國立臺灣大學管理學院資訊管理學系

碩士論文

Department of Information Management
College of Management
National Taiwan University

Master Thesis

考慮服務品質限制下利用路由選徑與資源配置防禦分散式阻絕服務攻擊

Defense against Distributed Denial-of-Service (DDoS) Attacks by Routing Assignment and Resource Allocation under Quality-of-Service (QoS) Constraints

郭承賓

Cheng-Bin Kuo

指導教授：林永松 博士

Advisor: Yeong-Sung Lin, Ph.D.

中華民國 96 年 7 月

July, 2007

國立臺灣大學碩士學位論文

口試委員會審定書

考慮服務品質限制下利用路由選徑與資源配置
防禦分散式阻絕服務攻擊

本論文係郭承賓君（學號 R94725045）在國立臺灣大學資訊管理學系、所完成之碩士學位論文，於民國 96 年 7 月 19 日承下列考試委員審查通過及口試及格，特此證明

口試委員：

所　　長：

謝 誌

能夠寫謝誌，心中的感觸五味雜陳，回顧過去兩年多的研究生涯，顛顛簸簸，一步步咬緊牙關往前走，一路上得到了很多的幫助，要感謝的人很多，因為有了你們的幫助，我才能夠順利完成學業。

首先要感謝我的恩師，林永松博士，這兩年來對我的諄諄教誨，每逢學生遇到瓶頸與難題時，恩師總是給予鼓勵以及思考上的啟發，讓學生獲益良多。恩師的風度翩翩，憂國憂民更是學生心目中的典範，從恩師身上學到的東西，在未來的路上，終身受用。承蒙口試委員孫雅麗博士、呂俊賢博士、顏宏旭博士、祝國忠博士於口試時給予寶貴的意見，讓學生的論文更加完備。

此外，要感謝實驗室的所有學長姐、學弟妹，其中，特別感謝柏皓學長的各方面指導、人生處事上的寶貴意見，以及中蓮學姐的協助，讓我的研究更加順利。學弟妹的給予的幫忙更讓我能夠專心完成論文。當然，非常感謝一起奮鬥的實驗室夥伴，翊恆、豈毅、坤道、俊維、雅芳、怡孜，你們的陪伴讓研究之路不孤單，曾經一起在實驗室打拼的日子，永生難忘，希望大家結束這段旅程後，都能夠展開心中美好的道路。

另外要感謝的是在我生活上給予協助的朋友,讓我能無後顧之憂的完成研究。感謝經常陪我一起度過低潮的賢祺、還有其他大學同學、研究所同學的陪伴。謝謝介紹工讀機會給我的亞真、信惠、克非，還有台灣微軟的雷歐娜、張天師、克里斯、黛比，感謝你們給我機會以及珍貴的建議。

也要感謝在精神上支持我的好朋友，穎可，感謝你支持我進入研究所、申請交換學生，你所給予過的鼓勵與支持，曾經是我能夠不斷走下去的動力。而在研究的最後關頭還有出國當交換學生的日子，要感謝好朋友，郁萍，你所帶來的歡樂與陪伴，讓我感到溫暖，讓我的精力源源不絕，完成研究與交換學生。

最後要感謝我的家人，姐姐慈芬，總是給予我各方面的支持與鼓勵、以及引導我向前，還有我摯愛的雙親，郭萬清先生與林梅香女士，你們對我的全心全意支持，提供了一個隨時歡迎我的避風港，讓我能夠不畏艱難的完成研究。完成學業的喜悅，最想與你們分享。

論文摘要

論文題目：考慮服務品質限制下利用路由選徑與資源配置

防禦分散式阻絕服務攻擊

作者：郭承賓 **九十六年七月**

指導教授：林永松　博士

隨著網路使用的普及，網路攻擊事件層出不窮，尤其是分散式阻絕服務攻擊，往往造成網路上服務提供者資源的損失以及使用者服務品質的權益受損。因此在遭受攻擊時，網路管理者為了維持使用者的服務品質，利用備用資源配置去良好地設計一個網路是有其需要的。

本論文中，在滿足服務品質限制下將利用路由選徑以及資源配置去防禦智慧型的分散式阻絕服務攻擊。我們將攻防的情境轉化成一個最大最小化的雙層數學規劃問題；內層問題（最小化）代表當一個網路遭受某種模式的攻擊時，網路管理者利用決定最少的防禦資源配置需求以及路由選徑策略與去維持網路內部使用者的服務品質，外層問題（最大化）則為網路管理者假設在給定攻擊流量時，有一攻擊者利用攻擊模式的調整以求最大化網路的整體防禦資源需求。為了求得最佳解，我們利用拉格蘭日鬆弛法為基礎的演算法來處理內層的問題，並利用次梯度法為基礎的演算法來解外層的問題。解出問題之後，我們預期發展出有效率且有效用的演算法。

關鍵詞：分散式阻絕服務攻擊、拉格蘭日鬆弛法、服務品質、路由選徑、資源配置

THESIS ABSTRACT

Defense against Distributed Denial-of-Service (DDoS) Attacks by Routing Assignment and Resource Allocation under Quality-of-Service (QoS) Constraints

NAME：CHENG-BIN KUO MONTH/YEAR：July 2007

ADVISOR：YEONG-SUNG LIN

As the popularity of networks is increasing, network attack events occur frequently, especially Distributed Denial-of-Service (DDoS) attacks. Upon such attacks, system resources are dramatically consumed and the Quality-of-Service (QoS) perceived by users significantly degrades. In order to achieve the objective of “continuity of services”, it is then essential that a network be well designed by spare resource allocation so as to maintain acceptable QoS levels upon such attacks.

In this thesis, the problem of defense against intelligent DDoS attacks by routing and budget allocation (RB) under QoS constraints is considered. This problem is formulated as a max-min integer programming problem, where the inner (minimization) problem is for network administrators to determine the minimum amount of defense budget required and effective internal routing policies so as to defend the network against a given pattern of DDoS attacks under given QoS requirements, while the outer (maximization) problem is for network administrators to evaluate the worst-case defense resource required when attacks adjust the patterns of DDoS attack flows (AF) under a fixed total attack power. A Lagrangean relaxation-based algorithm is proposed to solve the inner problem, while a subgradient-based algorithm is proposed to solve the outer problem. It is expected that efficient and effective algorithms be developed accordingly.

Keywords: Distributed Denial-of-Service, Lagrangean Relaxation, Quality-of-Service, Routing Assignment, Resource Allocation.

Table of Contents

List of Tables

List of Figures

Chapter 1 Introduction

1.1 Background

In the network attack events, Distributed Denial of Service (DDoS) attacks become common and it is easy to get all kinds of attack tools on the web nowadays [6]. When DDoS attacks happened, the network suffered performance degradation and waste of resources. The increasing popularity and utilization of the Internet raise the issue of defending against the DDoS attacks. The network administrators would like to find some solutions to mitigate the loss due to attacks. Thus, how to defense DDoS attacks becomes an important issue and the effect of defense mechanism is also critical.

Typically, a DDoS attack relies on an attacker remotely controlling numerous and widely distributed computers infected by viruses and Trojans. The attacker uses these botnets to send a flood of requests to a website or overwhelming packets to a network, which is often unable to resist, see (Figure 1 – 1). Some attacks are well known such as the February 2000 attack on popular websites including Yahoo, CNN, eBay, the attacks on the root DNS servers, and the May 2007 attack on Estonia by Russian hackers. Since many computers have become zombies, it is a relatively simple and cheap operation to execute attacks for an attacker. There are some online resources where someone can hire bots for cheap pay, so anyone could take down a site simply. It could get enough strength together, for instance, a 100Mbits DDoS attack. Some attack approaches are very common, such as TCP SYN Flooding attacks, ICMP Flooding attacks, and UDP Flooding attacks. Even some of them can be defended, but as time goes on, variations of DDoS attacks are made by attackers.

It is worth to note how to design a survivable network under attacks. Typically in reactive defense mechanisms, we have to detect the attacks first and then apply some mechanisms to protect our resources. Preventive mechanisms attempt to eliminate the possibility of DDoS attacks or ensure the legitimate clients are not denied. Detecting approaches are various and defense mechanisms are introduced by [2] [3] [4] [5]. Instead of detecting and defending attacks independently, it is critical to resist the attacks in a collaborative manner.

To measure the survivability of a network, we also have to evaluate what level of DDoS attacks that a network can sustain. Some metrics can be used to evaluate the survivability of network under attacks, including availability, connectivity and performance [8] [16] [17]. For availability, a client can reach the servers, which provide services. Thus, to fully disable communication, an attack would need to disable multiple servers or entry nodes of the network. Performance, as a network operator, we want to maintain the Quality of Service (QoS) of the network under attacks. The quality, which could be transmission time, of internal and external communication in a network should be guaranteed.

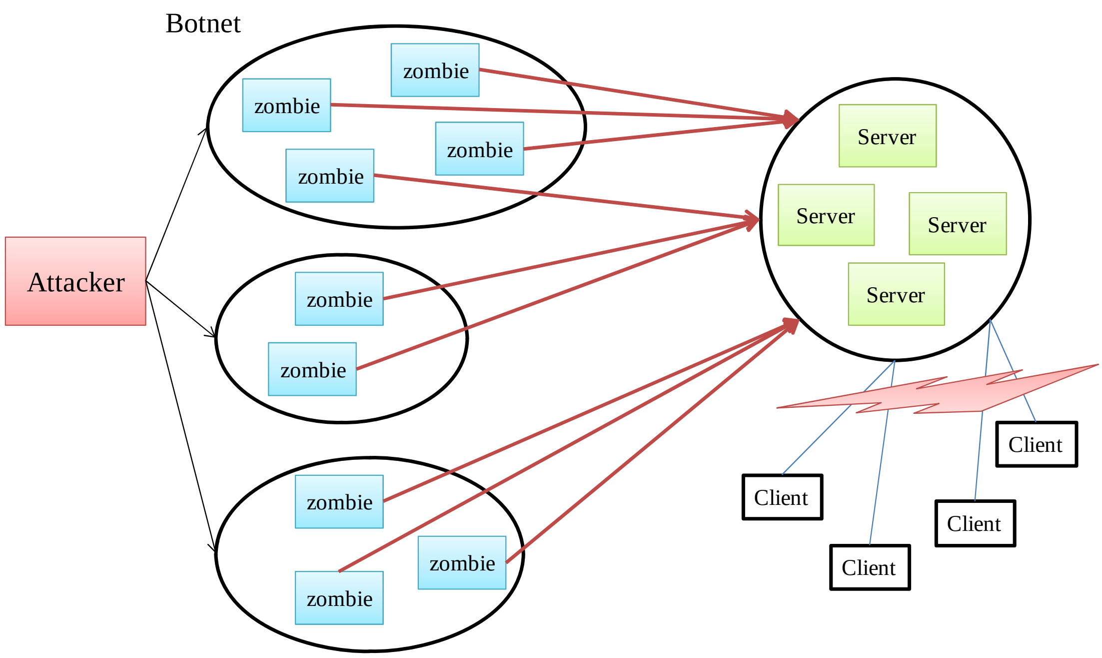


Figure 1 - 1 Attacker Uses Botnets to Attack the Victim

1.2 Motivation

Resources of network are consumed heavily and QoS degrades when suffering DDoS attacks. Several DDoS attacks detection and defense approaches are introduced, but the budget cannot be guaranteed. Also, defense approaches are variable, which makes it hard to be integrated, and seldom cover different types of DDoS attacks. Thus, we introduce network planning to maintain QoS under attacks on victim end. The concept of defense-in-depth is also considered (Figure 1 – 2).

Although the percentage of DDoS attack in security events is declining [21], it still has great impact on networks once malicious attackers want to breach them. So we want to design a survivable network, which can sustain the abnormal traffic while other defense mechanisms cannot work perfectly. In the attack and defense scenarios, we also consider the spare resource allocation by defenders [15]. What we try to do is to construct a mathematical model of attack and defense scenarios and therefore quantitative analysis can be applied. Consequently, budget spending on information security and loss due to attacks can be estimated more accurate.

In order to simulate the characteristics of real network, the network self-similarity is considered [10]. The nature of self-similarity of network traffic is well studied, but few of them discuss the phenomenon under attack situation. In our research, the impact on performance by network self-similarity and DDoS attacks are jointly considered since the attacks can be detected based on the nature of DDoS attacks, which influences the network self-similarity of traffic [4].

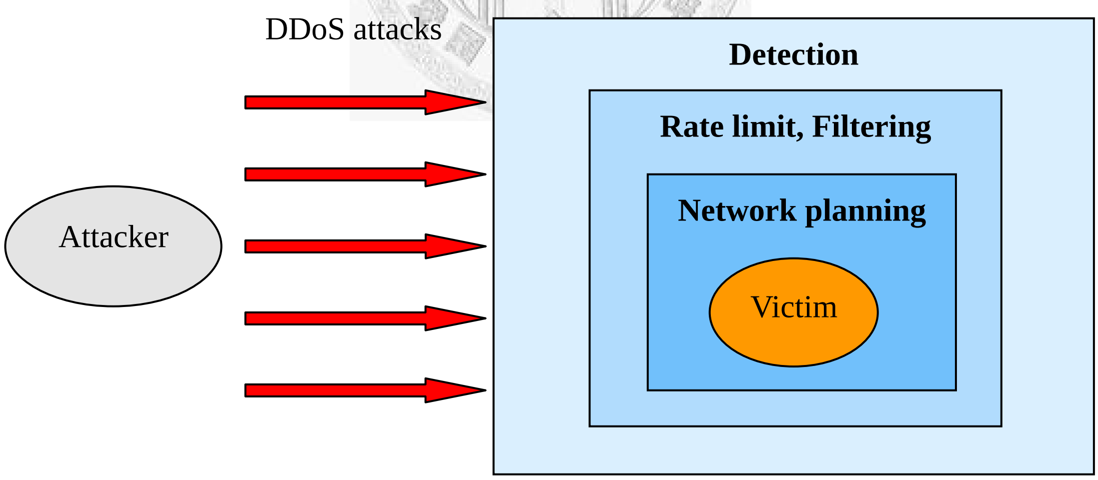


Figure 1 - 2 Defense-in-Depths

From the viewpoint of economy, we want to reduce the damage due to DDoS attacks, because the cost is huge while our society and economy are highly depended on

Internet. Instead of link or node attack, we focus on defending against the attackers whose objective is to exhaust the entire resources of the network.

1.3 Literature Survey

1.3.1 DDoS Attack and Defense

Although DDoS attacks tools are developed rapidly [6], the defense approaches are also studied by many researchers [3] [5] [8]. In [3], a router throttle is installed at selected upstream routers, and the throttle can be the leaky-bucket rate, which drops the attacker packets. Jelena Mirkovic et al. made taxonomy of DDoS attacks and defense mechanisms [1], and the attacks that target on key resource and degrading, which ties up only certain percentage of victim's resource, may not be detected easily. Besides, the integration of variable defense approaches is not easily achievable. Some defense mechanisms are separated by network areas, such as victim end, intermediate network, and source end [5]. Instead of deploying on source end and intermediate routers, we proposed the network planning and spare resource allocation in victim end to be a final defense of the collaborative defense (Figure 1 - 3). As mention before, another interesting finding is that detecting DDoS attacks by the self-similarity of traffic flows [4]. When the attacks occur, some phenomenon appears to affect the Hurst parameter values that differ from normal one. Instead of using detection and packets dropping, the survivable overlay network is proposed [8]. A survivable overlay network can resist the DoS attacks via rewire architecture and maximize the end-to-end connectivity between clients and servers. This kind of defense mechanism can also be a final defense of the victim network.

