Ver. v0.1.0 / 2025-10-02

3B情2 ネットワーク構築

嶋 直樹

2025-10-02

実験概要

1. 実験の全体目標

学生は2回の実験（各180分）を通して、多層的な内部ネットワークの構築、静的ルーティング、DNSによる名前解決、そしてNAPTによるインターネット接続までを実践的に体験し、その仕組みを深く理解することを目標とする。

2. ネットワーク構成

実験で構築するネットワークの全体像は以下の通りである。

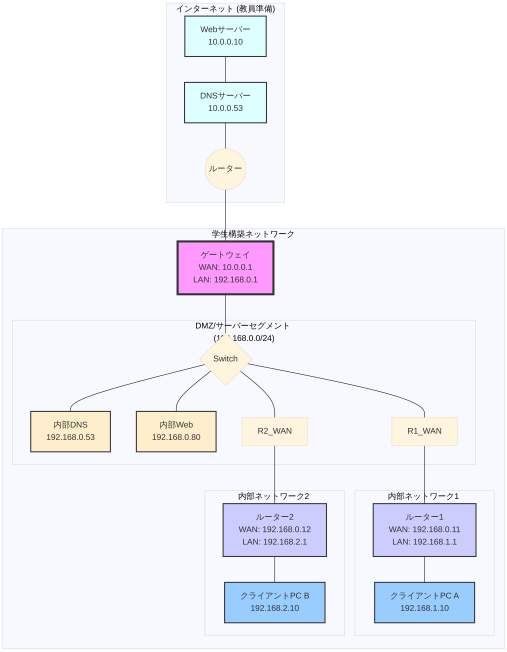

Figure 1: ネットワーク構成図

2.1. 構成要素

- インターネット領域（教員準備）:
  - 外部**DNS**サーバー **(`ExtDNS`):** `example.com` などのドメイン情報を持つDNSサーバー。
  - 外部**Web**サーバー **(`ExtWeb`):** インターネット上にあるWebサイトの役割を担うサーバー。
- 学生構築ネットワーク:
  - ゲートウェイ **(`GW`):** 組織全体の出入り口。WAN側とLAN側を持ち、NAPT機能で内部ネットワークをインターネットに接続する。
  - トランジットセグメント: GWと内部ルーターを接続する中継ネットワーク。

- 内部**DNS**サーバー **(IntDNS):** `corp.local` のような内部ドメインの名前解決を担う。
- 内部**Web**サーバー **(IntWeb):** 組織内でのみアクセス可能なWebサーバー。
- ルーター**1 (R1)** / ルーター**2 (R2):** 内部ネットワークをさらにセグメントに分割する。
- クライアント**PC A/B:** 各セグメントに属する端末。

2.2. IPアドレッシング計画

機器名	役割	インターフェース	IPアドレス	サブネットマスク	ゲートウェイ	DNSサーバー
インターネット領域						
ExtDNS	外部DNS	-	10.0.0.53	255.255.255.0	10.0.0.254	(自身)
ExtWeb	外部Web	-	10.0.0.80	255.255.255.0	10.0.0.254	10.0.0.53
学生構築ネットワーク						
GW	ゲートウェイ	WAN (eth0)	10.0.0.1	255.255.255.0	10.0.0.254	10.0.0.53
		LAN (eth1)	192.168.0.1	255.255.255.0	-	-
IntDNS	内部DNS	eth0	192.168.0.53	255.255.255.0	192.168.0.1	(自身), 10.0.0.53
IntWeb	内部Web	eth0	192.168.0.80	255.255.255.0	192.168.0.1	192.168.0.53
R1	ルーター1	WAN (eth0)	192.168.0.11	255.255.255.0	192.168.0.1	-
		LAN (eth1)	192.168.1.1	255.255.255.0	-	-
R2	ルーター2	WAN (eth0)	192.168.0.12	255.255.255.0	192.168.0.1	-
		LAN (eth1)	192.168.2.1	255.255.255.0	-	-
ClientA	クライアントA	eth0	192.168.1.10	255.255.255.0	192.168.1.1	192.168.0.53
ClientB	クライアントB	eth0	192.168.2.10	255.255.255.0	192.168.2.1	192.168.0.53

