

國立臺灣大學資訊管理研究所

碩士論文

Graduate Institute of Information Management

National Taiwan University

Master Thesis

防禦分散式阻絕服務攻擊之近似最佳化過濾及路由策略

Near Optimal Filtering and Routing Policies
against Distributed Denial-of-Service (DDoS) Attacks



江政祐

Henry Cheng-You, Chiang

指導教授：林永松 博士

Advisor: Frank Yeong-Sung Lin, Ph.D.

中華民國九十七年七月

July, 2008

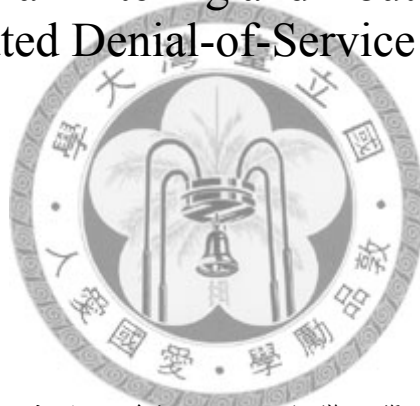


國立臺灣大學資訊管理研究所碩士論文

指導教授：林永松 博士

防禦分散式阻絕服務攻擊之近似最佳化過濾及路由策略

Near Optimal Filtering and Routing Policies
against Distributed Denial-of-Service (DDoS) Attacks



本論文係提交國立台灣大學
資訊管理學研究所作為完成碩士論文
學位所需條件之一部份

研究生：江政祐

中華民國九十七年七月



國立臺灣大學碩士學位論文
口試委員會審定書

防禦分散式阻絕服務攻擊

之近似最佳化過濾及路由策略

Near Optimal Filtering and Routing Policies
against

Distributed Denial-of-Service (DDoS) Attacks

本論文係 江政祐 君（學號 R95725005）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 97 年 7 月 15 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：

<u>趙敬仁</u>	<u>林永松</u>
<u>黃信輝</u>	<u>李俊亨</u>
<u>孫雅潔</u>	<u> </u>

所 長：

陳靜村



謝誌

轉眼間兩年的研究生涯就要過去了，留下的是滿懷的回憶與感激，謝謝一路上陪伴著我成長的家人與朋友，沒有你們就不會有今天的我。

首先感謝我的父母江銘泉與林素蘭，謝謝每當我遇到困難的時後你/妳們都會給我精神上的安慰，並且告訴我你/妳們人生中寶貴的經驗，使得我在迷途時有著一股支持的力量讓我找到光明的方向。每當深感壓力時，你/妳們更是會開導我，讓我像是如釋重負突破層層關卡到達最終的目的地，在此對你/妳們說聲謝謝。

在學業上最要感謝的就是林永松老師，每當有學業上的問題請教老師時，老師都會不顧其煩的幫助我得到問題的答案，特別記得有一次對於一個當時無法解決的問題，在經過與老師多次的討論後，最後終於解決了，此實讓我印象深刻與心存感激。在學業之外老師也給予我做人處事上重大的啟發，如何的待人處事與積極進取，此是何等的受用無窮。此外，也要感謝清大電機趙啟超教授、輔大資工呂俊賢教授以及本校孫雅麗教授與莊裕澤教授在口試時給予我論文上的建議與指導，如此讓我在研究的知識上能夠更加的充實，以及最終使得本篇論文能更嚴謹及完整，在此謝謝這幾位老師。

特別謝謝資安小組的所有成員，首先謝謝柏皓學長給予我論文上的建議以及進度的安排，如此使的我更積極的從事研究工作，另也謝謝學長在簡報與投影片上的指導，使得口試時能夠如此的順利。接著則是霽語、俊維學長、承賓學長、雅芳學姐、翊恆學長、豈毅學長及坤道學長，謝謝你/妳們給予我有關研究生涯時論文撰寫的經驗傳承，如此讓我在徬徨無助時能維持充份的自信。謝謝俊維學長在一開始帶我們對論文的入門，實受益良多。而對於一起度過研究生涯的奕廷、志浩與孜謙，我們終於走過來了，謝謝有你們的參與。有了與你們共同的研討才使得這篇論文能夠順利的完成。

謝謝學弟妹，睿斌、竣韋、培維、猷順、冠瑋、友仁及宴毅，謝謝你們在口試時購買的餐點以及適時的支援，如此讓我的論文口試能如此的順利。

最後感謝讓我有這個機會進入台大，完成了當時的夢想以及在學業上更加的精進，如今回想這一切都是值得的。

江政祐 謹識
于台大資訊管理研究所
民國九十七年七月

論文摘要

論文題目：防禦分散式阻絕服務攻擊之近似最佳化過濾及路由策略

作者：江政祐

九十七年七月

指導教授：林永松 博士

分散式阻絕服務攻擊已成為今日網際網路之嚴重威脅。在分散式阻絕服務攻擊發生時，眾多惡意封包佔據了網路伺服器的資源，導致合法使用者資源存取之困難。即使在使用了過濾器機制來防範分散式阻絕服務攻擊，仍無法保證合法使用者完全不受此攻擊之損害。



在本論文中，我們將分散式阻絕服務攻擊之攻擊與防禦情境模擬成一個兩階段的數學規劃問題。在內層問題中，防禦者試圖以分配其有限防禦資源來最大化受分散式阻絕服務攻擊損害之合法流量。而在外層問題則敘述分散式阻絕服務攻擊者之試圖以分配其有限攻擊資源來最小化合法流量。同時為了求得此問題的最佳解，我們採用以拉格蘭日鬆弛法為基礎的演算法來處理內層問題，而利用以次梯度法為基礎的演算法來處理外層問題。

關鍵詞：分散式阻絕服務攻擊、過濾器、數學規劃、資源配置、最佳化、拉格蘭日鬆弛法

THESIS ABSTRACT

GRADUATE INSTITUTE OF INFORMATION MANAGEMENT

NATIONAL TAIWAN UNIVERSITY

NAME: CHENG-YOU, CHIANG

MONTH/YEAR: JULY 2008

ADVISOR: YEONG-SUNG LIN

Near Optimal Filtering and Routing Policies against Distributed Denial-of-Service (DDoS) Attacks

Distributed Denial-of-Service (DDoS) attacks have become an impending threat toward today's Internet. During DDoS attacks, numerous malicious packets occupy a victim server and lead to the difficulty of the legitimate user's access. Even if the filtering thwarts DDoS attacks, no legitimate users can escape the collateral damage.

In this thesis, we model the DDoS attack-defense scenario as a two-level mathematical programming problem. In the inner problem, a defender tries to allocate the limited defense resources for the maximization of the legitimate traffic. In the outer problem, a DDoS attacker tries to allocate the limited attack resources in order to minimize the legitimate traffic. A Lagrangean relaxation-based algorithm is proposed to solve the inner problem, and a subgradient-based algorithm is proposed to solve the outer problem.

Keywords: Distributed Denial-of-Service, Filter, Mathematical Programming,

Resources Allocation, Optimization and Lagrangean Relaxation

Table of Contents

論文摘要	I
THESIS ABSTRACT	III
Table of Contents	IV
List of Tables	VI
List of Figures.....	VII
Chapter 1 Introduction.....	1
1.1 Background	1
1.2 Motivation.....	4
1.3 Literature Survey	7
1.3.1 DDoS Attacks	7
1.3.2 Survivability and Resource Allocation.....	11
1.3.3 Autonomous Systems	13
1.4 Proposed Approach	15
1.5 Thesis Organization	16
Chapter 2 Problem Formulation of the ARAS and FAS models.....	17
2.1 Problem Description	17
2.2 Problem Formulation of the ARAS Model.....	19
2.3 Problem Formulation of the FAS Model	30
Chapter 3 Solution Approach.....	35
3.1 Solution Approach for the FAS Model	35
3.1.1 Lagrangean Relaxation Method	35
3.1.2 Lagrangean Relaxation	39
3.1.3 The Dual Problem and the Subgradient Method.....	43
3.1.4 Getting Primal Feasible Solutions	44
3.2 Solution Approach for the ARAS Model	46
Chapter 4 Computational Experiments.....	49
4.1 Computational Experiments for the FAS Model	49
4.1.1 Simple Algorithm 1	49
4.1.2 Simple Algorithm 2	50
4.1.3 Experiment Environment.....	51
4.1.4 Experiment Results.....	55
4.1.5 Discussion of Results.....	68
4.2 Computational Experiments for the ARAS Model.....	70
4.2.1 Experiment Environment.....	70
4.2.2 Experiment Results.....	72
4.2.3 Discussion of Results.....	76

Chapter 5 Conclusion and Future Work.....	78
5.1 Conclusion	78
5.2 Future Work	80
References.....	82



List of Tables

Table 1.3.2-1 The Key Components of The Definition Regarding Survivability .	12
Table 2-1 Problem Assumption and Description of the ARAS Model.....	25
Table 2-2 Given Parameters of the ARAS Model.....	26
Table 2-3 Decision Variables of the ARAS Model	28
Table 2-4 Given Parameters of the FAS Model	31
Table 2-5 Decision Variables of the FAS Model.....	32
Table 3.1.4-1 Heuristic Algorithm for Getting Primal Feasible Solution.....	45
Table 3.2-1 Heuristic Algorithm for Solving ARAS Model	47
Table 3.2-2 Adjustment Procedure Algorithm.....	48
Table 4.1.1-1 Pseudo Code of Simple Algorithm 1	50
Table 4.1.1-2 Pseudo Code of Simple Algorithm 2	51
Table 4.1.3-1 Experiment Parameter Settings for LR in the FAS model	54
Table 4.1.3-2 Experiment Parameter Settings for the FAS model.....	54
Table 4.1.4-1 Experiment Results of Grid Network for the FAS Model ($ N =25$).	55
Table 4.1.4-2 Experiment Results of Mesh Network for the FAS Model ($ N =25$)	55
Table 4.1.4-3 Experiment Results of Random Network for the FAS Model ($ N =25$)	56
Table 4.1.4-4 Experiment Results of Grid Network for the FAS Model ($ N =49$).	56
Table 4.1.4-5 Experiment Results of Mesh Network for the FAS Model ($ N =49$)	56
Table 4.1.4-6 Experiment Results of Random Network for the FAS Model ($ N =49$)	56
Table 4.1.4-7 Experiment Results of Grid Network for the FAS Model ($ N =100$)	57
Table 4.1.4-8 Experiment Results of Mesh Network for the FAS Model ($ N =100$)	57
Table 4.1.4-9 Experiment Results of Random Network for the FAS Model ($ N =100$)	57
Table 4.2.1-1 Experiment Parameter Settings for the Adjustment Procedure in the ARAS model	71
Table 4.2.1-2 Experiment Parameter Settings for the ARAS model.....	71
Table 4.2.2-1 Experiment Results of Extra-Small Networks ($ N =25$)	72
Table 4.2.2-2 Experiment Results of Small Networks ($ N =49$).....	73

List of Figures

Figure 1.2-1 Types of Attacks or Misuse Detected in the last 12 Months.....	5
Figure 1.2-2 Dollar Amount Losses by Type of Attack	6
Figure 1.3.3-1 Different kinds of AS	14
Figure 2.2-1 The initial network topology	21
Figure 2.2-2 The network topology with only good user traffic	21
Figure 2.2-3	22
The maximum attack traffic obtained by the attack budget allocation.....	22
Figure 2.2-4 The (real) attack traffic sent by zombies to attack victim servers ...	22
Figure 2.2-5 The network topology with the aggregate traffic	23
Figure 2.2-6 The network topology with filters allocated.....	23
Figure 2.2-7 The network topology with the routing policy adopted.....	24
Figure 2.2-8 The rerouted (real) attack traffic	24
Figure 3-1 Concept of the Lagrangean Relaxation Method.....	37
Figure 3-3 Solution Approach to the ARAS model	47
Figure 4.1.3. Network Topologies:.....	53
(a) Grid Network (b) Mesh Network (c) Random Network.....	53
Figure 4.1.4-1	58
the Remaining Good Traffic of Grid Network with FL10 ($ N =25$)	58
Figure 4.1.4-2	58
the Remaining Good Traffic of Mesh Network with FL10 ($ N =25$)	58
Figure 4.1.4-3	59
the Remaining Good Traffic of Random Network with FL10 ($ N =25$)	59
Figure 4.1.4-4	59
the Remaining Good Traffic of Grid Network with FL10 ($ N =49$)	59
Figure 4.1.4-5	60
the Remaining Good Traffic of Mesh Network with FL10 ($ N =49$)	60
Figure 4.1.4-6	60
the Remaining Good Traffic of Random Network with FL10 ($ N =49$)	60
Figure 4.1.4-7	61
the Remaining Good Traffic of Grid Network with FL10 ($ N =100$)	61
Figure 4.1.4-8	61
the Remaining Good Traffic of Mesh Network with FL10 ($ N =100$)	61
Figure 4.1.4-9	62
the Remaining Good Traffic of Random Network with FL10 ($ N =100$)	62
Figure 4.1.4-10	62
the Remaining Good Traffic of Grid Network with different FL and RP	

(N =100)	62
Figure 4.1.4-11	63
the Remaining Good Traffic of Mesh Network with different FL and RP	
(N =100)	63
Figure 4.1.4-12	63
the Remaining Good Traffic of Random Network with different FL and RP	
(N =100)	63
Figure 4.1.4-13	64
the Remaining Good Traffic of Grid Network with different FL (N =49).....	64
Figure 4.1.4-14	64
the Remaining Good Traffic of Mesh Network with different FL (N =49).....	64
Figure 4.1.4-15	65
the Remaining Good Traffic of Random Network with different FL (N =49)	65
Figure 4.1.4-16	65
the Remaining Good Traffic of Grid Network with different FL (N =100).....	65
Figure 4.1.4-17	66
the Remaining Good Traffic of Mesh Network with different FL (N =100).....	66
Figure 4.1.4-18	66
the Remaining Good Traffic of Random Network with different FL (N =100) ..	66
Figure 4.1.4-19	67
the Remaining Good Traffic of Grid Network with different Attacker Budget	
(N =100)	67
Figure 4.1.4-20	67
the Remaining Good Traffic of Mesh Network with different Attacker Budget	
(N =100)	67
Figure 4.1.4-21	68
the Remaining Good Traffic of Random Network with different Attacker Budget	
(N =100)	68
Figure 4.2.2-1 the Remaining Good Traffic of Extra-small Networks at V1	73
Figure 4.2.2-2 the Remaining Good Traffic of Small Networks at V1	74
Figure 4.2.2-3 the Remaining Good Traffic of Extra-small Networks at V2	74
Figure 4.2.2-4 the Remaining Good Traffic of Small Networks at V2	75
Figure 4.2.2-5 the Remaining Good Traffic of Extra-small Networks at V3	75
Figure 4.2.2-6 the Remaining Good Traffic of Small Networks at V3	76

Chapter 1 Introduction

1.1 Background

The DDoS attack is regarded as a tough-to-solve problem in the existence of various network attacks today. Many scholars have proposed effective approaches to defend against this attack. However, the result is still far from the expectation. Legitimate users do usually suffer. Several forms of DDoS attacks exist but one we want to address is that the attacker sends hundreds of thousands unwanted packets to overwhelm a victim server [1]. Perhaps it is for the purpose of the illegal benefit or the military triumph. During the attack, legitimate users have difficulty in accessing the victim server due to the denied server resource. The malicious traffic occupies all available resources. Thus, the operation of the victim server is seriously diminished.

One who intuitively deals with the above-mentioned DDoS attack may come up with a trivial solution which is, simply, to discard unwanted packets and accept legitimate ones. However, the DDoS attack is not an easy problem. As the deficiency of a well-designed network protocol, IP address can be easily modified due to the IP

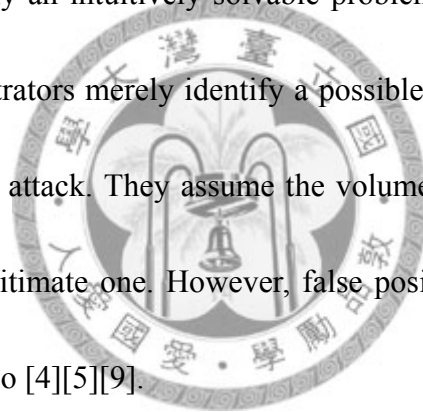
Spoofing [17]. An attacker can change its IP address by a packet field modification, which makes the detection of the attack origin difficult. A forgery legitimate packet may originate from an attack source. As a result, it is not as simple as one think to solve the problem so easily.

Nevertheless, thanks to the advent of the filtering mechanism, the threat from the DDoS attack may get relieved. In more details, the filtering mechanism is, as the attack starts, the victim server obtains the help from the upstream router which, in advance, regulate the incoming traffic. (Technically speaking, the incoming traffic here is spoken as the aggregate traffic, both the attack and legitimate traffic). With the filtering mechanism, the victim server can first signal upstream routers to install filters as the aggregate traffic exceeds a server traffic threshold. Following it, routers are capable of regulating the aggregate traffic. The victim server thus escapes the destiny of the inundation [2][3].

However, although the filtering mechanism is able to mitigate the DDoS attack, it will lead to another problem which is the impairment of the legitimate traffic (technically called the collateral damage). In the filtering, the legitimate traffic is, possibly, discarded. The legitimate traffic can not be precisely identified as the router

is to regulate the aggregate traffic. Eventually, filtering mechanism can not guarantee the maintenance of all legitimate traffic.

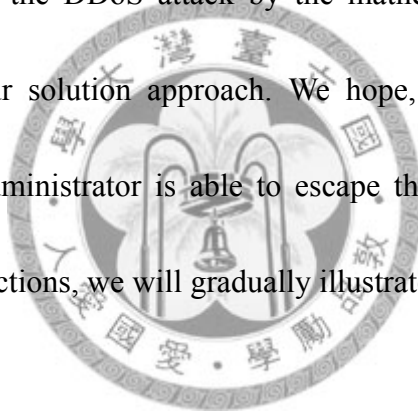
The reason is heavily associated with the design of the network protocol. The network protocol allows its users to so easily modify the content of a packet that an attacker is able to change its identity without any special effort [17]. No perfect rules now could be adopted to precisely identify the packet's legitimacy due to the awkward design. It is why an intuitively solvable problem becomes so complex. At present, network administrators merely identify a possible attack packet by its traffic volume during the DDoS attack. They assume the volume of the attack traffic is far more than that of the legitimate one. However, false positive or false negative may still happen in this scenario [4][5][9].



Due to the collateral damage, a naive approach comes out trying to avoid the damage. If we allocate filters to the routers nearest to the attacker, no legitimate users will thus get impaired. More specifically, it is the concept to install filters at all routers one hop away from the attacker. Ideally, it is perfect if the budget permits. However, due to the budget constraint, it is not practical. To install a filter certainly incurs the cost for the filter allocation. Besides, as filter itself, it also deserves some charges [7].

Therefore, under a reasonable circumstance, the budget will not be allowable to cover all expenses from the above concept.

In this moment, a strategic budget allocation becomes very critical. As the filtering is heavily associated with the budget, each filter allocation requires a careful estimation. In order to obtain the best result, we follow the guideline of the optimal resource allocation from [7][10][14] which helps us reach the final victory. In this thesis, we will formulate the DDoS attack by the mathematical programming and solve the problem by our solution approach. We hope, by adopting our solution approach, the network administrator is able to escape the collateral damage in the future. In the following sections, we will gradually illustrate our relevant works.



1.2 Motivation

According to the CSI/FBI Computer Crime and Security Survey (2007) [21], the percentage of the DDoS attack detected in that period does not change a lot, as shown in Figure 1.2-1, but the losses caused by the DDoS attack are still ranked high, as shown in Figure 1.2-2. Due to these reasons, we want to seriously investigate the impact of the DDoS attack.

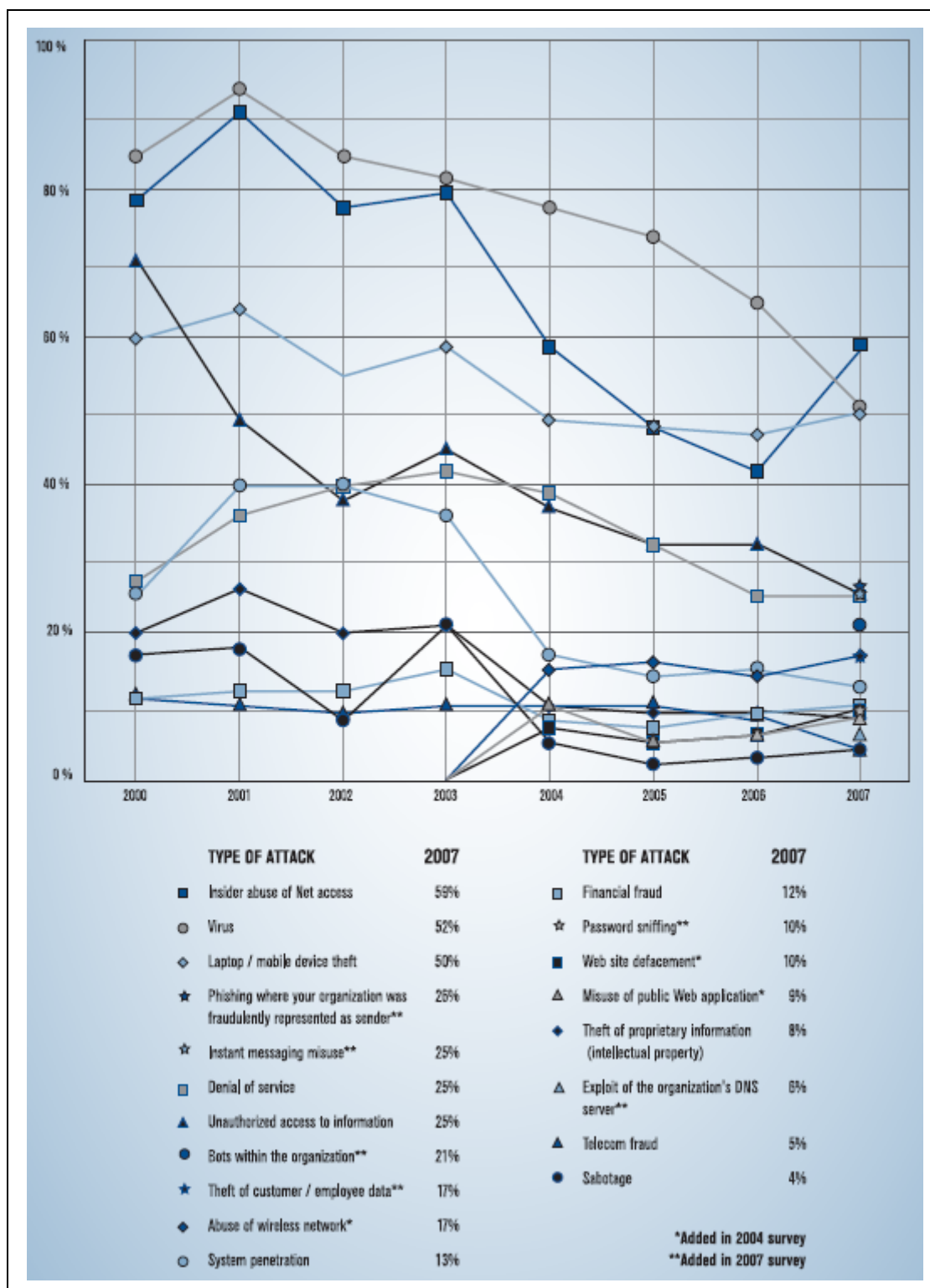


Figure 1.2-1 Types of Attacks or Misuse Detected in the last 12 Months [21]

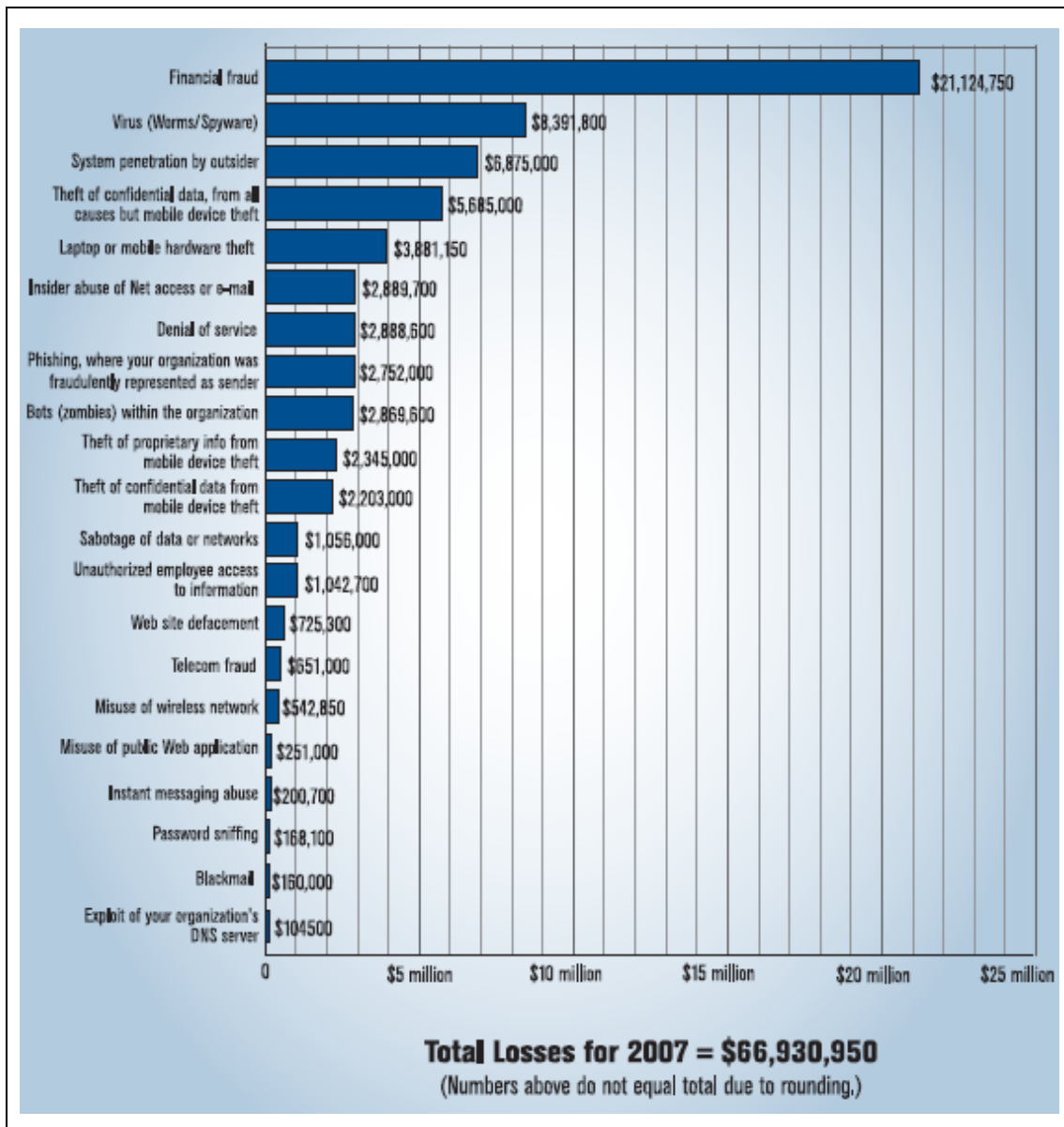
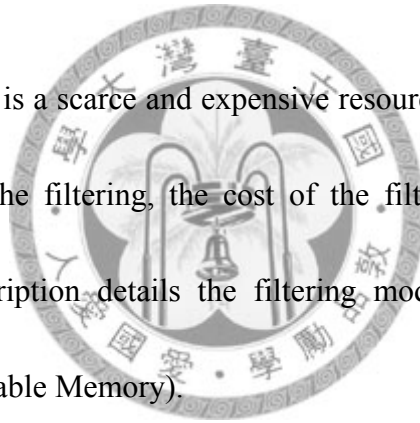


Figure 1.2-2 Dollar Amount Losses by Type of Attack [21]

Moreover, seldom works related to the DDoS attack use mathematical programming techniques to deal with the problem. We hope the DDoS attack can be more precisely formulated as a mathematical programming problem where we can obtain the optimal solution. [22] is one which formulates the DDoS attack in the timeframe of our work. However, there are still many aspects which can be further considered.

