# ネットワークサービスの設定

情報学部コンピュータ科学科 3 年 野原健汰 (学籍番号:101730279)
メールアドレス:nohara.kenta@e.mbox.nagoya-u.ac.jp

2019 年 5 月 16 日、5 月 23 日実施

# 目次

# 1 概要

aaaaaaaaaaaaaaaaaaaaaaaaaaaaaa

# 2 はじめに

aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa

## 2.1 実験の目的

aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa

## 2.2 実験機器

### 2.2.1 ルータ

aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa

### 2.2.2 WWW サーバ

aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa

### 2.2.3 PC

aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa

### 2.2.4 モニタ

aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa

### 2.2.5 PC 切替器

aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa

### 2.2.6 USB キーボード/USB マウス

aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa

#### 2.2.7　Ethernet クロスケーブル 2 本

aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa

#### 2.2.8　Ethernet ストレートケーブル 1 本

aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa

# 3　[課題 1] 初期環境の設定

## 3.1　目的と概要

本実験では、fdisk、df、lvdisplay 等のコマンドを実行し、その出力結果を確認することで各コマンドの意味、出力結果の情報を理解する。

## 3.2　実験方法

#### 3.2.1　環境の設定

ルータ (以後 machine1)、WWW サーバ (以後　machine 2)、PC(以後 machine 3) を PC 切替器を介してキーボード及びマウスと接続した。キーボードは PC 切替器のキーボード用端子に、マウスは PC 切替器のマウス用接続端子に接続した。PC 切替器と machine 1 及び machine 2、machine 3 は RGB ケーブルと USB ケーブルを用いて接続した。PC 切替器とモニタを RGB ケーブルを用いて接続した。Ethernet クロスケーブルを用いて machine 1 と machine 2、machine 1 と machine 3 の LAN インターフェース間を直接接続した。machine 1 と上位ネットワークは HUB を介して接続した。

以下の実験は machine 1 及び machine 2 でそれぞれ行った。

#### 3.2.2　Linux カーネルリリース番号の確認

以下のコマンドを実行し、現在動作している Linux カーネルを確認した。

```
uname -r
```

#### 3.2.3　ファイルシステムの確認

以下のコマンドを実行し、ファイルシステムのマウントポイントを記述した設定ファイル/etc/fstab の内容を確認した。

```
cat /etc/fstab
```

以下のコマンドを実行し、ディスクパーティションとマウントされているファイルシステムを確認した。

```
fdisk -l
df
```

以下のコマンドを実行し、論理ボリュームの内容を確認した。

```
lvdisplay
```

### 3.2.4 ホスト名の設定

- machine 1
  以下のコマンドを実行し、ホスト名を設定した。

  ```
  hostname icesc16.ice.nuie.nagoya-u.ac.jp
  ```

  vi エディタで設定ファイル/etc/hostname の内容を以下のように変更し、ホスト名の恒久的変更を設定した。

  ```
  #vi /etc/hostname
  127.0.0.1 icesc16.ice.nuie.nagoya-u.ac.jp localhost.localdomain localhost
  ```

- machine 2 以下のコマンドを実行し、ホスト名を設定した。

  ```
  hostname www6.ice.nuie.nagoya-u.ac.jp
  ```

  vi エディタで設定ファイル/etc/hostname の内容を以下のように変更し、ホスト名の恒久的変更を設定した。

  ```
  #vi /etc/hostname
  127.0.0.1 www6.ice.nuie.nagoya-u.ac.jp localhost.localdomain localhost
  ```

### 3.2.5 SELinux の状態確認と無効化

- 現在の設定状況の確認
  以下のコマンドを実行し、現在の設定状況を確認した。

  ```
  cat /etc/sysconfig/selinux
  getenforce
  ```

- SELinux の設定を permissive モードに変更
  以下のコマンドを実行し、SELinux の設定を permissive モードに変更した。

  ```
  setenforce 0
  ```

  変更したら、SELinux の設定ファイル /etc/sysconfig/selinux の内容を vi エディタで以下のように変

更し、SELinux の設定を恒久的にした。

SELINUX=permissive

## 3.3 実験結果

### 3.3.1 Linux カーネルリリース番号の確認の結果

- machine1

```
[root@localhost ~]# uname -r
3.10.0-862.el7.x86_64
```

- machine2

```
[root@localhost ~]# uname -r
3.10.0-862.el7.x86_64
```

### 3.3.2 ファイルシステムのマウントポイントを記述した設定ファイル/etc/fstab の内容を確認した結果

- machine1

```
[root@localhost ~]# cat /etc/fstab

#
#/etc/fstab
#Created by anaconda on Sat Apr 6 00:42:18 2019
#
#Accessible filesystems, by reference, are maintained under '/dev/disk'
#See man pages fstab(5), findfs(8), mount(8) and or blkid(8) for more info
#
/dev/mapper/centos-root / xfs defaults 0 0
UUID=5eb200c5-354e-419c-9306-bdcc507c72c5 /boot xfs defaults 0 0
/dev/mapper/centos-home /home xfs defaults 0 0
/dev/mapper/centos-swap swap swap defaults 0 0
```

- machine2

```
[root@localhost ~]# cat /etc/fstab

#
# /etc/fstab
# Created by anaconda on Sat Apr  6 00:42:18 2019
#
# Accessible filesystems, by reference, are maintained under '/dev/disk'
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info
#
/dev/mapper/centos-root / xfs  defaults   0 0
UUID=5eb200c5-354e-419c-9306-bdcc507c72c5 /boot  xfs  defaults  0 0
/dev/mapper/centos-home /home xfs    defaults  0 0
/dev/mapper/centos-swap swap  swap  defaults    0 0
```

1 列目はデバイス名、2 列目はマウントポイント、3 列目はファイルシステムの種類、4 列目はマウント時のオプション、5 列目はファイルシステムを dump するかしないかをそれぞれ表している。2 行目は linux ディスクの UUID を表している。UUID とは一意な識別子である。linux では windows とは異なり、ストレージデバイスを接続しただけではファイルの読み書きができない。また、linux は windows と異なり、パーティション単位でファイルのツリー構造を持つことができず、複数のパーティションも 1 つのツリー構造にまとめて扱う。デバイスにあるパーティションをこのツリー構造のどこかにディレクトリとして登録する作業がマウントであり、登録するディレクトリがマウントポイントである。(文献 [2] 参照)

### 3.3.3 ディスクパーティション確認の結果

- machine1

```
[root@localhost ~]# fdisk -l

Disk /dev/sda: 250.1 GB, 250059350016 bytes, 488397168 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk label type: dos
Disk identifier: 0x00016021

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1   *        2048     2099199     1048576   83  Linux
/dev/sda2         2099200   488396799   243148800   8e  Linux LVM

Disk /dev/mapper/centos-root: 53.7 GB, 53687091200 bytes, 104857600 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes


Disk /dev/mapper/centos-swap: 8321 MB, 8321499136 bytes, 16252928 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes


Disk /dev/mapper/centos-home: 187.0 GB, 186969489408 bytes, 365174784 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
```

fstab で確認したマウントされているデバイスのパーティション情報を表している。/dev/sda には 2 つのパーティションが作成されていることがわかる。

- machine2

script がとれていなかったため省略するが、machine1 と同じ結果であった。

### 3.3.4 マウントされているファイルシステム確認の結果

- machine1

```
[root@localhost ~]# df
Filesystem              1K-blocks    Used Available Use% Mounted on
```

```
/dev/mapper/centos-root  52403200 1081936  51321264   3% /
devtmpfs                  3936440       0   3936440   0% /dev
tmpfs                     3949112       0   3949112   0% /dev/shm
tmpfs                     3949112    8848   3940264   1% /run
tmpfs                     3949112       0   3949112   0% /sys/fs/cgroup
/dev/sda1                 1038336  145900    892436  15% /boot
/dev/mapper/centos-home 182498240   32944 182465296   1% /home
tmpfs                      789824       0    789824   0% /run/user/0
```

- machine2

```
[root@localhost ~]# df
Filesystem              1K-blocks    Used Available Use% Mounted on
/dev/mapper/centos-root  52403200 1007316  51395884   2% /
devtmpfs                  3913236       0   3913236   0% /dev
tmpfs                     3925932       0   3925932   0% /dev/shm
tmpfs                     3925932    9044   3916888   1% /run
tmpfs                     3925932       0   3925932   0% /sys/fs/cgroup
/dev/sda1                 1038336  146052    892284  15% /boot
/dev/mapper/centos-home 182498240   32944 182465296   1% /home
tmpfs                      785188       0    785188   0% /run/user/0
```

