Master's Thesis

Title

Measurement and Analysis of Network Traffic by Considering Applications' Dynamism

Supervisor

Professor Masayuki Murata

Author

Shinya Furuta

February 16th, 2009

Department of Information Networking

Graduate School of Information Science and Technology

Osaka University

Master’s Thesis

Measurement and Analysis of Network Traffic by Considering Applications’ Dynamism

Shinya Furuta

Abstract

An analysis of Internet traffic is carried out to detect the activity of routers and to design a traffic model for understanding abnormal traffic by using typical web applications. Even if text and image data are downloaded through a parallel connection at the time of reading a Web page, this is still considered independent communication. In addition, this has not been a particular problem so far because the need for simultaneous connections was previously relatively rare.

However, in the current network environment of high-speed operation and high-quality machines, the use of a large number of simultaneous connections, aimed at shortening the response time of applications, is becoming increasingly common. For example, Google Map connects with multiple servers for a single request and acquires map data. Therefore, simultaneous connections, which have not been considered a problem so far, may cause problems in the future such as the overflow of NAT (Network Address Translation) session tables. Consequently, it is necessary to reexamine the design of the Internet by analyzing traffic in terms of the characteristics of simultaneous connections.

Therefore,in this thesis, we perform traffic analysis that considers how TCP (Transmission Control Protocol) performs simultaneous connections. How to evaluate the connection depends on the setting of each application. This investigation is performed by considering each individual application. As a result, in this paper we clarified that it is sufficient if there exist about 700 address pools in the Unified Multiplex communication architecture.

Keywords

Traffic Analysis

Simultaneous Connection

IPv6

Contents

List of Figures

List of Tables

1 Introduction

Internet traffic has been analyzed for designing the Internet itself, traffic models for simulation, session tables for routers, and methods to detect abnormal traffic of a machine infected with a virus [1–5]. These experiments assume that the TCP (Transmission Control Protocol) connection is an independent communication session, and they are based on the premise that the number of TCP connections equals the number of total sessions.

In fact, for reading pages containing both text and images,this issue has arisen with applications such as web browsers, for reading with parallel connections, and download support software having a division-capable download function with plural connections. Even still, such usages have not raised major problems because they have been fairly few in number.

However, with a network environment of higher speeds and quality and the increase in machine resources, an increasing number of applications connect to many servers for a high transmission rate and quick response. For example, Google Map with Ajax (Asynchronous JavaScript + XML) shortens reading time by staying connected to multiple servers during the acquisition of map data, and it shortens the response time by reading as it predicts the next data. For example, in Firefox, a widely used web browser, the default setting for the maximum number of connections per server was 24 in version 2, but this increased to 30 in version 3. Furthermore, although it is said that the number of sustained connections for every server with HTTP /1.1 (Hypertext Transfer Protocol) should not exceed 2 in RFC2661 [6], the rated value of the maximum number of sustained connections for every server is set to 6 in Firefox [7]. Even Internet Explorer 8 has the same rated value of 6. In this way, the number of established connections at the same time tends to increase.

The current Internet was not designed by considering that a single application might establish a large number of connections. For example, there is the problem of errors occurring when connecting at a client-side in the Thunderbird e-mail application when the server side supports IMAP (Internet Message Access Protocol) service with Courier-IMAP. This causes a problem because the established setting in Thunderbird allows a maximum 5 connections for an IMAP server, while this setting is 4 for Courier-IMAP with the same IP [8]. Consequently, as IMAP illustrates, when offered additional services, a client sets up multiple connections. This results in the server manager assuming a lower number of users that can gain access at the same time.

In addition, in IPv4, much NAPT (Network Address Port Translation) [9] is used due to using a single global IP address at multiple terminals. NAPT has a session table maintaining the information of the connection and converts this from a private address to a global address. Therefore, NAPT cannot set up more connections than determined by the size of the session table. Accordingly, communication becomes temporarily impossible when the above type of connection, assumed at the time of design, must accommodate plural hosts using applications that establish many connections at the same time.

Use of the carrier grade NAT (Network Address Translation) has been proposed as one solution to the exhaustion of available IPv4 addresses [10]. Many providers assign one global address for a single user now. The carrier-grade NAT technology saves an IPv4 address to assign as a private address to a user by installing it between a backbone and the access network of the provider. However, the number of the users' connections is limited because a single global address is used by plural users. Therefore, it is necessary to consider the characteristics of an application in order to set a large quantity of connections using the type of service suitable for a number of users accessing through a single global address.

The above situation of the current Internet is constantly evolving, and the present status is far from the situation originally assumed, where a single client uses a single connection for one server and thus communicates. Therefore it is necessary to carry out a redesign of the Internet, but there has been no research investigating how a client connects each application with the server at the same time.

Therefore,in this thesis, we clarify whether the behavior of an application influences the number of TCP connections by measuring and analyzing the change in this number over time.

Finally, we propose the Unified Multiplex communication architecture to perform communication using one IP address for one session and then discarding this IP address at the end of the session; this is achieved by using the large address space available in IPv6. In this study, we use a large quantity of IP addresses but make a quantitative decision through our results because we have not yet performed an experiment on how much consumption suffers.

In Section 2, we describe our experimental environment and the applications tested, including their behavior. In Section 3, we analyze how the behavior of the applications in Section 2 influences the number of TCP connections. In Section 4, we give a summary of Unified Multiplex and consider the implications of this report's results. The conclusion and future work are given in Section 5.

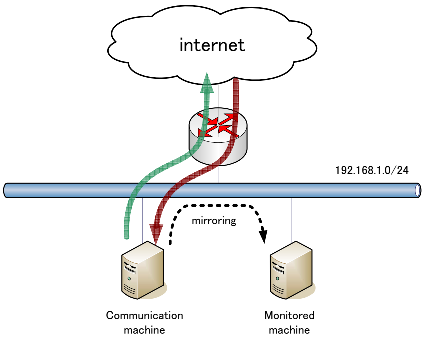


Figure 1: Network Environment

Table 1: Machines' Basic Specifications

	Communication machine	Monitoring machine
CPU	Pentium 4 3.06 GHz	Pentium 4 2.40 GHz
Memory	1.0 GB	1.0 GB
OS	Windows XP Profesional	FreeBSD Release 6.2

2 Data Gathering for Traffic Analysis

In this chapter, we explain the data acquisition method for performing traffic analysis with individual applications. Subsection 2.1 first describes our experimental environment, and subsection 2.2 overviews the studied applications. Finally, subsection 2.3 explains how the data were collected.

2.1 Experiment Environment

While operating an application as the object of study, the network environment has a machine generating traffic connected to a machine observing traffic as shown in Figure1. The observation performs mirroring at the port of a switching hub connecting the two machines and records the traffic values. The environment of the communication and monitoring machines is shown in Table 1.

2.2 Target Applications

There is now a very wide diversity of equipment and services deployed on the Internet, so it would be nearly impossible to cover all of them here. Therefore, this experiment investigates a representative set of services typically used by many Internet users. According to previous findings [11], such typical Internet usages include e-mail, information retrieval, acquisition of such information as current news, purchases of merchandise and services, and downloading of animation, images and so on. Therefore, this report investigates the performances of simultaneous connections with a Web browser, an e-mail client application, P2P file-sharing software, an instant messenger, and animation and jukebox software. Our method is to examine how traffic changes for a user action after defining the user actions for a particular application. Here, we insert an interval of approximately 1 minute between these user actions. Nevertheless, such controlled operation does approximate actual operation of these applications.