3. 各回の実験テーマ

第**1**回：内部ネットワークの構築と静的ルーティング

- 目標**:** ゲートウェイ、内部サーバー、ルーター、クライアントPCを接続し、静的ルーティングによって内部ネットワーク全体の通信を確立する。
- 内容**:**
  1. 各PC/サーバーへのIPアドレス設定。
  2. ルーター1およびルーター2でIPフォワーディングを有効化する。
  3. クライアントPC A/Bにデフォルトゲートウェイを設定する。
  4. ゲートウェイ(`GW`)に内部ネットワーク(`192.168.1.0/24`, `192.168.2.0/24`)への静的ルートを設定する。
  5. `ping` と `traceroute` を用いて ClientA <=> ClientB, ClientA <=> IntWeb などの通信を確認し、パケットの経路を追跡する。
  6. 内部DNSサーバーを構築し、`intweb.corp.local` のような内部ドメインの名前解決ができることを確認する。

第**2**回：**NAPT**によるインターネット接続

- 目標**:** ゲートウェイにNAPT（IPマスカレード）を設定し、内部ネットワークからインターネット上のサーバーへアクセスできるようにする。
- 内容**:**
  1. ゲートウェイ(`GW`)で `iptables` を用いてNAPT（IPマスカレード）を設定する。
  2. クライアントPC Aから外部Webサーバー(`10.0.0.80`)へ `ping` が通ることを確認する。

3. 外部DNSサーバー(10.0.0.53)を利用して www.example.com のような外部ドメインの名前解決ができることを確認する。
4. クライアントPC Aのブラウザから外部Webサーバーにアクセスする。
5. ゲートウェイで conntrack コマンドや iptables のカウンタを確認し、NAPTによるアドレス変換の様子を観察する。

4. レポート課題

- 考察事項
  - 問1: ClientAからClientBへ ping を送信した際、パケットはどのような経路を辿るか。traceroute の結果を基に、各ルーターのルーティングテーブルがどのように利用されるかを説明せよ。
  - 問2: ゲートウェイ(GW)に内部ネットワーク(192.168.1.0/24, 192.168.2.0/24)への静的ルートを設定しない場合、内部WebサーバーからClientAへの通信はなぜ失敗するのか説明せよ。
  - 問3: NAPTの役割とは何か。内部ネットワークのPCが1つのグローバルIPアドレス(10.0.0.1)を使ってインターネットと通信できる仕組みを、実験結果を交えて説明せよ。
  - 問4: 内部DNSと外部DNSの役割の違いは何か。ClientAが intweb.corp.local と www.example.com の両方にアクセスできるのはなぜか、名前解決のプロセスを追って説明せよ。

実験手順書：第1回内部ネットワークの構築と静的ルーティング

1. 実験の目的

この実験では、複数のルーターとサーバーを含む実践的な多層ネットワークをゼロから構築する。静的ルーティングを設定し、異なるネットワークセグメント間の通信を確立するとともに、内部DNSサーバーによる名前解決の仕組みを学ぶ。今回の目標は以下の通りである。

- 複数のネットワークセグメントにまたがるIPアドレッシングを計画し、設定できる。
- Linux PCをルーターとして機能させ、IPフォワーディングと静的ルートを設定できる。
- ping および traceroute を用いて、複雑なネットワークにおける通信経路を特定し、問題を診断できる。
- DNSサーバーを構築し、内部ドメインの名前解決を実現できる。

2. 実験環境

2.1. 必要な機材

- UbuntuがインストールされたPC: 6台 (GW, R1, R2, IntDNS, IntWeb, ClientA)
  - 注*: ClientB*役の*PC*も必要であるが、*1*台の*PC*で役割を交代しながら実験することも可能である。*GW, R1, R2*役の*PC*は*NIC*が*2*枚必要である。
- スイッチングハブ: 3台
- LANケーブル: 適量