1 列目はデバイス名、2 列目は全ディスク容量、3 列目は使用容量、4 列目は空き容量、5 列目は使用率、6 列目はマウントポイントを表している。

### 3.3.5 論理ボリュームの内容確認の結果

- machine1

```
[root@localhost ~]# lvdisplay
  --- Logical volume ---
  LV Path                /dev/centos/swap
  LV Name                swap
  VG Name                centos
  LV UUID                5S8Xtv-OZ7e-3krn-HqEs-8xHA-CYFl-BfapS4
  LV Write Access        read/write
  LV Creation host, time localhost, 2019-03-22 12:04:24 +0900
  LV Status              available
  # open                 2
  LV Size                7.75 GiB
  Current LE             1984
  Segments               1
  Allocation             inherit
  Read ahead sectors     auto
  - currently set to     256
  Block device           253:1

  --- Logical volume ---
  LV Path                /dev/centos/home
  LV Name                home
  VG Name                centos
  LV UUID                vanhId-NRrd-icZs-ugHk-PTLw-F5zJ-XgnuQb
  LV Write Access        read/write
  LV Creation host, time localhost, 2019-03-22 12:04:25 +0900
  LV Status              available
  # open                 1
```

```
  LV Size                <174.13 GiB
  Current LE             44577
  Segments               1
  Allocation             inherit
  Read ahead sectors     auto
  - currently set to     256
  Block device           253:2

  --- Logical volume ---
  LV Path                /dev/centos/root
  LV Name                root
  VG Name                centos
  LV UUID                zNAdo6-xqs9-t9PR-iKrf-C855-jD6d-Lv0l7q
  LV Write Access        read/write
  LV Creation host, time localhost, 2019-03-22 12:04:26 +0900
  LV Status              available
  # open                 1
  LV Size                50.00 GiB
  Current LE             12800
  Segments               1
  Allocation             inherit
  Read ahead sectors     auto
  - currently set to     256
  Block device           253:0
```

- machine2

```
[root@localhost ~]# lvdisplay
  --- Logical volume ---
  LV Path                /dev/centos/swap
  LV Name                swap
  VG Name                centos
  LV UUID                PohTZH-fMlu-Lf0h-Mdhh-Z2xs-I2QH-ZM8ush
  LV Write Access        read/write
  LV Creation host, time localhost, 2019-04-06 00:42:14 +0900
  LV Status              available
  # open                 2
  LV Size                7.75 GiB
  Current LE             1984
  Segments               1
  Allocation             inherit
  Read ahead sectors     auto
  - currently set to     256
  Block device           253:1

  --- Logical volume ---
  LV Path                /dev/centos/home
  LV Name                home
  VG Name                centos
  LV UUID                XYwPiS-jXOD-IKEh-Jzzo-NDvs-qfED-uV8ABr
  LV Write Access        read/write
  LV Creation host, time localhost, 2019-04-06 00:42:14 +0900
  LV Status              available
  # open                 1
  LV Size                <174.13 GiB
  Current LE             44577
  Segments               1
  Allocation             inherit
```

```
  Read ahead sectors     auto
  - currently set to     256
  Block device           253:2

  --- Logical volume ---
  LV Path                /dev/centos/root
  LV Name                root
  VG Name                centos
  LV UUID                Oz554A-DomL-OGcW-B3u7-yAir-UU6R-Fa3bE1
  LV Write Access        read/write
  LV Creation host, time localhost, 2019-04-06 00:42:16 +0900
  LV Status              available
  # open                 1
  LV Size                50.00 GiB
  Current LE             12800
  Segments               1
  Allocation             inherit
  Read ahead sectors     auto
  - currently set to     256
  Block device           253:0
```

swap、home、root という名前の 3 つの論理ボリュームが作成されている。論理ボリュームはボリュームグループ上に作成された仮想的なパーティションである。論理ボリュームを作成することで、物理ディスク及びそのパーティションをそのサイズに関わらず、単一のストレージ・ソースとして抽象化して把握できるようになる。(文献 [3] 参照)

### 3.3.6 現在の設定状況の確認の結果

- machine1

```
[root@icesc16 ~]# cat /etc/sysconfig/selinux
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#     enforcing - SELinux security policy is enforced.
#     permissive - SELinux prints warnings instead of enforcing.
#     disabled - No SELinux policy is loaded.
SELINUX=enforcing
# SELINUXTYPE= can take one of three two values:
#     targeted - Targeted processes are protected,
#     minimum - Modification of targeted policy. Only selected processes are protected.
#     mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

```
[root@icesc16 ~]# getenforce
Enforcing
```

- machine2

```
[root@www6 ~]# cat /etc/sysconfig/selinux

# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#     enforcing - SELinux security policy is enforced.
#     permissive - SELinux prints warnings instead of enforcing.
#     disabled - No SELinux policy is loaded.
```

```
SELINUX=enforcing
# SELINUXTYPE= can take one of three two values:
#     targeted - Targeted processes are protected,
#     minimum - Modification of targeted policy. Only selected processes are protected.
#     mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

```
[root@www6 ~]# getenforce
Enforcing
```

セキュリティ関連の Linux カーネル制御機能である SELinux が有効 (Enforcing) になっていることが分かる。SELinux はデフォルトで有効になっている。

### 3.3.7 SELinux の設定を permissive モードに変更した結果

- machine1

```
[root@icesc16 ~]# cat /etc/sysconfig/selinux
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#     enforcing - SELinux security policy is enforced.
#     permissive - SELinux prints warnings instead of enforcing.
#     disabled - No SELinux policy is loaded.
SELINUX=permissive
# SELINUXTYPE= can take one of three two values:
#     targeted - Targeted processes are protected,
#     minimum - Modification of targeted policy. Only selected processes are protected.
#     mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

- machine2

```
[root@www6 ~]# cat /etc/sysconfig/selinux
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#     enforcing - SELinux security policy is enforced.
#     permissive - SELinux prints warnings instead of enforcing.
#     disabled - No SELinux policy is loaded.
SELINUX=permissive
# SELINUXTYPE= can take one of three two values:
#     targeted - Targeted processes are protected,
#     minimum - Modification of targeted policy. Only selected processes are protected.
#     mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

SELinux を permissive に変更して機能を無効化した。これによりアクセス制御は無効になる (警告メッセージは出る)。

## 3.4 考察

fstab に書かれているファイルシステムと、df で見たファイルシステムの数が違う (?)。

# 4 [課題 2] ネットワーク設定

## 4.1 目的と概要

aaaaaaaaaaaaaaaaaaaaa

## 4.2 実験方法

以下は machine 1 及び machine 2 でそれぞれ行った。

### 4.2.1 現在の状況の確認

- ネットワークデバイスの確認以下のコマンドを実行し、ネットワークデバイスの確認を行った。

```
nmcli device status
nmcli device show
```

- ネットワーク接続の確認以下のコマンドを実行し、ネットワーク接続の確認を行った。

```
nmcli connection show –active
```

### 4.2.2 接続設定

- machine 1

  以下のコマンドを実行し、イーサネットデバイスを追加した。

```
nmcli c add type ethernet ifname enp1s0 con-name enp1s0
nmcli c add type ethernet ifname enp2s0 con-name enp2s0
nmcli c add type ethernet ifname enp3s0 con-name enp3s0
```

  以下のコマンドを実行し、OS 起動時の自動接続指定を行った。

```
nmcli c m enp1s0 connection.autoconnect yes
nmcli c m enp2s0 connection.autoconnect yes
nmcli c m enp3s0 connection.autoconnect yes
```

  以下のコマンドをそれぞれ実行し、IP アドレスの設定を行った。

```
nmcli c m enp1s0 ipv4.method manual ipv4.addresses “192.168.100.16/24”
nmcli c m enp2s0 ipv4.method manual ipv4.addresses “192.168.150.1/24”
nmcli c m enp3s0 ipv4.method manual ipv4.addresses “192.168.200.1/24”
```

  以下のコマンドを実行し、デフォルトゲートウェイの設定を行った (LAN1 のみ)。

```
nmcli c m enp1s0 gateway “192.168.100.1”
```

以下のコマンドを実行し、DNS サーバーの設定を行った。

```
nmcli c m enp1s0 ipv4.dns “10.10.1.2”
nmcli c m enp2s0 ipv4.dns “10.10.1.2”
nmcli c m enp3s0 ipv4.dns “10.10.1.2”
```

- machine 2 以下のコマンドを実行し、OS 起動時の自動接続指定を行った。

```
nmcli c m enp3s0 connection.autoconnect yes
```

#### 4.2.3 ネットワーク接続の有効化

以下のコマンドをそれぞれ実行し、ネットワーク接続を有効化した。

- machine 1

```
nmcli con down enp1s0
nmcli con up enp1s0
nmcli con down enp2s0
nmcli con up enp2s0
nmcli con down enp3s0
nmcli con up enp3s0
```

- machine 2

```
nmcli con down enp3s0
nmcli con up enp3s0
```

#### 4.2.4 各種情報の確認

以下のコマンドをそれぞれ実行し、各デバイスの設定の確認、ルーティング情報の確認を行った。

```
ip addr show
ip route
```

### 4.3 実験結果

#### 4.3.1 ネットワークデバイスの確認の結果

- machine 1

```
[root@icesc16 ~]# nmcli device status
```

```
DEVICE  TYPE      STATE         CONNECTION
enp1s0  ethernet  disconnected  --
enp2s0  ethernet  disconnected  --
enp3s0  ethernet  disconnected  --
enp4s0  ethernet  unavailable   --
lo      loopback  unmanaged     --
```