To defend against the DDoS attack, the defender (network administrator) effectively utilizes the available resources in this battle. Nevertheless, it is not objective to only consider the defender's behavior. From a comprehensive viewpoint, the attacker's behavior has to be included jointly. The attacker has its budget and resource allocation strategy. Accordingly, to resolve the issue, we hope the DDoS attack can be formulated in the offense-defense scenario where the game is between the attacker and the defender.

The filtering module is a scarce and expensive resource today. To defend against the DDoS attack under the filtering, the cost of the filter needs to be considered seriously. In [7], a description details the filtering module stored in the TCAM (Ternary Content Addressable Memory).



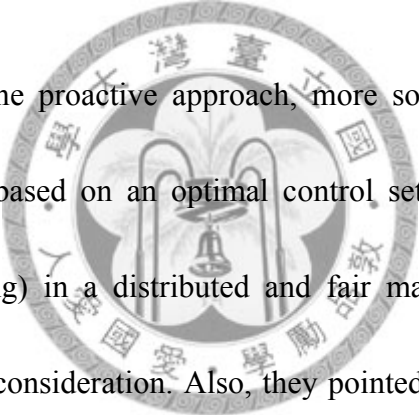
1.3 Literature Survey

In this section, we review some previous works relevant to DDoS attacks, survivability, resource allocation and autonomous system.

1.3.1 DDoS Attacks

During the DDoS attack, the server performance severely degrades and the

legitimate user seriously suffers. An effective approach to defending against the DDoS attack becomes an accelerating need. Some scholars [1][2][3] proposed effective approaches to tackle the DDoS attack by a proactive manner, in which a server under stress installs a router throttle (filter) at selected upstream routers. Hence, before aggressive packets converge to overwhelm the server, participating routers proactively regulate the concentration packet rates to a more moderate level, thus preventing an impending attack.



To further develop the proactive approach, more sophisticated algorithms are shown in [3]. In which, based on an optimal control setting, proposed algorithms achieve throttling (filtering) in a distributed and fair manner by taking important performance metrics into consideration. Also, they pointed out several objectives for these throttle algorithms such as **Fairness**, **Adaptiveness**, **Fast convergence** and **Stability**. Furthermore, the stability and convergence issues of these algorithms are also studied. With these evaluation criterions at hand, our work becomes more comprehensive.

To deeply enlighten us is the work in [7]. They treat the filtering mechanism under the DDoS attack as a resources allocation problem. Given the magnitude of the

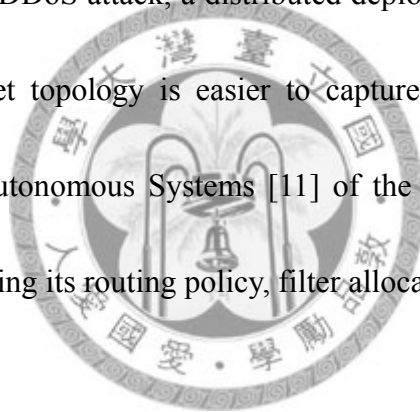
DDoS attack and the high cost of the filter today, the successful mitigation of the DDoS attack using the filtering crucially depends on the efficient allocation of filtering resources. By optimally allocating filters to attack sources, under a constraint on the number of filters, the objective to maximize the amount of the preserved good traffic can be well modeled. We will extend our work to further consider the zombie, routing, router capacity and so on.

Generally, it is not the traceback of a malicious packet but the requirement of a universal knowledge [8] which makes the defense against the DDoS attack become a tough problem. To protect victim servers from being overwhelmed by the aggregation traffic, the defender needs the help from the backbone service provider in order to follow the filtering mechanism. However, a problem about the universal knowledge appears. In what way, we can persuade backbone service providers to follow our filtering rules. Most of the time, these providers perhaps ignore such an urgent issue instead of considering the industrial benefit.

To resolve the issue, one way is to narrow our administrated areas where countering the DDoS attack becomes hopeful. In [8], Shiang Cheng and Qingguo Song proposed two perimeter-based defense mechanisms for Internet service

providers (ISPs) to provide the anti-DDoS service for their customers. These mechanisms rely completely on the edge routers to cooperatively identify the flooding sources and establish rate-limit filters to block the attack traffic. The system does not require any support from routers outside or inside the ISP.

To date, there is no authority defining Internet topology so that no one can show us a detailed picture of the Internet. This is a main challenge on many researches let alone the topology of the DDoS attack, a distributed deployment problem. Hopefully, one aspect of the Internet topology is easier to capture than the others. It is the topology made by the Autonomous Systems [11] of the Internet. The AS gives us much freedom on controlling its routing policy, filter allocation and router capacity.



One important concern in deploying any form of traffic analysis in the critical data-forwarding paths of the Internet is the performance. [6] presents a countermeasure against DDoS attacks, called the congestion-triggered packet sampling/packet filtering (CTPS/PF) architecture. With CTPS/PF, a packet sampling mechanism that is integrated with the congestion control mechanism at routers is used to detect DDoS attacks, and packet filters are activated only when sampling results warrant action. Dropped packets are sent to a CTPS system, which selects a subset of

such packets for statistical computations. Finally, if statistical results indicate anomaly (for example, a significant portion of packets contains bogus source addresses), then a control signal is sent to activate packet filters at input ports to remove malicious packets.

The current DDoS mitigation techniques are grouped into two categories in [20]. One is to mitigate the DDoS attack by two modules, adopted in [20], the attack detection module and the packet filtering module. At beginning, the attack detection module such as IDS (Intrusion Detection System) detects the malicious packet. Then, by the origin traceback, the original malicious host is blocked to prevent ongoing unwanted packets. Another technique is to regard the DDoS mitigation as a resource allocation problem, similar to our work. The technique strategically allocates network and server resources at an administrated area to prevent the resource consumption.

1.3.2 Survivability and Resource Allocation

As networks have gradually grown into large-scale systems, the definition for the network survivability can not be ignored as treated at past. System survivability is a critical part to our social and economic infrastructures as it provides many essential services to support our existence. If these systems are threatened and fail to provide

the required services, the consequences might be catastrophic and even fatal to our network.

The key components of the definition regarding survivability described in [12] are summarized in table 1.3.2-1. A standard definition of survivability for distributed network systems could be developed under the basis of these key components.

Table 1.3.2-1 The Key Components of The Definition Regarding Survivability

1. System: if the definition of survivability must vary, then at least the distributed network system environment for which it has been defined should be mentioned. The different types of essential services may warrant a special definition of survivability. In addition, whether the system is bounded or unbounded should be addressed.
2. Threat: a threat to a system may prevent the system from providing services to the user in the prescribed amount of time or may prevent the system from providing the services at all. Threats to a system can be categorized as accidental, intentional (malicious), or catastrophic. Accidental threats include software errors, hardware errors, and human errors. Intentional or malicious threats include sabotage, intrusion, or terrorist attacks. Catastrophic threats typically do not allow delivery of required service to the user, which includes acts of nature (thunderstorms, hurricanes, lightning, flood, earthquake, etc.), acts of war, and power failures.
3. Adaptability: in the event of a threat the system should have the capability to adapt to the threat and continue to provide the required service to the user.
4. Continuity of Service: services should be available to the user as defined by the requirements of the system and expected by the user, even in the event of a threat. Network performance should not appear to be degraded by the end user.
5. Time: services should be available to the user within the time required by the system and expected by the user.

In [14], Zeitlin tried to formulate the problem of the min-max integer resource allocation from both the attacker and the defender's point of view. The attacker has available M units and the defender N units. The collision of the attack and the defense occurs at “ n ” targets, meaning the attacker aims to destroy the maximum number of targets and the defender aims to minimize the destruction of targets. Therefore, the optimality condition is used to obtain a solution algorithm which is the allocation of total attacking (defending) resources among these targets.

1.3.3 Autonomous Systems

The ASs are usually classified depending on the way they manage the transit traffic [11]:

◆ Stub AS : has only one connection to another AS.
◆ Multi-homed AS : has two or more connections to other ASs but refuses to carry transit traffic.
◆ Transit AS : has two or more connections to other ASs and carries both local and transit traffic.

[11] keeps these definitions and add the following ones, considering the AS network as an undirected graph:

◆ Cycle AS : an AS that belongs to a cycle (i.e. it is on a closed path of disjoint ASs).
◆ Bridge AS : an AS which is not a cycle AS and is on a path connecting 2 cycle ASs.

[11] then divides the ASs into two exclusive broad categories:

♦ **In-mesh AS**: an AS which is a cycle AS or a bridge AS.

♦ **In-tree AS**: an AS which is not an in-mesh AS (i.e. it belongs to a tree).

[11] then defines the mesh as the set of in-mesh ASs and the forest as the set of in-tree ASs. All ASs in the forest can also be put into one of the next two exclusive categories:

♦ **Branch AS**: an in-tree AS of degree at least 2.

♦ **Leaf AS**: an in-tree AS of degree 1 (synonym of a stub).

Finally an AS can also have the following qualification(s):

♦ **Root AS**: An in-mesh AS which is the root of a tree (i.e. it is adjacent to two or more in-mesh ASs and to one or more in-tree ASs).

♦ **Relay AS**: an AS having exactly 2 connections.

♦ **Border AS**: an AS located on the diameter of the network.

♦ **Center AS**: an AS located on the radius of the network (i.e. belonging to the center of the network).

Figure 1.3.3-1 shows the different kinds of ASs in an inter-domain level network.

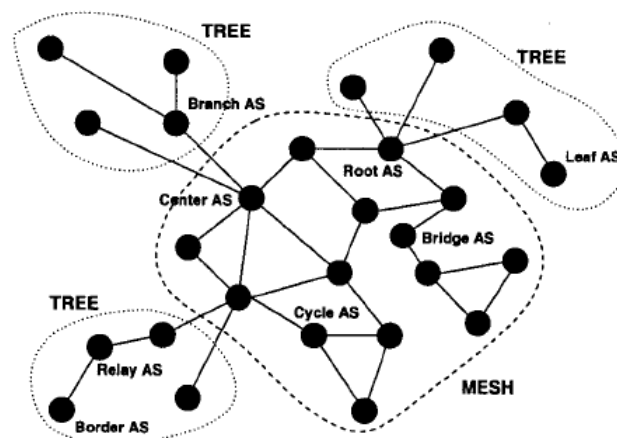


Figure 1.3.3-1 Different kinds of AS [11]

1.4 Proposed Approach

In our thesis, we solve the attacker resource allocation strategy (ARAS) problem and filter allocation strategy (FAS) by proposing a min-max mathematical model. It mathematically details the routing assignment, filter allocation and the DDoS attacker's strategies in the network. Furthermore, by applying the mathematical optimization technique to optimally solve the ARAS problem, the solution approach could be the network administrator's guideline to defend against the DDoS attack.

We formulate the problem as a mixed integer and linear programming (MILP) problem, where the problem objective is to maximize the total legitimate traffic for the defender in the DDoS attack using routing and filtering mechanism, subject to the defender's budget constraint. The DDoS attack's strategy is modeled in the outer problem, which is formulated as another MILP problem. The objective of the outer problem is to minimize the remaining legitimate traffic under a given defender's strategy and subject to the attacker's budget. We propose applying Lagrangean Relaxation method, combined with the subgradient method [13][14][19], to solve the inner problem. Ultimately, in solving the primal problem, a subgradient-based heuristic is proposed to adjust the attacker's budget allocation strategy according to the defender's strategy.

1.5 Thesis Organization

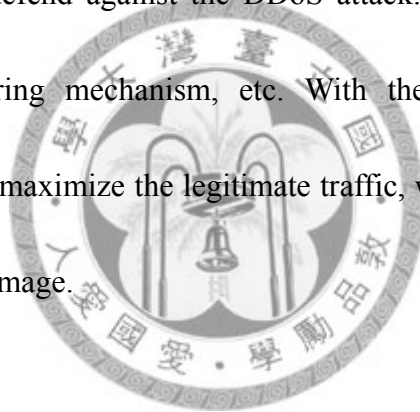
The remainder of the thesis is organized as follows. In Chapter 2, MILP formulations of the ARAS and the FAS problems are proposed. In Chapter 3, solution approaches to the ARAS and the FAS problems are proposed. In Chapter 4, the computational results of the ARAS and the FAS problems are presented. Finally, in Chapter 5, the conclusion and future works are described.



Chapter 2 Problem Formulation of the ARAS and FAS models

2.1 Problem Description

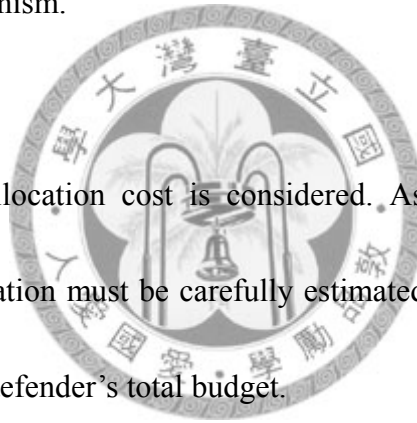
The problem we discuss here is a network administrator (defender) strategically utilizes its resources to defend against the DDoS attack. The resource may be the routing policy, the filtering mechanism, etc. With the available resources, the defender's objective is to maximize the legitimate traffic, which, in other words, is to minimize the collateral damage.



The collateral damage has to be seriously considered as the filtering deprives the legitimate traffic. The filtering literally helps mitigate the DDoS attack by regulating the aggregate traffic but it does not guarantee the maintenance of all legitimate traffic. Under the filtering, legitimate traffic is simultaneously discarded with the attack traffic. The maintenance of the remaining legitimate traffic, thus, becomes a critical issue.

In order to comprehensively describe our problem, we consider the related issues in defending against the DDoS attack.

First, the router capacity is considered. The filtering mechanism takes effect on top of the router but not all routers are capable of supporting the mechanism [15][16]. Some legacy routers exist in the network. The capacity constraint of a legacy router will refuse the filter function. In an AS, not all routers have the same capacity which allows the filtering mechanism.



Second, the filter allocation cost is considered. As the defender's budget is limited, each filter allocation must be carefully estimated. The total filter allocation cost can not exceed the defender's total budget.

Third, the attacker's budget is considered. The DDoS attacker has the attack budget which allows it to configure the DDoS attack. The worst-case scenario [15][16] is therefore assumed. Generally, the DDoS attacker compromises a cohort of intermediate hosts (zombies) before the attack. In our work, we consider the DDoS attacker is able to allocate the attack budget to compromise zombies for the attack traffic used in the DDoS attack. The DDoS attacker needs to strategically allocate

resources for the following attack. Overall, the attacker can decide budget allocation and attack traffic. Moreover, the legitimate traffic is still possibly sent from the zombie because it may be controlled only partially.

Fourth, there are multiple victim servers. The DDoS attacker tries to inundate multiple victim servers by making the aggregate traffic exceed the aggregate traffic threshold [8].

Besides, due to the importance of the network topology, a more controllable area is considered. We consider an AS (autonomous system) in our work [11]. The defender has a more effective management in an AS. The routing policy in the AS is also considered in our work. The defender routes traffic to maintain the remaining legitimate traffic. Seldom papers regarding DDoS attack take care of the routing. In our work, a legitimate packet or a malicious packet, as flowed into the AS, is strategically routed according to the defender's routing assignment.

2.2 Problem Formulation of the ARAS Model

To defend against the DDoS attack by the filtering and routing is modeled as a min-max optimization problem, where the objective is to minimize the maximized

legitimate traffic under the filtering. In the FAS model, a defender tries to maximize the remaining legitimate traffic by the filtering and routing assignment. In the ARAS model, the DDoS attacker tries to minimize the remaining legitimate traffic by the budget allocation and attack traffic.

To precisely illustrate the problem, we show the attack-defense scenario in the figures that follows. (Figure 2.2-1) is the initial network topology. At first, only good user (legitimate) traffic exists (Figure 2.2-2). Next, the DDoS attacker compromises zombies to obtain the maximum attack traffic by the attack budget allocation (Figure 2.2-3). Afterwards, DDoS attacker sends attack traffic to attack victim servers (Figure 2.2-4). As the victim servers experience the aggregate traffic exceeds the aggregate traffic threshold (Figure 2.2-5), filters are allocated to regulate the aggregate traffic (Figure 2.2-6). The routing policy is also adopted to reroute the attack traffic (Figure 2.2-7). Finally, the attack traffic is rerouted to finish the defense (Figure 2.2-8).