2.2. ネットワーク構成とIPアドレッシング

今回は、以下の図に示すネットワークを構築する。IPアドレスの設定ミスは通信失敗の主な原因となるため、各機器の役割とアドレスを常に確認すること。

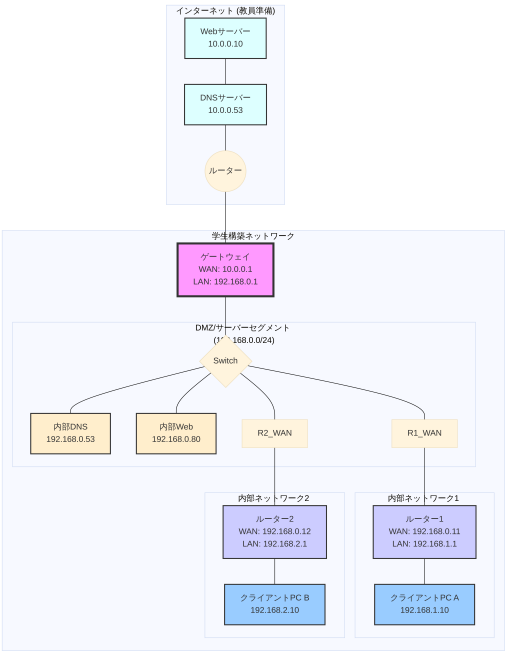

Figure 2: ネットワーク構成図

IPアドレス設定表

機器名	役割	インターフェース	IPアドレス/マスク	ゲートウェイ
GW	ゲートウェイ	LAN (eth1)	`192.168.0.1/24`	-
IntDNS	内部DNS	eth0	`192.168.0.53/24`	`192.168.0.1`
IntWeb	内部Web	eth0	`192.168.0.80/24`	`192.168.0.1`
R1	ルーター1	WAN (eth0)	`192.168.0.11/24`	`192.168.0.1`
		LAN (eth1)	`192.168.1.1/24`	-
R2	ルーター2	WAN (eth0)	`192.168.0.12/24`	`192.168.0.1`
		LAN (eth1)	`192.168.2.1/24`	-
ClientA	クライアントA	eth0	`192.168.1.10/24`	`192.168.1.1`
ClientB	クライアントB	eth0	`192.168.2.10/24`	`192.168.2.1`

3. 実験手順

ステップ**1**：物理接続と**IP**アドレス設定

構成図に従って各機器をLANケーブルで接続した後、すべてのPCに上記の表通りIPアドレスを設定する。ここではインターフェース名を `eth0`, `eth1` としているが、`ip a` で確認した実際の名前に置き換えること。

1. ルーター/ゲートウェイ **(GW, R1, R2)**

```
# GWでの設定例
sudo ip addr add 192.168.0.1/24 dev eth1
sudo sysctl -w net.ipv4.ip_forward=1

# R1での設定例
sudo ip addr add 192.168.0.11/24 dev eth0
sudo ip addr add 192.168.1.1/24 dev eth1
sudo sysctl -w net.ipv4.ip_forward=1

# R2での設定例
sudo ip addr add 192.168.0.12/24 dev eth0
sudo ip addr add 192.168.2.1/24 dev eth1
sudo sysctl -w net.ipv4.ip_forward=1
```

2. サーバー **(IntDNS, IntWeb)**

```
# IntDNSでの設定例
sudo ip addr add 192.168.0.53/24 dev eth0
```

3. クライアント **(ClientA, ClientB)**

```
# ClientAでの設定例
sudo ip addr add 192.168.1.10/24 dev eth0
```

ステップ**2**：静的ルーティング設定

各機器が「自分宛てでないパケットをどこへ送るべきか」を把握するために、ルーティング情報を設定する。

1. クライアントとサーバーのデフォルトゲートウェイ設定

各機器は、自分の所属するLANの出口となるルーターのIPアドレスをデフォルトゲートウェイとして設定する。

```
# ClientAでの設定例
sudo ip route add default via 192.168.1.1

# ClientBでの設定例
sudo ip route add default via 192.168.2.1

# IntDNS, IntWebでの設定例
sudo ip route add default via 192.168.0.1
```