1 列目はデバイス名、2 列目はコネクションタイプ、3 列目は接続状態、4 列目は接続を表している。machine1 では enp1s0、enp2s0、enp3s0 の 3 種類のネットワークデバイスを使用できる (この段階では接続されていない) ことがわかる。lo はローカルループバックを表している。

```
[root@icesc16 ~]# nmcli device show

GENERAL.DEVICE:                         enp1s0
GENERAL.TYPE:                           ethernet
GENERAL.HWADDR:                         00:E0:67:12:2D:B4
GENERAL.MTU:                            1500
GENERAL.STATE:                          30 (disconnected)
GENERAL.CONNECTION:                     --
GENERAL.CON-PATH:                       --
WIRED-PROPERTIES.CARRIER:               on

GENERAL.DEVICE:                         enp2s0
GENERAL.TYPE:                           ethernet
GENERAL.HWADDR:                         00:E0:67:12:2D:B5
GENERAL.MTU:                            1500
GENERAL.STATE:                          30 (disconnected)
GENERAL.CONNECTION:                     --
GENERAL.CON-PATH:                       --
WIRED-PROPERTIES.CARRIER:               on

GENERAL.DEVICE:                         enp3s0
GENERAL.TYPE:                           ethernet
GENERAL.HWADDR:                         00:E0:67:12:2D:B6
GENERAL.MTU:                            1500
GENERAL.STATE:                          30 (disconnected)
GENERAL.CONNECTION:                     --
GENERAL.CON-PATH:                       --
WIRED-PROPERTIES.CARRIER:               on

GENERAL.DEVICE:                         enp4s0
GENERAL.TYPE:                           ethernet
GENERAL.HWADDR:                         00:E0:67:12:2D:B7
GENERAL.MTU:                            1500
GENERAL.STATE:                          20 (unavailable)
GENERAL.CONNECTION:                     --
GENERAL.CON-PATH:                       --
WIRED-PROPERTIES.CARRIER:               off

GENERAL.DEVICE:                         lo
GENERAL.TYPE:                           loopback
GENERAL.HWADDR:                         00:00:00:00:00:00
GENERAL.MTU:                            65536
GENERAL.STATE:                          10 (unmanaged)
GENERAL.CONNECTION:                     --
GENERAL.CON-PATH:                       --
```

```
IP4.ADDRESS[1]:                         127.0.0.1/8
IP4.GATEWAY:                            --
IP6.ADDRESS[1]:                         ::1/128
IP6.GATEWAY:                            --
```

nmcli device status コマンドで確認したネットワークデバイスの詳細が表示されている。GENERAL.DEVICE はデバイス名、GENERAL.TYPE はコネクションタイプ、HWADDR は MAC アドレス、GENERAL.MTU は一度に送信できる最大のデータ量 (単位は byte)、GENERAL.STATE は接続状態をそれぞれ表している。ローカルループバックアドレスは自分自身を表す IP アドレスであり一般的に 127.0.0.1 が利用される。

- machine 2

```
[root@www6 ~]# nmcli device status

DEVICE  TYPE      STATE         CONNECTION
enp3s0  ethernet  disconnected  --
lo      loopback  unmanaged     --
wlp2s0  wifi      unmanaged     --
```

コマンドの見方は machine1 で示したため省略する。

machine2 では有線 LAN である enp3s0 のみ使用できる (この段階では接続されていない) ことがわかる。wlp2s0 は無線 LAN を表している。

```
[root@www6 ~]# nmcli device show

GENERAL.DEVICE:                         enp3s0
GENERAL.TYPE:                           ethernet
GENERAL.HWADDR:                         94:C6:91:A8:C9:54
GENERAL.MTU:                            1500
GENERAL.STATE:                          30 (disconnected)
GENERAL.CONNECTION:                     --
GENERAL.CON-PATH:                       --
WIRED-PROPERTIES.CARRIER:               on

GENERAL.DEVICE:                         lo
GENERAL.TYPE:                           loopback
GENERAL.HWADDR:                         00:00:00:00:00:00
GENERAL.MTU:                            65536
GENERAL.STATE:                          10 (unmanaged)
GENERAL.CONNECTION:                     --
GENERAL.CON-PATH:                       --
IP4.ADDRESS[1]:                         127.0.0.1/8
IP4.GATEWAY:                            --
IP6.ADDRESS[1]:                         ::1/128
IP6.GATEWAY:                            --

GENERAL.DEVICE:                         wlp2s0
GENERAL.TYPE:                           wifi
GENERAL.HWADDR:                         DC:8B:28:55:19:23
GENERAL.MTU:                            1500
GENERAL.STATE:                          10 (unmanaged)
GENERAL.CONNECTION:                     --
GENERAL.CON-PATH:                       --
IP4.GATEWAY:                            --
IP6.GATEWAY:                            --
```

コマンドの見方は machine1 で示したため省略する。

#### 4.3.2 ネットワーク接続の確認の結果

- machine 1

```
[root@icesc16 ~]# nmcli connection show --active
NAME  UUID  TYPE  DEVICE
```

- machine 2

```
[root@www6 ~]# nmcli connection show -active
NAME  UUID  TYPE  DEVICE
```

この段階では、ネットワーク接続されているデバイスが存在しないため何も表示されなかった。

#### 4.3.3 接続設定の結果

- machine 1

  イーサネットデバイスを追加した結果

```
[root@icesc16 ~]# nmcli c add type ethernet ifname enp1s0 con-name enp1s0
Warning: There is another connection with the name 'enp1s0'.
Reference the connection by its uuid '53ad581c-713c-414c-b4d1-e5a23ba31e89'
Connection 'enp1s0' (53ad581c-713c-414c-b4d1-e5a23ba31e89) successfully added.
[root@icesc16 ~]# nmcli c add type ethernet ifname enp2s0 con-name enp2s0
Warning: There is another connection with the name 'enp2s0'.
Reference the connection by its uuid '819df032-dd6f-4ea0-a0b7-4930ceb0d31f'
Connection 'enp2s0' (819df032-dd6f-4ea0-a0b7-4930ceb0d31f) successfully added.
[root@icesc16 ~]# nmcli c add type ethernet ifname enp3s0 con-name enp3s0
Warning: There is another connection with the name 'enp3s0'.
Reference the connection by its uuid '5b93b11d-e766-41ed-8c95-158f71acdc23'
Connection 'enp3s0' (5b93b11d-e766-41ed-8c95-158f71acdc23) successfully added.
```

  enp1s0、enp2s0、enp3s0 の 3 種類のイーサネットデバイスを追加できたことがわかる。

#### 4.3.4 ネットワーク接続の有効化の結果

- machine 1

```
[root@icesc16 ~]# nmcli con down enp1s0
Connection 'enp1s0' successfully deactivated
(D-Bus active path: /org/freedesktop/NetworkManager/ActiveConnection/11)
[root@icesc16 ~]# nmcli con up enp1s0
Connection successfully activated
(D-Bus active path: /org/freedesktop/NetworkManager/ActiveConnection/14)
[root@icesc16 ~]# nmcli con down enp2s0
Connection 'enp2s0' successfully deactivated
(D-Bus active path: /org/freedesktop/NetworkManager/ActiveConnection/17)
[root@icesc16 ~]# nmcli con up enp2s0
Connection successfully activated
(D-Bus active path: /org/freedesktop/NetworkManager/ActiveConnection/52)
[root@icesc16 ~]# nmcli con down enp3s0
```

```
Connection 'enp3s0' successfully deactivated
(D-Bus active path: /org/freedesktop/NetworkManager/ActiveConnection/20)
[root@icesc16 ~]# nmcli con up enp3s0
Connection successfully activated
(D-Bus active path: /org/freedesktop/NetworkManager/ActiveConnection/56)
```

- machine 2

```
[root@www6 ~]# nmcli con down enp3s0
Connection 'enp3s0' successfully deactivated (D-Bus active path:
/org/freedesktop/NetworkManager/ActiveConnection/2)
[root@www6 ~]# nmcli con up enp3s0
Connection successfully activated (D-Bus active path:
/org/freedesktop/NetworkManager/ActiveConnection/5)
```

一度 down して up することで有効化される。

#### 4.3.5 各種情報の確認

- machine 1

```
[root@icesc16 ~]# ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN
group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp1s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP
group default qlen 1000
    link/ether 00:e0:67:12:2d:b4 brd ff:ff:ff:ff:ff:ff
    inet 192.168.100.16/24 brd 192.168.100.255 scope global noprefixroute enp1s0
       valid_lft forever preferred_lft forever
    inet6 fe80::1a08:bb64:b6fa:6306/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
3: enp2s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP
group default qlen 1000
    link/ether 00:e0:67:12:2d:b5 brd ff:ff:ff:ff:ff:ff
    inet 192.168.150.1/24 brd 192.168.150.255 scope global noprefixroute enp2s0
       valid_lft forever preferred_lft forever
    inet6 fe80::f6cc:73c7:7638:c48c/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
4: enp3s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP
group default qlen 1000
    link/ether 00:e0:67:12:2d:b6 brd ff:ff:ff:ff:ff:ff
    inet 192.168.200.1/24 brd 192.168.200.255 scope global noprefixroute enp3s0
       valid_lft forever preferred_lft forever
    inet6 fe80::894a:645e:4f45:794d/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
5: enp4s0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc pfifo_fast state DOWN
group default qlen 1000
    link/ether 00:e0:67:12:2d:b7 brd ff:ff:ff:ff:ff:ff
```

```
[root@icesc16 ~]# ip route
default via 192.168.100.1 dev enp1s0 proto static metric 100
```

```
192.168.100.0/24 dev enp1s0 proto kernel scope link src 192.168.100.16 metric 100
192.168.150.0/24 dev enp2s0 proto kernel scope link src 192.168.150.1 metric 103
192.168.200.0/24 dev enp3s0 proto kernel scope link src 192.168.200.1 metric 104
```

