2018

Implementation of Cache Attack on Real Information Centric Networking System

Faustina J. Anto Morais
University of North Florida, faustina2903@gmail.com

IMPLEMENTATION OF CACHE ATTACK ON REAL INFORMATION CENTRIC NETWORKING SYSTEM

by

Faustina J. Anto Morais

A thesis submitted to the
School of Computing
in partial fulfillment of the requirements for the degree of

Master of Science in Computer and Information Sciences

UNIVERSITY OF NORTH FLORIDA
SCHOOL OF COMPUTING

August, 2018

The thesis "Implementation of Cache Attack on Real Information Centric Networking System" submitted by Faustina J. Anto Morais in partial fulfillment of the requirements for the degree of Master of Science in Computing and Information Sciences has been

Approved by the thesis committee: Date:

Dr. Swapnoneel Roy
Thesis Advisor and Committee Chairperson

Dr. Sanjay P. Ahuja

Dr. Sandeep Reddivari

Accepted for the School of Computing:

Dr. Sherif Elfayoumy
Director of the School

Accepted for the College of Computing, Engineering, and Construction:

Dr. Mark Tumeo
Dean of the College

Accepted for the University:

Dr. John Kantner
Dean of the Graduate School

ACKNOWLEGEMENTS

I would like to first thank my thesis advisor, Dr. Swapnoneel Roy, for his patient and helpful advice in the completion of my research work. This thesis would not have been possible without his direction and support. I would also like to thank my committee members, Dr. Sanjay P. Ahuja and Dr. Sandeep Reddivari, for their valuable suggestions. I also thank Professor Walt H. Schuller for guiding me in this process and being available to help me in the security laboratory where I did this work. This thesis provided me a great learning experience on security attacks and its impacts. It also taught me real patience in troubleshooting when something does not go as expected, a skill which is useful in the networking field outside of academics. Last but not the least, special thanks to my parents for supporting me throughout my studies at University of North Florida. Without their support, I wouldn't have been able to finish my Master's degree successfully.

CONTENTS

FIGURES

TABLES

ABSTRACT

Network security is an ongoing major problem in today's Internet world. Even though there have been simulation studies related to denial of service and cache attacks, studies of attacks on real networks are still lacking in the research. In this thesis, the effects of cache attacks in real information-centric networking systems were investigated. Cache attacks were implemented in real networks with different cache sizes and with Least Recently Used, Random and First In First Out algorithms to fill the caches in each node. The attacker hits the cache with unpopular content, making the user request that the results be fetched from web servers. The cache hit, time taken to get the result, and number of hops to serve the request were calculated with real network traffic. The results of the implementation are provided for different topologies and are compared with the simulation results.

CHAPTER 1

Introduction

The rise of Internet of Things (IoT) and the Internet being the primary source for exchange of information over the network has exploded in recent years. Information-centric networking is noteworthy in that it offers benefits such as improved efficiency, better scalability in bandwidth demand and better robustness. Improved efficiency is obtained with the temporary user data store in the cache and hence information centric network depends on caching for its implementation. Much of the research in the network security field concerns theoretical and simulation results. However, the results of network simulation often do not reveal problems that will affect the final result. Therefore, the primary focus of this research is to compare the result of a cache attack on a simulation with one on a real information centric network (ICN).

Information centric networks (ICNs) have pulled in considerable interest of recent researchers in computing [14, 20, 31, 37]. The major topic of research has been focused on the use of data caching to improve the performance of the network. However, research on security concerns of data caching is still lacking. An example of a scenario that lacks security could be a mobile ad hoc network (MANET), a self-organized network system without any secured infrastructure.

The effects of a Denial of Service (DoS) attack on simulated ICNs have been studied in [27]. The main goal of the attacker was to fill node caches with unpopular content, impeding caching in the ICN without increasing the likelihood of getting

detected. The attack originated from a malicious node that requested unpopular content at regular intervals. The decreased throughput and increased delay also led to higher energy consumption by the nodes of the ICN, thus reducing overall performance of the network. Using a network simulator, it was found [27] that such an attack was moderately successful against small scale networks. However, the potency of the attack rapidly decreased and became ineffective as the network size increased. In this research, Random, FIFO and LRU replacement policies were used to implement the cache attack considered different cache sizes and topologies. It is found that the attack was more effective against a First In First Out (FIFO) in comparison to a Least Recently Used (LRU) replacement policy.

1.1 Problem Statement

ICNs rely on caching to increase the performance of the network, thereby reducing the number of hops needed to respond to a query. ICNs have evolved from host-centric to content-centric, and the main problem in data exchange is content caching. The authors in [27] tested a DoS attack against a simulator that simulated ICNs with various network topologies. The attack was found to be more effective for smaller networks. But testing the effects of a DoS attack on the caches of a real ICN is currently missing from the literature and is an open problem. In this work, such an attack is tested over small real ICNs. Specifically, the scalability and effectiveness of a DoS attack on real ICNs is compared to those obtained from the simulator in [27].

1.2 Research Objective

The research initially started as a study of caching-related issues in ICNs. Being a comparatively new research area, most of the research on ICNs has focused on efficient routing and cache management policies to enhance the performance of ICNs (e.g. [42, 44, 48, 49, 63]). There has been much less research done on attacks related to caching. Motivation for this work came from [18], where the authors focused on detecting cache pollution attacks in Named Data Networks (NDNs). The paper proposed a new method for launching a denial of service (DoS) to attack NDNs and also proposed ways to improve cache pollution attacks. The concepts and methods specified by [18] for NDNs were used in this work for ICNs. Our main goal for this work was to develop a method for performing a DoS attack on real small ICNs ranging from 5 to 20 nodes and on different topologies including path and mesh topology.

CHAPTER 2

Background and Existing Research

In this chapter, the key concepts and definitions related to caching attacks in networking systems are discussed in order to understand the implementation of the proposed cache attack in real ICNs.

2.1 Information Centric Networking (ICN)

An ICN focuses on content objects that can be accessed or cached anywhere in the network rather than solely from the end hosts. The use of ICNs has become more prominent in recent times since 2010 due to heavy Internet traffic flow. With the evolution of the Internet from being host-centric to being network-centric, ICN aims to provide in-network caching to deliver content efficiently. The main reason to shift from host-centric to content-centric or information-centric networking is that the Internet is currently focused on delivering high volumes of digital data (e.g. 3D, 4D, movies, pictures) to users. The users need only the content of the data and are not interested in the source location.

Being a comparatively new research area, and most of the ICN research has focused on improving routing [7, 14], and not as much research on issues related to security and privacy [1, 20] of ICN. In [1], the researchers briefly discussed types and impacts of ICN attacks and how the attacker depends on the ICN attributes to perform

the attack. The impact of security requirements and the severity levels of attacks were also clearly stated with the solutions.

As shown in the Fig. 2.1, ICN communication takes place hop-to-hop from the user to the server or to the nearest hop to fulfill the user request.

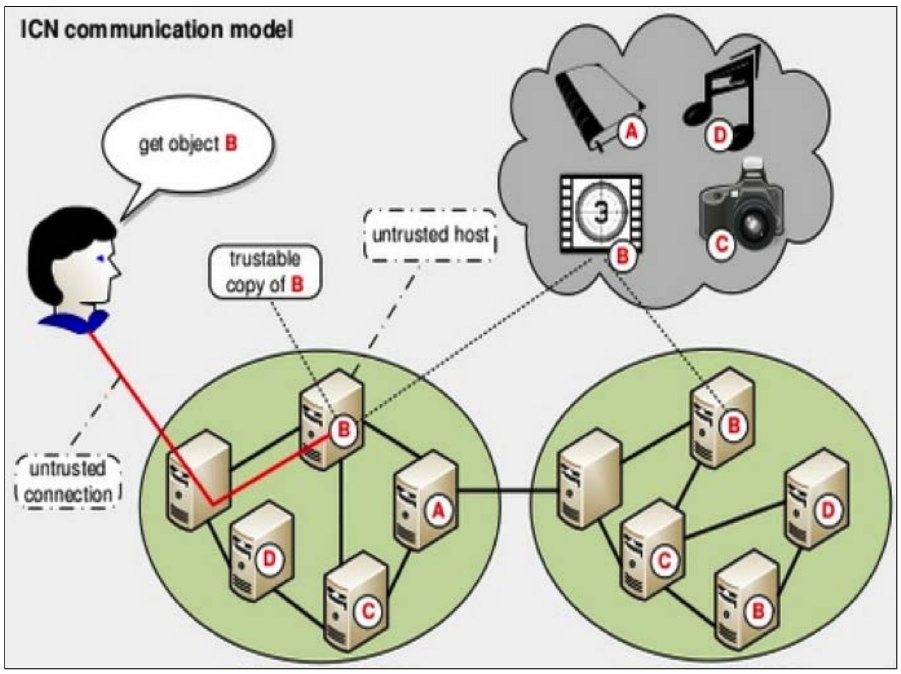

Figure 2.1: ICN Communication Model

The authors in [20] provided a brief assessment of DoS and distributed denial of service (DDoS) attacks and suggested a few possible countermeasures. The authors identified new types of DoS-based attacks, which can impact an ICN network interest flooding and content/cache poisoning and provided a brief discussion of their effects and countermeasures.

The researchers in [20] also discussed the privacy risks of ICN caching and continued investigating to the extent by which the nearby user, in the role of an

adversary, has requested the content and they proposed a solution to reduce such attacks. In [43], the authors presented a framework for delivering content in ICNs securely and with high availability.

In [58], the authors proposed a secure naming system that allows decoupling content authentication from its location in a network. The benefits of the proposed methodology in [58] are security of mapping functions between high-level names and cryptographic identifiers at the network level, authentication of the content with its provider regardless of the location from where it was retrieved, reduction of possible threats during the resolution procedure [20].

2.2 Cache attack

The cache memory is the temporary memory shared by multiple processes. The cache is accessible by the user and the attacker. This temporary memory is used to store the content invoked by the user for any requests to improve the efficiency for the future requests. Thereby if the user requests such content, it will be fetched directly from the cache itself or by the nearby hosts without reaching the web server to fetch the user data. A caching attack is defined as the filling of the cache with some random data.

Although simulation results are available that compare the performance of a caching attack in the ICN [27], there are no experimental results that evaluate the performance of cache attack on real ICNs. That is the focus of this research.

2.3 Related work

A considerable amount of research has proven the effectiveness of using data caching to increase performance, but research on the implementation of data caching in real networking systems and foiling attacks is still lagging behind.

2.3.1 Motivation for this Research

In designing a secure network, research involving the software simulation of attacks is not enough, and there have been experiments to test and compare simulation results with that on real network systems.

In [2, 3], firewalls and VPNs produced different results in real networks when compared to prototypes and virtual modeling software because the simulation results were not based on real network traffic.

In [38], the authors have performed a comparison between analytical and simulation research on ad hoc wireless networks.They identify factors in wireless networks like hills, obstacles, link asymmetries, and unpredictable fading not encapsulated by a simulator.