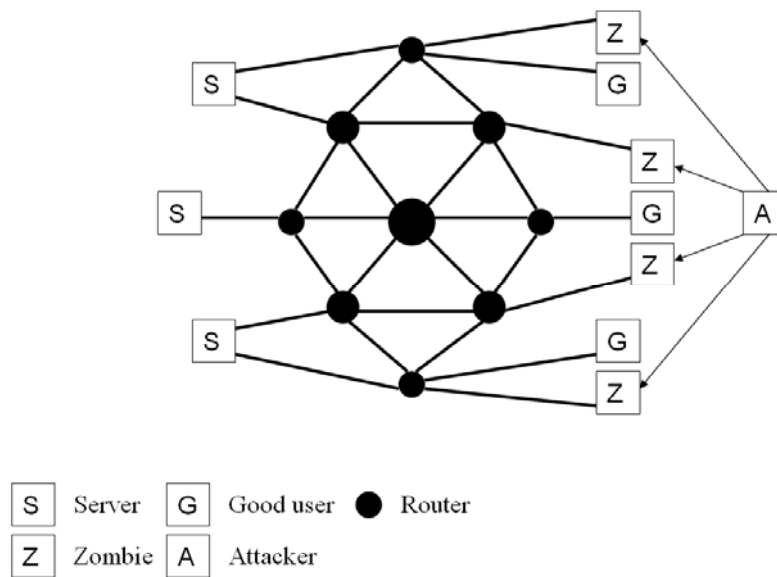


Figure 2.2-1 The initial network topology

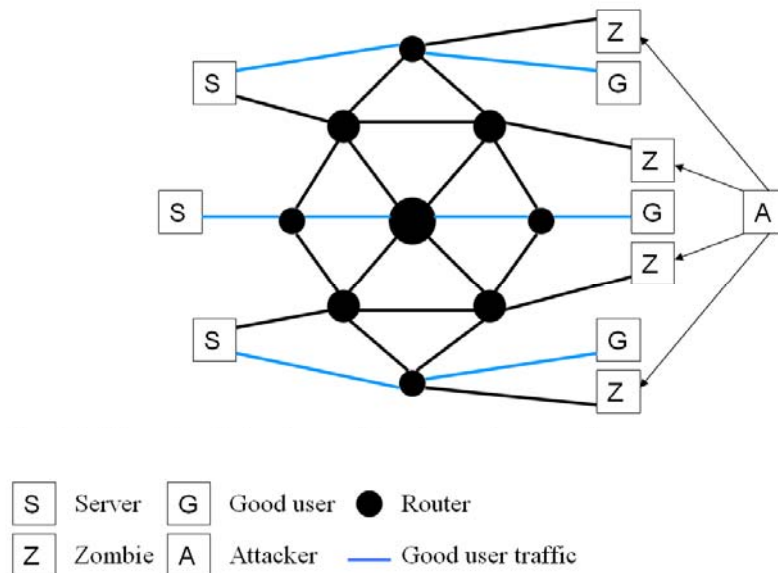


Figure 2.2-2 The network topology with only good user traffic

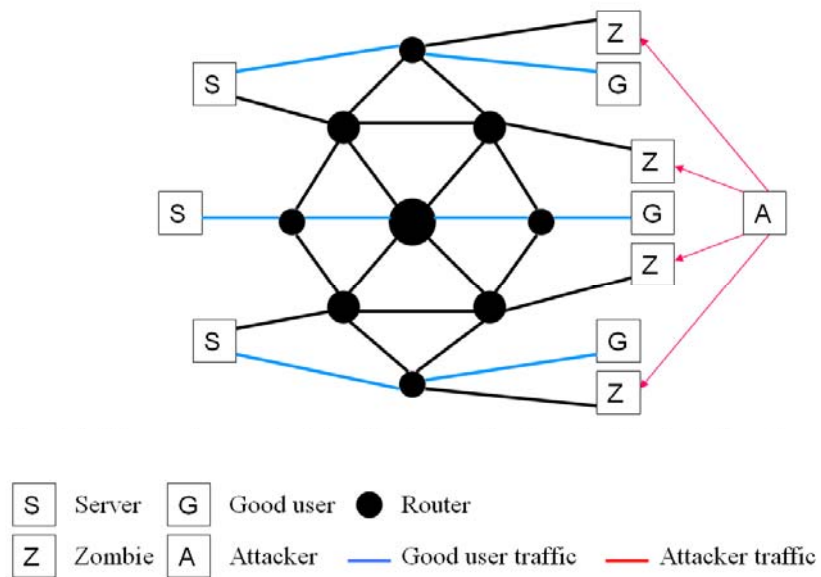


Figure 2.2-3
The maximum attack traffic obtained by the attack budget allocation

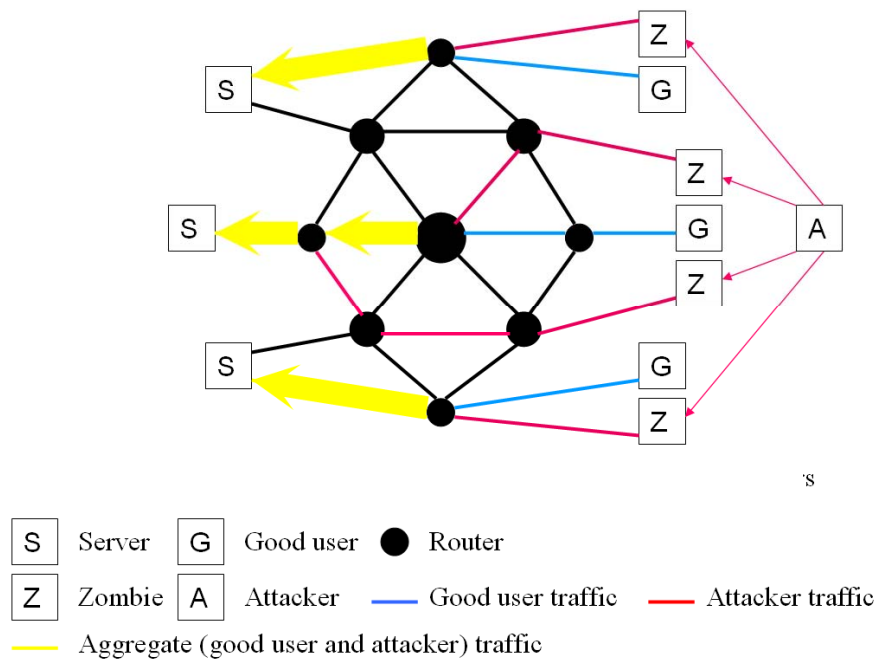


Figure 2.2-4 The (real) attack traffic sent by zombies to attack victim servers

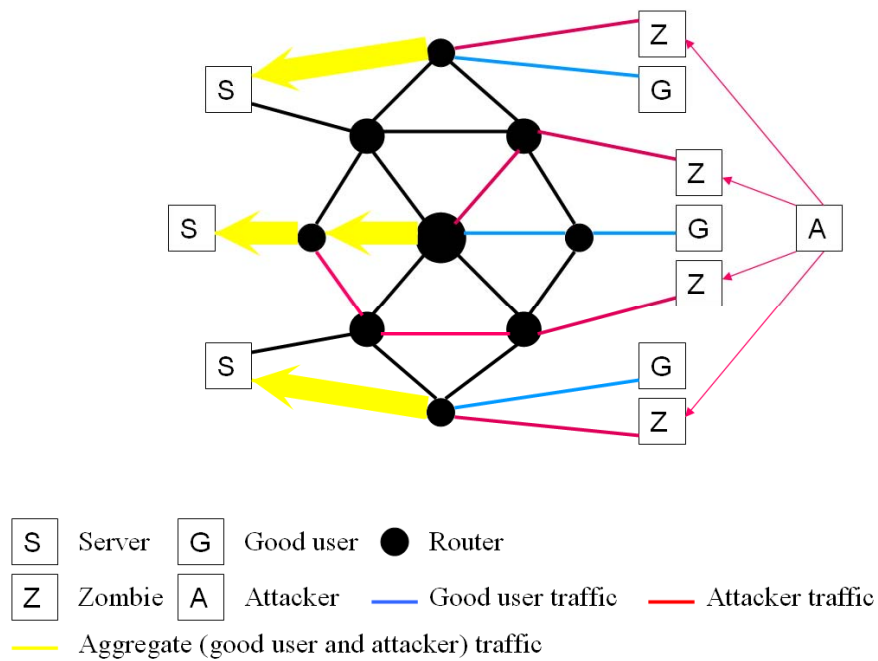


Figure 2.2-5 The network topology with the aggregate traffic

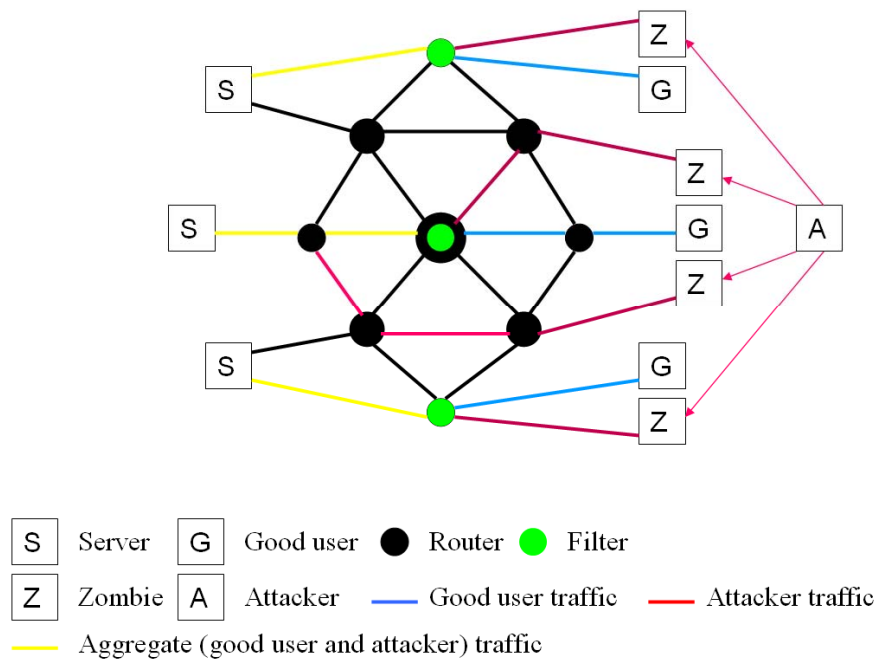


Figure 2.2-6 The network topology with filters allocated

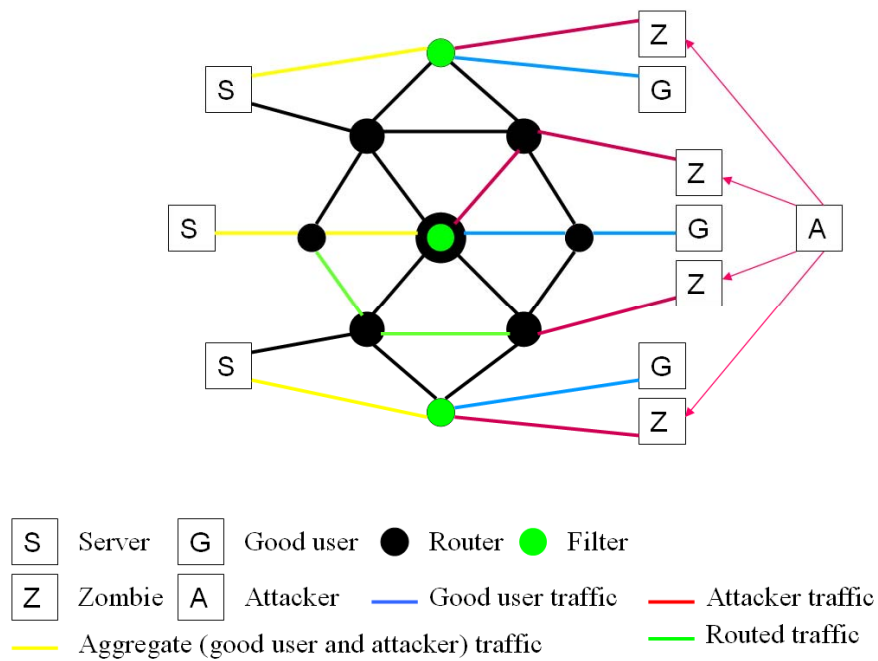


Figure 2.2-7 The network topology with the routing policy adopted

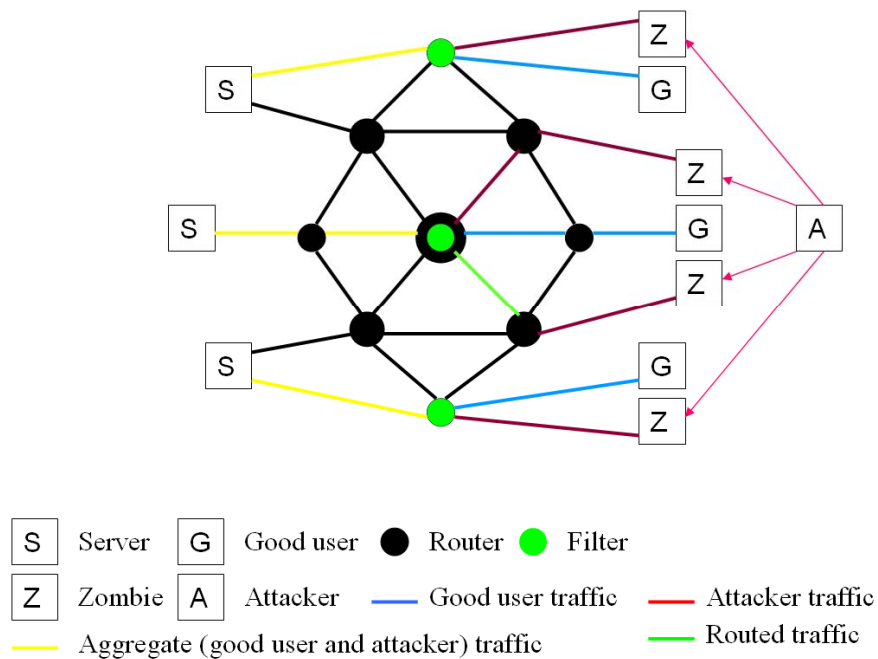


Figure 2.2-8 The rerouted (real) attack traffic

So far we have defined an optimization-based problem about which the detailed assumptions, objective, constraints and decision variables, are all listed in the following tables.

Table 2-1 Problem Assumption and Description of the ARAS Model

Assumption
<ul style="list-style-type: none"> ◆ The attacker attacks multiple victim servers in the network topology. ◆ The attacker compromises hosts as zombies. ◆ The attacker decides the attack traffic for each OD pair. ◆ The routing assignment of the attack traffic is determined by the defender. ◆ Only AS (Autonomous System) level networks are considered. ◆ The attacker is outside an AS. ◆ The zombie is outside an AS. ◆ The good traffic is also from zombies. ◆ The defender determines the routing assignment of the good traffic. ◆ The defender allocates filters to defend the attack traffic. ◆ The defender adopts the routing assignment to defend the attack traffic. ◆ IP Spoofing exists. ◆ The attack traffic is identified by its volume (which is far more than the good traffic volume). ◆ The filtering remaining rate is not zero. ◆ Both the attacker and the defender have complete information.
Given
<ul style="list-style-type: none"> ◆ The network topology (AS level) ◆ Multiple victim servers ◆ Zombie hosts ◆ The good traffic from zombies ◆ The defender's total budget ◆ The attacker's total budget ◆ The filter allocation cost ◆ The filtering remaining rate ◆ The routing path that the defender can choose ◆ The router capacity

Objective	
◆	To minimize the maximized good traffic
Subject to	
◆	The attack traffic and budget allocation
◆	The filter allocation
◆	The routing assignment
◆	The threshold of the aggregate traffic
◆	The threshold of the remaining good traffic percentage
◆	The filter allocation cost
◆	The router capacity
◆	The choice of the filtering remaining rate
To determine	
Defender :	
◆	The filter allocation strategy
◆	The filtering remaining rate
◆	The routing assignment
Attacker :	
◆	The budget allocation on zombies
◆	The attack traffic for each OD pair

We model the problem based on the above assumptions as a min-max mathematical programming problem. Notations and parameters used in this model are presented below.

Table 2-2 Given Parameters of the ARAS Model

Given Parameters	
Notation	Description
N	The index set of all nodes in an AS
E	The index set of all entry nodes, where $E \subset N$
S	The index set of all victim servers, where $S \subset N$
\hat{g}_v	The threshold of the aggregate traffic below which the aggregate traffic is regulated to defend the DDoS attack for a victim server v , where $v \in S$
Z	The index set of all zombies
ϕ_k	The threshold of the remaining good traffic percentage for zombie k that the remaining good traffic over the good traffic for zombie k must exceed to maintain the service quality, where $k \in Z$, $0 \leq \phi_k \leq 1$
R_i	The router capacity on a node i , where $i \in N$

B	The defender's total budget
A	The attacker's total budget
W_k	The index set of all OD pairs, where the origin is node o and the destination is node d , where $o \in E, d \in S, k \in Z$
P_w	The index set of all candidate paths of an OD pair w , where $w \in W_k$
N_p	The index set of all nodes on a candidate path p , where $p \in P_w, N_p \subset N$
δ_{pv}	The indicator function, which is 1 if a node v is on a path p ; and 0 otherwise (where $v \in N, p \in P_w$)
γ_w	The good traffic on an OD pair w , where $w \in W_k, k \in Z$
ω_i	All possible values, between 0 and 1, of F_i on a node i , where $i \in N$
C_i	The cost to allocate the filter on a node i , where $i \in N$
FC_i	The router capacity required to allocate the filter on a node i , where $i \in N$

Some points regarding these parameters need to be addressed here. The threshold of the aggregate traffic, \hat{g}_v , means the defender adopts the filtering and the routing assignment to route and regulate the aggregate traffic below this threshold to defend against the DDoS attack. In the meantime, in order to maintain the service quality of the good traffic, the threshold of the remaining good traffic percentage, ϕ_k , has to be considered also, which means the remaining good traffic after the filtering and the routing assignment over the original good traffic must exceed this threshold for each zombie. Both given parameters are crucial in the following formulation. All notations of decision variables are listed in Table 2-3.

Table 2-3 Decision Variables of the ARAS Model

Decision Variables	
Notation	Description
y_i	0 if the filter is allocated on a node i ; and 1 otherwise (where $i \in N$)
F_i	The filtering remaining rate on a node i , where $i \in N$
x_p	1 if a path p is selected as the routing path; and 0 otherwise (where $p \in P_w$)
C_w	The attack budget allocated on an OD pair w , where $w \in W_k, k \in Z$
$\zeta_w(C_w)$	The maximum attack traffic, which is the linear function of an OD pair w that is a function of the attack budget, where $w \in W_k, k \in Z$
β_w	The (real) attack traffic on an OD pair w , where $w \in W_k, k \in Z$

The mathematical model (IP 1) of the ARAS problem is formulated and shown as follows.

Objective function :

$$Z_{IP1} = \min_{C_w, \beta_w} \max_{F_i, x_p, y_i} \sum_{k \in Z} \sum_{w \in W_k} \sum_{p \in P_w} x_p \gamma_w \prod_{i \in N_p} F_i \quad (\text{IP 1})$$

Subject to :

$$\beta_w \geq 0 \quad \forall k \in Z, w \in W_k \quad (\text{IP 1.1})$$

$$\beta_w \leq \zeta_w(C_w) \quad \forall k \in Z, w \in W_k \quad (\text{IP 1.2})$$

$$\sum_{k \in Z} \sum_{w \in W_k} C_w \leq A \quad (\text{IP 1.3})$$

$$0 \leq C_w \leq A \quad \forall k \in Z, w \in W_k \quad (\text{IP 1.4})$$

$$y_i \leq F_i \quad \forall i \in N \quad (\text{IP 1.5})$$

$$y_i = 0 \text{ or } 1 \quad \forall i \in N \quad (\text{IP 1.6})$$

$$\sum_{p \in P_w} x_p = 1 \quad \forall k \in Z, w \in W_k \quad (\text{IP 1.7})$$

$$x_p = 0 \text{ or } 1 \quad \forall k \in Z, w \in W_k, p \in P_w \quad (\text{IP 1.8})$$

$$\sum_{k \in Z} \sum_{w \in W_k} \sum_{p \in P_w} x_p \delta_{pv} (\gamma_w + \beta_w) \prod_{i \in N_p} F_i \leq \hat{g}_v \quad \forall v \in S \quad (\text{IP 1.9})$$

$$\frac{\sum_{w \in W_k} \sum_{p \in P_w} x_p \gamma_w \prod_{i \in N_p} F_i}{\sum_{w \in W_k} \gamma_w} \geq \phi_k \quad \forall k \in Z \quad (\text{IP 1.10})$$

$$\sum_{i \in N} C_i \cdot (1 - y_i) \leq B \quad (\text{IP 1.11})$$

$$FC_i \cdot (1 - y_i) \leq R_i \quad \forall i \in N \quad (\text{IP 1.12})$$

$$F_i \in \omega_i \quad \forall i \in N. \quad (\text{IP 1.13})$$

Explanation of the Mathematical Formulation:

- **Objective function:** The objective is to minimize the maximized remaining good traffic. In the inner problem, the defender tries to maximize the remaining good traffic by the filtering and the routing assignment. In the outer problem, the attacker tries to minimize the remaining good traffic by the attack budget allocation and the attack traffic.
- **Constraint (IP 1.1)** enforces that the attack traffic must be nonnegative.
- **Constraint (IP 1.2)** requires that the (real) attack traffic, β_w , for an OD pair w must not exceed the maximum attack traffic, $\zeta_w(C_w)$, on an OD pair w .
- **Constraint (IP 1.3)** restricts that the total allocated attack budget, $\sum_{k \in Z} \sum_{w \in W_k} C_w$, must not exceed the attacker's total budget A .
- **Constraint (IP 1.4)** restricts that the attack budget allocated on an OD pair w must not exceed the attacker's total budget A .
- **Constraint (IP 1.5)** requires that the filtering on a node i is available as the filter is allocated.
- **Constraint (IP 1.6)** enforces that y_i is 0 if the filter is allocated on a node i ; and 1 otherwise.

- **Constraint (IP 1.7)** enforces that only one path is selected for an OD pair w .
- **Constraint (IP 1.8)** limits the value of x_p to 0 or 1.
- **Constraint (IP 1.9)** enforces that the aggregate traffic to each victim server v under the filtering and the routing assignment must not exceed the threshold of the aggregate traffic, \hat{g}_v , for each victim server v .
- **Constraint (IP 1.10)** enforces that the remaining good traffic percentage for each zombie k under the filtering and the routing assignment must exceed the threshold of the remaining good traffic percentage, ϕ_k , for each zombie k .
- **Constraint (IP 1.11)** restricts that the total filter allocation cost must not exceed the defender's total budget B .
- **Constraint (IP 1.12)** enforces that the router capacity required to allocate the filter on a node i must not exceed the router capacity on a node i .
- **Constraint (IP 1.13)** limits all possible values of the filtering remaining rate on a node i .

2.3 Problem Formulation of the FAS Model

It is very difficult to solve a two levels problem directly due to its intractable property. In order to break through the difficulty, we use a two-phase approach.

We formulate the defender's behavior in the FAS model where we assume the decision variables related to the attacker are given. They are marked gray in the table that follows. After the FAS problem is solved, the result is used as an input to the ARAS problem to develop an advanced budget allocation strategy. The given parameters of the FAS model are listed in Table 2-4.

Table 2-4 Given Parameters of the FAS Model

Given Parameters	
Notation	Description
N	The index set of all nodes in an AS
E	The index set of all entry nodes, where $E \subset N$
S	The index set of all victim servers, where $S \subset N$
\hat{g}_v	The threshold of the aggregate traffic below which the aggregate traffic is filtered to defend DDoS attack for a victim server v , where $v \in S$
Z	The index set of all zombies
ϕ_k	The threshold of the remaining good traffic percentage for zombie k that the remaining good traffic over the good traffic for zombie k must exceed to maintain the service quality, where $k \in Z$, $0 \leq \phi_k \leq 1$
R_i	The router capacity on a node i , where $i \in N$
B	The defender's total budget
A	The attacker's total budget
W_k	The index set of all OD pairs, where the origin is node o and the destination is node d , where $o \in E, d \in S, k \in Z$
P_w	The index set of all candidate paths of an OD pair w , where $w \in W_k$
N_p	The index set of all nodes on a candidate path p , where $p \in P_w, N_p \subset N$
δ_{pv}	The indicator function, which is 1 if a node v is on the path p ; and 0 otherwise (where $v \in N, p \in P_w$)
γ_w	The good traffic on an OD pair w , where $w \in W_k, k \in Z$
ω_i	All possible values, between 0 and 1, of F_i on a node i , where $i \in N$
C_i	The cost to allocate the filter on a node i , where $i \in N$
FC_i	The router capacity required to allocate the filter on a node i , where $i \in N$
C_w	The attack budget allocated on an OD pair w , where $w \in W_k, k \in Z$

$\zeta_w(C_w)$	The maximum attack traffic, which is the linear function of an OD pair w that is a function of the attack budget, where $w \in W_k, k \in Z$
β_w	The (real) attack traffic on an OD pair w , where $w \in W_k, k \in Z$

Note that C_w , $\zeta_w(C_w)$ and β_w are marked in gray, which are decision variables in the ARAS problem. However, as we hopes to solve the problem by a two phase approach, these variables become given parameters in the FAS model. Table 2-5 lists all decision variables used in the FAS model.

Table 2-5 Decision Variables of the FAS Model

Decision Variables	
Notation	Description
y_i	0 if the filter is allocated on a node i ; and 1 otherwise (where $i \in N$)
F_i	The filtering remaining rate on a node i , where $i \in N$
x_p	1 if a path p is selected as the routing path; and 0 otherwise (where $p \in P_w$)

The mathematical model (IP 2) of the FAS problem is shown as follows, which merely formulate the defender behavior. In this model, we transform the objective function from the maximization problem into the minimization problem. Then an auxiliary variable l_p is introduced to replace the product form $\prod_{i \in N_p} F_i$ in the objective function, constraint (IP 2.7) and constraint (IP 2.8).

Objective function :

$$\begin{aligned}
Z_{IP2} &= \max_{F_i, x_p, y_i} \sum_{k \in Z} \sum_{w \in W_k} \sum_{p \in P_w} x_p \gamma_w \prod_{i \in N_p} F_i \\
&\equiv \min_{F_i, x_p, y_i} - \sum_{k \in Z} \sum_{w \in W_k} \sum_{p \in P_w} x_p \gamma_w \prod_{i \in N_p} F_i \\
&\Rightarrow \min_{F_i, x_p, y_i} - \sum_{k \in Z} \sum_{w \in W_k} \sum_{p \in P_w} x_p \gamma_w l_p
\end{aligned} \tag{IP 2}$$

Subject to :

$$\prod_{i \in N_p} F_i \geq l_p \quad \forall k \in Z, w \in W_k, p \in P_w \tag{IP 2.1}$$

$$\varepsilon \leq l_p \leq 1 \quad \forall k \in Z, w \in W_k, p \in P_w \tag{IP 2.2}$$

$$y_i \leq F_i \quad \forall i \in N \tag{IP 2.3}$$

$$y_i = 0 \text{ or } 1 \quad \forall i \in N \tag{IP 2.4}$$

$$\sum_{p \in P_w} x_p = 1 \quad \forall k \in Z, w \in W_k \tag{IP 2.5}$$

$$x_p = 0 \text{ or } 1 \quad \forall k \in Z, w \in W_k, p \in P_w \tag{IP 2.6}$$

$$\begin{aligned}
&\sum_{k \in Z} \sum_{w \in W_k} \sum_{p \in P_w} x_p \delta_{pv} (\gamma_w + \beta_w) \prod_{i \in N_p} F_i \leq \hat{g}_v \\
&\Rightarrow \sum_{k \in Z} \sum_{w \in W_k} \sum_{p \in P_w} x_p \delta_{pv} (\gamma_w + \beta_w) l_p \leq \hat{g}_v
\end{aligned} \quad \forall v \in S \tag{IP 2.7}$$

$$\begin{aligned}
&\frac{\sum_{w \in W_k} \sum_{p \in P_w} x_p \gamma_w \prod_{i \in N_p} F_i}{\sum_{w \in W_k} \gamma_w} \geq \phi_k \\
&\Rightarrow \frac{\sum_{w \in W_k} \sum_{p \in P_w} x_p \gamma_w l_p}{\sum_{w \in W_k} \gamma_w} \geq \phi_k
\end{aligned} \quad \forall k \in Z \tag{IP 2.8}$$

$$\sum_{i \in N} C_i \cdot (1 - y_i) \leq B \tag{IP 2.9}$$

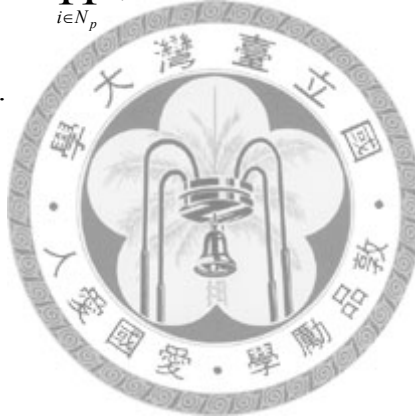
$$FC_i \cdot (1 - y_i) \leq R_i \quad \forall i \in N \tag{IP 2.10}$$

$$F_i \in \omega_i \quad \forall i \in N. \tag{IP 2.11}$$



Explanation of the Mathematical Formulation:

- **Objective function:** The objective is to maximize the remaining good traffic by the filtering and the routing assignment where the decision variables of the outer problem are given.
- **Constraint (IP 2.1) and (IP 2.2)** introduces auxiliary constraints which helps variable l_p to replace the $\prod_{i \in N_p} F_i$ in order to simplify the problem.
- **Constraint (IP 2.3) ~ (IP 2.11)** are equal to **Constraint (IP 1.5) ~ (IP 1.13)** except the product forms, $\prod_{i \in N_p} F_i$, in (IP 2.7) and (IP 2.8) are replaced by the auxiliary variable l_p .

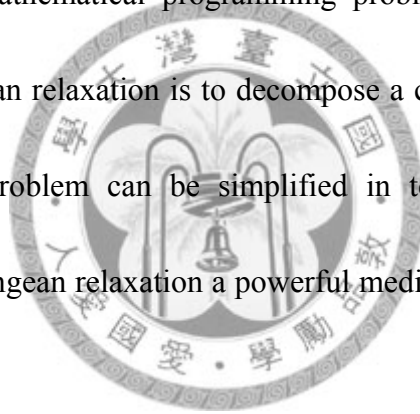


Chapter 3 Solution Approach

3.1 Solution Approach for the FAS Model

3.1.1 Lagrangean Relaxation Method

During the 1970s, the Lagrangean relaxation method had been greatly adopted to cope with large-scale mathematical programming problems [18]. The underlying concepts of the Lagrangean relaxation is to decompose a complex problem. Thus, an originally hard solved problem can be simplified in terms of complexities and difficulties making Lagrangean relaxation a powerful medium in solving optimization problems.



There are several applications by applying Lagrangean relaxation method such as integer programming, linear programming combinatorial optimization, and non-linear programming problems. As its efficiency and effectiveness in obtaining appropriate solutions, Lagrangean relaxation has become a generally recognized tool while dealing with mathematics-related problems. The method's performance is excellent, especially when dealing with large-scale mathematical programming applications

[19].

To remove out constraints and instead add them into the objective function with associated Lagrangean multipliers (μ) is the principle of the Lagrangean relaxation method [19]. The concept to form this method stems from an observation that many difficult integer programming problems can be formulated as a relatively easy problem with a set of side constraints. By adopting the transformation of the primal problem (P) into a Lagrangean relaxation problem (LR_u), we can divide the originally hard mathematical model into several independent subproblems and optimally conquer them by proper algorithms. Furthermore, some hints about the boundary of the objective function value could be offered by the Lagrangean relaxation method.



The feasibility of the result for (P) can be tested by solving (LR_u). Suppose all the constraints in (P) could be satisfied by the outcome. We will reach a primal feasible solution; otherwise, prosperous heuristics to tune the infeasible solution into a feasible one need to be constructed.

Moreover, in order to improve a solution quality, the adjustment by employing Lagrangean multipliers (μ) is required, which adjust the original heuristic to a

Lagrangian-based modified heuristic. Each feasible solution of (P) generates an upper bound (UB) of the optimal value of (P); thus, the optimal solution to the primal problem, indeed, appears between the Lagrangean LB and the primal feasible solution values.

The principal concepts of the Lagrangean relaxation method have been illustrated in Figure 3-1, and a detailed flow chart of it in Figure 3-2.