2. ルーターのルーティング設定

ルーターは、より複雑な経路情報を持つ。

```
# R1, R2のデフォルトゲートウェイ設定
# R1, R2ともに、未知の宛先はすべてGW(192.168.0.1)に送る
sudo ip route add default via 192.168.0.1

# GWの静的ルート設定
# GWは、192.168.1.0/24宛のパケットはR1(192.168.0.11)へ、
# 192.168.2.0/24宛はR2(192.168.0.12)へ送るように設定する
sudo ip route add 192.168.1.0/24 via 192.168.0.11
sudo ip route add 192.168.2.0/24 via 192.168.0.12
```

ステップ**3**：内部ネットワーク全体の疎通確認

`ping` と `traceroute` を使い、設定が正しいか確認する。

1. **ClientA**から様々な宛先へ **`traceroute`** を実行せよ。

```
# ClientAから同じセグメントのルーター(R1)へ
traceroute 192.168.1.1

# ClientAからGWへ
traceroute 192.168.0.1
# 経由地にR1(192.168.1.1)が表示されるはず

# ClientAから内部Webサーバーへ
traceroute 192.168.0.80
# 経由地にR1(192.168.1.1)とGW(192.168.0.1)が表示されるはず

# ClientAからClientBへ　最重要
traceroute 192.168.2.10
# 経由地にR1→GW→R2が表示されることを確認
```

2. **ping** で通信できることを確認する。上記 traceroute で確認したすべての宛先に ping が通るはずである。もし失敗する場合、traceroute の結果を手掛かりに、どの区間で問題が起きているか（IPアドレス設定ミス、ルート設定漏れなど）を切り分けること。

ステップ**4**：内部**DNS**サーバーの構築

IPアドレスではなく、分かりやすい名前でサーバーにアクセスできるようにする。

1. **IntDNS**サーバーに**BIND9**をインストールする。 bash # IntDNS**で実行** sudo apt update sudo apt install bind9
2. ゾーンファイルの設定を行う。 corp.local という内部ドメインを管理する設定を追加する。

```
# /etc/bind/named.conf.local に以下を追記
sudo nano /etc/bind/named.conf.local
```

```
zone "corp.local" {
    type master;
    file "/etc/bind/db.corp.local";
};
```

3. ゾーンファイルを作成する。 IPアドレスとドメイン名の対応を定義する。

```
# /etc/bind/db.corp.local を新規作成
sudo nano /etc/bind/db.corp.local
```

```
    604800
@       IN      SOA     intdns.corp.local. root.corp.local. (
                              2         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
@       IN      NS      intdns.corp.local.
intdns  IN      A       192.168.0.53
intweb  IN      A       192.168.0.80
```

4. 設定を反映させ、動作を確認する。bash # **設定ファイルの文法チェック** sudo named-checkconf # **ゾーンファイルの文法チェック** sudo named-checkzone corp.local /etc/bind/db.corp.local # BIND9**を再起動** sudo systemctl restart bind9

ステップ**5**：**DNS**による名前解決の確認

1. クライアントの**DNS**設定を変更する。ClientAが、構築したIntDNSサーバーに名前解決を問い合わせるように設定する。bash # ClientA**で実行** # /etc/resolv.conf **の** nameserver **を**IntDNS**のアドレスに書き換える** echo "nameserver 192.168.0.53" | sudo tee /etc/resolv.conf
2. 名前解決をテストする。nslookup コマンドで、intweb.corp.local のIPアドレスが正しく引けるか確認する。bash # ClientA**で実行** nslookup intweb.corp.local Server: 192.168.0.53 と Address: 192.168.0.80 が表示されれば成功である。
3. 名前での **ping** を試す。最後に、IPアドレスの代わりに名前で ping が通ることを確認する。bash ping intweb.corp.local

