電子情報工学科

工学専門実験　　報告

第　1　号

実　　験　　題　　目

C1 通信ネットワーク工学の基礎実験

通信工学コース　　第 3 班

池田　卓人

宇山　侑希

楠瀬　智也

平良　隼涼

長谷川　舞有

船越　大智

村田　悠斗

吉野　将裕

報　　告　　者

08D23159 番　　吉野　将裕　（通信工学コース）

電子メールアドレス：u589438g@ecs.osaka-u.ac.jp

令和 7 年　6 月 19 日

大阪大学工学部電子情報工学科

１.目的

パケットキャプチャログの分析を通じて、トランスポート層とアプリケーション層における代表的な通信プロトコルである User Datagram Protocol(UDP)、Transmission Control Protocol(TCP)、ならびに Hypertext Transfer Protocol(HTTP)の基本動作を理解。加えて、Web アプリケーションにおける輻輳現象の観察を通じて通信トラヒックの基礎を学習する。

２.第一週：パケットキャプチャによる通信ログ分析

パケットキャプチャを用いて、HTTP メッセージの送受信、ならびにそれに用いられるトランスポート層プロトコル TCP の動作を確認した。また、トランスポート層プロトコル UDP が、Domain Name System(DNS)により URL を IP アドレスに変換する際に使用されることを確認した。

2.1 実験環境

ハードウェア：

・Raspberry Pi 3 Model B+

ソフトウェア:

・Raspberry Pi OS 5.10(オペレーティング・システム)

・Wireshark 2.6.20(パケットキャプチャソフトウェア)

・Curl 7.64.0(HTTP をサポートするコマンドラインツール)

・Mozilla Firefox 78.9.0esr(Web ブラウザ)

2.2 理論

第 1 週で用いるシステムの模式図を図 1 に示す。以降では、手元で操作する端末(Raspberry Pi)を単にクライアントと呼ぶ。

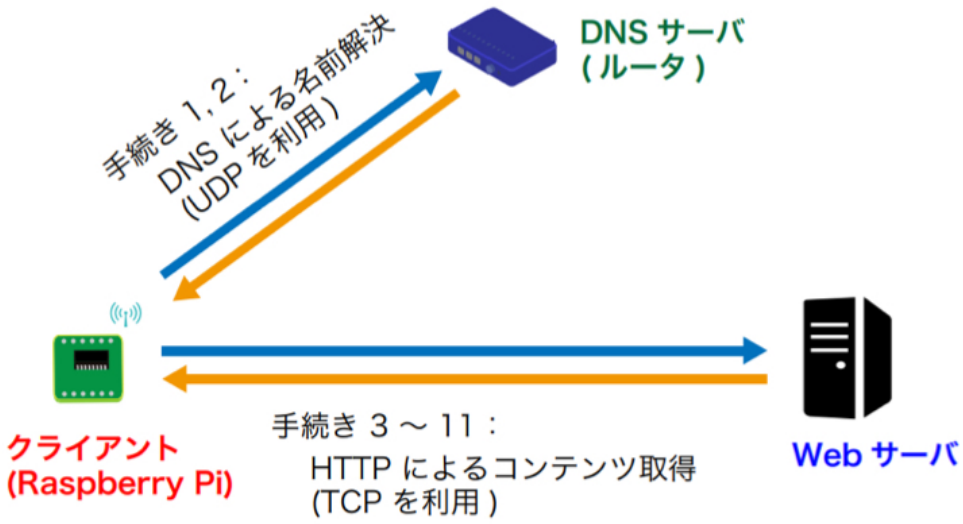

図 1　第 1 週で用いるシステムの模式図

コマンドラインツール「Curl」を使用して Web ページを閲覧する際には以下のような手順で通信が行われている。

・ドメイン名を IP アドレスに変換

1. クライアントは入力された URL のコンテンツを保持する Web サーバの IP アドレスを DNS サーバに問い合わせる(名前解決)。この通信には UDP が使用される。
2. DNS サーバは 1 の問い合わせ結果をクライアントに返す。この通信も、1 と同様に UDP が使用される。

・TCP コネクションの確立

3. クライアントは 2 で通知された IP アドレスと、HTTP におけるサーバ側ポート番号の組を宛先として、TCP コネクション開始要求(SYN)を送信する。
4. Web サーバは、3 で送信された SYN を受信し、それに対する確認応答(ACK)に加え、コネクション開始要求(SYN)をクライアントに送信する。
5. クライアントは 4 で送信された SYN/ACK を受信し、それに対する ACK を Web サーバに送信する。これによりクライアントと Web サーバの間に TCP コネクションが確立される。

・HTTP メッセージの送受信

6. クライアントは 5 で確立された TCP コネクションを通じて、Web サーバに「HTTP GET メッセージ」を格納したパケットを送信する。
7. Web サーバは、6 で送信されたパケットを受信し、ACK をクライアントに返す。また、このパ

ケットに含まれる「HTTP GET メッセージ」を読み込み、「HTTP Response メッセージ」をクライアントに返す。これにはクライアントが要求した Web ページのソースコード(HTML ファイル)が含まれている。

8. クライアントは、7 で送信された「HTTP Response メッセージ」を受信し、ACK を返す。

・TCP コネクションの終了

9. クライアントは、Web サーバにコネクション終了通知(FIN)を送信する。
10. Web サーバは、9 で送信された FIN を受信し、それに対する ACK に加え、コネクション終了通知(FIN)をクライアントに送信する。
11. クライアントは、10 で送信された FIN を受信し、それに対する ACK を Web サーバに送信する。

2.3 実験課題 1

この課題ではクライアントでターミナルを起動し、以下のコマンドを実行した。

```
$ curl http://www2b.comm.eng.osaka-u.ac.jp/~yoshiaki/C1/page1.html
```

実行結果は以下のようになった。

```
ubuntu@raspberrypi3:~ $ curl http://www2b.comm.eng.osaka-
u.ac.jp/~yoshiaki/C1/page1.html
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html lang="en">
<head>
<meta http-equiv="Content-Type" content="text/html"; charset="utf-8" />
<title>Page 1</title>
</head>
<body>
Hello, World!
</body>
</html>
```

ターミナルに Web ページのソースコードが表示されていることがわかる。

2.4 実験課題 2

以下に示す手順で、実験課題 1 のコマンドを入力した時に行われた一連の通信に関するログを取得した。

1. クライアントで Wireshark を起動し、パケットキャプチャを開始。
2. 実験課題 1 と同じコマンドを実行
3. Web ページのソースコードが標準出力に表示されたら、パケットキャプチャを停止。

実行結果は以下のようになった。

```
No.     Time        Source                Destination           Protocol Length Info
   8347 72.274886   192.168.12.66         10.144.0.1            DNS      88
Standard query 0x2cc8 A www2b.comm.eng.osaka-u.ac.jp
   8348 72.275140   192.168.12.66         10.144.0.1            DNS      88
Standard query 0x39c8 AAAA www2b.comm.eng.osaka-u.ac.jp
   8349 72.276049   10.144.0.1            192.168.12.66         DNS
104      Standard query response 0x2cc8 A www2b.comm.eng.osaka-u.ac.jp A
192.168.13.80
   8350 72.276242   10.144.0.1            192.168.12.66         DNS
160      Standard query response 0x39c8 AAAA www2b.comm.eng.osaka-u.ac.jp
CNAME genji.comm.eng.osaka-u.ac.jp SOA gene.comm.eng.osaka-u.ac.jp
   8351 72.277170   192.168.12.66         192.168.13.80         TCP
74       35016 → 80 [SYN] Seq=1007184416 Win=64240 Len=0 MSS=1460
SACK_PERM=1 TSval=3536644626 TSecr=0 WS=128
   8352 72.277751   192.168.13.80         192.168.12.66         TCP
74       80 → 35016 [SYN, ACK] Seq=2277431442 Ack=1007184417 Win=65160
Len=0 MSS=1460 SACK_PERM=1 TSval=1932884923 TSecr=3536644626 WS=128
   8353 72.277875   192.168.12.66         192.168.13.80         TCP
66       35016 → 80 [ACK] Seq=1007184417 Ack=2277431443 Win=64256 Len=0
TSval=3536644627 TSecr=1932884923
   8354 72.278158   192.168.12.66         192.168.13.80         HTTP
181      GET /~yoshiaki/C1/page1.html HTTP/1.1
   8355 72.278572   192.168.13.80         192.168.12.66         TCP
66       80 → 35016 [ACK] Seq=2277431443 Ack=1007184532 Win=65152 Len=0
TSval=1932884924 TSecr=3536644627
```

```
   8356  72.279152      192.168.13.80                192.168.12.66                HTTP
584      HTTP/1.1 200 OK  (text/html)
   8357  72.279238      192.168.12.66                192.168.13.80                TCP
66       35016 → 80 [ACK] Seq=1007184532 Ack=2277431961 Win=64000 Len=0
TSval=3536644628 TSecr=1932884925
   8358  72.280046      192.168.12.66                192.168.13.80                TCP
66       35016 → 80 [FIN, ACK] Seq=1007184532 Ack=2277431961 Win=64128
Len=0 TSval=3536644629 TSecr=1932884925
   8359  72.280709      192.168.13.80                192.168.12.66                TCP
66       80 → 35016 [FIN, ACK] Seq=2277431961 Ack=1007184533 Win=65152
Len=0 TSval=1932884926 TSecr=3536644629
   8360  72.280870      192.168.12.66                192.168.13.80                TCP
66       35016 → 80 [ACK] Seq=1007184533 Ack=2277431962 Win=64128 Len=0
TSval=3536644630 TSecr=1932884926
```

2.4.1 ドメイン名を IP アドレスに変換

ログでこの手続きを行っているのは、以下の部分である。

```
 8347  72.274886     192.168.12.66                10.144.0.1                   DNS          88
Standard query 0x2cc8 A www2b.comm.eng.osaka-u.ac.jp
  8349  72.276049     10.144.0.1                   192.168.12.66                DNS          104
Standard query response 0x2cc8 A www2b.comm.eng.osaka-u.ac.jp A 192.168.13.80
```

このログから、送信元 IP アドレス、受信元 IP アドレスを読み取ることができる。

次に、パケットキャプチャログの詳細を抜粋して示す。

・8347

```
User Datagram Protocol, Src Port: 57683, Dst Port: 53
    Source Port: 57683
    Destination Port: 53
    Length: 54
                                    (中略)
    Queries
        www2b.comm.eng.osaka-u.ac.jp: type A, class IN
            Name: www2b.comm.eng.osaka-u.ac.jp
```

[Name Length: 28]

[Label Count: 6]

Type: A (Host Address) (1)

Class: IN (0x0001)

・8349

User Datagram Protocol, Src Port: 53, Dst Port: 57683

Source Port: 53

Destination Port: 57683

Length: 70

(中略)

Queries

www2b.comm.eng.osaka-u.ac.jp: type A, class IN

Name: www2b.comm.eng.osaka-u.ac.jp

[Name Length: 28]

[Label Count: 6]

Type: A (Host Address) (1)

Class: IN (0x0001)

Answers

www2b.comm.eng.osaka-u.ac.jp: type A, class IN, addr 192.168.13.80

Name: www2b.comm.eng.osaka-u.ac.jp

Type: A (Host Address) (1)

Class: IN (0x0001)

Time to live: 1 (1 second)

Data length: 4

Address: 192.168.13.80

分析結果を表１にまとめた。

表１　ドメイン名を IP アドレスに変換する手続きの分析

No.	送信元 IP アドレス	受信先 IP アドレス	送信元 ポート番号	受信先 ポート番号	UDP セグメント長
8347	192.168.12.66	10.144.0.1	57683	53	54
8349	10.144.0.1	192.168.12.66	53	57683	70

アプリケーション層のメッセージの内容を以下に示す。

8347 : www2b.comm.eng.osaka-u.ac.jp の Web ページを保持する IP アドレスの問い合わせ

8349 : www2b.comm.eng.osaka-u.ac.jp の Web ページを保持する IP アドレスを送信

以上より、この 2 つの UDP パケットの送受信によって、クライアントは特定のドメイン名二対応する IP アドレスを取得することができることがわかる。

2.4.2 TCP コネクションの確立

ログでこの手続きを行っているのは、以下の部分である。

```
 8351 72.277170    192.168.12.66          192.168.13.80          TCP      74
35016 → 80 [SYN] Seq=1007184416 Win=64240 Len=0 MSS=1460 SACK_PERM=1
TSval=3536644626 TSecr=0 WS=128
 8352 72.277751    192.168.13.80          192.168.12.66          TCP      74
80 → 35016 [SYN, ACK] Seq=2277431442 Ack=1007184417 Win=65160 Len=0
MSS=1460 SACK_PERM=1 TSval=1932884923 TSecr=3536644626 WS=128
 8353 72.277875    192.168.12.66          192.168.13.80          TCP      66
35016 → 80 [ACK] Seq=1007184417 Ack=2277431443 Win=64256 Len=0
TSval=3536644627 TSecr=1932884923
```

次に、No.8351 のパケットキャプチャログの詳細を抜粋して示す。

```
Transmission Control Protocol, Src Port: 35016, Dst Port: 80, Seq: 1007184416, Len:
0
                                    (中略)
    Acknowledgment number: 0
```

ログから読み取ることができる情報を表 2、表 3 にまとめる。

表 2　TCP コネクションを確立するパケットの分析 1

No.	送信元 IP アドレス	受信先 IP アドレス	送信元 ポート番号	受信先 ポート番号	TCP セグメント長
8351	192.168.12.66	192.168.13.80	35016	80	0
8352	192.168.13.80	192.168.12.66	80	35016	0
8353	192.168.12.66	192.168.13.80	35016	80	0

表 3　TCP コネクションを確立するパケットの分析 2

No.	ON になっている TCP フラグ	シーケンス番号	ACK 番号
8351	SYN	1007184416	0
8352	SYN, ACK	2277431442	1007184417
8353	ACK	1007184417	2277431443

No.8351 において、クライアントが SYN を送信し、No.8352 の ACK 番号は、No.8351 のシーケンス番号に 1 を加えた ACK と SYN が返されている。同様に、No.8353 では No.8352 の ACK 番号と同じシーケンス番号の ACK が返されており、これら 3 つのパケットで TCP 接続の確立が行われたことがわかる。なお、この 3 つのパケットにはアプリケーション層のメッセージが存在しない。

2.4.3 HTTP メッセージの送受信

ログでこの手続きを行っているのは、以下の部分である。

```
8354 72.278158　192.168.12.66　192.168.13.80　HTTP　181
GET /~yoshiaki/C1/page1.html HTTP/1.1

8355 72.278572　192.168.13.80　192.168.12.66　TCP　66
80 → 35016 [ACK] Seq=2277431443 Ack=1007184532 Win=65152 Len=0
TSval=1932884924 TSecr=3536644627

8356 72.279152　192.168.13.80　192.168.12.66　HTTP　584
HTTP/1.1 200 OK （text/html）

8357 72.279238　192.168.12.66　192.168.13.80　TCP　66
```

