方法，大大降低了問題的複雜度，因此將等效頻寬的概念應用在網路容量限制考量之問題上，可以提供一個在研究允入控制及頻寬規劃分配上一個實用的架構。

在此篇論文中，我們首先提出了一個允入控制及路由規劃的通用性演算法，和一般允入控制的演算法不同的是，我們在決定是否允入新流量時除了考慮在目前路由架構下所剩餘的可用頻寬及服務品質管理的條件外，我們更進一步嘗試著調整目前的路由架構，將剩餘不連續的可用頻寬整合成連續可用的頻寬以供待允入之流量使用，如此能使得整體網路資源的使用最佳化，也等同於所獲利潤之最大化。但由於更動目前路由架構會造成流量的增加、資料封包的流失、增加現存網路使用者之傳輸延遲，因此本演算法的目標是在考慮以等效頻寬為條件的服務品質管理下，藉由最少（成本最低）的路由架構調整，來最大化系統的效能（收益）。

此外，我們也在以上述演算法為基礎下，提出了一個適用於群播網路的允入控制及路由規劃演算法。在群播網路上，調整路由架構變得較為複雜，群播樹的調整往往牽涉到群播樹建立時的條件限制，並且往往對網路產生更嚴重的影響，引發更大的成本，希望籍由此演算法的提出，可以有效地在儘量減少對現存網路使用者的影響下，最大化網路系統營運者的收益。

關鍵詞：等效頻寬、群播網路、允入控制、路由、網際網路通訊協定、網路規劃、最佳化、拉格蘭日鬆弛法、數學規劃。

THESIS ABSTRACT

GRADUATE INSTITUTE OF INFORMATION MANAGEMENT

NATIONAL TAIWAN UNIVERSITY

NAME：CHIN-HWA HU **MONTH/YEAR：JUL, 2003**

ADVISER： YEONG-SUNG LIN

Admission Control and QoS-constrained Routing Algorithms Considering Multicast Service and Traffic Rerouting

Because effective bandwidth transforms the complex non-linear influence between traffic flows into a linear expression, it reduces the complexity of the traffic model. The notion of effective bandwidths has provided an useful practical framework for call admission control and capacity planning problems.

In this thesis, we first propose a general call admission control and QoS-constrained routing algorithm. The difference between this proposed algorithm and other algorithms is that when we decide whether the new traffic flow is admitted, rerouting of existing traffic is considered. By moving some traffic from one link to another, it is possible to generate a new route that the QoS constraints, such like bandwidth, end-to-end delay will be satisfied for the new traffic flow. Utilization of the network, and revenue of network operators are maximized.

Besides, based on the above algorithm, we also propose a call admission and QoS-constrained routing algorithm that can be applied to networks supporting multicast services. In such networks, it is more complex to decide which set of traffic flows should be

rerouted to maximize network utilization. The constraints of building the multicast tree should also be considered when rerouting such set of traffic flows. Also, that will introduce more interference between traffic flows, and in turns costs more. By proposing this algorithm, we want to reduce the impact on the existing traffic flows as more as possible, and to maximize the total revenue of network operators at the same time.

Keywords: Effective Bandwidth, Multicast, Call Admission Control, Qos-Constrained Routing, Network Planning, Optimization, Lagrangean Relaxation Method, Mathematical Programming.

Contents

List of Tables

List of Figures

Chapter 1 Introduction

1.1 Background

With the popularity of the Internet, applications based on network service are growing rapidly. In order to meet the bandwidth requirements for those applications, network operators spend more and more to enlarge their network capacity, including setting up more new physical links, upgrading their existing links to higher transmission rate. In addition to enlarge the network capacity, because the Internet migrates to commercial enterprise, there is still one way to achieve that goal - network planning or called traffic engineering. Traffic engineering is the process of controlling how traffic flows through one's network in order to optimize resource utilization and network performance. At the mean time, it can provide Quality-of-Service (QoS). The ability to provide reliable QoS service may well become a crucial factor in influencing the customer's propensity to pay for networks. The current Internet operates in a best-effort manner, which is considered insufficient for QoS demanding applications. These applications, such as VoIP, VoD, MoD, video conferencing, Tele-Health require, at least benefit from QoS or some other form of prioritization guarantees on making successfully connection. To enable QoS and traffic engineering, admission control is needed, and plays an important role.

Those QoS demanding applications require a guaranteed level of QoS, to work properly. These QoS requirements may be in terms of a minimum bandwidth, bounded end-to-end delays, and the maximum packet loss rates suffered by a flow. Network operators that support such flows should guarantee such requirements in SLA (Service Level Agreement), and must be able to allocate and maintain their finite network resources to uphold their guarantees. Thus, the operators may also have to reject new traffic flows that would violate their promises. The process of deciding whether to accept or reject a new flow is called admission control. Nnetwork operators can also add some conditions when deciding whether to admit new traffic flows or not, even when the above requirements are already met. Operators may want to keep total throughput beyond a specified level, or preserve bandwidth for emergency, … , etc.

There are three basic components of admission control schemes: traffic descriptors, admission criteria, and measurement process. Figure 1.1 illustrates the relationship among the three components.

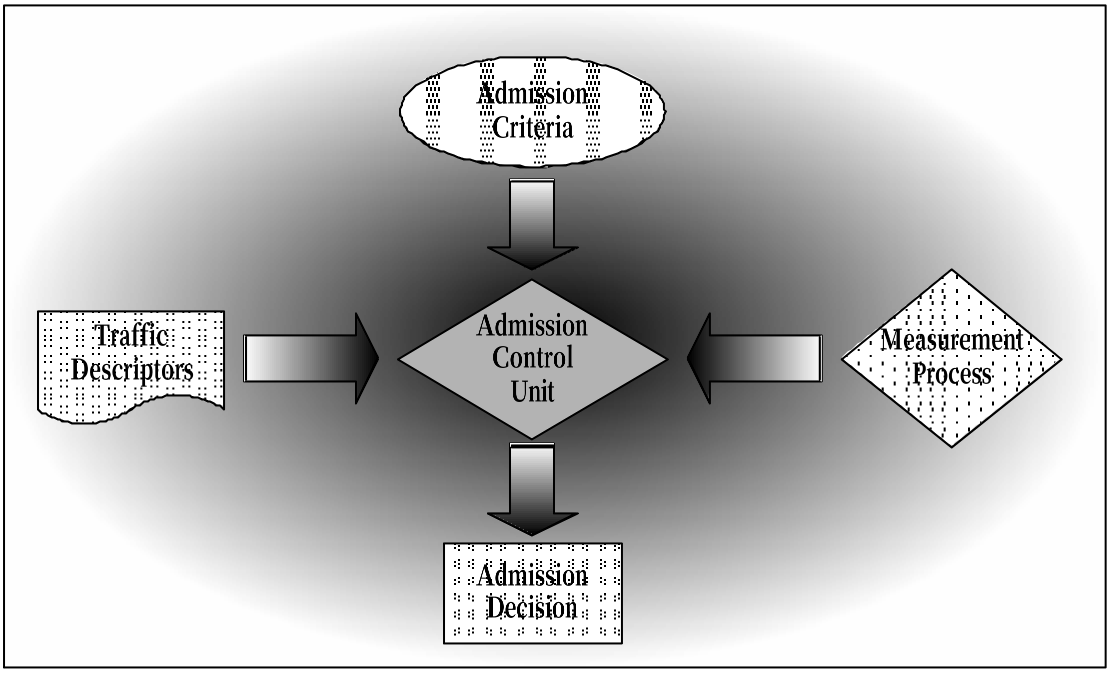

Figure 1.1 The relationship Between Basic Components of Admission Control

There are two basic approaches to admission control: the first, which we call the parameter-based approach, computes the amount of network resources required to support a set of flows given a priori flow characteristics; the second, the measurement-based approach, relies on measurement of actual traffic load in making admission decisions.

One of parameter-based approach, Simple-Sum algorithm, only ensure the sum of existing bandwidth requirements and the newly coming traffic flows doesn't exceed a specified threshold, for example, link capacity. The source behavior and the aggregate traffic arrival process are not considered. This is the simplest admission control algorithm and hence is being most widely implemented by switch and router vendors. Often, to ensure low queueing delay called for by controlled-load service, an approximation of the weighted fair queueing (WFQ) scheduling discipline is implemented with this admission control algorithm.

Examples of measurement-based approach are measured-sum, acceptance region [2] and efficient bandwidth. Not like "Simple Sum" algorithm just ensures the sum of existing reservations and the newly admitting traffic doesn't exceed network capacity, but "Measured Sum" algorithm uses measurement to estimate the load of existing traffic. "Acceptance Region" algorithm calculates an acceptance region that can maximize the reward of utilization against the penalty of packet loss. R. Guerin, H. Ahmadi, and M. Naghshineh give a comparison of above approaches [6]. Y.Y. Liu, P.H. Tu, and Z.F. Zhang give a simulation and analysis on measurement-based admission control algorithms [9].

To deal with QoS, the interference between each traffic flow should be considered as well. But through the systematic analysis from queuing theory, the interference is presented in non-linear mathematical equations forms. Its dynamic nature poses difficult traffic control problems when trying to achieve efficient use of network resources. This results in a complex traffic model when considering QoS constraints. Because all connections are statistically multiplexed at the physical layer and the bit rate of connections varies, a challenging problem is to characterize the effective bandwidth requirement of both individual connections and the aggregate bandwidth usage of connections multiplexed on a given link. At 1991, R. Guerin, H. Ahmadi, and M. Naghshineh proposed a concept of "equivalent bandwidth" [6]. The concept of equivalent bandwidth transforms the complex non-linear influence between traffic flows into a linear relationship, so it reduces the complexity of the traffic model. The notion of effective bandwidths has provided a useful practical framework for call admission control and capacity planning problem. Equivalent bandwidth will be discussed more detailed in section 1.3.

All of the above admission control algorithms decide whether to admit a new traffic based on residual resource which is computed from existing traffic and routing policy. In this thesis, a new approach of equivalent bandwidth measured-based admission control is proposed. The new algorithm is capable of rerouting some existing traffic from one link to another one. Thus, the utilization of the network could be maximized.

