

# VM웨어 2주차 6일

ACL이란?



## ACL(Access Control List)이란?

- **\*Access Control List (ACL)\***는 네트워크 보안의 중요한 개념으로, 네트워크를 통해 전달되는 데이터를 필터링하고 접근을 제어하는 데 사용됩니다. ACL을 통해 어떤 데이터가 네트워크를 통과할 수 있는지, 또는 차단해야 하는지를 결정할 수 있습니다. 쉽게 말해, ACL은 네트워크의 "보안 경비원" 역할을 합니다.

## 일상적인 비유로 이해하기

비유를 통해 설명하자면, ACL은 건물의 출입문에 설치된 보안 경비 시스템과 비슷합니다. 이 건물에는 특정 사람만 들어갈 수 있도록 하기 위해 경비원이 출입 명단을 가지고 있다고 상상해 보세요.

1. **출입 명단:** ACL은 이 경비원이 가지고 있는 명단과 같습니다. 이 명단에는 어떤 사람들이 들어올 수 있는지(허용), 어떤 사람들이 들어오면 안 되는지(거부)에 대한 정보가 적혀 있습니다.
2. **경비원의 역할:** 네트워크에서는 ACL이 경비원 역할을 합니다. ACL은 네트워크 트래픽을 필터링하여 특정 IP 주소나 특정 유형의 데이터를 허용하거나 거부합니다. 예를 들어, 회사 네트워크에서는 특정 외부 IP 주소로부터의 접근을 차단할 수 있습니다.
3. **정해진 규칙:** ACL은 명시적인 규칙으로 구성됩니다. 규칙은 특정 IP 주소나 네트워크 포트 번호를 기반으로 정의될 수 있으며, 이 규칙은 데이터가 네트워크를 통과할 때 ACL에 의해 평가됩니다. 예를 들어, "모든 직원은 근무 시간 동안에만 이 서버에 접근할 수 있다"와 같은 규칙을 설정할 수 있습니다.

## ACL의 주요 기능

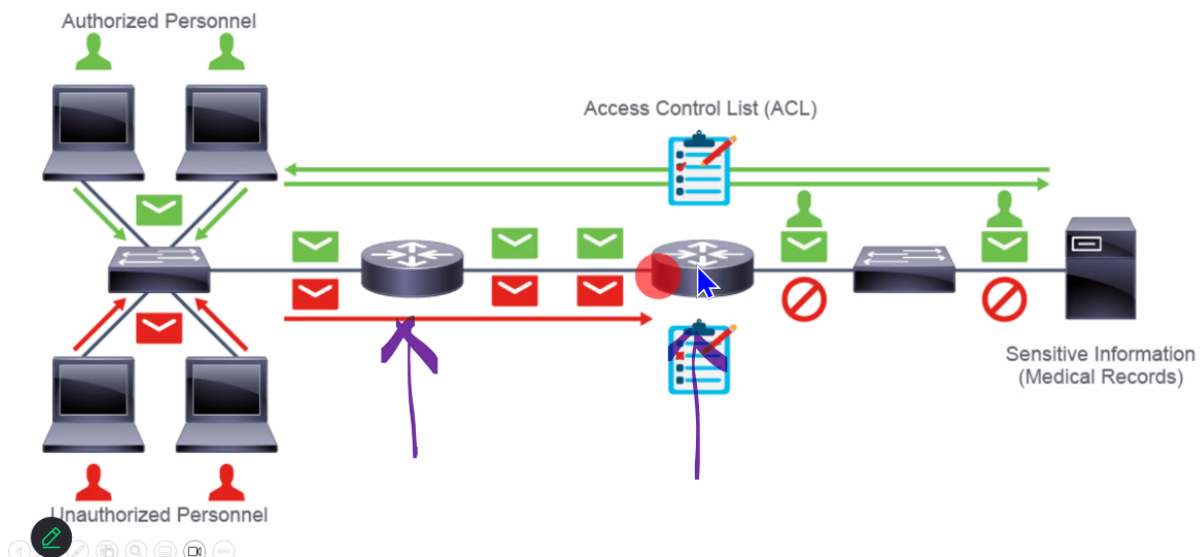
1. **보안 강화:** ACL은 네트워크에 접근할 수 있는 사람이나 시스템을 제한하여 네트워크 보안을 강화합니다. 불법 접근을 차단하고 민감한 정보를 보호합니다.
2. **트래픽 관리:** ACL을 사용하면 네트워크 트래픽을 관리할 수 있습니다. 예를 들어, 특정 종류의 트래픽(예: 비디오 스트리밍)을 제한하여 네트워크 대역폭을 관리할 수 있습니다.
3. **사용자 정의 필터링:** ACL은 매우 구체적인 필터링을 가능하게 합니다. IP 주소, 프로토콜 유형, 포트 번호 등을 기반으로 트래픽을 허용하거나 차단할 수 있습니다.