- machine 2

```
[root@www6 ~]# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN
group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp3s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP
 group default qlen 1000
    link/ether 94:c6:91:a8:c9:54 brd ff:ff:ff:ff:ff:ff
    inet 192.168.150.100/24 brd 192.168.150.255 scope global noprefixroute dynamic
enp3s0
       valid_lft 3435sec preferred_lft 3435sec
    inet6 fe80::15b7:4ca2:a0f3:9ba3/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
3: wlp2s0: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN
group default qlen 1000
    link/ether dc:8b:28:55:19:23 brd ff:ff:ff:ff:ff:ff
```

```
[root@www6 ~]# ip route
default via 192.168.150.1 dev enp3s0 proto dhcp metric 100
192.168.150.0/24 dev enp3s0 proto kernel scope link src 192.168.150.100 metric 100
```

以下に ip addr コマンドの見方を示す。

表 1　ip addr コマンドの意味

| | |
|---|---|
| lo、enp1s0、enp2s0、enp3s0、enp4s0、wlp2s0 | デバイス名<br>(lo はローカルループバック) |
| LOOPBACK、BROADCAST、<br>MULTICAST、UP、LOWER_UP | LOOPBACK... ループバック対応<br>BROADCAST... ブロードキャスト対応<br>MULTICAST... マルチキャスト対応<br>UP... ネットワークインタフェースが有効<br>LOWER_UP... デバイスにケーブルが繋がっている (有線 LAN) |
| mtu | 一度に送信できるデータの最大値 (byte)<br>(Linux 標準値 1500) |
| qdisc | ネットワークへの送信を待つデータの保存規則<br>pfifo_fast... パケットの優先度を考慮した qdisc(Linux デフォルト) |
| state | UP... ネットワークインターフェースが作動中<br>DOWN... ネットワークインターフェースが作動していない |
| group default | グループインターフェース |
| qlen | queue の長さ |
| link/loopback、link/either | MAC アドレス |
| brd ff:ff:ff:ff:ff:ff | ブロードキャストアドレス |
| inet、inet6 | inet...IPv4 アドレス<br>inet6...IPv6 アドレス |
| brd XXX.XXX.XXX.255 | ディレクティッドブロードキャストアドレス |
| scope | 送信先指定<br>global... 他ネットワークへのゲートウェイを<br>経由した unicast 通信による経路<br>link... 自身が属するネットワーク |
| valid_lft、preferred_lft | valid_lft... 有効な IPv4 アドレスの有効期限<br>preferred_lft... 適切な IPv4 アドレスの有効期限 |

以下に ip route コマンドの見方を示す。

表 2　ip route コマンドの意味

| | |
|---|---|
| via | ネクストホップのルーター |
| dev | 対象デバイス名 |
| proto | kernel... カーネルが自動生成した経路<br>dhcp...DHCP が自動生成した経路 |
| metric | ネットワーク間の仮想的な距離 |
| scope | 送信先を指定<br>link... 自身が属するネットワーク |
| src | 送信元を指定 |

### 4.4　考察

aaaaaaaaaaaaaaaaaaaaa

# 5　[課題 3]DHCP サービスの設定

## 5.1　目的と概要

aaaaaaaaaaaaaaaaaaaaa

## 5.2　実験方法

以下の実験は machine 1 で行った。

#### 5.2.1　稼働中のサービスの確認

以下のコマンドを打ち込み、稼働中のサービスの確認を行った。

```
systemctl list-units –type=service
```

#### 5.2.2　firewall の停止

以下のコマンドをそれぞれ打ち込み、firewall を一時的に停止した。

```
systemctl stop firewalld
systemctl status firewalld
```

#### 5.2.3　DHCP サービスのインストールと設定

- DHCP 関連パッケージの確認と DHCP サーバのインストール
  以下のコマンドを打ち込み、DHCP 関連パッケージの確認と DHCP サーバのインストールを行った。

yum list dhcp-
yum info dhcp
yum install dhcp

- 設定ファイルの修正
  設定ファイルを以下のように修正した。ゲートウェイ及びネットマスク、DNS サーバを指定した。
  リース期間はデフォルト 1 時間 (3600 秒)、最大 1 日 (86400 秒)、machine2 のリース期間は 30 日 (2592000 秒) に設定した。

```
#
# DHCP Server Configuration file.
#   see /usr/share/doc/dhcp*/dhcpd.conf.example
#   see dhcpd.conf(5) man page
#

default-lease-time 3600;
max-lease-time 86400;
option domain-name-servers 10.10.1.2;

subnet 192.168.150.0 netmask 255.255.255.0 {
    range 192.168.150.100 192.168.150.250;
    option routers 192.168.150.1;
}

subnet 192.168.200.0 netmask 255.255.255.0 {
    range 192.168.200.100 192.168.200.250;
    option routers 192.168.200.1;
}

host www2 {
    hardware ethernet 94:C6:91:A8:C6:B9;
    fixed-address 192.168.150.2;
    default-lease-time 2592000;
}
```

- DHCP サーバの起動と動作確認

systemctl restart dhcpd
systemctl status dhcpd

- DHCP サーバのブート時自動起動の設定

chkconfig dhcpd

## 5.3 実験結果

### 5.3.1 稼働中のサービスの確認の結果

```
[root@icesc16 ~]# systemctl list-units --type=service
```

```
UNIT                                                LOAD   ACTIVE SUB     DESCRIPTION
auditd.service
loaded active running Security Auditing Service
crond.service
loaded active running Command Scheduler
dbus.service
loaded active running D-Bus System Message Bus
firewalld.service
loaded active running firewalld - dynamic firewall daemon
getty@tty1.service
loaded active running Getty on tty1
irqbalance.service
loaded active running irqbalance daemon
kdump.service
loaded active exited  Crash recovery kernel arming
kmod-static-nodes.service
loaded active exited  Create list of required static device nodes for the curren
lvm2-lvmetad.service
loaded active running LVM2 metadata daemon
lvm2-monitor.service
loaded active exited  Monitoring of LVM2 mirrors, snapshots etc. using dmeventd
lvm2-pvscan@8:2.service
loaded active exited  LVM2 PV scan on device 8:2
network.service
loaded active exited  LSB: Bring up/down networking
NetworkManager-wait-online.service
loaded active exited  Network Manager Wait Online
NetworkManager.service
loaded active running Network Manager
polkit.service
loaded active running Authorization Manager
postfix.service
loaded active running Postfix Mail Transport Agent
rhel-dmesg.service
loaded active exited  Dump dmesg to /var/log/dmesg
rhel-domainname.service
loaded active exited  Read and set NIS domainname from /etc/sysconfig/network
rhel-import-state.service
loaded active exited  Import network configuration from initramfs
rhel-readonly.service
loaded active exited  Configure read-only root support
rsyslog.service
loaded active running System Logging Service
sshd.service
loaded active running OpenSSH server daemon
systemd-backlight@backlight:acpi_video0.service
loaded active exited  Load/Save Screen Backlight Brightness of backlight:acpi_vi
systemd-journal-flush.service
loaded active exited  Flush Journal to Persistent Storage
systemd-journald.service
loaded active running Journal Service
systemd-logind.service
loaded active running Login Service
systemd-random-seed.service
loaded active exited  Load/Save Random Seed
systemd-readahead-collect.service
loaded active exited  Collect Read-Ahead Data
```

```
systemd-readahead-replay.service
loaded active exited  Replay Read-Ahead Data
systemd-remount-fs.service
loaded active exited  Remount Root and Kernel File Systems
systemd-sysctl.service
loaded active exited  Apply Kernel Variables
systemd-tmpfiles-setup-dev.service
loaded active exited  Create Static Device Nodes in /dev
systemd-tmpfiles-setup.service
loaded active exited  Create Volatile Files and Directories
systemd-udev-trigger.service
loaded active exited  udev Coldplug all Devices
systemd-udevd.service
loaded active running udev Kernel Device Manager
systemd-update-utmp.service
loaded active exited  Update UTMP about System Boot/Shutdown
systemd-user-sessions.service
loaded active exited  Permit User Sessions
systemd-vconsole-setup.service
loaded active exited  Setup Virtual Console
tuned.service
loaded active running Dynamic System Tuning Daemon
LOAD   = Reflects whether the unit definition was properly loaded.
ACTIVE = The high-level unit activation state, i.e. generalization of SUB.
SUB    = The low-level unit activation state, values depend on unit type.
loaded units listed.Pass --all to see loaded but inactive units, too.
To show all installed unit files use 'systemctl list-unit-files'.
7mlines 1-47/47 (END)
lines 1-47/47 (END)
```