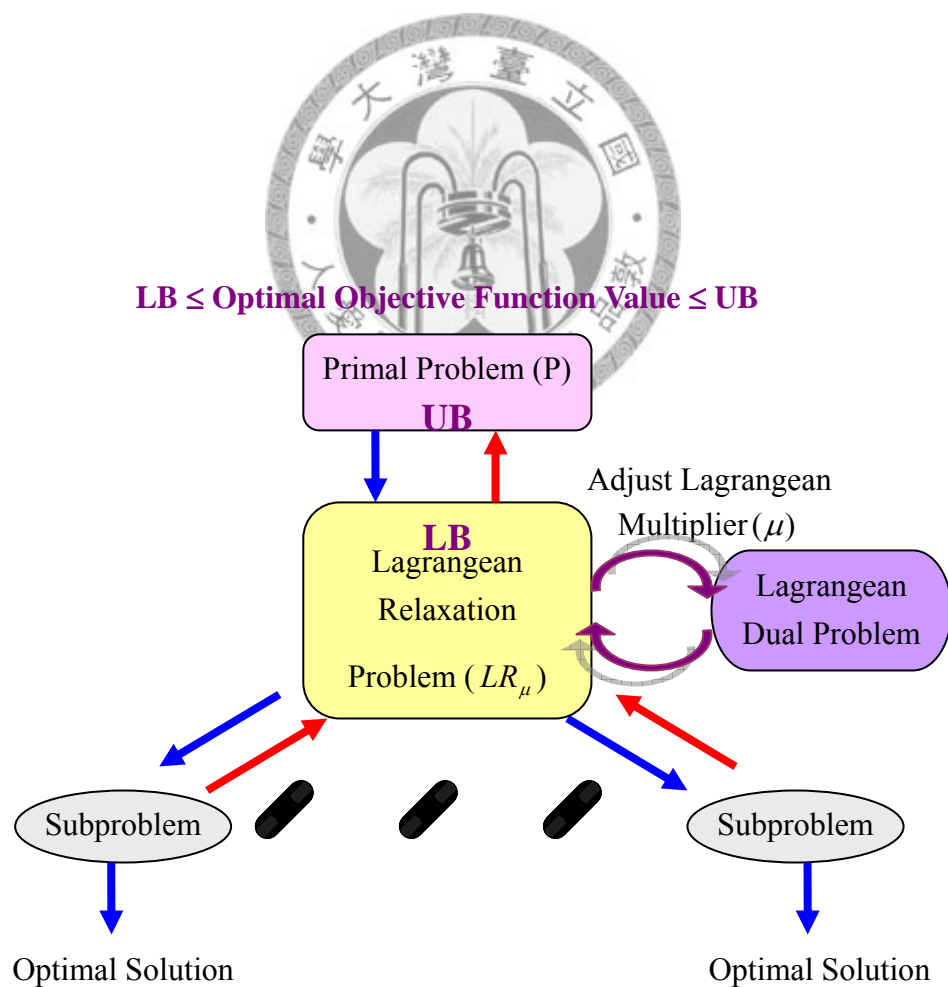


Figure 3-1 Concept of the Lagrangean Relaxation Method

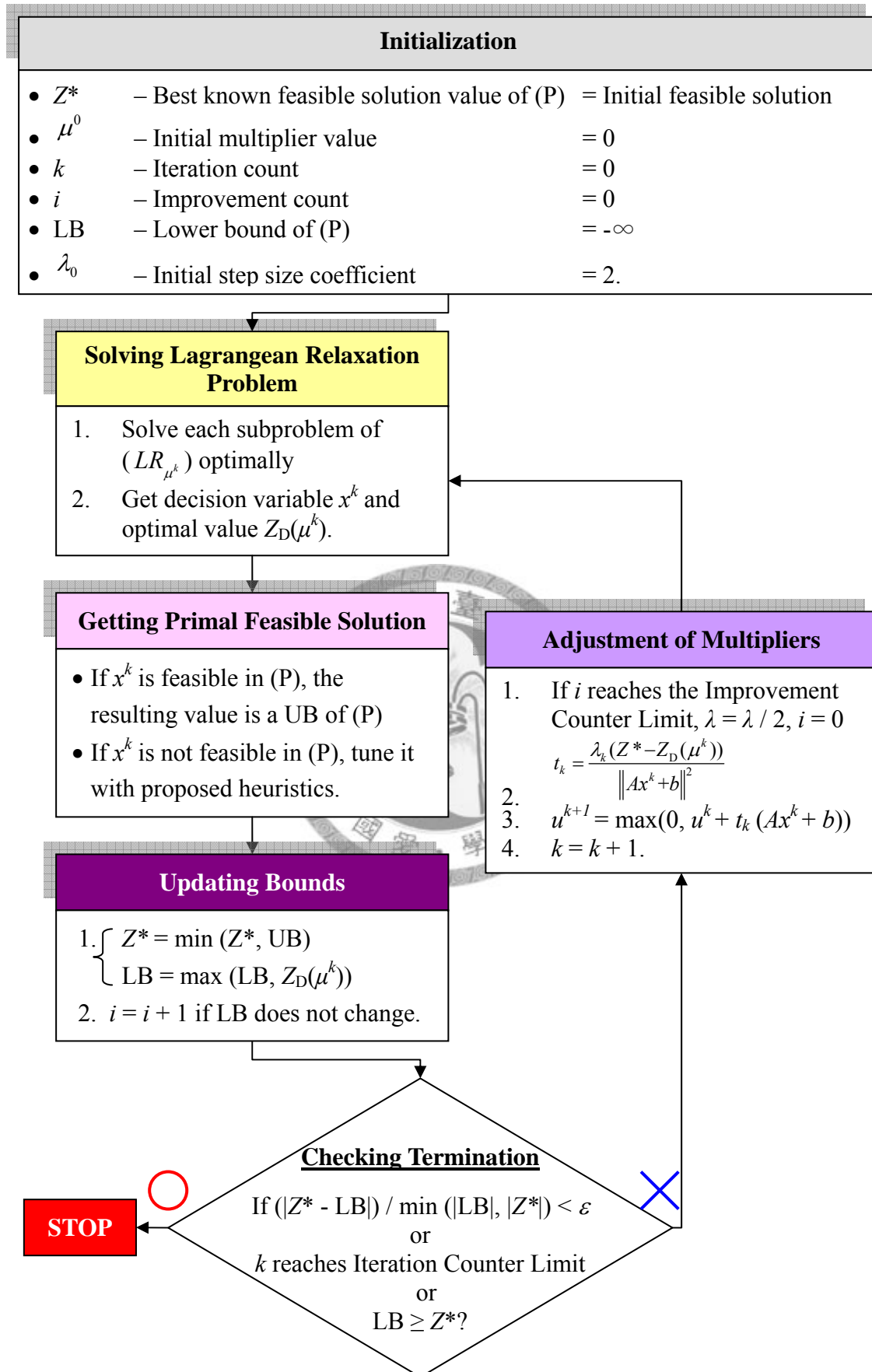


Figure 3-2 Lagrangean Relaxation Method Procedure

3.1.2 Lagrangean Relaxation

We apply the Lagrangean relaxation methodology to develop our solution approach. At first, we make an adjustment to the auxiliary constraint (IP 2.1) as it includes a value, $\prod_{i \in N_p} F_i$, calculated by a series of product, which makes this problem intractable and complex due to its non-linearity. We transform it from the product form into the logarithm form without losing its optimality.

$$\begin{aligned}
 \prod_{i \in N_p} F_i &\geq l_p \\
 \Rightarrow \log\left(\prod_{i \in N_p} F_i\right) &\geq \log(l_p) \\
 \Rightarrow \sum_{i \in N_p} \log F_i &\geq \log(l_p)
 \end{aligned}
 \quad \forall k \in Z, w \in W_k, p \in P_w \quad (\text{IP 2.1})$$

After the transformation, the Lagrangean relaxation method is applied to transform the primal problem (IP 2) into the following Lagrangean relaxation problem (LR 1), where Constraints (IP 2.1), (IP 2.3), (IP 2.7), (IP 2.8) and (IP 2.9) are relaxed.

$$\sum_{i \in N_p} \log F_i \geq \log(l_p) \quad \forall k \in Z, w \in W_k, p \in P_w \quad (\text{IP 2.1})$$

$$y_i \leq F_i \quad \forall i \in N \quad (\text{IP 2.3})$$

$$\sum_{k \in Z} \sum_{w \in W_k} \sum_{p \in P_w} x_p \delta_{pv} (\gamma_w + \beta_w) l_p \leq \hat{g}_v \quad \forall v \in S \quad (\text{IP 2.7})$$

$$\frac{\sum_{w \in W_k} \sum_{p \in P_w} x_p \gamma_w l_p}{\sum_{w \in W_k} \gamma_w} \geq \phi_k \quad \forall k \in Z \quad (\text{IP 2.8})$$

$$\sum_{i \in N} C_i \cdot (1 - y_i) \leq B \quad (\text{IP 2.9})$$

With a vector of Lagrangean multipliers, the Lagrangean relaxation problem of (IP 2) is transformed as follows.

$$\begin{aligned}
& Z_D(\mu_1, \mu_2, \mu_3, \mu_4, \mu_5) \\
& = \min_{F_i, x_p, y_i} - \sum_{k \in Z} \sum_{w \in W_k} \sum_{p \in P_w} x_p \gamma_w l_p \\
& + \sum_{k \in Z} \sum_{w \in W_k} \sum_{p \in P_w} \mu_{kwp}^1 [\log(l_p) - \sum_{i \in N_p} \log F_i] \\
& + \sum_{i \in N} \mu_i^2 [y_i - F_i] \\
& + \sum_{v \in S} \mu_v^3 [\sum_{k \in Z} \sum_{w \in W_k} \sum_{p \in P_w} x_p \delta_{pv} (\gamma_w + \beta_w) l_p - \hat{g}_v] \\
& + \sum_{k \in Z} \mu_k^4 [\phi_k \sum_{w \in W_k} \gamma_w - \sum_{w \in W_k} \sum_{p \in P_w} x_p \gamma_w l_p] \\
& + \mu^5 [\sum_{i \in N} C_i \cdot (1 - y_i) - B]
\end{aligned} \tag{LR 1}$$

Subject to:

$$\varepsilon \leq l_p \leq 1 \quad \forall k \in Z, w \in W_k, p \in P_w \tag{LR 1.1}$$

$$y_i = 0 \text{ or } 1 \quad \forall i \in N \tag{LR 1.2}$$

$$\sum_{p \in P_w} x_p = 1 \quad \forall k \in Z, w \in W_k \tag{LR 1.3}$$

$$x_p = 0 \text{ or } 1 \quad \forall k \in Z, w \in W_k, p \in P_w \tag{LR 1.4}$$

$$FC_i \cdot (1 - y_i) \leq R_i \quad \forall i \in N \tag{LR 1.5}$$

$$F_i \in \omega_i \quad \forall i \in N. \tag{LR 1.6}$$

The Lagrangean multipliers μ_2 , μ_3 and μ_4 are one dimensional vectors, and μ_1 is three dimensional vectors, where all multipliers are nonnegative. In order to solve the Lagrangean relaxation problem, we decompose (LR1) into independent subproblems shown below.

Subproblem 1.1 (related to decision variables x_p, l_p)

$$\begin{aligned}
 Z_{Sub1.1}(\mu_1, \mu_3, \mu_4) = \min & - \sum_{k \in Z} \sum_{w \in W_k} \sum_{p \in P_w} x_p \gamma_w l_p \\
 & + \sum_{k \in Z} \sum_{w \in W_k} \sum_{p \in P_w} \mu_{kwp}^1 [\log(l_p)] \\
 & + \sum_{v \in S} \mu_v^3 \left[\sum_{k \in Z} \sum_{w \in W_k} \sum_{p \in P_w} x_p \delta_{pv} (\gamma_w + \beta_w) l_p \right] \\
 & + \sum_{k \in Z} \mu_k^4 \left[- \sum_{w \in W_k} \sum_{p \in P_w} x_p \gamma_w l_p \right]
 \end{aligned} \tag{Sub 1.1}$$

Subject to:

$$\sum_{p \in P_w} x_p = 1 \quad \forall k \in Z, w \in W_k \tag{Sub 1.1.1}$$

$$x_p = 0 \text{ or } 1 \quad \forall k \in Z, w \in W_k, p \in P_w \tag{Sub 1.1.2}$$

$$\varepsilon \leq l_p \leq 1 \quad \forall k \in Z, w \in W_k, p \in P_w. \tag{Sub 1.1.3}$$

As (Sub 1.1) is a complicated problem, we have to make the simplification in order to optimally solve it. First, for the decision variable x_p , we narrow its region by a prior designation of the routing path. We designate a number of the routing path for the index set P_w . Moreover, as the decision variable l_p is associated with the logarithm, we restrict its value to between ε and 1. Finally, (Sub 1.1) can be further decomposed into $|W_k| \times |Z|$ independent sub problems. The algorithm for solving (Sub 1.1) is presented as the pseudo code in the following table.

For each OD pair $w \in W_k, k \in Z$ {

For each $p \in P_w$ {

1. Select p as the routing path.

2. For the selected path, record the best result by setting x_p to 1 and l_p to ε , 1 or its first derivative value at 0.

3. For the unselected path, record x_p and l_p by setting x_p to 0 and l_p to ε .

4. Record the sum of 2 and 3.

}

Set x_p and l_p to the condition where the sum calculated above is minimized.

}

The time complexity of (Sub 1.1) is $O(|S| \times |Z| \times \max_k |W_k| \times \max_w |P_w|)$.

Subproblem 1.2 (related to decision variable F_i)

$$Z_{Sub1.2}(\mu_1, \mu_2) = \min \sum_{k \in Z} \sum_{w \in W_k} \sum_{p \in P_w} \mu_{kwp}^1 \left[- \sum_{i \in N_p} \log F_i \right] - \sum_{i \in N} \mu_i^2 [F_i] \quad (\text{Sub 1.2})$$

Subject to:

$$F_i \in \omega_i \quad \forall i \in N. \quad (\text{Sub 1.2.1})$$

(Sub 1.2) can be further decomposed into $|N|$ independent sub problems, for which we must decide the F_i value of each node $i \in N$. Due to $F_i \in \omega_i$, we can solve (Sub 1.2) by the exhaustive search. After the best value of F_i for each node $i \in N$ is selected, the optimal solution of (Sub 1.2) can be found.

The time complexity of (Sub 1.2) is $O(|N| \times \max_i |\omega_i|)$.

Subproblem 1.3 (related to decision variable y_i)

$$Z_{Sub1.3}(\mu_2) = \min \sum_{i \in N} \mu_i^2 [y_i] + \mu^5 [\sum_{i \in N} C_i \cdot (1 - y_i)] \quad (\text{Sub 1.3})$$

Subject to:

$$y_i = 0 \text{ or } 1 \quad \forall i \in N \quad (\text{Sub 1.3.1})$$

$$FC_i \cdot (1 - y_i) \leq R_i \quad \forall i \in N. \quad (\text{Sub 1.3.2})$$

(Sub 1.3) can be further decomposed into $|N|$ independent sub problems. To solve (Sub 1.3) optimally, we first set y_i to 1 if the router capacity constraint, (Sub 1.3.2), is violated. Otherwise, we set y_i by comparing the result of setting y_i to either 0 or 1. y_i is finally set to the condition where the result is smaller. After the best value of y_i for each node $i \in N$ is set, the optimal solution of (Sub 1.3) can be found.

The time complexity of (Sub 1.3) is $O(|N|)$.

3.1.3 The Dual Problem and the Subgradient Method

By solving the above subproblems optimally, the Lagrangean relaxation problem (LR1) can be optimally solved also. According to the weak duality theorem [18], for any set of the multipliers $(\mu_1, \mu_2, \mu_3, \mu_4, \mu_5)$, $Z_{D1}(\mu_1, \mu_2, \mu_3, \mu_4, \mu_5)$ generates a Lower Bound (LB) of Z_{IP2} . Below, we construct a dual problem (D1) to obtain the tightest LB and solve it by the subgradient method [13] [19].

Dual problem (D1)

$$Z_D = \max Z_D(\mu_1, \mu_2, \mu_3, \mu_4, \mu_5) \quad (\text{D1})$$

Subject to: $\mu_1, \mu_2, \mu_3, \mu_4, \mu_5 \geq 0.$

Let a vector m be a subgradient of $Z_{D1}(\mu_1, \mu_2, \mu_3, \mu_4, \mu_5)$. Then, in iteration ψ of the subgradient procedure, the multiplier vector $\mu^\psi = (\mu_1^\psi, \mu_2^\psi, \mu_3^\psi, \mu_4^\psi, \mu_5^\psi)$ is updated by

$$\mu^{\psi+1} = \mu^\psi + t^\psi m^\psi,$$

where

$$m^\psi(\mu_1^\psi, \mu_2^\psi, \mu_3^\psi, \mu_4^\psi) = \{\log(l_p) - \sum_{i \in N_p} \log F, y_i - F, \sum_{k \in Z} \sum_{w \in W_k} \sum_{p \in P_w} x_p \delta_{pv} (\gamma_w + \beta_w) l_p - \hat{g}_v, \phi_k \sum_{w \in W_k} \gamma_w - \sum_{w \in W_k} \sum_{p \in P_w} x_p \gamma_w l_p, \sum_{i \in N} C_i \cdot (1 - y_i) - B\};$$

and the step size, t^ψ , is determined by

$$t^\psi = \lambda \frac{Z_{IP2}^* - Z_D(\mu^\psi)}{\|m^\psi\|^2}.$$

Z_{IP2}^* is the tightest upper bound (UB) of the primal objective function value found by iteration ψ . 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.

3.1.4 Getting Primal Feasible Solutions

By solving the subproblems optimally, we are able to obtain a primal feasible solution from the hint of the associated Lagrangean multipliers. The algorithm to obtain the primal feasible solution is detailed below.

Initially, the solution of the Subproblem 1.1 is used as the selected routing path since it's related to the routing assignment. Further, as the Subproblem 1.3 is related to the filter allocation, we use it as our filter allocation guideline. We hope to reach a solution by adjusting the filtering remaining rate on filter-allocated nodes. Next, the solution of the Subproblem 1.2 is used as our initial filtering remaining rate value, which requires to be, further, adjusted in order to satisfy the threshold of the aggregate traffic and the threshold of the remaining good traffic percentage. If one of the both thresholds is violated, the filtering remaining rate should be adjusted. On the contrary, if all constraints are satisfied, it is the eventual solution. The heuristic algorithm is presented as follows.

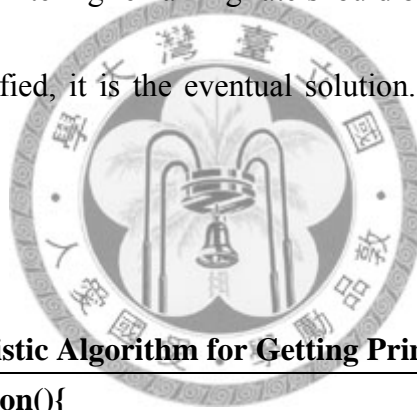


Table 3.1.4-1 Heuristic Algorithm for Getting Primal Feasible Solution

GetPrimalFeasibleSolution(){ Copy Result From Subproblem Solutions; Calculate Traffic Over Victim Server Aggregate Traffic Threshold Calculate Traffic Under Zombie Remaining Good Traffic Percentage Threshold While(AggregateTrafficIsOver GoodTrafficIsUnder){ If(AggregateTrafficOver > GoodTrafficUnder){ //solve the victim server with the most aggregate traffic over SolveMaxAggregateTrafficOver(ServNo) } else{ //solve the zombie with the most good traffic under SolveMaxGoodTrafficUnder(ZombieNo) } }
--

```

}

}

// NODE PASSED FREQUENCY :
// the time of a node passed for the routing assignment

SolveMaxAggregateTrafficOver( ServNo ){
    Calculate NODE PASSED FREQUENCY// From All Zombie To ServNo
    Sort Each Node By NODE PASSED FREQUENCY
    Set A Lower Filter Remaining Rate From The Most Passed Node
}

SolveMaxGoodTrafficUnder( ZombieNo ){
    Calculate NODE PASSED FREQUENCY// From ZombieNo to All Victim Server
    Sort Each Node By NODE PASSED FREQUENCY
    Set A Higher Filter Remaining Rate From The Most Passed Node
}

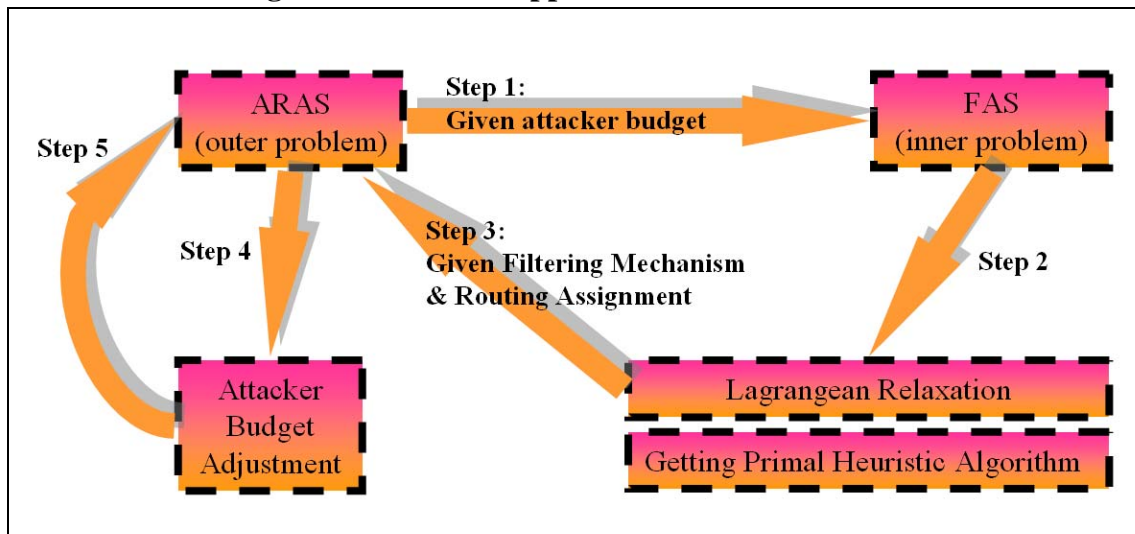
```

3.2 Solution Approach for the ARAS Model

The result from the FAS model means the best defense strategy under a given DDoS attack pattern. As we assumed, the DDoS attacker and defender have complete information. Therefore, both will maximize their benefits by coping with the opponent's strategy. In this point, after the FAS model is solved by the Lagrangean relaxation, the result of the FAS is inputted into the ARAS model. All decision variables about the DDoS attacker now are not given. The solution toward the ARAS model is, by the FAS input, to dynamically adjust attacker's budget allocation strategy until the equilibrium is reached. The interaction between FAS and ARAS is shown in

the following figure.

Figure 3-3 Solution Approach to the ARAS model



In order to solve the ARAS model, the adjustment of the attack budget is based on the concept that, for a path, if there are more nodes with filters, the good traffic tends to be influenced. Thus, we extract an amount of the attack budget on a path with the most filters and reallocate this amount to a path with the least filters. The detailed heuristic algorithm is presented as follows.

Table 3.2-1 Heuristic Algorithm for Solving ARAS Model

```

//Objective: minimize the maximized remaining good traffic
//Initialization
Init_Attack_Budget_Allocation_Strategy();
UB = -LR(); //the return value of LR() is negative due to the objective function
transformation in the FAS model
improvement_counter = 0
improvement_stage_counter = 0;
θ = 0.5; //initial step size coefficient

//Main Heuristic_ARAS procedure
FOR iteration = 1 TO ITERATION_COUNTER_LIMIT {
    Adjustment_Procedure(θ); //as shown in table that follows
    Z*IP 1 = -LR();
  
```

```

//Update UB
IF ( $Z^*_{IP\ 1} < UB$ ) {
     $UB = Z^*_{IP\ 1}$ ;
     $improvement\_counter = 0$ ;
}
ELSE {
     $improvement\_counter ++$ ;
}

//Update step size
IF  $improvement\_counter = IMPROVEMENT\_COUNTER\_LIMIT$  {
     $improvement\_counter = 0$ ;
     $improvement\_stage\_counter ++$ ;
     $\theta = \theta / 2$ ;
}
}

```

Table 3.2-2 Adjustment Procedure Algorithm

Initialization: Take the information of the routing assignment and filter allocation from the FAS problem.

Step 1. Calculate the number of nodes allocated with filters for each zombie to victim servers under the FAS routing assignment.

Step 2. Extract an amount of the attack budget from the path with the most filters allocated.

Step 3. Reallocate the above amount of the attack budget to the path with the least filters allocated.

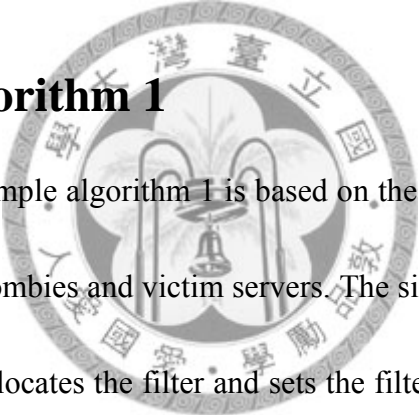
Step 4. Repeat the above steps until the remaining good traffic is minimized.

Chapter 4 Computational Experiments

4.1 Computational Experiments for the FAS Model

In this part, there are two simple algorithms proposed for the comparison that our proposed heuristic is more effective in the FAS model.

4.1.1 Simple Algorithm 1



The concept of the simple algorithm 1 is based on the greedy filtering remaining rate adjustment for both zombies and victim servers. The simple algorithm 1, initially, selects the routing path, allocates the filter and sets the filtering remaining rate. Then, after all of the setting are finished, the adjustment of the filtering remaining rate starts. First, the zombie is considered. If the threshold of the remaining good traffic for any zombie is violated, the filtering remaining rate is loosened. Until the threshold of the remaining good traffic for each zombie is satisfied, the victim server will be considered. If the threshold of the aggregate traffic for any victim server is violated, the filtering remaining rate is tightened. The pseudo code of the simple algorithm 1 is presented in the table that follows.

Table 4.1.1-1 Pseudo Code of Simple Algorithm 1

```
For each OD pair{  
    1. Select a routing path on which there is a node with the lowest filter allocation cost.  
    2. Allocate a filter on this node.  
    3. Check budget and router capacity constraints. If violated, ignore 2 and continue 1 ignoring this node.  
    4. Set the lowest filtering remaining rate for this node.  
}  
For each zombie{  
    While (The threshold of the remaining good traffic percentage is violated){  
        Set a higher filtering remaining rate to a (recorded) path from this zombie.  
        Record the next path //for the next setting  
    }  
}  
For each victim serv.{  
    While (The threshold of the aggregate traffic is violated){  
        Set a lower filtering remaining rate to a (recorded) path from this victim serv.  
        Record the next path //for the next setting  
    }  
}
```

4.1.2 Simple Algorithm 2

The concept of the simple algorithm 2 is quite similar to that of the simple algorithm 1 except the adjustment of the filtering remaining rate is in a global view for each zombie. If the threshold of the remaining good traffic for any zombie is violated, the filtering remaining rate is loosened once for all paths from this zombie. Such an adjustment will help satisfy the requirement more quickly but the victim server may receive more unwanted traffic. The pseudo code of the simple algorithm 2 is presented in the table that follows.