## 기술적인 구성 요소

1. **번호형 ACL:** ACL 규칙에 번호를 부여하여 순서를 정하는 방식입니다. 예를 들어, 1번 규칙이 2번 규칙보다 먼저 평가됩니다.
2. **이름형 ACL:** 번호 대신 이름을 사용하여 ACL을 정의하는 방식입니다. 더 직관적이고 관리하기 쉬운 방법입니다.
3. **표준 ACL:** 소스 IP 주소만을 기준으로 트래픽을 필터링합니다. 간단한 필터링 작업에 사용됩니다.
4. **확장 ACL:** 소스와 목적지 IP 주소, 프로토콜 유형, 포트 번호 등을 기준으로 더 정밀하게 트래픽을 필터링합니다. 더 복잡한 보안 요구 사항을 충족하는데 적합합니다.

이해를 돕기 위해 이러한 개념들에 대해 질문이 있으시면 말씀해 주세요! ACL의 구성과 사용 방법에 대한 추가적인 기술적 세부 사항을 더 깊이 다루어 드릴 수 있습니다.

## Explaining the Basics of ACL (Cont.)

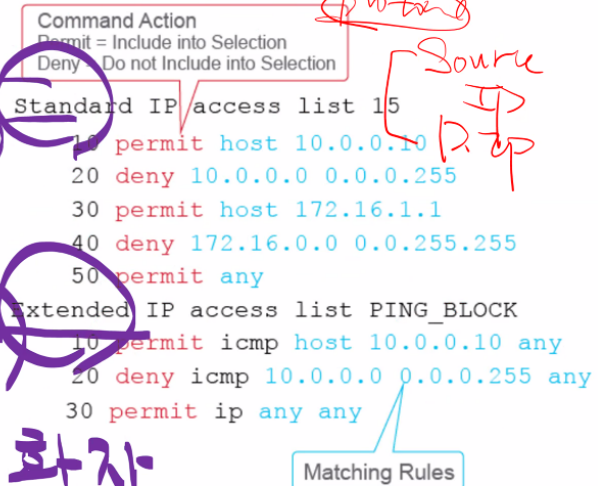


- 2번째 라우터 기준으로 왼쪽에서 요청하는 친구들은 S 소스다.
  - 1,2 소스는 허용하고, 3,4 소스는 불허하도록 2번째 라우터에 ACL을 설정한다.
- 1,2은 허용하고, 3,4는 불허하는 방법으로 IP를 사용한다. (표준ACL) 사용