In [12], the researchers stress the need to test “Internet protocols under varied conditions to determine whether they are robust and reliable". Specifically, they mention the following necessary simulation scenarios every network simulator should

capture:

1. Network topologies that define links and their characteristics.

2. Traffic models that specify sender and receiver locations and demands.

3. Network dynamics that include node and link failures

The researchers of this paper developed a common network simulator platform to encompass various scenarios on which other researchers can test various networking protocols. However, they concluded that their simulator still lacked aspects of a real network. These aspects may include:

1. Developing mechanisms for the successful integration of code from the user community; and

2. Developing tools for large-scale simulations with a diverse traffic mix.

These could only be captured by experimenting over real networks.

The researchers of [8] also identify shortcomings of network simulators. They point out that wide-area testbeds and custom simulators though valuable, have significant shortcomings which can only be studied through real networks. These approaches often lack the wide mix of network traffic and topologies found in real networks. Also, the repetition of experiments under controlled conditions can be difficult to expose the real network traffic.

The researchers in [5, 62] point out the importance of performing real experiments over simulations for neural networks. In particular, while using simulators, the experimental design chosen for the neural network training set can have a very high

impact on the predictive accuracy achieved by the simulator. Hence simulation results become limited in their accuracy.

These differences in results were the motivation for this research work. It has been attempted to test and compare results produced by [27] over a simulator to that of real ICNs. This attempt is made to investigate whether the cache attack indeed produces the same effect over real ICNs in presence of factors that are not present in a simulator, e.g. network traffic, delay in packet transmission due to network bandwidth, web-based viruses, etc.

CHAPTER 3

Implementation of Real ICN

In this research, cache attacks have been implemented over real information centric networking systems. To better understand the research, the technologies used in ICN network implementation are discussed in detail in this chapter.

3.1 Tools Used for Network Implementation

Technologies (tools) described in this section have been used to implement ICNs in this research work; these technologies are also used to implement a broad array of other networks.

3.1.1 Content Delivery

Content delivery is the service of copying the pages of a website to geographically dispersed servers and when the page is requested, dynamically identifying and serving page content from the closest server to the user, enabling faster delivery [29, 51, 54, 59, 60]. Content delivery works by placing cache servers at main access points around the world and using a routing code such as Hypertext Transfer Protocol (`HTTP`), which redirects the request for the web page to the closest server.

When a user clicks on the URL, which is content enabled, the content delivery network reroutes from the originating server to the cache server which is closest to the user. The cache then determines if the requested content exists in the cache server and then serves that content. If the content is new then it is retrieved from the web server and cached locally.

3.1.2 Squid Web Proxy Server

Squid (software) is a web proxy server daemon supporting `HTTP`, `HTTPS`, `FTP` and many more [10, 23, 25, 39, 56]. It is responsible for caching and reusing frequently requested web pages and thereby helps in reducing the response time. It also helps in caching web, `DNS` and other computer network lookups for groups of people sharing network resources. In order to optimize network throughput, Squid routes content requests to servers in many different ways to build cache server hierarchy. Squid is used by Internet service providers to increase network speed to their customers and also to LANs that share their Internet connection. Because of its proxy nature, which filters network traffic, it provides security and anonymity.
The configuration file to configure the Squid server is given in Fig. 3.1.

```
$vi/etc./wgetrc;
$cd /var/squid/squid.conf
```

Figure 3.1: Squid Configuration

3.1.3 Wget Function

GNU Wget (`Wget`) is a software package for retrieving content from web servers. It supports downloading files from `HTTP`, `HTTPS` and `FTP` protocols as well as retrieval through `HTTP` proxies [36, 45, 46, 47, 55]. It is a non-interactive command line tool that can easily invoke scripts and fetch data. Some of its advantages include robustness, recursive download, portability and support for proxy. The configuration file for `Wget` is given in Fig. 3.2.

```
$vi /etc./wgetrc
```

Figure 3.2: `Wget` Configuration

Wget can be invoked by a simple command line whose basic syntax is given in Fig. 3.3.

```
$wget [option] ··· [URL] ···
$wget https://www.unf.edu/
```

Figure 3.3: `Wget` Invoke

3.1.4 Rsyslog

`Rsyslog` is open-source software used on UNIX-like operating system for transmitting log messages in an IP network. It is helpful for logging information and

accepting inputs from different sources and transforming and outputting the results to different destinations, thus supporting both local and remote logging [13, 19, 26, 40, 53]. Every logged message contains a timestamp, a hostname field and a program name field. `Rsyslog` supports precise timestamps and writing directly to databases. `Rsyslog` is also used for implementing DoS attacks: a programmer could flood the `Rsyslog` daemon with `Syslog` messages resulting in the log files consuming all the remaining space on the file system [17, 30]. Some of the features of `Rsyslog` include excellent security and modular design. The configuration file for `Rsyslog` is given in Fig. 3.4.

```
$cd /etc/rsyslog.conf
```

Figure 3.4: `Rsyslog` Configuration

3.2 Implementation Methodology

In the real network, the ICN was initially set up with the cache and web servers. This setup allows the user to access or fetch data from the web server. The user request will be sent as query to the server, which then will be served by a nearby cache or the web server based on content availability. The basic steps in the methodology were:

1. Implement caching networks over nodes.
2. Implement querying for documents over the networks.
3. Assuming one or more nodes are compromised, implement the attack and evaluate the change in performance of the networks.

3.2.1 Overview of the Implemented ICNs

In this sub section, we provide an overview of the implemented ICNs on which we have tested the effect of the cache attack.

The two ICNs (with 4 and 8 nodes) implemented for this research worked exactly the same way as shown in the example of Fig. 3.5. The ICN in Fig. 3.5 contains 8 nodes. The last (rightmost node) is the web server that would contain all files the user would request.

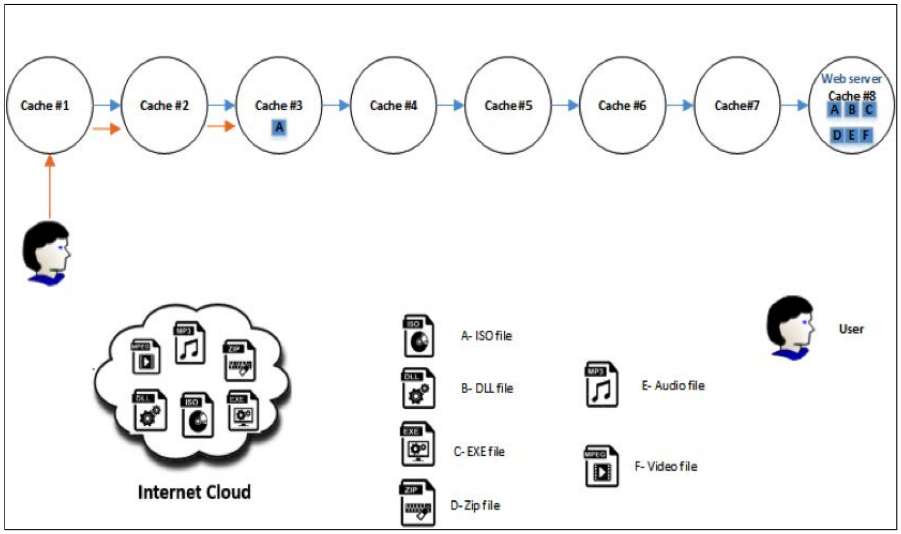

Figure 3.5: Overview of the ICNs

In network terminology, such a web server that owns a particular file, is called the custodian of that file. In the ICN of Fig. 3.5, the 7 remaining nodes could be users querying for files. Each of these nodes will have its own cache that stores any file relayed via that node. One such node in the ICN (the leftmost node) is designated as the user for illustration. When the user generates a query for a file, the query

propagates through all the nodes in the route. The file is searched for in the cache of any intermediate node in the path. The query is answered if the file is found in the cache of any of these nodes. If not, the query is answered only after reaching the web server, which then relays back the file via the intermediate nodes to serve the request. These intermediate nodes will store the file in their respective caches. The idea of ICN as mentioned before is a distributed system of nodes that would decrease the response times for queries using caches. It is expected that the next query for the same file by any user will not have to propagate to the web server, but will be catered by one of the intermediate nodes (being found in its cache). This way, the average response time for the ICN is reduced, enhancing the overall performance of the network.

3.2.2 Specification of the Implemented ICNs

Each node in the ICNs contains a hash table to direct a query to the next hop to serve the user's request. The network topologies considered in this research were line and mesh. Detailed description of these topologies are given in Chapter 4. In the real world (e.g. videos of Youtube), some documents (files in our case, and videos in case of Youtube) have much higher popularity over others. For example, some videos in Youtube have over a million views, while there are others with less than 10 views. It is well known that web content request popularity follows a Zipfian-like distributions [11, 24, 33, 34, 35, 61]. The file popularity level in this work was assumed to follow the same Zipfian distribution. The common values considered in the Zipfian distribution are usually chosen as $\alpha = 0.65$ and $\alpha = 0.85$. Since the change in alpha value will have the same difference ratio, $\alpha = 0.65$ has been used throughout this experiment. The caches of all the nodes will be empty in

the initial phase, and thus a warm-up test was carried out to fill the caches before experimental results were calculated. Parameters used to measure performance were:

1. Average number of increase in hops for queries.
2. Average delay for each request in the network.

For each user's request, Dijkstra's algorithm [32] is used to find the shortest path to the custodian which could be either the cache of an intermediate node in case of a popular file, or the web server (if the file is not found in the cache of any intermediate node in the path). The cache sizes considered in this research are 10 MB and 40 MB, and the number of nodes considered are 4 and 11. For experimental purposes, the ICN with 4 nodes is considered as a small network and the ICN with 11 nodes is considered a mid-sized network. In this work, the effect of the cache attack on both these ICNs with the real network traffic and firewall in place is compared to the results stated in [27] for networks of same sizes over the simulator.

3.2.3 Idea of the Performed Cache Attack

The goal of the attacker is to feed caches of all nodes in the ICN with extremely unpopular content (files in this case) and therefore render every user's request unavailable in the cache forcing it hop till the web server. Hence the time taken for each request (query) will be more, increasing the average response time for query for the ICN, and reducing its efficiency. The assumption is the attacker has control over one or more nodes in the ICN from which the attacker generates

queries for extremely unpopular contents periodically in order to fill in the caches of the intermediate nodes with these unpopular contents. The node(s) controlled by the attacker is (are) called compromised node(s).

This cache attack is performed on the both the ICNs (with 4 and 11 nodes) with one compromised node. Then the same attack is repeated with two compromised nodes for both the ICNs, and also with four and eight compromised nodes for the ICN with 11 nodes. To recall, according to [27], in the large network, the impact of cache attack was less whereas in the smaller network, the impact was higher.

CHAPTER 4

Testbed Setup

In this chapter, we illustrate the experimental setup in details. At the onset, we define some terms that we use throughout in this thesis.

4.1 Terminologies Used

- Nodes: Nodes are defined as any machine in the network (ICN).
- Requester Nodes: These are the nodes that would periodically request data (files) in the network (ICN). Requester nodes are also referred to as users or user nodes in this work.
- Attacker Nodes: These are the nodes that would periodically request extremely unpopular data (files) in the network (ICN). Attacker nodes are also referred to as compromised nodes in this work. The intent of the attacker as mentioned before is to fill out caches with unpopular content.
- Web Server: This node contains all files the requester nodes would request. This node is also referred to as a custodian node in this work.