35016 → 80 [ACK] Seq=1007184532 Ack=2277431961 Win=64000 Len=0 TSval=3536644628 TSecr=1932884925

次に、それぞれのパケットキャプチャログの詳細を抜粋して示す。

・8354

Transmission Control Protocol, Src Port: 35016, Dst Port: 80, Seq: 1007184417, Ack: 2277431443, Len: 115

(中略)

Flags: 0x018 (PSH, ACK)

(中略)

Hypertext Transfer Protocol

GET /~yoshiaki/C1/page1.html HTTP/1.1¥r¥n

[Expert Info (Chat/Sequence): GET /~yoshiaki/C1/page1.html HTTP/1.1¥r¥n]

[GET /~yoshiaki/C1/page1.html HTTP/1.1¥r¥n]

[Severity level: Chat]

[Group: Sequence]

Request Method: GET

Request URI: /~yoshiaki/C1/page1.html

Request Version: HTTP/1.1

Host: www2b.comm.eng.osaka-u.ac.jp¥r¥n

User-Agent: curl/7.64.0¥r¥n

Accept: */*¥r¥n

¥r¥n

[Full request URI: http://www2b.comm.eng.osaka-u.ac.jp/~yoshiaki/C1/page1.html]

[HTTP request 1/1]

[Response in frame: 8356]

・8356

Transmission Control Protocol, Src Port: 80, Dst Port: 35016, Seq: 2277431443, Ack: 1007184532, Len: 518

(中略)

Flags: 0x018 (PSH, ACK)

(中略)

Hypertext Transfer Protocol

HTTP/1.1 200 OK¥r¥n

[Expert Info (Chat/Sequence): HTTP/1.1 200 OK¥r¥n]

[HTTP/1.1 200 OK¥r¥n]

[Severity level: Chat]

[Group: Sequence]

Response Version: HTTP/1.1

Status Code: 200

[Status Code Description: OK]

Response Phrase: OK

Date: Tue, 11 May 2021 04:04:01 GMT¥r¥n

Server: Apache/2.4.41 (Ubuntu)¥r¥n

Last-Modified: Thu, 17 Sep 2020 03:12:39 GMT¥r¥n

ETag: "10a-5af79c1bb058b"¥r¥n

Accept-Ranges: bytes¥r¥n

Content-Length: 266¥r¥n

[Content length: 266]

Vary: Accept-Encoding¥r¥n

Content-Type: text/html¥r¥n

¥r¥n

[HTTP response 1/1]

[Time since request: 0.000993430 seconds]

[Request in frame: 8354]

[Request URI: http://www2b.comm.eng.osaka-u.ac.jp/~yoshiaki/C1/page1.html]

File Data: 266 bytes

このログから、それぞれのパケットにおける送信元ポート番号、受信先ポート番号、TCP セグメント長、シーケンス番号、ACK 番号、ON になっている TCP フラグを読み取ることができる。各パケットの情報を表 4 と表 5 にまとめる。

表 4　HTTP メッセージの送受信の手続きの分析 1

No.	送信元 IP アドレス	受信先 IP アドレス	送信元 ポート番号	受信先 ポート番号	TCP セグメント長
8354	192.168.12.66	192.168.13.80	35016	80	115
8355	192.168.13.80	192.168.12.66	80	35016	0
8356	192.168.13.80	192.168.12.66	80	35016	518
8357	192.168.12.66	192.168.13.80	35016	80	0

表 5　HTTP メッセージの送受信の手続きの分析 2

No.	ON になっている TCP フラグ	シーケンス番号	ACK 番号
8354	PSH, ACK	1007184417	2277431443
8355	ACK	2277431443	1007184532
8356	PSH, ACK	2277431443	1007184532
8357	ACK	1007184532	2277431961

アプリケーション層のメッセージの内容を以下に示す。

8354 : Web サーバに「HTTP GET メッセージ」を送信して
　　/~yoshiaki/C1/page1.html のページを要求

8355 : なし

8356 : 「HTTP GET メッセージ」に対する「HTTP Response メッセージ」を送信

8357 : なし

表 5 から No.8355 のシーケンス番号と No.8354 の ACK 番号が等しいことから、これは No.8354 の「HTTP GET メッセージ」に対する ACK であることが確認できる。同様に、No.8357 のパケットのシーケンス番号と No.8356 のパケットの ACK 番号が等しいため、No.8356 の「HTTP Response メッセージ」に対する ACK であることがわかる。

2.4.4 TCP コネクションの終了

ログでこの手続きを行っているのは、以下の部分である。

```
 8358 72.280046    192.168.12.66          192.168.13.80          TCP       66
35016 → 80 [FIN, ACK] Seq=1007184532 Ack=2277431961 Win=64128 Len=0
TSval=3536644629 TSecr=1932884925
 8359 72.280709    192.168.13.80          192.168.12.66          TCP       66
80 → 35016 [FIN, ACK] Seq=2277431961 Ack=1007184533 Win=65152 Len=0
TSval=1932884926 TSecr=3536644629
 8360 72.280870    192.168.12.66          192.168.13.80          TCP       66
35016 → 80 [ACK] Seq=1007184533 Ack=2277431962 Win=64128 Len=0
TSval=3536644630 TSecr=1932884926
```

このログから読み取ることができる各パケットの情報を表 6 と表 7 にまとめる。

表 6　TCP コネクションを終了する手続きの分析 1

No.	送信元 IP アドレス	受信先 IP アドレス	送信元 ポート番号	受信先 ポート番号	TCP セグメント長
8358	192.168.12.66	192.168.13.80	35016	80	0
8359	192.168.13.80	192.168.12.66	80	35016	0
8360	192.168.12.66	192.168.13.80	80	35016	0

表 7　TCP コネクションを終了する手続きの分析 2

No.	ON になっている TCP フラグ	シーケンス番号	ACK 番号
8358	ACK, FIN	1007184532	2277431961
8359	ACK, FIN	2277431961	1007184533
8360	ACK	1007184533	2277431962

No.8358 のパケットでは、FIN を Web サーバに送信し、No.8359 でその ACK が返されている。同様に、No.8360 でも No.8359 の FIN に対する ACK 番号が送信されており、コネクションの終了処理が進んでいることがわかる。また、この 3 つのパケットにはいずれもアプリケーション層のメッセージは存在しない。

2.5 実験課題 3

この課題では、実験課題 2 と同様の手順で Web ブラウザ(Firefox)から以下にアクセスした際の通信ログを分析する。

http://www2b.comm.eng.osaka-u.ac.jp/~yoshiaki/C1/page1.html

取得したのは以下のログである。このログを分析し、結果をまとめる。

No. Time Source Destination Protocol Length Info
688 17.977035 192.168.12.66 10.144.0.1 DNS 84 Standard query 0x140f A detectportal.firefox.com
689 17.979484 10.144.0.1 192.168.12.66 DNS 489 Standard query response 0x140f A detectportal.firefox.com CNAME detectportal.prod.mozaws.net CNAME prod.detectportal.prod.cloudops.mozgcp.net A 34.107.221.82 NS ns-cloud-e4.googledomains.com NS ns-cloud-e1.googledomains.com NS ns-cloud-e3.googledomains.com NS ns-cloud-e2.googledomains.com A 216.239.32.110 A 216.239.34.110 A 216.239.36.110 A 216.239.38.110 AAAA 2001:4860:4802:32::6e AAAA 2001:4860:4802:34::6e AAAA 2001:4860:4802:36::6e AAAA 2001:4860:4802:38::6e
710 18.273600 192.168.12.66 10.144.0.1 DNS 71 Standard query 0xbc85 A mozilla.org
711 18.274560 10.144.0.1 192.168.12.66 DNS 304 Standard query response 0xbc85 A mozilla.org A 44.236.48.31 A 44.235.246.155 A 44.236.72.93 NS ns7-66.akam.net NS ns1-240.akam.net NS ns4-64.akam.net NS ns5-65.akam.net A 184.85.248.65 A 193.108.91.240 A 84.53.139.64 A 96.7.49.66 AAAA 2600:1401:2::f0
712 18.274845 192.168.12.66 10.144.0.1 DNS 71 Standard query 0x3aba A mozilla.org
714 18.275518 10.144.0.1 192.168.12.66 DNS 304 Standard query response 0x3aba A mozilla.org A 44.236.48.31 A 44.235.246.155 A 44.236.72.93 NS ns7-66.akam.net NS ns1-240.akam.net NS ns4-64.akam.net NS ns5-65.akam.net A 184.85.248.65 A 193.108.91.240 A 84.53.139.64 A

96.7.49.66 AAAA 2600:1401:2::f0

715 18.276111 192.168.12.66 10.144.0.1 DNS 84 Standard query 0xa2bf A detectportal.firefox.com

716 18.276852 10.144.0.1 192.168.12.66 DNS 489 Standard query response 0xa2bf A detectportal.firefox.com CNAME detectportal.prod.mozaws.net CNAME prod.detectportal.prod.cloudops.mozgcp.net A 34.107.221.82 NS ns-cloud-e4.googledomains.com NS ns-cloud-e1.googledomains.com NS ns-cloud-e3.googledomains.com NS ns-cloud-e2.googledomains.com A 216.239.32.110 A 216.239.34.110 A 216.239.36.110 A 216.239.38.110 AAAA 2001:4860:4802:32::6e AAAA 2001:4860:4802:34::6e AAAA 2001:4860:4802:36::6e AAAA 2001:4860:4802:38::6e

717 18.277491 192.168.12.66 10.144.0.1 DNS 84 Standard query 0xfdb4 A detectportal.firefox.com

720 18.278186 10.144.0.1 192.168.12.66 DNS 489 Standard query response 0xfdb4 A detectportal.firefox.com CNAME detectportal.prod.mozaws.net CNAME prod.detectportal.prod.cloudops.mozgcp.net A 34.107.221.82 NS ns-cloud-e4.googledomains.com NS ns-cloud-e1.googledomains.com NS ns-cloud-e3.googledomains.com NS ns-cloud-e2.googledomains.com A 216.239.32.110 A 216.239.34.110 A 216.239.36.110 A 216.239.38.110 AAAA 2001:4860:4802:32::6e AAAA 2001:4860:4802:34::6e AAAA 2001:4860:4802:36::6e AAAA 2001:4860:4802:38::6e

743 18.901046 192.168.12.66 10.144.0.1 DNS 95 Standard query 0x2777 A content-signature-2.cdn.mozilla.net

744 18.903489 10.144.0.1 192.168.12.66 DNS 368 Standard query response 0x2777 A content-signature-2.cdn.mozilla.net CNAME d2nxq2uap88usk.cloudfront.net A 13.33.29.20 A 13.33.29.2 A 13.33.29.106 A 13.33.29.125 NS ns-1295.awsdns-33.org NS ns-365.awsdns-45.com NS ns-704.awsdns-24.net NS ns-1811.awsdns-34.co.uk A 205.251.197.15 A 205.251.194.192

779 19.077282 192.168.12.66 10.144.0.1 DNS 97 Standard query 0xb487 A firefox.settings.services.mozilla.com

781 19.079781 10.144.0.1 192.168.12.66 DNS

330 Standard query response 0xb487 A firefox.settings.services.mozilla.com A 99.86.193.95 A 99.86.193.125 A 99.86.193.83 A 99.86.193.91 NS ns-1364.awsdns-42.org NS ns-1627.awsdns-11.co.uk NS ns-166.awsdns-20.com NS ns-972.awsdns-57.net A 205.251.192.166 A 205.251.195.204

1018 20.715609 192.168.12.66 10.144.0.1 DNS 85 Standard query 0xe902 A push.services.mozilla.com

1019 20.718021 10.144.0.1 192.168.12.66 DNS 305 Standard query response 0xe902 A push.services.mozilla.com CNAME autopush.prod.mozaws.net A 35.163.208.27 NS ns-614.awsdns-12.net NS ns-377.awsdns-47.com NS ns-1260.awsdns-29.org NS ns-1986.awsdns-56.co.uk A 205.251.196.236 A 205.251.194.102

1023 20.770565 192.168.12.66 10.144.0.1 DNS 85 Standard query 0xc06b A push.services.mozilla.com

1024 20.771246 10.144.0.1 192.168.12.66 DNS 305 Standard query response 0xc06b A push.services.mozilla.com CNAME autopush.prod.mozaws.net A 35.163.208.27 NS ns-614.awsdns-12.net NS ns-377.awsdns-47.com NS ns-1260.awsdns-29.org NS ns-1986.awsdns-56.co.uk A 205.251.196.236 A 205.251.194.102

1025 20.772946 192.168.12.66 10.144.0.1 DNS 85 Standard query 0x0761 A push.services.mozilla.com

1026 20.773595 10.144.0.1 192.168.12.66 DNS 305 Standard query response 0x0761 A push.services.mozilla.com CNAME autopush.prod.mozaws.net A 35.163.208.27 NS ns-614.awsdns-12.net NS ns-377.awsdns-47.com NS ns-1260.awsdns-29.org NS ns-1986.awsdns-56.co.uk A 205.251.196.236 A 205.251.194.102

1076 21.098356 192.168.12.66 10.144.0.1 DNS 77 Standard query 0xfe6b A ocsp.digicert.com

1077 21.099169 10.144.0.1 192.168.12.66 DNS 373 Standard query response 0xfe6b A ocsp.digicert.com CNAME cs9.wac.phicdn.net A 117.18.237.29 NS ns3.phicdn.net NS ns1.phicdn.net NS ns4.phicdn.net NS ns2.phicdn.net A 72.21.80.5 A 72.21.80.6 A 192.229.254.5 A

192.229.254.6 AAAA 2606:2800:1::5 AAAA 2606:2800:1::6 AAAA 2606:2800:e::5 AAAA 2606:2800:e::6

2622 34.946600 192.168.12.66 10.144.0.1 DNS 88 Standard query 0x2089 A www2b.comm.eng.osaka-u.ac.jp

2623 34.947637 10.144.0.1 192.168.12.66 DNS 104 Standard query response 0x2089 A www2b.comm.eng.osaka-u.ac.jp A 192.168.13.80

2639 35.055745 192.168.12.66 192.168.13.80 TCP 74 35516 → 80 [SYN] Seq=3906406002 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=3543356595 TSecr=0 WS=128

2640 35.056306 192.168.13.80 192.168.12.66 TCP 74 80 → 35516 [SYN, ACK] Seq=35308251 Ack=3906406003 Win=65160 Len=0 MSS=1460 SACK_PERM=1 TSval=1939596895 TSecr=3543356595 WS=128

2641 35.056438 192.168.12.66 192.168.13.80 TCP 66 35516 → 80 [ACK] Seq=3906406003 Ack=35308252 Win=64256 Len=0 TSval=3543356596 TSecr=1939596895

2642 35.056808 192.168.12.66 192.168.13.80 HTTP 436 GET /~yoshiaki/C1/page1.html HTTP/1.1