これで第1回の実験は終了である。ここまでの設定と確認結果をレポートにまとめよ。

実験手順書：第2回 NAPTによるインターネット接続

1. 実験の目的

前回の実験で構築した内部ネットワークを模擬的なインターネットに接続する。組織の出入り口となるゲートウェイにNAPT（Network Address Port Translation）を設定し、複数のプライベートIPアドレスを持つPCが一つのグローバルIPアドレスを共有してインターネットと通信する仕組みを学ぶ。今回の目標は以下の通りである。

- NAPT（IPマスカレード）の概念と必要性を理解する。
- `iptables` を用いてLinuxルーターにNAPTを設定できる。
- 内部ネットワークからインターネット上のサーバーへアクセスできることを確認する。
- DNSのフォワーディング機能を設定し、内部ドメインと外部ドメインの両方を名前解決できる仕組みを構築する。
- NAPTによるアドレス変換の様子を観察し、その動作を説明できる。

2. 実験環境

- 前提条件: 第1回の実験で構築したネットワークが、すべての設定を含めて正しく動作していること。
- 追加の構成: ゲートウェイ(GW)を教員が準備したインターネットセグメントに接続する。

IPアドレス設定表（第2回追加分）

機器名	役割	インターフェース	IPアドレス/マスク	ゲートウェイ	DNSサーバー
GW	ゲートウェイ	WAN (eth0)	`10.0.0.1/24`	`10.0.0.254`	`10.0.0.53`
ExtDNS	外部DNS	-	`10.0.0.53/24`	`10.0.0.254`	(自身)
ExtWeb	外部Web	-	`10.0.0.80/24`	`10.0.0.254`	`10.0.0.53`

インターネット側のルーター（*`10.0.0.254`*）は教員が管理する。

3. 実験手順

ステップ**1**：ゲートウェイ**(GW)**のインターネット接続

GWをインターネットに接続し、自身がインターネットと通信できるように設定する。

1. **WAN**側インターフェースの設定 GWのWAN側インターフェース（例: `eth0`）にグローバルIPアドレスを設定する。`bash # GWで実行 sudo ip addr add 10.0.0.1/24 dev eth0`
2. デフォルトゲートウェイの設定 GW自身にとってのインターネットへの出口を設定する。これによりGWは未知の宛先のパケットをすべてインターネット側ルーター(`10.0.0.254`)へ転送する。`bash # GWで実行 sudo ip route add default via 10.0.0.254`
3. 疎通確認 GWからインターネット上のサーバー（ExtWeb: `10.0.0.80`）へ `ping` が通ることを確認する。`bash # GWで実行 ping 10.0.0.80` この時点では、内部のクライアント*PC*から *`ping`* は通らない。戻りのパケットがクライアント*PC*のプライベート*IP*アドレスに届かないためである。

ステップ**2**：**NAPT**（**IP**マスカレード）の設定

内部ネットワークのPCがGWのグローバルIPアドレス（`10.0.0.1`）を借りてインターネットと通信できるように、`iptables` を使ってNAPTを設定する。

1. **`iptables`** ルールの追加 「POSTROUTING」チェーン、すなわちパケットがルーターから出ていく直前に送信元アドレスを書き換える（マスカレードする）ルールを追加する。

```
# GWで実行
# -t nat: natテーブルを操作する
# -A POSTROUTING: POSTROUTINGチェーンに追加する
# -o eth0: 出力インターフェースがeth0（WAN側）の場合
# -j MASQUERADE: 送信元IPアドレスをeth0のIPアドレスに書き換える
sudo iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
```

2. 設定の確認 追加したルールを確認する。bash　　# GWで実行　　sudo iptables -t nat -L POSTROUTING -v
3. 内部クライアントからの疎通確認 ClientAから再度ExtWeb (10.0.0.80) へ ping を実行する。今回は成功するはずである。bash　　# ClientAで実行　　ping 10.0.0.80

ステップ**3**：**DNS**フォワーダーの設定

内部DNSサーバー(IntDNS) が、自身のゾーンに存在しないドメイン（例: www.example.com）に関する問い合わせを外部DNSサーバー(ExtDNS)へ転送するように設定する。