2.2.1 Web Browser

Because a Web browser connects with a server at random, the sustained HTTP 1.1 connection is specified in RFC2616 so that the burden on the server is not too large, and thus the number of connections is limited to 2 at maximum [6]. This setting depends on the application. In the case of Firefox 3, the maximum number of connections is 15 for every HTTP 1.0 server, 6 for every HTTP 1.1 server; for IE (Internet Explorer) 6, this number is 4 for every HTTP 1.0 server and 2 for every HTTP 1.1 server. In addition, a ”keep-alive” function is used to ensure that communication is performed regularly by confirming that a sustained connection is effective. When a keep-alive packet does not arrive, even when a no-communication state continues beyond a constant time, this function will cut the connection. This cut-off time is set depending on the implementation of the application; for example, it is 300 seconds in Firefox 3 and 60 seconds in IE 6.

While considering the difference in the implementation mentioned above, the observation of network traffic uses two kinds each of Firefox and Internet Explorer as Web browsers during use in Web Service. There are various Web services, but here we conduct an investigation into Google Map, Amazon, and YouTube. In the following, we show the steps taken in these observations for each application.

In Google Map the user performs these steps:

(1) Display Google Map (`http://maps.google.co.jp/maps`),

(2) Search for ”Osaka,”

(3) Display a street view of Hanshin Expressway Route 11,

(4) Click a direction arrow toward the east,

(5) Close.

In Amazon, the user performs the following steps:

(1) Access the top page of Amazon (`http://www.amazon.co.jp/`),

(2) Access books / comics / magazines → Japanese books,

(3) Access ”see details of Top 100 Japanese books,”

(4) Click the first displayed book,

(5) Close.

In YouTube the user performs these steps:

(1) Access the top page of YouTube (`http://www.youtube.com/`),

(2) Search for ”Osaka,”

(3) Try to play the animation (length of the animation is 2:53),

(4) Close.

2.2.2 e-Mail Client Applications

Our traffic analysis also examines the state when Thunderbird and Outlook are used as e-mail client applications. Two accounts are set as shown in Table . In this table, POP3 is Post Office Protocol, SMTP is Simple Mail Transfer Protocol, and SSL is Secure Sockets Layer.

The steps used for the e-mail clients are as follows:

(1) Receive the email of account 1,

Table 2: Account Settings

	recieve protocol	send protocol	recieve server	send server
No.1	IMAP over SSL	SMTP over SSL	mailsrv.nal.ics.es.osaka-u.ac.jp	mailsrv.nal.ics.es.osaka-u.ac.jp
No.2	POP3 over SSL	SMTP over SSL	POP3.gmail.com	smtp.gmail.com

(2) Choose the e-mail folders of account 1 sequentially and receive a message (in folder access, takes about 15 seconds, and choose 10 folders in total),

(3) Receive a message in account 2,

(4) Transmit the e-mail with account 1,

(5) Transmit the e-mail with account 1 again,

(6) Close the e-mail client application.

The e-mail client is used to analyze how traffic occurs for the above user actions.

2.2.3 P2P file-sharing software

We now consider the time needed to use BitTorrent as P2P file-sharing software. A notable characteristic of BitTorrent is that all users are involved in the total file distribution according to the rule of having to transmit part of one's file in order to receive a part of a file from a partner. In BitTorrent, the user performs the following actions:

(1) Click a Torrent file and start a download,

(2) Finish downloading the file,

(3) Keep the application running for a while,

(4) Quit BitTorrent.

A Torrent file is a file including the link to the IP address of a machine that is exchanging the desired data. FreeBSD7.1-Release-i386-bootonly, which can be acquired at `http://torrents.freebsd.org:8080/`, is used in this report.

2.2.4 Instant Messenger

We used Windows Live Messenger as an instant messenger. Window Live Messenger is an application that has many functions such as audio communication and file sharing, but in this report we observe traffic only in the case of using text chat.

A text chat is generally performed as (1) The starting client sends the request to start the chat with an address to a server, (2) A demand is forwarded to the other client, (3) The chat starts. More concretely, the user follows these steps:

(1) Enter the signature,

(2) Transmit a message to the client at the address (exchanged five times every 15 seconds),

(3) Keep a conversation window open for 10 minutes,

(4) Close the conversation window but keep it available for 10 minutes,

(5) Close Windows Live messenger.

2.2.5 Music/Movie Player Application

Recent music/movie player applications possess not only functions for sound streaming and photo browsing but also the capability to access online stores or net-based radio stations. The online store shows the cover-jacket images of a large number of albums or audio/video files, and it allows the user to download many files in one access session. In this report, we use the popular iTunes Store. In this case, a user performs these steps:

(1) Access the iTunes Store,

(2) Access the Music section,

(3) Access the ”Top Albums,”

(4) Click the first displayed album,

(5) Try to play it,

(6) Quit iTunes.

3 Analysis of The Network Traffic

In this chapter, we analyze the network traffic data collected in section 2. In subsection 3.1, we analyze the results of measuring the network traffic of particular clients during one day. In section 3.2, detailed analysis is given for traffic while operating every application.

3.1 Summary of Gathered Traffic Data

A summary of traffic data gathered in the analyses of Section 2 is shown in Tables and . There are data for every application, and the total number of connections is counted by the number of SYN/ACK packets. However, redundant data are omitted when the address of the origin of a message transmission, the address of the next message transmission, the port number of the origin, and the port number of the next transmission are the same. The other items are represented using the summary function of Wireshark.

3.2 Method of Traffic Analysis

In this section, we explain our analysis method for collecting traffic data. In this report, we analyze the following four items:

(1) Transition of the number of connections by time

(2) Transition of the number of requests for the connection by time

(3) Distribution of the connected time of the connection

(4) Distribution of the request intervals of the connection

The ”number of connections” is considered the number of connections required to establish a TCP state. This implies that a TCP state is ESTABLISHED at the state when a FIN packet or an RST packet is received or before a message is transmitted after the transmission of a SYN packet, after a SYN/ACK packet corresponding to the SYN packet is received. This is used to determine how an action of the user application affects the number of connections at the same time. In other words, this assists our investigation by showing how this number changes over time. The connect time of these connections is from the reception time for a SYN/ACK packet to the time a FIN or RST packet is received or transmitted as a message. By examining the connect

Table 3: Network Trace Data Used in Study

Dataset	Duration (seconds)	Number of Packets
Firefox-Google Map	270.740	4136
IE-Google Map	270.092	1329
Firefox-Amazon	255.628	2457
IE-Amazon	270.858	939
Firefox-Youtube	466.371	9481
IE-Youtube	402.448	9013
Thunderbird	479.043	804
Outlook	415.951	740
Live Messenger	1425.754	1212
iTunes	356.135	3548
BitTorrent	17472.038	303370
ALL	23250.815	223342

Table 4: Trace Data Details

Dataset	Number of Connections	Overall Transfer Data (bytes)	Mean Data Rate (bytes/sec)	Mean Packets Size (bytes)
Firefox-Google Map	67	3126015	11546.192	15.277
IE-Google Map	55	844533	3126.836	4.921
Firefox-Amazon	55	1568843	6137.245	638.254
IE-Amazon	33	536696	1959.313	565.171
Firefox-Youtube	52	8585174	18408.426	905.511
IE-Youtube	29	8397473	20866.000	931.707
Thunderbird	11	276658	577.523	344.102
Outlook	6	303680	730.087	410.378
Live Messenger	60	463623	325.177	382.527
iTunes	13	3054236	8550.784	858.297
BitTorrent	8270	1439427012	8238.461	474.479
ALL	5696	136021480	5850.181	609.028

time of every connection, we can determine whether an application requires a short or a long time to acquire information for a user action. There is a request interval for the connection every time a SYN packet of the ESTABLISHED connection is sent. This is used to investigate the ratio of connections being connected at the same time.