Table 4.1.1-2 Pseudo Code of Simple Algorithm 2

```
For each OD pair{  
    1. Select a routing path on which there is a node with the lowest filter allocation  
    cost.  
    2. Allocate a filter on this node.  
    3. Check budget and router capacity constraints. If violated, ignore 2 and continue 1  
    ignoring this node.  
    4. Set the lowest filtering remaining rate for this node.  
}  
For each zombie{  
    While (The threshold of the remaining good traffic percentage is violated)  
        Set a higher filtering remaining rate to all paths from this zombie.  
}  
For each victim serv.{  
    While (The threshold of the aggregate traffic is violated){  
        Set a lower filtering remaining rate to a (recorded) path from this victim serv.  
        Record the next path //for the next setting  
    }  
}
```

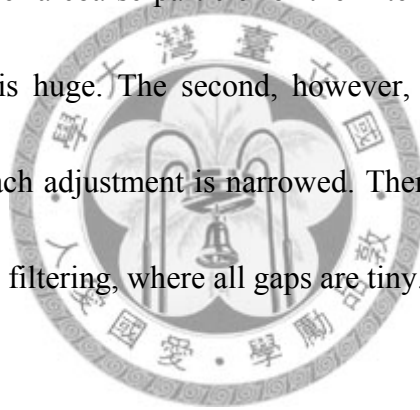
4.1.3 Experiment Environment

The proposed algorithms for solving the FAS model are all coded in C++ with Microsoft Visual Studio 2005 and executed on a computer with Intel® Core™ 2 CPU 1.86GHz, 1.00GB RAM. The Iteration Counter Limit and Improve Counter Limit are set to 2000 and 60 respectively. The step size scalar, λ , is initialized as 2 and is halved if the objective function value, Z_D , is not improved after times of Improve Counter Limit.

To examine the scalability our proposed heuristic, three kinds of network

topologies are considered, in which two are regular networks and one is irregular network. The first regular network is grid network topology shown in Figure 4.1.3(a). The second regular network is mesh network topology shown in Figure 4.1.3(b). The third network is random network topology shown in Figure 4.1.3(c).

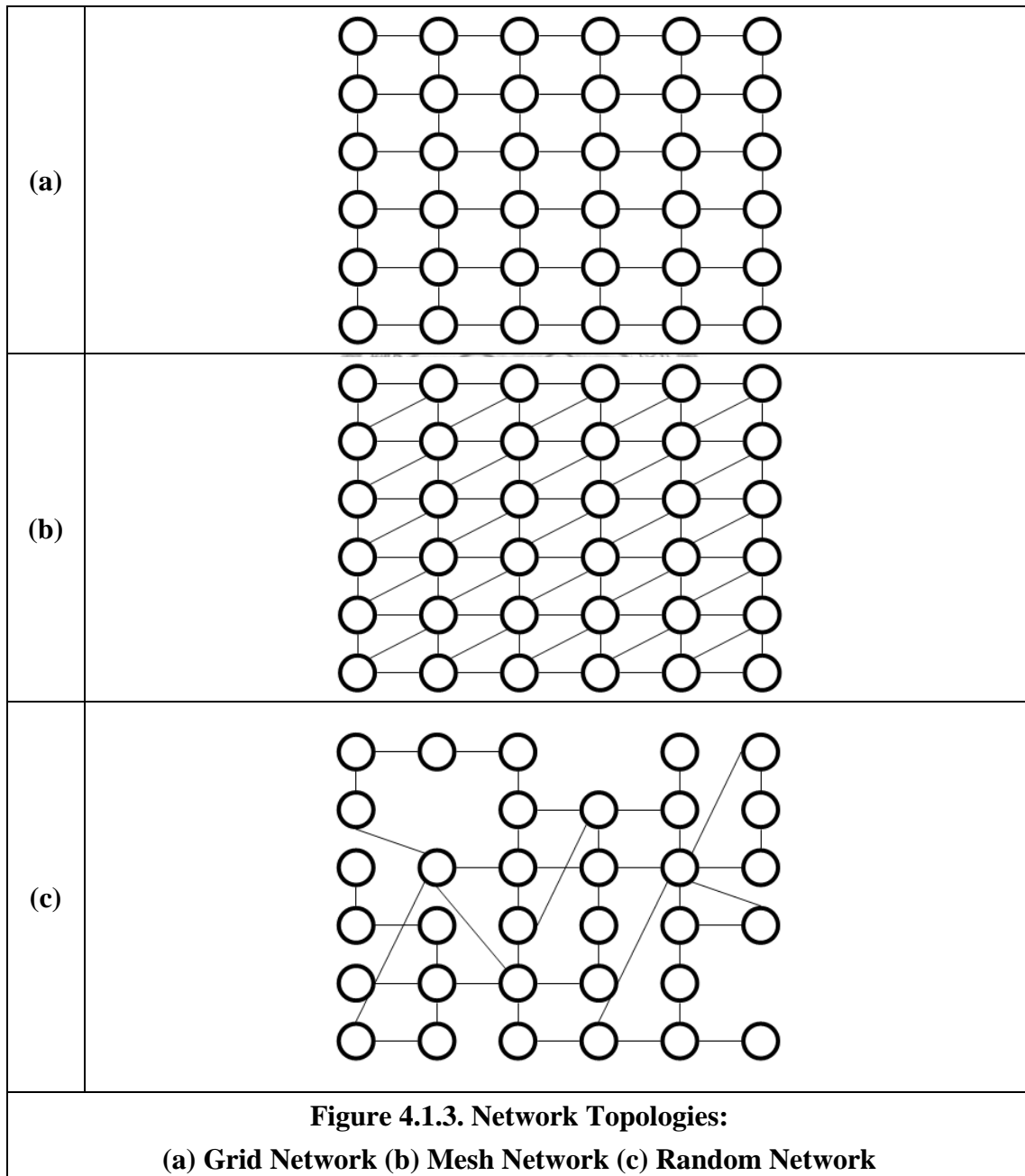
Moreover, in order to more objectively observe the remaining good traffic after filtering, we design three kinds of filtering level adjustment strategy. The first is FL3, where we give the defender a coarse particle for the filtering. The gap between each filtering rate adjustment is huge. The second, however, is a more moderate, FL5, where the gap between each adjustment is narrowed. Then, the third, FL10, is a tiny particle adjustment for the filtering, where all gaps are tiny.



As for the routing policy, we also design three kinds of routing assignment to estimate the remaining good traffic. The first is RP3 where only three routing paths exist for each OD pair selection. Only one will be chosen for an OD pair. Then, the second is RP6 where the routing paths are extended to six paths. Finally, the third is RP9 which gives nine paths.

Since our work is related to the discussion of multiple victim servers under the

DDoS attack, we further design the scenario, network topologies with different number of victim servers. From the typical one victim server, V1, and a more robust two victim server, V2, to the most robust, three victim servers, V3. We hope by examining different number of victim servers, we can find a guideline to maintain more legitimate traffic under the limited budget.



The parameters and scenarios adopted in our experiment are listed in the table below. The Lagrangean relaxation related parameters are also listed.

Table 4.1.3-1 Experiment Parameter Settings for LR in the FAS model

Parameters of Lagrangean relaxation in the FAS model	
Parameters	Value
Iteration Counter Limit	2000
Improve Counter Limit	60
Initial UB	0
Initial Multiplier Value	All multipliers are initiated to be 0
Initial Scalar of Step Size λ	2
Test Platform	CPU: Intel® Core™ 2@1.86GHz RAM: 1GB OS: Windows XP with SP2

Table 4.1.3-2 Experiment Parameter Settings for the FAS model

Parameters of the FAS model	
Parameters	Value
Test Topology	Grid networks Mesh networks Random networks
Number of Nodes N	25, 49, 100
Simple Algorithm	Simple Algorithm 1 Simple Algorithm 2
Victim Server	V1 V2 V3
Filter Level	FL3 FL5 FL10
Routing Policy	RP3 RP6 RP9
Total Attacker's Budget	200~800

4.1.4 Experiment Results

To evaluate the remaining good traffic under different scenarios, we show the experiment results in the following tables. In each table, the LR value means the remaining good traffic calculated by the optimal feasible solution from the Lagrangean relaxation; The UB value means the upper bound of LR; and SA1 and SA2 are the remaining good traffic from the simple algorithm 1 and the simple algorithm 2. We further calculate the gap between LR and UB by $\frac{UB - LR}{LR} \times 100\%$ to examine the quality of LR. The improvement ration of LR to SA1 and SA2 is calculated by $\frac{LR - SA1}{SA1} \times 100\%$ and $\frac{LR - SA2}{SA2} \times 100\%$.

Table 4.1.4-1 Experiment Results of Grid Network for the FAS Model (|N|=25)

Test Topology: Grid Network							
Victim Serv.	LR	UB	Gap	SA1	Imp. R. to SA1	SA2	Imp. R. to SA2
V1	6.90	10.10	46.36	6.90	0.00	6.27	10.00
V2	11.60	13.11	12.99	11.50	0.94	10.45	11.03
V3	24.48	27.24	11.25	16.88	45.00	15.90	53.96

Table 4.1.4-2 Experiment Results of Mesh Network for the FAS Model (|N|=25)

Test Topology: Mesh Network							
Victim Serv.	LR	UB	Gap	SA1	Imp. R. to SA1	SA2	Imp. R. to SA2
V1	9.18	10.10	9.97	9.18	0.00	9.18	0.00
V2	12.30	13.13	6.79	12.30	0.00	12.30	0.00
V3	24.51	27.24	11.14	19.68	24.51	17.31	41.58

Table 4.1.4-3 Experiment Results of Random Network for the FAS Model ($|N|=25$)

Test Topology: Random Network							
Victim Serv.	LR	UB	Gap	SA1	Imp. R. to SA1	SA2	Imp. R. to SA2
V1	9.18	10.10	9.97	9.18	0.00	9.18	0.00
V2	13.07	13.11	0.35	12.30	6.23	11.50	13.67
V3	23.73	27.24	14.77	21.67	9.52	17.02	39.39

Table 4.1.4-4 Experiment Results of Grid Network for the FAS Model ($|N|=49$)

Test Topology: Grid Network							
Victim Serv.	LR	UB	Gap	SA1	Imp. R. to SA1	SA2	Imp. R. to SA2
V1	9.42	10.10	7.15	7.59	24.18	7.59	24.18
V2	11.40	13.11	15.00	10.55	8.08	9.27	22.94
V3	25.58	27.24	6.47	19.93	28.34	14.78	73.06

Table 4.1.4-5 Experiment Results of Mesh Network for the FAS Model ($|N|=49$)

Test Topology: Mesh Network							
Victim Serv.	LR	UB	Gap	SA1	Imp. R. to SA1	SA2	Imp. R. to SA2
V1	9.18	10.10	9.97	9.18	0.00	9.18	0.00
V2	12.30	13.11	6.59	10.80	13.90	11.18	10.00
V3	25.82	27.24	5.48	22.71	13.68	18.57	39.03

Table 4.1.4-6 Experiment Results of Random Network for the FAS Model ($|N|=49$)

Test Topology: Random Network							
Victim Serv.	LR	UB	Gap	SA1	Imp. R. to SA1	SA2	Imp. R. to SA2
V1	9.18	10.10	9.97	9.18	0.00	8.35	10.00
V2	12.30	13.11	6.60	11.88	3.55	9.27	32.64
V3	25.82	27.24	5.48	23.57	9.56	18.73	37.88

Table 4.1.4-7 Experiment Results of Grid Network for the FAS Model ($|N|=100$)

Test Topology: Grid Network							
Victim Serv.	LR	UB	Gap	SA1	Imp. R. to SA1	SA2	Imp. R. to SA2
V1	8.35	10.10	20.96	7.59	10.00	7.59	10.00
V2	11.40	13.11	15.00	8.95	27.37	8.47	34.62
V3	28.53	29.61	3.79	22.98	24.13	23.45	21.65

Table 4.1.4-8 Experiment Results of Mesh Network for the FAS Model ($|N|=100$)

Test Topology: Mesh Network							
Victim Serv.	LR	UB	Gap	SA1	Imp. R. to SA1	SA2	Imp. R. to SA2
V1	9.18	10.10	9.97	8.35	10.00	9.18	0.00
V2	12.30	13.11	6.59	12.30	0.00	9.27	32.64
V3	25.91	27.24	5.11	17.05	51.97	16.56	56.47

Table 4.1.4-9 Experiment Results of Random Network for the FAS Model ($|N|=100$)

Test Topology: Random Network							
Victim Serv.	LR	UB	Gap	SA1	Imp. R. to SA1	SA2	Imp. R. to SA2
V1	9.18	10.10	9.97	9.18	0.00	8.35	10.00
V2	12.30	13.12	6.65	11.15	10.34	9.27	32.64
V3	25.91	27.24	5.11	17.68	46.53	17.59	47.29

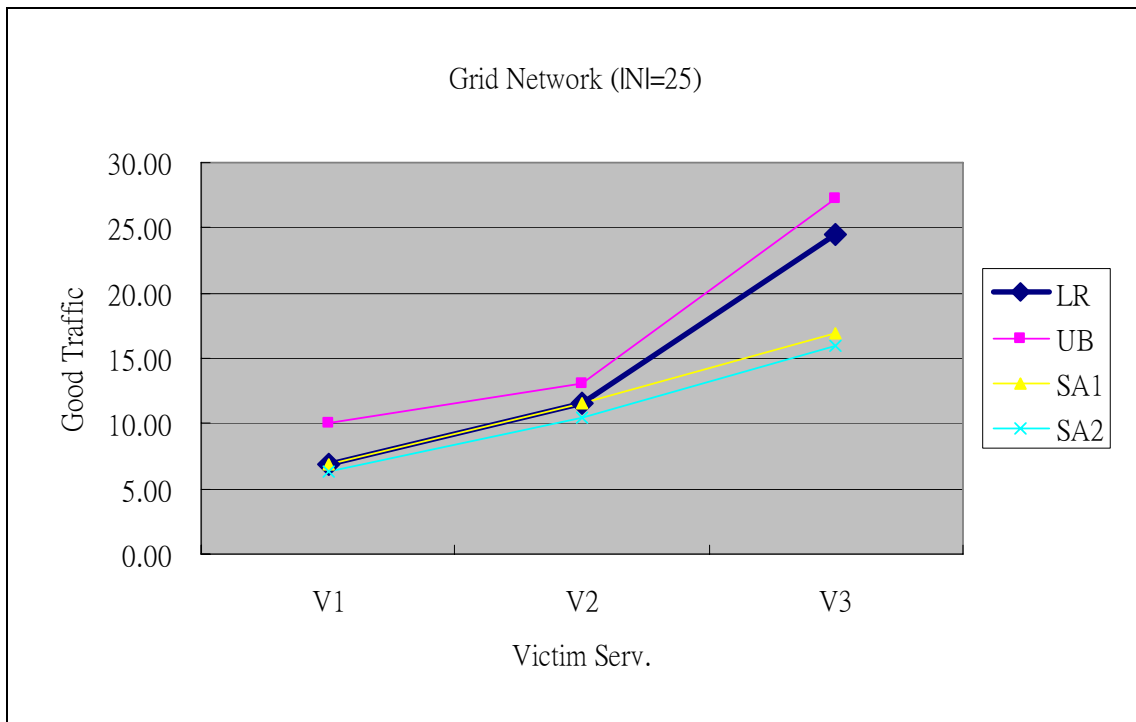


Figure 4.1.4-1
the Remaining Good Traffic of Grid Network with FL10 ($|N|=25$)

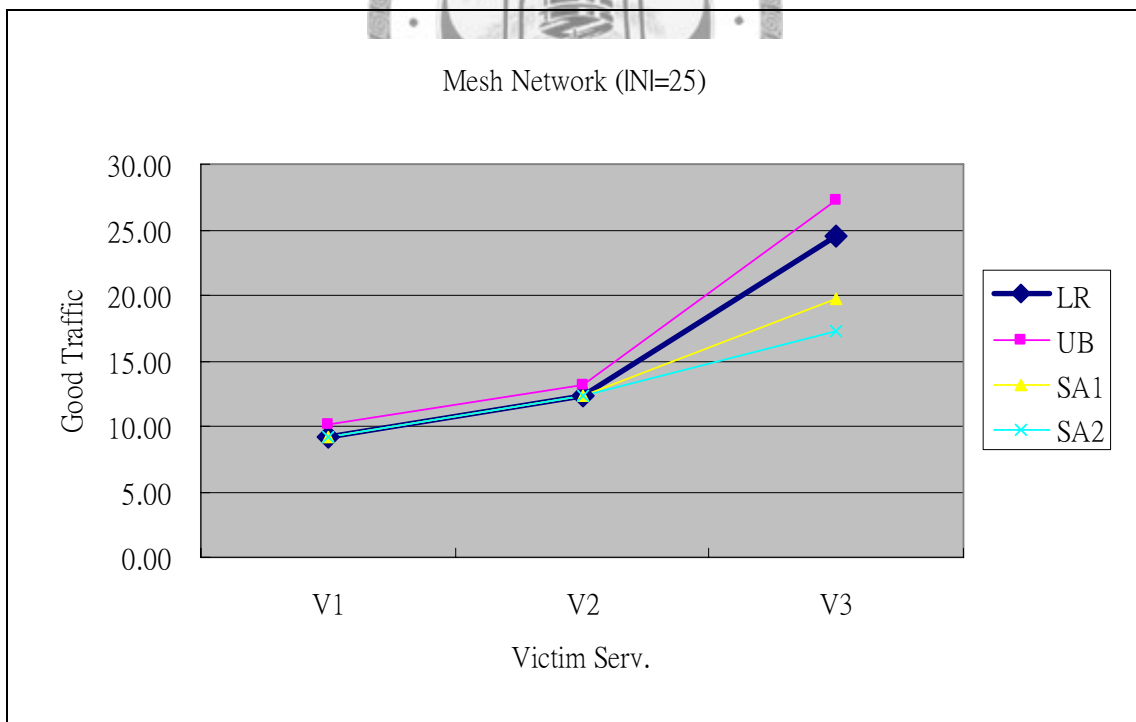


Figure 4.1.4-2
the Remaining Good Traffic of Mesh Network with FL10 ($|N|=25$)

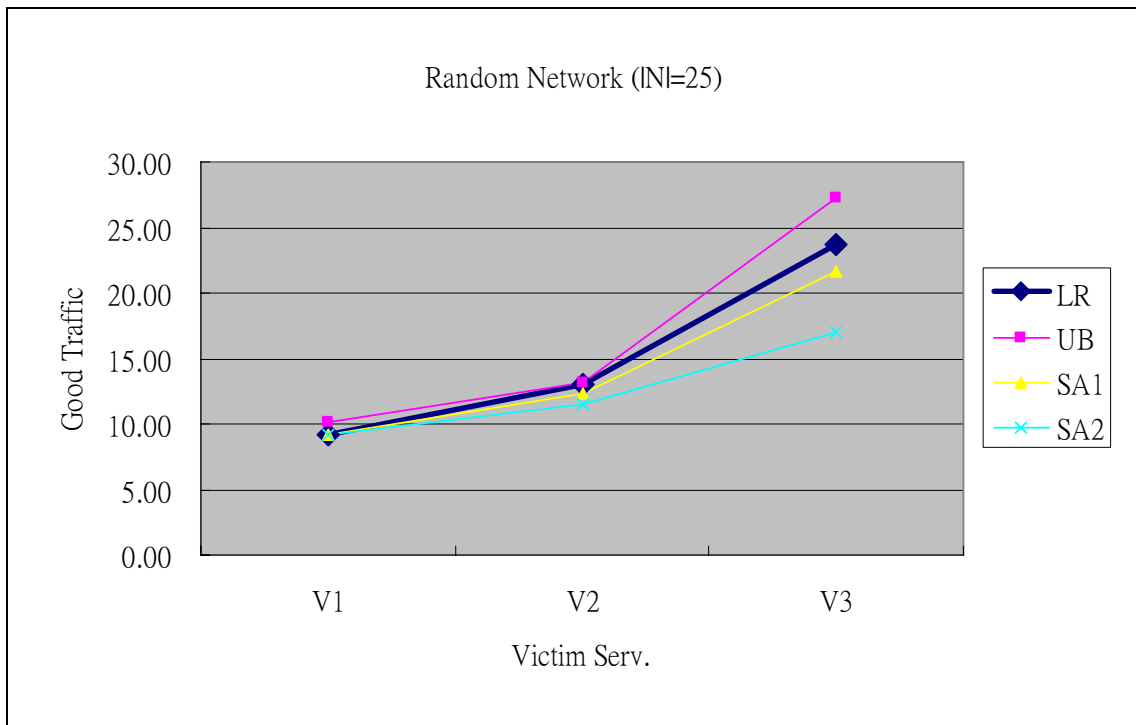


Figure 4.1.4-3
the Remaining Good Traffic of Random Network with FL10 ($|N|=25$)

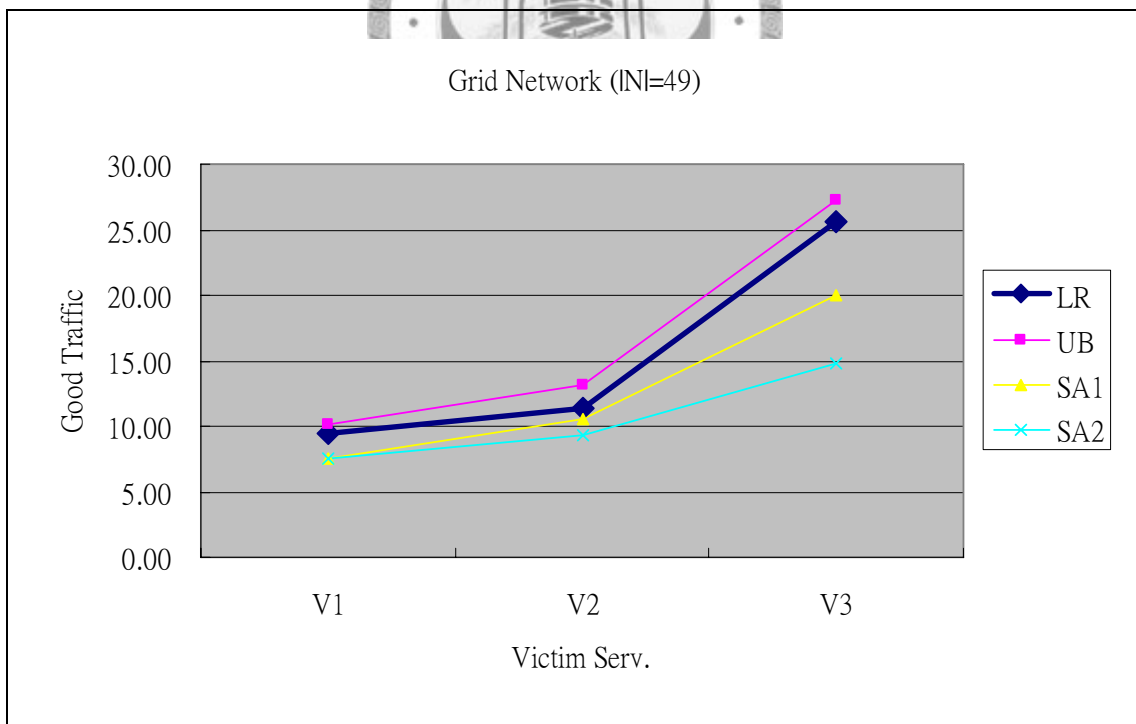


Figure 4.1.4-4
the Remaining Good Traffic of Grid Network with FL10 ($|N|=49$)

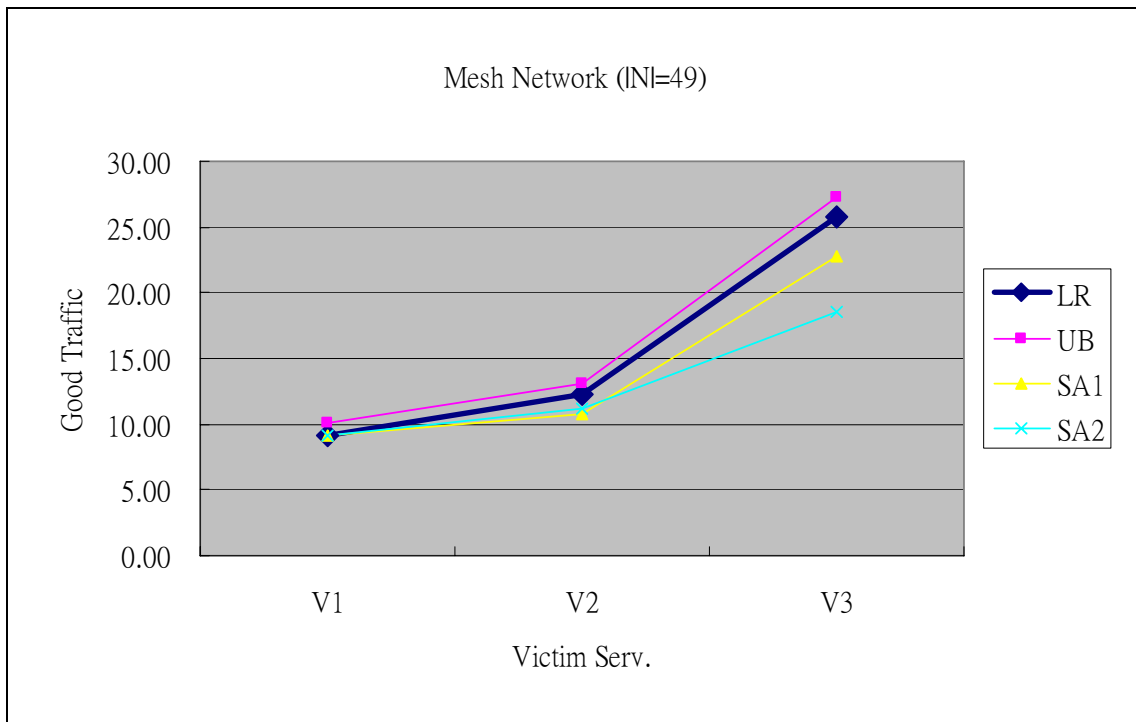


Figure 4.1.4-5
the Remaining Good Traffic of Mesh Network with FL10 ($|N|=49$)