## ACL Overview

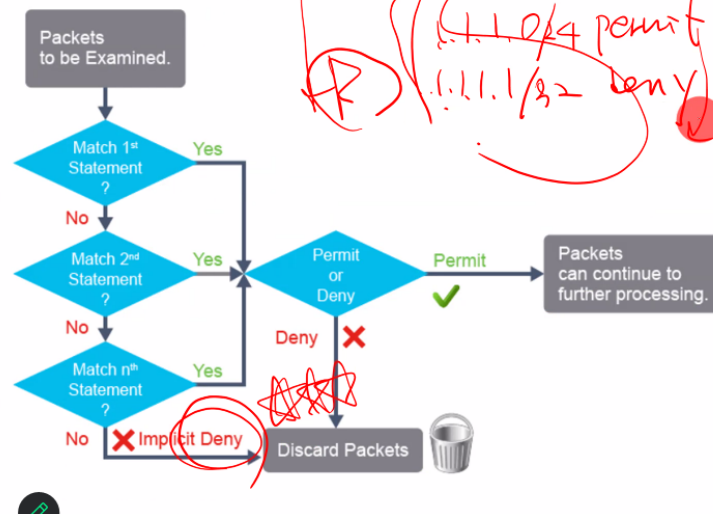
The following ACL features are important to remember:

- Ordered series of statements
- ACL statements specify the following:
  - Action
    - Permit
    - Deny
  - Matching rule
- ACLs are applied for the following:
  - Traffic filtering
  - Selecting traffic
    - To be analyzed, forwarded, or processed in other ways



- 확장ACL은 IP, 포트, 프로토콜 등을 구분해서 더욱 정밀하게 트래픽을 제어한다.

## ACL Operation



ACL tests include the following:

- An ACL consists of a series of permit and deny statements.
- An ACL is consulted in top-down order.
- The first match executes the **permit** or **deny** action and stops further ACL matching.
- There is an implicit **deny all** statement at the end of each ACL.

- 좁은 범위의 IP를 위에 올리고 넓은 범위의 IP를 밑으로 내려라.
- 순서에 따라서 1번 디나이하고 이후에는 허용해버린다.



## 1. Top-Down Order로 동작한다는 것

ACL이 "top-down order"로 동작한다는 것은, ACL에 정의된 규칙들이 위에서 아래로 순차적으로 적용된다는 것을 의미합니다. 이 방식을 이해하기 위해서는 ACL이 어떻게 구성되고 작동하는지 알 필요가 있습니다.

- **위에서 아래로 순서대로 검사:** ACL은 여러 개의 규칙으로 구성되며, 이 규칙들은 리스트 형태로 나열됩니다. 패킷이 ACL에 들어오면, 가장 위에 있는 첫 번째 규칙부터 순차적으로 하나씩 검사합니다. 만약 어떤 규칙과 일치하면, 그 규칙에 정의된 액션(허용 또는 거부)을 즉시 수행하고, 더 이상의 규칙은 검사하지 않습니다.

- 예를 들어, ACL이 다음과 같이 구성되어 있다고 가정해 봅시다:

1. `permit 192.168.1.0/24` (192.168.1.0/24 네트워크를 허용)
2. `deny 192.168.1.5` (192.168.1.5를 거부)

- 이 경우, 만약 패킷의 소스 IP가 192.168.1.5라면, 첫 번째 규칙인 `permit 192.168.1.0/24`가 먼저 일치하여 허용됩니다. 그러므로 그 패킷은 ACL의 나머지 규칙(즉, 192.168.1.5를 거부하는 규칙)을 더 이상 검사하지 않고 네트워크를 통과하게 됩니다.

- **첫 번째 일치가 중요:** 따라서 ACL의 규칙을 정의할 때, 가장 구체적인 규칙을 먼저 배치하고, 덜 구체적인 규칙을 그 다음에 배치하는 것이 중요합니다. 그렇지 않으면 구체적인 규칙이 적용되기 전에 일반적인 규칙이 먼저 적용되어 의도한 대로 작동하지 않을 수 있습니다.

## 2. Implicit Deny All

**Implicit Deny All**(묵시적 거부)은 ACL의 모든 규칙을 통과하지 못한 패킷을 자동으로 차단하는 기본적인 보안 메커니즘입니다. 이 개념을 이해하려면 ACL의 기본 동작 방식을 알아야 합니다.

- **명시적 규칙이 없는 경우:** ACL 리스트의 끝에는 명시적으로 설정되지 않은 거부 규칙이 자동으로 존재합니다. 이를 "묵시적 거부"라고 합니다. 즉, ACL에 명시적으로 허용(permit) 또는 거부(deny)되지 않은 패킷은 기본적으로 차단됩니다.