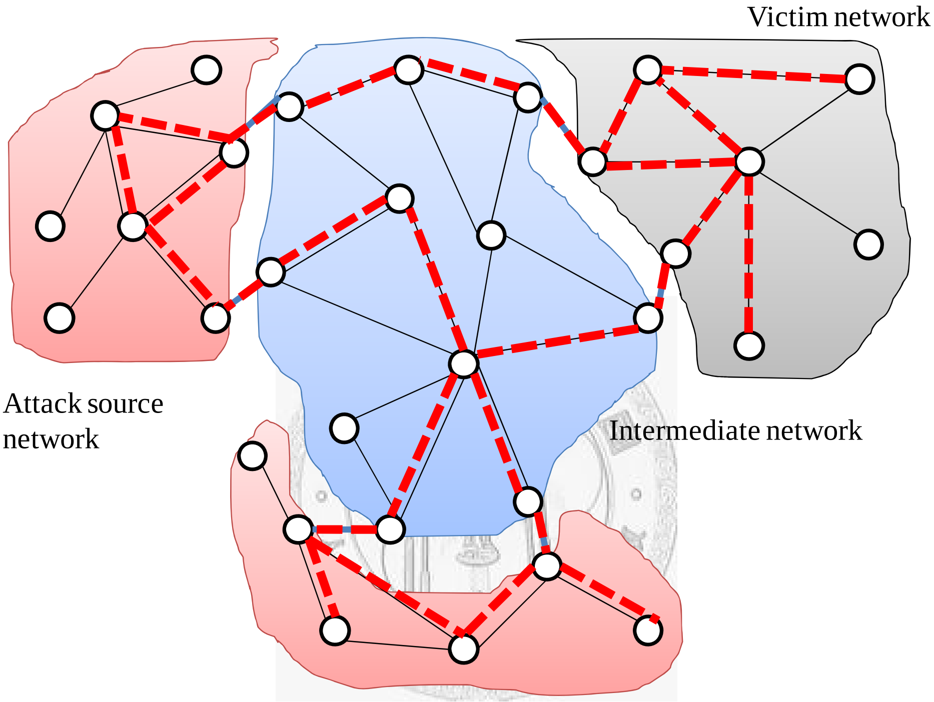


Figure 1 - 3 Defenses by Network Areas

1.3.2 Characteristics of the Network

W. E. Leland et al. [10] discuss the self-similarity of Ethernet network, and the degree of self-similarity is measured by Hurst parameter. The mechanisms to estimate Hurst parameter are already well studied. Generally after tracing the network traffic, the traffic rate (packets/time unit) is viewed as a time series and a statistic approach is used to calculate the Hurst parameter, which measures the degree of self-similarity. Usually the Hurst parameter (H) is between 0 and 1, if $0.5 < H \leq 1$ then we claim the traffic is self-similar. The Ethernet traffic is observed and the value of H is between 0.7~0.8 [10]. In addition, the mixed normal and abnormal traffic is also self-similar [11] [14] (Figure 1 - 4), and measured by Hurst parameter (H). The traffic with self-similarity will impact the performance, including transmission delay and delay jitter [12] [13]. Because of self-similarity, the network traffic is bustier than some typical traffic model, such as Poisson arrival process. The present traffic models are constructed with self-similarity or near self-similarity. In queueing theory, the S/M/1 and MMPP/M/1 models are proposed to simulate the real network traffic, where S means self-similar arrival process and MMPP is Markov Modulated Poisson Process [10]. The performance can be analyzed more accurate under an appropriate traffic model. The impacts of self-similar network traffic on queueing delay raise the congestion control problem, it is important to analyze the network performance with self-similar traffic [13].

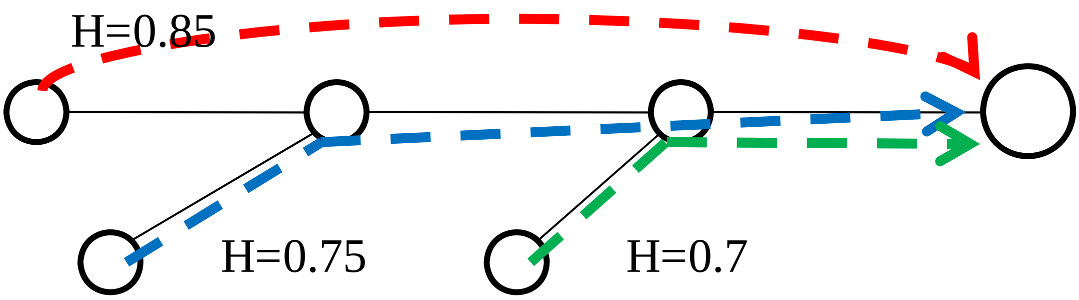


Figure 1 - 4 Hurst Parameter Value of Aggregate Flow

1.3.3 Network Survivability

J. C. Knight et al. [17] defined the survivability of network and D. M. Nicol et al. surveyed different types of measures of survivability [16]. A network cannot easily to be claimed survival or failed, the degree of the survivability of a network can be measured by several ways, including connectivity, availability, reliability, and performability. Different types of attackers have obviously different impact on the survivability of the network, especially the harder attackers, which impact the network most [23]. The survivability of network in our model is considered by performance measure. The QoS should be satisfied under DDoS attacks, and then we claim the network is survivable.

1.4 Proposed Approach

In this paper, a max-min mathematical model is proposed to describe the routing assignment and budget allocation of a network administrator and DDoS attacks strategies of an attacker. After solving the problem optimally, we can provide a guide line for network administrator to resist the abnormal traffic produced by DDoS attacks.

The primal max-min problem is formulated as a mixed integer and linear programming (MILP) problem, where the objective of problem is to maximize the total budget used by network administrator to resist the attacks, subject to different level of attacks. The total budget is derived from the inner problem, which is formulated as another MILP problem. The objective of inner problem is to minimize the total defense budget used to resist attacks, subject to QoS requirements. We proposed Lagrangean Relaxation method, which is conjunction with the subgradient method [18] [19], to solve RB problem. Furthermore, a subgradient-based heuristic, which adjusted the attacks strategies according to budget allocation by network administrator, is proposed to solve the primal max-min problem.

1.5 Thesis Organization

The remainder of the thesis is organized as follows. In Chapter 2, MILP formulations of primal max-min and RB problems are described. In Chapter 3, solution approaches to the problems are proposed. In Chapter 4, the computational results of the problems are presented. The last chapter, Chapter 5, we make conclusions and indicate the directions of future work.

Chapter 2 Problem Formulation

2.1 Problem Description

The problem we address is that how a network administrator defends against DDoS attacks by routing assignment and resource allocation under the QoS constraints. When the AS is suffered by DDoS attacks, the abnormal traffic and overwhelming quantity of packets consume the key resources of the AS. The network administrator would like to maintain the QoS in an acceptable level by using routing assignment, which will prevent too much traffic load in the same communication links, and allocating defense budget to network components, such as bandwidth, CPU power, and server buffer, in order to enhance the communication quality. In the mean time, the objective of network administrator is to minimized total defense budget to satisfy the QoS constraints for each O-D (origin-destination) pair.

The attacker outside the AS will attack the network by DDoS attacks, which send overwhelming packets. Instead of link and node attacks, the objective of the attacker is to exhaust the resources of the network by deciding the destination, which entry node to be passed, and the volume of each attack flow. The meaning of exhausting the resources of the network indicates maximizing the total defense budget used by the network administrator.

It is not trivial for both attacker and network administrator to make decisions to achieve their objectives, and therefore we proposed a mathematical model to solve this problem. After solving the problem, we expected to provide a guide line for network administrator to defense the attacks when the attacker uses different attack strategies. In order to make the model more realistic, we also consider that the network traffic has self-similarity, measured by Hurst parameter, which impacts the QoS of the network.

2.2 Problem Formulation

We model the problem as a max-min problem. The inner problem represents that for a given DDoS attack strategy, the defender uses routing assignment and budget allocation (RB) decision variables to minimize the total defense budget under QoS constraints. The outer problem represents that for a given routing assignment and budget allocation strategy, the attacker uses DDoS attack decision variables, which determine the volume of abnormal traffic to designate destination from specific entry node, to maximize the total budget. We formulate the max-min problem as an attack flow adjustment versus routing assignment and budget allocation (AFRB) problem.

The AS can be modeled as a graph, and it has several entry nodes, common nodes, dummy nodes, and directed links. Besides physical directed links, we use the node splitting technique to consider the node level communication. Therefore, each node generates a virtual link. For the convenience of modeling, we also assume that each entry node will be assigned two dummy nodes; one represents attack source, and the other represents normal external traffic source. All dummy nodes will be viewed as in the AS (Figure 2-1). The attacker executed DDoS attacks and sent specific volume of abnormal traffic to designate destination nodes via different entry nodes, and then the defender tried to defend against attacks by routing assignment and budget allocation (Figure 2-2), (Figure 2-3). Defender also wanted to satisfy links and nodes capacity constraints and QoS requirements under the attacks (Figure 2-4).

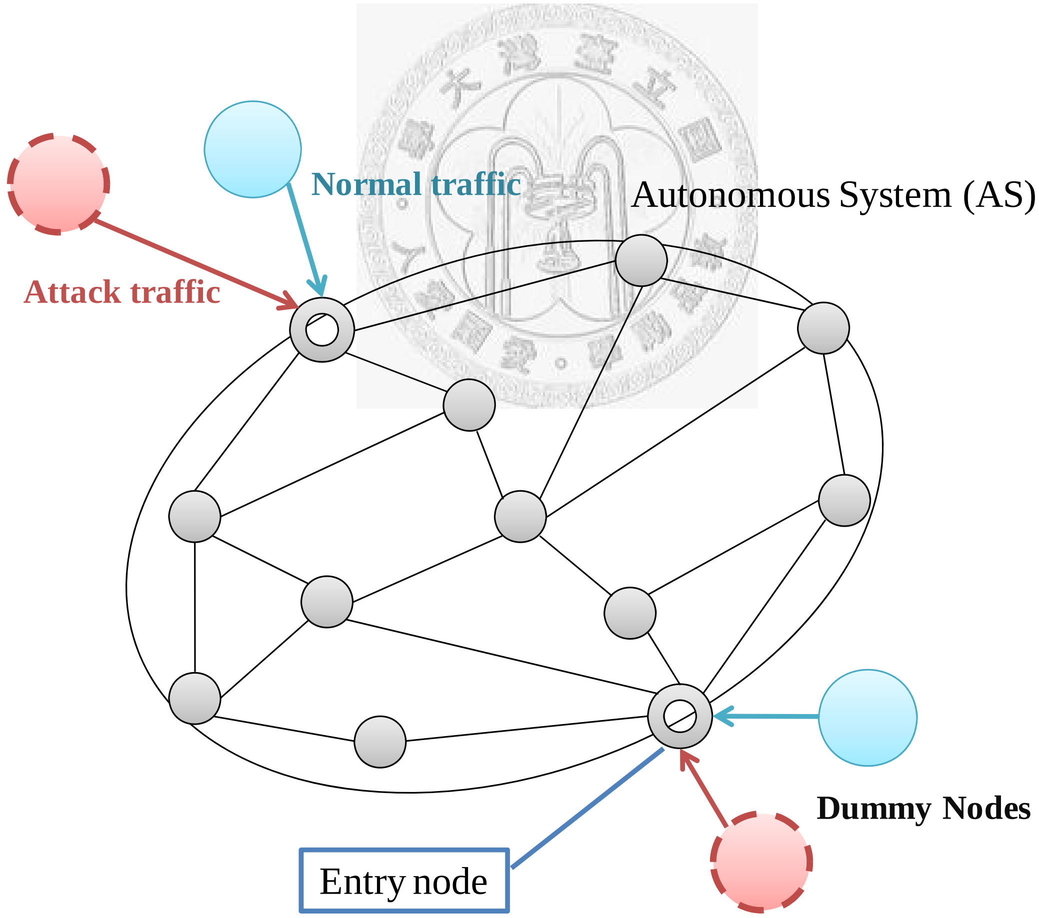


Figure 2 - 1 Graph of the Autonomous System (AS)