### 5.3.2 DHCP 関連パッケージの確認結果

```
[root@icesc16 ~]# yum list dhcp-\*
Installed Packages
dhcp-common.x86_64                                      12:4.2.5-68.el7.centos
@anaconda
dhcp-libs.x86_64                                        12:4.2.5-68.el7.centos
@anaconda
Available Packages
dhcp.x86_64                                             12:4.2.5-68.el7.centos.1
base
dhcp-common.x86_64                                      12:4.2.5-68.el7.centos.1
base
dhcp-devel.i686                                         12:4.2.5-68.el7.centos.1
base
dhcp-devel.x86_64                                       12:4.2.5-68.el7.centos.1
base
dhcp-libs.i686                                          12:4.2.5-68.el7.centos.1
base
dhcp-libs.x86_64                                        12:4.2.5-68.el7.centos.1
base
```

```
[root@icesc16 ~]# yum info dhcp
Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
 * base: mirrors.cat.net
```

```
 * extras: mirrors.cat.net
 * updates: ftp.jaist.ac.jp
Available Packages
Name        : dhcp
Arch        : x86_64
Epoch       : 12
Version     : 4.2.5
Release     : 68.el7.centos.1
Size        : 513 k
Repo        : base/7/x86_64
Summary     : Dynamic host configuration protocol software
URL         : http://isc.org/products/DHCP/
License     : ISC
Description : DHCP (Dynamic Host Configuration Protocol) is a protocol which allows
            : individual devices on an IP network to get their own network
            : configuration information (IP address, subnetmask, broadcast address,
            : etc.) from a DHCP server. The overall purpose of DHCP is to make it
            : easier to administer a large network.
            :
            : To use DHCP on your network, install a DHCP service (or relay agent),
            : and on clients run a DHCP client daemon.  The dhcp package provides
            : the ISC DHCP service and relay agent.
```

### 5.4 考察

aaaaaaaaaaaaaaaaaaaaa

# 6 [課題 4] ファイアウォールの設定

## 6.1 目的と概要

aaaaaaaaaaaaaaaaaaaaa

## 6.2 実験方法

machine1 について下記実験を行った。

#### 6.2.1 パケット転送の有効化

設定ファイル /etc/sysctl.d/10-ipv4.conf を作成し、以下を追加した。

```
net.ipv4.ip_forward=1
```

設定ファイルを変更したら、以下のコマンドで有効化した。

```
sysctl –system
reboot
```

### 6.2.2 firewalld から iptables-services への変更

以下のコマンドを実行し、Netfilter のフロントエンドを firewalld から iptables に変更した。

```
systemctl stop firewalld
systemctl disable firewalld
yum install iptables-services
systemctl start iptables
systemctl status iptables
systemctl enable iptables
```

### 6.2.3 サンプルの実行

サンプルファイルを以下のコマンドで machine1 にコピーして実行した。

```
scp bv0572197@ssh.ice.nuie.nagoya-u.ac.jp:/pub1/jikken/cs-net/iptables-sample.sh
sh iptables-sample.sh
```

### 6.2.4 iptables の設定状況の確認

以下のコマンドを実行し、iptables の設定状況を確認した。

```
iptables -L -n
```

以下のコマンドを実行し、現在のルールを確認した。

```
iptables-save
```

### 6.2.5 設定スクリプトの修正

外部のパソコンで以下のようなファイルを作り、scp コマンドで machine1 にコピーした。

```
#!/bin/sh

PATH=/sbin:/bin:/usr/bin:/usr/sbin

## 変数の定義
EXTERNAL_INTERFACE="enp1s0"        # 外側インタフェースの名前
DMZ_INTERFACE="enp2s0"             # DMZ インタフェースの名前
INTERNAL_INTERFACE="enp3s0"        # 内側インタフェースの名前

# 外側インタフェースのアドレスIP
IPADDR=‘ip addr show $EXTERNAL_INTERFACE | \
 sed -e ’s/^.*inet \([^ \/]*\).*$/\1/p’ -e d‘
# 内部ネットワーク・アドレス
INTERNAL_LAN=‘ip addr show $INTERNAL_INTERFACE | \
 sed -e ’s/^.*inet \([^ ]*\).*$/\1/p’ -e d‘
```

```
# ネットワーク・アドレスDMZ
DMZ_LAN=‘ip addr show $DMZ_INTERFACE | \
 sed -e ’s/^.*inet \([^ ]*\).*$/\1/p’ -e d‘

ANYWHERE="0.0.0.0/0"

## 以下の設定を実行している間はパケットの転送を停止する
echo 0 > /proc/sys/net/ipv4/ip_forward

## すでに設定されているルールを消去する
iptables -F
iptables -F -t nat

## ポリシーの初期設定 -> しない場合の扱いmatch
iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP

## ループバック・インタフェースの入出力を許可する
iptables -A INPUT -i lo -j ACCEPT
iptables -A OUTPUT -o lo -j ACCEPT

################################################################################
##
## チェーンの設定（デフォルト拒否）INPUT
##

iptables -A INPUT -i $EXTERNAL_INTERFACE -p tcp -m state --state NEW -m tcp \
    --dport 22 -j ACCEPT
iptables -A INPUT -i $INTERNAL_INTERFACE -p tcp -m state --state NEW -m tcp \
    --dport 22 -j ACCEPT
iptables -A INPUT -i $INTERNAL_INTERFACE -p udp -m state --state NEW -m udp \
    --dport 67 -j ACCEPT
iptables -A INPUT -i $INTERNAL_INTERFACE -p icmp -j ACCEPT
iptables -A INPUT -i $DMZ_INTERFACE -p udp -m state --state NEW -m udp \
    --dport 67 -j ACCEPT
iptables -A INPUT -i $DMZ_INTERFACE -p icmp -j ACCEPT
iptables -A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT


################################################################################
##
## チェーンの設定（デフォルト拒否）OUTPUT
##

iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p tcp -m state --state NEW -m tcp \
    --dport 22 -j ACCEPT
iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p tcp -m state --state NEW -m tcp \
    --dport 80 -j ACCEPT
iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p tcp -m state --state NEW -m tcp \
    --dport 443 -j ACCEPT
iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p udp -m state --state NEW -m udp \
    --dport 53 -j ACCEPT
iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p icmp -j ACCEPT
iptables -A OUTPUT -o $INTERNAL_INTERFACE -p icmp -j ACCEPT
iptables -A OUTPUT -o $DMZ_INTERFACE -p tcp -m state --state NEW -m tcp \
    --dport 22 -j ACCEPT
```

```
iptables -A OUTPUT -m state --state RELATED,ESTABLISHED -j ACCEPT


##############################################################################
##
## チェーンの設定（デフォルト拒否）FORWARD
##

iptables -A FORWARD -i $INTERNAL_INTERFACE -o $EXTERNAL_INTERFACE -p tcp \
 -m state --state NEW,ESTABLISHED -m tcp --dport 22 -j ACCEPT

iptables -A FORWARD -i $INTERNAL_INTERFACE -o $EXTERNAL_INTERFACE -p udp \
-m state --state NEW,ESTABLISHED -m udp --dport 53 -j ACCEPT
iptables -A FORWARD -i $INTERNAL_INTERFACE -o $EXTERNAL_INTERFACE -p tcp \
-m state --state NEW,ESTABLISHED -m tcp --dport 80 -j ACCEPT
iptables -A FORWARD -i $INTERNAL_INTERFACE -o $EXTERNAL_INTERFACE -p tcp \
-m state --state NEW,ESTABLISHED -m tcp --dport 443 -j ACCEPT
iptables -A FORWARD -i $INTERNAL_INTERFACE -o $EXTERNAL_INTERFACE -p icmp \
-j ACCEPT
iptables -A FORWARD -i $EXTERNAL_INTERFACE -o $DMZ_INTERFACE -p tcp \
-m state --state NEW,ESTABLISHED -m tcp --dport 80 -j ACCEPT
iptables -A FORWARD -i $EXTERNAL_INTERFACE -o $DMZ_INTERFACE -p tcp \
-m state --state NEW,ESTABLISHED -m tcp --dport 443 -j ACCEPT
iptables -A FORWARD -i $DMZ_INTERFACE -o $EXTERNAL_INTERFACE -p tcp \
-m state --state NEW,ESTABLISHED -m tcp --dport 22 -j ACCEPT
iptables -A FORWARD -i $DMZ_INTERFACE -o $EXTERNAL_INTERFACE -p udp \
-m state --state NEW,ESTABLISHED -m udp --dport 53 -j ACCEPT
iptables -A FORWARD -i $DMZ_INTERFACE -o $EXTERNAL_INTERFACE -p tcp \
-m state --state NEW,ESTABLISHED -m tcp --dport 80 -j ACCEPT
iptables -A FORWARD -i $DMZ_INTERFACE -o $EXTERNAL_INTERFACE -p tcp \
-m state --state NEW,ESTABLISHED -m tcp --dport 443 -j ACCEPT
iptables -A FORWARD -i $DMZ_INTERFACE -o $EXTERNAL_INTERFACE -p icmp -j ACCEPT
iptables -A FORWARD -i $INTERNAL_INTERFACE -o $EXTERNAL_INTERFACE -p \
icmp -j ACCEPT
iptables -A FORWARD -i $INTERNAL_INTERFACE -o $DMZ_INTERFACE -m state \
--state NEW,ESTABLISHED -j ACCEPT

iptables -A FORWARD -i $DMZ_INTERFACE -o $INTERNAL_INTERFACE -m state \
--state RELATED,ESTABLISHED -j ACCEPT
iptables -A FORWARD -i $DMZ_INTERFACE -o $EXTERNAL_INTERFACE -m state \
--state RELATED,ESTABLISHED -j ACCEPT
iptables -A FORWARD -i $EXTERNAL_INTERFACE -o $DMZ_INTERFACE -m state \
--state RELATED,ESTABLISHED -j ACCEPT
iptables -A FORWARD -i $EXTERNAL_INTERFACE -o $INTERNAL_INTERFACE -m \
state --state RELATED,ESTABLISHED -j ACCEPT
##############################################################################
##
## の設定NAT
##

iptables -A POSTROUTING -t nat -s $INTERNAL_LAN -o $EXTERNAL_INTERFACE -j SNAT \
    --to-source $IPADDR

iptables -A POSTROUTING -t nat -s $DMZ_LAN -o $EXTERNAL_INTERFACE -j SNAT \
    --to-source $IPADDR
```

```
iptables -A PREROUTING -t nat -p tcp --dport 80 -i $EXTERNAL_INTERFACE -j DNAT \
 --to-destination 192.168.150.2
iptables -A PREROUTING -t nat -p tcp --dport 443 -i $EXTERNAL_INTERFACE -j DNAT \
 --to-destination 192.168.150.2

################################################################################
##
## 設定の保存
##
#/etc/init.d/iptables save active

## パケットの転送を開始する
echo 1 > /proc/sys/net/ipv4/ip_forward

exit 0
```