4.2 Cache Setup

The code that accomplishes any cache activity such as filling the data in the cache and fetching the content has been written using Python scripting language. The open source Squid web proxy has been installed in every node including the web server. The cache proxies contents were stored in the main memory and ensured that the system was not shut down during the experiment.

The experiments were made without attacker and with 1 and 2 attacker(s) were implemented on both the ICNs with 4 and 11 nodes. Additionally, experiments with 4 and 8 attackers were performed with the ICN with 11 nodes (caches). The user sends a `HTTP` request as shown in the Fig. 4.1, which then seeks the content on the caches of the intermediate nodes in the path, and then onto the web server (if the content was not found in any of the caches of the intermediate nodes).

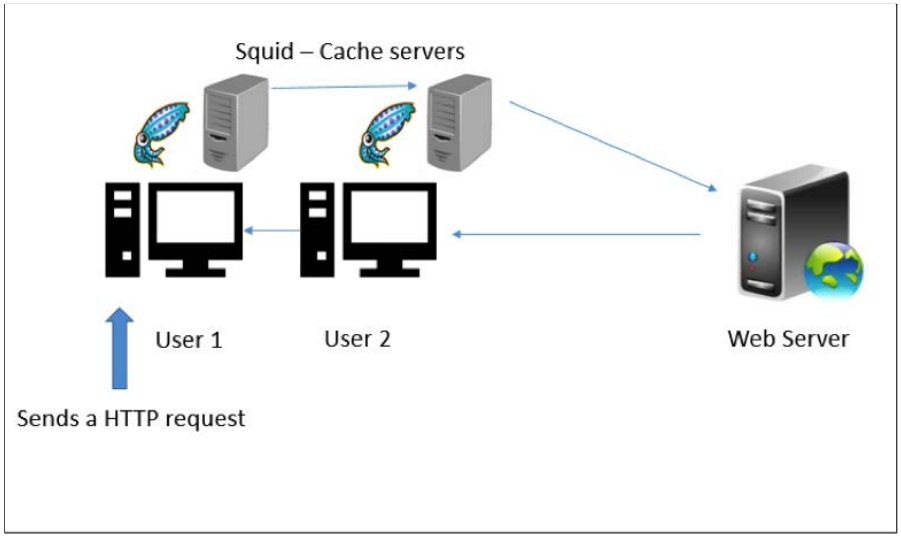

Figure 4.1: Squid Cache Setup with Web Server

Each node was allocated a static (specific) IP address throughout the experiments. The caches were stored in the main memory of each node, and therefore by default, Dynamic Host Configuration Protocol (DHCP) is invoked to assign new IP addresses to the nodes each time the ICN is restarted [6, 21, 22, 41, 50]. To implement static IP address configurations, Dynamic Host Configuration Protocol (DHCP) was disabled and a static IP address was defined for each node as shown in Fig. 4.2.

```
$vi /var/log/squid/ip.txt
192.168.103.205 cache 0
192.168.103.206 cache 1
192.168.103.207 cache 2
192.168.103.208 cache 3
192.168.103.209 cache 4
192.168.103.209 cache 5
192.168.103.209 cache 6
192.168.103.209 cache 7
192.168.103.209 cache 8
192.168.103.209 cache 9
192.168.103.209 cache 10
192.168.103.209 cache 11 / webserver
```

Figure 4.2: Static IP Address Definitions

The following entries were added at the end of the configuration file to disable DHCP, and the file was saved as shown in Fig. 4.3.

```
# disable ipv6
net.ipv6.conf.all.disable_ipv6 = 1
net.ipv6.conf.default.disable_ipv6 = 1
net.ipv6.conf.lo.disable_ipv6 = 1
```

Figure 4.3: Entries to the Configuration file

4.3 Firewall Configuration

The Linux `IPtable` configuration file was used to define the incoming and outgoing ports to accept or deny incoming content. This is the process which secured the network and monitored the traffic flow within the network. Also the use of the firewall differentiates the experimental results from simulation results (of [27]), since this experiments involve real network traffic. The configuration file for the firewall is shown in Fig. 4.4.

```
$cd var/log/squid/iptables.config
$cd var/log/squid/squid.config
```

Figure 4.4: Firewall Configuration

4.4 Web Server Setup

One of the nodes was considered to be the web server where all the files were stored. Squid proxy was set up in the web server as well. The files in the web server could be fetched from any other node per the user request using the `wget` function. As shown in Fig. 4.5, the first line defines the web server with an IP address of `192.168.103.254`. The next line is a query for the file `file1` requested

from this web server. The attacker also used the same technique to fetch files and populate caches of other nodes with unpopular content (files).

```
$192.168.103.254 cache 11 / webserver
$wget http://192.168.103.254/file1
```

Figure 4.5: Web Server Setup

4.5 Topology Definition

The two topologies considered were path and mesh topology. Path topology is defined as the path of order n that has n vertices denoted by P_n [15]. Among these two topologies, the mesh topology exhibits the real world scenario. But in a full mesh topology, every node of the network is connected to every other node. In an ICN, however this is not feasible since this defeats the purpose of caching (since every node becomes a neighbor of the web server). Therefore to better represent an ICN, a partial mesh network has been implemented and used for the experiments. Both these networks were implemented to compare results of the simulation of [27] with those of real ICNs. As shown in the Fig. 4.6, the caches were built on each work station (node) along with the web server. In this topology, the user request and the attacker request will be generated at random. The network is full duplex.

Fig. 4.8 and Fig. 4.7 show the representation of the partial mesh topologies with 4 and 11 nodes respectively used in this work.

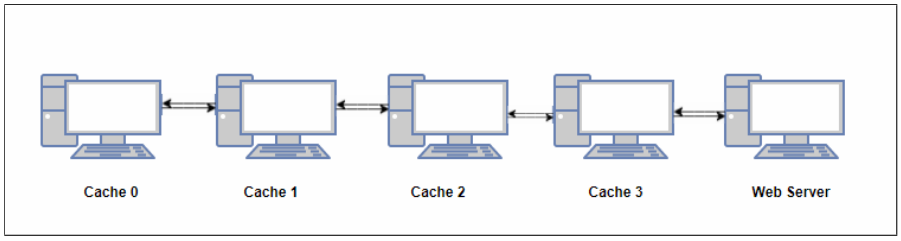

Figure 4.6: Path (P_n) Topology

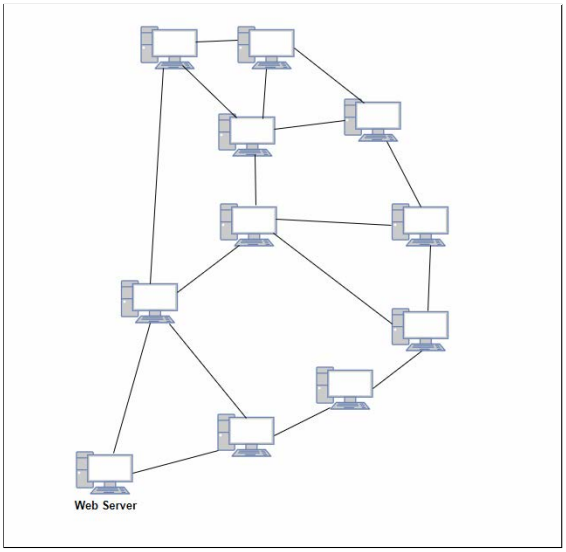

Figure 4.7: Partial Mesh Topology with 11 Nodes

In the partial mesh topology, each node has at least two neighbors [52]. An exception to that has been made in Fig. 4.8, where the web server is connected to only one node. Each user request will use the Dijkstra's algorithm [32] to find the shortest path to fetch the file either from the cache of another node (including its own cache), or from the web server in case the cache of any intermediate node does not contain the requested file.

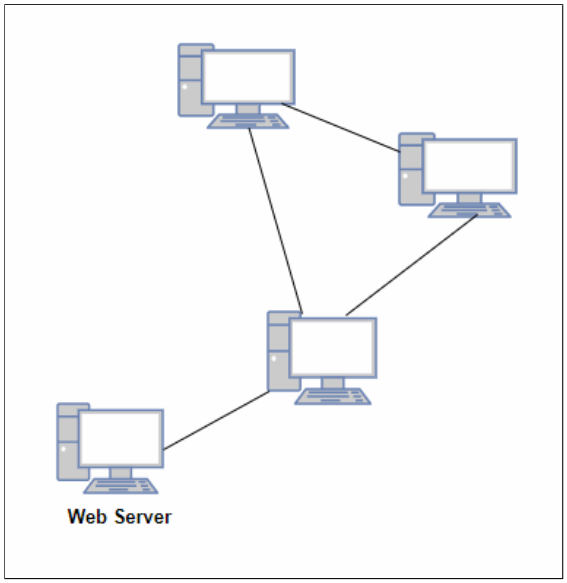

Figure 4.8: Partial Mesh Topology with 4 Nodes

4.6 Hardware and Software Specifications

4.6.1 Hardware Specifications

For this research work, all the caches/nodes had the same configuration as described in Table 4.1.

4.6.2 Software Specifications

Linux flavor Ubuntu14.0 AMI was mounted on the workstations. Java JDK version 1.7 was installed, along with Python 2.6. The firewall was configured using Linux

Operating System	Gnome Ubuntu 16.04
Processor	Intel Core 2 Quad Q9300 2.5 GHz x4
OS Type	64 bit
Memory Size	3.7 GB
Disk Size	28.2 GB

Table 4.1: Hardware Configuration

IPtable. Squid open source proxy was installed on every node (including the web server node). Squid configuration file was configured in each node to specify the node and the next hop, thereby implementing line and partial mesh topologies.

CHAPTER 5

Results and Analysis

Once the ICN was set up with the line or partial mesh topology, the system was ready to implement the cache attack on the ICN. The smaller network consisted of 4 nodes and the larger network consisted of 11 nodes.

In [27], the researchers found in a simulation environment that the attack was considerably more effective with FIFO replacement policy over LRU and random. In this research therefore LRU, Random, and FIFO policies have been used to compare the results with [27]. There were a fixed number of files with a fixed size for each experiment scenario. The popularity of each file was assigned based on a Zipfian distribution with a Zipfian α value of 0.65. This α is the most common Zipfian value. Another common value of α, (0.85) has been used in [27] for very large networks of more than 100 nodes, which were not considered in this research (due to lack of resources).

It is to be noted, since the value of α remains constant throughout the set of experiments, the same set experiments performed with a different value of α will only change numerical results, but not the overall conclusions obtained the experiments. The results of using a different value for α will thus be numerically but not qualitatively different, although a fixed value of $\alpha = 0.65$ has been used in these experiments.

5.1 Evaluation Scenarios

Once the popularities of the files were set with the given Zipfian distribution α value, the effectiveness of the attack and the average time taken for the ICN to respond to a request without and in presence of attacker(s) was measured. For this research cache sizes of 10 MB and 40 MB were chosen. The different values of parameters used for both path and partial mesh scenarios are given in Table 5.1.

5.2 P_n ICN Results

In the P_n ICN scenario, the rate of requests from the normal and attacker nodes are the same. The aim of this scenario was to compare the results obtained out of this experiment with a similar scenario of [27]. This scenario also proves the attack to be possible on a real ICN and thus establishes its validity.