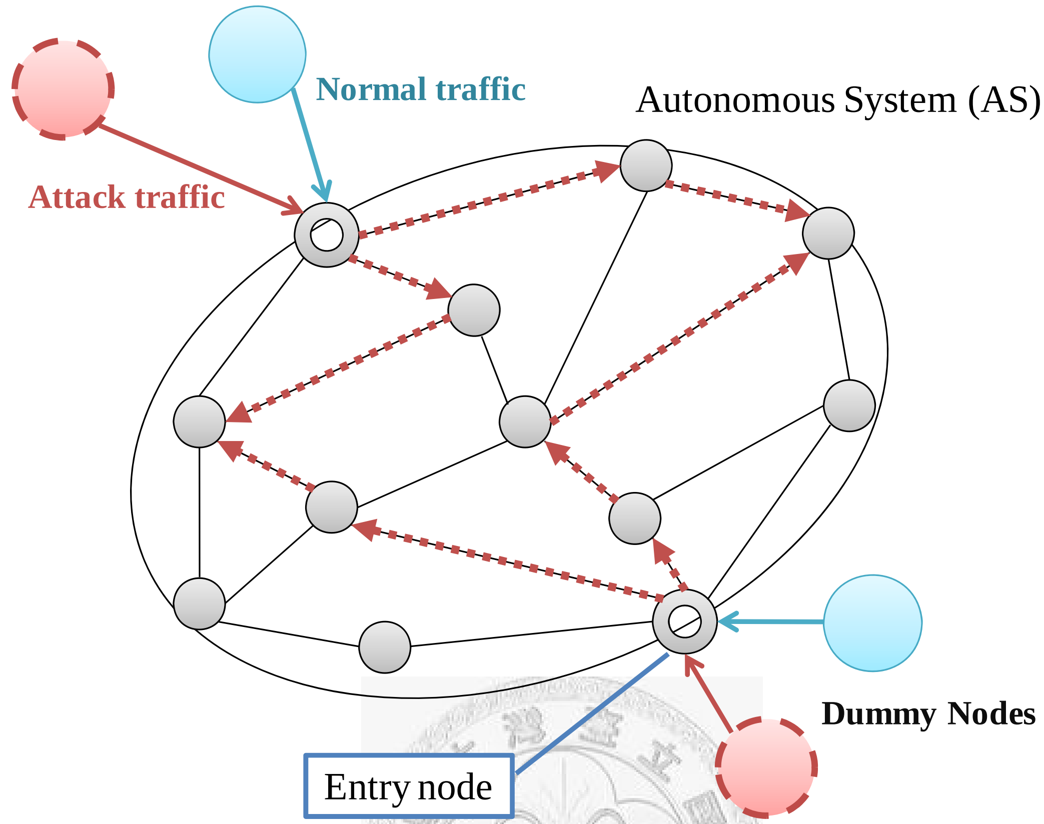


Figure 2 - 2 Attacker Executed DDoS Attacks

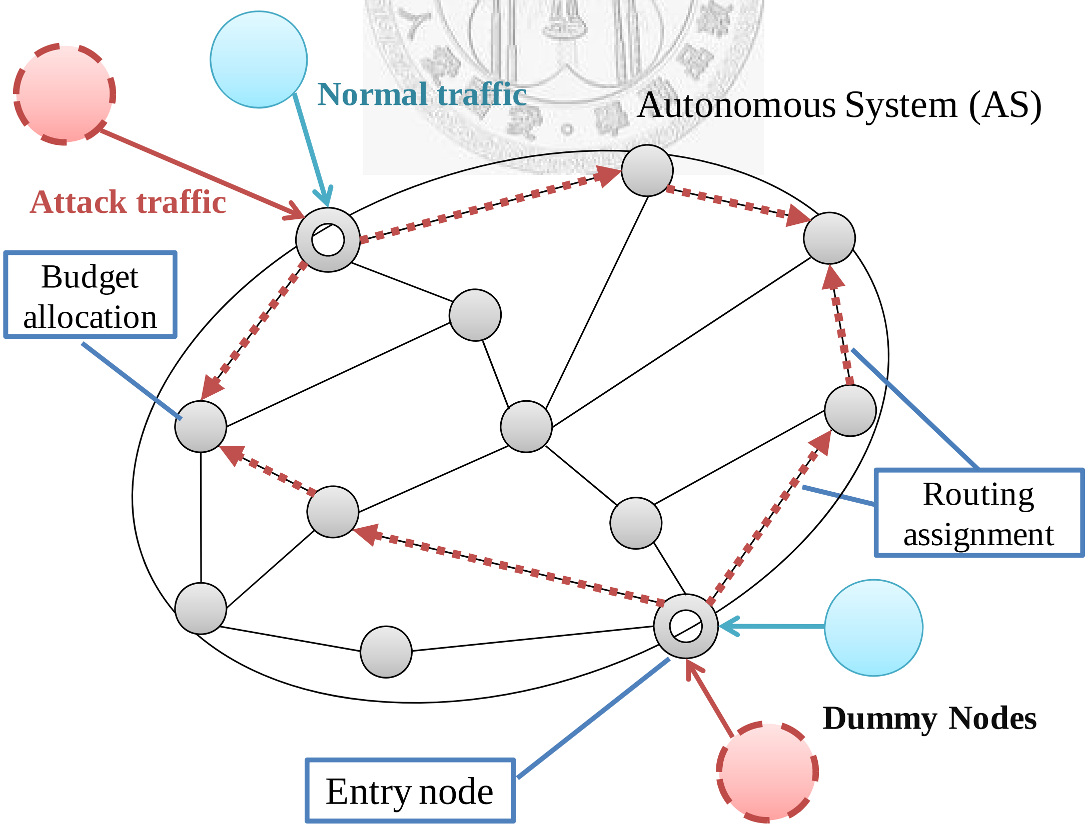


Figure 2 - 3 Defender Decided Routing Assignment and Budget Allocation

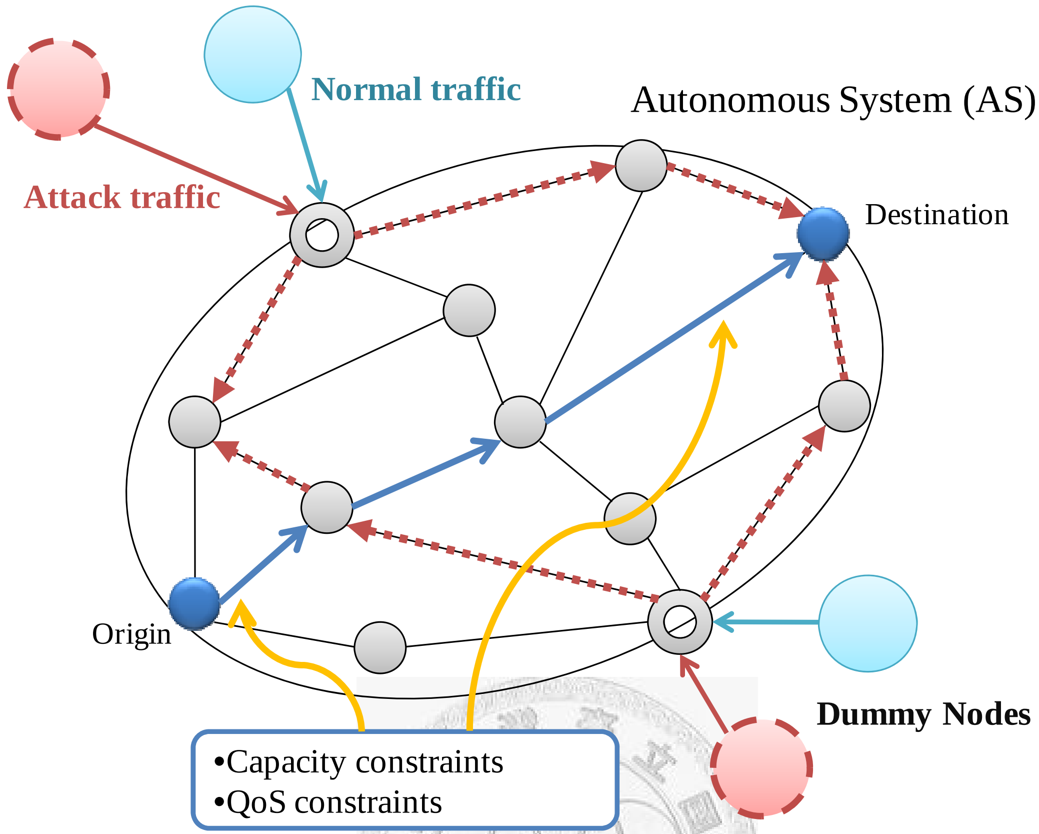
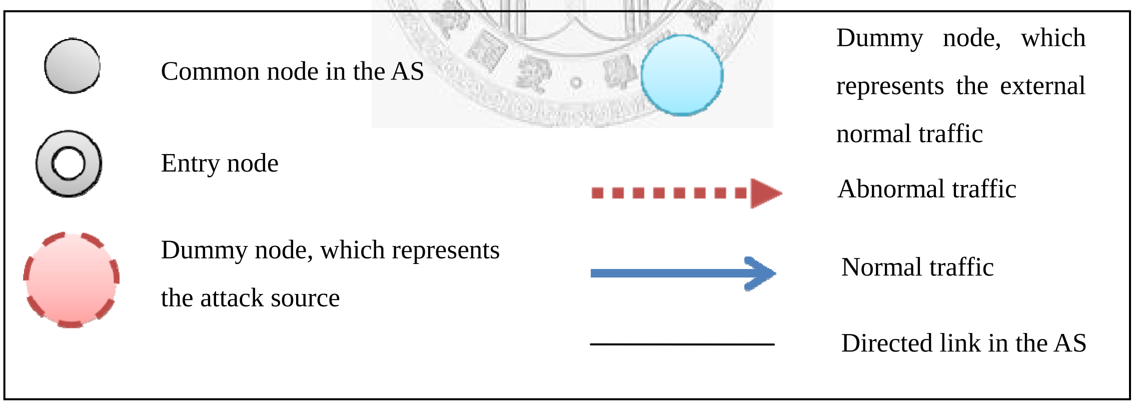


Figure 2 - 4 Requirements of Capacity and QoS



In this scenario, defender would like to select an optimal path for each O-D pair to transmit data and allocate budget to nodes or links whose capacities need to be enhanced. In the mean time, attacker wants to use abnormal traffic, which is well designed to specific destination, to violate the QoS and maximize total defense budget.

The assumptions and descriptions of the model are given in Table 2-1.

Table 2 - 1 Problem Assumption and Description

Assumptions
1. The network administrator can decide the routing assignment of the autonomous system (AS).
2. The network administrator can allocate the budget to network components to enhance the bandwidth, buffer, and CPU power.
3. For each O-D pair, the network administrator will select an optimal path to transmit the data under QoS requirements.
4. Both attacker and administrator have complete information of the AS.
5. Instead of link and node attacks, the objective of attacker, who is outside the AS, is to exhaust the resources of the AS.
6. Attack flows can enter the AS via one or many entry nodes.
7. The destination node and traffic volume of each attack flow are decided by attacker.
8. The traffic has self-similarity, which is measured by Hurst parameter.
Given
1. The network topology
2. The end-to-end normal traffic requirements
3. The end-to-end delay QoS requirements

4. The estimated Hurst parameter of the traffic for each O-D pair

Objective

- To maximize the minimized total defense budget

Subject to

1. Routing constraints
2. Link and node capacity constraints
3. End-to-end delay QoS constraints
4. Characteristics of the Hurst parameter

To determine

Defender:

1. The budget allocation strategy
2. The routing assignment of the AS

Attacker:

For each attack flow:

1. The volume of abnormal traffic
2. Destination node
3. Which entry node to be passed

We model the problem above as a max-min mathematical programming problem. The given parameters are defined in Table 2-2.

Table 2 - 2 Given Parameters of the Model

Given Parameters	
Notation	**Description**
N	The index set of all nodes in the autonomous system (AS)
L	The set of directed communication links, $L = L_1 \cup L_2$
L_1	The set of directed communication links, and each link is between two nodes
L_2	The set of virtual links between two splitting nodes for all nodes in the AS
W	The set of all Origin-Destination (O-D) pairs
W_{att}	The set of O-D pairs, and all the source nodes are attack source nodes, where $W_{att} \subset W$
P_w	The set of all candidate paths of an O-D pair *w*, where $w \in W$
δ_{pl}	The indicator function which is 1 if *l* is on the path *p* and 0 otherwise, where $p \in P_w$, $w \in W$
B_l	The set of budget configurations of a link *l*, where $l \in L$
γ_{att}	Total abnormal traffic produced by attacker
β_w	(packets/sec), the traffic requirement for O-D pair *w*, $w \in W - W_{att}$
D_w	The maximum allowable end-to-end delay for O-D pair *w*, $w \in W - W_{att}$
H_w	The Hurst parameter to measure the degree of self-similarity of the traffic for O-D pair *w*, where $w \in W$
H_{LB}	The Hurst parameter, which is a lower bound, to denote the degree of self-similarity of a link

The set L_2 is composed of virtual links which are generated by node splitting (Figure 2 - 5).

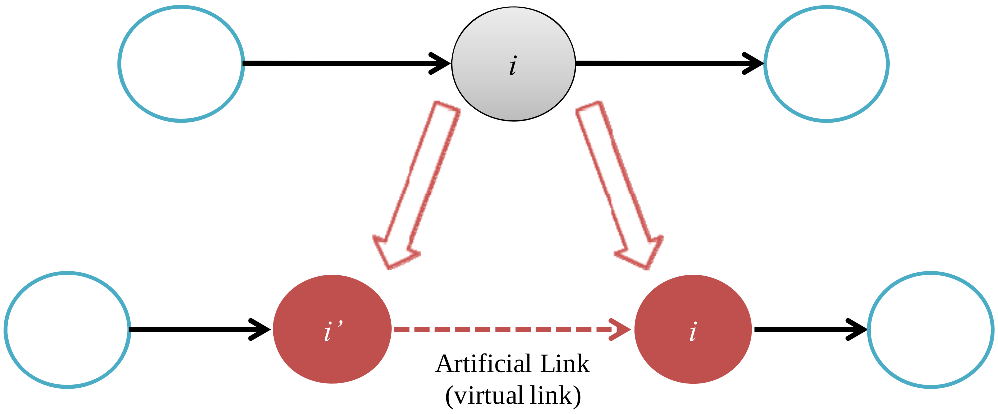


Figure 2 - 5 Node Splitting

Table 2 - 3 Decision Variables of the Model

Decision Variables	
Notation	**Description**
γ_w	Abnormal traffic from an attack source to a designated destination, produced by the attacker, where $w \in W_{att}$
b_l	The budget allocation to directed link *l*, where $b_l \in B_l$ and $l \in L$
g_l	The aggregate traffic flow on link *l*, $l \in L$
c_l	(packets/sec), the capacity of each link $l \in L$, which is equal to $\hat{c}_l(b_l)$
d_l	The mean traffic delay of each link $l \in L$, which is equal to function $\hat{d}_l(c_l,\ g_l,\ H_l)$
H_l	The Hurst parameter to measure the degree of self-similarity of the aggregate flow on directed link *l*, $l \in L$ (the aggregate flow consists of independent traffic sources)
x_p	A routing decision variable which is 1 when path $p \in P_w$ is used to transmit the packets by O-D pair *w*, where $w \in W$, and 0 otherwise

t_{wl}	An auxiliary decision variable is 1 if l is used by an O-D pair w and 0 otherwise, where $l \in L$, $w \in W$

In the primal max-min problem, attacker can control the decision variable γ_w, which represents the volume of abnormal traffic from one attack source to designated destination. It is noteworthy that when attacker decided the attack source, the entry node to be passed is also determined because each attack source is modeled as a dummy node linked with one entry node. The model is formulated as the following problem (IP1).

Objective function:

$$Z_{\text{IP1}} = \max_{\gamma_w} \min_{b_l, x_p} \left[\sum_{l \in L} b_l \right] \qquad \textbf{(IP1)}$$

Subject to:

$$b_l \in B_l \qquad \forall l \in L \qquad \text{(IP1.1)}$$

$$\gamma_{att} = \sum_{w \in W_{att}} \gamma_w \qquad \text{(IP1.2)}$$

$$\gamma_w \geq 0 \qquad \forall w \in W_{att} \qquad \text{(IP1.3)}$$

$$\sum_{w \in W - W_{att}} \sum_{p \in P_w} x_p \delta_{pl} \beta_w + \sum_{w \in W_{att}} \sum_{p \in P_w} x_p \delta_{pl} \gamma_w = g_l \qquad \forall l \in L \qquad \text{(IP1.4)}$$

$$0 \leq g_l \leq c_l = \hat{c}_l(b_l) \qquad \forall l \in L \qquad \text{(IP1.5)}$$

$$\sum_{p \in P_w} x_p \delta_{pl} H_w \leq H_l \qquad \forall w \in W,\ l \in L \qquad \text{(IP1.6)}$$

$$H_l \in \left\{ H_{LB},\ \sum_{p\in P_w} x_p \delta_{pl} H_w \right\} \qquad \forall w\in W,\ l\in L \qquad \text{(IP1.7)}$$

$$d_l = \hat{d}_l(c_l, g_l, H_l) \qquad \forall l\in L \qquad \text{(IP1.8)}$$

$$\sum_{l\in L} d_l \sum_{p\in P_w} x_p \delta_{pl} \le D_w \qquad \forall w\in W \qquad \text{(IP1.9)}$$

$$\sum_{p\in P_w} x_p = 1 \qquad \forall w\in W \qquad \text{(IP1.10)}$$

$$\sum_{p\in P_w} x_p \delta_{pl} = t_{wl} \qquad \forall w\in W,\ l\in L \qquad \text{(IP1.11)}$$

$$x_p = 0 \text{ or } 1 \qquad \forall p\in P_w, \forall w\in W \qquad \text{(IP1.12)}$$

$$t_{wl} = 0 \text{ or } 1 \qquad \forall w\in W,\ l\in L. \qquad \text{(IP1.13)}$$

Explanation of the Mathematical Formulation:

- Objective function: The objective is to maximize the minimized total defense budget $\sum_{l\in L} b_l$. In the RB problem, defender tries to minimize the total defense budget allocated to the network. In the AFRB problem, the attacker tries to maximize the total defense budget.
- Constraint (IP1.1) indicates the budget allocated to network components is a kind of configuration, which belongs to a configuration set, $B_{l.}$
- Constraint (IP1.2) requires that the total abnormal traffic must not exceed a given value γ_{att} .