2643 35.057355 192.168.13.80 192.168.12.66 TCP 66 80 → 35516 [ACK] Seq=35308252 Ack=3906406373 Win=64896 Len=0 TSval=1939596896 TSecr=3543356597

2644 35.058177 192.168.13.80 192.168.12.66 HTTP 627 HTTP/1.1 200 OK (text/html)

2645 35.058247 192.168.12.66 192.168.13.80 TCP 66 35516 → 80 [ACK] Seq=3906406373 Ack=35308813 Win=64000 Len=0 TSval=3543356598 TSecr=1939596897

2707 35.533754 192.168.12.66 192.168.13.80 HTTP 334 GET /favicon.ico HTTP/1.1

2708 35.534299 192.168.13.80 192.168.12.66 TCP 66 80 → 35516 [ACK] Seq=35308813 Ack=3906406641 Win=64640 Len=0 TSval=1939597373 TSecr=3543357073

```
   2709  35.534821      192.168.13.80              192.168.12.66              HTTP
572      HTTP/1.1 404 Not Found  (text/html)
   2710  35.534867      192.168.12.66              192.168.13.80              TCP
66       35516 → 80 [ACK] Seq=3906406641 Ack=35309319 Win=64000 Len=0
TSval=3543357075 TSecr=1939597373
   3222  40.534947      192.168.13.80              192.168.12.66              TCP
66       80 → 35516 [FIN, ACK] Seq=35309319 Ack=3906406641 Win=64640
Len=0 TSval=1939602374 TSecr=3543357075
   3223  40.535294      192.168.12.66              192.168.13.80              TCP
66       35516 → 80 [FIN, ACK] Seq=3906406641 Ack=35309320 Win=64128
Len=0 TSval=3543362075 TSecr=1939602374
   3224  40.535752      192.168.13.80              192.168.12.66              TCP
66       80 → 35516 [ACK] Seq=35309320 Ack=3906406642 Win=64640 Len=0
TSval=1939602374 TSecr=3543362075
```

2.5.1 ドメイン名を IP アドレスに変換

ログでこの手続きを行っているのは、以下の部分である。

```
  2622  34.946600     192.168.12.66              10.144.0.1                 DNS        88
Standard query 0x2089 A www2b.comm.eng.osaka-u.ac.jp
  2623  34.947637     10.144.0.1                 192.168.12.66              DNS        104
Standard query response 0x2089 A www2b.comm.eng.osaka-u.ac.jp A 192.168.13.80
```

このログから、送信元 IP アドレス、受信元 IP アドレスを読み取ることができる。

次に、No.2622 のパケットキャプチャログの詳細を抜粋して示す。

```
User Datagram Protocol, Src Port: 49142, Dst Port: 53
    Source Port: 49142
    Destination Port: 53
    Length: 54
                                    (中略)
    Queries
        www2b.comm.eng.osaka-u.ac.jp: type A, class IN
            Name: www2b.comm.eng.osaka-u.ac.jp
```

```
        [Name Length: 28]
        [Label Count: 6]
        Type: A (Host Address) (1)
        Class: IN (0x0001)
```

このログから送信元ポート番号、受信先ポート番号、UDP セグメント長を読み取ることができる。

同様に、No.2623 のパケットキャプチャログの詳細を抜粋して示す。

```
User Datagram Protocol, Src Port: 53, Dst Port: 49142
    Source Port: 53
    Destination Port: 49142
    Length: 70
                                    (中略)
    Queries
        www2b.comm.eng.osaka-u.ac.jp: type A, class IN
            Name: www2b.comm.eng.osaka-u.ac.jp
            [Name Length: 28]
            [Label Count: 6]
            Type: A (Host Address) (1)
            Class: IN (0x0001)
    Answers
        www2b.comm.eng.osaka-u.ac.jp: type A, class IN, addr 192.168.13.80
            Name: www2b.comm.eng.osaka-u.ac.jp
            Type: A (Host Address) (1)
            Class: IN (0x0001)
            Time to live: 1 (1 second)
            Data length: 4
            Address: 192.168.13.80
```

No.2623 のパケットについてもログから送信元ポート番号、受信先ポート番号、UDP セグメント長を読み取ることができる。

分析結果を表 8 にまとめる。

表 8　ドメイン名を IP アドレスに変換する手続きの分析

No.	送信元 IP アドレス	受信先 IP アドレス	送信元 ポート番号	受信先 ポート番号	UDP セグメント長
2622	192.168.12.66	10.144.0.1	49142	53	54
2623	10.144.0.1	192.168.12.66	53	49142	70

アプリケーション層のメッセージの内容を以下に示す。

2622 : www2b.comm.eng.osaka-u.ac.jp の Web ページを保持する IP アドレスの問い合わせ

2623 : www2b.comm.eng.osaka-u.ac.jp の Web ページを保持する IP アドレスを送信

以上より、２つの UDP パケットの送受信によってクライアントは特定のドメイン名に対応する IP アドレスを取得できていることがわかる。

2.5.2 TCP コネクションの確立

ログでこの手続きを行っているのは、以下の部分である。

```
  2639 35.055745    192.168.12.66          192.168.13.80          TCP       74
35516 → 80 [SYN] Seq=3906406002 Win=64240 Len=0 MSS=1460 SACK_PERM=1
TSval=3543356595 TSecr=0 WS=128
  2640 35.056306    192.168.13.80          192.168.12.66          TCP       74
80 → 35516 [SYN, ACK] Seq=35308251 Ack=3906406003 Win=65160 Len=0
MSS=1460 SACK_PERM=1 TSval=1939596895 TSecr=3543356595 WS=128
  2641 35.056438    192.168.12.66          192.168.13.80          TCP       66
35516 → 80 [ACK] Seq=3906406003 Ack=35308252 Win=64256 Len=0
TSval=3543356596 TSecr=1939596895
```

次に、No.2639 のパケットキャプチャログの詳細を抜粋して示す。

```
Transmission Control Protocol, Src Port: 35516, Dst Port: 80, Seq: 3906406002, Len:
0
                                (中略)
Acknowledgment number: 0
```

ログから読み取ることができる情報を表 9、表 10 にまとめる。

表 9　TCP コネクションを確立する手続きの分析 1

No.	送信元 IP アドレス	受信先 IP アドレス	送信元 ポート番号	受信先 ポート番号	TCP セグメント長
2639	192.168.12.66	192.168.13.80	35516	80	0
2640	192.168.13.80	192.168.12.66	80	35516	0
2641	192.168.12.66	192.168.13.80	35516	80	0

表 10　TCP コネクションを確立する手続きの分析 2

No.	ON になっている TCP フラグ	シーケンス番号	ACK 番号
2639	SYN	3906406002	0
2640	SYN, ACK	35308251	3906406003
2641	ACK	3906406003	35308252

以上より、実験課題 2 の No.8351 から No.8353 のパケットと同じ役割であることがわかる。

2.5.3 HTTP メッセージの送受信

ログでこの手続きを行っているのは、以下の部分である。

```
 2642 35.056808    192.168.12.66          192.168.13.80          HTTP      436
GET /~yoshiaki/C1/page1.html HTTP/1.1
  2643 35.057355    192.168.13.80          192.168.12.66          TCP       66
80 → 35516 [ACK] Seq=35308252 Ack=3906406373 Win=64896 Len=0
TSval=1939596896 TSecr=3543356597
  2644 35.058177    192.168.13.80          192.168.12.66          HTTP
627      HTTP/1.1 200 OK  (text/html)
  2645 35.058247    192.168.12.66          192.168.13.80          TCP       66
35516 → 80 [ACK] Seq=3906406373 Ack=35308813 Win=64000 Len=0
TSval=3543356598 TSecr=1939596897
  2707 35.533754    192.168.12.66          192.168.13.80          HTTP
334      GET /favicon.ico HTTP/1.1
  2708 35.534299    192.168.13.80          192.168.12.66          TCP       66
80 → 35516 [ACK] Seq=35308813 Ack=3906406641 Win=64640 Len=0
```

TSval=1939597373 TSecr=3543357073

2709 35.534821 192.168.13.80 192.168.12.66 HTTP 572 HTTP/1.1 404 Not Found (text/html)

2710 35.534867 192.168.12.66 192.168.13.80 TCP 66 35516 → 80 [ACK] Seq=3906406641 Ack=35309319 Win=64000 Len=0 TSval=3543357075 TSecr=1939597373

次に、それぞれのパケットキャプチャログの詳細を抜粋して示す。

・2642

Transmission Control Protocol, Src Port: 35516, Dst Port: 80, Seq: 3906406003, Ack: 35308252, Len: 370

(中略)

Flags: 0x018 (PSH, ACK)

(中略)

Hypertext Transfer Protocol

GET /~yoshiaki/C1/page1.html HTTP/1.1¥r¥n

[Expert Info (Chat/Sequence): GET /~yoshiaki/C1/page1.html HTTP/1.1¥r¥n]

[GET /~yoshiaki/C1/page1.html HTTP/1.1¥r¥n]

[Severity level: Chat]

[Group: Sequence]

Request Method: GET

Request URI: /~yoshiaki/C1/page1.html

Request Version: HTTP/1.1

Host: www2b.comm.eng.osaka-u.ac.jp¥r¥n

User-Agent: Mozilla/5.0 (X11; Linux armv7l; rv:78.0) Gecko/20100101 Firefox/78.0¥r¥n

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8¥r¥n

Accept-Language: en-US,en;q=0.5¥r¥n

Accept-Encoding: gzip, deflate¥r¥n

DNT: 1¥r¥n

Connection: keep-alive¥r¥n

Upgrade-Insecure-Requests: 1¥r¥n

¥r¥n

[Full request URI: http://www2b.comm.eng.osaka-u.ac.jp/~yoshiaki/C1/page1.html]

[HTTP request 1/2]

[Response in frame: 2644]

[Next request in frame: 2707]

・2644

Transmission Control Protocol, Src Port: 80, Dst Port: 35516, Seq: 35308252, Ack: 3906406373, Len: 561

(中略)

Flags: 0x018 (PSH, ACK)

(中略)

Hypertext Transfer Protocol

HTTP/1.1 200 OK¥r¥n

[Expert Info (Chat/Sequence): HTTP/1.1 200 OK¥r¥n]

[HTTP/1.1 200 OK¥r¥n]

[Severity level: Chat]

[Group: Sequence]

Response Version: HTTP/1.1

Status Code: 200

[Status Code Description: OK]

Response Phrase: OK

Date: Tue, 11 May 2021 05:55:53 GMT¥r¥n

Server: Apache/2.4.41 (Ubuntu)¥r¥n

Last-Modified: Thu, 17 Sep 2020 03:12:39 GMT¥r¥n

ETag: "10a-5af79c1bb058b-gzip"¥r¥n

Accept-Ranges: bytes¥r¥n

Vary: Accept-Encoding¥r¥n

Content-Encoding: gzip¥r¥n

Content-Length: 224¥r¥n

[Content length: 224]

Keep-Alive: timeout=5, max=100¥r¥n

Connection: Keep-Alive¥r¥n

Content-Type: text/html¥r¥n

¥r¥n

[HTTP response 1/2]

[Time since request: 0.001369270 seconds]

[Request in frame: 2642]

[Next request in frame: 2707]

[Next response in frame: 2709]

[Request URI: http://www2b.comm.eng.osaka-u.ac.jp/~yoshiaki/C1/page1.html]

Content-encoded entity body (gzip): 224 bytes -> 266 bytes

File Data: 266 bytes

・2707

Transmission Control Protocol, Src Port: 35516, Dst Port: 80, Seq: 3906406373, Ack: 35308813, Len: 268

(中略)

Flags: 0x018 (PSH, ACK)

(中略)

Hypertext Transfer Protocol

GET /favicon.ico HTTP/1.1¥r¥n

[Expert Info (Chat/Sequence): GET /favicon.ico HTTP/1.1¥r¥n]

[GET /favicon.ico HTTP/1.1¥r¥n]

[Severity level: Chat]

[Group: Sequence]

Request Method: GET

Request URI: /favicon.ico

Request Version: HTTP/1.1

Host: www2b.comm.eng.osaka-u.ac.jp¥r¥n

User-Agent: Mozilla/5.0 (X11; Linux armv7l; rv:78.0) Gecko/20100101 Firefox/78.0¥r¥n

Accept: image/webp,*/*¥r¥n

Accept-Language: en-US,en;q=0.5¥r¥n

Accept-Encoding: gzip, deflate¥r¥n

DNT: 1¥r¥n

Connection: keep-alive¥r¥n

¥r¥n

[Full request URI: http://www2b.comm.eng.osaka-u.ac.jp/favicon.ico]

[HTTP request 2/2]

[Prev request in frame: 2642]

[Response in frame: 2709]

・2709

Transmission Control Protocol, Src Port: 80, Dst Port: 35516, Seq: 35308813, Ack: 3906406641, Len: 506

(中略)

Flags: 0x018 (PSH, ACK)

(中略)

Hypertext Transfer Protocol

HTTP/1.1 404 Not Found¥r¥n

[Expert Info (Chat/Sequence): HTTP/1.1 404 Not Found¥r¥n]

[HTTP/1.1 404 Not Found¥r¥n]

[Severity level: Chat]

[Group: Sequence]

Response Version: HTTP/1.1

Status Code: 404

[Status Code Description: Not Found]

```
        Response Phrase: Not Found
    Date: Tue, 11 May 2021 05:55:54 GMT¥r¥n
    Server: Apache/2.4.41 (Ubuntu)¥r¥n
    Content-Length: 290¥r¥n
        [Content length: 290]
    Keep-Alive: timeout=5, max=99¥r¥n
    Connection: Keep-Alive¥r¥n
    Content-Type: text/html; charset=iso-8859-1¥r¥n
    ¥r¥n
    [HTTP response 2/2]
    [Time since request: 0.001067291 seconds]
    [Prev request in frame: 2642]
    [Prev response in frame: 2644]
    [Request in frame: 2707]
    [Request URI: http://www2b.comm.eng.osaka-u.ac.jp/favicon.ico]
    File Data: 290 bytes
Line-based text data: text/html (9 lines)
    <!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">¥n
    <html><head>¥n
    <title>404 Not Found</title>¥n
    </head><body>¥n
    <h1>Not Found</h1>¥n
    <p>The requested URL was not found on this server.</p>¥n
    <hr>¥n
    <address>Apache/2.4.41 (Ubuntu) Server at www2b.comm.eng.osaka-u.ac.jp Port
80</address>¥n
    </body></html>¥n
```

このログから読み取ることができる各パケットの情報を表 11 と表 12 にまとめる。

表 11　HTTP メッセージの送受信の手続きの分析 1

No.	送信元 IP	受信先 IP	送信元	受信先	TCP

	アドレス	アドレス	ポート番号	ポート番号	セグメント長
2642	192.168.12.66	192.168.13.80	35516	80	370
2643	192.168.13.80	192.168.12.66	80	35516	0
2644	192.168.13.80	192.168.12.66	80	35516	561
2645	192.168.12.66	192.168.13.80	35516	80	0
2707	192.168.12.66	192.168.13.80	35516	80	268
2708	192.168.13.80	192.168.12.66	80	35516	0
2709	192.168.13.80	192.168.12.66	80	35516	506
2710	192.168.12.66	192.168.13.80	35516	80	0