1. **BIND9**のオプション設定を編集する。 bash　　# IntDNSで実行　　sudo nano /etc/bind/named.conf.options
2. **forwarders** を追加する。 options ブロック内に以下の2行を追加または修正する。

```
options {
    directory "/var/cache/bind";

    // ↓↓↓↓ この2行を追加 ↓↓↓↓
    forwarders {
        10.0.0.53; // 外部DNSサーバーのIPアドレス
    };
    forward only;
    // ↑↑↑↑ ここまで ↑↑↑↑

    dnssec-validation auto;
    listen-on-v6 { any; };
};
```

3. **BIND9**を再起動する。 bash　　# IntDNSで実行　　sudo systemctl restart bind9

ステップ**4**：最終的な動作確認

ClientAから、すべてのネットワーク機能が統合的に動作することを確認する。

1. 外部ドメインの名前解決 ClientAから nslookup で外部ドメイン（例: www.example.com、教員から指示されたドメイン名）が引けることを確認する。bash　　# ClientAで実行　　nslookup www.example.com この問い合わせは ClientA -> R1 -> GW -> IntDNS -> GW -> ExtDNS という経路を辿り、応答が返る。
2. 内部ドメインの名前解決 引き続き、内部ドメイン（intweb.corp.local）も引けることを確認する。bash　　# ClientAで実行　　nslookup intweb.corp.local
3. **Web**ブラウザでのアクセス ClientAにGUI環境が整っていれば、Webブラウザを起動し内部Webサーバー (http://intweb.corp.local) と外部Webサーバー (http://www.example.com) の両方にアクセスできることを確認すること。

ステップ**5**：**NAPT**の動作観察

NAPTが実際にどのように動作しているかを確認するため、GWでコネクショントラッキングテーブルを観察する。

1. **conntrack** ツールのインストール bash　　# GWで実行　　sudo apt update sudo apt install conntrack

2. コネクションの監視 ClientAで外部のWebサイトに `ping` を実行し続けた状態で、GWで以下のコマンドを実行する。

```
# ClientAで実行
ping www.example.com

# GWで実行
sudo conntrack -L
```

出力からICMP通信を探し、`src=192.168.1.10`（ClientA）のパケットが `[UNREPLIED]` または `[ASSURED]` のセクションで `src=10.0.0.1`（GWのWAN側IP）へ書き換えられている様子を確認する。これがNAPTによるアドレス変換の証拠である。

4. 課題

この実験結果を基に、レポートの考察の章で以下の問いに答えよ。

- 問: NAPTの役割とは何か。内部ネットワークのPC（`192.168.1.10`）が一つのグローバルIPアドレス（`10.0.0.1`）を使ってインターネット上のサーバーと通信できる仕組みを、`conntrack` の結果などを交えて具体的に説明せよ。
- 問: DNSフォワーダーを設定する理由を説明せよ。設定しなかった場合、ClientAから `www.example.com` への名前解決はどうなるか、そのプロセスを追って考察せよ。

レポート課題：実践的ネットワークシステムの構築

1. 課題の概要

2回にわたる「実践的ネットワークシステムの構築」実験で得られた結果と考察をレポートとしてまとめて提出せよ。このレポートは、実験を通して構築した多層的なネットワークに関する理解度を評価するためのものである。静的ルーティング、NAPT、DNSといった現代のネットワークを支える重要な技術要素について、自ら構築・確認した事実に基づいて論理的に説明する能力が求められる。