- Constraint (IP1.3) requires the abnormal traffic from an attack source to a designate destination must be nonnegative.
- Constraint (IP1.4) calculates the aggregate flow on link *l*, including the normal and abnormal traffic, and internal and external traffic as well.
- Constraint (IP1.5) denotes that the aggregate flow on link *l* must not exceed the capacity, which is a function of b_l.
- Constraint (IP1.6) estimates the Hurst parameter value of aggregate flow on link *l*, and the value is no smaller than the maximum Hurst parameter value of independent traffic sources.
- Constraint (IP1.7) denotes the Hurst parameter value of aggregate flow on link *l* belongs to a set, which is composed of Hurst parameter values of independent traffic sources and a lower bound.
- Constraint (IP1.8) denotes that the mean traffic delay on link *l* is a function of three parameters, capacity, aggregate flow, and Hurst parameter value.
- Constraint (IP1.9) requires the transmission delay of each O-D pair must not exceed the end-to-end delay QoS requirement.
- Constraint (IP1.10) enforces that each O-D pair can choose only one path from the candidate paths to transmit data.
- Constraint (IP1.11) binds the relation among t_{wl}, x_p, and δ_{pl}, so that we can

use this relation to simplify the problem and make it easier to solve.

- Constraint (IP1.12) enforces that if a path is chosen, then the $x_p = 1$, otherwise $x_p = 0$.
- Constraint (IP1.13) enforces that if a link is chosen by O-D pair *w*, then the $t_{wl} = 1$, otherwise $t_{wl} = 0$.

2.3 Problem Formulation of the RB Problem

For solving the primal problem, we try to analyze the RB problem first. The meaning of RB problem is that given an attack pattern by attacker, the defender has to minimize the total defense budget by adjusting the routing assignment and budget allocation. The QoS requirements also must be satisfied when the attacker uses different attack patterns each time. The problem assumptions of RB problem are the same as the original max-min problem. The given parameters are defined in Table 2-4.

Table 2 - 4 Given Parameters of RB Problem

Given Parameters	
Notation	**Description**
N	The index set of all nodes in the autonomous system (AS)
L	The set of directed communication links, $L = L_1 \cup L_2$

L_1	The set of directed communication links, and each link is between two nodes
L_2	The set of virtual links between two splitting nodes for all nodes in the AS
W	The set of all Origin-Destination (O-D) pairs
W_{att}	The set of O-D pairs, and all the source nodes are attack source nodes, where $W_{att} \subset W$
P_w	The set of all candidate paths of an O-D pair *w*, where $w \in W$
δ_{pl}	The indicator function which is 1 if *l* is on the path *p* and 0 otherwise, where $p \in P_w$, $w \in W$
B_l	The set of budget configurations of a link *l*, where $l \in L$
α_w	(packets/sec), the traffic from O-D pair *w*, where $w \in W$
D_w	The maximum allowable end-to-end delay for O-D pair *w*, $w \in W - W_{att}$
H_w	The Hurst parameter to measure the degree of self-similarity of the traffic for O-D pair *w*, where $w \in W$
H_{LB}	The Hurst parameter, which is a lower bound, to denote the degree of self-similarity of a link

The abnormal traffic γ_w, produced by attacker to designate destination by specific entry node becomes given parameter of RB problem now. Furthermore, we can simplify two given parameters γ_w and β_w into one given parameter α_w, which denotes the traffic of O-D pair *w*. The decision variables of defender are defined in Table 2-5.

Table 2 - 5 Decision Variables of RB Problem

Decision Variables	
Notation	**Description**
b_l	The budget allocation to directed link *l*, where $b_l \in B_l$ and $l \in L$
g_l	The aggregate traffic flow on link *l*, $l \in L$
c_l	(packets/sec), the capacity of each link $l \in L$, which is equal to $\hat{c}_l(b_l)$
d_l	The mean traffic delay of each link $l \in L$, which is equal to function $\hat{d}_l(c_l, g_l, H_l)$
H_l	The Hurst parameter to measure the degree of self-similarity of the aggregate traffic flow on directed link *l*, $l \in L$ (aggregate traffic flow consists of independent traffic sources)
x_p	A routing decision variable which is 1 when path $p \in P_w$ is used to transmit the packets by O-D pair *w*, where $w \in W$, and 0 otherwise
t_{wl}	An auxiliary decision variable is 1 if *l* is used by an O-D pair *w* and 0 otherwise, where $l \in L$, $w \in W$

The network administrator has to decide the value of b_l, then the capacity of link *l* was decided. The decision variable x_p can determine which path will be used by an O-D pair. Besides, for solving the problem easier, we substituted the relation in (IP1.11) into (IP1.7) and (IP1.9) to get (IP2.5) and (IP2.7). The RB problem is formulated as (IP2).

Objective function:

$$Z_{\mathrm{IP2}} = \min_{b_l, x_p} \left[\sum_{l \in L} b_l \right] \quad \textbf{(IP2)}$$

Subject to:

$$b_l \in B_l \qquad \forall l \in L \qquad \text{(IP2.1)}$$

$$\sum_{w \in W} \sum_{p \in P_w} x_p \delta_{pl} \alpha_w = g_l \qquad \forall l \in L \qquad \text{(IP2.2)}$$

$$0 \le g_l \le c_l = \hat{c}_l(b_l) \qquad \forall l \in L \qquad \text{(IP2.3)}$$

$$\sum_{p \in P_w} x_p \delta_{pl} H_w \le H_l \qquad \forall w \in W,\ l \in L \qquad \text{(IP2.4)}$$

$$H_l \in \{H_{LB},\ t_{wl} H_w\} \qquad \forall w \in W,\ l \in L \qquad \text{(IP2.5)}$$

$$d_l = \hat{d}_l(c_l, g_l, H_l) \qquad \forall l \in L \qquad \text{(IP2.6)}$$

$$\sum_{l \in L} d_l t_{wl} \le D_w \qquad \forall w \in W \qquad \text{(IP2.7)}$$

$$\sum_{p \in P_w} x_p = 1 \qquad \forall w \in W \qquad \text{(IP2.8)}$$

$$\sum_{p \in P_w} x_p \delta_{pl} = t_{wl} \qquad \forall w \in W,\ l \in L \qquad \text{(IP2.9)}$$

$$x_p = 0 \text{ or } 1 \qquad \forall p \in P_w, \forall w \in W \qquad \text{(IP2.10)}$$

$$t_{wl} = 0 \text{ or } 1 \qquad \forall w \in W,\ l \in L. \qquad \text{(IP2.11)}$$

Explanation of the Mathematical Formulation:

- Objective function: the objective function is to minimize the total defense budget allocated to network components.
- Constraint (IP2.1) is the same as Constraint (IP1.1) in the original max-min problem (IP1).
- Constraints (IP2.2) ~ (IP2.11) are the same as Constraints (IP1.4) ~ (IP1.13) in the original max-min problem.

Chapter 3 Solution Approaches

3.1 Lagrangean Relaxation Method

The Lagrangean relaxation method was first used to solve large-scale mathematical programming problems during the 1970s [19]. An important concept of the method is “decomposition”, which reduces the complexities and difficulties of the primal problem. Because of its efficiency and effectiveness in solving many complicate programming problems, Lagrangean relaxation has become one of the most popular tools to solve optimization problem. The applications of it include integer programming, linear programming combinatorial optimization, and non-linear programming problems. The performance of Lagrangean relaxation is excellent, especially in solving large-scale mathematical programming problems [18].

When we are solving some difficult programming problems, the problems can be modeled as a set of constraints and then we apply Lagrangean relaxation method to transform the problem become an easier solvable form. In this method, we first relax some constraints, and add them into the objective function with associated Lagrangean multipliers (μ). The concept is as if we add some penalties to primal problem, when we violate the relaxed constraints, the effect of the penalties will occur. After relaxation, we get a new objective function and the other constraints, and the new problem (LR problem) is formed. Then we can decompose the LR problem into several subproblems, and each subproblem can be optimally solved by using some existing algorithms.

Taking minimization problem as an example, the LR problem will provide a lower bound ($Z_D(\mu)$) to primal problem. We hope that the lower bound can achieve the objective function value of primal problem as tight as possible, so we derive another new problem, which is called Lagrangean dual problem. After tuning the Lagrangean multiplier (μ) iteration by iteration, we can get a tightest lower bound of primal problem.

From the above procedure, we always can get some hints of solving primal problem. We then apply some heuristic approaches to get the feasible solutions, which provide an upper bound (UB) of the objective function value of primal problem. Intuitively, the optimal objective function value of primal problem is between lower bound (objective function value of LR problem) and upper bound (objective function value of primal problem).

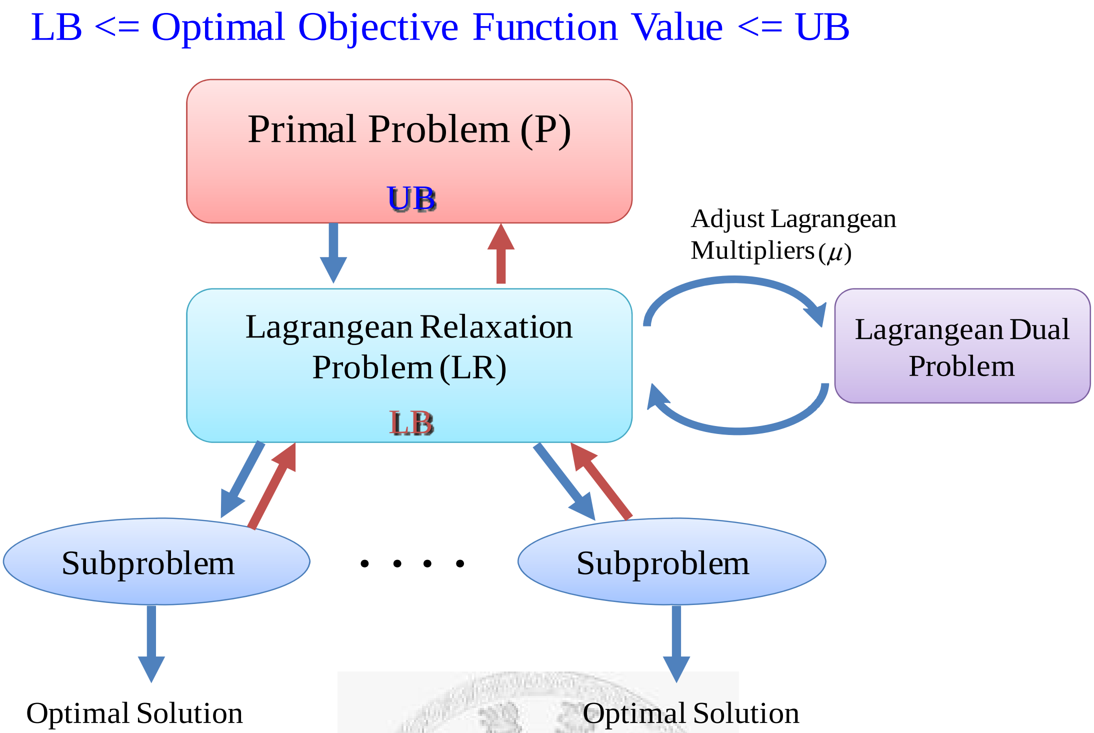


Figure 3 - 1 Concept of LR

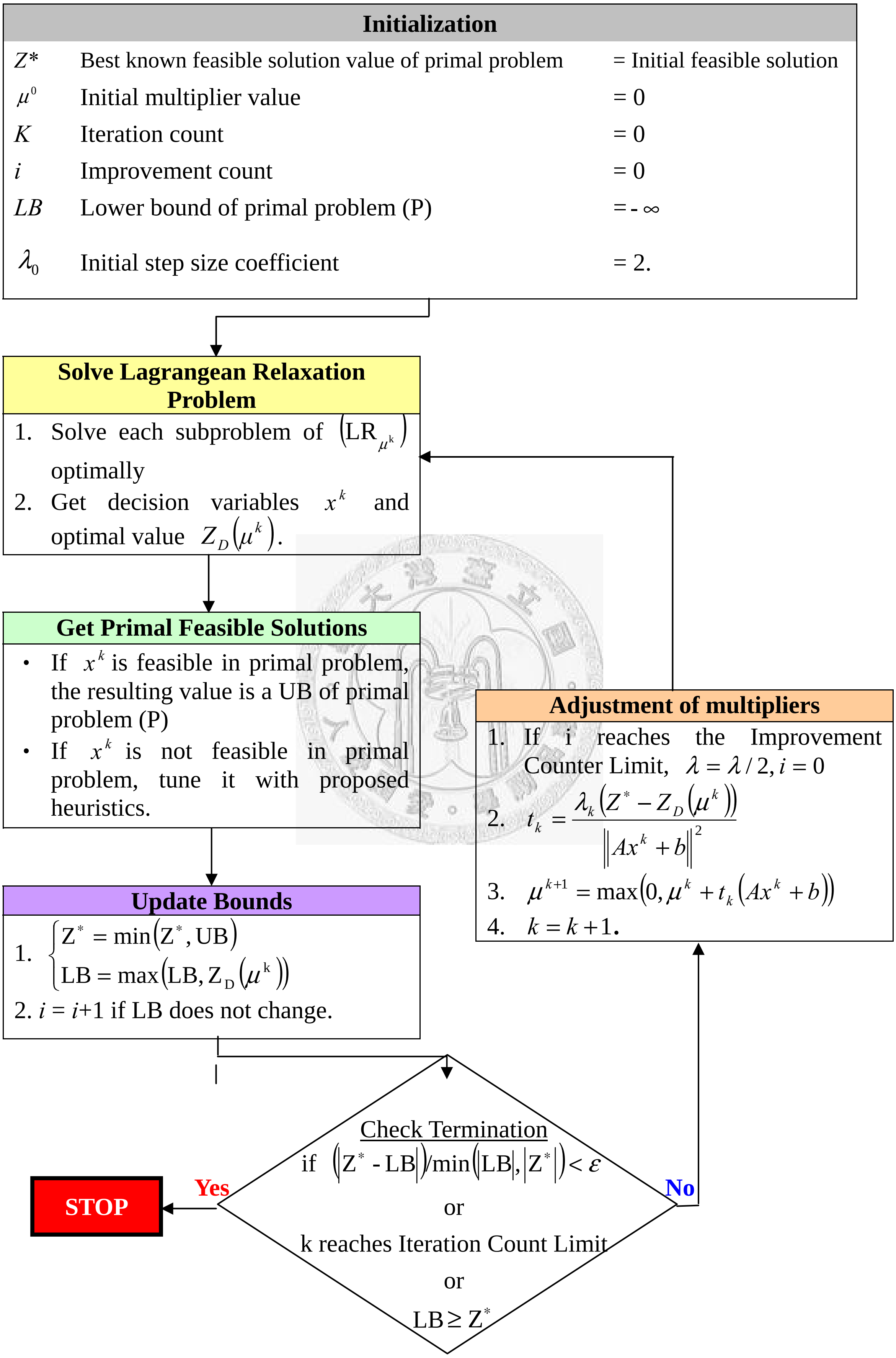


Figure 3 - 2 Lagrangean Relaxation Method Procedure

3.2 The Solution Approach for the RB Problem

After reformulate the problem as (IP2), we apply the Lagrangean relaxation to solve the problem. Constraints (IP2.2) and (IP2.9) can be relaxed to $\sum_{w \in W} \sum_{p \in P_w} x_p \delta_{pl} \alpha_w \le g_l$ and $\sum_{p \in P_w} x_p \delta_{pl} \le t_{wl}$ without violating the original meaning of (IP2). Then we relax constraints (IP2.2), (IP2.4), (IP2.7), and (IP2.9) and add them, multiplied with associated Lagrangean multipliers, to the objective function of (IP2). The following Lagrangean relaxation problem (LR1) is obtained.