3.2.1 Traffic Analysis of Web Browser

Here, we analyze the traffic visiting sites such as Google Map, Amazon, and Youtube using Firefox and IE.

Google Map

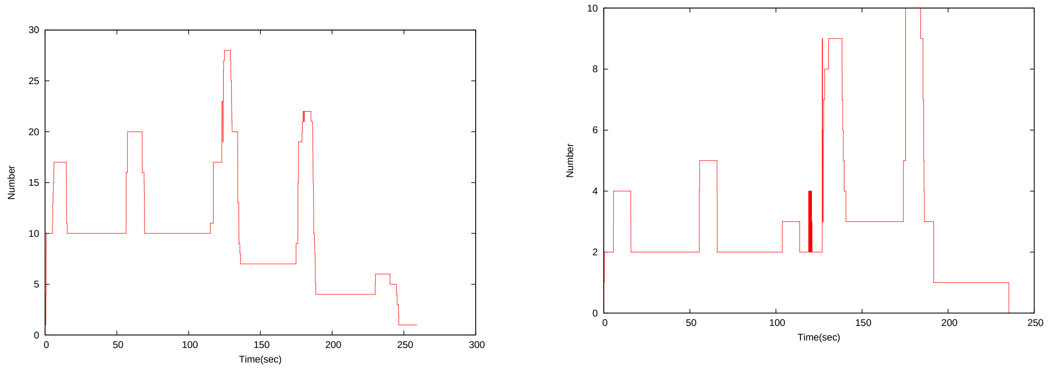
Figures 2 to 5 show the number of simultaneous connections, the number of requests, the distribution of connect time, and the distribution of request intervals when using Google Map.

First, let's consider the traffic characteristics when we visit Google Map. When the top page of Google Map is displayed, the Web browser establishes connections to the server that maintains a search screen and map data. For quickly updating map data by the user's action, it persistently connects for a relatively long time (keep-alive timeout: about 120 seconds) to reduce the overhead of establishing a connection. It finishes the connections used to load data for such items as scroll bars or photos of searched objects in a shorter time (Keep-alive timeout is about 20 seconds), since such data do not demand as quick a response as map data. The reason why a time lag of about 120 seconds occurs after a connection request is that time is needed to display a thumbnail after the street view finds a candidate location.

Figures 2 and 3 show the number of connections and the number of requests for a connection when we visit Google Map. Firefox differs from IE in that the numbers of maximum persistent connections per server are 10 and 2, respectively.

The user performs the action about every 60 seconds, so a connection request is also transmitted about every 60 seconds. The number of connections has four peaks roughly corresponding to the number of actions. Figure 4 shows cumulative distributions of connect time when Google Map is being viewed. Connect time can be classified into three groups: for 20 seconds or more, within 0 20 seconds, and for nearly 0 seconds.

Figure 5 shows cumulative distribution of connection-request intervals when Google Map is



(a) Google Map's Network Traffic (Firefox)

(b) Google Map's Network Traffic (Internet Explorer)

Figure 2: Number of Established Connections (Google Map)

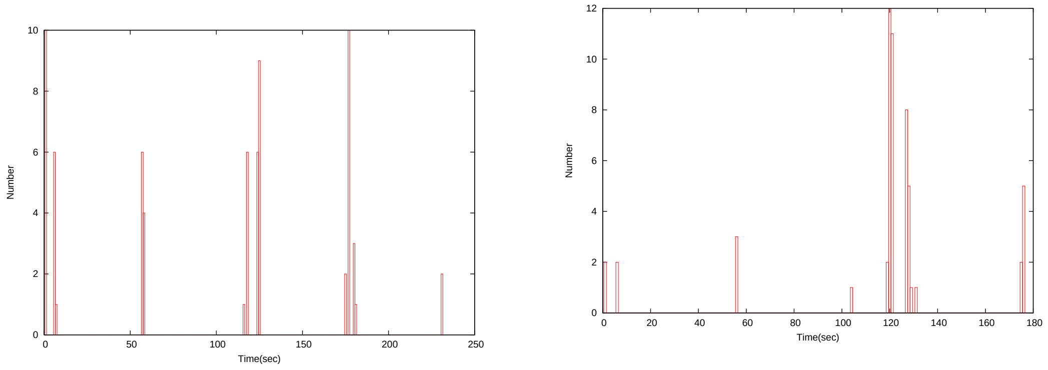
being viewed. A connection rate of 80 percent or more implies nearly simultaneous requests. Figure 3(a) shows that the timing is just behind the user's action, at just a slight time lag. Since this site uses Ajax, there are connections after acquiring an XML file that are described as server addresses.

Amazon

Figures 6 to 9 show the number of simultaneous connections, the number of requests, distribution of connect time, and distribution of request intervals when using Amazon's web site.

Various rankings and a picture of the products are published on Amazon's top page. Because the connection of each element is set, it can be confirmed that a large number of requests are transmitted in figure7. Amazon immediately performs time-out of the connection for the static contents such as the menu or logo and then connects to the ranking on the server for a long time.

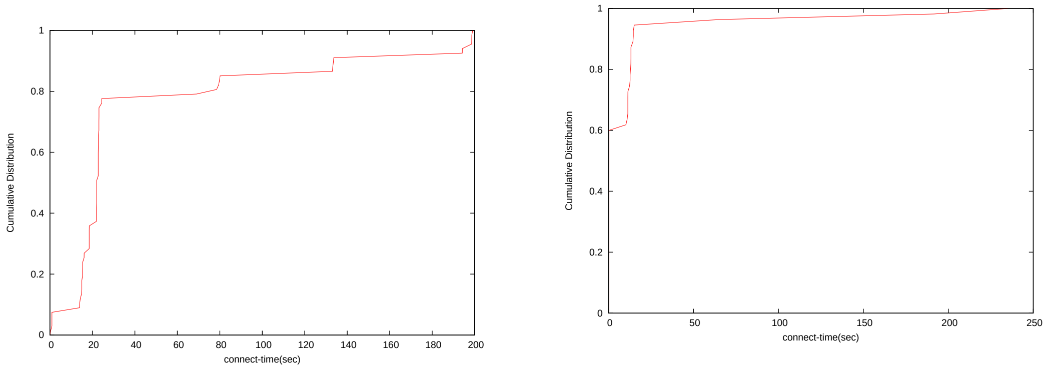
While continuous connection is maintained in figure 6(a), a number of simultaneous connections are being made during a period of about 60 seconds in figure 6(b). This is caused by the persistent connection of IE carrying out a timeout within the 60 seconds. Therefore, figure 8 shows that a connection for 60 seconds or less is almost the same case. The connection group at about 40 seconds comes from the time-out of a server storing static contents.