表 12　HTTP メッセージの送受信の手続きの分析 2

No.	ON になっている TCP フラグ	シーケンス番号	ACK 番号
2642	PSH, ACK	3906406003	35308252
2643	ACK	35308252	3906406373
2644	PSH, ACK	35308252	3906406373
2645	ACK	3906406373	35308813
2707	PSH, ACK	3906406373	35308813
2708	ACK	35308813	3906406641
2709	PSH, ACK	35308813	3906406641
2710	ACK	3906406641	35309319

アプリケーション層のメッセージの内容を以下に示す。

2642: Web サーバに「HTTP GET メッセージ」を送信して/~yoshiaki/C1/page1.html のページを要求。gzip 形式または deflate 形式での圧縮要求。

2643 : なし

2644 : 「HTTP GET メッセージ」に対する「HTTP Response メッセージ」を送信。http://www2b.comm.eng.osaka-u.ac.jp のソースコードを含む。gzip 形式で圧縮されている。

2645 : なし

2707: Web サーバに「HTTP GET メッセージ」を送信して www2b.comm.eng.osaka-u.ac.jp の favicon.ico を要求している。

2708 : なし

2709 : 「HTTP GET メッセージ」に対する「HTTP Response メッセージ」を送信。

404 Not Found

2710 : なし

No.2642 で HTTP GET により、html を圧縮形式で要求し、No.2644 でその応用として圧縮された html が返された。No.2707 では favicon.ico の取得要求が送られたが、No.2709 で「404 Not Found」が返された。なお、No.2642~2645 は実験課題 2 の No.8354~8357 と同様の役割を持つ。favicon.ico の要求は Firefox による自動送信であり、2.4.3 では見られなかった。

2.5.4 TCP コネクションの終了

ログでこの手続きを行っているのは、以下の部分である。

```
  3222 40.534947    192.168.13.80          192.168.12.66          TCP      66
80 → 35516 [FIN, ACK] Seq=35309319 Ack=3906406641 Win=64640 Len=0
TSval=1939602374 TSecr=3543357075
  3223 40.535294    192.168.12.66          192.168.13.80          TCP      66
35516 → 80 [FIN, ACK] Seq=3906406641 Ack=35309320 Win=64128 Len=0
TSval=3543362075 TSecr=1939602374
  3224 40.535752    192.168.13.80          192.168.12.66          TCP      66
80 → 35516 [ACK] Seq=35309320 Ack=3906406642 Win=64640 Len=0
TSval=1939602374 TSecr=3543362075
```

このログから読み取ることができる各パケットの情報を表 13 と表 14 にまとめる。

表 13　TCP コネクションを終了する手続きの分析 1

No.	送信元 IP アドレス	受信先 IP アドレス	送信元 ポート番号	受信先 ポート番号	TCP セグメント長
3222	192.168.13.80	192.168.12.66	80	35516	0
3223	192.168.12.66	192.168.13.80	35516	80	0
3224	192.168.13.80	192.168.12.66	80	35516	0

表 14　TCP コネクションを終了する手続きの分析 2

No.	ON になっている TCP フラグ	シーケンス番号	ACK 番号
3222	ACK, FIN	35309319	3906406641

3223	ACK, FIN	3906406641	35309320
3224	ACK	35309320	3906406642

全体的な流れは実験課題 2 の No.8358～No.8360 と似ているが、今回は Web サーバが先に FIN を送信している点が大きな違いである。実験課題 2 では、クライアントが通信終了を判断したのに対し、実験課題 3 では favicon.ico の応答後に Web サーバが終了を判断しているためである。なお、この 3 つのパケットにはいずれもアプリケーション層のメッセージは存在しない。

2.6 実験課題 4

この課題では、実験課題 2、実験課題 3 と同様の手順で Web ブラウザ(Firefox)から以下にアクセスした際の通信ログを分析する。

http://www2b.comm.eng.osaka-u.ac.jp/~yoshiaki/C1/page2.html

取得したのは以下のログである。このログを分析し、結果をまとめる。

No. Time Source Destination Protocol Length Info

1019 18.650095 192.168.12.66 10.144.0.1 DNS 88 Standard query 0xfa8f A www2b.comm.eng.osaka-u.ac.jp

1020 18.651080 10.144.0.1 192.168.12.66 DNS 104 Standard query response 0xfa8f A www2b.comm.eng.osaka-u.ac.jp A 192.168.13.80

1032 18.760729 192.168.12.66 192.168.13.80 TCP 74 35530 → 80 [SYN] Seq=268385703 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=3543596857 TSecr=0 WS=128

1033 18.761271 192.168.13.80 192.168.12.66 TCP 74 80 → 35530 [SYN, ACK] Seq=4149200285 Ack=268385704 Win=65160 Len=0 MSS=1460 SACK_PERM=1 TSval=1939837156 TSecr=3543596857 WS=128

1034 18.761396 192.168.12.66 192.168.13.80 TCP 66 35530 → 80 [ACK] Seq=268385704 Ack=4149200286 Win=64256 Len=0 TSval=3543596857 TSecr=1939837156

1035 18.761708 192.168.12.66 192.168.13.80 HTTP

436 GET /~yoshiaki/C1/page2.html HTTP/1.1

1036 18.762236 192.168.13.80 192.168.12.66 TCP 66 80 → 35530 [ACK] Seq=4149200286 Ack=268386074 Win=64896 Len=0 TSval=1939837157 TSecr=3543596858

1037 18.763016 192.168.13.80 192.168.12.66 HTTP 635 HTTP/1.1 200 OK (text/html)

1038 18.763111 192.168.12.66 192.168.13.80 TCP 66 35530 → 80 [ACK] Seq=268386074 Ack=4149200855 Win=64000 Len=0 TSval=3543596859 TSecr=1939837158

1084 19.170575 192.168.12.66 192.168.13.80 HTTP 420 GET /~yoshiaki/C1/helloworld.png HTTP/1.1

1085 19.171240 192.168.13.80 192.168.12.66 TCP 66 80 → 35530 [ACK] Seq=4149200855 Ack=268386428 Win=64640 Len=0 TSval=1939837566 TSecr=3543597267

1086 19.171658 192.168.13.80 192.168.12.66 TCP 1514 80 → 35530 [ACK] Seq=4149200855 Ack=268386428 Win=64640 Len=1448 TSval=1939837566 TSecr=3543597267 [TCP segment of a reassembled PDU]

1087 19.171759 192.168.12.66 192.168.13.80 TCP 66 35530 → 80 [ACK] Seq=268386428 Ack=4149202303 Win=64000 Len=0 TSval=3543597268 TSecr=1939837566

1088 19.171891 192.168.13.80 192.168.12.66 HTTP 659 HTTP/1.1 200 OK (PNG)

1089 19.171954 192.168.12.66 192.168.13.80 TCP 66 35530 → 80 [ACK] Seq=268386428 Ack=4149202896 Win=64000 Len=0 TSval=3543597268 TSecr=1939837566

1116 19.299075 192.168.12.66 192.168.13.80 HTTP 334 GET /favicon.ico HTTP/1.1

1117 19.299679 192.168.13.80 192.168.12.66 TCP 66 80 → 35530 [ACK] Seq=4149202896 Ack=268386696 Win=64384 Len=0 TSval=1939837694 TSecr=3543597395

```
   1118  19.300124      192.168.13.80              192.168.12.66              HTTP
572      HTTP/1.1 404 Not Found  (text/html)
   1119  19.300232      192.168.12.66              192.168.13.80              TCP
66       35530 → 80 [ACK] Seq=268386696 Ack=4149203402 Win=64000 Len=0
TSval=3543597396 TSecr=1939837695
   1643  24.300291      192.168.13.80              192.168.12.66              TCP
66       80 → 35530 [FIN, ACK] Seq=4149203402 Ack=268386696 Win=64384
Len=0 TSval=1939842695 TSecr=3543597396
   1644  24.300712      192.168.12.66              192.168.13.80              TCP
66       35530 → 80 [FIN, ACK] Seq=268386696 Ack=4149203403 Win=64128
Len=0 TSval=3543602397 TSecr=1939842695
   1645  24.301194      192.168.13.80              192.168.12.66              TCP
66       80 → 35530 [ACK] Seq=4149203403 Ack=268386697 Win=64384 Len=0
TSval=1939842696 TSecr=3543602397
```

2.6.1 ドメイン名を IP アドレスに変換

ログでこの手続きを行っているのは、以下の部分である。

```
  1019  18.650095     192.168.12.66              10.144.0.1                 DNS        88
Standard query 0xfa8f A www2b.comm.eng.osaka-u.ac.jp
  1020  18.651080     10.144.0.1                 192.168.12.66              DNS        104
Standard query response 0xfa8f A www2b.comm.eng.osaka-u.ac.jp A 192.168.13.80
```

次に、パケットキャプチャログの詳細を抜粋して示す。

・1019

```
User Datagram Protocol, Src Port: 35759, Dst Port: 53
    Source Port: 35759
    Destination Port: 53
    Length: 54
    Checksum: 0xd7c2 [unverified]
    [Checksum Status: Unverified]
    [Stream index: 15]
```

Domain Name System (query)

Transaction ID: 0xfa8f

Flags: 0x0100 Standard query

(中略)

Queries

www2b.comm.eng.osaka-u.ac.jp: type A, class IN

Name: www2b.comm.eng.osaka-u.ac.jp

[Name Length: 28]

[Label Count: 6]

Type: A (Host Address) (1)

Class: IN (0x0001)

[Response In: 1020]

・1020

User Datagram Protocol, Src Port: 53, Dst Port: 35759

Source Port: 53

Destination Port: 35759

Length: 70

Checksum: 0xb66a [unverified]

[Checksum Status: Unverified]

[Stream index: 15]

Domain Name System (response)

Transaction ID: 0xfa8f

Flags: 0x8580 Standard query response, No error

(中略)

Queries

www2b.comm.eng.osaka-u.ac.jp: type A, class IN

Name: www2b.comm.eng.osaka-u.ac.jp

[Name Length: 28]

[Label Count: 6]

Type: A (Host Address) (1)

```
        Class: IN (0x0001)
    Answers
        www2b.comm.eng.osaka-u.ac.jp: type A, class IN, addr 192.168.13.80
            Name: www2b.comm.eng.osaka-u.ac.jp
            Type: A (Host Address) (1)
            Class: IN (0x0001)
            Time to live: 1 (1 second)
            Data length: 4
            Address: 192.168.13.80
    [Request In: 1019]
    [Time: 0.000984841 seconds]
```

分析結果を表にまとめる。

表 15　ドメイン名を IP アドレスに変換する手続きの分析

No.	送信元 IP アドレス	受信先 IP アドレス	送信元ポート番号	受信先ポート番号	UDP セグメント長
1019	192.168.12.66	10.144.0.1	35759	53	54
1020	10.144.0.1	192.168.12.66	53	35759	70

アプリケーション層のメッセージの内容を以下に示す。

1019 : www2b.comm.eng.osaka-u.ac.jp の Web ページを保持する IP アドレスの問い合わせ

1020 : www2b.comm.eng.osaka-u.ac.jp の Web ページを保持する IP アドレスを送信

以上より、この 2 つのパケットの通信によって、クライアントは入力された Web サーバに対応する IP アドレス(192.168.13.80)を取得することができることがわかる。実験課題 2、実験課題 3 と同じ流れであることが確認できた。

2.6.2 TCP コネクションの確立

ログでこの手続きを行っているのは、以下の部分である。

```
  1032 18.760729     192.168.12.66          192.168.13.80          TCP        74
35530 → 80 [SYN] Seq=268385703 Win=64240 Len=0 MSS=1460 SACK_PERM=1
TSval=3543596857 TSecr=0 WS=128
  1033 18.761271     192.168.13.80          192.168.12.66          TCP        74
```

```
80 → 35530 [SYN, ACK] Seq=4149200285 Ack=268385704 Win=65160 Len=0
MSS=1460 SACK_PERM=1 TSval=1939837156 TSecr=3543596857 WS=128
  1034 18.761396    192.168.12.66          192.168.13.80          TCP       66
35530 → 80 [ACK] Seq=268385704 Ack=4149200286 Win=64256 Len=0
TSval=3543596857 TSecr=1939837156
```

次に、No.1032 のパケットキャプチャログの詳細を抜粋して示す。

```
Transmission Control Protocol, Src Port: 35530, Dst Port: 80, Seq: 268385703, Len: 0
                        (中略)
    Acknowledgment number: 0
```

ログから読み取ることができる情報を表 16、表 17 にまとめる。

表 16　TCP コネクションを確立する手続きの分析 1

No.	送信元 IP アドレス	受信先 IP アドレス	送信元 ポート番号	受信先 ポート番号	TCP セグメント長
1032	192.168.12.66	192.168.13.80	35530	80	0
1033	192.168.13.80	192.168.12.66	80	35016	0
1034	192.168.12.66	192.168.13.80	35016	80	0

表 17　TCP コネクションを確立する手続きの分析 2

No.	ON になっている TCP フラグ	シーケンス番号	ACK 番号
1032	SYN	268385703	0
1033	SYN, ACK	4149200285	268385704
1034	ACK	268385704	4149200286

No.1032 のパケットでは、クライアントが Web サーバに SYN を送信、No.1032 の SYN に対する確認応答(ACK)と SYN が送信されていることがわかる。同様に、No.1033 の SYN に対する ACK が送信されていることがわかる。なお、この 3 つのパケットにはアプリケーション層のメッセージが存在しない。実験課題 2、実験課題 3 と同じ手順であることが確認できる。

2.6.3 HTTP メッセージの送受信

ログでこの手続きを行っているのは、以下の部分である。

```
  1035 18.761708    192.168.12.66    192.168.13.80    HTTP
436    GET /~yoshiaki/C1/page2.html HTTP/1.1
  1036 18.762236    192.168.13.80    192.168.12.66    TCP
66     80 → 35530 [ACK] Seq=4149200286 Ack=268386074 Win=64896 Len=0
TSval=1939837157 TSecr=3543596858
  1037 18.763016    192.168.13.80    192.168.12.66    HTTP
635    HTTP/1.1 200 OK  (text/html)
  1038 18.763111    192.168.12.66    192.168.13.80    TCP
66     35530 → 80 [ACK] Seq=268386074 Ack=4149200855 Win=64000 Len=0
TSval=3543596859 TSecr=1939837158
  1084 19.170575    192.168.12.66    192.168.13.80    HTTP
420    GET /~yoshiaki/C1/helloworld.png HTTP/1.1
  1085 19.171240    192.168.13.80    192.168.12.66    TCP
66     80 → 35530 [ACK] Seq=4149200855 Ack=268386428 Win=64640 Len=0
TSval=1939837566 TSecr=3543597267
  1086 19.171658    192.168.13.80    192.168.12.66    TCP
1514   80 → 35530 [ACK] Seq=4149200855 Ack=268386428 Win=64640
Len=1448 TSval=1939837566 TSecr=3543597267 [TCP segment of a reassembled
PDU]
  1087 19.171759    192.168.12.66    192.168.13.80    TCP
66     35530 → 80 [ACK] Seq=268386428 Ack=4149202303 Win=64000 Len=0
TSval=3543597268 TSecr=1939837566
  1088 19.171891    192.168.13.80    192.168.12.66    HTTP
659    HTTP/1.1 200 OK  (PNG)
  1089 19.171954    192.168.12.66    192.168.13.80    TCP
66     35530 → 80 [ACK] Seq=268386428 Ack=4149202896 Win=64000 Len=0
TSval=3543597268 TSecr=1939837566
  1116 19.299075    192.168.12.66    192.168.13.80    HTTP
334    GET /favicon.ico HTTP/1.1
  1117 19.299679    192.168.13.80    192.168.12.66    TCP
```