1.2 Motivation

Almost all admission control algorithms only consider the residual resources of the network based on the existing routing topology to decide whether to admit new traffic flows or not. But it is possible that the residual resource can fulfill the QoS requirements of the new admitting traffic flows. Because those links with residual resources are disconnected. If they are disconnected, there will be no continuous paths found for new traffic flows to route on. But, under this scenario, it doesn't mean that the residual resource is not sufficient to admit new traffic flows. If the network operators reject the new coming traffic flow, the network utilization is not optimized, and either the network is not maximized. We propose an algorithm trying to solve this problem. If we try to move some traffic from one link to another properly, we can have a new "continuous" path to meet the QoS requirements of new traffic flows, and, therefore, they could be admitted. At the same time, revenue is maximized.

But changing traffic flow from one link to another can also introduce interference between traffic flows, which in turns impacts other traffic flows in the network. Such interference could be, for example, packet-loss, increasing of transmission delay, …, etc. And yet another issue to be mentioned is the cost of re-routing process. All above do have serious impacts on the network. What is to be maximized is the total system throughput (revenue). At mean while, try to minimize the degree of impact on the existing traffic flows of the network.

In [8], the authors propose an admission control and routing algorithm for networks supporting PVC service. The objective of admission control is to maximize the system's throughput (revenue) subject to (1) the Quality of Service constraint for each user pair and (2) the constraint that the ratio of the total routing table modification cost and the corresponding revenue for admitting the new user(s) not exceed a given bound. In networks supporting PVC, connections are set up at the service subscription time by manually modifying the routing tables residing in the switches along the chosen paths via a network management system.

In this paper, a similar algorithm but supporting any network and with the ability to provide QoS service is proposed. In this model, whenever a new traffic flow comes, whether the residual network resource is sufficient to provide a QoS-guaranteed service for the new traffic flow or not is decided. If not, try to do some change on the routing topology to optimize the resource utilization and to see if the QoS constraints of new incoming traffic could be satisfied after the change. At the same time, those existing traffic flows should also meet its QoS constraints on the impact of admitting new traffic flow, such as end-to-end delay.

In addition, an interesting scenario is considered: networks supporting multicast. In such scenario, the cost of link changing should be considered group-wised. At the mean time, the integrity of the multicast group should be ensured.

The objective is also the maximization of whole system's throughput (revenue) subject to (1) the QoS constraint for each user pair and (2) to minimize the total cost of changing routing topology due to admit a new traffic flow.

1.3 Literature Survey

1.3.1 Equivalent Bandwidth

In high-speed network architectures, several classes of traffic streams with widely varying traffic characteristics are statistically multiplexed and share common switching and transmission resources. Because all connections are statistically multiplexed at the physical layer and the bit rate of connections varies, a challenging problem is to characterize the effective bandwidth requirement on a given link. Admission control depends much on the characterization to decide if and how to accept incoming traffic flows. The equivalent capacity of a set of connections multiplexed on a link is defined as the amount of bandwidth required to achieve a desired GOS, e.g., buffer overflow probability, maximum end-to-end delay, given the offered aggregate bit rate generated by the connections. It is a function of individual connection characteristics and available network resources such as buffers or bandwidth. The goal of equivalent capacity is to capture key connection parameters that influence bandwidth allocation. Therefore, the equivalent capacity computation focuses on the bandwidth requirement of the bit rate generated by sources. The types of "sources" include both individual users as well as more complex sources, such as output of a multiplexer. R. Guerin, H. Ahmadi, and M. Naghshineh, propose a computationally simply approximation for the equivalent capacity or bandwidth requirement of a single or multiplexed connections on the basis of buffer overflow probability [6].

The equivalent capacity is computed from the combination of two different approaches, one based on a fluid-flow model and the other on the approximation of the stationary bit rate distribution. These two approaches capture different aspect of behavior of multiplexed connection, while remaining computationally simple. The aspect is significant for admission control, because "if" and "how" to handle new incoming traffic flows is to be decided. And because of the simplicity of computation, it allows for the real-time computation of admission decision-making.

Fluid-Flow Approximation

The first approximation for the equivalent capacity is based on a fluid-flow mode. In this model, the bit rate generated by a number of multiplexed connections is represented as a continuous flow of bits with intensity varying according to the state of an underlying continuous-time Markov chain. This Markov chain is obtained from the superposition of the sources associated width each connection. A two-state Markov source is characterized by it peak rate R_{peak}, utilization r, and mean burst period b. The aggregate bit rate is directed to a buffer which is emptied at a constant rate c. What we want to calculate is the minimum value of c, expressed as $\hat{C}$, that, for a given buffer size x, ensures a buffer overflow probability smaller than e. The value $\hat{C}$ is the equivalent capacity of the multiplexed connections. Please reference [6] for detailed computation of equivalent capacity. Only the results are listed here.

Single Source:

$$\hat{C} \simeq \frac{a b(1-r)R_{peak} - x + \sqrt{\left[a b(1-r)R_{peak} - x\right]^2 + 4xa b r(1-r)R_{peak}}}{2a b(1-r)} \qquad (1)$$

where $a = \ln\left(1/e\right)$

Multiple Sources:

$$\hat{C}_{(F)} = \sum_{i=1}^{N} \hat{c}_i \qquad (2)$$

where $\hat{c}_i$ are determined from (1), N is the number of multiplexed sources

Stationary Approximation

$$\hat{C}_{(s)} \simeq m + a's \quad \text{with} \quad a' = \sqrt{-2\ln(e) - \ln(2p)} \qquad (3)$$

where m is the mean aggregate bit rate ($m = \sum_{i=1}^{N} m_i$), and s is the standard deviation of the aggregate bit rate ($s^2 = \sum_{i=1}^{N} s_i^2$)

Now, combine the two approximatio ns into a single expression. The equivalent capacity $\hat{C}$ is taken to be the minimum of $\hat{C}_{(F)}$ and $\hat{C}_{(S)}$:

$$\hat{C} = \min\left\{ m + a's , \sum_{i=1}^{N} \hat{c}_i \right\} \qquad (4)$$

Y.Y. Liu, P.H. Tu, and Z.F. Zhang develop the delay probability distribution function (PDF) and the approximate expression for the equivalent bandwidth using fluid flow method[9]. Their study is on packet delay in buffers, and First Come First Serve principle is adopted. In the paper, the traffic model is also two-state Markov process with ON state and OFF state duration are exponential distributed with parameter b and a respectively. When in ON state, the source generate packets at peek rate l . e is defined as the probability that the packet delay in buffer exceeds a given threshold D. Again, the detailed description of computation is skipped, only the result is listed. The equivalent bandwidth under delay constrains, as the given D and e , is as below:

$$C_d = \frac{N l \left(N a D - \ln e \right)}{N \left(a + b \right) D - \ln e} \qquad (5)$$

where N, still, is the number of traffic multiplexed on the link

1.3.2 Quality of Service Routing

The Internet has recently become an important communications channel [11]. The Internet was used in the 1980s and the beginning of 1990s by research and education communities for computer data transmission: electronic mail, network news and file transfers. The most demanding application from the service quality point of view was a network remote logon as an interactive application. The bandwidth required was small and occasional delay variations of order of several seconds could be tolerated [11].

Routing deployed in today's Internet is focused on connectivity and typically supports only one type of datagram service called "best effort" [14]. That means it will try its best to forward user traffic, but can provide no guarantees regarding loss rate, bandwidth, delay, delay jitter, etc. For example, packets can be dropped indiscriminately in the event of congestion. This kind of service works fine for some traditional applications (such as FTP and email). Recently many interactive or real-time services have been introduced and at the same time the economical importance of the Internet has grown. Transmitting interactive real-time media is the greatest challenge in packet-based networks, such as IP networks. The end-to-end delay, the delay variations (jitter), and the packet loss must not exceed some limits or usability of the service degrades badly [10]. It's intolerable for newly emerged real-time, multimedia applications, which require high bandwidth, low delay, and low delay jitter. In other words, these new applications require better transmission services than "best-effort". Thus, the study of Quality-of-Service (QoS) is very important nowadays.

Current Internet routing protocol [2], e.g. OSPF, RIP, use "shortest path routing" which is optimized for a single arbitrary metric, administrative weight or hop count. Alternate paths with acceptable but non-optimal cost can't be used to route traffic. QoS-based routing must extend the current routing paradigm in three basic ways. First, to support traffic using integrated-services class of services, multiple paths between node pairs will have to be calculated. Such calculation requires the distribution of routing metrics, such like delay and available bandwidth. If the metrics change frequently, routing updates become more frequently and consuming more network bandwidth and router CPU cycles.

Second, today's opportunistic routing will shift traffic to a "better" path as soon as it is found even if the service requirement is satisfied. Such rerouting can introduce routing oscillations as traffic shifts back and forth between alternate paths. Furthermore, delay variation and jitter experienced by end users are increased.

Third, as mentioned earlier, today's optimal path routing algorithms do not support alternate routing. If the best existing path cannot admit a new flow, the associated traffic cannot be forwarded even if an adequate alternate path exists.

The objectives of QoS-based routing are:

- **Dynamic determination of feasible paths**: QoS-based routing is supposed to dynamically, not find being configured statically, a path to satisfy end user's requirements. If there are several feasible paths available, the selection is based on some policy constraints, like minimum cost.

- **Optimization of resource usage**: QoS-based routing is expected to direct network traffic in an efficient way that can maximize the total network throughput. Such a routing scheme can be the basis for efficient network engineering.
- **Graceful performance degradation**: When network is in heavy load, QoS-based routing is expected to give better performance (e.g. better throughput) than best-effort routing, which can degrade the performance dramatically.

The followings are some traffic handling mechanisms:

802.1p: 802.1p is a traffic-handling mechanism for supporting QoS in IEEE 802 technology LANs. 802.1p defines a field in the layer-2 header of 802 packets that can carry one of eight priority values. Typically, hosts or routers sending traffic into a LAN will mark each transmitted packet with the appropriate priority value. LAN devices, such as switches, bridges and hubs, are expected to treat the packets accordingly (by making use of underlying queuing mechanisms). The scope of the 802.1p priority mark is limited to the LAN. Once packets are carried off the LAN, through a layer-3 device, the 802.1p priority is removed.