(a) Google Map’s Network Traffic (Firefox)

(b) Google Map’s Network Traffic (Internet Explorer)

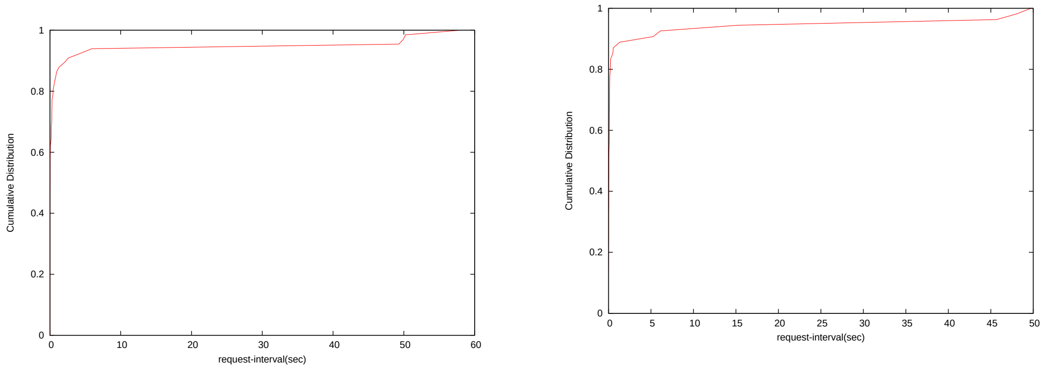
Figure 3: Number of Requests (Google Map)



(a) Google Map’s Network Traffic (Firefox)

(b) Google Map’s Network Traffic (Internet Explorer)

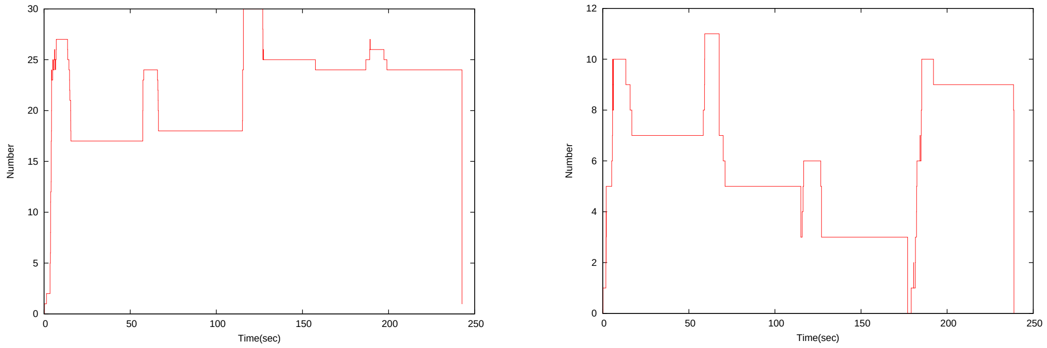
Figure 4: Distribution of Connect Time



(a) Google Map's Network Traffic (Firefox)

(b) Google Map's Network Traffic (Internet Explorer)

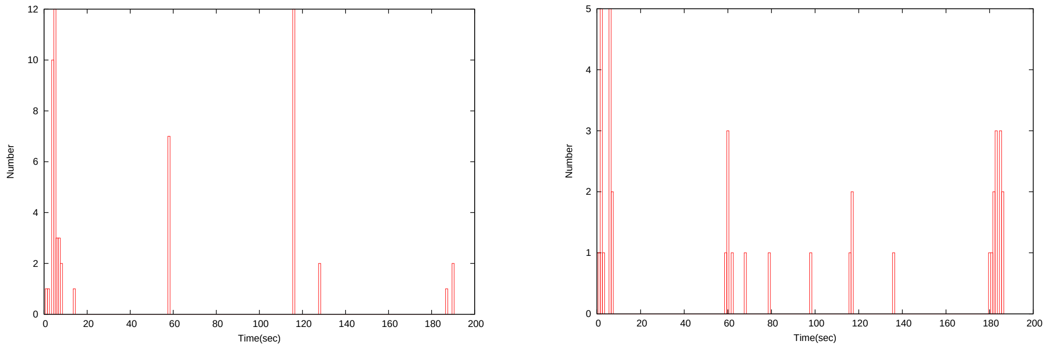
Figure 5: Distribution of Request Intervals



(a) Firefox

(b) Internet Explorer

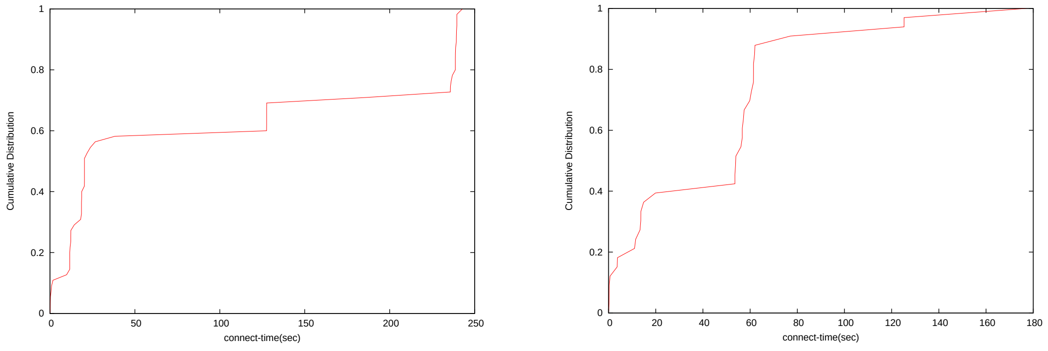
Figure 6: Number of Established Connections (Amazon)



(a) Firefox

(b) Internet Explorer

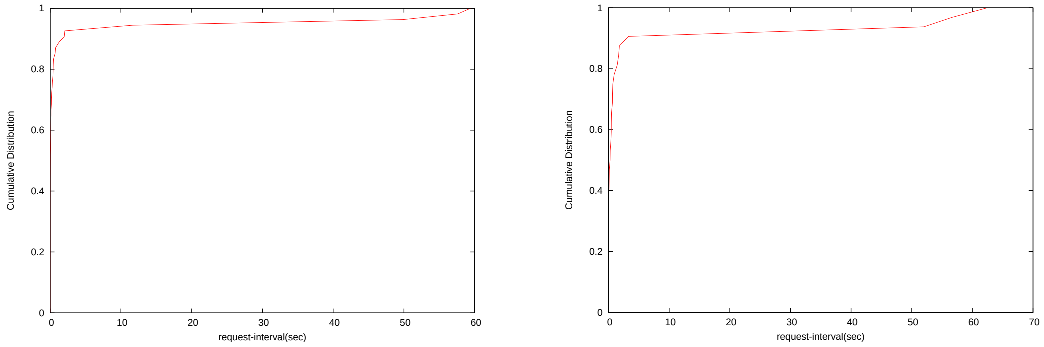
Figure 7: Number of Requests (Amazon)



(a) Firefox

(b) Internet Explorer

Figure 8: Distribution of Connect Time (Amazon)



(a) Firefox

(b) Internet Explorer

Figure 9: Distribution of Request Intervals (Amazon)

Youtube

Figures 10 to 13 show the number of simultaneous connections, the number of requests, distribution of connection time, and distribution of request intervals when the Youtube Web site is used.

The top page of Youtube consists of video thumnails and an introduction of the video that is currently viewed. The introduction of the video changes after a certain period, and the connection is set up for every selection. Watching a video sets up a single connection, and the connection is terminated when the video is finished. Other videos are recommended after the initial video is finished. A connection is then set up to display the thumbnail of the recommended video (260 and 300 seconds in Figure 11).