Parameter	Values
Policy	LRU, Random, FIFO
#nodes	4, 11
Cache size	10MB, 40MB
File size	1MB
#files	400
% of unpopular files	120%
#attackers	0, 1, 2, 4, 8

Table 5.1: Parameters for both P_n and Partial Mesh ICN

As stated in Table 5.1, each of the replacement policies LRU, FIFO, and Random

were tested in this scenario. The ICN with 4 nodes was tested with 0, 1, and 2 attackers, and that with 11 nodes was tested with 0, 1, 2, 4, and 8 attackers. Additionally, going by the results of [27], the number of unpopular files each attacker can request has been fixed at 120% of the cache size in MB. Therefore with a cache size of 10MB, the attacker requests at least 12 popular files, and with a cache size of 40MB, the attacker always requests at least 48 popular files. The total number of files used in the experiments is 400.

In this section, we present the results for the following parameters as listed in Table 5.2 for illustration and discussion.

Parameter	Values
Policy	LRU
#nodes	4
Cache size	10MB, 40MB
File size	1MB
#files	400
% of unpopular files	120%
#attackers	0, 1, 2

Table 5.2: Parameters for P_4 with LRU

In Fig. 5.1, the effect of cache attack in the P_4 ICN nodes is illustrated.

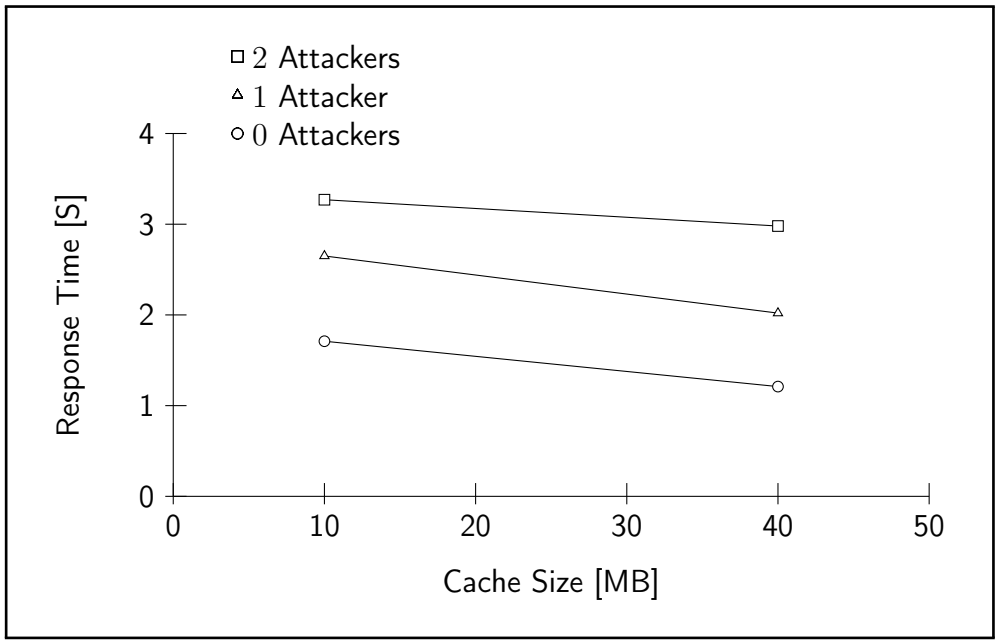

Figure 5.1: P_4 ICN with LRU

All result sets were obtained after the initial warm-up that fills the cache with random data. As is exhibited in Fig. 5.1, it is found that the impact of the cache attack appears to increase with the increase in the number of attackers, since the response time for requests increases with the increase in the number of attackers. This result is consistent with [27] for the same parameters thus validating this attack on a real ICN.

Another result of experiments with the P_{11} is also illustrated. The parameters for that is specified in Table 5.3.

In Fig. 5.2, the effect of cache attack in the P_{11} ICN is illustrated.

Parameter	Values
Policy	LRU
#nodes	11
Cache size	10MB, 40MB
File size	1MB
#files	400
% of unpopular files	120%
#attackers	0, 1, 2, 4, 8

Table 5.3: Parameters for P_{11} with LRU

Again as is exhibited in Fig. 5.2, it is found that the impact of the cache attack appears to increase with the increase in the number of attackers for the P_{11} ICN.

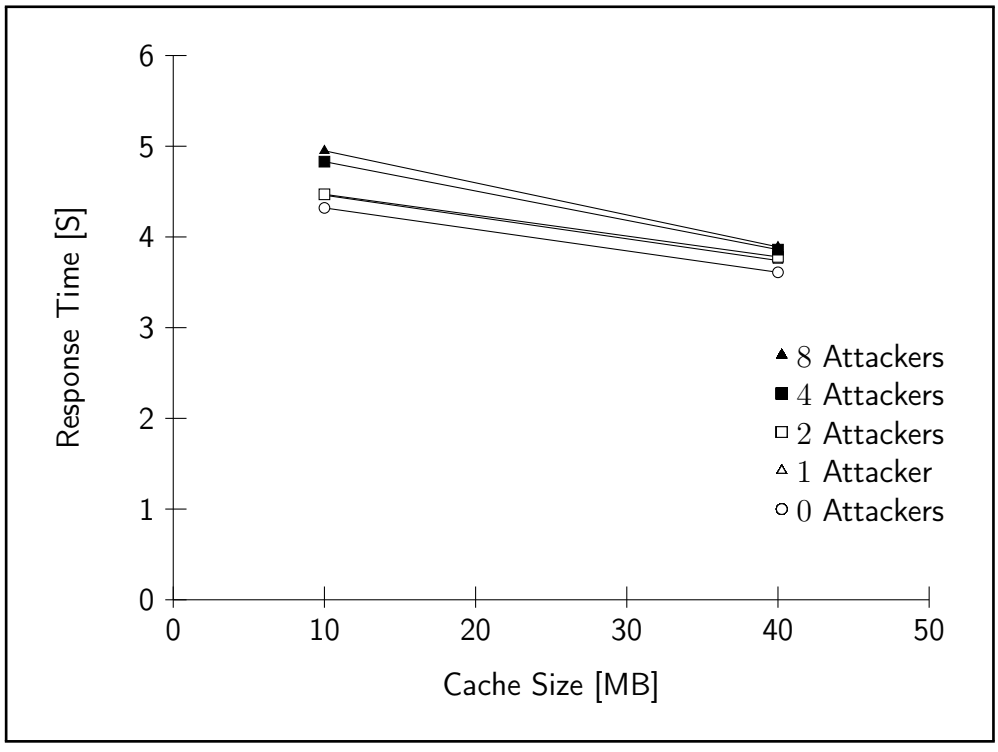

Figure 5.2: P_{11} ICN with LRU

Additional results of the P_n ICN scenario has been put in Appendix A.1.

5.3 Partial Mesh ICN Results

In this section, we present the results for the partial mesh ICN. The following parameters as listed in Table 5.4 for illustration and discussion.

Parameter	Values
Policy	LRU
#nodes	4
Cache size	10MB, 40MB
File size	1MB
#files	400
% of unpopular files	120%
#attackers	0, 1, 2

Table 5.4: Parameters for Partial Mesh ICN for 4 Nodes with LRU

In Fig. 5.3, the effect of cache attack in the P_4 ICN is illustrated.

All result sets were obtained after the initial warm-up that fills the cache with random data. As is exhibited in Fig. 5.3, it is found that the impact of the cache attack increases with the increase in the number of attackers, since the response time for requests increases with the increase in the number of attackers. This result is consistent with [27] for the same parameters thus validating this attack on a real

ICN.

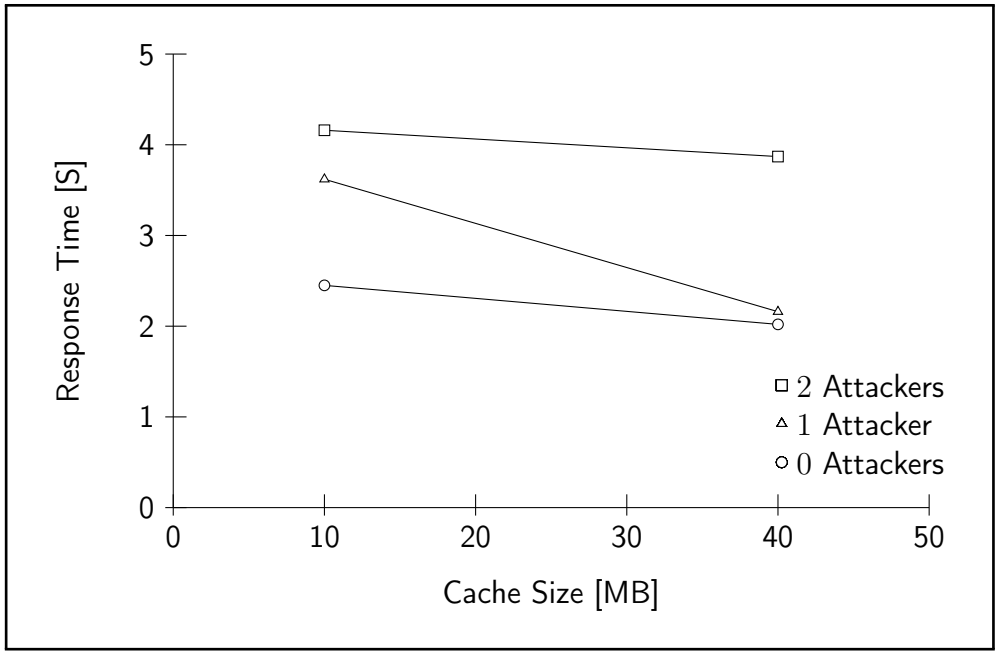

Figure 5.3: Partial Mesh ICN with 4 Nodes with LRU

Another result of experiments with the partial mesh ICN with 11 nodes is also illustrated. The parameters for that is specified in Table 5.5.

Parameter	Values
Policy	LRU
#nodes	11
Cache size	10MB, 40MB
File size	1MB
#files	400
% of unpopular files	120%
#attackers	0, 1, 2, 4, 8

Table 5.5: Parameters for Partial Mesh ICN for 11 Nodes with LRU

In Fig. 5.4, the effect of cache attack in the partial mesh ICN with 11 nodes is illustrated.