3.2.1 Lagrangean Relaxation

(LR1)

$$
\begin{aligned}
Z_{\mathrm{D}}(\mu^1, \mu^2, \mu^3, \mu^4) = \min_{b_l, x_p} & \left[\sum_{l \in L} b_l \right. \\
& + \sum_{l \in L} \mu_l^1 \left[\sum_{w \in W} \sum_{p \in P_w} x_p \delta_{pl} \alpha_w - g_l \right] \\
& + \sum_{w \in W} \sum_{l \in L} \mu_{wl}^2 \left[\sum_{p \in P_w} x_p \delta_{pl} H_w - H_l \right] \\
& + \sum_{w \in W} \mu_w^3 \left[\sum_{l \in L} \hat{d}_l (c_l, g_l, H_l) t_{wl} - D_w \right] \\
& \left. + \sum_{w \in W} \sum_{l \in L} \mu_{wl}^4 \left[\sum_{p \in P_w} x_p \delta_{pl} - t_{wl} \right] \right]
\end{aligned}
$$

Subject to:

$$b_l \in B_l \qquad \forall l \in L \qquad \text{(LR1.1)}$$

$$H_l \in \{H_{LB},\ t_{wl} H_w\} \qquad \forall w \in W,\ l \in L \qquad \text{(LR1.2)}$$

$$0 \le g_l \le c_l = \hat{c}_l(b_l) \qquad \forall l \in L \qquad \text{(LR1.3)}$$

$$\sum_{p \in P_w} x_p = 1 \qquad \forall w \in W \qquad \text{(LR1.4)}$$

$$x_p = 0 \text{ or } 1 \qquad \forall p \in P_w, \forall w \in W \qquad \text{(LR1.5)}$$

$$t_{wl} = 0 \text{ or } 1 \qquad \forall w \in W,\ l \in L. \qquad \text{(LR1.6)}$$

The Lagrangean multipliers μ_1 and μ_3 are one dimensional vectors, and μ_2 and μ_4 are two dimensional vectors, all of them are nonnegative. To solve Lagrangean relaxation problem, we decompose (LR1) into two independent subproblems.

Subproblem 1 (related to decision variable x_p)

$$Z_{\text{Sub1}}(\mu^1,\mu^2,\mu^4)=\min \sum_{l\in L}\mu_l^1\left[\sum_{w\in W}\sum_{p\in P_w}x_p\delta_{pl}\alpha_w\right] \quad \textbf{(Sub1)}$$

$$+\sum_{w\in W}\sum_{l\in L}\mu_{wl}^2\left[\sum_{p\in P_w}x_p\delta_{pl}H_w\right]+\sum_{w\in W}\sum_{l\in L}\mu_{wl}^4\left[\sum_{p\in P_w}x_p\delta_{pl}\right]$$

$$=\min\left(\sum_{w\in W}\sum_{p\in P_w}\sum_{l\in L}x_p\delta_{pl}\left[\mu_l^1\alpha_w+\mu_{wl}^2H_w+\mu_{wl}^4\right]\right)$$

Subject to:

$$\sum_{p\in P_w}x_p=1 \qquad \forall w\in W \qquad \text{(LR1.4)}$$

$$x_p=0\text{ or }1 \qquad \forall p\in P_w,\forall w\in W. \qquad \text{(LR1.5)}$$

(Sub1) can be further decomposed into $|W|$ independent shortest path problem, with nonnegative arc weight $\left(\mu_l^1\alpha_w+\mu_{wl}^2H_w+\mu_{wl}^4\right)$. The arc weight is composed of traffic, burstiness of each O-D pair and μ_4, which implies that the we will select a frequently passed path from iteration to iteration. Each shortest path problem can be solved by Dijkstra's algorithm. The computational complexity is $O\left(|N|^2\right)$ for each source node.

Subproblem 2 (related to decision variables b_l, g_l, t_{wl}, H_l)

(Sub2)

$$Z_{\text{Sub2}}(\mu^1,\mu^2,\mu^3,\mu^4)=\min\ \left[\sum_{l\in L} b_l\right]$$

$$+\sum_{l\in L}\mu_l^1[-g_l]+\sum_{w\in W}\sum_{l\in L}\mu_{wl}^2[-H_l]$$

$$+\sum_{w\in W}\mu_w^3\left[\sum_{l\in L}\hat{d}_l(c_l,g_l,H_l)t_{wl}\right]+\sum_{w\in W}\sum_{l\in L}\mu_{wl}^4[-t_{wl}]$$

Rewrite to:

$$\min\ \sum_{l\in L}\begin{bmatrix} b_l+\left(-\mu_l^1\right)g_l+\sum_{w\in W}\left(-\mu_{wl}^2\right)H_l \\ +\sum_{w\in W}\left(\hat{d}_l(c_l,g_l,H_l)\mu_w^3-\mu_{wl}^4\right)t_{wl}\end{bmatrix}$$

Subject to:

$b_l\in B_l \qquad \forall l\in L \qquad$ (LR1.1)

$H_l\in\{H_{LB},\ t_{wl}H_w\} \qquad \forall w\in W,\ l\in L \qquad$ (LR1.2)

$0\le g_l\le c_l=\hat{c}_l(b_l) \qquad \forall l\in L \qquad$ (LR1.3)

$t_{wl}=0$ or $1 \qquad \forall w\in W,\ l\in L \qquad$ (LR1.6)

(Sub2) can be further decomposed into $|L|$ independent subproblems, for each link *l* we obtain a problem as (Sub2.1).

Subproblem 2.1 (for each $l \in L$)

$$\min \begin{bmatrix} b_l + \left(-\mu_l^1\right) g_l + \sum_{w \in W} \left(-\mu_{wl}^2\right) H_l \\ + \sum_{w \in W} \left(\hat{d}_l(c_l, g_l, H_l) \mu_w^3 - \mu_{wl}^4 \right) t_{wl} \end{bmatrix} \quad \textbf{(Sub2.1)}$$

Subject to:

$$b_l \in B_l \quad \text{(LR1.1)}$$

$$H_l \in \{H_{LB},\ t_{wl} H_w\} \quad \forall w \in W \quad \text{(LR1.2)}$$

$$0 \le g_l \le c_l = \hat{c}_l(b_l) \quad \text{(LR1.3)}$$

$$t_{wl} = 0 \text{ or } 1 \quad \forall w \in W. \quad \text{(LR1.6)}$$

To solve the subproblem (Sub2.1), we first exhaustively assign the values of b_l and H_l, and next (LR1.2) is relaxed to $H_l \in \{H_{LB},\ H_w\}$ $\forall w \in W$, and substitute the (LR1.4) into objective function. The form of $\hat{d}_l(c_l, g_l, H_l)$ is provided by [13] as follows:

Table 3 - 1 G/M/1 Queueing Delay Approximation

Notation	Description
W_q	The average queueing delay of a G/M/1 queueing system
δ	A function of utilization (ρ) and Hurst Parameter (H)
H	Hurst parameter
μ	Service rate
ρ	utilization
$b_{i,H}$	A functions of H, where i = 0, 1, 2, 3
c_{xy}	coefficients

$$W_q = \frac{\delta}{\mu(1-\delta)}$$

$$\delta(\rho, H) = b_{3,H}\rho^3 + b_{2,H}\rho^2 + b_{1,H}\rho + b_{0,H}$$

$$\begin{bmatrix} b_{3,H} \\ b_{2,H} \\ b_{1,H} \\ b_{0,H} \end{bmatrix} = \begin{bmatrix} c_{33} & c_{32} & c_{31} & c_{30} \\ c_{23} & c_{22} & c_{21} & c_{20} \\ c_{13} & c_{12} & c_{11} & c_{10} \\ c_{03} & c_{02} & c_{01} & c_{00} \end{bmatrix} \begin{bmatrix} H^3 \\ H^2 \\ H \\ 1 \end{bmatrix}.$$

We let $\hat{d}_l(c_l, g_l, H_l) = W_q$, service rate = c_l, $\rho = \frac{g_l}{c_l}$, and $H = H_l$, hence we need to solve

Subproblem 2.1.1 (for each $\left(b_l \in B_l, H_l \in \{H_w, H_{LB}\}\right)$)

$$\min \left[\left(-\mu_l^1\right) g_l + \sum_{w\in W} \left(\mu_w^3 \frac{\delta}{c_l\left(1-\delta\right)} - \mu_{wl}^4 \right) t_{wl} \right] \qquad \textbf{(Sub2.1.1)}$$

Subject to:

$$0 \le g_l \le c_l = \hat{c}_l(b_l) \qquad \text{(LR1.3)}$$

$$t_{wl} = 0 \text{ or } 1 \qquad \forall w \in W \qquad \text{(LR1.7)}$$

$$\delta = b_{3,H_l}\left(\frac{g_l}{c_l}\right)^3 + b_{2,H_l}\left(\frac{g_l}{c_l}\right)^2 + b_{1,H_l}\left(\frac{g_l}{c_l}\right) + b_{0,H_l} \qquad \text{(LR1.8)}$$

$$\begin{bmatrix} b_{3,H_l} \\ b_{2,H_l} \\ b_{1,H_l} \\ b_{0,H_l} \end{bmatrix} = \begin{bmatrix} c_{33} & c_{32} & c_{31} & c_{30} \\ c_{23} & c_{22} & c_{21} & c_{20} \\ c_{13} & c_{12} & c_{11} & c_{10} \\ c_{03} & c_{02} & c_{01} & c_{00} \end{bmatrix} \begin{bmatrix} H_l^{\ 3} \\ H_l^{\ 2} \\ H_l \\ 1 \end{bmatrix}. \qquad \text{(LR1.9)}$$

We can focus on solving g_l and t_{wl} in (Sub2.1.1) where a similar problem was solved in [9].

The algorithm of solving (Sub2.1.1) is as follows:

Step1. Solve $\mu_w^3 \dfrac{\delta}{c_l\left(1-\delta\right)} - \mu_{wl}^4 = 0$ for each O-D pair *w*, call them the break points of g_l.

Step2. Sort these break points and drop infeasible values, where feasible region is defined in (LR1.3), and denoted as $g_l^1, g_l^2, \ldots g_l^n$.

Step3. At each interval $g_l^i \le g_l \le g_l^{i+1}$, t_{wl} is 1 if $\mu_w^3 \frac{\delta}{c_l(1-\delta)} - \mu_{wl}^4 \le 0$ and is 0 otherwise.

Step4. Within the interval $g_l^i \le g_l \le g_l^{i+1}$, the local minimal is either at a boundary point, g_l^i or g_l^{i+1}, or at g_l^*, where

$$\begin{cases} f\left(g_l^*\right) \le f\left(g_l\right) \\ f\left(g_l\right) = \left[-\mu_l^1 g_l + e_l \frac{\delta}{c_l(1-\delta)}\right] \\ e_l = \sum_{w \in W} \mu_w^3 t_{wl}. \end{cases}$$

To simplify finding the solution of g^*, we assume that the utilization is discrete and search the local optimal solution by increasing 0.001 of the value of utilization.

Step5. The global minimum point of (Sub2.1.1) can be found by comparing these local minimum points.

After finding the optimal solution of (Sub2.1.1), the optimal solution of (Sub2.1) can be found.

The algorithm of solving (Sub2.1) is as follows:

Step1. Assign a value to b_l

Step2. Assign a value to H_l

Step3. Solve (Sub2.1.1) for each set (b_l, H_l), and get a local minimum objective function value

Step4. Compare these local minimum objective function values, and then find the global minimum objective function value and the optimal solutions of b_l, H_l, g_l, t_{wl}.

The computational complexity of (Sub2.1) is $O\left(|B_l| \times |W|^2 \times \log|W|\right)$ for each link.

3.2.1 The Dual Problem and the Subgradient Method

To solve the above subproblems optimally, the Lagrangean Relaxation problem (LR1) can be solved optimally. According to the weak duality theorem [20], for the set of the multipliers $(\mu^1, \mu^2, \mu^3, \mu^4)$, $Z_{\text{D1}}(\mu^1, \mu^2, \mu^3, \mu^4)$ generates a Lower Bound (LB) of Z_{IP2}. Next we construct a dual problem (D1) to obtain the tightest LB and solve it by the subgradient method [18] [19].

Dual problem (D1)

$$Z_{\text{D}} = \max Z_{\text{D}}(\mu^1, \mu^2, \mu^3, \mu^4) \qquad \text{(D1)}$$

Subject to: $\mu^1, \mu^2, \mu^3, \mu^4 \geq 0$

Let a vector m be a subgradient of $Z_{\mathrm{D1}}(\mu^1, \mu^2, \mu^3, \mu^4)$. Then in iteration k of the subgradient procedure, the multiplier vector $\pi = (\mu^1, \mu^2, \mu^3, \mu^4)$ is updated from

$\pi^{k+1} = \pi^k + t^k m^k$,

and where

$$m^k(\mu^1, \mu^2, \mu^3, \mu^4) = \left(\sum_{w \in W} \sum_{p \in P_w} x_p \delta_{pl} \alpha_w - g_l,\ \sum_{p \in P_w} x_p \delta_{pl} H_w - H_l,\ \sum_{l \in L} d_l t_{wl} - D_w,\ \sum_{p \in P_w} x_p \delta_{pl} - t_{wl} \right).$$

The step size t^k is determined by

$$t^k = \lambda \frac{Z^*_{\mathrm{IP2}} - Z_{\mathrm{D}}\left(\pi^k\right)}{\left\| m^k \right\|^2}.$$

Z^*_{IP2} is the tightest upper bound (UB) of the primal objective function value found from iteration k. Note that λ is a scalar between 0 and 2, and usually initiated with the value of 2 and halved if the best objective function value does not improve within a given iterations.