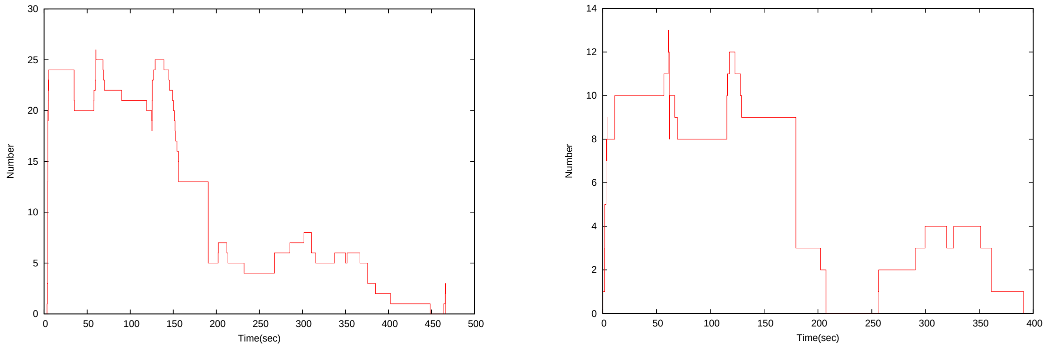
Figure 12 does not show a characteristic like Google Map and Amazon. Table shows the average and the variance of connection time when connecting to each server. Here, it is understood that the difference at connect time originates in the connected server, although the variance of connect time to 66.249.89.118 is large. The connect time to 66.249.89.118 varies among times of 80 seconds, about 120 seconds, and about 200 seconds, and this difference is caused by the packet being transmitted before persistent connection time out and thus when the timer is reset.

3.2.2 Traffic Analysis of Mail Client Applications

Figure 14 shows the traffic characteristics when the Thunderbird e-mail client application is used. In Figure 14(a), the number of connections continues to increase up to five for 60-120 seconds

Table 5: Statistics of the Connect Time for Every Server of Youtube (Firefox)

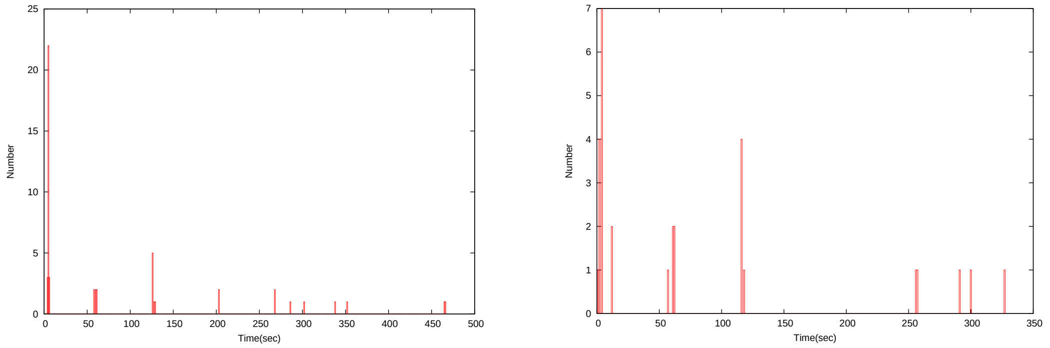
IP Address	Number of connection	average(seconds)	variance
60.254.154.75	2	305.41165	0.001299602
66.249.89.118	23	144.7383304	1653.967444
74.125.14.24	7	36.49555714	3.53854716
74.125.14.28	3	35.77323333	9.786968542
209.85.175.102	2	12.68815	0.002953922
210.153.90.116	5	0.21562	0.007994982



(a) Firefox

(b) Internet Explorer

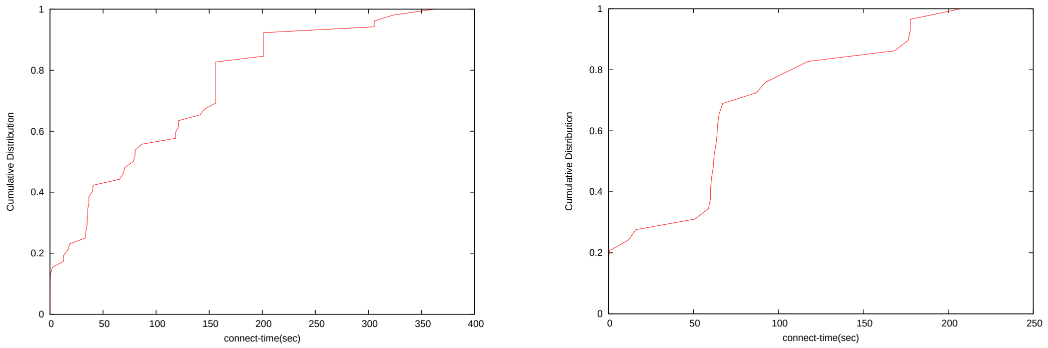
Figure 10: Number of Established Connections (Youtube)



(a) Firefox

(b) Internet Explorer

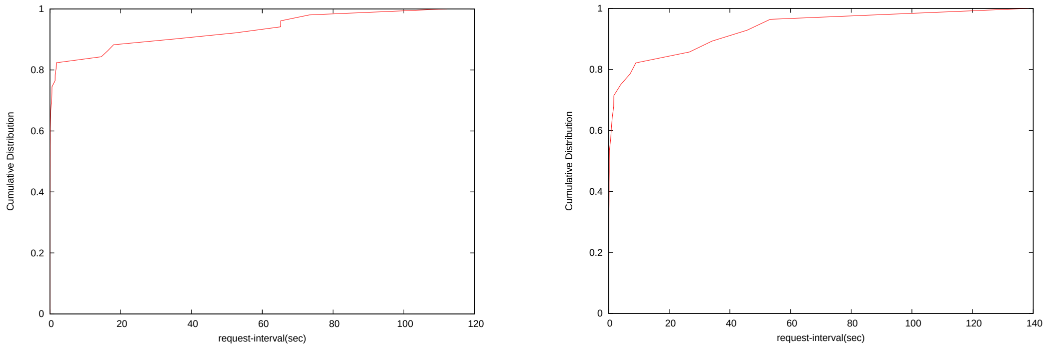
Figure 11: Number of Requests (Youtube)



(a) Firefox

(b) Internet Explorer

Figure 12: Distribution of Connection Time (Youtube)



(a) Youtube’s Traffic (Firefox)

(b) Youtube’s Traffic (Internet Explorer)

Figure 13: Distribution of Request Intervals (Youtube)

after receiving the e-mail in account 1 . Thunderbird sets up a single connection for each folder of IMAP, but the maximum number of connections per server is restricted to five. Thunderbird increases to three connections at about 250 seconds. E-mail is received by POP3, and the remaining two HTTP connections display the start page of Thunderbird. When the certification and download of the message are over, the connection is finished. Thunderbird connects with a server periodically thereafter to check for messages. SMTP certifies and sends messages in the same way as POP3 does.