66 80 → 35530 [ACK] Seq=4149202896 Ack=268386696 Win=64384 Len=0 TSval=1939837694 TSecr=3543597395

1118 19.300124 192.168.13.80 192.168.12.66 HTTP 572 HTTP/1.1 404 Not Found (text/html)

1119 19.300232 192.168.12.66 192.168.13.80 TCP 66 35530 → 80 [ACK] Seq=268386696 Ack=4149203402 Win=64000 Len=0 TSval=3543597396 TSecr=1939837695

次に、それぞれのパケットキャプチャログの詳細を抜粋して示す。

・1035

Transmission Control Protocol, Src Port: 35530, Dst Port: 80, Seq: 268385704, Ack: 4149200286, Len: 370

(中略)

Flags: 0x018 (PSH, ACK)

(中略)

Hypertext Transfer Protocol

GET /~yoshiaki/C1/page2.html HTTP/1.1¥r¥n

(中略)

Host: www2b.comm.eng.osaka-u.ac.jp¥r¥n

User-Agent: Mozilla/5.0 (X11; Linux armv7l; rv:78.0) Gecko/20100101 Firefox/78.0¥r¥n

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8¥r¥n

Accept-Language: en-US,en;q=0.5¥r¥n

Accept-Encoding: gzip, deflate¥r¥n

DNT: 1¥r¥n

Connection: keep-alive¥r¥n

Upgrade-Insecure-Requests: 1¥r¥n

¥r¥n

[Full request URI: http://www2b.comm.eng.osaka-u.ac.jp/~yoshiaki/C1/page2.html]

[HTTP request 1/3]

[Response in frame: 1037]

[Next request in frame: 1084]

・1037

Transmission Control Protocol, Src Port: 80, Dst Port: 35530, Seq: 4149200286, Ack: 268386074, Len: 569

(中略)

Flags: 0x018 (PSH, ACK)

(中略)

Hypertext Transfer Protocol

HTTP/1.1 200 OK¥r¥n

(中略)

Date: Tue, 11 May 2021 05:59:53 GMT¥r¥n

Server: Apache/2.4.41 (Ubuntu)¥r¥n

Last-Modified: Thu, 17 Sep 2020 03:12:39 GMT¥r¥n

ETag: "119-5af79c1bbefeb-gzip"¥r¥n

Accept-Ranges: bytes¥r¥n

Vary: Accept-Encoding¥r¥n

Content-Encoding: gzip¥r¥n

Content-Length: 232¥r¥n

[Content length: 232]

Keep-Alive: timeout=5, max=100¥r¥n

Connection: Keep-Alive¥r¥n

Content-Type: text/html¥r¥n

¥r¥n

・1084

Transmission Control Protocol, Src Port: 35530, Dst Port: 80, Seq: 268386074, Ack: 4149200855, Len: 354

(中略)

Flags: 0x018 (PSH, ACK)

(中略)

Hypertext Transfer Protocol

GET /~yoshiaki/C1/helloworld.png HTTP/1.1¥r¥n

(中略)

Host: www2b.comm.eng.osaka-u.ac.jp¥r¥n

User-Agent: Mozilla/5.0 (X11; Linux armv7l; rv:78.0) Gecko/20100101 Firefox/78.0¥r¥n

Accept: image/webp,*/*¥r¥n

Accept-Language: en-US,en;q=0.5¥r¥n

Accept-Encoding: gzip, deflate¥r¥n

DNT: 1¥r¥n

Connection: keep-alive¥r¥n

Referer: http://www2b.comm.eng.osaka-u.ac.jp/~yoshiaki/C1/page2.html¥r¥n

¥r¥n

[Full request URI: http://www2b.comm.eng.osaka-u.ac.jp/~yoshiaki/C1/helloworld.png]

[HTTP request 2/3]

[Prev request in frame: 1035]

[Response in frame: 1088]

[Next request in frame: 1116]

・1088

Transmission Control Protocol, Src Port: 80, Dst Port: 35530, Seq: 4149202303, Ack: 268386428, Len: 593

(中略)

Flags: 0x018 (PSH, ACK)

(中略)

Hypertext Transfer Protocol

HTTP/1.1 200 OK¥r¥n

(中略)

Date: Tue, 11 May 2021 05:59:54 GMT¥r¥n

Server: Apache/2.4.41 (Ubuntu)¥r¥n

Last-Modified: Thu, 17 Sep 2020 03:12:39 GMT¥r¥n

ETag: "6dc-5af79c1ba3a6b"¥r¥n

Accept-Ranges: bytes¥r¥n

Content-Length: 1756¥r¥n

[Content length: 1756]

Keep-Alive: timeout=5, max=99¥r¥n

Connection: Keep-Alive¥r¥n

Content-Type: image/png¥r¥n

¥r¥n

・1116

Transmission Control Protocol, Src Port: 35530, Dst Port: 80, Seq: 268386428, Ack: 4149202896, Len: 268

(中略)

Flags: 0x018 (PSH, ACK)

(中略)

Hypertext Transfer Protocol

GET /favicon.ico HTTP/1.1¥r¥n

(中略)

Host: www2b.comm.eng.osaka-u.ac.jp¥r¥n

User-Agent: Mozilla/5.0 (X11; Linux armv7l; rv:78.0) Gecko/20100101 Firefox/78.0¥r¥n

Accept: image/webp,*/*¥r¥n

Accept-Language: en-US,en;q=0.5¥r¥n

Accept-Encoding: gzip, deflate¥r¥n

DNT: 1¥r¥n

Connection: keep-alive¥r¥n

¥r¥n

[Full request URI: http://www2b.comm.eng.osaka-u.ac.jp/favicon.ico]

[HTTP request 3/3]

[Prev request in frame: 1084]

[Response in frame: 1118]

・1118

Transmission Control Protocol, Src Port: 80, Dst Port: 35530, Seq: 4149202896, Ack: 268386696, Len: 506

(中略)

Flags: 0x018 (PSH, ACK)

(中略)

Hypertext Transfer Protocol

HTTP/1.1 404 Not Found¥r¥n

[Expert Info (Chat/Sequence): HTTP/1.1 404 Not Found¥r¥n]

[HTTP/1.1 404 Not Found¥r¥n]

[Severity level: Chat]

[Group: Sequence]

Response Version: HTTP/1.1

Status Code: 404

[Status Code Description: Not Found]

Response Phrase: Not Found

Date: Tue, 11 May 2021 05:59:54 GMT¥r¥n

Server: Apache/2.4.41 (Ubuntu)¥r¥n

Content-Length: 290¥r¥n

[Content length: 290]

Keep-Alive: timeout=5, max=98¥r¥n

Connection: Keep-Alive¥r¥n

Content-Type: text/html; charset=iso-8859-1¥r¥n

このログから読み取ることができる各パケットの情報を表 18 と表 19 にまとめる。

表 18 HTTP メッセージの送受信の手続きの分析 1

No.	送信元 IP アドレス	受信先 IP アドレス	送信元 ポート番号	受信先 ポート番号	TCP セグメント長
1035	192.168.12.66	192.168.13.80	35530	80	370
1036	192.168.13.80	192.168.12.66	80	35530	0
1037	192.168.13.80	192.168.12.66	80	35530	569
1038	192.168.12.66	192.168.13.80	35530	80	0
1084	192.168.12.66	192.168.13.80	35530	80	354
1085	192.168.13.80	192.168.12.66	80	35530	0
1086	192.168.13.80	192.168.12.66	80	35530	1448
1087	192.168.12.66	192.168.13.80	35530	80	0
1088	192.168.13.80	192.168.12.66	80	35530	593
1089	192.168.12.66	192.168.13.80	35530	80	0
1116	192.168.12.66	192.168.13.80	35530	80	268
1117	192.168.13.80	192.168.12.66	80	35530	0
1118	192.168.13.80	192.168.12.66	80	35530	506
1119	192.168.12.66	192.168.13.80	35530	80	0

表 19　HTTP メッセージの送受信の手続きの分析 2

No.	ON になっている TCP フラグ	シーケンス番号	ACK 番号
1035	PSH, ACK	268385704	4149200286
1036	ACK	4149200286	268386074
1037	PSH, ACK	4149200286	268386074
1038	ACK	268386074	4149200855
1084	PSH, ACK	268386074	4149200855
1085	ACK	4149200855	268386428
1086	PSH, ACK	4149200855	268386428
1087	ACK	268386428	4149202303
1088	PSH, ACK	4149202303	268386428
1089	ACK	268386428	4149202896
1116	PSH, ACK	268386428	4149202896

1117	ACK	4149202896	268386696
1118	PSH, ACK	4149202896	268386696
1119	ACK	268386696	4149203402

アプリケーション層のメッセージの内容を以下に示す。

1035: Web サーバに「HTTP GET メッセージ」を送信し html 要求。

1036 : なし

1037 : 「HTTP GET メッセージ」に対する「HTTP Response メッセージ」を送信。
http://www2b.comm.eng.osaka-u.ac.jp のソースコードを含む。

1038 : なし

1084: Web サーバに「HTTP GET メッセージ」を送信して/~yoshiaki/C1/helloworld.png の画像ファイルを gzip または deflate で圧縮することを要求している。

1085 : なし

1086 : なし

1087 : なし

1088 : 「HTTP GET メッセージ」に対する「HTTP Response メッセージ」を送信。
/~yoshiaki/C1/helloworld.png の HTML ファイルを含んでいる。

1089 : なし

1116: Web サーバに「HTTP GET メッセージ」を送信して favicon.ico を要求。

1117 : なし

1118 : 「HTTP GET メッセージ」に対する「HTTP Response メッセージ」を送信。
404 Not Found (favicon.ico が見つからなかった)

1119 : なし

No.1035~No.1038 は実験課題 2・3 と同様の通信である。No.1084～No.1089 では、画像ファイル”helloworld.png”を要求・取得しており、これは画像を含むページ特有の通信である。No.1116～No.1119 では favicon.ico の取得を試みたが失敗しており、実験課題 3 と同じである。

2.5.4 TCP コネクションの終了

ログでこの手続きを行っているのは、以下の部分である。

```
  1643 24.300291    192.168.13.80          192.168.12.66          TCP       66
80 → 35530 [FIN, ACK] Seq=4149203402 Ack=268386696 Win=64384 Len=0
```

```
TSval=1939842695 TSecr=3543597396
  1644 24.300712    192.168.12.66          192.168.13.80          TCP      66
35530 → 80 [FIN, ACK] Seq=268386696 Ack=4149203403 Win=64128 Len=0
TSval=3543602397 TSecr=1939842695
  1645 24.301194    192.168.13.80          192.168.12.66          TCP      66
80 → 35530 [ACK] Seq=4149203403 Ack=268386697 Win=64384 Len=0
TSval=1939842696 TSecr=3543602397
```

このログから読み取ることができる各パケットの情報を表 20 と表 21 にまとめる。

表 20　TCP コネクションを終了する手続きの分析 1

No.	送信元 IP アドレス	受信先 IP アドレス	送信元 ポート番号	受信先 ポート番号	TCP セグメント長
1643	192.168.13.80	192.168.12.66	80	35530	0
1644	192.168.12.66	192.168.13.80	35530	80	0
1645	192.168.13.80	192.168.12.66	80	35530	0

表 21　TCP コネクションを終了する手続きの分析 2

No.	ON になっている TCP フラグ	シーケンス番号	ACK 番号
1643	ACK, FIN	4149203402	268386696
1644	ACK, FIN	268386696	4149203403
1645	ACK	4149203403	268386697

No.1643 から No.1645 のパケットは、実験課題 2、実験課題 3 と同じ処理を行なっていることがわかる。この 3 つのパケットにはいずれもアプリケーション層のメッセージは存在しない。

3.第二週：Web アプリケーションにおける輻輳現象の分析

画像識別を行う Web アプリケーションを題材として、情報システムにおいて生じる輻輳現象を観察する。クライアントと Web サーバは LAN ケーブルとスイッチングハブを介して接続しており、Web サーバ上では画像識別を行う Web アプリケーションが動作している。クライアントから Web サーバへ HTTP POST メッセージとして画像ファイルを送信すると、Web サーバは「ResNet

V2 101」と呼ばれる深層ニューラルネットワークを使用して画像のラベルを識別し、識別結果を HTTP Response メッセージとしてクライアントに返す。今回実験対象とするシステムの概略図は以下の図 2 である。

図 2　第 2 週で用いるシステムの模式図

このシステムでは、クライアントが画像を送信可能である最小間隔と比べて、Web サーバが画像識別に要する時間の方がはるかに大きいという特徴がある。したがって、クライアントが最大の送信頻度で画像を送り続けると Web サーバの処理能力を超過してしまい、遅延時間が極めて増大するという結果を招く。本実験では、このような状況において、送信頻度が遅延時間に与える影響、並びに遅延時間との適切な兼ね合いをとることができる適切な送信頻度の決定方法について、実測に基づいた考察を行う。

3.1 理論

本実験のシステムは、待ち行列モデルを用いてモデル化される。本実験で考える単一サーバ待ち行列モデルは、待合室と単一のサーバで構成され、外部から次々に到着するジョブをサーバが順番に処理する。このモデルの模式図を図 3 に示す。

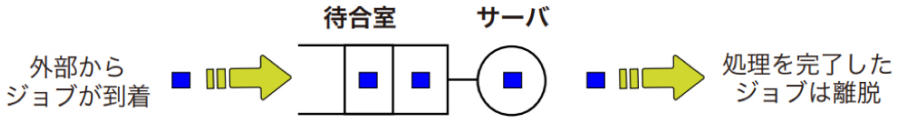

図 3　単一サーバ待ち行列モデル

単一サーバ待ち行列モデルは、具体的には以下の手続きに従って動作する。

(a)ある頻度で、外部からジョブが待合室に到着する。

(a-1)ジョブの到着時点においてサーバが空いていた場合、

到着ジョブは即座にサーバで処理を受け始める。

(a-2)ジョブの到着時点において他のジョブが処理中であった場合、

到着ジョブは待合室で待機する。

(b)ジョブの処理が完了した時点において、

(b-1)待合室にジョブが存在する場合、サーバは次のジョブの処理を開始する。

(b-2)待合室にジョブが存在しない場合、サーバは次のジョブの到着まで待機する。

本実験においては、待ち行列モデルにおける「待合室」および「サーバ」は、Web サーバ上に存在する「バッファ」および「演算装置(CPU)」に相当する。また、「外部」とは Web サーバの外側のことを表し、実際には到着ジョブはクライアントが生成してサーバに送出したものである。