Differentiated Services (Diffserv): Diffserv is a layer-3 QoS mechanism. defines a field in the layer-3 header of IP packets, called the diffserv codepoint (DSCP). Typically, hosts or routers sending traffic into a diffserv network will mark each transmitted packet with the appropriate DSCP. The DSCP is a six-bit field, spanning the fields formerly known as the type-of-service (TOS) fields and the IP precedence fields. Routers within the diffserv network use the DSCP to classify packets and apply specific queuing or scheduling behavior (known as a per-hop behavior or PHB) based on the results of the classification.

Integrated Services (Intserv): Intserv is a service framework. there are two services defined within this framework, guaranteed service and the controlled load service. The guaranteed service promises to carry a certain traffic volume with a quantifiable, bounded latency. The controlled load service agrees to carry a certain traffic volume with the 'appearance of a lightly loaded network'. These are quantifiable services in the sense that they are defined to provide quantifiable QoS to a specific quantity of traffic.

1.3.3 Traffic Engineering

Traffic engineering is the process of controlling how traffic flows through one's network in order to optimize resource utilization and network performance. Traffic engineering is needed in Internet initially because current interior gateway protocols (IGPs) always use the shortest paths to forward traffic. Using shortest paths conserves network resources, but may also cause the following problems.

The shortest paths from different sources overlap at some links, causing congestion on those links. The traffic from a source to a destination exceeds the capacity of the shortest path, while a longer path between these two routers is underutilized.

There is a debate on whether network capacity will one day become so cheap and abundant that these two problems will be eliminated. This debate is beyond the scope of this article. Here we simply note that currently all ISPs have the above problems. By performing traffic engineering in their networks, ISPs can greatly optimize resource utilization and network performance. Revenue can be increased without large investments in upgrading network infrastructure. Therefore, traffic engineering is definitely useful for ISPs now.

Traffic engineering is difficult to do with IGPs in large networks for the following reasons:

Among the equal-cost multipaths from a source, every path will have an equal share of load. This equal ratio cannot be changed. Therefore, one of the paths may end up carrying significantly more traffic than other paths because it also carries traffic from other sources. Local sharing cannot be done among multiple paths of different costs. Modifying an IGP metric to trigger some traffic shift tends to have side effects, and undesirable traffic shifts may also be triggered.

In order to do traffic engineering effectively, the Internet Engineering Task Force (IETF) introduced MPLS (discussed later), constraint-based routing, and an enhanced link state IGP.

1.4 Proposed Approach

We model the problems as nonlinear convex integer mathematical programming problems. We will develop heuristics and apply the Lagrangean relaxation method to solve the problems. In Lagrangian Relaxation, subgradient method is used to find extreme point.

Chapter 2 Problem Formulation

2.1 Problem Description

The problem to be solved is that, how to decide a minimum cost path set so that network operators could admit new incoming traffic flows as many as possible to maximize total operating revenue. Two models are defined. One is for unicast networks, and the other one is for networks supporting multicast services.

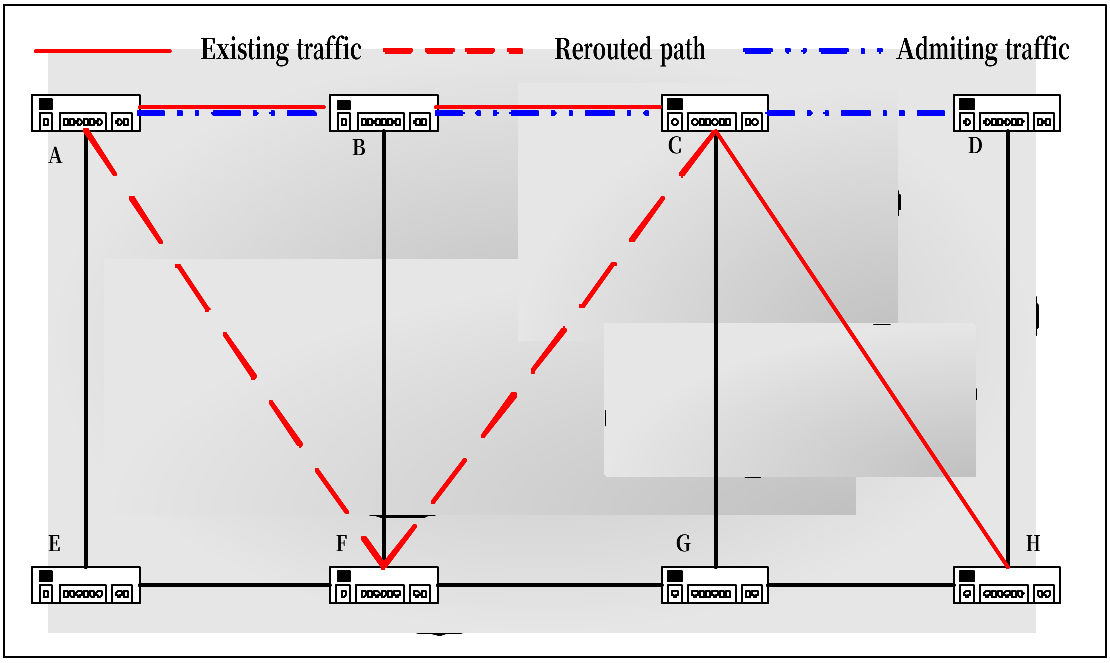

Figure 2.1 Reroute existing traffic flows to Admit New Traffic

Take figure 2.1 for example, the existing traffic is routed on path (A,B,C,H). Here comes a new traffic on O-D pair (A,D). Supposing this traffic has a high bandwidth requirement, and only link (A,B) and link (B,C) have sufficient bandwidth. But now, part of their bandwidth is used to transmit existing traffic flows, the residual bandwidth is not enough for the incoming traffic flow. In ordinary admission algorithms, the new traffic would be rejected. But, in fact, residual bandwidth is indeed enough to admit the new traffic. In our proposed algorithm, we will try to reroute the existing traffic to a new path (A,F,C,H), so that all bandwidth of (A,B) and (B,C) is released for the new traffic to route on. Thus, the new traffic would be admitted. Not only system throughput (utilization) but also operating revenue is increased.

In this algorithm, some issues should be discussed more detailed:

- How to decide the cost of rerouting paths?
- What are the constraints of rerouting?
- How are the new incoming traffics routed?

The cost of the rerouted paths is calculated using transmission delay, hop counts of the path, the amount of traffic flow routed on the path, and a "rank coefficient" of the path. At first, transmission delay is discussed. When a path is rerouted, network operators should temporarily stop to transmit all traffic flows on the path, after a period of time sufficient for the traffic to transmit to destination, and then restart to transmit data on the new path. The reason why stop first is to drain out all traffic flows on the old path to ensure the packets are not out-of-order due to path rerouting. So, delay is incurred. But, how to calculate the delay is proper to this model? There are two different ways to calculate that. From sender's point of view, the delay it suffers is just equal to the end-to-end delay of the originally path. On the other hand, from the receiver's point of view, the delay is equal to the end-to-end delay of the new path. Here, we choose the first scenario. This decision means that the longer the end-to-end delay of one path is, the less likely the path is rerouted.

Second, intuitively, the traffic on a path will suffer a longer delay if hop counts of the path are larger. Third, the amount of traffic flow of on each path is an important factor in calculating rerouting cost. More traffic flows will incur longer delay. Besides, larger buffer size is needed to store the traffic received at original router of the path.

The last factor is the "rank coefficient". In recent years, applications based on Internet vary one from another. Each application has its own QoS requirements. So, we proposed such a coefficient when calculating rerouting cost. It is an indicator that represents the importance of a traffic flow. Based on the QoS requirements in SLA, network operators may set different values for different traffic flow. With a larges rank coefficient, a traffic flow is less likely to be rerouted. For example, for a real-time application, like VoD, or video-conference, network operators can set a higher value than those paths with ordinary e-mail traffics.

In conclusion, the cost of rerouting a traffic flow will be the end-to-end delay multiplied by the amount of traffic flows, and then, its rank coefficient. The hop count already is included when calculating end-to-end delay, so it is not multiplied again.

In this problem, the equivalent bandwidth is used as a basis of QoS constraints measurement. How to calculate equivalent bandwidth is describe in section 1.3. Each traffic flow has its own end-to-end delay requirements. When a traffic flow is rerouted, the end-to-end delay on the new path should be also less than this specified value.

As to the routing of the new traffic flows, non-linear programming skill is adopted. The above constraints are well formulated as a set of mathematical expressions. The objective function is to maximize the total revenue. New traffic flows are admitted as many as possible, and after admitting new traffics, try to find a rerouting policy that has a less cost. The objective function is the maximum of revenue minus the cost of rerouting existing traffic flows. And then, a mathematical approach called Lagrangean Relaxation Method is used to solve this problem.

In multicast scenario, the path delay constraint is replaced by group-wised delay constraint. In addition to those constraints in unicast scenario, yet another set of constraints should be satisfied: the integrity constraints of a multicast group. That is, all destinations of the multicast group have to find to routing path, or the multicast group can not be admitted.