3.2.2 Getting Primal Feasible Solutions

To obtain the primal feasible solutions to the primal RB problem (IP2), solutions of Lagrangean relaxation problems (LR1) are considered. For example, if a solution of (LR1) is feasible to (IP2), say, the capacity constraints and QoS constraints are satisfied, and then it is considered as a primal feasible solution to (IP2). If it is not feasible to (IP2), then we can modify it to be a feasible primal solution. Hence, a getting primal feasible solutions heuristic algorithm is developed.

The algorithm of solving (IP2) is as follows:

Initial Step. Read the information from (LR1), including:

1. Use Lagrangean multipliers μ^4 as a priority for each O-D pair
2. Assign a routing path x_p, where obtained from (LR1), for each O-D pair
3. Each O-D pair is marked as in a **Waiting Queue** with priority.
4. Construct a **Candidate Queue**, where all O-D pairs in **Candidate Queue** are viewed as sending data in the network.
5. Setup a *Max_Searching_Limit* and a *Searching_Counter*, where $Max\_Searching\_Limit = \frac{|W|^2}{4}$ and *Searching_Counter* = 0.

After initial step, we repeat the following steps, and the algorithm terminates either a feasible solution is found or no feasible solutions are found in some iterations.

Step1. Pop the front O-D pair of **Waiting Queue** to get into **Path Checking Process**.

Step2. Run **Candidate Queue Checking Process**.

Step3. Run **Searching Limit Checking Process**.

Path Checking Process (for input O-D pair)

Step1. Check whether the current candidate path of O-D pair is feasible, if it is feasible, the O-D pair is put into **Candidate Queue** and stop this process, otherwise go to next step.

Step2. Find a minimum end-to-end delay routing path for O-D pair.

Step3. Assign the budget to the path to satisfy the capacity constraints. Whether the path is feasible or not, put the O-D pair into **Candidate Queue.**

Candidate Queue Checking Process (for each O-D pair in the queue)

Step1. Construct a scenario that all O-D pairs in **Candidate Queue** are sending data, rerouting for each O-D pair to get a minimum end-to-end delay path

Step2. Check end-to-end delay constraints, if all the candidate paths in **Candidate Queue** are feasible, go to **Step5**, otherwise go to **Step3**.

Step3. For each O-D pair with infeasible candidate path, calculating the ***gain*** by adding one more unit budget for each link. The gain is defined as follows:

$gain = d_l\left(b_l\right) - d_l(b_l + 1)$, for each link l on candidate path.

Step4. Finding the maximum gain to add one more unit budget to the link. Repeat **Step3** and **Step4** until the candidate path satisfies the end-to-end delay constraints. If all links on candidate path reaches maximum budget limit and the candidate path is still infeasible, put the O-D pair into **Waiting Queue**. If any O-D pair is sent to **Waiting Queue**, increase *Searching_Counter*.

Step5. If the **Waiting Queue** is empty, stop the algorithm (the feasible solution is found for all O-D pairs).

Searching Limit Checking Process

Step1. If *Searching_Counter* > *Max_Searching_Limit*, go to next step otherwise stop this process.

Step2. If all links in the network reach the maximum budget limits, stop the algorithm (unable to find feasible solutions) otherwise continue next step.

Step3. Set all links in the network to maximum budget. Pop the front of **Candidate Queue** and find a minimum end-to-end delay routing path for the O-D pair, then send it to **Waiting Queue** until **Candidate Queue** is empty. In the end, double the *Max_Searching_Limit*.

3.3 Simple Algorithms

For comparing the performance with the heuristic algorithm developed in Lagrangean relaxation method, we propose two simple algorithms to solve (IP2). The algorithms are described as follows:

Simple Algorithm 1

Step1. Find a minimum end-to-end delay routing path for each O-D pair.

Step2. Allocate budget to satisfied capacity constraints and end-to-end QoS constraints.

Step3. If any infeasible candidate paths exist, go to next step otherwise stop the algorithm (find the feasible solution for all O-D pair).

Step4. For each O-D pair with infeasible candidate path, repeat **Step1** again. If all links in the network reach the maximum budget limit and any infeasible candidate paths exist, stop the algorithm (unable to find feasible solutions).

Simple Algorithm 2

Step1. Use aggregate flow on links as arc weights and run shortest path algorithm to find a routing path for each O-D pair.

Step2. Allocate budget to satisfied capacity constraints and end-to-end QoS constraints.

Step3. If any infeasible candidate paths exist, go to next step otherwise stop the algorithm (find the feasible solution for all O-D pair).

Step4. For each O-D pair with infeasible candidate path, repeat **Step1** again. If all links in the network reach the maximum budget limit and any infeasible candidate paths exist, stop the algorithm (unable to find feasible solutions).

The concepts of simple algorithm 1 and simple algorithm 2 are very similar, and the only difference is in Step1.

3.4 The Solution Approach for the AFRB Problem

The outcome of RB problem indicates the best defense strategy under a given attack pattern. As mention earlier, the objective of AFRB problem is to maximize the total defense budget by adjusting the decision variable γ_w, where $w \in W$. From the perspective of an attacker, he can control the volume of attack flow, destination node of attack flow, and which entry node to be passed.

In this kind of scenario, we propose a heuristic algorithm to simulate the behavior of an attacker, whose objective is to exhaust the resources of the network. The main idea of the algorithm is based on attack flow adjustment procedure upon the routing paths and budget allocation decided by network administrator. The relation of RB and AFRB problems is showed in Figure 3 – 3.

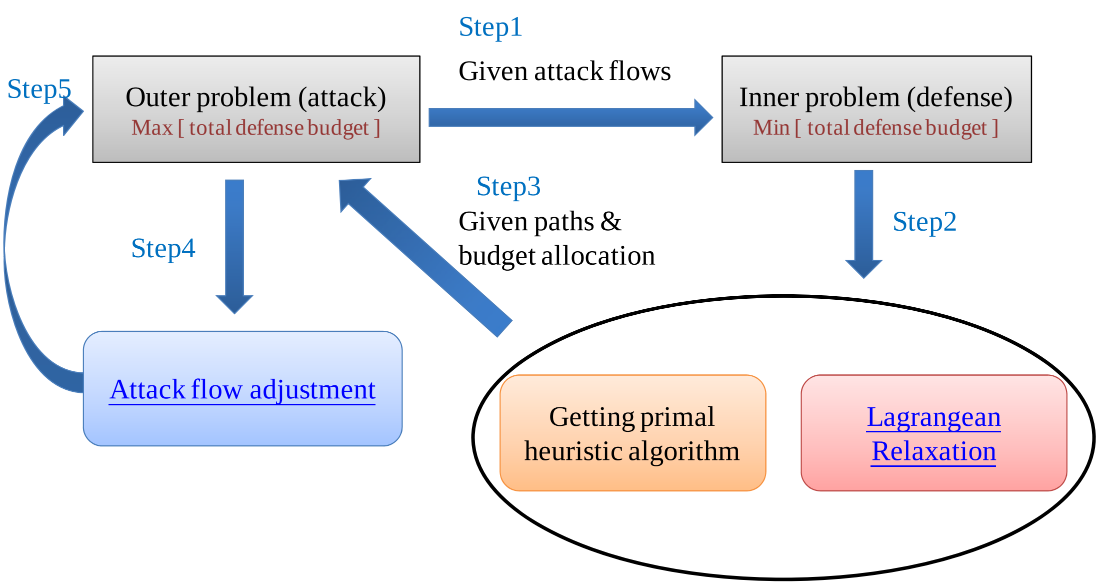


Figure 3 - 3 Solution Approach for the AFRB Problem

The mathematical model of AFRB problem is formulated in (IP1) and the heuristic algorithm is showed in Table 3 – 2.

Table 3 - 2 The Heuristic Algorithm for AFRB Problem

Objective: maximize the minimized total defense budget (max min Z_{IP1})

Initialization: LB (lower bound) = 0

//LR() is the optimal objective function value of (IP2)

WHILE *improvement_counter* <= *improvement_counter_limit* and *iteration* <= *iteration_counter_limit* {

Attack_Flow_Adjustment_Procedure;

Z^*_{IP1} = LR();

IF (Z^*_{IP1} > LB) {

LB = Z^*_{IP1};

improvement_counter = 0;

}

ELSE{

improvement_counter++;

}

iteration++;

}

The attack flow adjustment procedure is described as below:

Table 3 - 3 Attack Flow Adjustment Procedure

Initialization: 1. initial attack flow allocation, get the information of routing paths and budget allocation from RB problem.

2. total attack flow is given

Step1. Use Lagrangean multiplier μ_1 as arc weights to evaluate the importance of each routing path.

Step2. Try to extract one unit attack flow from routing path with lower weight to the path with higher weight.

Step3. Calculate the new total defense budget.

Step4. Find the maximum gain of each attack flow unit, where the gain is defined as

$$gain = new\ total\ defense\ budget - current\ total\ defense\ budget$$

Step5. Repeat the steps above until the total defense budget is maximized.

Chapter 4 Computational Experiments

4.1 Computational Experiments of RB Problem

4.1.1 Experimental Environments

The proposed algorithms for the RB problem are coded in Visual C++ and run on PCs with an INTEL Pentium 4 (2.40GHz) CPU. The *Iteration_Counter_Limit* and *Improvement_Counter_Limit* are set to 800 and 20 respectively. The step size scalar, λ, is initialized to 2 and is halved if the objective function value, Z_D, is not improved in *Improvement_Counter_Limit* iterations. All Lagrangean Multipliers are initialized to be 0 and initial UB is set to 10^{10} to represent infinity value Table 4 - 1, Table 4 - 2.

Three kinds of network topologies are tested, random network, grid network, and mesh network. Each network consists of 9 nodes and 4 dummy nodes Figure 4-1, Figure 4-2, Figure 4-3. Each link has ten kinds of budget configurations and each node has twenty kinds of budget configurations. The capacity of link and node is a function of budget, and the convex form is considered. The total attack flow is tested from 0 to 350 packets per second and the maximum allowable end-to-end delay are set to 600 ms and 900 ms for in and cross AS QoS requirements. Basic normal traffic requirements of in and cross AS are set to 2 and 4 packets per second respectively. The Hurst parameters of internal flow and external flow are set to 0.7 and 0.75 which can express the characteristic of network self-similarity but are not too bursty to affect the whole network Table 4 - 3.

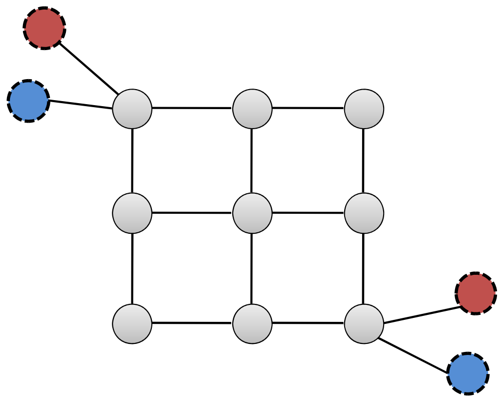


Figure 4 - 1 Grid Network

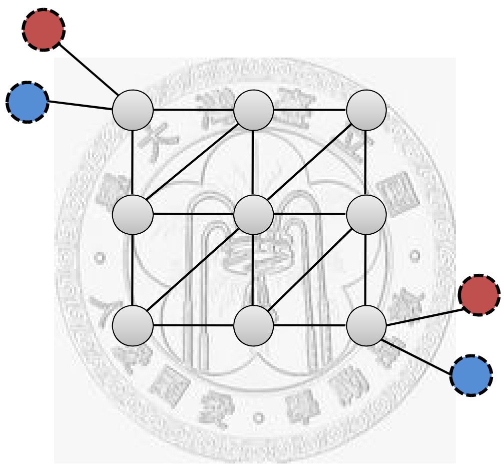


Figure 4 - 2 Mesh Network

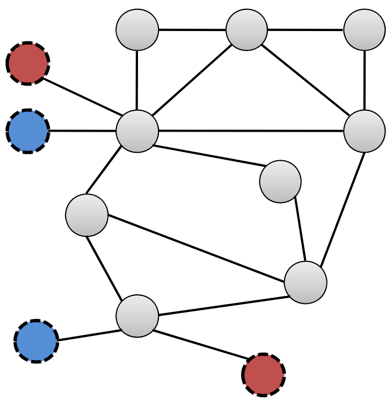


Figure 4 - 3 Random Network

Table 4 - 1 Test Platform

Test Platform	
CPU	INTEL Pentium 4 (2.4 GHz)
RAM	1 GB
Operation System	Microsoft Windows XP
Development Platform	Microsoft Visual Studio 2005
Programming Language	C++

Table 4 - 2 Parameters of LR

Parameters	**Values**
Iteration Counter Limit	800
Improvement Counter Limit	20
Initial UB	10^{10}
Initial Lagrangean Multipliers	$\mu^1, \mu^2, \mu^3, \mu^4 = 0$
Initial Scalar of Step Size	2

Table 4 - 3 Parameters of RB Problem

Parameters	**Value**
Testing Topology	Random networks, Grid networks, Mesh networks
Network Size	9 nodes and 4 dummy nodes, 16 nodes and 6 dummy nodes, 20 nodes and 6 dummy nodes
Budget Configurations	Link: $B_l = \{1, 2 \ldots, 50\}$

	Node (virtual link): $B_l = \{1, 2 \ldots, 100\}$
Link Capacity	Link: $c_l = 1 + 50 \times LN(1 + b_l \times 10)$, Node (virtual link): $c_l = 1 + 70 \times LN(1 + b_l \times 20)$ (packets/sec)
Total Attack Flow	0 ~ 350 (packets/sec)
Maximum Allowable End-to-End Delay	In the AS: 600 (ms) Cross the AS: 900 (ms)
Hurst Parameter	Inner Normal Traffic: 0.7 External Normal Traffic: 0.75 Attack Flow: 0.85

4.1.2 Computational Experiments

Many literatures pointed out the effects of Hurst parameter, thus we first test the different Hurst parameter values of attack flow Table 4 - 4.