時刻 0 に最初のジョブが到着し、その後$\mathrm{N}-1$個のジョブが到着するものとする。n番目($\mathrm{n}=0,1,\cdots,\mathrm{N}-1$)のジョブの到着時刻を$\alpha_n(\alpha_{n-1}<\alpha_n)$とする。ただし、0 番目に到着したジョブについては、$\alpha_0=0$とする。G_nを$\mathrm{n}-1$番目とn番目のジョブの到着間隔、すなわち式(1)のように定義する。

$$G_n=\alpha_n-\alpha_{n-1}\ (n=1,2,\cdots,\mathrm{N}-1) \qquad (1)$$

次に、n番目($\mathrm{n}=0,1,\cdots$)のジョブがサービスを完了する時刻をβ_nとし、そのジョブの遅延時間をD_nと定義する。なお、遅延時間はジョブが到着してからサービスを完了するまでにかかる時間を表すため、式(2)のように定義できる。

$$D_n=\beta_n-\alpha_n\ (n=1,2,\cdots,\mathrm{N}-1) \qquad (2)$$

W_nを他のジョブの処理が完了するまでの待合室での待ち時間、H_nをサーバにおけるそのジョブの処理時間を表す。この 2 つを用いて遅延時間を表すと、式(3)のようになる。

$$D_n=W_n+H_n\ \ (n=1,2,\cdots,\mathrm{N}-1) \qquad (3)$$

先着順処理の単一サーバ待ち行列では、次の漸化式(Lindley 方程式)が成立する。

$$D_n=\max(0,D_{n-1}-G_n)+H_n\ (n=1,2,\cdots,\mathrm{N}-1) \qquad (4)$$

式(4)を変形することにより、以下の式(5)が得られる。

$$D_n=\max(G_n,D_{n-1})+H_n-G_n\ (n=1,2,\cdots,\mathrm{N}-1) \qquad (5)$$

この式において$G_n\geq D_{n-1}$となるとき、n番目のジョブは待ち時間なく処理を開始できるため、式(6)が成り立つ。

$$D_n=H_n \qquad (6)$$

一方、$G_n<D_{n-1}$となるときはn番目のジョブは 1 つ前のジョブの処理が完了するのを待つ必要があるため、式(7)のようになる。

$$D_n=D_{n-1}+H_n-G_n \qquad (7)$$

式(7)より、$G_n < D_{n-1}$の場合には一つ前のジョブの遅延時間D_{n-1}と比較して、遅延時間D_nは$H_n - G_n$だけ大きくなることがわかる。

従って、ジョブの処理時間H_nと比べてジョブの到着間隔G_nが小さい状況が継続した場合、遅延時間D_nは際限なく増加し続ける。

$\mathrm{A}(t)(t \geq 0)$を時刻tまでに到着したジョブの総数と定義すると、式(8)が成り立つ。ただし、$\mathbb{I}(A)$は、Aが真であるときに 1、偽であるときに 0 をとる関数(指示関数)を表す。

$$\mathrm{A}(t) = \sum_{n=0}^{N-1} \mathbb{I}\{\alpha_n \leq t\} \quad (8)$$

同様に、$\mathrm{D}(t)(t \geq 0)$を時刻tまでにサービスを完了して離脱したジョブの総数と定義する。

$$\mathrm{D}(t) = \sum_{n=0}^{N-1} \mathbb{I}\{\beta_n \leq t\} \quad (9)$$

時刻t において、待合室やサーバに滞在しているジョブの総数$\mathrm{L}(t)$は式(10)のように表される。

$$\mathrm{L}(t) = \mathrm{A}(t) - \mathrm{D}(t) \ \ (\mathrm{t} \geq 0) \quad (10)$$

$\mathrm{T}(0 \leq T \leq \beta_N)$を最後のジョブが離脱するまでのある一時刻とし、時間$(0, T]$における平均到着率$\bar{\lambda}(T)$と平均離脱率$\bar{\mu}(T)$をそれぞれ式(11)、(12)で定義する。なお、$\bar{\lambda}(T)$は単位時間当たりの到着数、$\bar{\mu}(T)$は単位時間当たりの離脱数を表す。

$$\bar{\lambda}(T) = \frac{A(T) - 1}{T} \quad (11)$$

$$\bar{\mu}(T) = \frac{D(T)}{T} \quad (12)$$

平均到着率$\bar{\lambda}(T)$は到着間隔$G_n(n = 1, 2, \cdots, \mathrm{N} - 1)$の平均と式(13)のように関係付けられる。以下の議論では、$\mathrm{T} \geq \alpha_1$を仮定する。平均間隔$\bar{\tau}(T)$を時間$(0, T]$に発生した到着の平均間隔とすると、式(13)のように定義できる。

$$\bar{\tau}(T) = \frac{1}{A(T) - 1} \sum_{n=1}^{A(T)-1} G_n \quad (13)$$

$\mathrm{A(T)} \leq \mathrm{N} - 2$のとき、次の不等式が成り立つ。

$$\sum_{n=1}^{A(T)-1} G_n \leq T < \sum_{n=1}^{A(T)-1} G_n + G_{A(T)} \quad (14)$$

また、$\mathrm{A(T)} \leq \mathrm{N} - 2$のとき、次の不等式が成り立つ。

$$\bar{\tau}(T) \leq \frac{1}{\bar{\lambda}(T)} < \bar{\tau}(T) + \frac{G_{A(T)}}{A(T) - 1} \quad (15)$$

A(T)が十分に大きいとき、式(15)の最右辺における第二項は無視できるほど小さいとみなして良いため、式(16)が得られる。このとき、平均到着率は平均到着間隔の逆数にほぼ等しくなる。

$$\bar{\lambda}(T) \cong \frac{1}{\bar{\tau}(T)} \quad (16)$$

一方、平均離脱率は到着間隔G_nと処理時間H_nの両方に依存する。到着ジョブの負荷に対してサーバの処理能力より十分高いとき、すなわち、処理時間が到着間隔に比べて小さいとき、時刻Tでの滞留ジョブ数L(T)は 0 に近似できる。このとき、A(T) ≅ $D(T)$となるため、式(17)が成り立つ。

$$\bar{\mu}(T) \cong \bar{\lambda}(T) \quad (17)$$

この式はサーバの処理能力が到達ジョブの負荷と比べて高いときにのみ成り立つ。

ここで、時刻$[0,T]$における平均処理時間を$\bar{B}(T)$とし、以下の式(18)で定義する。ただし、D(T) ≥ 1を仮定する。

$$\bar{B}(T) = \frac{1}{D(T)} \sum_{n-1}^{D(T)-1} H_n \quad (18)$$

到着ジョブの負荷がサーバの処理能力を超えてしまった場合、つまり処理時間が到着間隔に比べて大きいときはサーバが常にジョブを処理している状態となる。そのため時刻t ∈ $[0,T]$において滞留ジョブ数L(t)は 1 以上の値をとる。ジョブの離脱間隔は処理時間と等しくなるため、式(19)が成り立つ。

$$\bar{\mu}(T) \cong \frac{1}{\bar{B}(T)} \quad (19)$$

ただしこの式は、サーバの処理能力が到達ジョブの負荷より低いときにのみ成り立つ。

以上より平均離脱率$\bar{\mu}(T)$は以下のような特徴をもつ。

・サーバの処理能力が到着ジョブの負荷と比べて高い
　→平均離脱率$\bar{\mu}(T)$と平均到着率$\bar{\lambda}(T)$はほぼ等しい
・サーバの処理能力が到着ジョブの負荷と比べて低い
　→平均離脱率$\bar{\mu}(T)$と平均処理時間$\bar{B}(T)$の逆数はほぼ等しい

式(10)、(11)、(12)より、時刻Tにおける滞留ジョブ数L(T)は式(20)で与えられる。

$$\mathrm{L}(T) = \left(\bar{\lambda}(T) - \bar{\mu}(T)\right) \times T \quad (20)$$

以上より、滞留ジョブ数L(T)について次の結論を得る。

・サーバの処理能力が到着ジョブの負荷と比べて高い
　→滞留ジョブ数は比較的小さな値にとどまる。
・サーバの処理能力が到着ジョブの負荷と比べて低い
　→滞留ジョブ数は時間経過とともに際限なく増加する。

3.2 演習課題 1

先着順処理の単一サーバ待ち行列では、次の漸化式(Lindley 方程式)が成立する。

$$D_n = \max(0, D_{n-1} - G_n) + H_n \quad (n = 1, 2, \cdots, \mathrm{N} - 1)$$

この定理を証明する。

(i)n番目のジョブの到着時点においてサーバが空いていた場合

到着後すぐにサーバで処理されることとなり待ち時間は 0 なので

$$W_n = 0$$

(ii) n番目のジョブの到着時点において他のジョブが処理中であった場合

n番目のジョブは$\mathrm{n} - 1$番目のジョブの処理が終了するまで待合室で待機することになる。よって待機時間は、n 番目のジョブが到着してから n-1 番目に到着したジョブの処理が終わるまでの時間と一致するため、以下のように表すことができる。

$$W_n = \beta_{n-1} - \alpha_n = (\beta_{n-1} - \alpha_{n-1}) - (\alpha_n - \alpha_{n-1})$$

式(1)、(2)より、W_nは到着間隔G_nと遅延時間D_nを用いて以下のように表すことができる。

$$W_n = D_{n-1} - G_n$$

また、$W_n > 0$より$D_{n-1} - G_n > 0$である。

(i),(ii)より、

$$W_n = \max(0, D_{n-1} - G_n)$$

である。式(3)より、

$$D_n = \max(0, D_{n-1} - G_n) + H_n \quad (n = 1, 2, \cdots, \mathrm{N} - 1)$$

3.3.1 演習課題 2-補題 1 の証明

$\mathrm{A(T)} \leq \mathrm{N} - 2$のとき、次の不等式が成り立つ。

$$\sum_{n=1}^{A(T)-1} G_n \leq T < \sum_{n=1}^{A(T)-1} G_n + G_{A(T)}$$

この補題を証明する。

まず、最左辺について考える。式(1)で表したG_nの定義より、

$$\sum_{n=1}^{A(T)-1} G_n = \sum_{n=1}^{A(T)-1} \alpha_n - \alpha_{n-1} = (\alpha_1 - \alpha_0) + (\alpha_2 - \alpha_1) + \cdots + \left(\alpha_{A(T)-1} - \alpha_{A(T)-2}\right)$$

定義より、$\alpha_0 = 0$なので、

$$\sum_{n=1}^{A(T)-1} G_n = \alpha_{A(T)-1} - \alpha_0 = \alpha_{A(T)-1}$$

不等式の最右辺についても同様に考える。$G_{A(T)} = \alpha_{A(T)} - \alpha_{A(T)-1}$より

$$\sum_{n=1}^{A(T)-1} G_n + G_{A(T)} = \alpha_{A(T)-1} + \left(\alpha_{A(T)} - \alpha_{A(T)-1}\right) = \alpha_{A(T)}$$

以上より、補題を示すためには以下の不等式(22)が成り立つことを示せばよい。

$$\alpha_{A(T)-1} \leq T < \alpha_{A(T)} \qquad (22)$$

$A(T)$は時間Tまでに到着したジョブの総数であるため、0番目から$(A(T)-1)$番目のジョブは時間Tよりはやく到着していることになる。従って、$(A(T)-1)$番目のジョブの到着時間$\alpha_{A(T)-1}$はT以下になる。しかし、$A(T)$番目以降のジョブは時間Tより遅く到着する。そのため、$\alpha_{A(T)}$はTよりも大きくなる。

以上より、不等式(22)は成り立つ。よって、補題 1 の不等式も成立する。

3.3.2 演習課題 2-定理 2 の証明

$\mathrm{A(T)} \leq \mathrm{N} - 2$のとき、次の不等式が成り立つ。

$$\bar{\tau}(T) \leq \frac{1}{\bar{\lambda}(T)} < \bar{\tau}(T) + \frac{G_{A(T)}}{A(T)-1}$$

この定理を証明する。

補題 1 において示した不等式の各辺を$A(T)-1$で割ると

$$\frac{1}{A(T)-1}\sum_{n=1}^{A(T)-1} G_n \leq \frac{T}{A(T)-1} < \frac{1}{A(T)-1}\sum_{n=1}^{A(T)-1} G_n + \frac{G_{A(T)}}{A(T)-1}$$

式(11)で表した平均到着率$\bar{\lambda}(T)$の定義と、式(13)で表した時間$(0,T]$に発生した到着の平均間隔$\bar{\tau}(T)$の定義より

$$\bar{\tau}(T) \leq \frac{1}{\bar{\lambda}(T)} < \bar{\tau}(T) + \frac{G_{A(T)}}{A(T)-1}$$

となる。定理 2 の不等式も示された。

3.3 実験環境

クライアント側ハードウェア

・Raspberry Pi 3 Model B+

クライアント側ソフトウェア

・Ubuntu 20.04 LTS

・Python 3.8.2

サーバ側ハードウェア

・HP ProDesk 400 G4 DM/CT

サーバ側ソフトウェア

・Ubuntu 20.04 LTS

・TensorFlow 2.3.1

・TFServe 0.3, ResNet V

・Python 3.8.2

3.4　実験課題 5

この課題では、クライアントでターミナルを開き、ホームディレクトリ直下の client ディレクトリに移動する。その後、以下のコマンドを実行した。

```
$ python3 client.py
```

実行結果は以下のようになった。

```
ubuntu@raspberrypi3:~/client $ python3 client.py
converted_500/test_324.png
[{'class': 'German shepherd, German shepherd dog, German police dog, alsatian',
'prob': 21.546550750732422}, {'class': 'kelpie', 'prob': 17.637371063232422}, {'class':
'toy terrier', 'prob': 14.025994300842285}, {'class': 'malinois', 'prob':
11.318921089172363}]
```

実際に送信されたサーバへ送信された画像は test_324.png である。この画像を図 4 に示す。

図 4　サーバに送信された画像(test_324.png)

実行結果より、送信された画像はゲルマンシェパードである可能性が最も高いとわかる。test_324.png も犬の画像であることから識別結果は正しいと考えることができる。

3.5　実験課題 6

この課題では、実験課題 2 と同様の手順で、実験課題 5 のコマンドを入力したときに行われた一連の通信に関するログを取得する。手順は以下に示す。

1. クライアントで Wireshark を起動し、パケットキャプチャを開始。
2. 実験課題 5 と同じコマンドを実行。
3. Web ページのソースコードが標準出力に表示されたら、パケットキャプチャを停止。