2.2 Problem Notations and Formulation

2.2.1 Model 1: Unicast

Notation

Given Parameters	
Notation	**Descriptions**
L	Set of links in the network
C_l	For each link $l \in L$, the link capacity
$W^{'}$	Set of existing O-D pairs
$W^{''}$	Set of new O-D pairs whose admittance into network is to be determined
W	$W = W^{'} \cup W^{''}$
g_w	Equivalent bandwidth of O-D pair $w \in W$
l_w	Mean traffic requirement of O-D pair $w \in W$
a_w	Revenue from admitting O-D pair $w \in W$ into the network
P_w	Set of elementary directed paths in the network
$p_w^{'}$	Artificial path introduced to carry the rejected sessions
$P_w^{'}$	$P_w^{'} = P_w \cup \{p_w^{'}\}$
x_p'	1 if path $p \in P_w^{'}$ is used to transmit the packets for O-D pair w on original routing decision Otherwise, 0
d_{pl}	A function which returns 1 when link l is on path p Otherwise, 0

Q_w	Maximum allowed delay time for the O-D pair w in SLA
H_w	Maximum allowed hop counts for the OD-pair w in SLA
R_w	Rank coefficient of O-D pair w
C	A transformation coefficient from rerouting cost to revenue

Table 2-1 Notations of Given Parameters

Decision Variables	
Notation	**Descriptions**
x_p	1 if path $p \in P_w^{'}$ is used to transmit the packet for O-D pair w Otherwise, 0

Table 2-2 Notations of Decision Variables

Formulation

Objective function:

$$Z_{IP1} \quad = \quad \max \sum_{w \in W''} \sum_{p \in P_w} a_w x_p - C \sum_{w \in W'} \sum_{p \in P_w} x'_p (1 - x_p)(t_w + 3s_w)\, l_w R_w \qquad \text{(IP1)}$$

As we want to deal with a minimization problem, an equivalent expression is:

$$Z_{IP2} \quad = \quad \min \sum_{w \in W'} \sum_{p \in P_w} -a_w x_p + C \sum_{w \in W'} \sum_{p \in P_w} x'_p (1 - x_p)(t_w + 3s_w)\, l_w R_w \qquad \text{(IP2)}$$

subject to:

$$\sum_{w \in W} \sum_{p \in P_w} x_p g_w d_{pl} \le C_l \qquad \forall l \in L \qquad (1)$$

$$\sum_{p \in P_w^{'}} x_p = 1 \qquad \forall w \in W^{"} \qquad (2)$$

$$\sum_{p \in P_w} x_p = 1 \qquad \forall w \in W' \qquad (3)$$

$$x_p = 0 \text{ or } 1 \qquad \forall p \in P'_w, w \in W \qquad (4)$$

$$l_w \le g_w \qquad \forall w \in W \qquad (5)$$

$$\sum_{l \in L} \sum_{p \in P_w} x_p d_{pl} \le H_w \qquad \forall w \in W \qquad (6)$$

$$t_w = \sum_{l \in L} \frac{x'_p d_{pl}}{g_w - l_w} \qquad \forall p \in P_{w,} \forall w \in W' \qquad (7)$$

$$s_w = \sqrt{\sum_{l \in L} \frac{x'_p d_{pl}}{(g_w - l_w)^2}} \qquad \forall p \in P_{w,} \forall w \in W'. \qquad (8)$$

The objective function is to maximize total revenue from admitting a new traffic flow. The first item is the total revenue received from customers. The second item is the rerouting cost. In the second item, $t_w + 3s_w$ is a reasonable upper bound for end-to-end delay. Considering the set $(x'_p, x_p), x'_p(1 - x_p)$ returns 1 only when the value of the set is (1,0), otherwise, it will return 0. This means only when the path p is rerouted, the rerouting cost will be calculated. The term $x'_p(1 - x_p)d_{pl}$ ensures only the links on those paths which are rerouted are summed over. $\sum_{l \in L} x'_p(1 - x_p)d_{pl}(t_w + 3s_w)$ is the total end-to-end delay of the path.

Constraint (1) requires that the aggregate equivalent bandwidth does not exceed the capacity of each link. Constraints (2), (3) and (4) require that all of the traffic between any O-D pair must be transmitted over exactly one routing path. Constraint (2) requires the new O-D pairs should choose one path from the path set including physical paths and artificial paths. When artificial path is selected, the O-D pair is rejected. Constraint (3) requires the existing O-D pairs can only choose one path from physical paths. No originally admitted traffic should be rejected after admitting new traffic. Constraint (6) is the hop-count constraint. Constraint (7) and (8) are the end-to-end delay and standard deviation derived from an *M/M/1* queueing model respectively.

About the equivalent bandwidth g_w, we can obtain it from H_w. To explain that, an *M/M/1* queueing model is used.[1] On each link, delay can be expressed as $1/(g_w - l_w)$. The hop count is H_w, end-to-end delay would be $\frac{H_w}{g_w - l_w}$. Hence, an expression is hold:

$$\frac{H_w}{g_w - l_w} \le Q_w \qquad (2.1)$$

Thus, g_w an be set to its minimum value, which is: $l_w + (H_w / Q_w)$.

2.2.2 Model 2: Multicast

Notation

Given Parameters	
Notation	**Descriptions**
L	Set of links in the network
C_l	For each link $l \in L$, the link capacity
G'	Set of existing groups
G''	Set of new groups whose admittance to be determined

[1] Though *M/M/1* model can't fully describe real traffic on the Internet, for illustration purpose, M/M/1 model can provide an acceptable analysis result.

G	$\{G' \cup G''\}$
D_g	Set of destinations of the multicast group g, $g \in G$
h_g	The minimum number of hops to the farthest destination node in multicast group g.
g_g	Equivalent bandwidth of multicast group g, $g \in G$
l_g	Mean traffic requirement of multicast group g, $g \in G$
a_g	Revenue of admitting multicast group g into the network, $g \in G$
P_{gd}	Set of elementary directed paths in the network, $g \in G$
p'_{gd}	Artificial path introduced to carry the rejected session, $g \in G$
P'_{gd}	$P'_{gd} = P_{gd} \cup \{p'_{gd}\}$, $g \in G$
x'_{pgd}	1 if path $p \in P'_{gd}$ is used to transmit the traffic for multicast group g on original routing topology, and 0 otherwise
d_{pl}	A function which returns 1 when link l is on path p, and 0, otherwise
y'_{gl}	1 if multicast group g is routed on link l on the original routing topology, and 0 otherwise
Q_g	Maximum allowed delay time for the multicast group in SLA
H_g	Maximum allowed hop counts for the multicast group g in SLA
R_g	Rank coefficient of multicast group g
C	A transformation coefficient from rerouting cost to revenue

Table 2-3 Notation of Given Parameters

Decision Variables	
Notation	**Descriptions**
x_{pgd}	1 if path $p \in P_{gd}^{'}$ is used to transmit the traffic for multicast group *g* destined for destination d, and 0 otherwise.
y_{gl}	1 if multicast group *g* is routed on link $l \in L$, $g \in G$, and 0 otherwise
z_g	1 if multicast group *g* is admitted, and 0 otherwise
S_g	The maximum transmission delay of a multicast group *g* from root to all destinations

Table 2-4 Notation of Decision Variables

Objective function:

$$Z_{IP3} = \max \sum_{g \in G''} a_g z_g - C \sum_{g \in G'} S_g l_g R_g \quad \text{(IP3)}$$

As we want to deal with a minimization problem, an equivalent expression is :

$$Z_{IP4} = \min \sum_{g \in G''} -a_g z_g + C \sum_{g \in G'} S_g l_g R_g \quad \text{(IP4)}$$

subject to:

$$\sum_{g \in G} g_g y_{gl} \le C_l \qquad \forall l \in L \quad (1)$$

$$\sum_{p \in P_{gd}} x_{pgd} = 1 \qquad \forall d \in D_g, \forall g \in G'' \quad (2)$$

$$\sum_{p \in P'_{gd}} x_{pgd} = 1 \qquad \forall d \in D_g, \forall g \in G' \qquad (3)$$

$$x_{pgd} = 0 \text{ or } 1 \qquad (4)$$

$$l_g \leq g_g \qquad \forall g \in G \qquad (5)$$

$$\sum_{l \in L} \sum_{p \in P_{gd}} x_{pgd} d_{pl} \leq H_g \qquad \forall d \in D_g, \forall g \in G \qquad (6)$$

$$\sum_{d \in D_g} \sum_{p \in P_{gd}} x_{pgd} d_{pl} \leq y_{gl} \left|D_g\right| \qquad \forall g \in G, l \in L \qquad (7)$$

$$y_{gl} = 0 \text{ or } 1 \qquad \forall g \in G \qquad (8)$$

$$\sum_{l \in L} y_{gl} \geq \max\{h_g, \left|D_g\right|\} \qquad \forall g \in G \qquad (9)$$

$$z_g = \sum_{p \in P'_{gd}} x_{pgd} \qquad \forall d \in D_g, \forall g \in G^{"} \qquad (10)$$

$$z_g = 0 \text{ or } 1 \qquad (11)$$

$$t_{gd} = \sum_{l \in L} \frac{x'_{pgd} d_{pl}}{g_g - l_g} \qquad \forall p \in Pgd, \; \forall d \in Dg, \forall g \in G \qquad (12)$$

$$s_{gd} = \sqrt{\sum_{l \in L} \frac{x'_{pgd} d_{pl}}{\left(g_g - l_g\right)^2}} \qquad \forall p \in Pgd, \; \forall d \in Dg, \forall g \in G \qquad (13)$$

$$S_g \geq \sum_{l \in L} \sum_{p \in P'_{gd}} x'_p (1 - x_p)\left(t_{gd} + 3 s_{gd}\right) \qquad \forall d \in D_g, \forall g \in G' \qquad (14)$$

$$S_g \leq Q_g \qquad \forall g \in G'. \qquad (15)$$

Constraint (1) requires that the aggregate equivalent bandwidth does not exceed the capacity of each link. Constraints (2), (3) and (4) require that all of the traffic between any S-D (Source-Destination) pair must be transmitted over exactly one path. Constraint (2) requires the new groups should choose paths from the path set including physical paths and artificial paths. Constraint (3) requires the existing groups can only choose paths from physical paths. No originally admitted groups should be rejected after admitting new groups.

Constraint (5) requires the calculated equivalent bandwidth is larger than the mean traffic requirement of the multicast group. Constraint (6) is the hop-count constraint. Constraint (7) and (8) are referred to as the tree constraints. Constraint (8) and (9) require that for each multicast group, the number of links used should be more than the larger value of the height of the routing tree of the group and the number of destinations of that group. Constraint (10) and (11) referred as the integrity constraint They require that in a multicast group, for each destination, a path is selected only when the multicast group is admitted. Constraint (12) describes the mean delay on the path. Constraint (13) describes the standard deviation of the delay on the path. Constraint (14) requires that the decision variable S_g should be larger than the maximum transmission delay of every S-D pair for each group. Constraint (15) requires that the maximum transmission delay can not be larger than the given delay in SLA, Q_g.

Chapter 3 Solution Approach

3.1 Lagrangean Relaxation Method

In the 1970s [2], Lagrangean relaxation methods were used in scheduling and solving general integer programming problems. Lagrangean relaxation can provide proper solutions for those problems. It is a flexible solution approach. In fact, it has become one of the best tools for solving optimization problems such as integer programming, linear programming combinatorial optimization, and non-linear programming. Lagrangean relaxation has several advantages, for example, Lagrangean relaxation could decompose complex mathematical models in many different ways into some stand-alone subproblems. Then, we can optimally solve the subproblems using any proper algorithm [2][4].