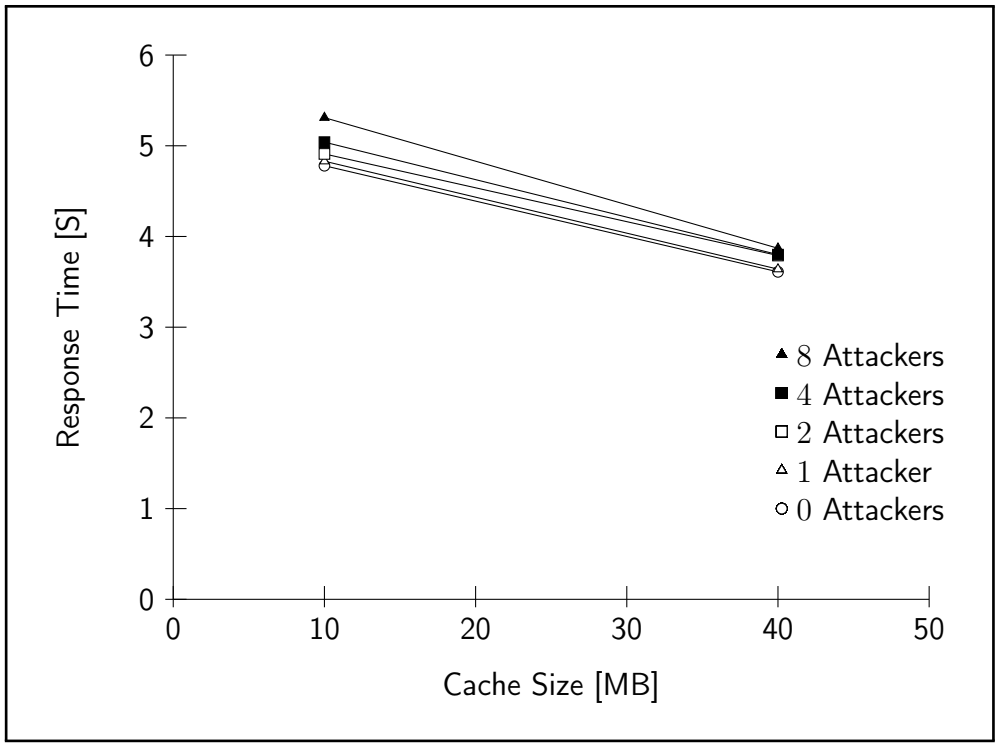

Figure 5.4: Partial Mesh ICN with 11 Nodes with LRU

Again as is exhibited in Fig. 5.2, it is found that the impact of the cache attack increases with the increase in the number of attackers for the P_{11} ICN.

Additional results of the partial mesh ICN scenario has been put in Appendix A.2.

5.4 Comparison with other Simulation Results

Fig. 5.5 shows the trends of similar experiments conducted in a simulator over a P_n ICN with comparable parameters. These trends are extremely similar to Fig. 5.1.

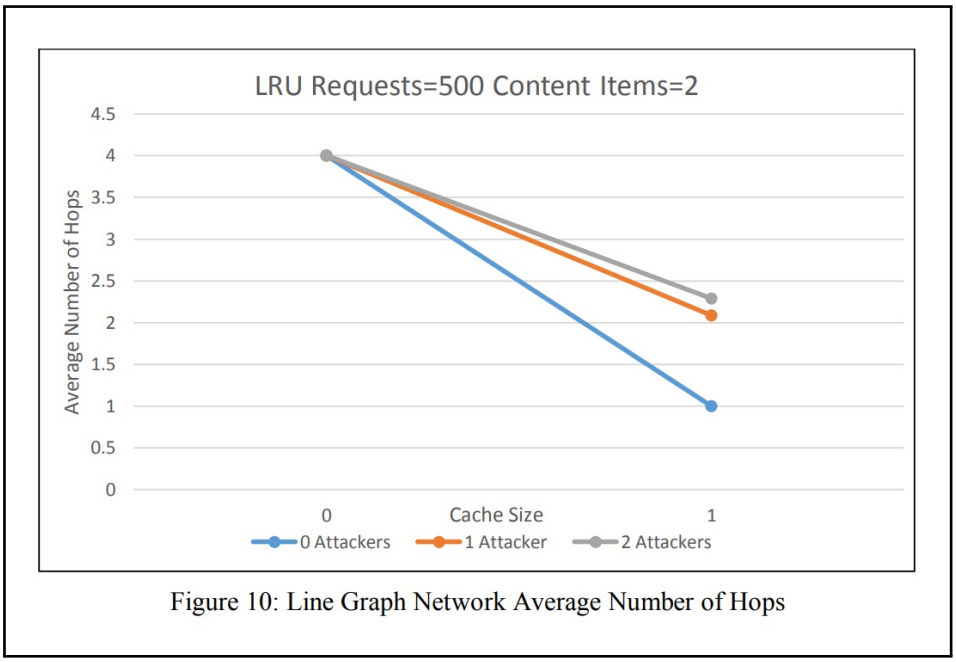

Figure 5.5: Simulation on a P_n ICN

Comparing the trends of Fig 5.5 and Fig 5.1, the difference in the beginning is due to the fact that no cache was used in the starting point by [27] for the experimental scenario of Fig. 5.5, while caches of size 10 MB were used at the starting point in this work. This validates the fact that the simulation results could be replicated in the real world, and it confirms that this cache attack is a threat for ICNs.

5.5 Comparison of Trends over Different Cache Replacement Policies

Next the effect of the cache attack has been compared for different cache policies on both kinds of ICN. In particular, Fig. 5.6 compares the effect of the cache attack on a path network topology of P_{11} with cache replacement policies LRU, FIFO, and Random. For each replacement policy, the percentage increase in average response time for requests with 0 attackers and 8 attackers respectively with cache sizes 10 MB and 40 MB have been reported in the plots. As is observed from Fig. 5.6, the effect of the cache attack is the most for FIFO, and the least for LRU replacement policy for the P_n ICN.

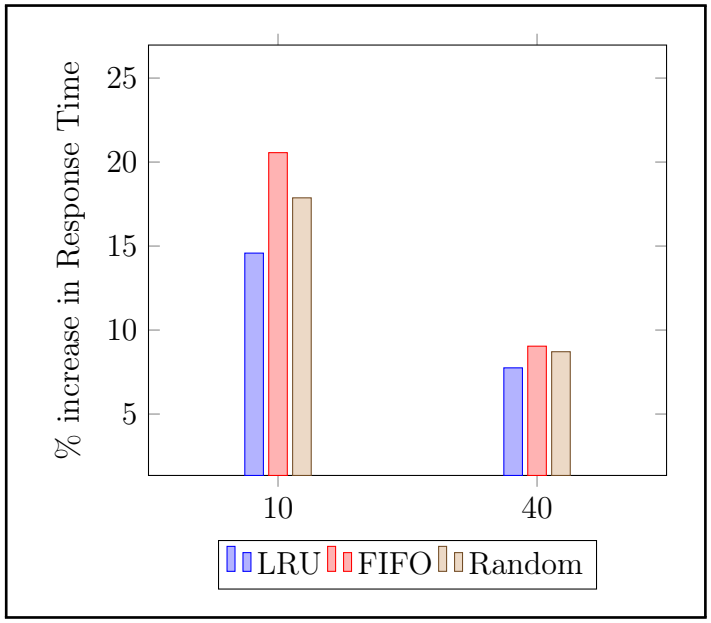

Figure 5.6: P_{11} ICN

Fig. 5.7 compares the effect of the cache attack on the partial mesh network of 11 nodes with cache replacement policies LRU, FIFO, and Random. Again, for each

replacement policy, the percentage increase in average response time for requests with 0 attackers and 8 attackers respectively with cache sizes 10 MB and 40 MB have been reported in the plots. As is observed from Fig. 5.7, the effect of the cache attack is again the most for FIFO, and the least for LRU replacement policy for the P_n ICN. But one difference here is for a cache size of 40, the difference in effect between the policies is more pronounced in the partial mesh ICN than the P_n ICN. This is probably due to the complicated topological structure of the mesh network over the line network.

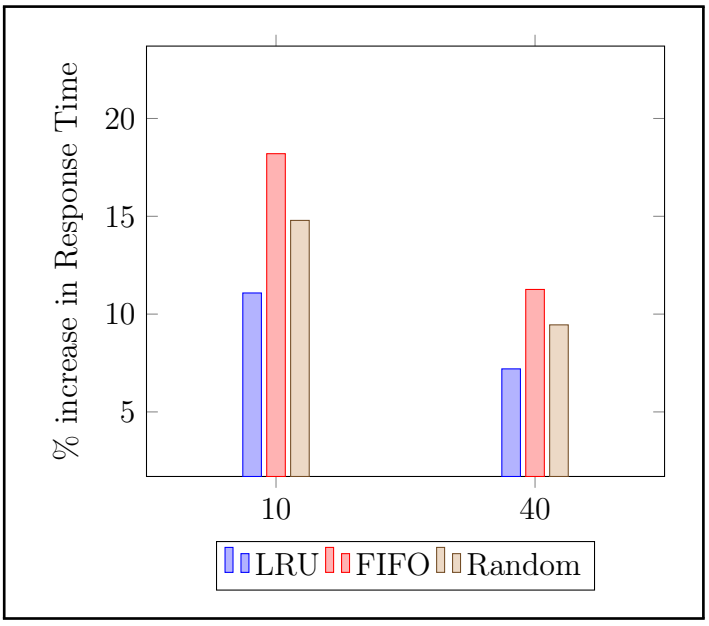

Figure 5.7: Partial Mesh ICN with 11 Nodes.

LRU has been found to be the best policy in various different network settings in different research works (e.g. [4, 9, 57]). Therefore the performance of LRU policy against the cache attack is not surprising. In [27], simulation also had shown

similar results proving LRU as the best replacement policy .

The response time in the network is an important factor that includes the transmission delay (Delay or latency is the time taken for a unit of data to be transmitted across a network link) to the destination, the processing time at both source and destination, the delay along the path, and the transmission time back to the source. The difference in the average response time has been seen in the real ICN from this experiment which were similar to the simulator results.

CHAPTER 6

Conclusions and Future Work

6.1 Results Summary

This research work focused on testing real time ICN systems to determine the results on the real network and to compare the same with the prior simulation results [27] on the effect of a cache attack on such systems (this work was mentioned as a future research direction in [27, 28]). During the experiments we discovered other factors affecting network performance which had previously gone unnoticed in simulations. The factors include network traffic, delay in packet transmission due to network bandwidth, web-based viruses, and an attack on a web server which has no backup server. These were the factors considered in this experiment which uses the laboratory networking system.

The result trends were almost the same as those in the simulations even after considering the real network factors which proves that the cache attack had high impact in the smaller network and was less effective for larger ICNs.

6.2 Implementation Challenges

To set up the laboratory network, an information centric network initially consisting of four caches was set up and preliminary results were obtained. The cache was

stored in the main memory under the assumption that the system power would never go off. Unfortunately, during a hurricane, the building lost its power and the research setup was disturbed. All the IP's also changed because by default IP addresses were dynamically assigned. The entire cache setup with the web server was rebuilt again, this time with the static IP addresses. Therefore, the recommendation to for future researchers is to use static IP addresses for the nodes during this experiment.

6.3 Future Research Directions

This research can be further extended by evaluating the performance under cache attack of larger (30 or more nodes) ICNs in the public cloud such as Microsoft Azure or Amazon Web Services. Further, the results can be repeated with the addition of smarter attacking techniques, which considers the wait time (e.g. characteristic time [16]) for each attack to flood the cache with unpopular content. In this research, since wait time was not considered even if the attacker hit the cache with zero time interval, it did not affect performance but merely raised the number of attacks on the ICN. This study extended the results of cache attacks in simulated networks to real ICNs, and will be helpful for further advancing research in real networking systems.

REFERENCES

[1] Abdallah, E. G., Hassanein, H. S., and Zulkernine, M. A survey of security attacks in information-centric networking. IEEE Communications Surveys & Tutorials 17, 3 (2015), 1441–1454.

[2] Akbaş, D., and Gümüşkaya, H. modeling and analysis of an enterprise network and its security structures. In Electrical, Electronics and Computer Engineering (ELECO), 2010 National Conference on (2010), IEEE, pp. 598–602.

[3] Akbaş, D., and Gümüşkaya, H. Real and opnet modeling and analysis of an enterprise network and its security structures. Procedia Computer Science 3 (2011), 1038–1042.