2. レポートの構成

以下の構成に従ってレポートを作成すること。

1. 表紙
   - レポートタイトル、学科、学年、学生番号、氏名、提出日を明記すること。
2. はじめに（目的）
   - 本実験全体の目的（多層ネットワークの構築、ルーティング、NAPT、DNSの連携など）を、自分の言葉で簡潔にまとめること。
3. 実験環境
   - 実験で構築した最終的なネットワーク構成図を提示すること（実験手順書に示した図を参考に、自分で作成すること）。
   - 実験で使用したIPアドレッシング計画の表を記載すること。
4. 実験手順と結果
   - 各実験（第1回・第2回）の主要なステップについて、実行した内容と得られた結果を記述すること。
   - 特に、ルーティングテーブルの設定（`ip route`）、`traceroute` の実行結果、`iptables` のルール、`conntrack` の観察結果、`nslookup` の結果など、客観的な事実を正確に記録することを重視する。
   - 結果を示す際には、ターミナルの出力やGUIのスクリーンショットを適切に活用し、図や表を用いて分かりやすく整理すること。
5. 考察
   - 実験結果に基づき、以下の問いに答えること。単に用語を調べるだけでなく、実験のどの結果からそう言えるのかを明確に示しながら論理的に考察を展開すること。

考察の問い

- 問**1:** ClientA (`192.168.1.10`) から ClientB (`192.168.2.10`) へ `ping` を実行した際、パケットはどのような経路を辿るか。`traceroute` の結果を示し、その各ホップ（R1, GW, R2）で、それぞれの機器のルーティングテーブルがどのように参照され、パケットが次のホップへ転送されるのかを具体的に説明せよ。
- 問**2:** ゲートウェイ(GW)に、内部ネットワーク（`192.168.1.0/24` および `192.168.2.0/24`）への静的ルートを設定した。もしこの設定がなかった場合、ClientA から外部Webサーバー (`ExtWeb`) への `ping` はどうなるか。「行き」と「戻り」のパケットの経路をそれぞれ考察し、通信が成立しない理由を説明せよ。
- 問**3:** NAPT（IPマスカレード）の役割とは何か。内部ネットワークにいる複数のPC（ClientA, ClientBなど）が、ゲートウェイの持つ単一のグローバルIPアドレス (`10.0.0.1`) を共有してインターネット上のサーバーと通信できる仕組みを、`iptables` の設定ルールや `conntrack` の観察結果を根拠として説明せよ。
- 問**4:** 内部DNSサーバー (`IntDNS`) と外部DNSサーバー (`ExtDNS`) の役割の違いは何か。ClientAが `intweb.corp.local`（内部ドメイン）と `www.example.com`（外部ドメイン）の両方の名前解決ができるのはなぜか。それぞれのドメインに対する `nslookup` の問い合わせが、どのような経路で処理されるのかをステップごとに説明せよ。

---

6. 結論（まとめ）
   - 実験全体を通して学んだこと、特に難しかった点やそれをどのように解決したか、さらに今後の学習で探求したいと感じた点などを自由に記述すること。
7. 参考文献（任意）
   - レポート作成にあたり、参考にした書籍やWebサイトがあれば記載すること。

3. 評価基準

提出されたレポートは、以下のルーブリックに基づいて評価する。

評価項目	A (優)	B (良)	C (可)	D (不可)
結果の記録の正確性と明瞭性	コマンド出力や`traceroute`の結果などが要点を押さえて整理され、誰が読んでも分かりやすく正確に記録されている。	結果は記録されているが、情報の整理が不十分であったり、説明が不足している点がある。	結果の記録に不足や誤りが多く、第三者が内容を再現することが困難である。	結果がほとんど記録されていない、または不正確である。
考察の論理性と深さ	各問いに対し、実験結果とネットワークの動作原理を関連付けた論理的で深い考察がなされている。	各問いに対し、実験結果に基づいた考察がなされているが、分析が表面的であったり、論理の飛躍が見られる。	考察が単なる用語説明に終始していたり、実験結果との関連性が薄い。	考察がほとんどなされていない、または見当違いである。
ネットワーク技術の統合的理解	ルーティング、NAPT、DNSといった複数の技術が連携して一つのシステムとして動作している様子を統合的に理解し、説明できている。	各技術の個別の役割は理解しているが、それらの連携についての説明が不十分である。	技術の理解に根本的な誤りが多く見られる。	ほとんど理解できていない。