Lagrangean relaxation lets us to find out the boundary of our objective function, so we can use it to implement heuristic solutions for getting feasible solutions. Lagrangean relaxation is a flexible solution strategy that permits modelers to exploit the underlying structure in any optimization problem by relaxing complicating constraints. This method permits us to "pull apart" models by removing constraints and place them in the objective function with associated Lagrangean multipliers. The optimal value of the relaxed problem is always a lower bound (for minimization problems) on the objective function value of the problem. To obtain the best lower bound, we need to choose a minimization multiplier so that the optimal value of the Lagrangean subproblem is as large as possible. We can solve the Lagrangean multiplier problem in a variety of ways. The subgradient optimization technique is the most popular technique for solving Lagrangean multipliers problems [2][4].

Figure 3.1 explains Lagrangean relaxation in a straightforward way. Figure 3.2 gives a detailed procedure for Lagrangean relaxation.

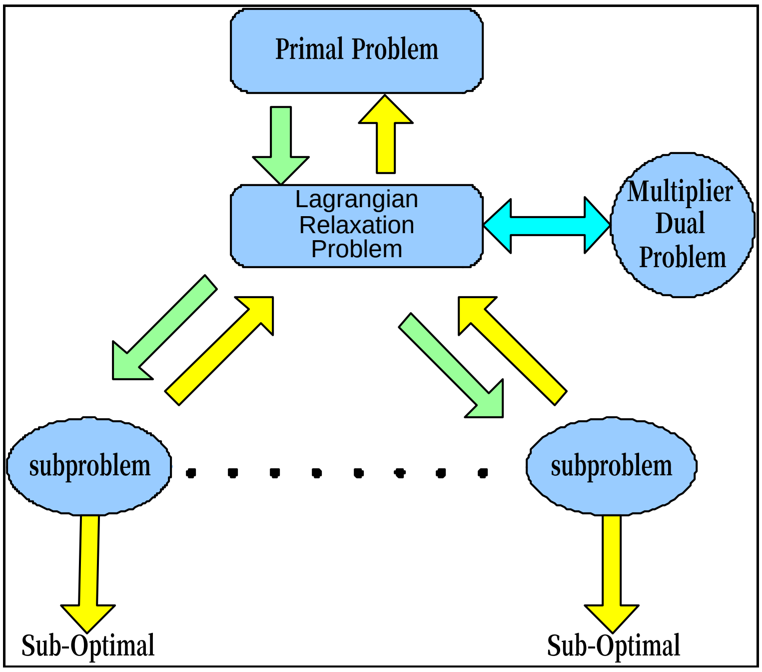

Figure 3.1 Lagrangean Relaxation Illustration

Figure 3.2 Lagrangean Relaxation Procedures

3.2 Model 1 : the Unicast Model

3.2.1 Solution Approach

By using the Lagrangean Relaxation method, we can transform the primal problem (IP2) into the following Lagrangean Relaxation problem (LR1) where Constraint (1) is relaxed.

3.2.2 Lagrangean Relaxation

For a non-negative Lagrangean multiplier, a Lagrangean Relaxation problem of (IP2) is given by

Optimization problem (LR1):

$$Z_{D1}(u) = \min \sum_{w\in W^{"}} \sum_{p\in P_w} -a_w x_p + C \sum_{w\in W^{'}} \sum_{p\in P_w} \sum_{l\in L} x_p^{'} (1-x_p)(t_w + 3s_w)\, l_w R_w + \sum_{l\in L} u_l \Big(\sum_{w\in W} \sum_{p\in P_w} x_p g_w d_{pl} - c_l\Big) \qquad \text{(LR1)}$$

subject to:

$\sum_{p\in P_w^{'}} x_p$	$=$	1	$\forall w\in W^{"}$	(1)
$\sum_{p\in P_w} x_p$	$=$	1	$\forall w\in W^{'}$	(2)
x_p	$=$	0 or 1	$\forall p\in P_w', w\in W$	(3)
l_w	$\le$	g_w	$\forall w\in W$	(4)
$\sum_{l\in L}\sum_{p\in P_w} x_p d_{pl}$	$\le$	H_w	$\forall w\in W$	(5)
t_w	$=$	$\sum_{l\in L} \frac{x_p' d_{pl}}{g_w - l_w}$	$\forall p\in P_{w,} \forall w \in W^{'}$	(6)
s_w	$=$	$\sqrt{\sum_{l\in L} \frac{x_p' d_{pl}}{(g_w - l_w)^2}}$	$\forall p\in P_{w,} \forall w \in W^{'}$.	(7)

where u_l are Lagrangean multipliers and are ≥ 0. This problem can be decomposed to two subproblems.

Optimization problem (LR1.1.1):

$$Z_{Sub_{1.1.1}}(u_l) = \min \sum_{l\in L} \sum_{w\in W^{''}} \sum_{p\in P_w} (u_l g_w d_{pl} - a_w) x_p \qquad \text{(LR 1.1)}$$

subject to :

$$\sum_{p\in P_w^{'}} x_p = 1 \qquad \forall w \in W^{''} \qquad (1)$$

$$x_p = 0 \text{ or } 1 \qquad \forall p \in P_w', w \in W \qquad (2)$$

$$\sum_{l\in L} \sum_{p\in P_w} x_p d_{pl} \leq H_w \qquad \forall w \in W\ . \qquad (3)$$

This subproblem is related to to-be-admitted O-D pairs and can be decomposed to $|w|$ subproblems. Each subproblem is a shortest path problem with hop constraint. $u_l g_w$ is the link cost, and H_w is the hop constraint. It can be solved by Bellman-Ford algorithm. If $u_l g_w - a_w \leq 0$, then set $\sum_{p\in P_w} x_p$ to be 1, otherwise, 0.

Optimization problem (LR1.1.2):

$$Z_{Sub_{1.1.2}}(u_l) = \min \sum_{w\in W} \sum_{p\in P_w} \sum_{l\in L} [u_l g_w d_{pl} - C x_p^{'} (t_w + 3s_w) l_w R_w] x_p \qquad \text{(LR 1.2)}$$

subject to :

$$\sum_{p\in P_w} x_p = 1 \qquad \forall w \in W^{'} \qquad (1)$$

$$x_p = 0 \text{ or } 1 \qquad \forall p \in P_w', w \in W \qquad (2)$$

$$\sum_{l\in L} \sum_{p\in P_w} x_p d_{pl} \leq H_w \qquad \forall w \in W \qquad (3)$$

$$t_w = \sum_{l\in L} \frac{x_p' d_{pl}}{g_w - l_w} \qquad \forall p \in P_{w,} \forall w \in W^{'} \qquad (4)$$

$$s_w = \sqrt{\sum_{l \in L} \frac{x'_p d_{pl}}{(g_w - l_w)^2}} \qquad \forall p \in P_{w,} \forall w \in W'. \qquad (5)$$

This subproblem is related to existing O-D pairs, and can be decomposed to $|w|$ subproblems. Each subproblem is a shortest path problem with hop constraint. $u_l g_w$ is the link cost, and H_w is the hop constraint. Each subproblem can be solved by Bellman-Ford algorithm. The path discovered by Bellman-Ford algorithm could be the same as the original path, or different from the original path. Here, we should compare the two calculating result for both situation, and take the smaller one. The reason is that if rerouting the O-D pair will introduce larger value of $Z_{D1}(u)$, we should let the O-D pair use the original path to maximize total revenue.

3.2.3 The Dual Problem and the Subgradient Method

According to the weak Lagrangean duality theorem, for any $u \geq 0$, $Z_{D1}(u)$ is a lower bound of $IP2$. The following dual problem (D1) is then constructed to calculate the tightest lower bound.

Dual Problem (D1)

$$Z_{d1} = \max Z_{D1}(u) \qquad (D1)$$

Subject to:

$$u_l \geq 0.$$

The most popular method to solve the dual problem is the subgradient method. Let g be a subgradient of $Z_{D1}(u)$. Then, in iteration k of the subgradient optimization procedure, the multiplier vector $p = (u)$ is updated by $p^{k+1} = p^k + t^k g^k$. The step size t^k is determined by $t^k = d \frac{Z_{IP2}^h - Z_{D1}(p_k)}{\|g^k\|^2}$. Z_{IP2}^h is the primal objective function value for a heuristic solution. d is a constant between 0 and 2.

3.3 Model 2 : the Multicast Model

3.3.1 Solution Approach

By using the Lagrangean Relaxation method, we can transform the primal problem (IP4) into the following Lagrangean Relaxation problem (LR2) where Constraints (1)、(7)、(10)、(14) are relaxed.