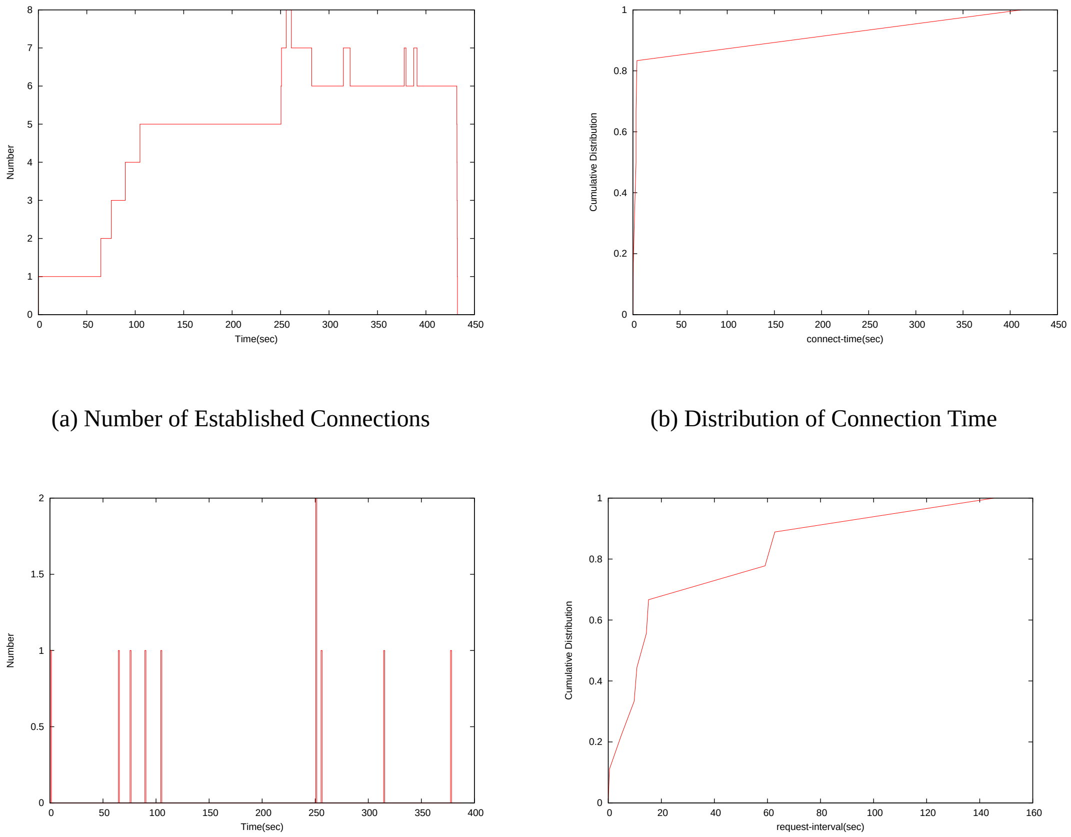
Figure 15 shows the appearance of the traffic when Outlook is used. The IMAP connection continues maintaining a single connection per server with Outlook whereas Thunderbird connects every folder in an IMAP connection. In Figure 15(a), Outlook continues to maintain a single connection per server, whereas Thunderbird sets up a single connection per folder. Outlook connects for the download of the message separately from the certification connection and terminates the connection when a download is completed. Outlook's SMTP protocol performs certification and sending of messages in another connection like POP3.

3.2.3 Traffic Analysis of Instant Messenger

Figures 16(a) to 16(f) show the number of simultaneous connections, the number of requests, distribution of connect time, and distribution of request intervals when we use Windows Live Messenger. In addition, Figure 16(c) shows the number of simultaneous connections for 0 to 120 seconds, and Figure 16(d) shows the number of requests for 0 to 120 seconds.

First, when the Messenger is started, a connection is made for certification of the user. Next, the application always connects to a Microsoft server to grasp the state of the client. It accesses a server having all of the contents and acquires group-constitution information and a list of members, the advertisement contents of the account, and the expression function contents tab provided by the advertisement sponsor. It then connects to a Microsoft server when the user starts a conversation with a member and exchanges messages through the server. This connection is not related to the state of the conversation window of the partner. A connection is started when a conversation window is opened and ends when it is closed.

In Figure 16(a), the messenger sets up a connection regularly without the user actually initiating an action. These connections acquire an advertisement and deliver the news displayed by the lower part of the messenger. Figure 16(f) shows that connections always begin at 270 seconds.



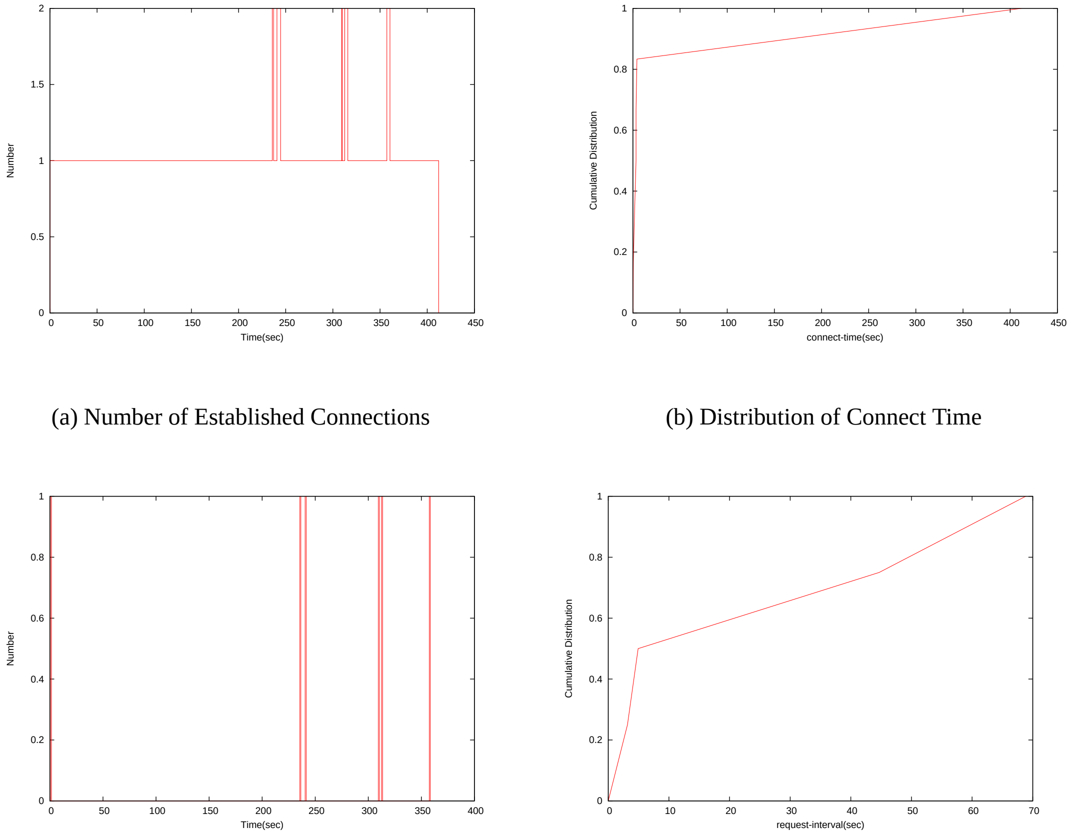
(a) Number of Established Connections

(b) Distribution of Connection Time

(c) Number of Requests

(d) Distribution of Request Intervals

Figure 14: Thunderbird's Network Traffic



(a) Number of Established Connections

(b) Distribution of Connect Time

(c) Number of Requests

(d) Distribution of Request Intervals

Figure 15: Outlook’s Network Traffic

Figure 16(e) shows that a short connection of one second or less exists in 20% of the cases and a connection exists for about 60 seconds in about 80% of the cases. Connections to the contents delivery server are performed by persistent connections of HTTP/1.1, and this connection continues until time-out.