[4] Al-Zoubi, H., Milenkovic, A., and Milenkovic, M. Performance evaluation of cache replacement policies for the spec cpu2000 benchmark suite. In Proceedings of the 42nd annual Southeast regional conference (2004), ACM, pp. 267–272.

[5] Alam, F. M., McNaught, K. R., and Ringrose, T. J. A comparison of experimental designs in the development of a neural network simulation metamodel. Simulation Modelling Practice and Theory 12, 7 (2004), 559 – 578. Simulation in Operational Research.

[6] Aura, T., Roe, M., and Murdoch, S. Dynamic host configuration protocol, Aug. 7 2012. US Patent 8,239,549.

[7] Badov, M., Seetharam, A., Kurose, J., Firoiu, V., and Nanda, S. Congestion-aware caching and search in information-centric networks. In Proceedings of the 1st ACM Conference on Information-Centric Networking (2014), ACM, pp. 37–46.

[8] Bajaj, S., Breslau, L., Estrin, D., Fall, K., Floyd, S., Haldar, P., Handley, M., Helmy, A., Heidemann, J., Huang, P., et al. Improving simulation for network research.

[9] Balamash, A., and Krunz, M. An overview of web caching replacement algorithms. IEEE Communications Surveys & Tutorials 6, 2 (2004).

[10] Barish, G., and Obraczke, K. World wide web caching: Trends and techniques. IEEE Communications magazine 38, 5 (2000), 178–184.

[11] Breslau, L., Cao, P., Fan, L., Phillips, G., and Shenker, S. Web caching and zipf-like distributions: Evidence and implications. In INFOCOM'99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE (1999), vol. 1, IEEE, pp. 126–134.

[12] Breslau, L., Estrin, D., Fall, K., Floyd, S., Heidemann, J., Helmy, A., Huang, P., McCanne, S., Varadhan, K., Xu, Y., et al. Advances in network simulation. Computer 33, 5 (2000), 59–67.

[13] Cao, P., Badger, E. C., Kalbarczyk, Z. T., Iyer, R. K., Withers, A., and Slagell, A. J. Towards an unified security testbed and security analytics framework. In Proceedings of the 2015 Symposium and Bootcamp on the Science of Security (2015), ACM, p. 24.

[14] Chai, W. K., He, D., Psaras, I., and Pavlou, G. Cache "less for more" in information-centric networks. In International Conference on Research in Networking (2012), Springer, pp. 27–40.

[15] Chartrand, Gary and Zhang, Ping A first course in graph theory Courier Corporation , 2013.

[16] Che, H., Wang, Z., and Tung, Y. Analysis and design of hierarchical web caching systems. In INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE (2001), vol. 3, IEEE, pp. 1416–1424.

[17] Choi, S.-K., Yang, C.-H., and Kwak, J. System hardening and security monitoring for iot devices to mitigate iot security vulnerabilities and threats. KSII Transactions on Internet & Information Systems 12, 2 (2018).

[18] Conti, M., Gasti, P., and Teoli, M. A lightweight mechanism for detection of cache pollution attacks in named data networking. Computer Networks 57, 16 (2013), 3178–3191.

[19] DeConinck, A., Bonnie, A., Kelly, K., Sanchez, S., Martin, C., Mason, M., Brandt, J., Gentile, A., Allan, B., Agelastos, A., et al. Design and implementation of a scalable monitoring system for trinity. Proc. Cray User's Group (2016).

[20] Demetrio, L., Paolo, M., and Massimo, M. Numerical simulation of shot noise in disordered graphene2013 22nd international conference on noise and fluctuations (icnf). In ICNF2013 (2013), IEEE, pp. 1–4.

[21] Droms, R. Dynamic host configuration protocol. Tech. rep., 1997.

[22] Droms, R., Bound, J., Volz, B., Lemon, T., Perkins, C., and Carney, M. Dynamic host configuration protocol for ipv6 (dhcpv6). Tech. rep., 2003.

[23] Fan, L., Cao, P., Almeida, J., and Broder, A. Z. Summary cache: a scalable wide-area web cache sharing protocol. IEEE/ACM Transactions on Networking (TON) 8, 3 (2000), 281–293.

[24] Fayazbakhsh, S. K., Lin, Y., Tootoonchian, A., Ghodsi, A., Koponen, T., Maggs, B., Ng, K., Sekar, V., and Shenker, S. Less pain, most of the gain: Incrementally deployable icn. In ACM SIGCOMM Computer Communication Review (2013), vol. 43, ACM, pp. 147–158.

[25] Gadde, S., Chase, J., and Rabinovich, M. A taste of crispy squid. In Workshop on Internet Server Performance (1998), vol. 199, Citeseer, p. 1.

[26] Gerhards, R. rsyslog: going up from 40k messages per second to 250k. In Linux Kongress (2010).

[27] Gouge, J., Seetharam, A., and Roy, S. On the scalability and effectiveness of a cache pollution based dos attack in information centric networks. In Computing, Networking and Communications (ICNC), 2016 International Conference on (2016), IEEE, pp. 1–5.

[28] Gouge, J. B. A targeted denial of service attack on data caching networks.

[29] Haghighi, A. A., Heydari, S. S., and Shahbazpanahi, S. Dynamic qos-aware resource assignment in cloud-based content-delivery networks. IEEE Access 6 (2018), 2298–2309.

[30] Hassan, S. R., Pazardzievska, J., and Bourgeois, J. Distributed denial of service (ddos) attack detection by reducing the security alerts in grid computing networks.

[31] Jacobson, V., Smetters, D. K., Thornton, J. D., Plass, M. F., Briggs, N. H., and Braynard, R. L. Networking named content. In Proceedings of the 5th international conference on Emerging networking experiments and technologies (2009), ACM, pp. 1–12.

[32] Johnson, D. B. A note on dijkstra's shortest path algorithm. Journal of the ACM (JACM) 20, 3 (1973), 385–388.

[33] Kiskani, M. K., and Sadjadpour, H. R. Capacity of cellular networks with femtocache. In Computer Communications Workshops (INFOCOM WKSHPS), 2016 IEEE Conference on (2016), IEEE, pp. 9–14.

[34] Kiskani, M. K., and Sadjadpour, H. R. Multihop caching-aided coded multicasting for the next generation of cellular networks. IEEE Transactions on Vehicular Technology 66, 3 (2017), 2576–2585.

[35] Kiskani, M. K., and Sadjadpour, H. R. Throughput analysis of decentralized coded content caching in cellular networks. IEEE Transactions on Wireless Communications 16, 1 (2017), 663–672.

[36] Kolias, C., Kambourakis, G., Stavrou, A., and Voas, J. Ddos in the iot: Mirai and other botnets. Computer 50, 7 (2017), 80–84.

[37] Koponen, T., Chawla, M., Chun, B.-G., Ermolinskiy, A., Kim, K. H., Shenker, S., and Stoica, I. A data-oriented (and beyond) network architecture. In ACM SIGCOMM Computer Communication Review (2007), vol. 37, ACM, pp. 181–192.

[38] Kotz, D., Newport, C., Gray, R. S., Liu, J., Yuan, Y., and Elliott, C. Experimental evaluation of wireless simulation assumptions. In Proceedings of the 7th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems (2004), ACM, pp. 78–82.

[39] Mahanti, A., Williamson, C., and Eager, D. Traffic analysis of a web proxy caching hierarchy. IEEE Network 14, 3 (2000), 16–23.

[40] Matulis, P. Centralised logging with rsyslog. Canonical Technical White Paper (2009).

[41] Miller, F. P., Vandome, A. F., and McBrewster, J. Dynamic host configuration protocol.

[42] Ming, Z., Xu, M., and Wang, D. Age-based cooperative caching in information-centric networking. In Computer Communication and Networks (ICCCN), 2014 23rd International Conference on (2014), IEEE, pp. 1–8.

[43] Misra, S., Tourani, R., and Majd, N. E. Secure content delivery in information-centric networks: Design, implementation, and analyses. In Proceedings of the 3rd ACM SIGCOMM workshop on Information-centric networking (2013), ACM, pp. 73–78.

[44] Muscariello, L., Carofiglio, G., and Gallo, M. Bandwidth and storage sharing performance in information centric networking. In Proceedings of the ACM SIGCOMM workshop on Information-centric networking (2011), ACM, pp. 26–31.

[45] Nelson, M. L., McCown, F., Smith, J. A., and Klein, M. Using the web infrastructure to preserve web pages. International Journal on Digital Libraries 6, 4 (2007), 327–349.

[46] Niksi, H. Gnu wget. available from the master gnu archive site prep. ai, 1998.

[47] Niksic, H., et al. April 2005. gnu wget 1.10. Free Software Foundation.

[48] Psaras, I., Chai, W. K., and Pavlou, G. Probabilistic in-network caching for information-centric networks. In Proceedings of the second edition of the ICN workshop on Information-centric networking (2012), ACM, pp. 55–60.

[49] Saino, L., Psaras, I., and Pavlou, G. Hash-routing schemes for information centric networking. In Proceedings of the 3rd ACM SIGCOMM workshop on Information-centric networking (2013), ACM, pp. 27–32.

[50] Schulzrinne, H. Dynamic host configuration protocol (dhcpv4 and dhcpv6) option for civic addresses configuration information. Tech. rep., 2006.

[51] Sivasubramanian, S., Richardson, D. R., and Marshall, B. E. Content delivery reconciliation, Apr. 24 2018. US Patent 9,954,934.

[52] Small, D., Spencer, T., and Ou, C. Methods and apparatus to implement a partial mesh virtual private local area network service, June 3 2014. US Patent 8,743,740.

[53] Tsukahara, Y., Tomine, T., and Sugiura, K. Host information acquisition in lan environment. In Ubiquitous and Future Networks (ICUFN), 2011 Third International Conference on (2011), IEEE, pp. 284–289.

[54] Wang, W., Niyato, D., Wang, P., and Leshem, A. Decentralized caching for content delivery based on blockchain: A game theoretic perspective. arXiv preprint arXiv:1801.07604 (2018).

[55] Wang, Y., Ye, S., and Li, X. Understanding current ipv6 performance: a measurement study. In Computers and Communications, 2005. ISCC 2005. Proceedings. 10th IEEE Symposium on (2005), IEEE, pp. 71–76.

[56] Wessels, D., and Claffy, K. Icp and the squid web cache. IEEE Journal on Selected Areas in Communications 16, 3 (1998), 345–357.

[57] Wierzbicki, A., Leibowitz, N., Ripeanu, M., and Wozniak, R. Cache replacement policies revisited: The case of p2p traffic. In Cluster Computing and the Grid, 2004. CCGrid 2004. IEEE International Symposium on (2004), IEEE, pp. 182–189.

[58] Wong, W., and Nikander, P. Secure naming in information-centric networks. In Proceedings of the Re-Architecting the Internet Workshop (2010), ACM, p. 12.

[59] Xu, Q., Su, Z., Zheng, Q., Luo, M., and Dong, B. Secure content delivery with edge nodes to save caching resources for mobile users in green cities. IEEE Transactions on Industrial Informatics 14, 6 (2018), 2550–2559.

[60] Yellin, D., and Shalvi, O. Efficient content delivery over wireless networks using guaranteed prefetching at selected times-of-day, May 1 2018. US Patent 9,961,159.