3.3.2 Lagrangean Relaxation

For a vector of non-negative Lagrangean multipliers, a Lagrangean Relaxation problem of (IP2) is given by

Optimization problem (LR2):

$$
\begin{aligned}
Z_{D2}(a,u,v,w) = \min \sum_{g\in G^{"}} -a_g z_g + C\sum_{g\in G} S_g\, l_g R_g + \sum_{l\in L} a_l \Big(\sum_{g\in G} g_g\, y_{gl} - c_l\Big) + \\
\sum_{l\in L}\sum_{g\in G} u_{gl}\Big(\sum_{d\in D_g}\sum_{p\in P_{gd}} x_{pgd} d_{pl} - y_{gl}\,|D_g|\Big) + \\
\sum_{g\in G^{"}}\sum_{d\in D_g} v_{gd}\Big(z_g - \sum_{p\in P'_{gd}} x_{pgd}\Big) + \\
\sum_{g\in G}\sum_{d\in D_g} w_{gd}\Big(\sum_{p\in P_{gd}} x'_p (1-x_p)(t_{gd} + 3s_{gd}) - S_g\Big)
\end{aligned}
\qquad \text{(LR 2)}
$$

subject to :

$\sum_{p\in P_{gd}} x_{pgd}$	=	1	$\forall d \in D_g, \forall g \in G^{"}$	(1)
$\sum_{p\in P'_{gd}} x_{pgd}$	=	1	$\forall d \in D_g, \forall g \in G'$	(2)
x_{pgd}	=	0 or 1		(3)

$$l_g \quad \le \quad g_g \qquad \forall g \in G \qquad (4)$$

$$\sum_{l \in L} \sum_{p \in P_{gd}} x_{pgd} d_{pl} \quad \le \quad H_g \qquad \forall g \in G \qquad (5)$$

$$y_{gl} \quad = \quad 0 \text{ or } 1 \qquad g \in G \qquad (6)$$

$$\sum_{l \in L} y_{gl} \quad \ge \quad \max\{h_g, |D_g|\} \qquad \forall g \in G \qquad (7)$$

$$z_g \quad = \quad 0 \text{ or } 1 \qquad g \in G \qquad (8)$$

$$t_{gd} \quad = \quad \sum_{l \in L} \frac{x'_{pgd} d_{pl}}{g_g - l_g} \qquad \forall p \in Pgd \quad \forall d \in Dg, \forall g \in G \qquad (9)$$

$$s_{gd} \quad = \quad \sqrt{\sum_{l \in L} \frac{x'_{pgd} d_{pl}}{\left(g_g - l_g\right)^2}} \qquad \forall p \in Pgd \quad \forall d \in Dg, \forall g \in G \qquad (10)$$

$$S_g \quad \ge \quad \sum_{p \in P'_{gd}} x'_p (1 - x_p) d_{pl} \left(t_{gd} + 3s_{gd}\right) \qquad \forall d \in D_g, \forall g \in G \qquad (11)$$

$$S_g \quad \le \quad Q_g \qquad g \in G. \qquad (12)$$

where $a_l, u_{gl}, v_{gd}, w_{gd}$ are Lagrangean multipliers and are all ≥ 0. To solve (LR2), we can decompose (LR2) into the following four independent and easily solvable optimization subproblems.

Subproblem 2.1: (related to decision variable x_{gpd})

$$Z_{sub2.1}(u_{gl}, v_{gd}, w_{gd}) = \min \sum_{l \in L} \sum_{g \in G} u_{gl} \sum_{p \in P_{gd}} x_{pgd} d_{pl} - \sum_{g \in G} \sum_{d \in D_g} v_{gd} \sum_{p \in P_{gd}} x_{pgd} +$$

$$\sum_{g \in G} \sum_{d \in D_g} w_{gd} \sum_{p \in P_{gd}} \sum_{l \in L} -x'_{pgd} x_{pgd} (t_{gd} + 3s_{gd})$$

$$= \min \sum_{g \in G} \sum_{d \in D_g} \sum_{p \in P_{gd}} \sum_{l \in L} [u_{gl} d_{pl} - w_{gd} x'_{pgd} (t_{gd} + 3s_{gd}) - v_{gd}] x_{pgd}$$

(LR2.1)

subject to:

$$\sum_{p \in P'_{gd}} x_{pgd} = 1 \quad \forall d \in D_g, \forall g \in G^{*} \quad (1)$$

$$\sum_{p \in P_{gd}} x_{pgd} = 1 \quad \forall d \in D_g, \forall g \in G' \quad (2)$$

$$x_{pgd} = 0 \text{ or } 1 \quad (3)$$

$$\sum_{l \in L} \sum_{p \in P_{gd}} x_{pgd} d_{pl} \leq H_g \quad \forall d \in D_g, \forall g \in G. \quad (4)$$

This subproblem can be further decomposed to two subproblems.

Optimization problem (LR 2.1.1):

$$Z_{sub_{2.1.1}}(u,v) = \min \sum_{g \in G'} \sum_{d \in D_g} \sum_{p \in P_{gd}} \sum_{l \in L} (u_{gl} d_{pl} - v_{gd}) x_{pgd} \quad \text{(LR2.1.1)}$$

subject to

$$\sum_{p \in P'_{gd}} x_{pgd} = 1 \quad \forall d \in D_g, \forall g \in G^{*} \quad (1)$$

$$x_{pgd} = 0 \text{ or } 1 \quad (2)$$

$$\sum_{l \in L} \sum_{p \in P_{gd}} x_{pgd} d_{pl} \leq H_g \quad \forall d \in D_g, \forall g \in G. \quad (4)$$

This subproblem can be further decomposed to $|G||D_g|$ independent hop-count constrained shortest path problems with nonnegative arc weights. The link cost is u_{gl}, and hop-count constraint is H_g. This can be solved by using Bellman-Ford algorithm. If $u_{gl} - v_{gd}$ is negative, $\sum_{p \in P_{gd}} x_{pgd}$ is 1, otherwise, $\sum_{p \in P_{gd}} x_{pgd}$ is 0.

Optimization problem(LR 2.1.2):

$$Z_{sub_{2.1.2}}(u,w) = \min \sum_{g \in G'} \sum_{d \in D_g} \sum_{p \in P_{gd}} \sum_{l \in L} [u_{gl} d_{pl} - w_{gd} x'_{pgd} (t_{gd} + 3s_{gd})] x_{pgd} \quad \text{(LR2.1.2)}$$

subject to

$$\sum_{p \in P'_{gd}} x_{pgd} = 1 \qquad \forall d \in D_g, \forall g \in G' \qquad (1)$$

$$x_{pgd} = 0 \text{ or } 1 \qquad (2)$$

$$\sum_{l \in L} \sum_{p \in P_{gd}} x_{pgd} d_{pl} \leq H_g \qquad \forall d \in D_g, \forall g \in G. \qquad (4)$$

$$t_{gd} = \sum_{l \in L} \frac{x'_{pgd} d_{pl}}{g_g - l_g} \qquad \begin{array}{l} \forall p \in Pgd \\ \forall d \in Dg, \forall g \in G \end{array} \qquad (5)$$

$$s_{gd} = \sqrt{\sum_{l \in L} \frac{x'_{pgd} d_{pl}}{\left(g_g - l_g\right)^2}} \qquad \begin{array}{l} \forall p \in Pgd \\ \forall d \in Dg, \forall g \in G \end{array} \qquad (6)$$

This subproblem can be further decomposed to $|G||D_g|$ independent hop-count constrained shortest path problems with nonnegative arc weights. The link cost is u_{gl}, and hop-count constraint is H_g. This can be solved by using Bellman-Ford algorithm. If $u_{gl} - w_{gd} x'_{pgd}(t_{gd} + 3s_{gd})$ is negative, $\sum_{p \in P_{gd}} x_{pgd}$ is 1, otherwise, $\sum_{p \in P_{gd}} x_{pgd}$ is 0.

Subproblem 2.2: (related to decision variable y_{gl})

$$Z_{sub2.2}(u_{gl}, a_l) = \min\left(\sum_{g \in G} \sum_{l \in L} a_l g_g y_{gl} - \sum_{l \in L} \sum_{g \in G} u_{gl} y_{gl} \right)$$

$$= \min \sum_{l \in L} \sum_{g \in G} (a_l g_g - u_{gl} |D_g|) y_{gl} \qquad \text{(LR2.2)}$$

subject to

$$y_{gl} = 0 \text{ or } 1 \qquad \forall g \in G \qquad (1)$$

$$\sum_{l \in L} y_{gl} \geq \max\{h_g, |D_g|\} \qquad \forall g \in G. \qquad (2)$$

The algorithm to solve subproblem 2.2 is stated as follows:

Step1. Compute $\max\{h_g, |D_g|\}$ for multicast group *g*.

Step2. Compute the number of negative coefficient $a_l g_g - u_{gl} |D_g|$ for all links on multicast group *g*.

Step3. If the number of negative coefficient is greater than or equal to $\max\{h_g, |D_g|\}$ for multicast group *g*, then assign the corresponding negative coefficient of y_{gl} to 1 and 0 otherwise.

Step4. If the number of negative coefficient is no greater than $\max\{h_g, |D_g|\}$ for multicast group *g*, then assign the corresponding negative coefficient of y_{gl} to 1. Then, assign ($\max\{h_g, |D_g|\}$ − the number of negative coefficient of y_{gl}) numbers of smallest positive coefficient of y_{gl} to 1 and 0 otherwise.

Subproblem 2.3: (related to decision variable z_g)

$$Z_{Sub2.3}(v_{gd}) = \min \sum_{g \in G''} \sum_{d \in D_g} (v_{gd} - a_g) z_g \qquad \text{(LR2.3)}$$

subject to

$$z_g = 0 \text{ or } 1 \qquad g \in G.$$

The subproblem is to determine z_g, and can be further decomposed to $|G''|$ subproblems. For each subproblem, there are two cases to consider:

Case 1. If $\sum_{d \in D_g} (v_{gd} - a_g) \geq 0$, then $z_g = 0$.

Case 2. If $\sum_{d \in D_g} (v_{gd} - a_g) < 0$, then $z_g = 1$.

Subproblem 2.4: (related to decision variable s_g)

$$Z_{Sub2.4}(w_{gd}) = \min \sum_{g \in G} \sum_{d \in D_g} (Cg_g R_g - w_{gd}) S_g \qquad \text{(LR2.4)}$$

subject to:

$$S_g \quad \le \quad Q_g \qquad g \in G^{'}. \qquad (1)$$

The subproblem is to determine S_g, and can be further decomposed to $|G^{'}|$ subproblems. There are two cases to consider:

Case 1: If $\sum_{d \in D_g} Cg_g R_g - w_{gd} \le 0$, then S_g is set to be its maximum value, Q_g.

Case2: If $\sum_{d \in D_g} Cg_g R_g - w_{gd} > 0$, then S_g is set to be its minimum value, 0.

3.3.3 The Dual Problem and the Subgradient Method

According to the weak Lagrangean duality theorem, for any $u_{gl}, v_{gd}, w_{gd}, a_l \ge 0$ $Z_{D2}(a, u, v, w)$ is a lower bound of $IP4$. The following dual problem (D2) is then constructed to calculate the tightest lower bound.