#### 6.2.6 確認用ソフトウェアのインストール

machine1 及び machine2 において以下のコマンドを実行し、ファイアウォール設定の確認時に利用するクライアントソフトウェアをインストールした。

```
yum info lynx
yum install lynx
yum info bind-utils
yum install bind-utils
rpm -ql bind-utils — grep /usr/bin
```

#### 6.2.7 ファイアウォールの動作状況の最終確認

ここで、lynx コマンドや ping コマンドを用いて、machine1 から外部ネットワークへの接続、machine2 から外部ネットワークへの接続、machine3 から外部ネットワークへの接続、machine3 から machine1 への接続、machine3 から machine2 への接続、外部ネットワークから machine1 への接続が可能であることを確認した。

#### 6.2.8 最終状態を保存

以下のコマンドを実行し、最終状態を保存した。

```
iptables-save > /etc/sysconfig/iptables
```

次に machine2 について下記実験を行った。

#### 6.2.9 firewalld の動作状態を確認

以下のコマンドを実行し、firewalld の動作状態を確認した。

```
systemctl status firewalld
```

#### 6.2.10　現在のゾーンの許可状態を確認

以下のコマンドを実行し、現在のゾーンの許可状態を確認した。

```
firewalld-cmd –get-active-zones
```

#### 6.2.11　必要なサービス許可設定の追加

以下のコマンドを実行し、必要なサービス (http や https など) を追加した。

```
firewall-cmd –add-service=dhcpv6-client
firewall-cmd –add-service=http
firewall-cmd –add-service=https
firewall-cmd –list-all
```

#### 6.2.12　動作確認

SSH、HTTP/HTTPS、DHCPv6client、すべての ICMP パケットのみ許可し、その他の接続は全て拒否されることを確認した。

#### 6.2.13　各設定の永続化

以下のコマンドを実行し、各設定の永続化を行った。

```
firewall-cmd –permanent –add-sevice=dhcpv6-client
firewall-cmd –permanent –add-service=http
firewall-cmd –permanent –add-service=https
```

#### 6.2.14　firewall の再起動

以下のコマンドを実行し、firewall の再起動を行った。

```
firewalld-cmd –reload
```

### 6.3　実験結果

#### 6.3.1　パケット転送の有効化を行った結果

```
[root@icesc16 ~]# sysctl --system
* Applying /usr/lib/sysctl.d/00-system.conf ...
* Applying /usr/lib/sysctl.d/10-default-yama-scope.conf ...
kernel.yama.ptrace_scope = 0
* Applying /etc/sysctl.d/10-ipv4.conf ...
net.ipv4.ip_forward = 1
* Applying /usr/lib/sysctl.d/50-default.conf ...
kernel.sysrq = 16
kernel.core_uses_pid = 1
```

```
net.ipv4.conf.default.rp_filter = 1
net.ipv4.conf.all.rp_filter = 1
net.ipv4.conf.default.accept_source_route = 0
net.ipv4.conf.all.accept_source_route = 0
net.ipv4.conf.default.promote_secondaries = 1
net.ipv4.conf.all.promote_secondaries = 1
```

### 6.3.2 iptables の設定状況を確認した結果

```
[root@icesc16 ~]# iptables -L -n
Chain INPUT (policy DROP)
target     prot opt source               destination
ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0
ACCEPT     tcp  --  0.0.0.0/0            0.0.0.0/0           state NEW tcp dpt:22
ACCEPT     tcp  --  0.0.0.0/0            0.0.0.0/0           state NEW tcp dpt:22
ACCEPT     udp  --  0.0.0.0/0            0.0.0.0/0           state NEW udp dpt:67
ACCEPT     icmp --  0.0.0.0/0            0.0.0.0/0
ACCEPT     udp  --  0.0.0.0/0            0.0.0.0/0           state NEW udp dpt:67
ACCEPT     icmp --  0.0.0.0/0            0.0.0.0/0
ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0           state RELATED,ESTABLISHED

Chain FORWARD (policy DROP)
target     prot opt source               destination
ACCEPT     tcp  --  0.0.0.0/0            0.0.0.0/0           state NEW,ESTABLISHED tcp dpt:22
ACCEPT     icmp --  0.0.0.0/0            0.0.0.0/0
ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0           state RELATED,ESTABLISHED

Chain OUTPUT (policy DROP)
target     prot opt source               destination
ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0
ACCEPT     tcp  --  0.0.0.0/0            0.0.0.0/0           state NEW tcp dpt:22
ACCEPT     tcp  --  0.0.0.0/0            0.0.0.0/0           state NEW tcp dpt:80
ACCEPT     tcp  --  0.0.0.0/0            0.0.0.0/0           state NEW tcp dpt:443
ACCEPT     udp  --  0.0.0.0/0            0.0.0.0/0           state NEW udp dpt:53
ACCEPT     icmp --  0.0.0.0/0            0.0.0.0/0
ACCEPT     icmp --  0.0.0.0/0            0.0.0.0/0
ACCEPT     tcp  --  0.0.0.0/0            0.0.0.0/0           state NEW tcp dpt:22
ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0           state RELATED,ESTABLISHED
```

### 6.3.3 iptables のルールを表示した結果

```
root@icesc16 ~]# iptables-save
# Generated by iptables-save v1.4.21 on Thu May 23 14:28:44 2019
*nat
:PREROUTING ACCEPT [0:0]
:INPUT ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
:POSTROUTING ACCEPT [0:0]
-A POSTROUTING -s 192.168.200.0/24 -o enp1s0 -j SNAT --to-source 192.168.100.16
COMMIT
# Completed on Thu May 23 14:28:44 2019
# Generated by iptables-save v1.4.21 on Thu May 23 14:28:44 2019
*filter
:INPUT DROP [0:0]
:FORWARD DROP [0:0]
```

```
:OUTPUT DROP [0:0]
-A INPUT -i lo -j ACCEPT
-A INPUT -i enp1s0 -p tcp -m state --state NEW -m tcp --dport 22 -j ACCEPT
-A INPUT -i enp3s0 -p tcp -m state --state NEW -m tcp --dport 22 -j ACCEPT
-A INPUT -i enp3s0 -p udp -m state --state NEW -m udp --dport 67 -j ACCEPT
-A INPUT -i enp3s0 -p icmp -j ACCEPT
-A INPUT -i enp2s0 -p udp -m state --state NEW -m udp --dport 67 -j ACCEPT
-A INPUT -i enp2s0 -p icmp -j ACCEPT
-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A FORWARD -i enp3s0 -o enp1s0 -p tcp -m state --state NEW,ESTABLISHED -m tcp --dport 22
-j ACCEPT
-A FORWARD -i enp3s0 -o enp1s0 -p icmp -j ACCEPT
-A FORWARD -i enp1s0 -o enp3s0 -m state --state RELATED,ESTABLISHED -j ACCEPT
-A OUTPUT -o lo -j ACCEPT
-A OUTPUT -o enp1s0 -p tcp -m state --state NEW -m tcp --dport 22 -j ACCEPT
-A OUTPUT -o enp1s0 -p tcp -m state --state NEW -m tcp --dport 80 -j ACCEPT
-A OUTPUT -o enp1s0 -p tcp -m state --state NEW -m tcp --dport 443 -j ACCEPT
-A OUTPUT -o enp1s0 -p udp -m state --state NEW -m udp --dport 53 -j ACCEPT
-A OUTPUT -o enp1s0 -p icmp -j ACCEPT
-A OUTPUT -o enp3s0 -p icmp -j ACCEPT
-A OUTPUT -o enp2s0 -p tcp -m state --state NEW -m tcp --dport 22 -j ACCEPT
-A OUTPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
COMMIT
# Completed on Thu May 23 14:28:44 2019
```

#### 6.3.4 firewalld の動作状態の確認結果

```
[root@localhost ~]# systemctl status firewalld
firewalld.service - firewalld - dynamic firewall daemon
   Loaded: loaded (/usr/lib/systemd/system/firewalld.service; enabled; vendor preset: enabled)
   Active: mactive (running) since Thu 2019-05-23 13:00:52 JST; 4h 49min ago
     Docs: man:firewalld(1)
 Main PID: 762 (firewalld)
   CGroup: /system.slice/firewalld.service └─
           762 /usr/bin/python -Es /usr/sbin/firewalld --nofork --nopid
```