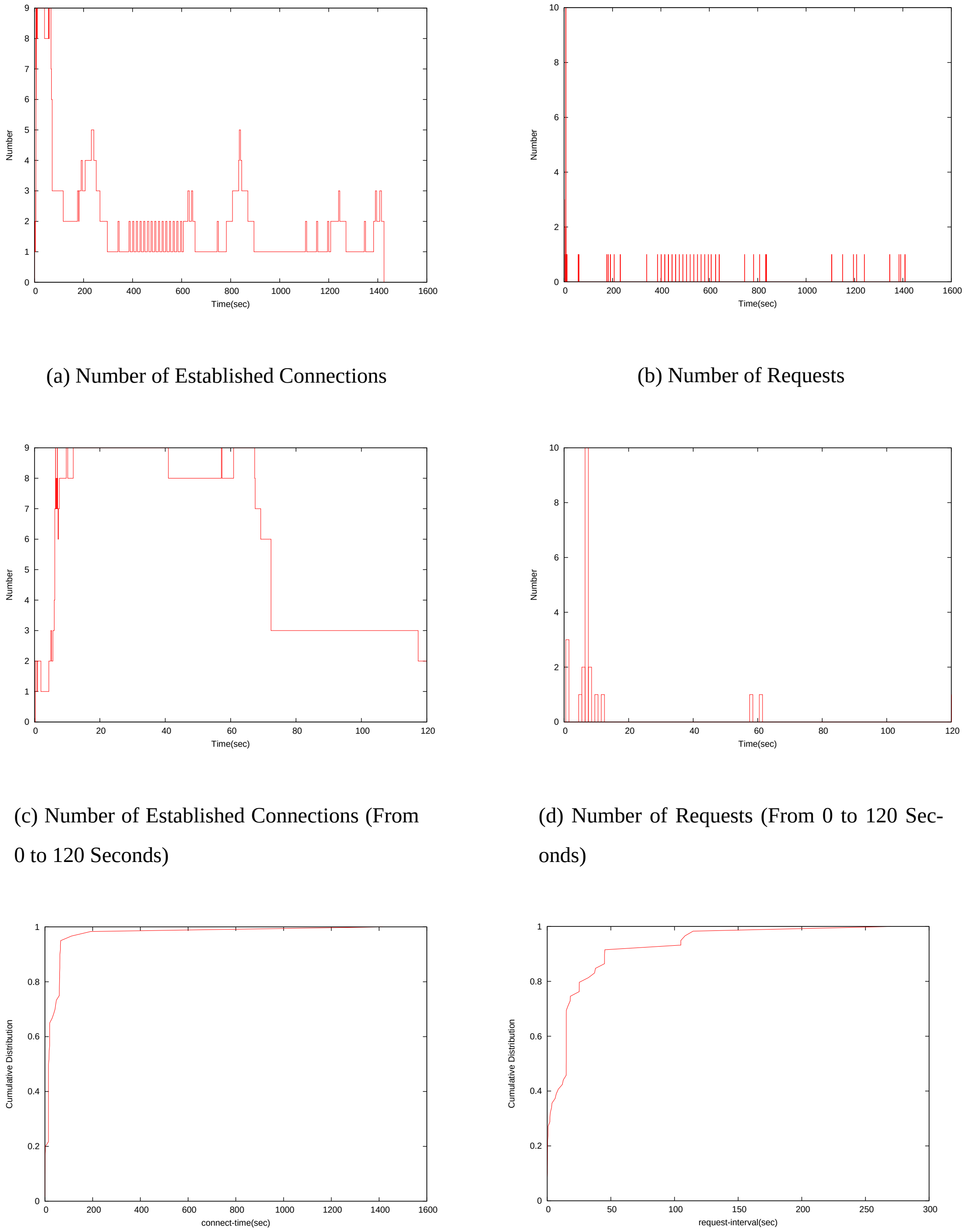
3.2.4 Traffic Analysis of P2P File-Sharing Applications

Figures 17(a) to 17(d) show the number of simultaneous connections, the number of requests, distribution of connect time, and distribution of request intervals when using BitTorrent.

The file downloading ends for the first time after about 30 minutes. In Figure 17(a), the number of simultaneous connections decreases at about 1800 seconds. However, BitTorrent continues setting up a connection intermittently as long as it continues starting to upload in other nodes.

In Figure 17(c), about 90 percent of the connections are finished within 180 seconds. In Figure 17(d), the interval of the request was performed within 10 seconds, and 90 percent of the connections were finished around 200 seconds at maximum.

As mentioned above, BitTorrent is capable of up to about 50 simultaneous connections. Such connections are short but frequent.



(a) Number of Established Connections

(b) Number of Requests

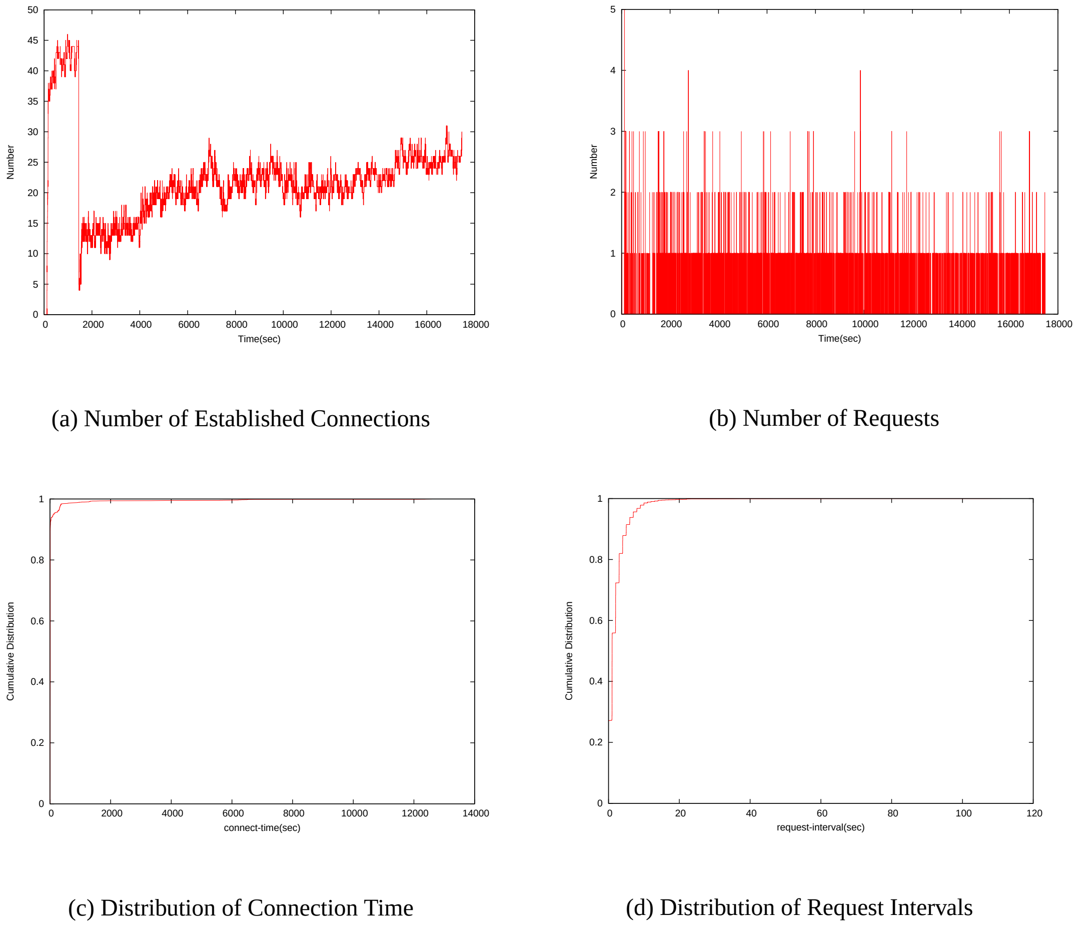
(c) Number of Established Connections (From 0 to 120 Seconds)

(d) Number of Requests (From 0 to 120 Seconds)

(e) Distribution of Connection Time

(f) Distribution of Request Intervals

Figure 16: Windows Live Messenger's NetworkTraffic



(a) Number of Established Connections

(b) Number of Requests

(c) Distribution of Connection Time

(d) Distribution of Request Intervals

Figure 17: BitTorrent's Network Traffic

4 Unified Multiplex Communication Architectures

We propose a Unified Multiplex communication architecture that maintains addresses only for service and is disposable. In this approach, it is possible to use abundant address spaces and plural addresses in IPv6.