Dual Problem (D1)

$$Z_{d2} \quad = \quad \max Z_{D2}(a, u, v, w) \qquad \text{(D2)}$$

Subject to:

$$a_l, u_{gl}, v_{gd}, w_{gd} \quad \ge \quad 0.$$

The most popular method to solve the dual problem is the subgradient method. Let g be a subgradient of $Z_{D2}(a,u,v,w)$. Then, in iteration k of the subgradient optimization procedure, the multiplier vector $p=(a,u,v,w)$ is updated by $p^{k+1}=p^k+t^k g^k$. The step size t^k is determined by $t^k = d\,\dfrac{Z_{IP4}^{h}-Z_{D2}(p_k)}{\left\|g^k\right\|^2}$. Z_{IP2}^{h} is the primal objective function value for a heuristic solution. d is a constant between 0 and 2.

Chapter 4 Getting Primal Feasible Solution

4.1 Heuristics for the Unicast Model

To calculate primal feasible solutions for the unicast model, solutions of the Lagrangean Relaxation problems are considered. By using Lagrangean relaxation and the subgradient method as our tools to solve these problems, we can get not only a theoretical lower bound of primal feasible solution, but also some hints to help us get primal feasible solution. As some constraints are relaxed from original problems to obtain easier-solved problems, the set of decision variables we obtain from $Z_{D1(u)}$ may not be a valid solution set. We need to develop some heuristics to tune these decision variables so that they may constitute a feasible solution. In this section, we describe the detail of these heuristics.

4.1.1 ReArrangeExisting Heuristic

As we relax the link capacity constraint, try to reroute some O-D pairs, aggregated flow on some links may violate link capacity constraint. We should reroute those O-D pairs that introduce the violation. In all O-D pairs, sort all O-D pairs by its transmission delay. Then reroute O-D pairs in this sequence, till the aggregate flow of existing O-D pairs on every link doesn't exceed the link capacity.

4.1.2 Reroute Heuristic

After the ReArrangeExisting heuristic, we can make sure that all links at least not over-loaded before new traffic admitted. But, based on the decision variable obtained in the subproblem, with new O-D pairs admitted, not all traffic are not over-loaded. So we try to reroute those O-D pairs that use these links. The steps of reroute heuristic are as following:

1. Sort all links by the amount of "exceeding flow" (aggregate flow – link capacity).

2. For each link, sort every O-D pair in the sequence of the number of over-loaded links used.

3. Then reroute these O-D pairs in sequence. When rerouting these O-D pairs, Bellman-Ford algorithm is used. The link cost is multiplier u_l for each link *l*. If the aggregated flow of the link plus the flow request of the O-D pair will cause the link capacity constraint be violated, or the link capacity constraint is already violated, the link can not be used. The link cost is set to be MAX.

4.1.3 Drop Heuristic

After the Reroute heuristic, we have tried to rearrange all traffic flows which are admitted. But there may be some links also over-loaded. So, we have to drop some admitted O-D pairs to meet the capacity constraint. To do this, the Drop heuristic is developed, the steps of the Drop heuristic are as following:

1. Sort all to-be-admitted O-D pairs by the value: $\sum_{p \in P_w} \sum_{l \in L} u_l g_w d_{pl} - a_w$.

2. Drop the O-D pairs in sequence.

3. check the link capacity if satisfied. If not, repeat step 1 and step 2.

4.1.4 Add Heuristic

After the Drop heuristic, the link capacity is satisfied. Thus, the solution set is feasible. But in order to maximize the revenue from the network, those O-D pairs that are not admitted should have a second chance to be admitted using the residual capacity. So we develop the Add heuristic. The steps of the Add heuristic are as following:

1. Sort all not-yet-admitted O-D pairs by the value: $\sum_{p \in P_w} \sum_{l \in L} u_l g_w d_{pl} - a_w$.

2. Try to route each O-D pair using Bellman-Ford algorithm in sequence. The link cost is the value of multiplier u_l . On each link, if the aggregated flow will cause the link capacity on one link be violated, the link can not be used. The link cost is set to be MAX.

4.2 Heuristic for the Multicast Model

To calculate primal feasible solution for the multicast model, solutions of the Lagrangean Relaxation problems are considered. By using Lagrangean relaxation and the subgradient method as our tools to solve these problems, we can get not only a theoretical lower bound of primal feasible solution, but also some hints to help us get primal feasible solution. As some constraints are relaxed from original problems to obtain easier-solved problems, the set of decision variables we obtain from $Z_{D2}(a, u, v, w)$ may not be a valid solution set. We need to develop some heuristics to tune these decision variables so that they may constitute a feasible solution. In this section, we describe the detail of these heuristics.

4.2.1 CheckGroup Heuristic

As we relax the relation between x_{pgd} and z_g, after we solve the subproblems this constraint may be violated. So we first should check for this constraint. Check each z_g, if z_g is set to be 1, check if all destinations of group g find a routing path. If not, set z_g =0, and set corresponding x_{pgd} to be 0. If z_g is set to be 0, set all corresponding x_{pgd} to be 0.

4.2.2 ReArrangeExisting Heuristic

As we relax the link capacity constraint, trying to reroute some multicast groups, the link capacity constraint on some links may be violated. We should reroute those multicast groups that introduce the violation. For all multicast groups, sort them by transmission delay. Then reroute them in the order of transmission delay, till the link capacity constraint on all links are satisfied.

4.2.3 Reroute Heuristic

After the ReArrangeExisting heuristic, we can make sure that all links are at least not over-loaded before new multicast groups admitted. But, based on the decision variables obtained in subproblems, with new multicast groups admitted, some links may be over-loaded. So we should reroute those groups that use these over-loaded links. The steps of reroute heuristic are as following:

1. Sort all links by the amount of "exceeding flow" (aggregate flow – link capacity).

2. For each link, sort all multicast groups in the sequence of the number of over-loaded links used.

3. Then reroute these multicast groups in sequence using Dijkstra algorithm. The link cost is the value of multiplier u_{gl}. If admitting the multicast group will cause link capacity constraint be violated, or the link capacity is already violated, the link can not be used. The link cost is set to be MAX.

4.2.3 Drop Heuristic

After the Reroute heuristic, we have tried to rearrange all multicast groups that are admitted. But there may be some links also over-loaded still. So, we have to drop some admitted multicast groups to meet the capacity constraint. To doing this, the Drop heuristic is developed, the steps of the Drop heuristic are as following:

1. Sort all to-be-admitted multicast groups by the value: $\sum_{d \in D_g} (v_{gd} - a_g)$.

2. Drop the multicast groups in sequence.

3. Check if the link capacity constraint is satisfied. If not, repeat step 1 and 2.

4.2.4 Add Heuristic

After the Drop heuristic, the link capacity is satisfied. Thus, the solution set is feasible. But to maximize the revenue from the network, those multicast groups that are not admitted should have a second chance to be admitted using the residual capacity. So we develop the Add heuristic. The steps of the Add heuristic are as following:

1. Sort all not-yet-admitted multicast groups by the value: $\sum_{d \in D_g} (v_{gd} - a_g)$.

2. Try to find a routing tree for each multicast group using Bellman-Ford algorithm. The

link cost is the value of multiplier u_{gl}. On each link, if admitting the multicast group will cause the link constraint on the link violated, the link can not be used. The link cost is set to be MAX.

Chapter 5 Computational Experiments

In order to prove that our heuristics are good enough, we also implement two simple algorithms to compare with our heuristics.

5.1 Simple Algorithm for the Unicast Model

Find a shortest path with hop constraint, H_w, using Bellman-Ford algorithm for each to-be-admitted O-D pairs. Then apply the Reroute heuristic, the Drop heuristic, the Add heuristic in sequence. When Bellman-Ford algorithm is used, the link cost is set to be 1.

5.2 Simple Algorithm for the Multicast Model

Find a shortest path with hop constraint, H_g, using Bellman-Ford algorithm for each destination of all to-be-admitted multicast groups. Then apply the Reroute heuristic, the Drop heuristic, the Add heuristic in sequence. When Bellman-Ford algorithm is used, the link cost is set to be 1.

5.3 Assumptions, Parameters, and Cases

Number Of Iteration	1000
Maximum Unimprovement Counter	100
Begin to Get Primal Solution	200
Initial Upper Bound	0
Initial Scalar of Step Size	2

Table 5-1 Common Parameters

These algorithms are coded in C, and run on a Pentium 4 2.0G PC with 512 MB RAM. In our implementation, the initial upper bounds, Z_{IP2}^{h}, Z_{IP4}^{h}, for unicast model and multicast model respectively are set to be 0. This means at worst case, no new O-D pairs or multicast groups can be admitted. All multipliers are set to be 0 initially.

We have tested two algorithms on two networks, Mesh and GTE, with 9 and 12 nodes. These topologies are shown in Figure 5-1 and 5-2. For each network, we have 3 test cases. The first one is all to-be-admitted traffics can be totally admitted without any existing traffic being rerouted. The second is all to-be-admitted traffics can be admitted only when some existing traffics are rerouted. The final case is not all to-be-admitted traffics can be admitted even existing traffics are rerouted.

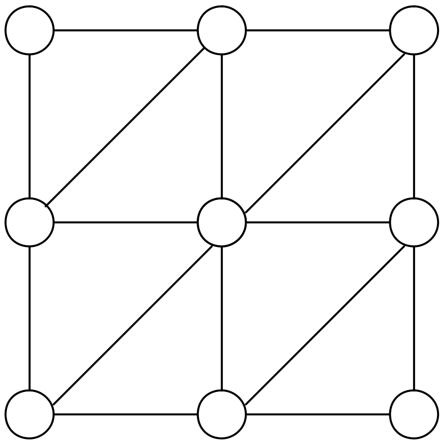

Figure 5-1 9-node 16-link Mesh Network

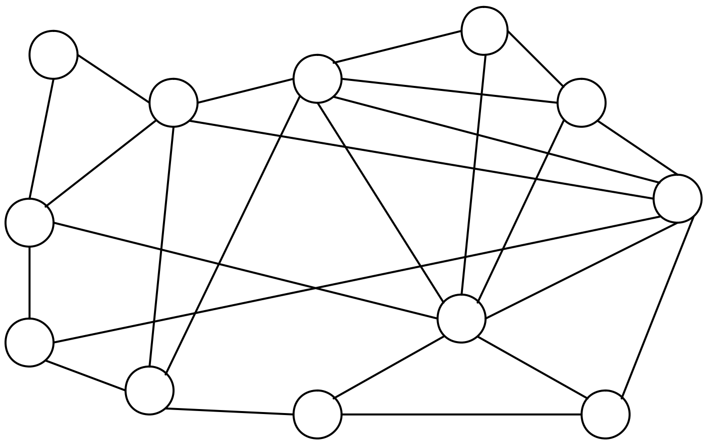

Figure 5-2 12-node 25-link GTE network

5.3.1 Cases for the Unicast Model

For Mesh network, we have 6 existing O-D pairs, and 3 to-be-admitted O-D pairs. Each to-be-admitted O-D pair has revenue 10 respectively. For GTE network, we have three test cases. Two of them have 8 existing O-D pairs, 4 to-be-admitted O-D pairs. Each to-be-admitted O-D pair has revenue 10, too. Another test case for GTE network, we have 9 existing O-D pairs, and 5 to-be-admitted OD-pairs. Each to-be-admitted O-D pair also has revenue 10, too.