[61] Yuen, W. H., and Schulzrinne, H. Wsn01-3: Improving search efficiency using bloom filters in partially connected ad hoc networks: A location-centric analysis. In Global Telecommunications Conference, 2006. GLOBECOM'06. IEEE (2006), IEEE, pp. 1–5.

[62] Yun, S., Lee, J., Chung, W., Kim, E., and Kim, S. A soft computing approach to localization in wireless sensor networks. Expert Systems with Applications 36, 4 (2009), 7552–7561.

[63] Zhang, G., Li, Y., and Lin, T. Caching in information centric networking: A survey. Computer Networks 57, 16 (2013), 3128–3141.

APPENDIX A

Additional Results

A.1 P_n ICN

Parameter	Values
Policy	FIFO
#nodes	4
Cache size	10MB, 40MB
File size	1MB
#files	400
% of unpopular files	120%
#attackers	0, 1, 2

Table A.1: Parameters for P_4 ICN with FIFO

Parameter	Values
Policy	FIFO
#nodes	11
Cache size	10MB, 40MB
File size	1MB
#files	400
% of unpopular files	120%
#attackers	0, 1, 2, 4, 8

Table A.2: Parameters for P_{11} ICN with FIFO

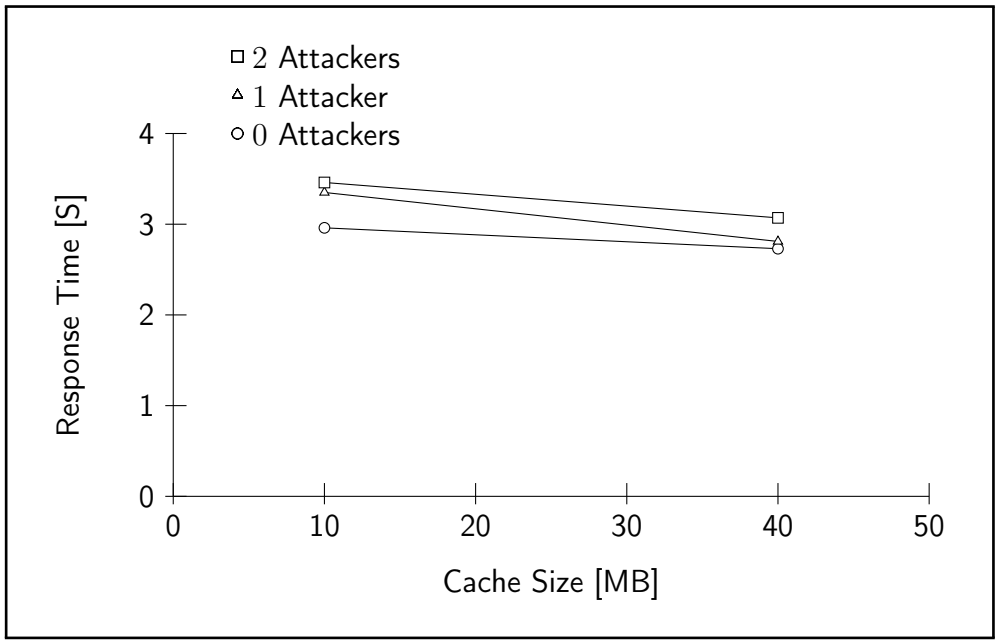

Figure A.1: P_4 with FIFO

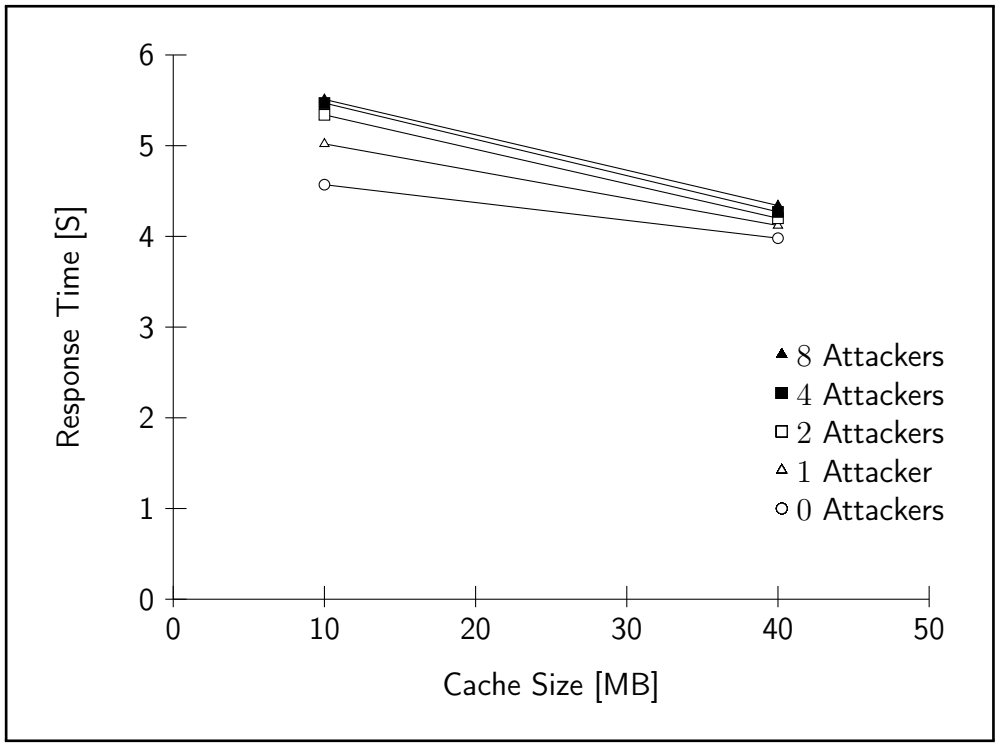

Figure A.2: P_{11} with FIFO

Parameter	Values
Policy	Random
#nodes	4
Cache size	10MB, 40MB
File size	1MB
#files	400
% of unpopular files	120%
#attackers	0, 1, 2

Table A.3: Parameters for P_4 with Random

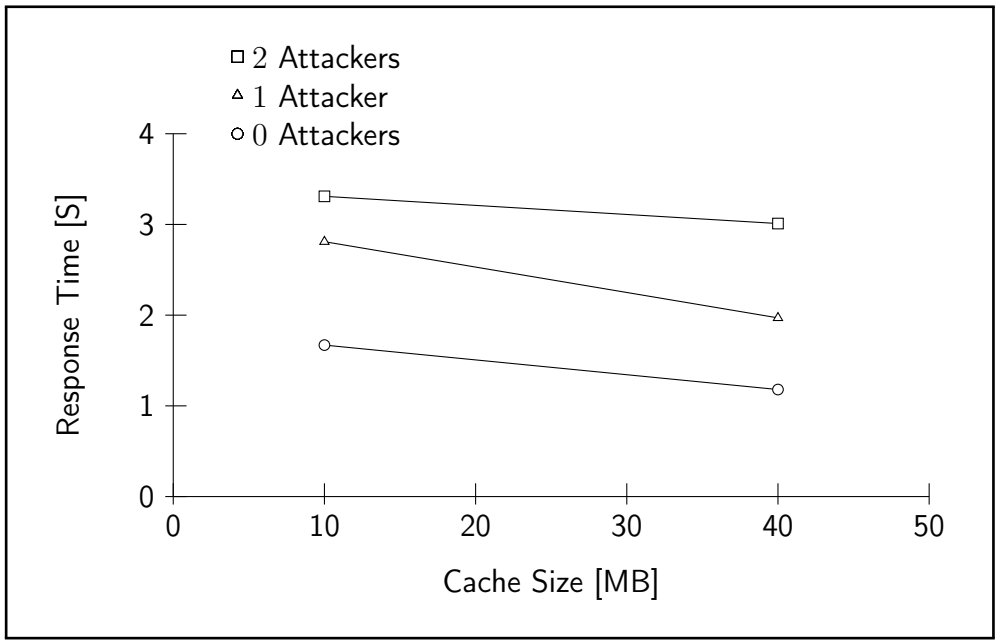

Figure A.3: P_4 with Random

Parameter	Values
Policy	Random
#nodes	11
Cache size	10MB, 40MB
File size	1MB
#files	400
% of unpopular files	120%
#attackers	0, 1, 2, 4, 8

Table A.4: Parameters for P_{11} with Random

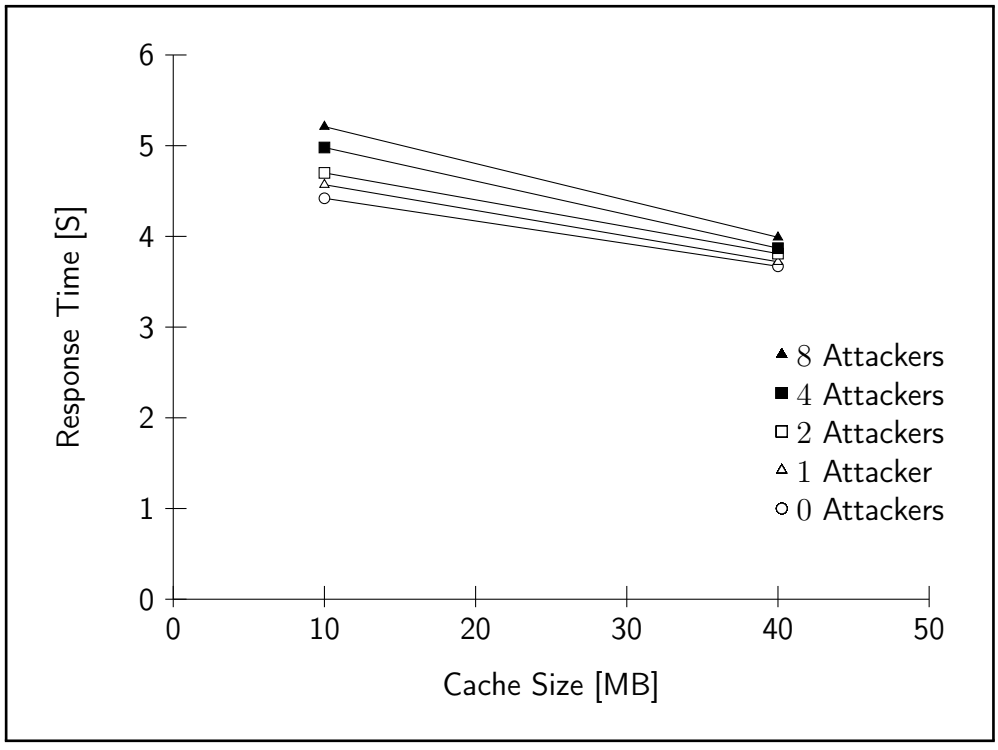

Figure A.4: P_{11} with Random

Parameter	Values
Policy	FIFO
#nodes	4
Cache size	10MB, 40MB
File size	1MB
#files	400
% of unpopular files	120%
#attackers	0, 1, 2