評価項目	A (優)	B (良)	C (可)	D (不可)
問題解決能力	実験中に発生した問題に対し、原因を自ら切り分け、解決に至るプロセスがレポートから読み取れる。	問題の発生と結果は記述されているが、解決へのアプローチが不明確である。	問題解決の試みが見られない。	実験を完遂できていない。

4. 提出について

- 提出期限: 第2回実験の翌週の授業開始時
- 提出形式: A4用紙に印刷して提出、または指示された電子的手段で提出すること。

補助資料1：Linuxネットワークコマンドチートシート

この資料は、実験で使用する主要なLinuxネットワークコマンドの基本的な使い方をまとめたものである。より複雑なネットワークに対応するため、`iptables` や `bind9` 関連のコマンドを追加している。

1. `ip` コマンド (基本)

ネットワークインターフェースやルーティングテーブルを管理する基本コマンドである。

- インターフェースとIPアドレスの表示 `bash ip address show # 短縮形: ip a`
- IPアドレスの追加 (例: `192.168.1.10/24` を `eth0` に) `bash sudo ip addr add 192.168.1.10/24 dev eth0`
- ルーティングテーブルの表示 `bash ip route show # 短縮形: ip r`
- デフォルトゲートウェイの追加 (例: `192.168.1.1`) `bash sudo ip route add default via 192.168.1.1`
- 静的ルートの追加 (例: `192.168.1.0/24` へのパケットを `192.168.0.11` へ転送) `bash sudo ip route add 192.168.1.0/24 via 192.168.0.11`

2. `ping` & `traceroute` (疎通確認)

- 指定ホストとの疎通確認 `bash ping 192.168.1.10`
- 指定ホストまでの経路追跡 `bash traceroute 192.168.1.10`

3. `iptables` (ファイアウォール & NAPT)

Linuxカーネルのパケットフィルタリング機能（Netfilter）を操作するコマンドであり、NAPTの設定に用いる。

- 現在のNATテーブルのルールを表示 `bash sudo iptables -t nat -L -v`
- NAPT（IPマスカレード）ルールの追加 (WAN側インターフェースが `eth0` の場合、`eth0` から出ていくパケットの送信元を `eth0` のIPアドレスに書き換える) `bash sudo iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE`
- NATテーブルのルールをすべて削除 `bash sudo iptables -t nat -F`

4. `conntrack` (コネクショントラッキング)

NAPTによって変換されている通信セッションの状態を確認できる。

- `conntrack` ツールのインストール `bash sudo apt update sudo apt install conntrack`

- 現在のコネクショントラックテーブルを表示 `bash sudo conntrack -L`
- 特定の送信元**IP**アドレスでフィルタリング (例: ClientA `192.168.1.10`) `bash sudo conntrack -L -s 192.168.1.10`

5. BIND9 (DNSサーバー)

DNSサーバー `BIND9` の管理用コマンドをまとめる。

- **BIND9**のインストール `bash sudo apt update sudo apt install bind9`
- 設定ファイルの構文チェック `bash sudo named-checkconf`
- ゾーンファイルの構文チェック (例: `corp.local` ゾーン) `bash sudo named-checkzone corp.local /etc/bind/db.corp.local`
- **BIND9**サービスの再起動 `bash sudo systemctl restart bind9`
- **BIND9**サービスのステータス確認 `bash sudo systemctl status bind9`

6. `nslookup` & `dig` (DNSクライアント)

DNSサーバーに問い合わせて名前解決を行うコマンドである。

- **`nslookup`** でドメイン名を問い合わせる `bash nslookup intweb.corp.local`
- **`dig`** で詳細な情報を問い合わせる `bash dig www.example.com`

7. その他

- **IP**フォワーディングの有効化 (ルーター機能の有効化) `bash sudo sysctl -w net.ipv4.ip_forward=1`
- **DNS**サーバー設定ファイルの確認・編集 (クライアントPCが参照するDNSサーバーのアドレスを管理する)

```bash
# 確認
cat /etc/resolv.conf

# 編集 (例: 192.168.0.53 を指定)
echo "nameserver 192.168.0.53" | sudo tee /etc/resolv.conf
```