実行結果は以下のようになった。

No.	Time	Source	Destination	Protocol	Length	Info
2140	18.969328	192.168.12.66	10.144.0.1	DNS	92	Standard query 0x3ba2 A exp1.rnea.comm.eng.osaka-u.ac.jp
2141	18.969419	192.168.12.66	10.144.0.1	DNS	92	Standard query 0x43a2 AAAA exp1.rnea.comm.eng.osaka-u.ac.jp
2142	18.970377	10.144.0.1	192.168.12.66	DNS	108	Standard query response 0x3ba2 A exp1.rnea.comm.eng.osaka-u.ac.jp A 192.168.12.23
2143	18.971896	10.144.0.1	192.168.12.66	DNS	144	Standard query response 0x43a2 No such name AAAA exp1.rnea.comm.eng.osaka-u.ac.jp SOA gene.comm.eng.osaka-u.ac.jp
2144	18.972480	192.168.12.66	192.168.12.23	TCP	74	42522 → 5000 [SYN] Seq=4044280001 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=3712487604 TSecr=0 WS=128
2145	18.974209	192.168.12.23	192.168.12.66	TCP	74	5000 → 42522 [SYN, ACK] Seq=1438445294 Ack=4044280002 Win=65160 Len=0 MSS=1460 SACK_PERM=1 TSval=2689562944 TSecr=3712487604 WS=128
2146	18.974294	192.168.12.66	192.168.12.23	TCP	66	42522 → 5000 [ACK] Seq=4044280002 Ack=1438445295 Win=64256 Len=0 TSval=3712487606 TSecr=2689562944
2147	18.974476	192.168.12.66	192.168.12.23	IPA	227	unknown 0x53
2148	18.974618	192.168.12.66	192.168.12.23	IPA	1514	unknown 0x4e
2149	18.974640	192.168.12.66	192.168.12.23	IPA	1514	unknown 0x6e
2150	18.974654	192.168.12.66	192.168.12.23	IPA	1514	unknown 0xbc
2151	18.974667	192.168.12.66	192.168.12.23	IPA	1514	unknown 0x23

2152 18.974681 192.168.12.66 192.168.12.23 IPA 1514 unknown 0xb5

2153 18.974734 192.168.12.66 192.168.12.23 RSL 1514 ip.access PDCH DEACTIVATION

2154 18.974761 192.168.12.66 192.168.12.23 IPA 1514 unknown 0x61

2155 18.974775 192.168.12.66 192.168.12.23 IPA 1514 unknown 0x87

2156 18.974788 192.168.12.66 192.168.12.23 IPA 520 OSMO EXT unknown 0xe6

2157 18.975679 192.168.12.23 192.168.12.66 TCP 66 5000 → 42522 [ACK] Seq=1438445295 Ack=4044280163 Win=65024 Len=0 TSval=2689562946 TSecr=3712487606

2158 18.975684 192.168.12.23 192.168.12.66 TCP 66 5000 → 42522 [ACK] Seq=1438445295 Ack=4044285955 Win=61312 Len=0 TSval=2689562946 TSecr=3712487606

2159 18.975688 192.168.12.23 192.168.12.66 TCP 66 5000 → 42522 [ACK] Seq=1438445295 Ack=4044287403 Win=60288 Len=0 TSval=2689562946 TSecr=3712487606

2160 18.975693 192.168.12.23 192.168.12.66 TCP 66 5000 → 42522 [ACK] Seq=1438445295 Ack=4044288851 Win=59392 Len=0 TSval=2689562946 TSecr=3712487606

2161 18.975920 192.168.12.23 192.168.12.66 TCP 66 5000 → 42522 [ACK] Seq=1438445295 Ack=4044290299 Win=58368 Len=0 TSval=2689562946 TSecr=3712487606

2162 18.975924 192.168.12.23 192.168.12.66 TCP 66 5000 → 42522 [ACK] Seq=1438445295 Ack=4044291747 Win=57472 Len=0 TSval=2689562946 TSecr=3712487606

2163 18.975929 192.168.12.23 192.168.12.66 TCP 66 5000 → 42522 [ACK] Seq=1438445295 Ack=4044292201 Win=57088 Len=0 TSval=2689562946 TSecr=3712487606

2167 19.161365 192.168.12.23 192.168.12.66 IPA 83 unknown 0x54

2168 19.161501 192.168.12.66 192.168.12.23 TCP 66 42522 → 5000 [ACK] Seq=4044292201 Ack=1438445312 Win=64256 Len=0 TSval=3712487793 TSecr=2689563131

```
   2169 19.161369     192.168.12.23          192.168.12.66          IPA        434
unknown 0x6e
   2171 19.163987     192.168.12.66          192.168.12.23          TCP        66
42522 → 5000 [FIN, ACK] Seq=4044292201 Ack=1438445681 Win=64128 Len=0
TSval=3712487796 TSecr=2689563131
   2172 19.165384     192.168.12.23          192.168.12.66          TCP        66
5000 → 42522 [ACK] Seq=1438445681 Ack=4044292202 Win=64128 Len=0
TSval=2689563135 TSecr=3712487796
```

3.5.1 ドメイン名を IP アドレスに変換

ログでこの手続きを行っているのは、以下の部分である。

```
 2140 18.969328     192.168.12.66          10.144.0.1             DNS        92
Standard query 0x3ba2 A exp1.rnea.comm.eng.osaka-u.ac.jp
 2142 18.970377     10.144.0.1             192.168.12.66          DNS        108
Standard query response 0x3ba2 A exp1.rnea.comm.eng.osaka-u.ac.jp A 192.168.12.23
```

次に、No.2140 のパケットキャプチャログの詳細を抜粋して示す。

```
User Datagram Protocol, Src Port: 34238, Dst Port: 53
    Source Port: 34238
    Destination Port: 53
    Length: 58
                                        (中略)
    Queries
        exp1.rnea.comm.eng.osaka-u.ac.jp: type A, class IN
            Name: exp1.rnea.comm.eng.osaka-u.ac.jp
            [Name Length: 32]
            [Label Count: 7]
            Type: A (Host Address) (1)
            Class: IN (0x0001)
```

同様に、No.2142 のパケットキャプチャログの詳細を抜粋して示す。

```
User Datagram Protocol, Src Port: 53, Dst Port: 34238
    Source Port: 53
```

Destination Port: 34238

Length: 74

(中略)

Queries

exp1.rnea.comm.eng.osaka-u.ac.jp: type A, class IN

Name: exp1.rnea.comm.eng.osaka-u.ac.jp

[Name Length: 32]

[Label Count: 7]

Type: A (Host Address) (1)

Class: IN (0x0001)

Answers

exp1.rnea.comm.eng.osaka-u.ac.jp: type A, class IN, addr 192.168.12.23

Name: exp1.rnea.comm.eng.osaka-u.ac.jp

Type: A (Host Address) (1)

Class: IN (0x0001)

Time to live: 2525 (42 minutes, 5 seconds)

Data length: 4

Address: 192.168.12.23

分析結果を表 22 にまとめた。

表 22　ドメイン名を IP アドレスに変換する手続きの分析

No.	送信元 IP アドレス	受信先 IP アドレス	送信元ポート番号	受信先ポート番号	UDP セグメント長
2140	192.168.12.66	10.144.0.1	34238	53	58
2142	10.144.0.1	192.168.12.66	53	34238	74

アプリケーション層のメッセージの内容を以下に示す。

2140 : exp1.rnea.comm.eng.osaka-u.ac.jp の Web ページを保持する IP アドレスの問い合わせ

2142 : exp1.rnea.comm.eng.osaka-u.ac.jp の Web ページを保持する IP アドレスを送信

以上より、この 2 つのパケットの通信によって、クライアントは入力された Web サーバに対応

する IP アドレス(192.168.12.23)を取得することができることがわかる。

3.5.2 TCP コネクションの確立

ログでこの手続きを行っているのは、以下の部分である。

```
  2144 18.972480    192.168.12.66          192.168.12.23          TCP      74
42522 → 5000 [SYN] Seq=4044280001 Win=64240 Len=0 MSS=1460
SACK_PERM=1 TSval=3712487604 TSecr=0 WS=128
  2145 18.974209    192.168.12.23          192.168.12.66          TCP      74
5000 → 42522 [SYN, ACK] Seq=1438445294 Ack=4044280002 Win=65160 Len=0
MSS=1460 SACK_PERM=1 TSval=2689562944 TSecr=3712487604 WS=128
  2146 18.974294    192.168.12.66          192.168.12.23          TCP      66
42522 → 5000 [ACK] Seq=4044280002 Ack=1438445295 Win=64256 Len=0
TSval=3712487606 TSecr=2689562944
```

ログから読み取ることができる情報を表 23、表 24 にまとめる。

表 23　TCP コネクションを確立する手続きの分析 1

No.	送信元 IP アドレス	受信先 IP アドレス	送信元ポート番号	受信先ポート番号	TCP セグメント長
2144	192.168.12.66	192.168.12.23	42522	5000	0
2145	192.168.12.23	192.168.12.66	5000	42522	0
2146	192.168.12.66	192.168.12.23	42522	5000	0

表 24　TCP コネクションを確立する手続きの分析 2

No.	ON になっている TCP フラグ	シーケンス番号	ACK 番号
2144	SYN	4044280001	0
2145	SYN, ACK	1438445294	4044280002
2146	ACK	4044280002	1438445295

実験課題 2、実験課題 3、実験課題 4 と同様の手順で TCP コネクションが確立されているとわかる。この 3 つのパケットにはアプリケーション層のメッセージが存在しない。

3.5.3 HTTP メッセージの送受信

ログでこの手続きを行っているのは、以下の部分である。

2147 18.974476 192.168.12.66 192.168.12.23 IPA 227 unknown 0x53

2148 18.974618 192.168.12.66 192.168.12.23 IPA 1514 unknown 0x4e

2149 18.974640 192.168.12.66 192.168.12.23 IPA 1514 unknown 0x6e

2150 18.974654 192.168.12.66 192.168.12.23 IPA 1514 unknown 0xbc

2151 18.974667 192.168.12.66 192.168.12.23 IPA 1514 unknown 0x23

2152 18.974681 192.168.12.66 192.168.12.23 IPA 1514 unknown 0xb5

2153 18.974734 192.168.12.66 192.168.12.23 RSL 1514 ip.access PDCH DEACTIVATION

2154 18.974761 192.168.12.66 192.168.12.23 IPA 1514 unknown 0x61

2155 18.974775 192.168.12.66 192.168.12.23 IPA 1514 unknown 0x87

2156 18.974788 192.168.12.66 192.168.12.23 IPA 520 OSMO EXT unknown 0xe6

2157 18.975679 192.168.12.23 192.168.12.66 TCP 66 5000 → 42522 [ACK] Seq=1438445295 Ack=4044280163 Win=65024 Len=0 TSval=2689562946 TSecr=3712487606

2158 18.975684 192.168.12.23 192.168.12.66 TCP 66 5000 → 42522 [ACK] Seq=1438445295 Ack=4044285955 Win=61312 Len=0 TSval=2689562946 TSecr=3712487606

2159 18.975688 192.168.12.23 192.168.12.66 TCP 66 5000 → 42522 [ACK] Seq=1438445295 Ack=4044287403 Win=60288 Len=0 TSval=2689562946 TSecr=3712487606

2160 18.975693 192.168.12.23 192.168.12.66 TCP 66 5000 → 42522 [ACK] Seq=1438445295 Ack=4044288851 Win=59392 Len=0 TSval=2689562946 TSecr=3712487606

2161 18.975920 192.168.12.23 192.168.12.66 TCP 66 5000 → 42522 [ACK] Seq=1438445295 Ack=4044290299 Win=58368 Len=0 TSval=2689562946 TSecr=3712487606

2162 18.975924 192.168.12.23 192.168.12.66 TCP 66

```
5000 → 42522 [ACK] Seq=1438445295 Ack=4044291747 Win=57472 Len=0
TSval=2689562946 TSecr=3712487606
 2163 18.975929 192.168.12.23 192.168.12.66 TCP 66
5000 → 42522 [ACK] Seq=1438445295 Ack=4044292201 Win=57088 Len=0
TSval=2689562946 TSecr=3712487606
 2167 19.161365 192.168.12.23 192.168.12.66 IPA 83
unknown 0x54
 2168 19.161501 192.168.12.66 192.168.12.23 TCP 66
42522 → 5000 [ACK] Seq=4044292201 Ack=1438445312 Win=64256 Len=0
TSval=3712487793 TSecr=2689563131
```

次に、それぞれのパケットキャプチャログの詳細を抜粋して示す。

・2147

```
Transmission Control Protocol, Src Port: 42522, Dst Port: 5000, Seq: 4044280002, Ack:
1438445295, Len: 161
                                        (中略)
    Flags: 0x018 (PSH, ACK)
                                        (中略)
IPA protocol ip.access, type: unknown 0x53
    DataLen: 20559
    Protocol: Unknown (0x53)
```

・2148

```
Transmission Control Protocol, Src Port: 42522, Dst Port: 5000, Seq: 4044280163, Ack:
1438445295, Len: 1448
                                        (中略)
    Flags: 0x010 (ACK)
                                        (中略)
IPA protocol ip.access, type: unknown 0x4e
    DataLen: 35152
    Protocol: Unknown (0x4e)
```

・2149

Transmission Control Protocol, Src Port: 42522, Dst Port: 5000, Seq: 4044281611, Ack: 1438445295, Len: 1448

(中略)

Flags: 0x010 (ACK)

(中略)

IPA protocol ip.access, type: unknown 0x6e

DataLen: 56045

Protocol: Unknown (0x6e)

・2150

Transmission Control Protocol, Src Port: 42522, Dst Port: 5000, Seq: 4044283059, Ack: 1438445295, Len: 1448

(中略)

Flags: 0x010 (ACK)

(中略)

IPA protocol ip.access, type: unknown 0xbc

DataLen: 2298

Protocol: Unknown (0xbc)

・2151

Transmission Control Protocol, Src Port: 42522, Dst Port: 5000, Seq: 4044284507, Ack: 1438445295, Len: 1448

(中略)

Flags: 0x010 (ACK)

(中略)

IPA protocol ip.access, type: unknown 0x23

DataLen: 2368

Protocol: Unknown (0x23)

・2152

Transmission Control Protocol, Src Port: 42522, Dst Port: 5000, Seq: 4044285955, Ack: 1438445295, Len: 1448

(中略)

Flags: 0x018 (PSH, ACK)

(中略)

IPA protocol ip.access, type: unknown 0xb5

DataLen: 55784

Protocol: Unknown (0xb5)

・2153

Transmission Control Protocol, Src Port: 42522, Dst Port: 5000, Seq: 4044287403, Ack: 1438445295, Len: 1448

(中略)

Flags: 0x010 (ACK)

(中略)

IPA protocol ip.access, type: unknown 0x14

DataLen: 3900

Protocol: Unknown (0x14)