Table 4 - 4 Different Hurst Parameter Values of Attack Flow

Network Topology	Total Traffic (packets/sec)	H = 0.75	H = 0.8	H = 0.85
		Total Defense Budget (units)		
Random Networks	0	39	39	39
	50	39	39	40
	100	39	39	41
	150	41	42	42
	200	43	44	46
	250	47	49	52
	300	53	56	61
	350	66	67	78
Grid Networks	0	33	33	33
	50	33	33	33
	100	33	33	35
	150	35	35	37
	200	36	38	40
	250	39	41	47
	300	44	49	57
	350	55	59	74
Mesh Networks	0	41	41	41
	50	41	41	41
	100	41	41	42
	150	43	43	43
	200	43	45	45
	250	47	47	50
	300	50	51	55
	350	56	61	68

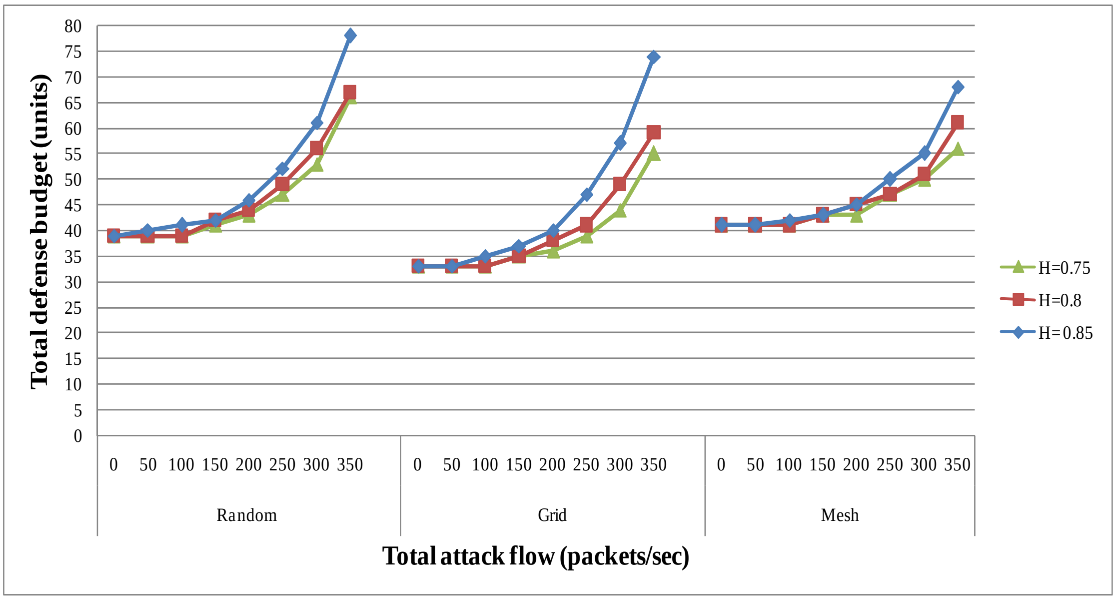


Figure 4 - 4 Different Hurst Parameter Values of Attack Flow

In the following experiments of RB problem, we fix the Hurst parameter value of attack flow to be 0.85, which shows high degree of self-similarity, to compare the solution quality of LR Table 4 - 5. The improvement ratios of LR are also listed on Table 4 - 6.

Table 4 - 5 Solution Quality for RB Problem

Network Topology	Total Traffic (packets/sec)	Total Defense Budget (units)			
		SA1	SA2	LR	LB
Random Networks	0	39	39	39	39
	50	40	40	40	39
	100	41	41	41	39
	125	42	42	41	39.1287
	150	43	43	42	39.4812
	175	46	45	44	40.3809
	200	47	48	46	41.9295
	250	55	55	52	45.78
	300	66	66	61	49.0242
	350	79	84	75	64.5828
Grid Networks	0	33	33	33	33
	50	33	33	33	33
	100	35	35	35	33
	125	36	35	35	33
	150	38	37	37	34.125
	175	40	40	38	37.0276
	200	45	43	40	34.9825
	250	54	51	47	37.4435
	300	70	67	57	41.1632
	350	92	85	74	49.9805
Mesh Networks	0	41	41	41	41
	50	41	41	41	41
	100	43	43	42	41
	125	43	43	43	41
	150	44	44	43	41.7592
	175	45	45	44	43.8229
	200	46	47	45	43.6886
	250	52	51	50	47.9996
	300	60	58	55	48.9665
	350	72	73	68	65.2899

Table 4 - 6 Improvement Ratios for RB Problem

Network Topology	Total Traffic (packets/sec)	Improvement Ratio to SA1 (%)	Improvement Ratio to SA2 (%)	Gap to LB (%)
Random Networks	0	0.00	0.00	0.00
	50	0.00	0.00	2.56
	100	0.00	0.00	5.13
	125	2.38	2.38	4.78
	150	2.33	2.33	6.38
	175	4.35	2.22	8.96
	200	2.13	4.17	9.71
	250	5.45	5.45	13.59
	300	7.58	7.58	24.43
	350	5.06	10.71	16.13
Grid Networks	0	0.00	0.00	0.00
	50	0.00	0.00	0.00
	100	0.00	0.00	6.06
	125	2.78	0.00	6.06
	150	2.63	0.00	8.42
	175	5.00	5.00	2.63
	200	11.11	6.98	14.34
	250	12.96	7.84	25.52
	300	18.57	14.93	38.47
	350	19.57	12.94	48.06
Mesh Networks	0	0.00	0.00	0.00
	50	0.00	0.00	0.00
	100	2.33	2.33	2.44
	125	0.00	0.00	4.88
	150	2.27	2.27	2.97
	175	2.22	2.22	0.40
	200	2.17	4.26	3.00
	250	3.85	1.96	4.17
	300	8.33	5.17	12.32
	350	5.56	6.85	4.15

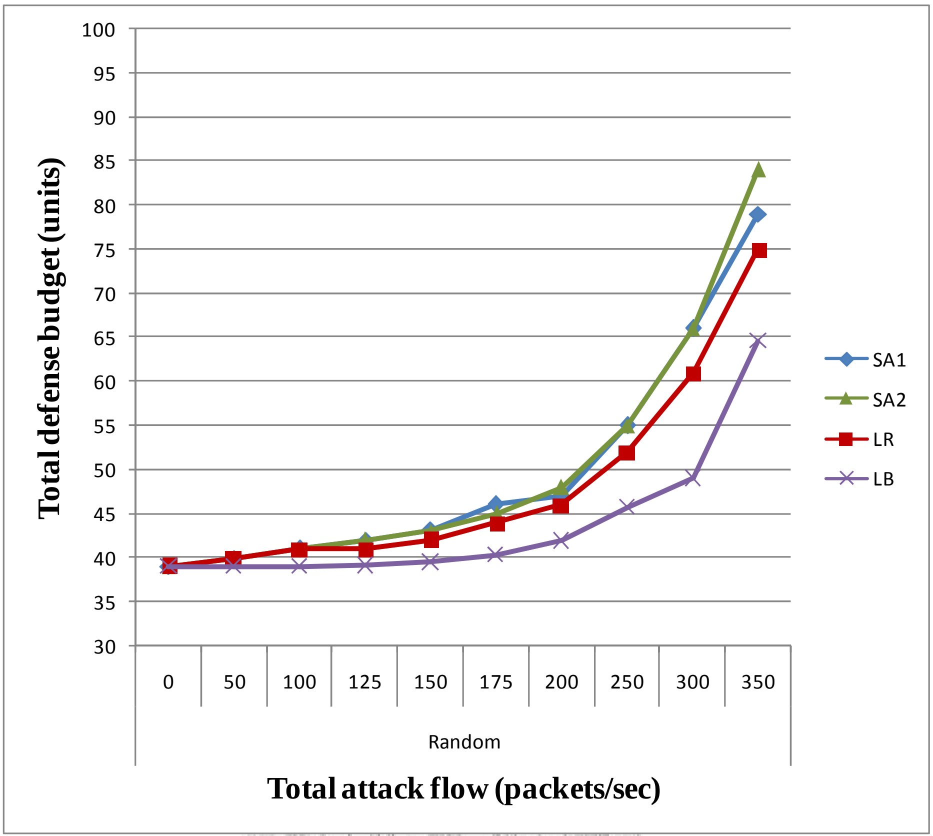


Figure 4 - 5 Total Defense Budget under Different Total Attack Flows in the Random Network

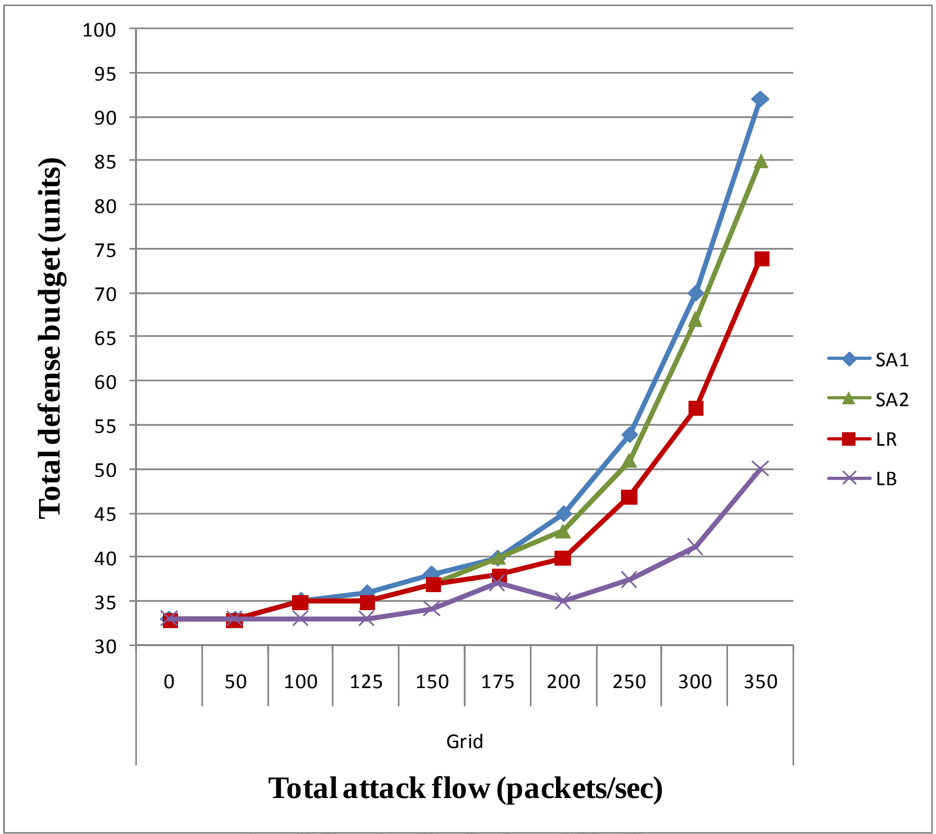


Figure 4 - 6 Total Defense Budget under Different Total Attack Flows in the Grid Network

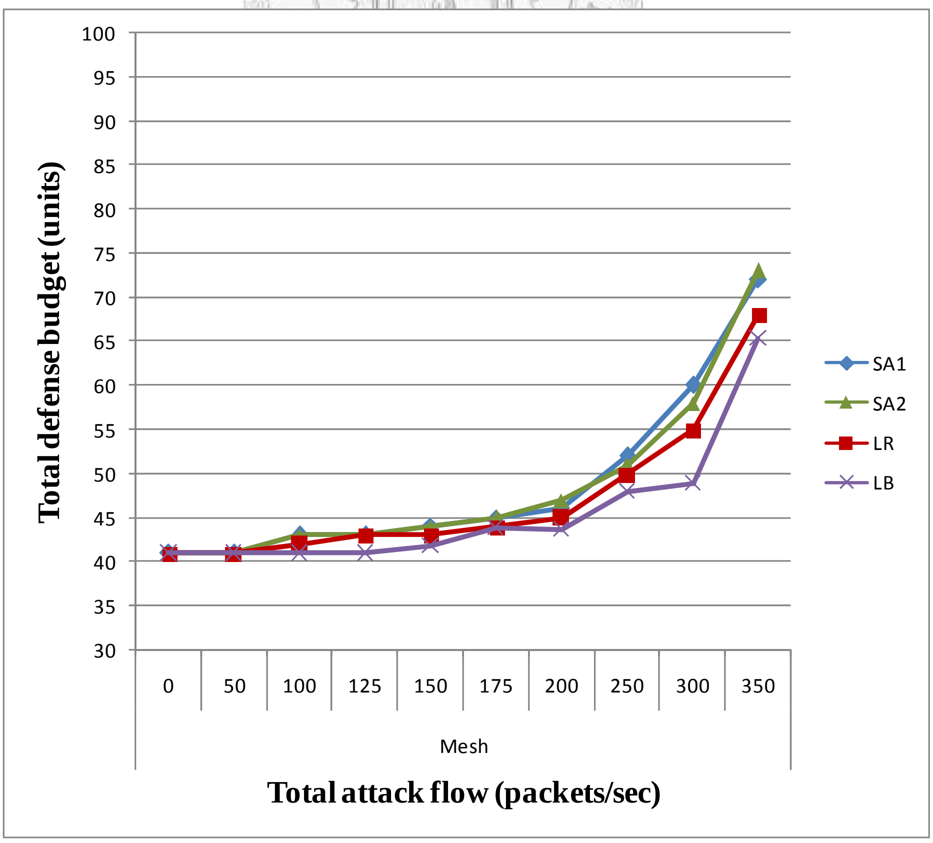


Figure 4 - 7 Total Defense Budget under Different Total Attack Flows in the Mesh Network

Table 4 - 7 Solution Quality of RB Problem in Different Network Scale

Network Topology	Network Scale (Number of O-D Pairs)	Total Defense Budget (units)			
		SA1	SA2	LR	LB
Random Networks	128	59	61	52	50
	246	94	88	84	77.3215
	390	205	190	164	150.982
	566	698	373	233	197.373
Grid Networks	128	49	48	44	39
	246	89	86	77	70.3934
	390	193	180	169	157.375
	566	1848	425	357	291.889
Mesh Networks	128	52	54	44	39
	246	87	86	82	74.8068
	390	162	156	117	99.3095
	566	2070	350	196	165.224

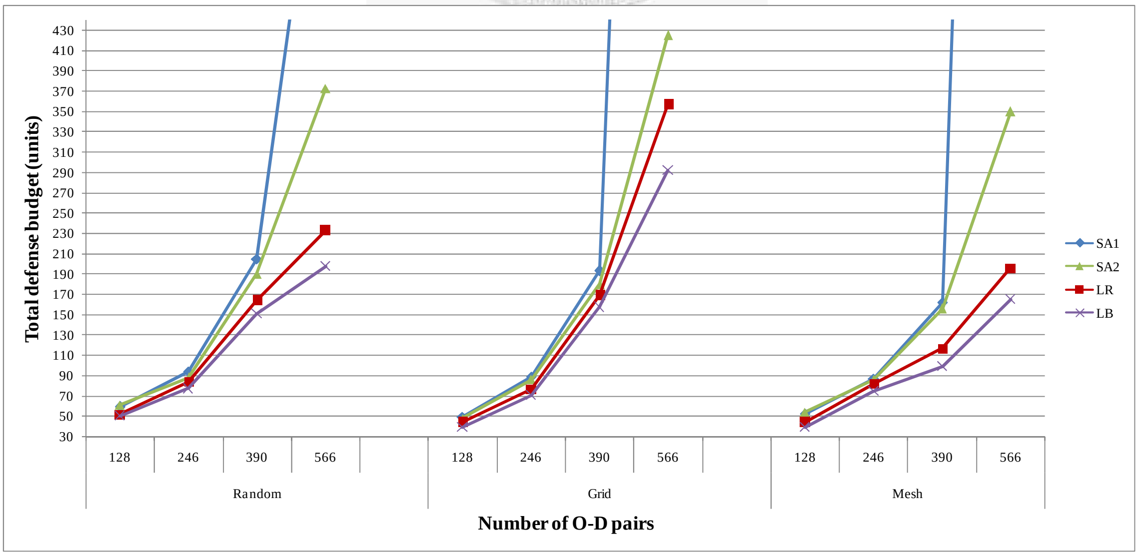


Figure 4 - 8 Total Defense Budget under Different Network Scale

4.1.3 Discussion of Results

Figure 4 - 4 shows the effects of different Hurst parameter values assigned to attack flows, and the effects on total defense budget become obvious while the total attack flow increases. The total defense budget is increasing rapidly when the Hurst parameter value of attack flow is set to 0.85. In order to display the burstiness of DDoS attack flows, which usually behave like ON/OFF traffic sources, we set H = 0.85 to attack flows, H = 0.75 to external normal traffic, and H = 0.7 to internal normal traffic in the following experiments.