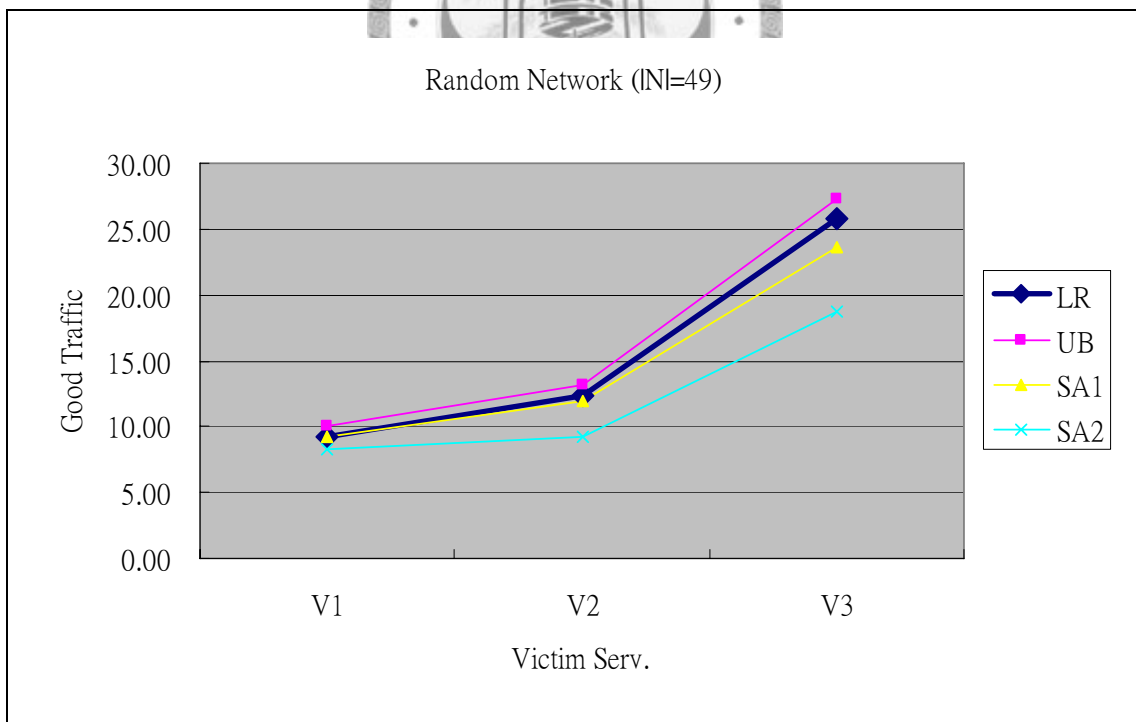


Figure 4.1.4-6
the Remaining Good Traffic of Random Network with FL10 ($|N|=49$)

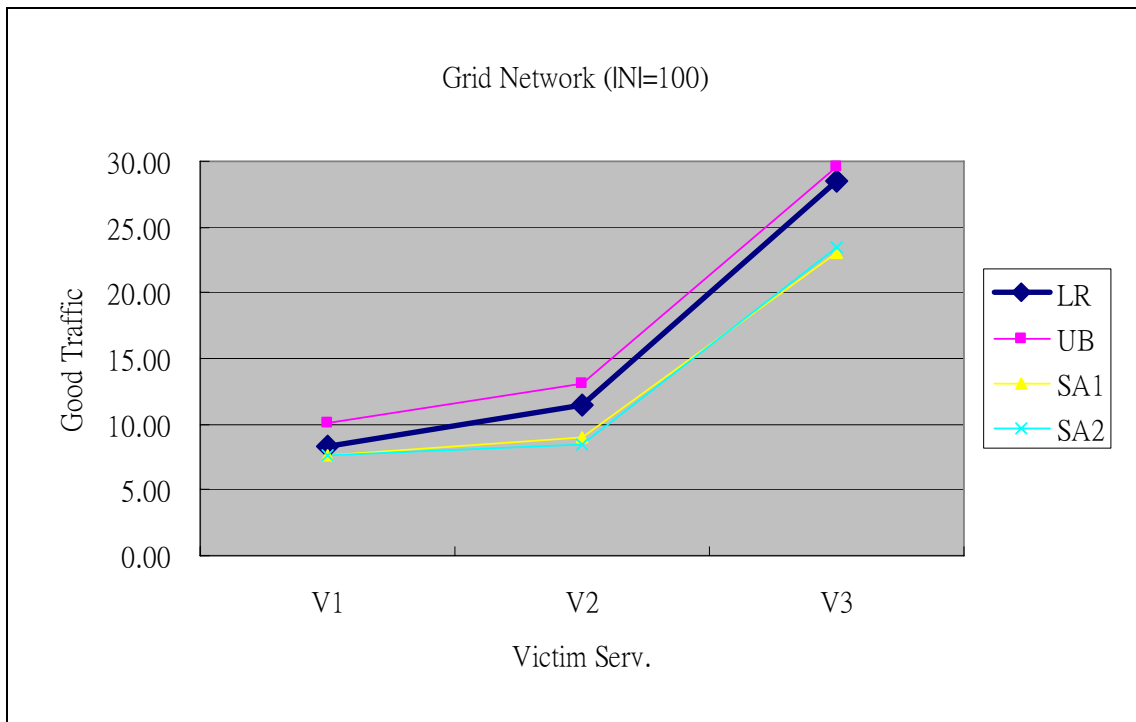


Figure 4.1.4-7
the Remaining Good Traffic of Grid Network with FL10 ($|N|=100$)

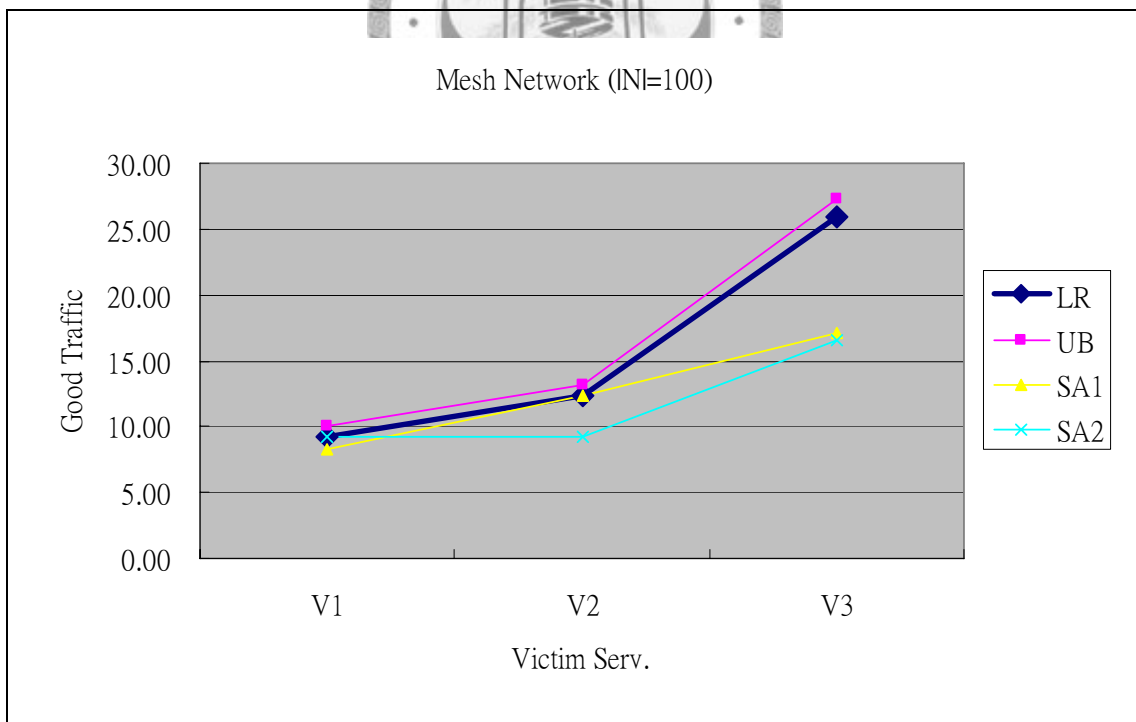


Figure 4.1.4-8
the Remaining Good Traffic of Mesh Network with FL10 ($|N|=100$)

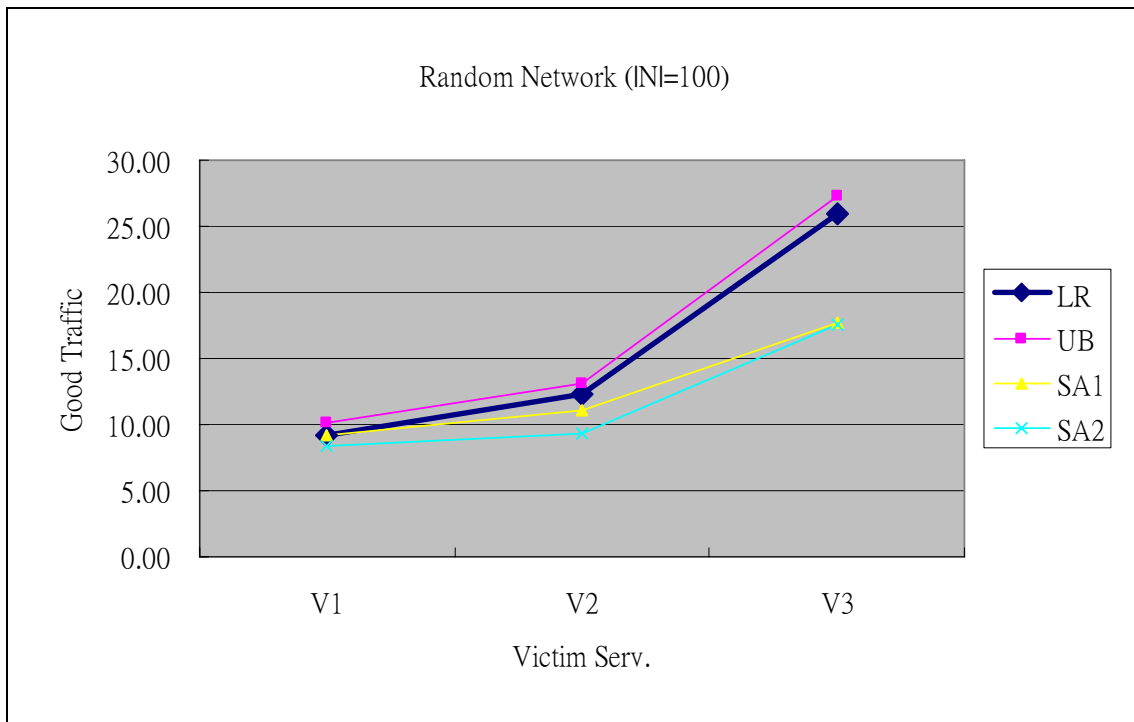


Figure 4.1.4-9
the Remaining Good Traffic of Random Network with FL10 ($|N|=100$)

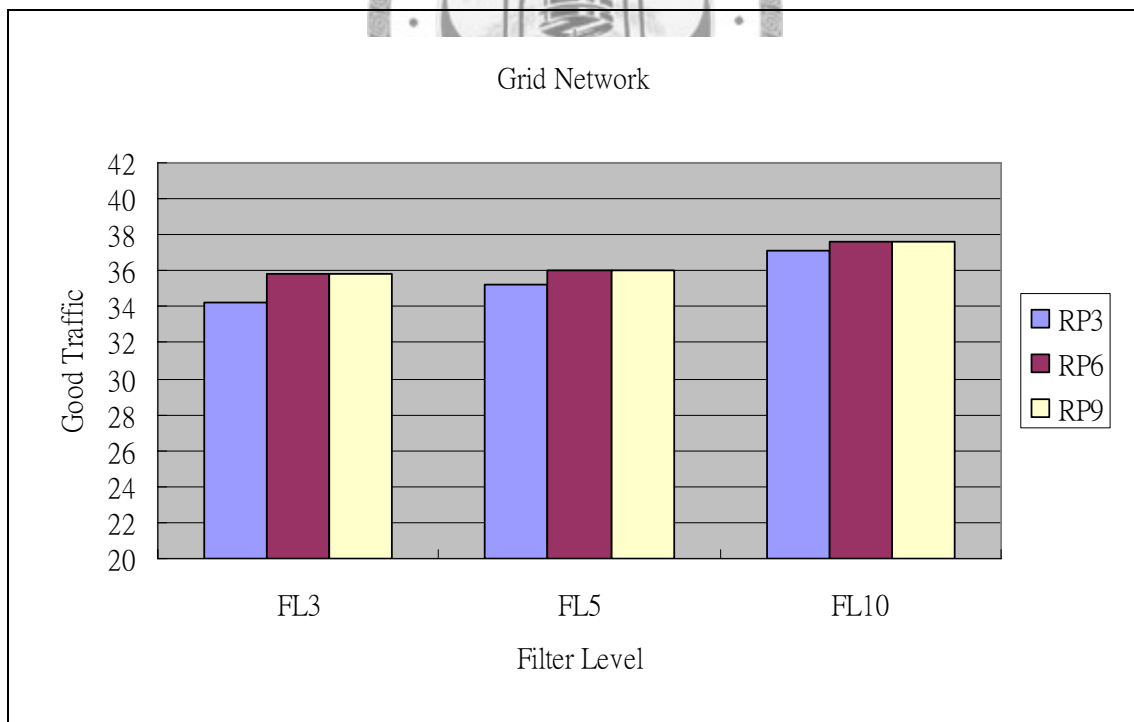


Figure 4.1.4-10
the Remaining Good Traffic of Grid Network with different FL and RP
($|N|=100$)

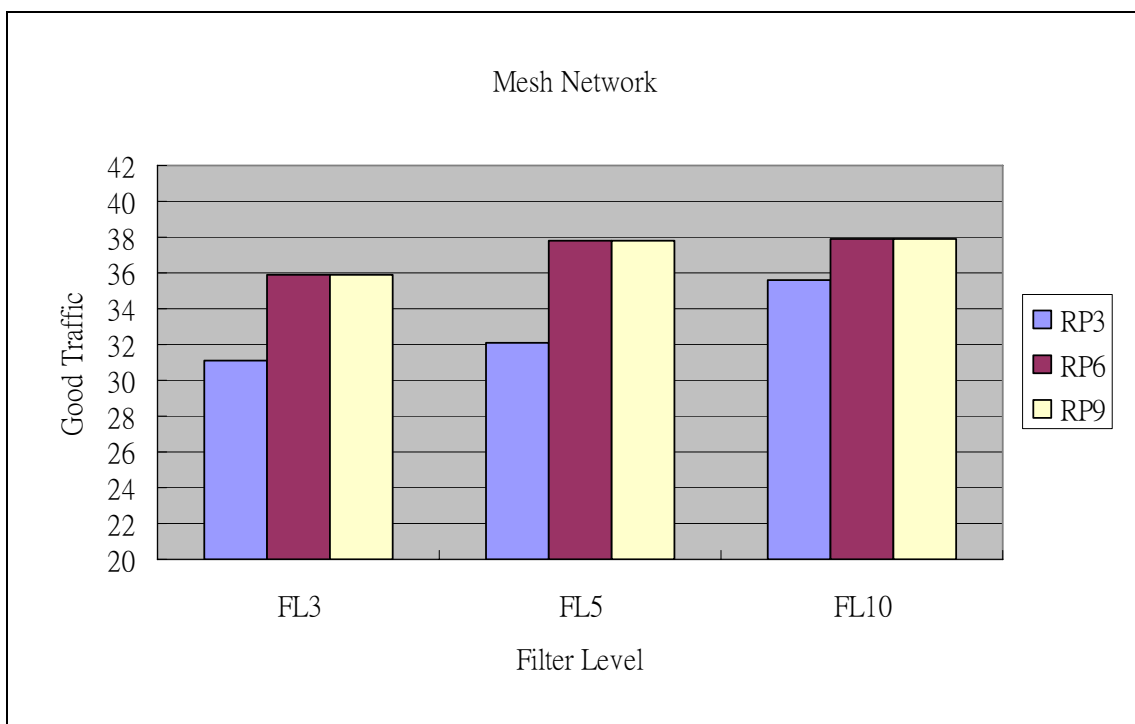


Figure 4.1.4-11
the Remaining Good Traffic of Mesh Network with different FL and RP
(|N|=100)

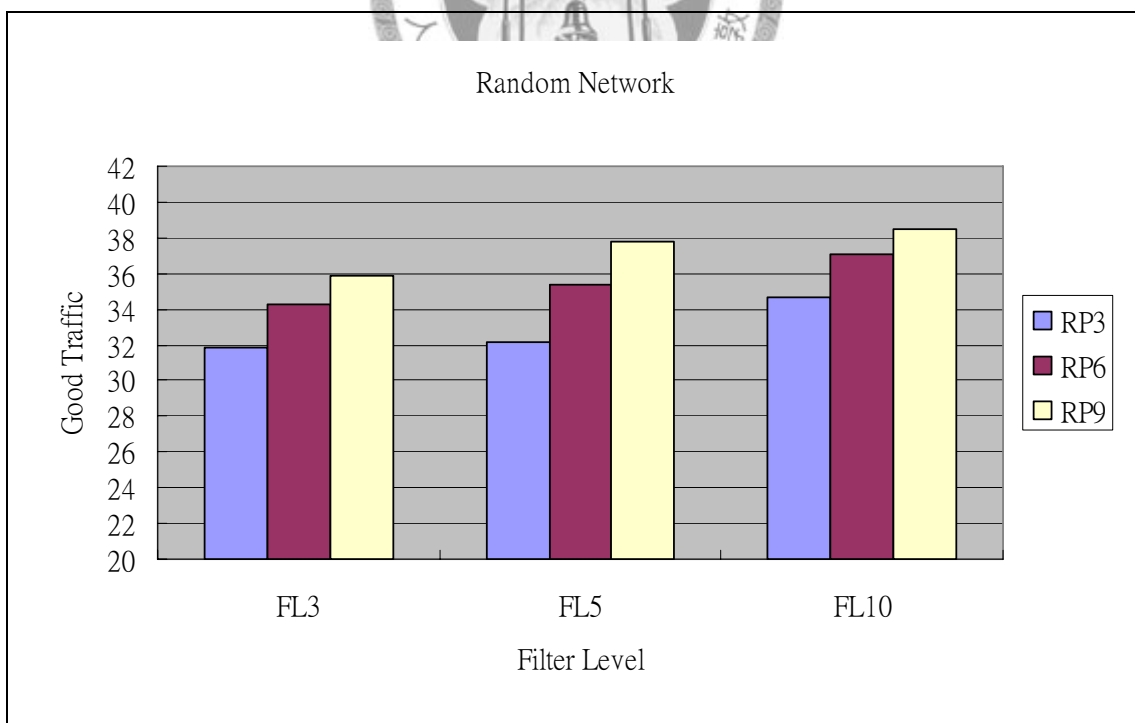


Figure 4.1.4-12
the Remaining Good Traffic of Random Network with different FL and RP
(|N|=100)

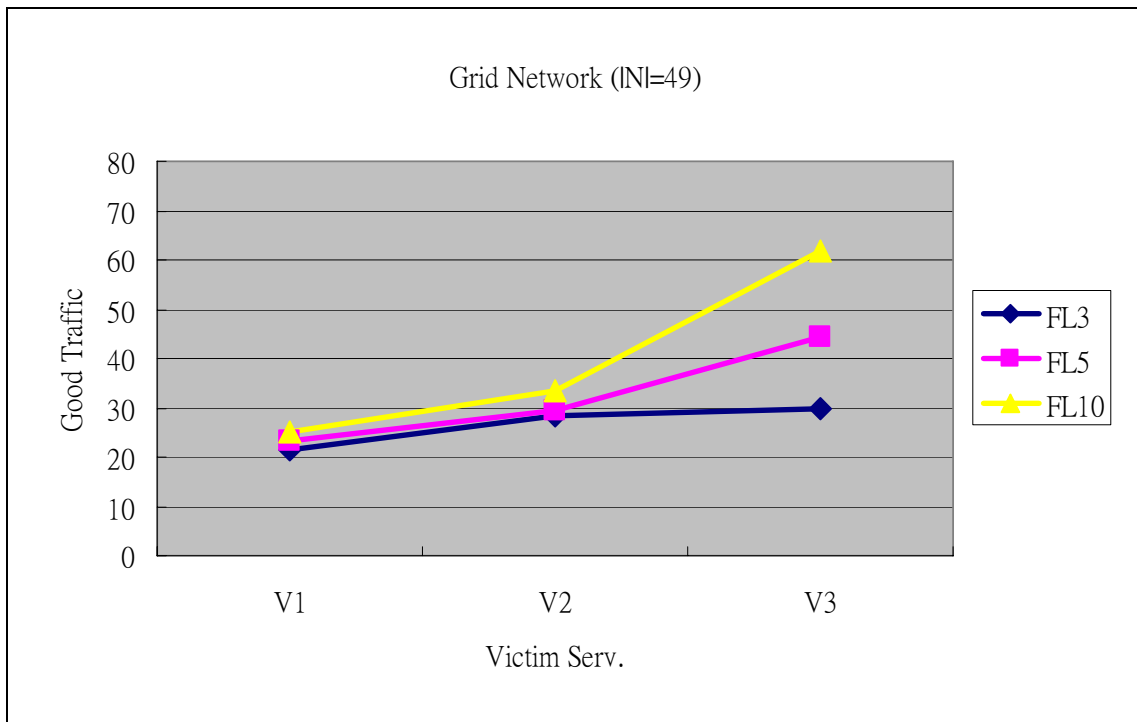


Figure 4.1.4-13
the Remaining Good Traffic of Grid Network with different FL ($|N|=49$)

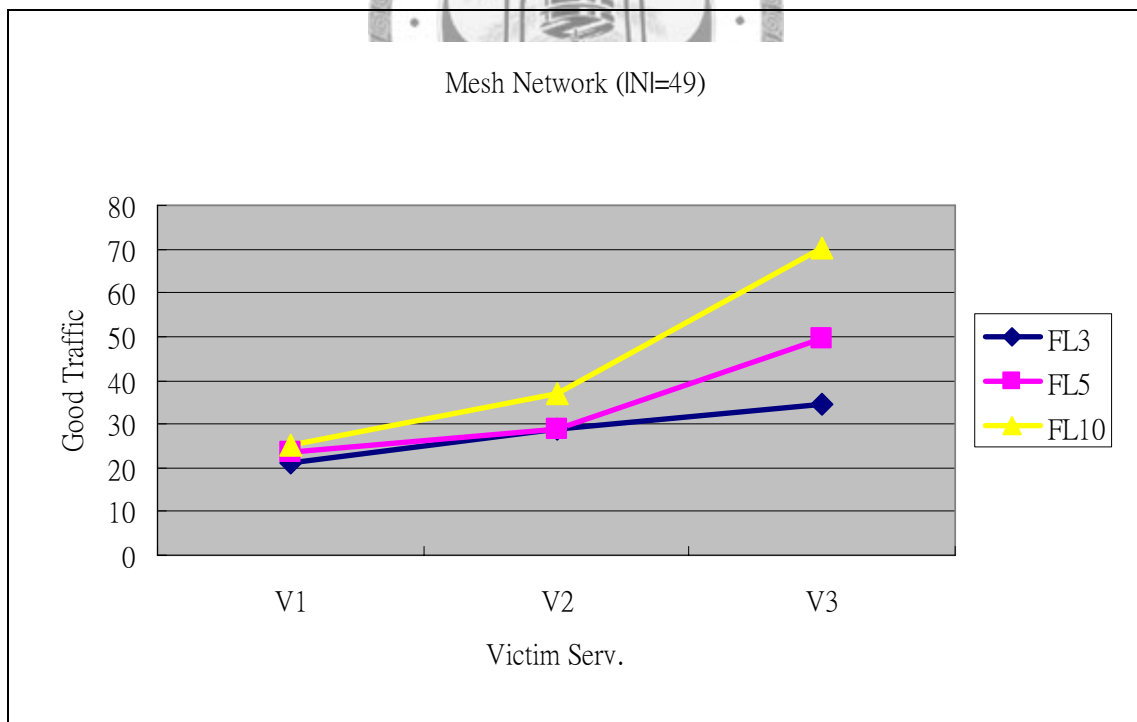
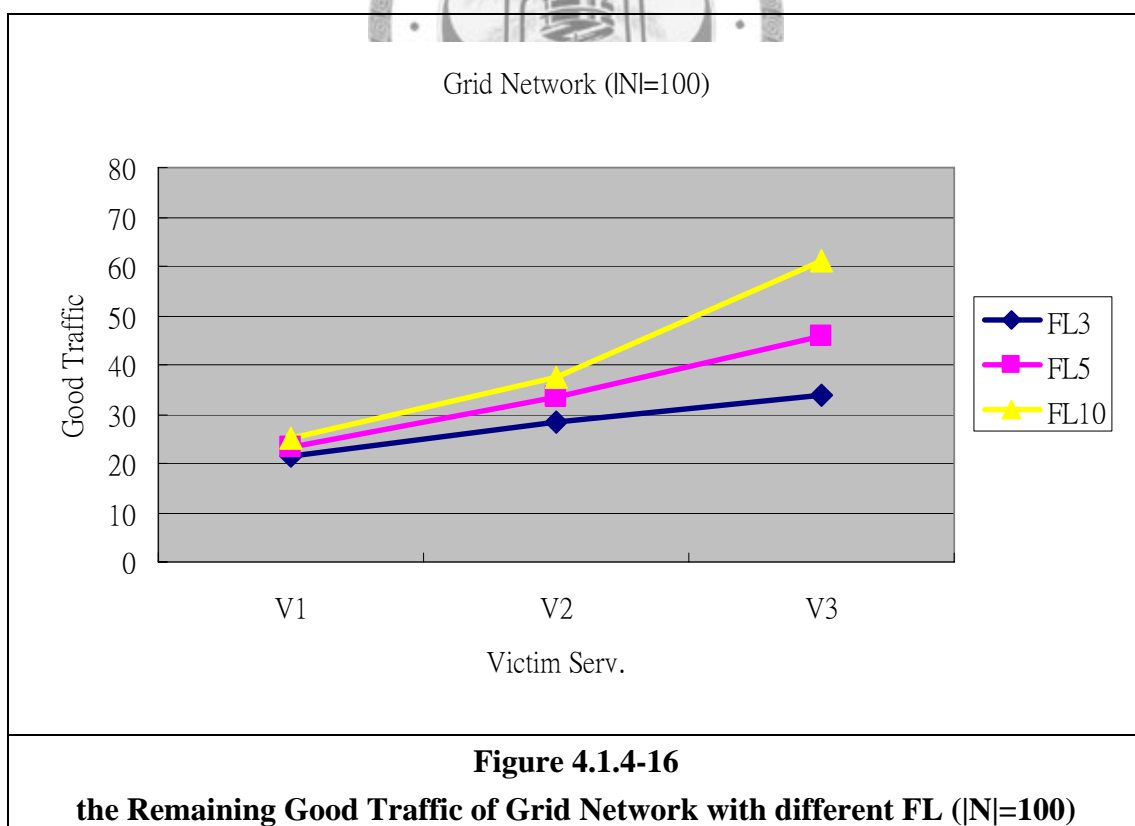
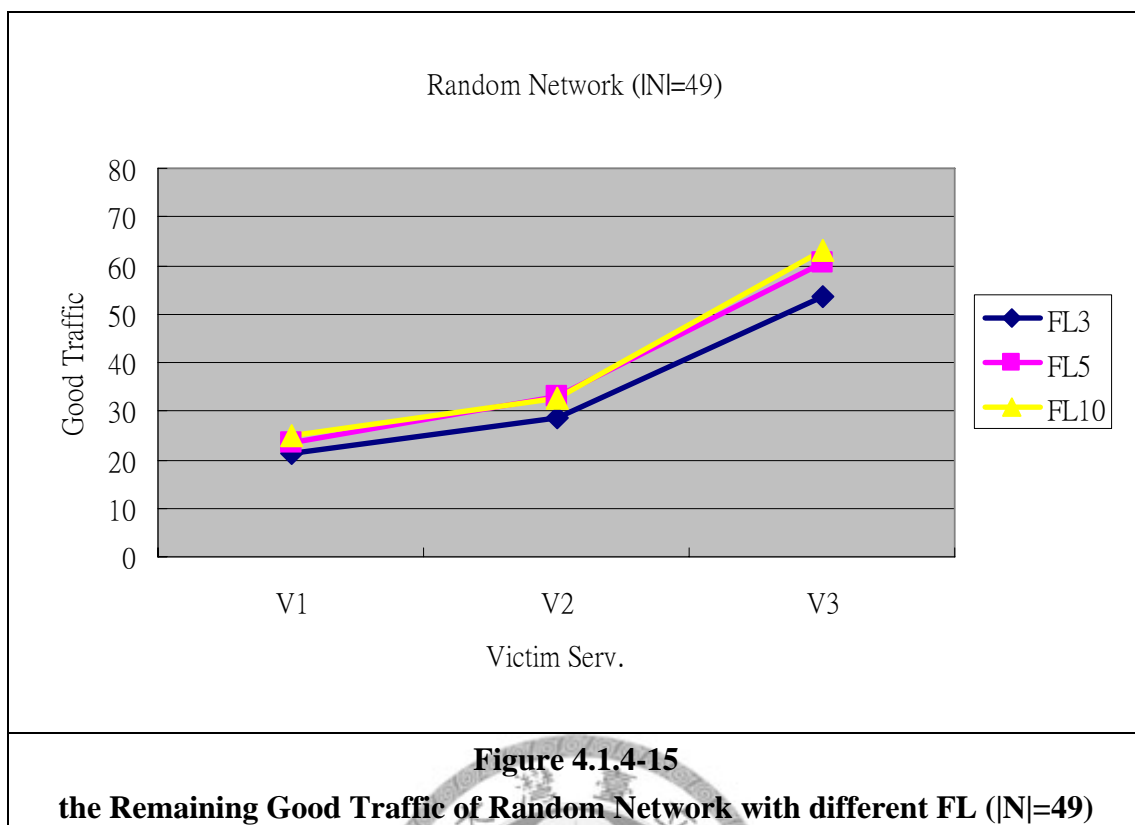