・2154

Transmission Control Protocol, Src Port: 42522, Dst Port: 5000, Seq: 4044288851, Ack: 1438445295, Len: 1448

(中略)

Flags: 0x010 (ACK)

(中略)

IPA protocol ip.access, type: unknown 0x61

DataLen: 32247

Protocol: Unknown (0x61)

・2155

Transmission Control Protocol, Src Port: 42522, Dst Port: 5000, Seq: 4044290299, Ack: 1438445295, Len: 1448

(中略)

Flags: 0x010 (ACK)

(中略)

IPA protocol ip.access, type: unknown 0x87

DataLen: 40278

Protocol: Unknown (0x87)

・2156

Transmission Control Protocol, Src Port: 42522, Dst Port: 5000, Seq: 4044291747, Ack: 1438445295, Len: 454

(中略)

Flags: 0x018 (PSH, ACK)

(中略)

IPA protocol ip.access, type: OSMO EXT

DataLen: 40686

Protocol: OSMO EXT (0xee)

Osmo ext protocol: Unknown (0xe6)

・2167

Transmission Control Protocol, Src Port: 5000, Dst Port: 42522, Seq: 1438445295, Ack: 4044292201, Len: 17

(中略)

Flags: 0x018 (PSH, ACK)

(中略)

IPA protocol ip.access, type: unknown 0x54

DataLen: 18516

Protocol: Unknown (0x54)

このログから読み取ることができる各パケットの情報を表 25 と表 26 にまとめる。

表 25　HTTP メッセージの送受信の手続きの分析 1

No.	送信元 IP アドレス	受信先 IP アドレス	送信元 ポート番号	受信先 ポート番号	TCP セグメント長
2147	192.168.12.66	192.168.12.23	42522	5000	161
2148	192.168.12.66	192.168.12.23	42522	5000	1448
2149	192.168.12.66	192.168.12.23	42522	5000	1448
2150	192.168.12.66	192.168.12.23	42522	5000	1448
2151	192.168.12.66	192.168.12.23	42522	5000	1448
2152	192.168.12.66	192.168.12.23	42522	5000	1448
2153	192.168.12.66	192.168.12.23	42522	5000	1448
2154	192.168.12.66	192.168.12.23	42522	5000	1448
2155	192.168.12.66	192.168.12.23	42522	5000	1448
2156	192.168.12.66	192.168.12.23	42522	5000	454
2157	192.168.12.23	192.168.12.66	5000	42522	0
2158	192.168.12.23	192.168.12.66	5000	42522	0
2159	192.168.12.23	192.168.12.66	5000	42522	0
2160	192.168.12.23	192.168.12.66	5000	42522	0
2161	192.168.12.23	192.168.12.66	5000	42522	0
2162	192.168.12.23	192.168.12.66	5000	42522	0
2163	192.168.12.23	192.168.12.66	5000	42522	0
2167	192.168.12.23	192.168.12.66	5000	42522	17
2168	192.168.12.66	192.168.12.23	42522	5000	0

表 26　HTTP メッセージの送受信の手続きの分析 2

No.	ON になっている TCP フラグ	シーケンス番号	ACK 番号
2147	PSH, ACK	4044280002	1438445295
2148	ACK	4044280163	1438445295
2149	ACK	4044281611	1438445295
2150	ACK	4044283059	1438445295
2151	ACK	4044284507	1438445295

2152	PSH, ACK	4044285955	1438445295
2153	ACK	4044287403	1438445295
2154	ACK	4044288851	1438445295
2155	ACK	4044290299	1438445295
2156	PSH, ACK	4044291747	1438445295
2157	ACK	1438445295	4044280163
2158	ACK	1438445295	4044285955
2159	ACK	1438445295	4044287403
2160	ACK	1438445295	4044288851
2161	ACK	1438445295	4044290299
2162	ACK	1438445295	4044291747
2163	ACK	1438445295	4044292201
2167	PSH, ACK	1438445295	4044292201
2168	ACK	4044292201	1438445312

No.2147～No.2156 のパケットではクライアントが連続して Web サーバへデータを送信しており、これはネットワークの輻輳により ACK が返ってこないからだと考えることができる。通常、クライアントは送信したデータに対する ACK をまってからつぎの送信を行うが、ACK が帰ってこない場合は再送を行う仕組みとなっているため、クライアント側で再送が繰り返されたと考えられる。No.2157～No.2163 のパケットでは、Web サーバがクライアントに、ACK を連続で送信しており、これらの ACK 番号と、No.2147～No.2156 のシーケンス番号を照らし合わせると、一部のパケットに対しては ACK 応答が対応しているが、No.2148～No.2150 に対応する ACK が存在しないことから、これらのパケットはネットワークの輻輳などによってサーバに届かなかった可能性が高い。No.2167 では web サーバが画像の識別結果をクライアントに返しており、No.2168 では、No.2167 のパケットに対する ACK が送信されている。これらから、通信が正常に再開され、アプリケーション層の処理が進行したことがわかる。

3.5.4 TCP コネクションの終了

ログでこの手続きを行っているのは、以下の部分である。

```
 2169  19.161369      192.168.12.23              192.168.12.66              IPA          434
unknown 0x6e
```

```
 2171 19.163987     192.168.12.66          192.168.12.23          TCP      66
42522 → 5000 [FIN, ACK] Seq=4044292201 Ack=1438445681 Win=64128 Len=0
TSval=3712487796 TSecr=2689563131
 2172 19.165384     192.168.12.23          192.168.12.66          TCP      66
5000 → 42522 [ACK] Seq=1438445681 Ack=4044292202 Win=64128 Len=0
TSval=2689563135 TSecr=3712487796
```

次に、No.2169 のパケットキャプチャログの詳細を抜粋して示す。

```
Transmission Control Protocol, Src Port: 5000, Dst Port: 42522, Seq: 1438445312, Ack:
4044292201, Len: 368
                                    (中略)
    Flags: 0x019 (FIN, PSH, ACK)
```

このログから読み取ることができる各パケットの情報を表 27 と表 28 にまとめる。

表 27　TCP コネクションを終了する手続きの分析 1

No.	送信元 IP アドレス	受信先 IP アドレス	送信元 ポート番号	受信先 ポート番号	TCP セグメント長
2169	192.168.12.23	192.168.12.66	5000	42522	368
2171	192.168.12.66	192.168.12.23	42522	5000	0
2172	192.168.12.23	192.168.12.66	5000	42522	0

表 28　TCP コネクションを終了する手続きの分析 2

No.	ON になっている TCP フラグ	シーケンス番号	ACK 番号
2169	ACK, PSH, FIN	1438445312	4044292201
2171	ACK, FIN	4044292201	1438445681
2172	ACK	1438445681	4044292202

No.2169 のパケットでは、クライアントからの画像の識別結果が送信されると同時に、TCP コネクション終了要求である FIN が含まれている。これに対して、No. 2171 のパケットでは web サーバが ACK を返し、No.2172 のパケットでは web サーバからの FIN に対する ACK が送信されており、正常なコネクションの終了処理が行われたことが確認できる。また、この 3 つのパケットにはいずれもアプリケーション層のメッセージは存在しない。

3.6　実験課題 7

送出間隔 T を変化させながら、以下のコマンドを実行した。

```
$ python3 client.py –num_images 100 –interval T –show_statistics
```

(a)送出間隔を変化させた時の画像のインデックスと遅延時間の関係グラフ

送出間隔 T= 0.05, 0.1, 0.2 ,0.5 ,1.0[s]のときのグラフを重ねてプロットしたら、図 5 のようになった。

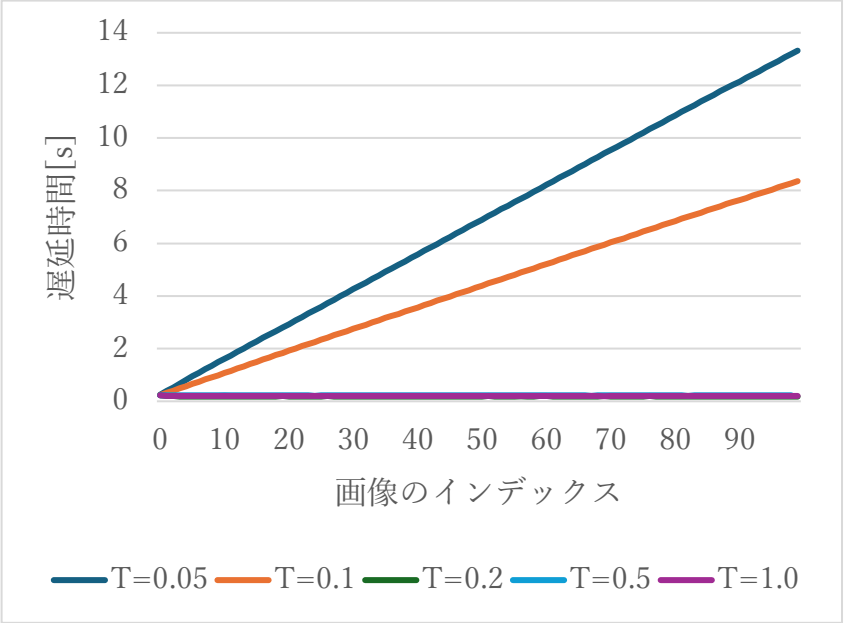

図 5　画像のインデックスと遅延時間の関係グラフ

送出間隔 T の値に関わらず比例関係を示し、T が大きくなるほど傾きが小さくなった。特に T=0.2 ,0.5 ,1.0 のグラフの傾きはほぼ 0 となり、T が小さいと負荷が増すことで遅延時間が大きくなると考えられる。

(b)画像送出頻度と平均遅延時間の関係グラフ

送出間隔 T を 0.05-0.2(0.01 刻み)、0.3-1.0(0.1 刻み)について、横軸を画像送出頻度、縦軸を平均遅延時間としてプロットした。

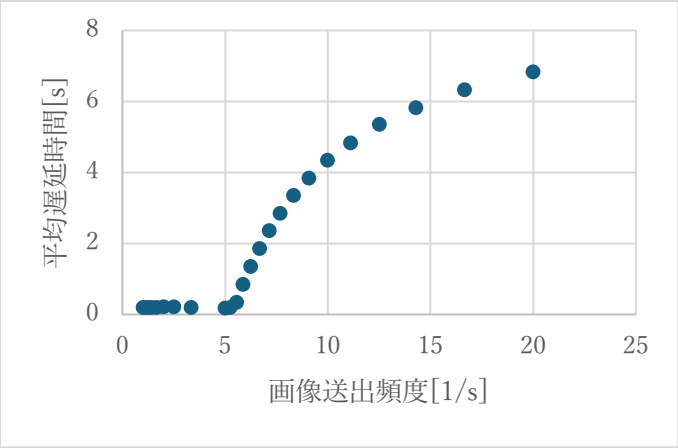

図 6　画像送出頻度と平均遅延時間の関係グラフ

画像送出頻度が 5 よりも大きい場合は常に平均遅延時間が増加していく。これは(a)のグラフによって得られた結果と一致している。画像送出頻度が 5 以下、すなわち T が 0.2 以上の時に平均遅延時間がほぼ 0 となっていることがわかる。

(c)サーバの最大処理能力の推定

3.1 の理論でまとめたように、サーバの処理能力が到着ジョブの負荷を下回ると、滞留ジョブが増加し、平均遅延時間が増大する。図 6 より、画像送出頻度が 5 枚/秒を超えると遅延が顕著に増えるため、サーバの最大処理能力はおよそ 5[枚/秒]であると推定できる。

(d)平均遅延時間と送出間隔 T の和が最小となるような画像送出頻度の特定

横軸を画像送出頻度、縦軸を平均時間と送出間隔の和としてプロットしたら図 7 のようになった。

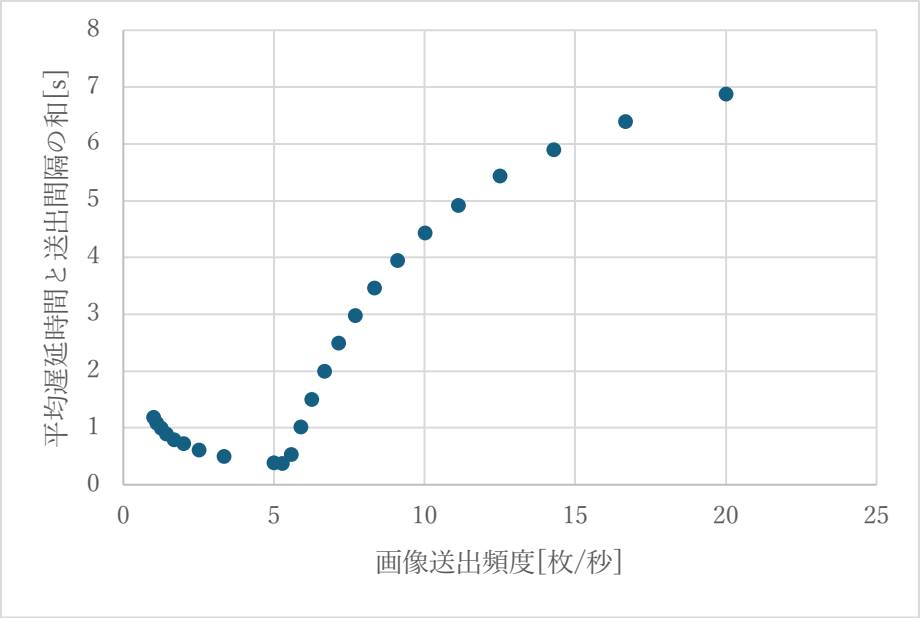

図 7 画像送出頻度、平均遅延時間と送出間隔の和の関係グラフ

このグラフより、平均遅延時間と送出間隔の和が最小となるのは T=0.19[s]の時である。この時の画像送出頻度は 5.263[/s]である。

4.感想

情報通信基礎 2 の授業や個人での web アプリケーション開発において post man を利用するなど、TCP の 3 -ウェイハンドシェイクの流れや HTTP メソッドのリスエスト/レスポンスについて、概念として抽象的には抑えることができていたが、今回の実験のように実際にパケットの流れを可視化してログを読むということは初めてだった。これまで、HTTP メソッドは身近なもので、TCP については学問として学んだものといったイメージが強く、結びつけることができていなかった。今回の実験を通して、普段なんとなく利用している DNS や通信の仕組みについて、UDP, TCP, HTTP の基本動作、通信トラヒック工学の基礎について学びを深めることができた。

5.参考文献

・電子情報工学専門実験 C1 通信ネットワーク工学の基礎
・Wireshark で通信プロトコルを見る
https://future-architect.github.io/articles/20210823b/
・GPU 推論サーバの待ち行列モデル　滝根研究室
http://www2b.comm.eng.osaka-u.ac.jp/~yoshiaki/slides/pdf/2019_q-symp.pdf
・5 章　待ち行列ネットワークモデル　電子情報通信学会知識ベース
https://www.ieice-hbkb.org/files/05/05gun_01hen_05.pdf