First, we give an outline of the Unified Multiplex communication architecture in subsection 4.1. Next, in subsection 4.2 we describe the kind of disposable address that is dynamically generated with Unified Multiplex. Finally, in subsection 4.3, we consider this architecture from the results obtained in the preceding sections.

4.1 Outline of Unified Multiplex Communication Architectures

In the present communication architecture, the IP address of a node is fixed at each interface of the node. Therefore, the same IP address is used whenever the node communicates.

However, it is not desirable from a security viewpoint for an IP address to be fixed. This is because an address used for information that is transmitted to an outsider can become a weapon for a malicious attack. Moreover, access information could be extracted from an IP address, which can easily become a problem from the viewpoint of privacy.

The Unified Multiplex communication architecture we advocate is an IP that uses the identifier of a node and the previous node. Its mechanism provides privacy and security by generating an address dynamically and handling it as a disposable address. This can be achieved because the user dynamically generates an interface ID that can be set up freely among the vast address spaces (2128) contained in IPv6. This space is very large (264 pieces) and robust against the traditional brute force attacks, and thus it is almost impossible to gain unauthorized access from the outside. This approach has the advantage that an address is already invalid at the stage of leaking because it is disposable; therefore, even if the address of the service leaks to a third party, there is little exposure to risk.

4.2 Address Type of Unified Communication

EA is an address type that incorporates information suitable to the ephemeral port of legacy communication equipment. The ephemeral port is a port used by a general client to avoid specifying a port number beforehand, and it instead determines the port number (port not used among 49152

65535) that can be used at the time of connection to a server. The Unified communication setting is the address we assume the client uses. The client identifies a session using only EA, which is dynamically generated for every session and then thrown away at the end of a session. Therefore, there is only a very small possibility that user information in the EA could be derived by an address because there is little information related to the address during the brief period of service.

SSA is the address that a server uses in Unified communication. It is suitable for well-known ports and IP addresses in legacy communications, and it makes possible the identification of a session by using only SSA. This address indicates only a specific reliable communication site, and it makes it difficult for a third party who is not told the address to access it. Moreover, it is dynamically generated for every session, as with EA, and the address is canceled when a session ends. Therefore, the shorter the duration of use, the more difficult it is for legacy communication techniques to overcome the protection of information at a destination address. Moreover, the improvement of security can be expected beyond that of legacy communications by assuming that a third party's access is made more difficult by not opening the address to the public and thus making an attack difficult to deliver.

4.3 Consideration of Number of Address Pools

It is necessary to allocate addresses only for the service in each server and the client dynamically in Unified Multiplex. It is also necessary to perform DAD (Duplicated Address Detection) before an address is used. DAD takes about 60 seconds. In order to avoid the time lag due to DAD, the address used by an address pool function is pooled beforehand, and, if necessary, the system carries out addition to a pool, extraction, and assignment to a socket. The address pool, that is, the complete pre-DAD uncertain pool of addresses assigned to the socket, will, if necessary, be consumed by the address. In order to be able to respond without delay to address consumption, even if there is a large degree of it, it is necessary to have a pool of addresses to some extent. Therefore, it is necessary to correct the addresses if necessary to take into account each cause of consumption.

The number of address pools is estimated from the analysis of section 4. We set up a great many connections when using a tab in Firefox (nearly 500 connections may be required just to open ten tabs at the same time). E-mail or a messenger client always sets up fewer that 10 connections, and BitTorrent sets up 160 connections on average. As mentioned above, it is sufficient to have about 700 address pools.

5 Conclusion

In this paper, we analyzed network traffic for each of several applications and investigated how each application makes connections simultaneously. First, we confirmed that a Web browser incorporating multi-use Ajax can make a large number of connections simultaneously. Unified Multiplex communication architecture was developed to cope with such large numbers of connections, regardless of the number of address pools considered maximum. As mentioned above, we clarified that about 700 address pools are sufficient for operations.

A remaining task is to investigate, in a similar way, the connection requirements for consumption connection models. It is possible to control the number of address pools by the dynamic operation of an application.

Acknowledgements

First, I would like to express my sincere gratitude to my supervisor, Professor Masayuki Murata of Osaka University, for his continuous support and valuable advice throughout my studies, and for providing me this precious study opportunity in his laboratory.

I would like to especially express my deepest appreciation to Associate Professor Shingo Ata of Osaka City University, for his careful guidance and invaluable firsthand advice. No work in this thesis would have been possible without his support.

My deepest thanks also go to Dr. Hiroshi Kitamura of NEC Corporation, who gave important advice and assistance. He made an effort to secure my internship at NEC, and gave me instruction that cannot be obtained in school education, which I call the proper mental attitude as a member of society.

I am very grateful to Associate Professor Naoki Wakamiya and Assistant Professor Shin'ichi Arakawa of Osaka University, for their appropriate guidance.

I must deeply thank Ms. Morimoto, a member of the university clerical staff, for much help and encouragement at times.

The help of Ms. Kaoru Masumoto, Ms. Miwa Takahashi, and Ms. Haesung Hwang made it possible for me to complete this article, and I owe them a great debt of thanks.

Finally, I would like to extend my gratitude to my family for their understanding, support, encouragement and sacrifices throughout my studies.

References

[1] M. Nabe, K. Baba, M. Murata, and H. Miyahara, “Analysis and modeling of WWW traffic for designing internet access networks,” *IEICE Transaction Communication*, vol. J80-B-I, pp. 428–437, Jan. 1997.

[2] K. Cho, K. Fukuda, H. Esaki, and A. Kato, “The impact and implications of the growth in residential user-to-user traffic,” in *Proceedings of the 2006 conference on Applications, technologies, architectures, and protocols for computer communications*, pp. 207–218, Oct. 2006.

[3] H.-K. Choi and J. O. Limb, “A behavioral model of web traffic,” in *Proceedings of the Seventh Annual International Conference on Network Protocols*, pp. 327–334, Nov. 1999.

[4] L. Limwiwatkul and A. Rungsawang, “Distributed denial of service detection using TCP/IP header and traffic measurement analysis,” *International Symposium on Communications and Information Technology (ISCIT)*, vol. 1, pp. 605–610, Oct. 2004.

[5] B. A. Mah, “An empirical model of http network traffic,” *IEEE Computer and Communications Societies, Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 7, pp. 592–600, Apr. 1997.

[6] R. T. Fielding, J. Gettys, J. C. Mogul, H. F. Nielsen, L. Masinter, P. J. Leach, and T. Berners-Lee, “Hypertext transfer protocol – HTTP/1.1,” *RFC2616*, June 1999.

[7] bugzilla. available online at `https://bugzilla.mozilla.org/show_bug.cgi?id=423377`.

[8] bugzilla. available online at `https://bugzilla.mozilla.org/show_bug.cgi?id=92072`.

[9] P. Srisuresh and M. Holdrege, “IP network address translator (NAT) terminology and considerations,” *RFC2663*, Aug. 1999.

[10] Ministry of Internal Affairs and Communications, “The study report about a smooth IPv6 shift of the internet (in japanese).” available online at `http://www.soumu.go.jp/s-news/2008/pdf/080617_2_bt1.pdf`.

[11] Institute for Information and Communications Policy, “Statistical outlook of the internet in japan (in japanese).” available online at http://www.soumu.go.jp/iicp/chousakenkyu/data/research/survey/telecom/2007/2007-1-01-2.pdf, 2007.