Figure 4.1.4-14
the Remaining Good Traffic of Mesh Network with different FL ($|N|=49$)



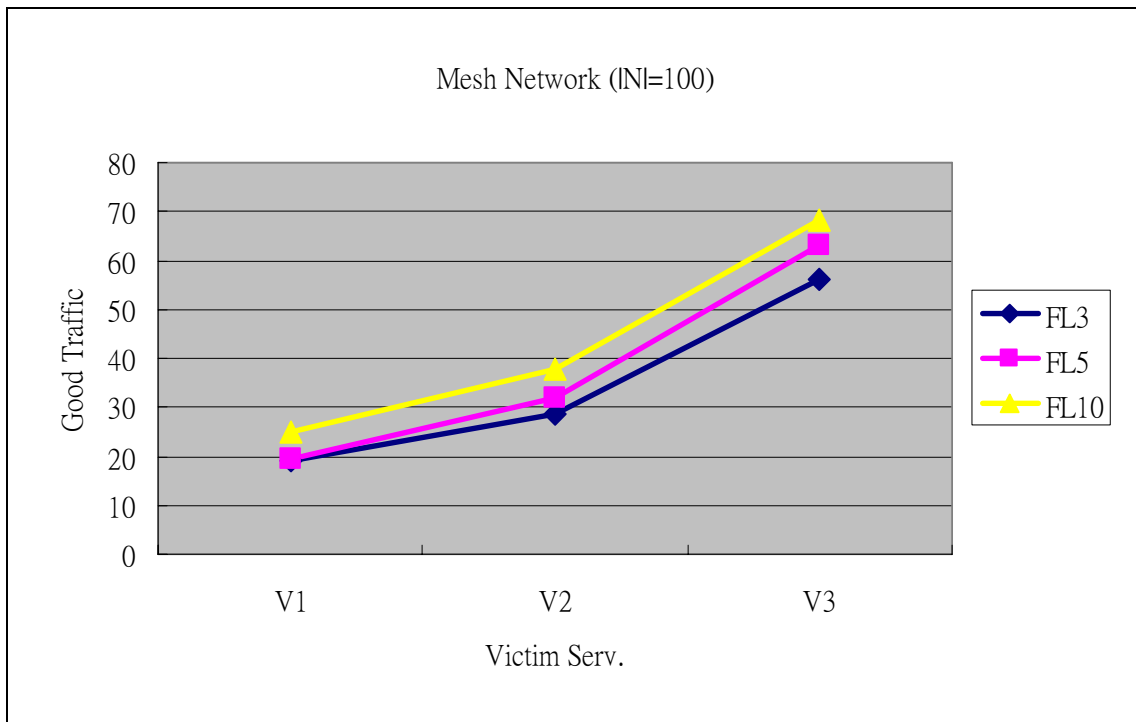


Figure 4.1.4-17
the Remaining Good Traffic of Mesh Network with different FL ($|N|=100$)

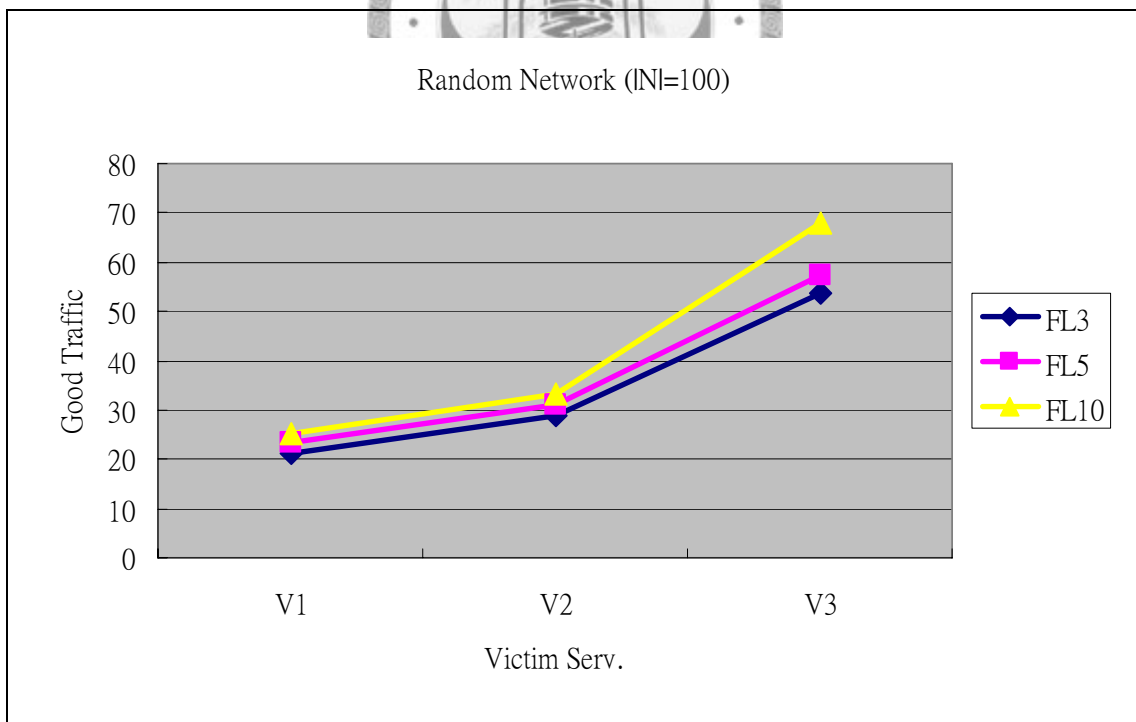


Figure 4.1.4-18
the Remaining Good Traffic of Random Network with different FL ($|N|=100$)

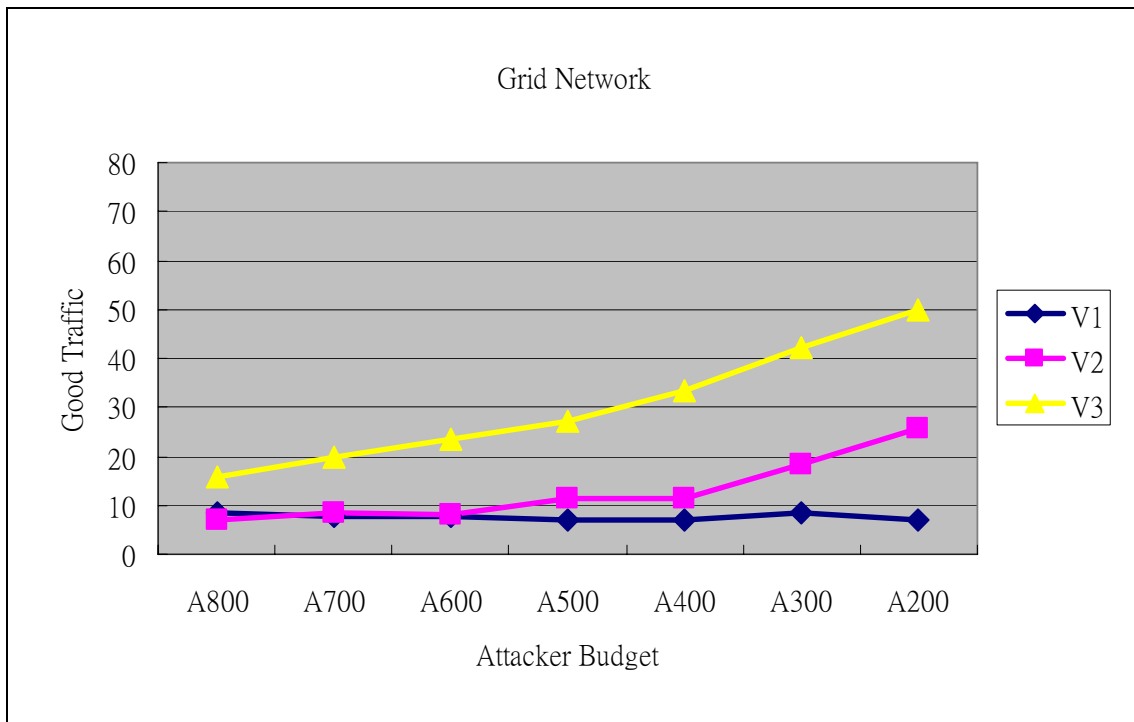


Figure 4.1.4-19
the Remaining Good Traffic of Grid Network with different Attacker Budget
 ($|N|=100$)

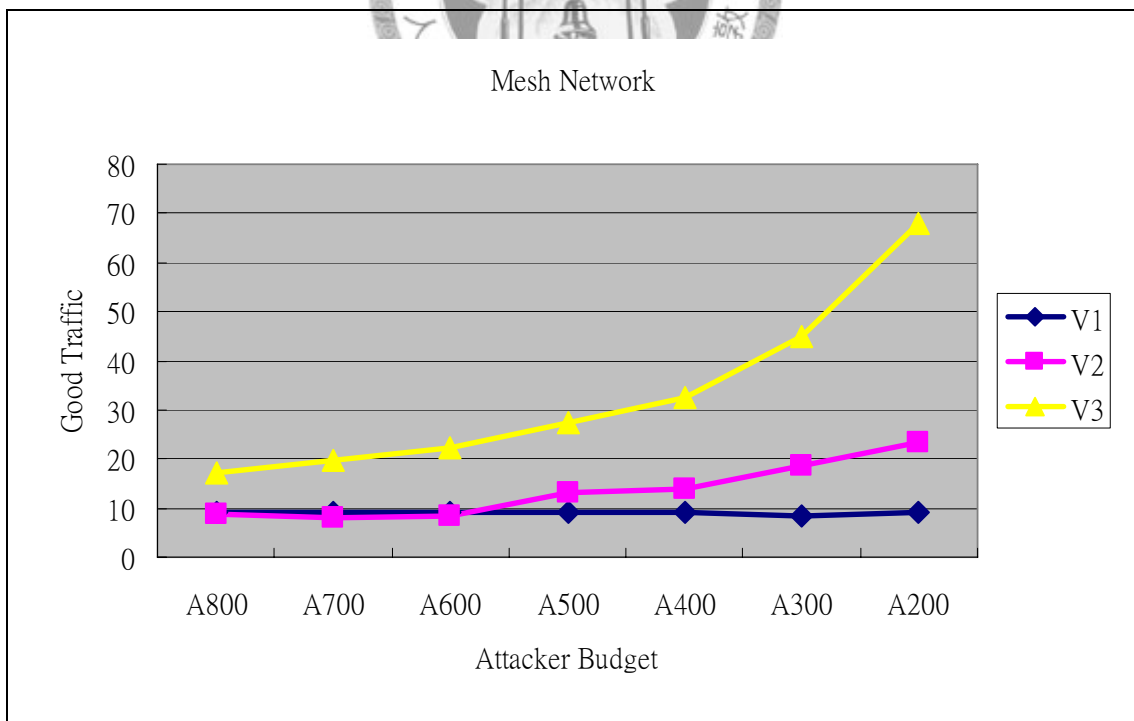
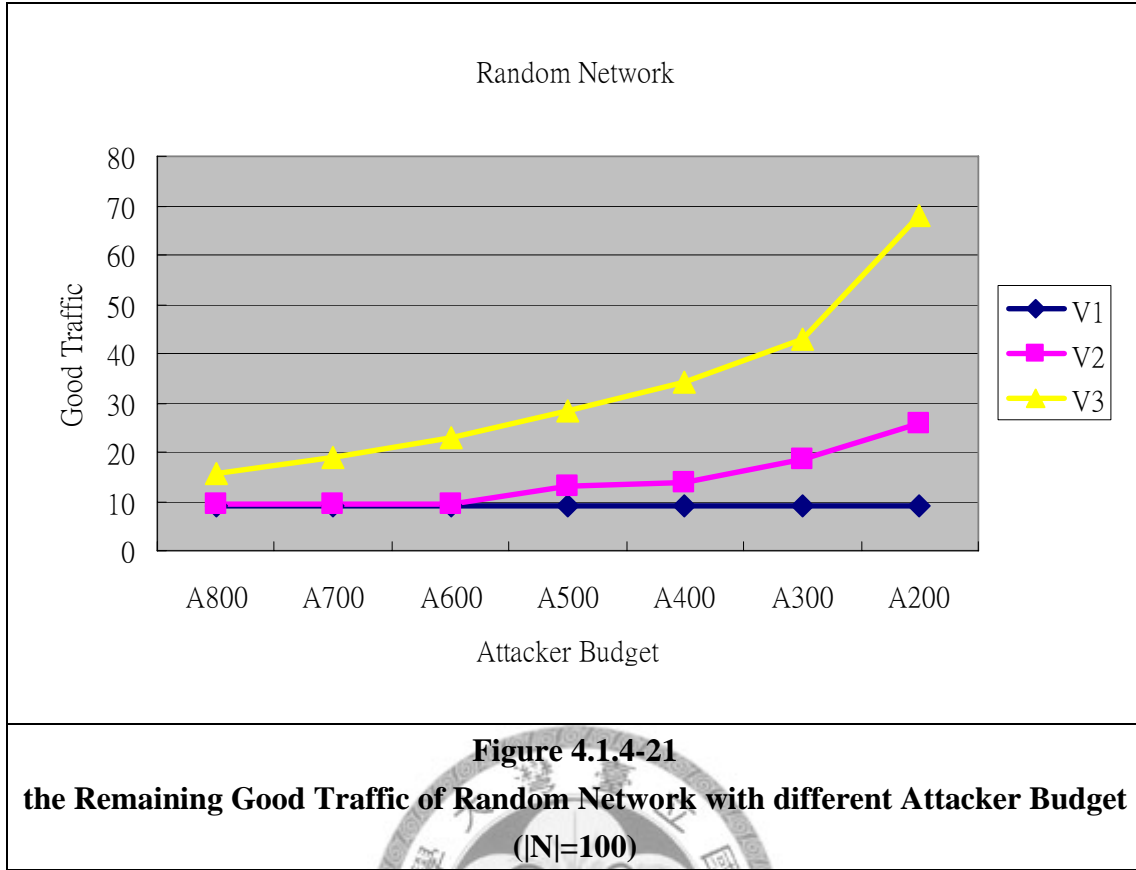


Figure 4.1.4-20
the Remaining Good Traffic of Mesh Network with different Attacker Budget
 ($|N|=100$)



4.1.5 Discussion of Results

Figure 4.1.4-1 to Figure 4.1.4-9 compares the solution quality of our proposed Lagrangean relaxation-based algorithm with the simple algorithm 1 and the simple algorithm 2, and shows the gap between LR and UBs.

- From these figures, we can obviously find that our proposed heuristic outperforms the two simple algorithms in all cases, although sometimes the simple algorithm 1 or the simple algorithm 2 has the same performance compared with our heuristic. Overall, this is enough to indicate that our proposed Lagrangean relaxation-based algorithm is able to solve the FAS model in various network topologies. Moreover, gaps between LR and UBs are small, which

further means that our proposed heuristic is a near-optimal solution approach.

Because our proposed heuristic adopts the hints provided by the LR, it certainly has better and firmer performance than both simple algorithms. We take a look from V2 to V3 in these figures. We can find the solution quality of our proposed heuristic improves quickly. The difference between the simple algorithms and our proposed heuristic could be found obviously here.

Figure 4.1.4-10 to Figure 4.1.4-12 observes the remaining good traffic under different RPs and FLs. In Figure 4.1.4-10 and 11, we can find it is sensitive for maintaining the good traffic from RP3 to RP6 in grid and mesh network. No matter under FL3, FL5 or FL10, the trend all exists. However, from RP6 to RP9, the sensitivity disappears. In Figure 4.1.4-12, for the random network, this sensitivity can not be found.

Figure 4.1.4-13 to Figure 4.1.4-18 observes the remaining good traffic under different FLs from network size ($|N|=49$) to ($|N|=100$). The FL10 performs better than the FL5 and the FL5 performs better than the FL3 in all three different network topologies and different victim servers. Because the FL10 gives more choices for the filtering adjustment, it is likely to maintain more remaining good traffic.

Figure 4.1.4-19 to Figure 4.1.4-21 observes the remaining good traffic under different total attacker's budget at network size ($|N|=100$). In all network topologies, if total attacker's budget increases, the remaining good traffic will decrease. However, on the contrary, if the total attacker's budget decreases, the remaining good traffic will increase. In V1, the trend is not obvious because the total attacker's budget 200 is enough to overwhelm one victim server.

4.2 Computational Experiments for the ARAS Model

4.2.1 Experiment Environment

The proposed algorithms for solving the ARAS model are all coded in C++ with Microsoft Visual Studio 2005 and executed on a computer with Intel® Core™ 2 CPU 1.86GHz, 1.00GB RAM. The Iteration Counter Limit and Improve Counter Limit are set to 100 and 10 respectively. The step size scalar, θ , is initialized as 0.5 and is halved if the objective function value, Z_{IP1} , is not improved after times of Improve Counter Limit.

In the ARAS model, the attacker tries to minimize the remaining good traffic under the defender's filtering mechanism and routing assignment. After each round of the attack, the defender adjusts the filtering mechanism and routing assignment

according to the attacker's budget reallocation. Finally, the equilibrium is reached.

Here, three kinds of filtering level adjustment and network topologies are tested including different numbers of victim servers. The parameters and scenarios adopted in our experiment are listed in the table below.

Table 4.2.1-1 Experiment Parameter Settings for the Adjustment Procedure in the ARAS model

Parameters of Adjustment Procedure	
Parameters	Value
Iteration Counter Limit	100
Improve Counter Limit	10
Initial Scalar of Step Size θ	0.5
Test Platform	CPU: Intel® Core™ 2@1.86GHz RAM: 1GB OS: Windows XP with SP2

Table 4.2.1-2 Experiment Parameter Settings for the ARAS model

Parameter of the ARAS model	
Parameters	Value
Test Topology	Grid networks
	Mesh networks
	Random networks
Number of Nodes N	25, 49
Victim Server	V1
	V2
	V3
Filter Level	FL3
	FL5
	FL10
Routing Policy	RP3

4.2.2 Experiment Results

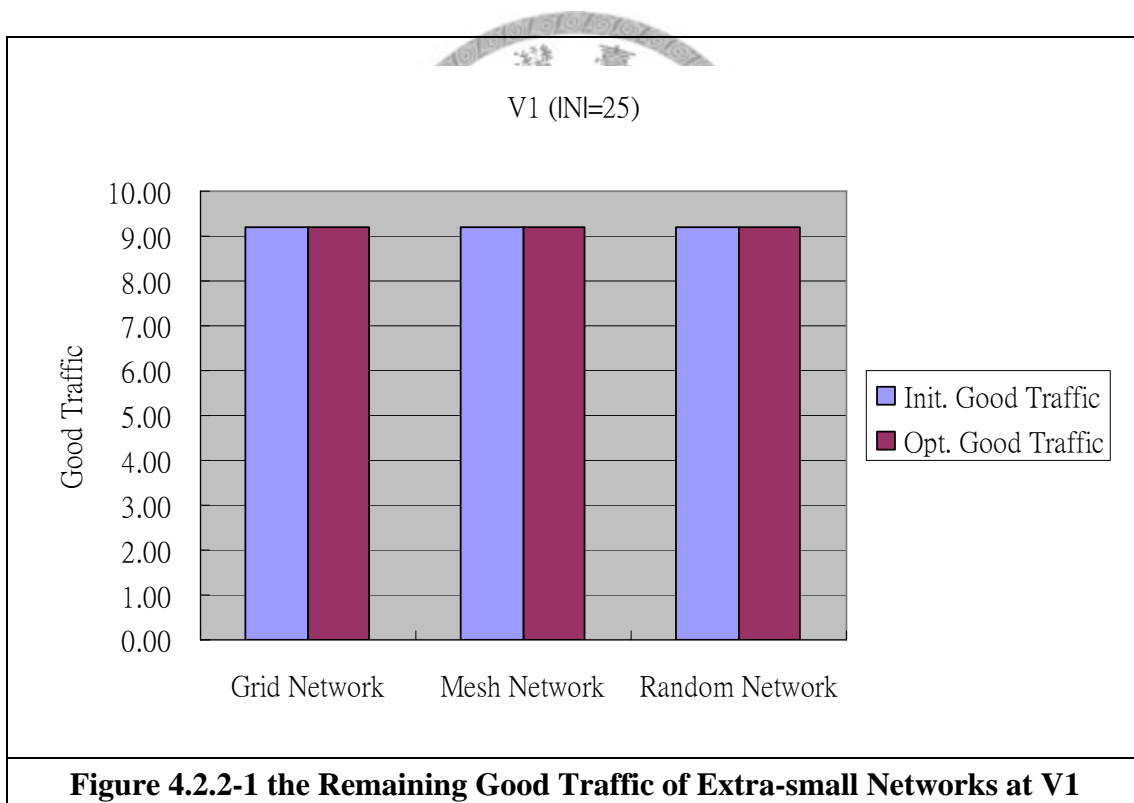
To evaluate the remaining good traffic under different scenarios in the ARAS model, we show the experiment results in the following tables. In each table, the Init. Good Traffic value means the remaining good traffic under the initial defender's filtering mechanism and routing assignment; The Opt. Good Traffic value means the equilibrium of the remaining good traffic resulting from the attacker's budget reallocation strategy. The improvement ratio, Imp. Ratio of Opt. Good Traffic, is calculated by $\frac{\text{Init. Good Traffic} - \text{Opt. Good Traffic}}{\text{Init. Good Traffic}} \times 100\%$.