5.3.2 Cases for the Multicast Model

For Mesh network, we have 6 existing multicast groups, and 3 to-be-admitted multicast groups. Each to-be-admitted multicast group has revenue 10 respectively. For GTE network, we have two test cases. One has 8 existing multicast groups, 4 to-be-admitted multicast groups. Each to-be-admitted multicast group has revenue 10, too. Another test case for GTE network, we have 9 multicast groups, and 5 to-be-admitted multicast groups. Each to-be-admitted multicast group also has revenue 10, too.

5.4 Experimental Results

5.4.1 Experimental Results for the Unicast Model

Representative results have been selected and list below for the purpose of demonstration.

Cases		SA	LR	Lower Bound	Error Rate	CPU Time (sec)
Mesh Network	1	-30	-30	-30	0%	31.35
	2	-29.1038	-29.1038	-30	3%	31.51
	3	-10	-19.1033	-30	29.36%	31.78
GTE Network	1	-40	-40	-40	0%	37.42
	2	-38.5766	-38.7872	-40	3%	37.61
	3	-20	-28.6748	-40	28.31%	38.03
GTE Network	1	-40	-40	-40	0%	37.33
	2	-38.9118	-39.0273	-40	2.43%	37.57
	3	-20	-28.4487	-40	28.87%	37.92
GTE Network	1	-50	-50	-50	0%	37.68
	2	-48.5127	-48.6284	-50	2.74%	38.01
	3	-30	-38.2103	-48.8082	21.71%	38.21

Table 5-2 Experimental Results of the Unicast Model

5.4.2 Experimental Results for the Multicast Model

Representative results have been selected and list below for the purpose of demonstration.

Cases		SA	LR	Lower Bound	Error Rate	CPU Time (sec)
Mesh Network	1	-30	-30	-30	0%	39.63
	2	-28.7833	-28.8126	-30	3.9%	40.55
	3	-10	-18.7231	-30	37.59%	41.02
GTE Network	1	-40	-40	-40	0%	50.21
	2	-37.8462	-38.1672	-40	4.58%	51.34
	3	-20	-28.3417	-40	29.14%	52.16
GTE Network	1	-50	-50	-50	0%	55.76
	2	-47.2125	-47.6184	-50	4.76%	57.03
	3	-30	-37.8213	-42.2251	10.43%	58.34

Table 5-3 Experimental Results of the Multicast Model

5.5 Result Discussion

According to our experimental result, we can see that the result of SA is not good as LR. The reason is that, in SA method, we do not use any informative parameters to be link costs. We just set each link cost to be 1, and find a minimum-hop shortest path. No matter how heavy the traffic is on the link, the cost of the link is always the same. Thus, the path we found will not change according to the status of the whole network.

But in LR method, we use multipliers to be the link cost when we try to reroute or find paths for O-D pair or for R-D pair of multicast groups. Values of multipliers are tuned by gradient method iteration by iteration. When a link is over-loaded, the value of that link will increase iteration by iteration, and that link will not be a good choice for constructing a path. For existing O-D pairs or R-D pairs of multicast groups, they are thus "rerouted according to the status of the network". Once rerouting happens, new traffic will have "another" chance to be admitted.

In this problem, we can use linear programming relaxation to explain the error rate. Because of the integer constraint, there will be a bound between lower bound and the result of heuristic. We call it duality gap. If we eliminate the integer constraint, for example, we can use more than one path to transmit the traffic of one O-D pair, more traffics will be transmitted over the network. Thus, the network utilization can be maximized. And the corresponding calculated revenue, the lower bound, will become higher than the result of our heuristic.

Chapter 6 Summary and Conclusions

6.1 Summary

In this thesis, our work emphasizes on considering rerouting existing traffics while deciding to admit new traffics. When we take the rerouting of existing traffic into consideration, new traffics will have more chance to be admitted. At the same time, the network utilization is maximized. But rerouting does cost. If the revenue from admitting a new traffic is more than the rerouting cost, than the new traffic is not worth admitting. We use equivalent bandwidth for QoS constraints.

We proposed two models: one is for unicast model, and the other one is for multicast model. For both model, we formulate the problem in mathematical formulations. Our objective function is to maximize total revenue minus rerouting cost. Then we use Lagrangean Relaxation method to solve the problem. While applying this methodology, we relax some complicated constraints to make the problem more easily solvable. Then we decompose the problem into several subproblems. We analyze the subproblems and optimally solve these subproblems. We develop several heuristics to obtain primal feasible solution.

We implement the algorithms in C code, and test them using two well-known networks.

In the experiment result, we have a nice result. Applying our algorithms, existing traffics are rerouted and new traffics can use those links originally used by existing traffics. The network utilizations are improved.

6.2 Conclusions

The contribution of this thesis would be take rerouting and QoS into consideration together while doing admission control. With QoS, network operators can provide controllable services with better quality. And users of the network will be more satisfied and willing to pay more for the service. Admission control with trying to reroute existing traffic flows will increase the network utilization. Network operators can provide more service without adding capacity.

In this thesis, we develop an easily implemented algorithm to solve this problem. We use equivalent bandwidth for QoS constraints. An easily calculated closed-form expression makes the algorithm more efficient. And also, this algorithm can be applied to many situations with different QoS constraints without a widely modification on problem formulation and structure. Only the calculation of equivalent bandwidth would be changed.

According to our experimental results, the computation time is not long and the qualities of results are also not bad. Comparing to simple algorithms, in heavy-loaded environments, it has a very well performance. Multipliers do take effect on this situation. The values of the multipliers are updated according to the decision variables (network flow allocation) in previous iteration. And they affect the routing decision in the current iteration. Multipliers help the algorithm to make better routing decisions. And finally, converges to an optimal solution.

6.3 Future Works

In this paper, we only use end-to-end delay as our QoS constraints. There are still many aspects for QoS constraints, for example, delay jitter. One can develop his own formulation of equivalent bandwidth for different aspects of QoS constraints.

Another, there are different methods for calculating rerouting cost and deciding relative parameters. What we propose is only one of them. One can try to develop a new formulation to calculate rerouting cost.

Also, this can apply to wireless networks. Rerouting in wireless networks cost more and is more complex to calculate corresponding cost. The cost of rerouting not only includes delay, but also the cost to handover between access points. The resource of access points, the number of channels, should also be taken into consideration. The model and formulation will become more complex, and also more interesting.

References

[1] A.W. Berger, W. Whitt, "*Effective bandwidths with priorities*", IEEE/ACM Transactions on Networking, Vol.6, Issue.4, Aug. 1998, pp.447-460.

[2] E. Crawley, R. Nair, B. Rajagopalan, H. Sandick, "*A Framework for QoS-based Routing in the Internet*", RFC2386, Aug. 1998.

[3] M.L. Fisher, "*The Lagrangean relaxation method for solving integer programming problems*", Management Science, Vol. 27, pp.1-18, 1981.

[4] A.M. Geoffrion, "*Lagrangean relaxation and its use in integer programming*", Math. Programming Study, Vol.2, 1974, pp.82-114.

[5] R.J. Gibbens, F.P. Kelly, and P.B. Key. "*A Decision-Theoretic Approach to Call Admission Control in ATM Networks*", IEEE Journal of Selected Areas in Communication, Vol.13, Aug. 1995.

[6] R. Guerin, H. Ahmadi, M. Naghshineh, "*Equivalent capacity and its application to bandwidth allocation in high-speed networks*", Selected Areas in Communications, IEEE Journal on, Vol.9, Issue.7, Sep. 1991 pp.968-981.

[7] S. Jamin, S.J. Shenker and P.B. Danzig, " *Comparison of Measurement-Based Admission Control Algorithms for Controlled-Load Service*", in Proceedings of the Conference on Computer Communications, IEEE INFOCOM'97, Kobe, Japan, Apr. 1997, pp. 973.

[8] Y.S. Lin, C.T. Chen, "*Admission Control and Routing Algorithms for Networks Supporting the Permanent Virtual Connection (PVC) Service*", 1995.

[9] Y.Y. Liu, P.H. Tu, Z.F. Zhang, "*Equivalent bandwidth estimation under delay constraints*", Communication Technology Proceedings, 2000. WCC - ICCT 2000. International Conference on, Vol.2, 2000, pp.1465-1469.

[10] M. Luoma, M. Peuhkuri, and T. Yletyinen, "*Quality of service for IP voice services - is it necessary?*", In Proceedings of Voice, Video and Data '98, Nov. 1998, pp.242-253.

[11] V. Paxon, S. Floyd, "*Why we don't know how to simulate theInternet*", In Proceedings of the 1997 Winter Simulation Conference, Dec. 1997.

[12] M. Peuhkuri, "*IP Quality of Service*", 1999.

[13] D. Tse and M. Grossglauser, "*Measurement-Based Call Admission Control: Analysis and Simulation*", IEEE INFOCOM '97, Kobe, Japan, Apr. 1997.

[14] Z. Wang and J. Crowcroft, "*QoS Routing for Supporting Resource Reservation*", IEEE JSAC, Sep. 1996.

[15] H.H. Yen, "*Network Planning and Capacity Management for Computer and Logistic Networks*", 2001.

簡　　歷

姓　　名：胡晉華

出 生 地：台灣省台南縣

出 生 日：中華民國六十六年八月一日

學　　歷：八十四年九月至八十八年六月

國立台灣大學資訊管理學系

九十年九月至九十二年七月

國立台灣大學資訊管理學研究所