#### 6.3.5 現在のゾーンの許可状態の確認結果

```
[root@www6 ~]# firewalld-cmd --get-active-zones
public
  interfaces: enp3s0
```

#### 6.3.6 各設定の永続化設定の結果

```
[root@icesc16 ~]# firewall-cmd --permanent --add-service=dhcp6-client
Warning: ALREADY_ENABLED: dhcpv6-client
success
[root@icesc16 ~]# firewall-cmd --permanent --add-service=http
success
[root@icesc16 ~]# firewall-cmd --permanent --add-service=https
success
```

### 6.4 考察

aaaaaaaaaaaaaaaaaaaaa

# 7 [課題 5]WWW サービスの設定

## 7.1 目的と概要

aaaaaaaaaaaaaaaaaaaaa

## 7.2 実験方法

machine2 で下記の実験を行った。

#### 7.2.1 firewall の停止

以下のコマンドを実行し、firewall の停止を行った。

```
systemctl stop firewalld
systemctl status firewalld
```

#### 7.2.2 WWW サーバのインストール

以下のコマンドを実行し、Apache WWW サーバと https 対応モジュールをインストールした。

```
yum info httpd
yum info mod_ssl
yum install httpd
yum install mod_ssl
```

#### 7.2.3 SSL/TLS 自己署名証明書の作成

/etc/pki/tls/certs に移動した。

以下のコマンドを実行し、秘密鍵を作成した。

```
openssl genrsa -aes128 1024 > server.key
```

以下のコマンドを実行し、パスフレーズの除去を行った。

```
openssl rsa -utf8 -in server.key -out server.key
```

以下のコマンドを実行し、CSR を作成した。

```
openssl req -utf8 -new -key server.key -out server.csr
```

以下のコマンドを実行し、有効期間 1 年のサーバ証明書を作成した。

```
openssl x509 -in server.csr -out server.crt -days 365 -req -signkey server.key
```

#### 7.2.4 WWW サーバの設定ファイルにサーバ証明書を指定

設定ファイル /etc/httpd/conf.d/ssl.conf を以下のように変更した。

```
SSLProtocol +TLSv1.2
SSLCertificateFile /etc/pki/tls/certs/server.crt
SSLCertificateKeyFile /etc/pki/tls/certs/server.key
```

#### 7.2.5 WWW サーバの主設定ファイルを修正

Apache WWW サーバ主設定ファイル/etc/httpd/conf/httpd.conf の ServerName ディレクティブを group6a に変更した。

#### 7.2.6 WWW サーバの起動

以下のコマンドを実行し、WWW サーバを実行した。

```
systemctl restart httpd
```

#### 7.2.7 動作確認

machine3 と外部テスト端末から http、https 両方の動作を確認した。

#### 7.2.8 WWW サーバのブート時自動起動を指定

以下のコマンドを実行し、WWW サーバのブート時自動起動を指定した。

```
systemctl enable httpd
```

## 7.3　実験結果

### 7.3.1　machine3 から https サーバに接続した結果

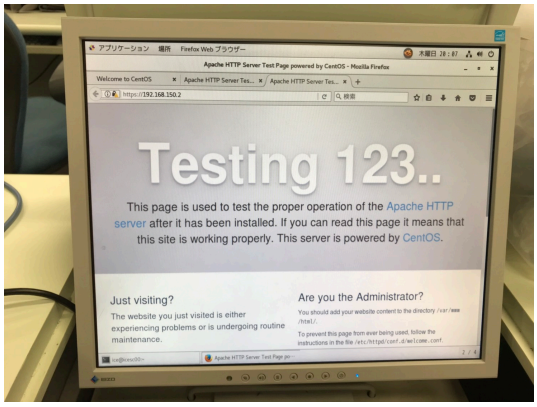


図 1　machine3 から https サーバに接続した結果

### 7.3.2　外部テスト端末から http サーバ接続した結果

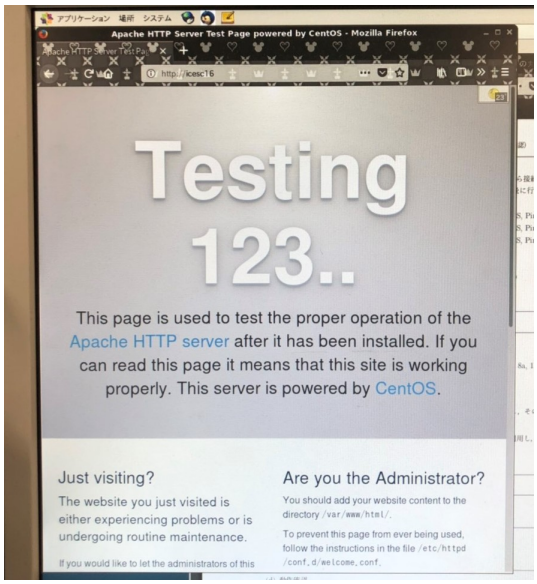


図 2　外部テスト端末から http サーバに接続した結果

図 1、2 のように接続が確認できた。

### 7.3.3　自己署名証明書の確認

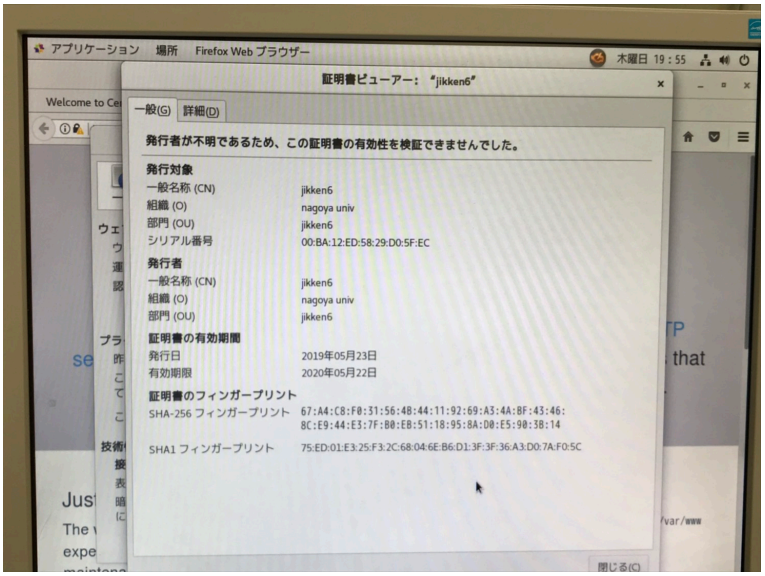


図 3　証明書の詳細

設定した名前であることが確認できた。

### 7.3.4　WWW サーバのブート時自動起動を指定した結果

再起動を行い、systemctl status httpd コマンドで確認した結果、active になっていることが確認できた。

## 7.4　考察

aaaaaaaaaaaaaaaaaaaaa

# 8　まとめ

aaaaaaaaaaaaaaaaaaaaa

# 9　[調査課題 1]TCP パケットのヘッダ情報及び IP パケットのヘッダ情報

ここでは、TCP パケットのヘッダ情報及び IP パケットのヘッダ情報について調査した

## 9.1　TCP パケットのヘッダ情報

TCP パケットのヘッダ情報の概要は以下の図で示される。

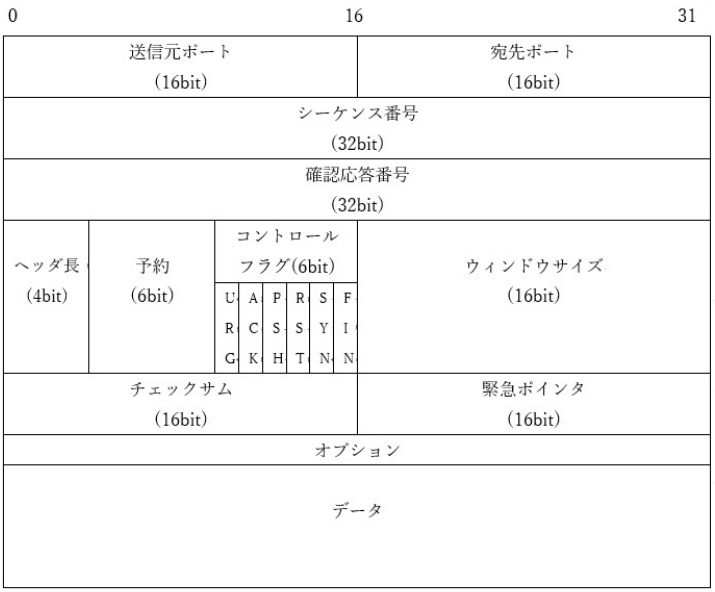


図 4　TCP パケットヘッダ情報の概要

### 9.1.1　送信元ポート

送信元のアプリケーションを識別するための番号

### 9.1.2　宛先ポート

宛先のアプリケーションを識別するための番号

### 9.1.3　シーケンス番号

送信したデータの順序を示す番号で、送信するデータ 1 バイトごとにシーケンス番号を 1 つずつ増やし、$2^{32}$ を超えるとまた同じ値を繰り返す。

### 9.1.4　確認応答番号

受信したデータに対して受信位置をバイト位置で表すフィールドで、受信が完了したデータ位置のシーケンス番号 +1 を返す。

### 9.1.5　ヘッダ長

TCP データが始まる位置を表すフィールドで、TCP ヘッダの直後にデータ部が続くため、TCP データの長さを示している。

### 9.1.6　コントロールフラグ

URG、ACK、PSH、RST、SYN、FIN の 6 つのビットで構成されており、これらのビットは 1 が入るときに意味を成す。

#### 9.1.7 URG

urgent の略で、緊急データが含まれていることを示すフラグ。デフォルトでは 0 が入っており、1 で ON になる。

#### 9.1.8 ACK

acknowledge の略で、有効な ACK 番号が TCP ヘッダに含まれていることを示すフラグ。TCP の 3 ウェイハンドシェイク時の最初を除き、TCP パケットは全て ACK のフラグが ON になっている。