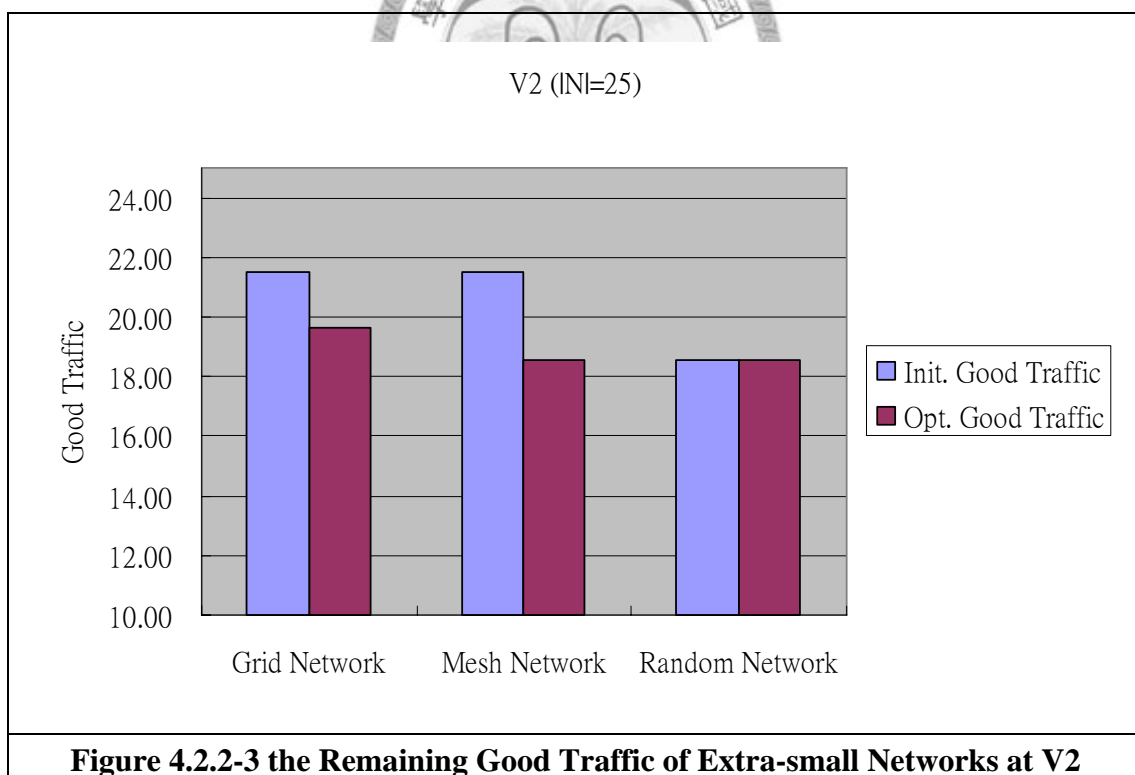
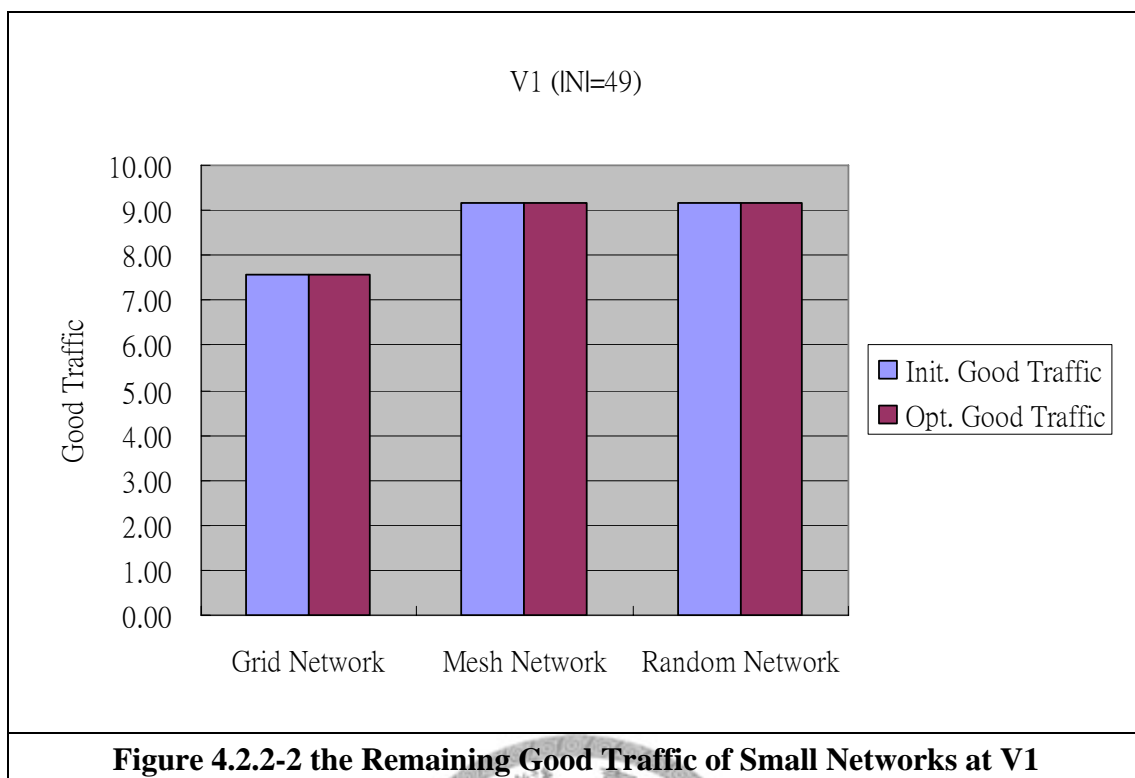
Table 4.2.2-1 Experiment Results of Extra-Small Networks (|N|=25)

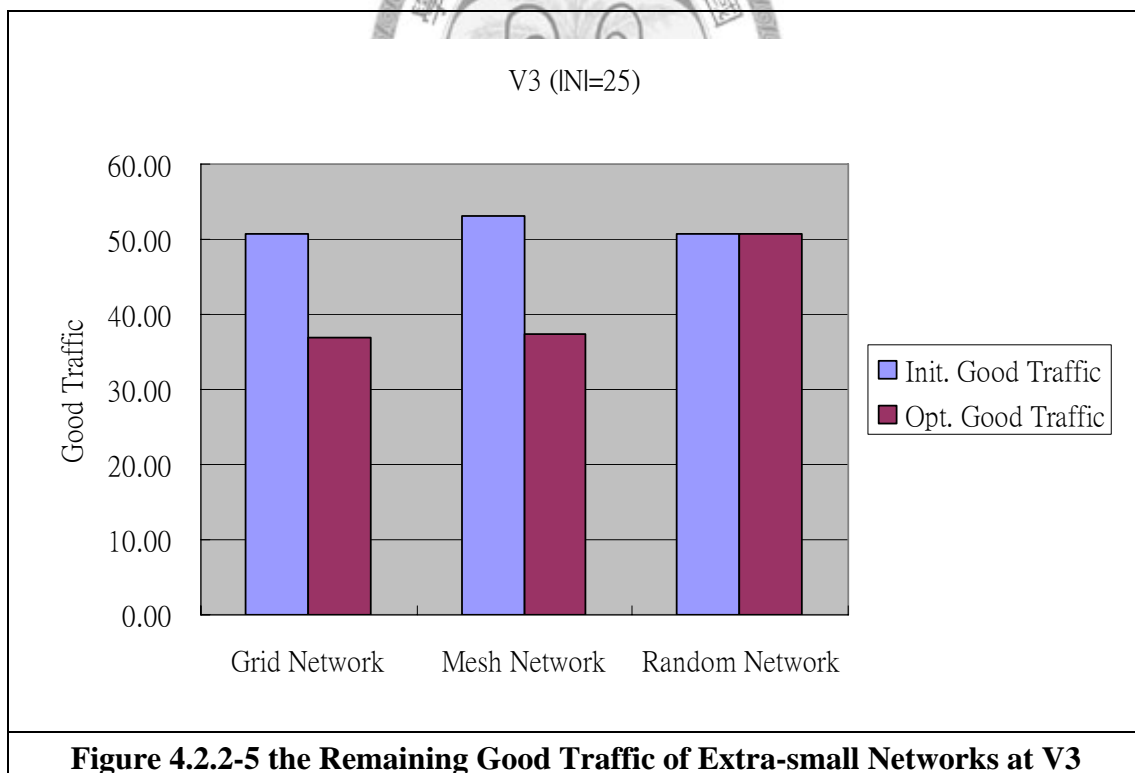
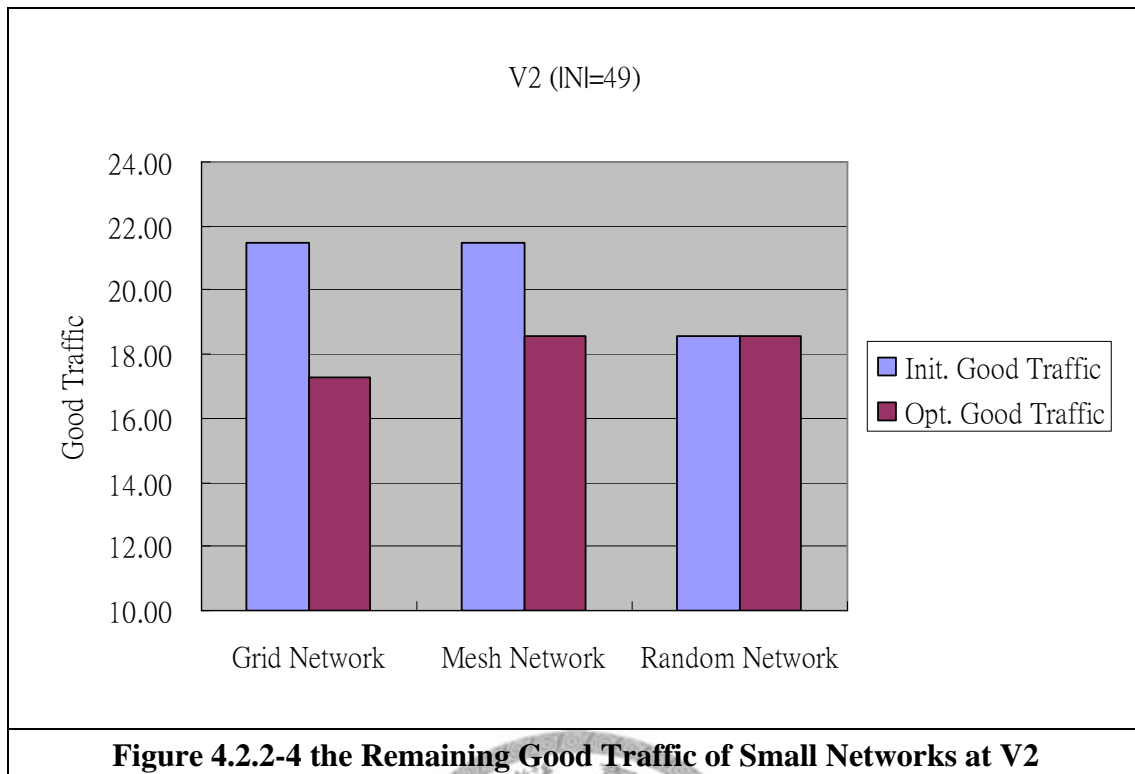
Network Topology	Victim Serv.	Init. Good Traffic	Opt. Good Traffic	Imp. Ratio of Opt. Good Traffic
Grid Network	V1	9.18	9.18	0.00
	V2	21.50	19.66	8.56
	V3	50.79	36.89	27.37
Mesh Network	V1	9.18	9.18	0.00
	V2	21.50	18.57	13.61
	V3	53.06	37.45	29.41
Random Network	V1	9.18	9.18	0.00
	V2	18.55	18.55	0.00
	V3	50.79	50.79	0.00

Table 4.2.2-2 Experiment Results of Small Networks ($|N|=49$)

Network Topology	Victim Serv.	Init. Good Traffic	Opt. Good Traffic	Imp. Ratio of Opt. Good Traffic
Grid Network	V1	7.59	7.59	0.00
	V2	21.50	17.28	19.61
	V3	55.10	49.73	9.74
Mesh Network	V1	9.18	9.18	0.00
	V2	21.50	18.55	13.73
	V3	53.06	51.85	2.28
Random Network	V1	9.18	9.18	0.00
	V2	18.55	18.55	0.00
	V3	53.06	53.06	0.00







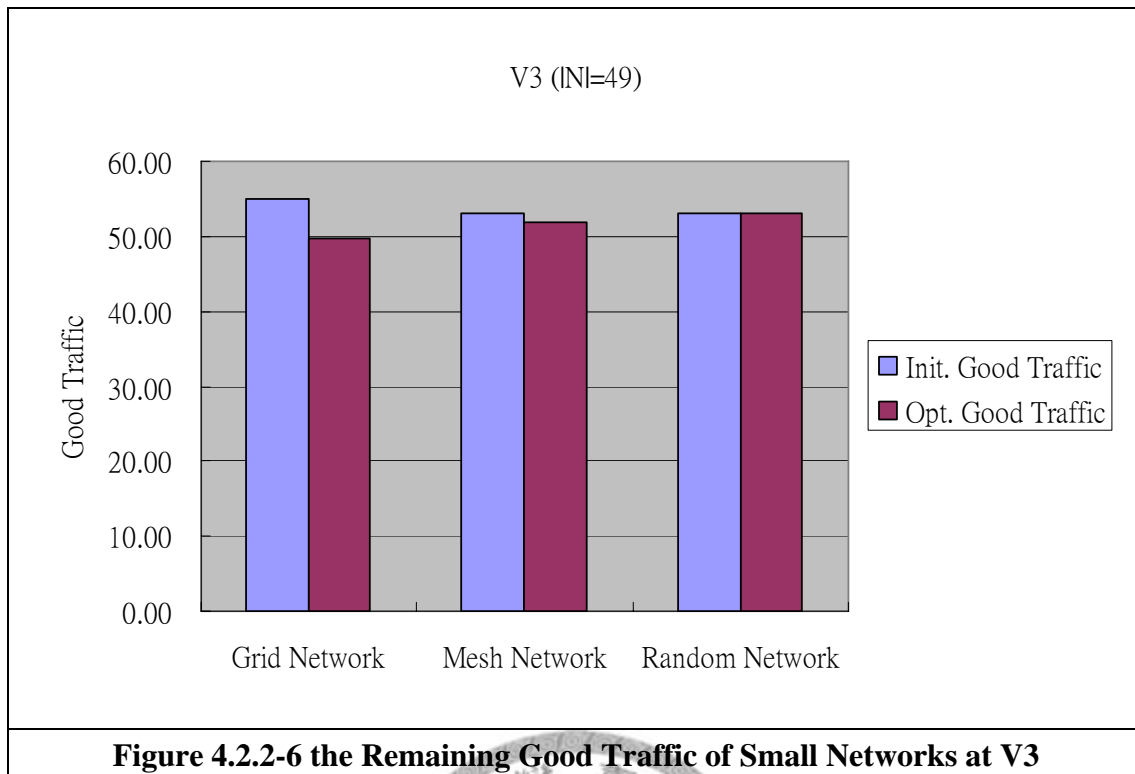


Figure 4.2.2-6 the Remaining Good Traffic of Small Networks at V3

4.2.3 Discussion of Results

Figure 4.2.2-1 ~ 6 displays the equilibrium remaining good traffic under different network topologies, numbers of nodes, and different numbers of victim servers. From these figures, we can find the following trends.

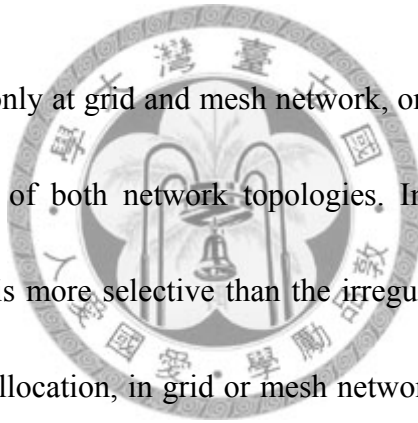
- Figure 4.2.2-1 and Figure 4.2.2-2 show the remaining good traffic under different network topologies at V1. In both figures, we observe no variation of the remaining good traffic at three topologies after the attacker's budget reallocation. The reason for the unchanged remaining good traffic is due to the V1. Initially, it is at the equilibrium. The attacker's total budget is quite enough to overwhelm V1. Thus, no matter how the attacker changes the budget allocation strategy, the

remaining good traffic still holds.

- Figure 4.2.2-3 and Figure 4.2.2-4 show the remaining good traffic under different network topologies at V2. In both figures, we could observe the variation of the remaining good traffic at both grid and mesh networks. Compared with V1, now the attacker's total budget is not enough to overwhelm V2. The initial status is not at the equilibrium. Thus, the variation appears.

- As for the variation only at grid and mesh network, one possible reason is due to the regular property of both network topologies. In the regular network, the routing path chosen is more selective than the irregular network. If the attacker changes the budget allocation, in grid or mesh network, the defender is likely to choose a new routing path for a new filtering, which mainly makes the variation. However, in the random network, because the choice of the routing path is not as rich as both networks above, the defender is less likely to make a new filtering.

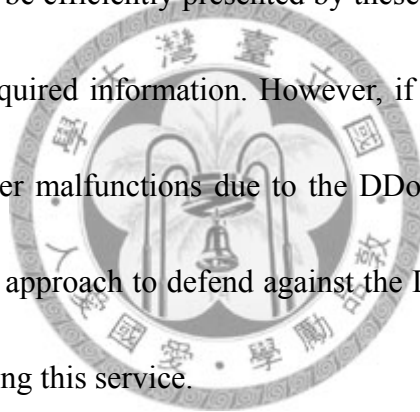
- Figure 4.2.2-5 and Figure 4.2.2-6 have similar trends as Figure 4.2.2-3 and Figure 4.2.2-4 except the remaining good traffic of both Init. Good Traffic and Opt. Good Traffic is preserved more.



Chapter 5 Conclusion and Future Work

5.1 Conclusion

Internet has become an important place to provide the daily information. Among many information-supportable media, the web server plays the most crucial role. Information request could be efficiently presented by these servers. Only with a finger click, you can find the required information. However, if one day or even one hour, the main operational server malfunctions due to the DDoS attack, the financial loss would be inestimable. An approach to defend against the DDoS attack, now, is rather urgent if we hope continuing this service.



In this thesis, we have successfully illustrated the attack-defense scenario in terms of the DDoS attack, where the DDoS attacker attempts to strategically allocate its budget to influence the legitimate user so as to minimize the legitimate traffic, while the defender tries to defend against the attacker by effective filtering mechanism and routing assignment. Ultimately, the equilibrium is reached by both the attack's and defender's strategies.

The main contribution in our work is the proposed mathematical model to describe the ARAS and FAS problems, which formulates the interaction between the DDoS attacker and the defender. Seldom previous works related to defend against DDoS attack consider using mathematical models to describe the DDoS attack scenario. We especially emphasize the remaining legitimate traffic for it's a critical factor for today you and me.

Moreover, as the scenario of multiple victim servers is considered, we find if the number of the victim server increases, the remaining good traffic could be preserved more. This further indicates if, in a network environment, more (victim) servers exist, the defender is more powerful to protect the legitimate user and, on the contrary, the attacker is less likely to penetrate the defense.

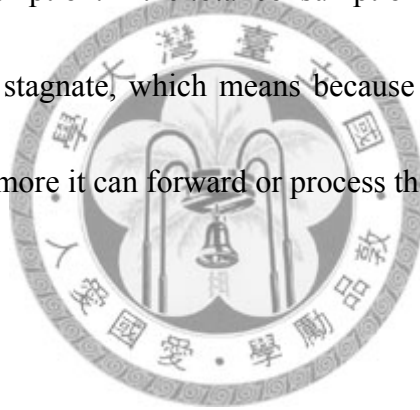
From the result of computational experiments, if there are more choices for the filtering remaining rate and routing paths, the result would be better. This is very important to derive our DDoS defense guideline, which is “the richer the choice is, the stronger the defender gets.”

5.2 Future Work

Some relevant issues and concepts which could extend our research are listed as follows.

- **Nodal Capacity**

In this consideration, we focus on nodal capacity consumption. We assume each node (router) in the topology we consider has a capacity limit. Not only filter allocation but filtering affects the nodal capacity. Both incur an amount of nodal capacity consumption. If the total consumption exceeds the nodal capacity limit, the node will stagnate, which means because the current nodal capacity reaches its limit, no more it can forward or process the upcoming traffic.



- **OD Pair Filtering**

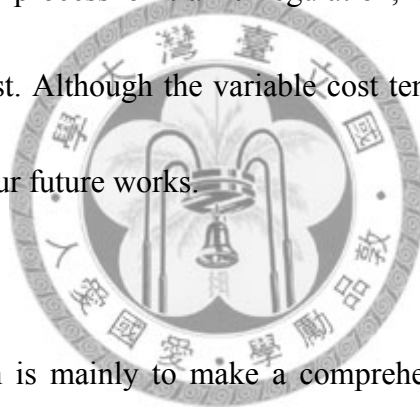
In this consideration, we put emphases on OD pair filtering. There are two aspects addressed here.

The first is “Filter OD Pair Handling” where the concept is the filter has an OD pair handling limit. The filter can just handle a number of OD pairs for a node. If the current OD pair handled reaches the limit, the remaining OD pair will not be regulated. The second is “Filter OD Pair Rate.” In this concept, the

filtering emphasizes each OD pair for a node. If a specific OD pair contains more aggregate traffic, the regulation can focus on this pair. The filtering rate is, thus, different for each OD pair

- **Filter Fixed and Variable Cost**

In this consideration, we focus on cost. We assume filter has a fixed cost incurred at the filter allocation. Different nodes have different allocation cost. Furthermore, in the process of traffic regulation, each filter rate adjustment incurs a variable cost. Although the variable cost tends to be insignificant, it is still meaningful in our future works.



The above extension is mainly to make a comprehensive consideration of the filtering mechanism. In the future, the follow-up research will be continued to improve the work.

References

- [1] K.Y. Yau, F. Liang and C.S. Lui, "On Defending Against Distributed Denial-of-Service Attacks with Server-centric RouterThrottles," *CERIAS Tech Report* 2001-39.
- [2] K.Y. Yau, C.S. Lui, F. Liang and Y. Yam, "Defending Against Distributed Denial-of-Service Attacks With Max-Min Fair Server-Centric Router Throttles," *IEEE/ACM Transactions on Networking*, Volume 13, Number 1, pp. 29-42, February 2005.
- [3] C.W. Tan, D.M. Chiu, C.S. Lui and K.Y. Yau, "A Distributed Throttling Approach for Handling High Bandwidth Aggregates," *IEEE Transactions on Parallel and Distributed Systems*, Volume 18, Number 7, pp. 983-995, July 2007.
- [4] H. Wang, C. Jin and K.G. Shin, "Defense Against Spoofed IP Traffic Using Hop-Count Filtering," *IEEE/ACM Transactions on Networking*, Volume 15, Number 1, pp. 40-53, February 2007.
- [5] C. Jin, H. Wang and K.G. Shin, "Hop-Count Filtering: An Effective Defense Against Spoofed DDoS Traffic," *Proc. of the 10th ACM conference on Computer and Communications Security*, pp. 30-41, 2003.
- [6] Y. Huang and J.M. Pullen, "Countering Denial-of-Service Attacks Using Congestion Triggered Packet Sampling and Filtering," *Proc. of the Computer*

Communications and Networks, pp. 490-494, 2001.

[7] K.E. Defrawy, A. Markopoulou and K. Argyraki, "Optimal Filtering for DDoS Attacks," *arXiv:cs/0612066*, 12 December 2006.

[8] S. Chen and Q. Song, "Perimeter-Based Defense Against High Bandwidth DDoS Attacks," *IEEE Transactions on Parallel and Distributed Systems*, Volume 16, Number 6, pp. 526-537, June 2005.

[9] C.C. Zou, N. Duffield, D. Towsley and W. Gong, "Adaptive Defense Against Various Network Attacks," *IEEE Journal on Selected Areas in Communications*, Volume 24, Number 10, pp. 1877-1888, October 2006.

[10] G. Levitin, "Optimal Defense Strategy Against Intentional Attacks," *IEEE Transactions on Reliability*, Volume 56, Number 1, pp. 148-157, March 2007.

[11] D. Magoni and J.J. Pansiot, "Analysis of the Autonomous System Network Topology," *ACM SIGCOMM Computer Communication Review*, Volume 31, Issue 3, pp. 26-37, July 2001.

[12] V.R. Westmark, "A Definition for Information System Survivability," *IEEE Proc. of the 37th Hawaii International Conference on System Sciences*, Volume 9, 2004.

[13] M.L. Fisher, "The Lagrangean Relaxation Method for Solving Integer Programming Problems," *Management Science*, Volume 27, Number 1, pp. 1-18, January 1981.

- [14] Z. Zeitlin, "Integer Allocation Problems of Min-Max Type with Quasiconvex Separable Functions," *Operations Research*, Volume 29, Number 1, pp. 207-211, January-February, 1981.
- [15] F.Y.S. Lin, P.H. Tsang and Y.L. Lin, "Near Optimal Protection Strategies Against Targeted Attacks on the Core Node of a Network," *Proc. of the Second International Conference on Availability, Reliability and Security*, 2007.
- [16] F.Y.S. Lin, C.L. Tseng and P.H. Tsang, "Near Optimal Attack Strategies for the Maximization of Information Theft," *Proc. of the The Second International Conference on Availability, Reliability and Security*, pp. 213-222, 2007.
- [17] W. Stallings, *Cryptography and Network Security*, 4th Edition, 2005.
- [18] A.M. Geoffrion, "Lagrangian Relaxation and its Use in Integer Programming," *Mathematical Programming Study*, Volume 2, pp. 82-114, 1974.
- [19] M.L. Fisher, "An Application Oriented Guide to Lagrangean Relaxation," *Interfaces*, Volume 15, Number 2, pp. 10-21, April 1985.
- [20] R. Chen, J.M. Park, and R. Marchany, "A Divide-and-Conquer Strategy for Thwarting Distributed Denial-of-Service Attacks," *IEEE Transactions on Parallel and Distributed Systems*, Volume 18, Issue 5, pp. 577 – 588, May 2007.
- [21] L.A. Gordon, M.P. Loeb, W. Lucyshyn, and R. Richardson, "2007 CSI/FBI Computer Crime and Security Survey," *Computer Security Institute*, 2007,

<http://GoCSI.com>.

- [22] K.E. Defrawy, A. Markopoulou and K. Argyraki, “Optimal Allocation of Filters Against DDoS Attacks,” *Information Theory and Applications Workshop*, pp. 140-149, 2007.



簡歷

姓名：江政祐

出生地：台灣省台北市

生日：中華民國七十年三月十五日



學歷：九十二年九月至九十五年六月

國立政治大學 資訊管理學系學士

九十五年九月至九十七年六月

國立台灣大學 資訊管理研究所碩士