Table A.5: Parameters for Partial Mesh ICN for 4 nodes with FIFO

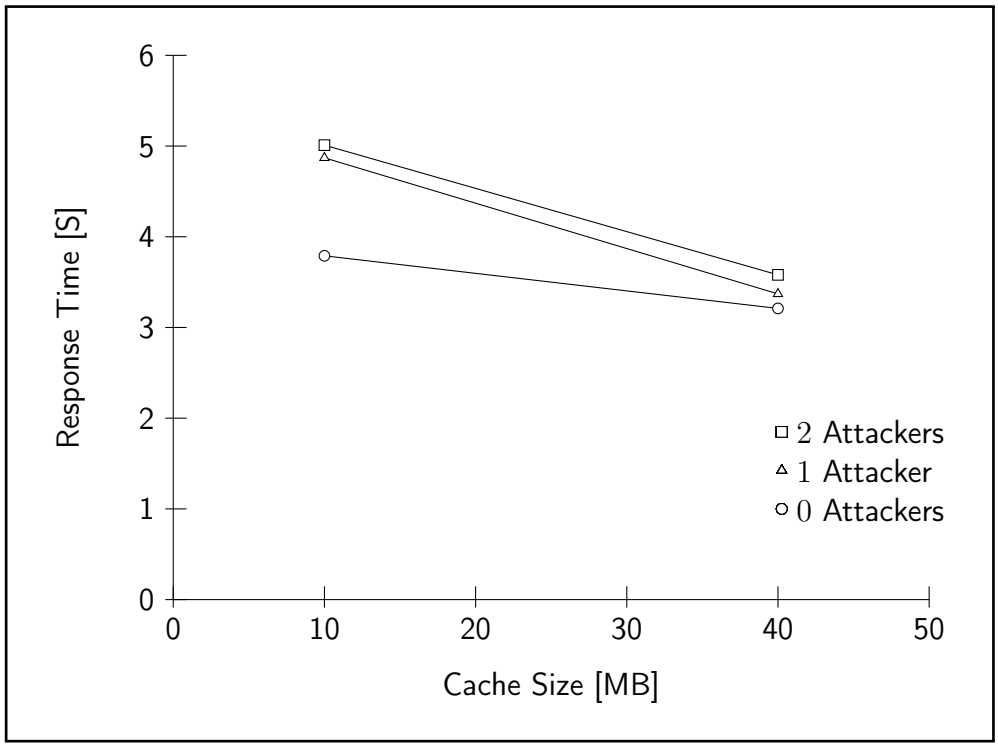

Figure A.5: Partial Mesh ICN with 4 nodes with FIFO

Parameter	Values
Policy	FIFO
#nodes	11
Cache size	10MB, 40MB
File size	1MB
#files	400
% of unpopular files	120%
#attackers	0, 1, 2, 4, 8

Table A.6: Parameters for Partial Mesh ICN for 11 nodes with FIFO

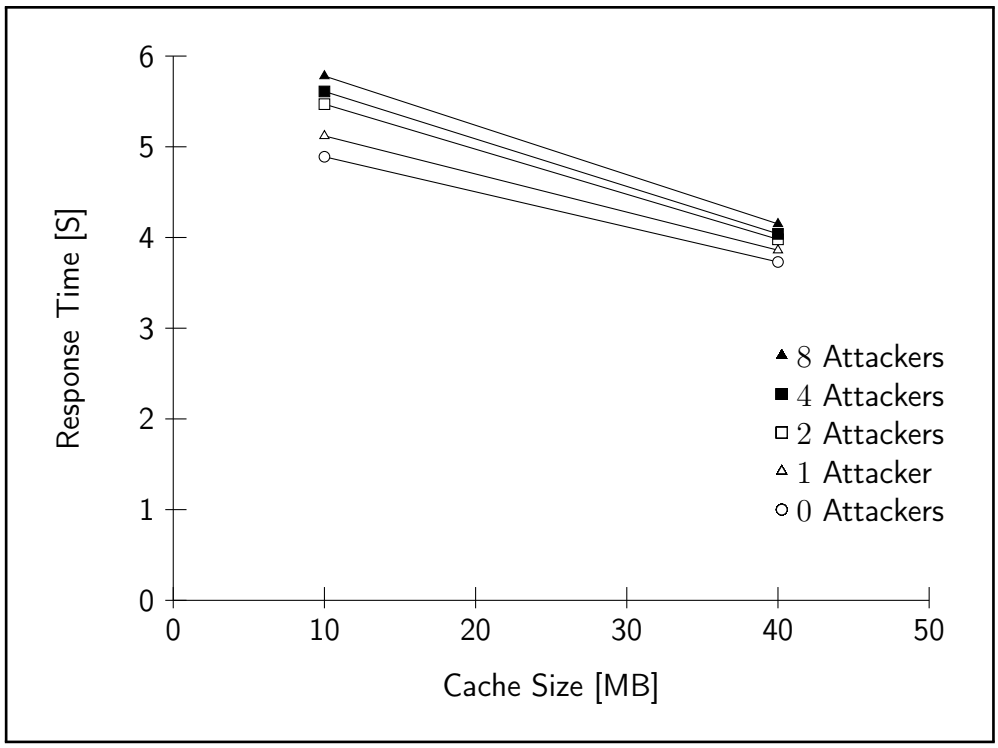

Figure A.6: Partial Mesh ICN with 11 nodes with FIFO

Parameter	Values
Policy	Random
#nodes	4
Cache size	10MB, 40MB
File size	1MB
#files	400
% of unpopular files	120%
#attackers	0, 1, 2

Table A.7: Parameters for Partial Mesh ICN for 4 nodes with Random

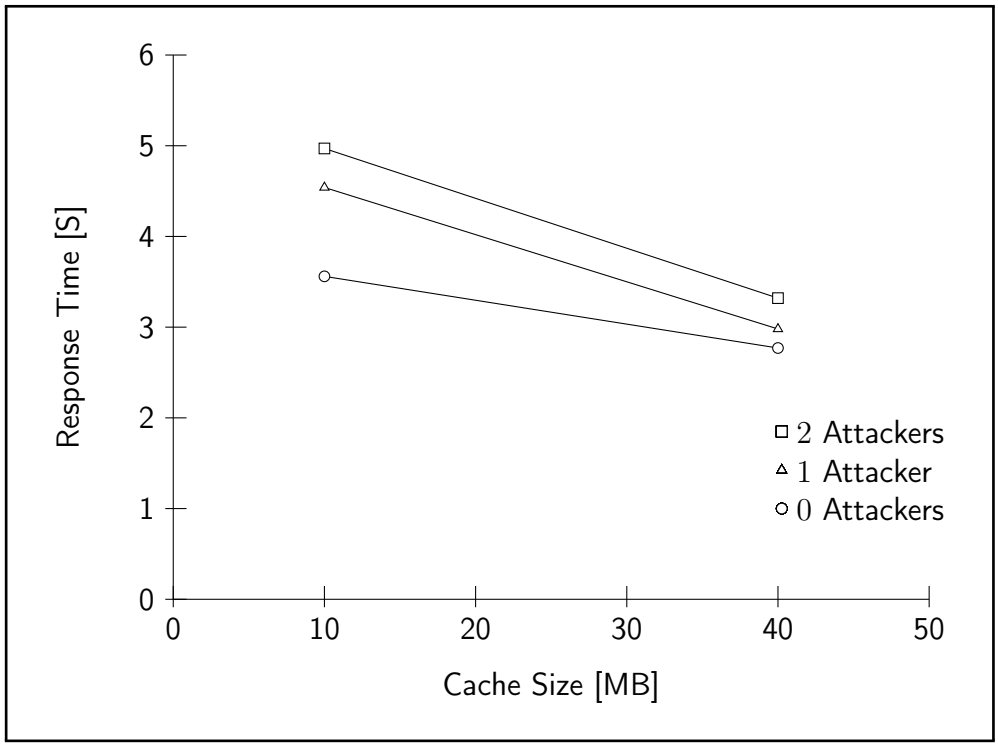

Figure A.7: Partial Mesh ICN with 4 nodes with Random

Parameter	Values
Policy	Random
#nodes	11
Cache size	10MB, 40MB
File size	1MB
#files	400
% of unpopular files	120%
#attackers	0, 1, 2, 4, 8

Table A.8: Parameters for Partial Mesh ICN for 11 nodes with Random

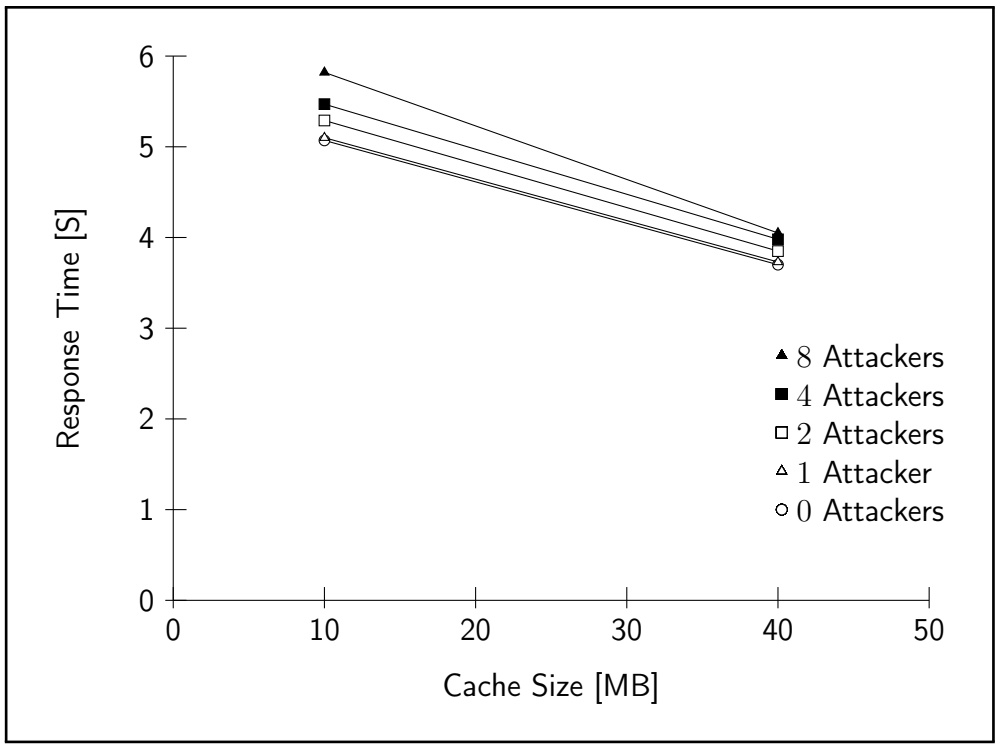

Figure A.8: Partial Mesh ICN with 11 nodes with Random

VITA

Faustina J. Anto Morais received a bachelor's degree in Information Technology from College of Engineering Guindy, Anna University in India in 2015, and aims to receive her Master of Science in Computer and Information Sciences from University of North Florida by August 2018. Dr. Swapnoneel Roy of School of Computing, University of North Florida is Faustina's thesis advisor. Faustina has been an intern in several companies during her studies and has experience working on-campus as a Graduate Teaching Assistant during her course of study. She also earned her distinction level Cyber Security and Internet of Things (IoT) courses from Massachusetts Institute of Technology. She currently works as an IT Systems Engineer for Blue Cross and Blue Shield of Florida in Jacksonville, Florida. Her expertise includes Cyber Security, IoT, Platform as a service (PaaS) automation and networking. Faustina's long-term dream is to run a women-based non-profit organization.