From Figure 4 - 5 to 4 - 6, we can observe the LR costs less total defense budget than SA1 and SA2 in the same total attack flow, and the improvement ratios of LR to SA1 and SA2 are increasing when we enlarge the total attack flow. In the mean time, we observe that the LR performs better in the random network and grid network. The reason of this result might be that an O-D pair can have more candidate paths to send data in the mesh network so that SA1 and SA2 can find good routing paths for each O-D pair. Besides, the LR method provides us a LB to exam the solution qualities, the error gap between LR and LB are shown on column 5 of Table 4 - 5.

4.2 Computational Experiments of AFRB Problem

The experimental environments are basically the same as RB problem, and additional parameters for the AFRB problem are used in attack flow adjustment procedure. The maximum iteration limit and improvement iteration limit of AFRB problem are 50 and 5 respectively. For each iteration of AFRB problem, we need to run *Iteration_Counter_Limit* LR iterations to optimally solve RB problem first, and then run the attack flow adjustment procedure.

4.2.1 Computational Experiments

For comparing the solution quality of AFRB problem, the initial attack flow allocation is compared. The initial attack flow allocation is based on the link degree of network nodes Table 4 - 8.

Table 4 - 8 Experimental Results of AFRB Problem

Network Topology	Total Traffic (packets/sec)	Total Defense Budget		Total Defense Budget Increasing Ratios (%)
		Initial Attack Flow Allocation	Attack Flow Adjustment	
Random Networks	0	39	39	0.00
	50	40	40	0.00
	100	41	46	12.20
	125	41	52	26.83
	150	42	63	50.00
	175	44	73	65.91
	200	46	110	139.13

Grid Networks	0	33	33	0.00
	50	33	35	6.06
	100	35	42	20.00
	125	35	51	45.71
	150	37	70	89.19
	175	38	97	155.26
	200	40	125	212.50
Mesh Networks	0	41	41	0.00
	50	41	41	0.00
	100	42	47	11.90
	125	43	51	18.60
	150	43	59	37.21
	175	44	67	52.27
	200	45	87	93.33

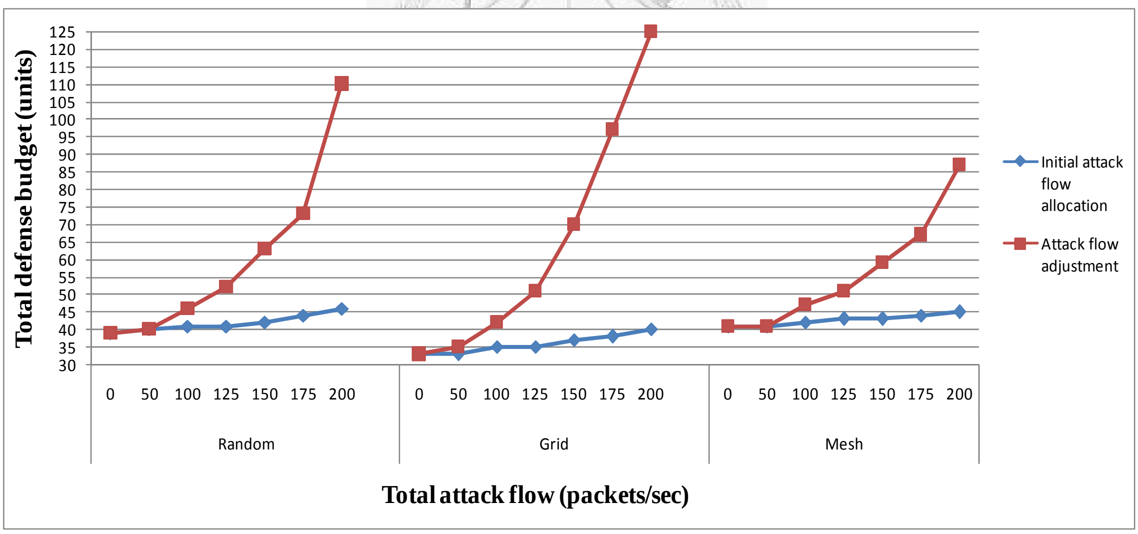


Figure 4 - 9 Total Defense Budget after Attack Flow Adjustment Procedure

For comparing different network topologies purpose, we have to notice the basic total defense budget. The meaning of the basic network total defense budget is that each link has a basic budget configuration, which is one unit initially, thus different network topologies have different basic budget.

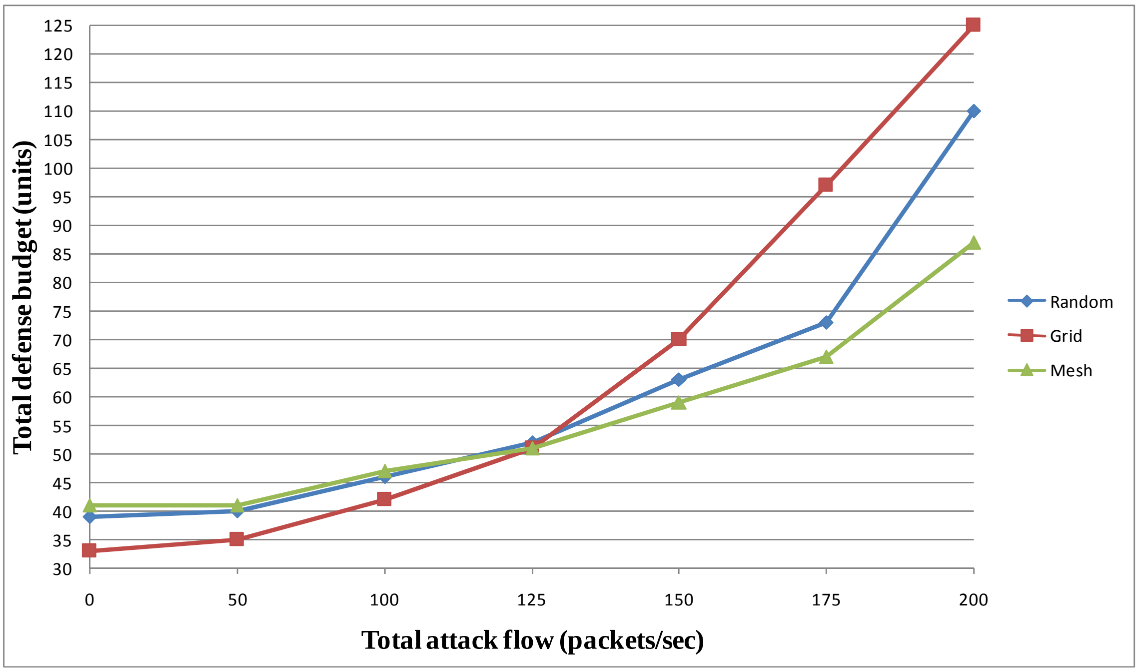


Figure 4 - 10 Total Defense Budget of Different Network Topologies under Attacks

4.2.2 Discussion of Results

In Figure 4 - 9, we can observe that after the attack flow adjustment, total defense budget increases and rises dramatically when the total attack flow excesses a threshold in a given maximum budget limit. The effects on total defense budget differs in different network topologies; in the 9 nodes and 4 dummy nodes grid network, it can sustain less attack flow volume than random network and mesh network under the same QoS requirements and maximum total budget limit on each link.

To research the reasons why the grid network can sustain less attack flow volume than the random network and the mesh network, we inference that because an O-D pair has less candidate paths for network administrator to choose in the grid network, which has less links than the random network and the mesh network that can be observed from Figure 4 - 1 to Figure 4 - 3.

Chapter 5 Conclusion and Future Work

5.1 Conclusion

Even so many network security commercial products are developed nowadays, it is still hard to defense DDoS attacks perfectly, and we research on defense against the attacks in the victim end network. Another observation is that the DDoS attacks with higher network self-similarity than normal network traffic do consume more resources of the network, and the QoS requirements are hard to be satisfied as well. In this thesis, the defense mechanism proposed for the network administrator performs better than simple heuristic algorithms in grid, random, and mesh networks. In contrary an intelligent attacker who has more attack power will finally exhaust the resources of the network.

The first contribution of the thesis is that we propose a mathematical model to analyze this kind of DDoS attacks and defense scenario. The scenario can be analyzed by the AFRB problem and the RB problem, besides we also propose a good solution approach to the RB problem and the AFRB problem as well. For network administrator, we provide a defense mechanism to defense the DDoS attacks executed by the attacker whose objective is to exhaust the entire resources of the network. Also, the performance of the defense mechanism in different network topologies is considered and analyzed. Furthermore, the network self-similarity is considered in our mathematical model, and first we capture the aggregate characteristic of self-similar traffics, and then we setup the DDoS attack flows with higher Hurst parameter value because the On/Off characteristic of DDoS attack traffic is recognized easily.

5.2 Future Work

We highlight three issues to be our future work. First, we want to expand the mathematical model to several AS to achieve scalability and the concept of collaborative defense. Next, if we want to model the concept of collaborative defense, the features of DDoS attacks detection and filtering must be considered. Hence, we want to add these features into our mathematical model and research on the effects of attacks detection and filtering probabilities to the network.

Another issue is that in this thesis, we take the end-to-end delay to be the QoS requirements but do not include the end-to-end delay jitter. This is due to that we have not found a suitable form of the delay jitter, which is a function of utilization and Hurst parameter. We hope to find or developed a suitable form of delay jitter in the future. Last issue is that we expect to test more network topologies to verify our solution approaches and enlarge the network size as possible as we can.

References

[1] J. Mirkovic, P. Reiher, “A Taxonomy of DDoS Attack and DDoS Defense Mechanisms”, *ACM SIGCOMM Computer Communications Review*, Vol. 34, No. 2, April 2004.

[2] H. Wang, D. Zhang, K. G. Shin, “Change-Point Monitoring for the Detection of DoS Attacks”, *IEEE Transactions on Dependable and Secure Computing*, Vol. 1, No. 4, Octorber-December 2004.

[3] D. K. Y. Yau, J. C. S. Lui, F. Liang, Y. Yam, “Defending Against Distributed Denial-of-Service Attacks with Max-Min Fair Server-Centric Router Throttles”, *IEEE/ACM Transactions on Networking,* Vol. 13, No. 1, February 2005.

[4] Y. Xiang, Y. Lin, W. L. Lei, S. J. Huang, “Detecting DDoS attack based on network self-similarity”, *IEE Proc.-Commun*, Vol. 151, No. 3, June 2004.

[5] J. Mirkovic, P. Reiher, “D-WARD: A Source-End Defense against Flooding Denial-of-Service Attacks”, *IEEE Transactions on Dependable and Secure Computing*, Vol. 2, No. 3, July-September 2005.

[6] D. Dittrich, “Distributed Denial of Service (DDoS) Attacks and Tools Page”, http://staff.washington.edu/dittrich/misc/ddos/

[7] S. Hansman, R. Hunt, “A Taxonomy of Network and Computer Attacks”, *Computers & Security*, Vol. 24, pp. 31-43, 2005.

[8] T. Bu, S. Norden, T. Woo, "A survivable DoS-Resistant Overlay Network", *Computer Networks*, Vol. 50, pp. 1281-1301, 2006.

[9] K. T. Cheng, F. Y. S. Lin, "Near-Optimal Delay Constrained Routing in Virtual Circuit Networks", *IEEE INFOCOM*, 2001.

[10] W. E. Leland, M. S. Taqqu, W. Willinger, D. V. Wilson, "On the Self-Similar Nature of Ethernet Traffic (Extended Version)", *IEEE/ACM Transactions on Networking*, Vol. 2, No. 1, February 1994.

[11] Q. Yu, Y. Mao, T. Wang, F. Wu, "Hurst Parameter Estimation and Characteristics Analysis of Aggregate Wireless LAN Traffic", *In Proceeding IEEE International Communications, Circuits and Systems Conference*, 2005.

[12] J. Domanska, "The influence of traffic self-similarity on QoS mechanisms", *IEEE SAINT-W'05*, 2005.

[13] Y. G. Kim, A. Shiravi, P. S. Min, "Prediction-Based Routing through Least Cost Delay Constraint", *Proceedings of the 18th International Parallel and Distributed Processing Symposium (IPDPS'04)*.

[14] G. Mazzini, R. Rovatti, G. Setti, "Self-Similarity in Max/Average Aggregated Processes", *Proceedings of the International Symposium on Circuits and Systems*, 2004.

[15] Z. Zeitlin, "Integer Allocation Problems of Min-Max Type with Quasiconvex

Separable Functions”, *Operations Research*, Vol. 29, No. 1, January-February 1981.

[16] D. M. Nicol, W. H. Sanders, K. S. Trivedi, “Model-Based Evaluation: From Dependability to Security”, *IEEE Transactions on Dependable and Secure Computing*, Vol. 1, No. 1, January-March 2004.

[17] J. C. Knight, E. A. Strunk, K. J. Sullivan, “Towards a Rigorous Definition of Information System Survivability”, *Proceeding of the DARPA Information Survivability Conference and Exposition (DISCEX 2003)*, Vol. 1, pp. 78-89, April 2003.

[18] M. L. Fisher, “The Lagrangean Relaxation Method for Solving Integer Programming Problems”, *Management Science*, Vol. 27, No. 1, pp. 1-18, January 1981.

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

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

[21] L. A. Gordon, M. P. Loeb, W. Lucyshyn, and R. Richardson, “2006 CSI/FBI Computer Crime and Security Survey”, *Computer Security Institute*, 2006, http://GoCSI.com/

[22] Matpack C++ Numerics and Graphics Library, http://www.matpack.de/

[23] J. McDermott, "Attack-Potential-Based Survivability Modeling for High-Consequence Systems", *IEEE International Workshop on Information Assurance*, 2005.

簡歷

姓名：　郭承賓

出生地：台灣省桃園縣

出生日：中華民國七十一年十一月二十三日

學歷：

九十年九月至九十四年六月

國立政治大學 資訊管理學系 學士

九十四年九月至九十六年十二月

國立台灣大學 資訊管理學研究所 碩士