#### 9.1.9 PSH

push の略で、受信したデータをバッファリングせずにすぐにアプリケーションに引き渡すように要求するためのフラグ。

#### 9.1.10 RST

reset の略で、TCP 接続を中断、拒否したい場合にセットされるフラグ。RST フラグを ON にしたパケットを送信することで、現在の TCP 接続を強制終了させることができる。

#### 9.1.11 SYN

synchronize の略で、TCP の 3 ウェイシェイクハンド時のオープン処理の開始に双方のそれぞれが SYN フラグを ON にして ACK 番号を同期させる。以降のパケットにはセットされない。

#### 9.1.12 FIN

finish の略で、TCP 接続を終了させるためセットされるフラグ。双方から FIN フラグが ON である TCP パケットを送信することで TCP 接続は終了する。

#### 9.1.13 ウィンドウサイズ

受信側が一度に受信することができるデータ量を送信側に通知するために使用される。送信側は、この値のデータ量を超えて送信することはできない。

#### 9.1.14 チェックサム

TCP ヘッダとデータ部分のエラーチェックを行うために使用される値が入れられる。

#### 9.1.15 緊急ポインタ

コントロールフラグの URG が 1 である場合にのみ使用されるフィールドであり、緊急データの開始位置を示す情報が入れられる。

#### 9.1.16 オプション

TCP 接続の特性を設定するために利用される可変長のフィールド。(性能を向上させるために利用する)MSS のやり取りなどに用いられる。TCP ヘッダの長さを 32 ビットの整数にするように必要に応じて空データのパディング (0) が埋められる。

### 9.1.17　データ

TCP のデータ部であり、TCP 接続がタイムアウトしても切断されないようにデータを含まない TCP ヘッダだけのパケットを送る場合もある。

## 9.2　IP パケットのヘッダ情報

次に、IP パケットのヘッダ情報についてまとめる (文献 [1] 参照)
IP パケットのヘッダ情報の概要は以下の図で示される。

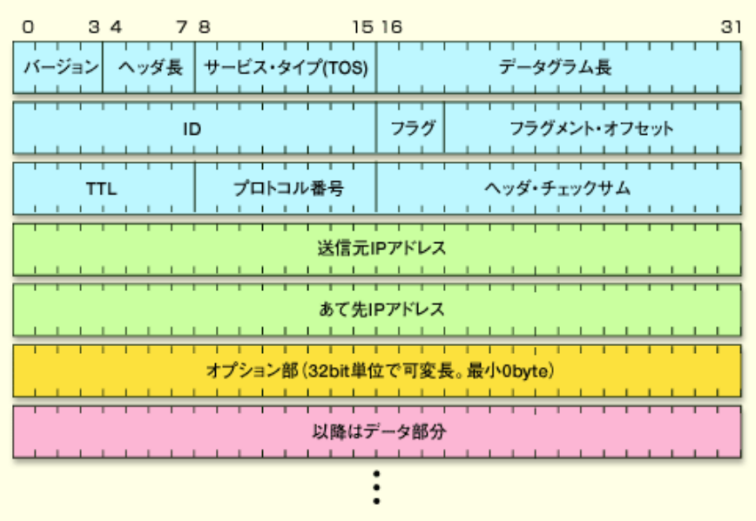


図 5　IP パケットヘッダ情報の概要 (文献 [1] から引用)

### 9.2.1　バージョン (4bit)

バージョンフィールドは IP プロトコルのバージョンを表現するために使われる。IPv4 を用いる場合はこのフィールドでは常に 4(2 進数では 0010) になっており、IPv6 を用いる場合は常に 6(2 進数では 0110) になっているため、同ネットワーク媒体に IPv4 と IPv6 のパケットは混在していても区別することができる。

### 9.2.2　ヘッダ長 (4bit)

ヘッダ長フィールドは、IP ヘッダ部分 (固定長部分 + オプション部分) のサイズを表すために使われる。4bit 幅しかないため 0 から 15 までしか表すことができないが、ヘッダの長さは 32bit 単位で数えるため実際の最大長は 60bytes まで表すことが可能である。また、IP ヘッダの固定長部分は常に 20bytes あるためこのフィールドの最小値は 5 であり、最大値は 15 である。

### 9.2.3　サービス・タイプ (TOS)(8bit)

サービス・タイプフィールドは IP パケットの優先度などを表す TOS を指定するために使われる。しかし、現在使われている TCP/IP ネットワークではこのフィールドによる TOS 指定はほとんど使用されておらず

意味を持っていないことが多い。

#### 9.2.4 データグラム長 (16bit)

データグラム長フィールドは IP パケット全体 (IP ヘッダ部分 + データ部分) のサイズを byte 単位で表すために使われる。このフィールドは 16bit 幅であるため IP パケットのサイズの最大長 (送信可能なデータ + ヘッダ) は 64Kbytes である。このフィールドの最小値は 20+ データ部分の長さ (byte) である。

#### 9.2.5 ID(16bit)

ID フィールドは、IP フラグメンテーションの際に、IP パケットを識別するために使われる。サイズの大きなデータを送信する場合、IP フラグメンテーションによって IP パケットをいくつか分割して小さくしてから送信するが、このフラグメント化されたパケットが後で 1 つの IP パケットに再構成するための目印として使われる。

#### 9.2.6 フラグ (3bit)

フラグフィールドはフラグメンテーションにおいて利用される、特殊なフラグ情報を表すために使われる。3bit 分のデータがあるが実際使われているのは 2bit である。

- MFbit
  MFbit はフラグメントがさらに続くかどうかを表すために使われる。1 つの IP パケットをいくつか分割した場合、最後部のパケットではこの MFbit を 0 にし、その他のパケットでは MFbit を 1 にする。つまりこの bit が 1 ならばフラグメント化された IP パケットがさらに後ろに続くことを表している。
- DFbit
  DFbit は IP パケットを分断してはいけないという指示を与えるために使われる。IP パケットのフラグメント化とその再構成は少なからず複雑な操作が必要になり、性能の低いコンピュータなどではその実現が難しい。こういう場合に DFbit を使うことでフラグメンテーション処理を行わずに TCP/IP 通信を実現することが可能である。

#### 9.2.7 フラグメント・オフセット (16bit)

フラグメント・オフセットフィールドは、フラグメント化された IP パケットにおいてフラグメントのどの部分が IP パケット中に含まれているかを示すオフセット数値を表すために使われる。

#### 9.2.8 TTL(8bit)

TTL フィールドは IP パケットの寿命を表すために使われる。IP パケットを送信するコンピュータはこのフィールドに適当な数値をセットし、その決められた寿命の間だけしか IP パケットが生存できないようにする。

#### 9.2.9 プロトコル番号 (8bit)

プロトコル番号フィールドは、上位のトランスポート層のネットワーク・プロトコルの種類を表す番号を格納するために使われる。

#### 9.2.10　ヘッダ・チェックサム (16bit)

ヘッダ・チェックサムフィールドはヘッダ部分 (固定部分 + オプション部分) のチェックサム (整合性を検査するためのデータ) を表すために使われる。このチェックサム計算では「1 の補数演算」という特別なアルゴリズムが利用される。通常のコンピュータでは「2 の補数演算」によるチェックサムがよく使われるが IP では計算が高速かつ十分な信頼性が確保できる「1 の補数演算」が使われる。

#### 9.2.11　送信元 IP アドレス (32bit)

送信元 IP アドレスフィールドは、送信元のコンピュータの IP アドレスを表すために使われる。

#### 9.2.12　宛先 IP アドレス (32bit)

宛先 IP アドレスフィールドは、IP パケットの送信先コンピュータの IP アドレスを表すために使われる。

#### 9.2.13　オプション (32bit)

オプションフィールドは、IP パケットの送信に伴い、様々な付加的機能を実現するために使われる。通常はオプションは利用されないが、IP パケットの通過のログやルーティングなどで利用される。

## 10　[調査課題 2]TCP/IP 通信におけるブロードキャストの役割

aaaaaaaaaaaaaaaaaaaaa

## 11　[調査課題 3]TCP パケット送信の過程

aaaaaaaaaaaaaaaaaaaaa

## 12　[調査課題 4] 各サービスにおけるセキュリティ強化の動向

aaaaaaaaaaaaaaaaaaaaa

## 13　[調査課題 5]SSL/TLS を用いるプロトコル

aaaaaaaaaaaaaaaaaaaaa

## 参考文献


[1] atmarkIT IP パケットの構造と IP フラグメンテーション
https://www.atmarkit.co.jp/ait/articles/0304/04/news001_2.html(2019 年 5 月 22 日参照)

[2] 日経 xTECH　マウント
https://tech.nikkeibp.co.jp/it/article/Keyword/20070207/261214/(2019 年 5 月 25 日参照)

[3] IBM
https://www.ibm.com/developerworks/jp/linux/library/l-lvm2/index.html(2019 年 5 月 25 日参照)