- 예를 들어, ACL이 다음과 같이 정의되어 있다고 가정합시다:

1. `permit 192.168.1.0/24` (192.168.1.0/24 네트워크를 허용)

- 만약 패킷의 소스 IP가 192.168.2.1이라면, 이 패킷은 명시된 규칙과 일치하지 않으므로 ACL의 끝에 있는 묵시적 거부 규칙에 의해 자동으로 차단됩니다.

- **보안상의 이유:** 묵시적 거부는 네트워크 보안에서 중요한 역할을 합니다. ACL에 명시적으로 허용되지 않은 모든 트래픽을 자동으로 거부함으로써, 잠재적인 보안 위협으로부터 네트워크를 보호할 수 있습니다. 이는 네트워크 관리자가 예상하지 못한 트래픽이나 비정상적인 접근을 차단하는 기본적인 방어선을 제공하기 때문입니다.

## 요약

- **Top-Down Order:** ACL은 위에서 아래로 순차적으로 규칙을 검사합니다. 가장 먼저 일치하는 규칙이 발견되면 그 규칙에 따라 행동을 결정하고 더 이상의 규칙은 검사하지 않습니다. 따라서 규칙을 설정할 때 그 순서가 매우 중요합니다.
- **Implicit Deny All:** ACL 리스트의 끝에는 명시적으로 정의되지 않은 "거부" 규칙이 항상 존재합니다. 이것은 ACL에 명시적으로 허용되지 않은 모든 트래픽을 자동으로 차단하는 기본 보안 메커니즘입니다.

이 두 가지 개념이 ACL의 효과적인 동작을 이해하고 설정하는 데 중요한 이유입니다. 추가적인 질문이 있거나 더 깊이 알고 싶은 부분이 있다면 알려주세요!



네, 맞습니다. ACL을 설계할 때 좁은 범위의 IP를 위에 올리고 넓은 범위의 IP를 아래로 내리는 것이 좋은 설계 원칙입니다. 이 원칙은 ACL의 **Top-Down Order** 방식 때문에 중요합니다. 이제 이 원칙이 왜 중요한지 자세히 설명해 드리겠습니다.

## 좁은 범위를 위에, 넓은 범위를 아래에 배치하는 이유

1. **세밀한 제어를 우선 적용:** 좁은 범위의 IP 규칙은 특정한 IP 주소나 서브넷에 대해 매우 세밀한 제어를 제공합니다. 이러한 규칙은 매우 특정한 행동을 요구할 때 필요합니다. 예를 들어, 특정 서버의 접근을 제어하거나 중요한 네트워크 장치에 대한 접근을 제한할 때가 그렇습니다. 이런 경우, 좁은 범위의 규칙을 먼저 배치하면 ACL이 더 구체적이고 중요한 규칙을 먼저 평가하여 적절한 조치를 취할 수 있습니다.

- 예: `deny 192.168.1.5` (특정 IP 거부)
- `permit 192.168.1.0/24` (서브넷 허용)

위의 예에서 특정 IP `192.168.1.5`를 먼저 거부하고, 나머지 서브넷 `192.168.1.0/24`를 허용합니다. 이렇게 하면 세부적인 규칙이 먼저 적용되어 의도한 대로 동작하게 됩니다.

2. **일반 규칙을 나중에 적용:** 넓은 범위의 IP 규칙은 보다 일반적인 제어를 제공합니다. 넓은 범위를 허용하거나 거부하는 규칙을 위에 배치하면, 좁은 범위의 규칙이 평가되기 전에 일반 규칙이 먼저 적용될 수 있습니다. 이 경우, 특정한 제어가 무시될 수 있습니다.

- 예: 만약 `permit 192.168.1.0/24`를 먼저 두고, `deny 192.168.1.5`를 그 다음에 두면, `192.168.1.5`는 서브넷 규칙에 의해 이미 허용되었기 때문에 거부 규칙이 무시됩니다. 따라서 이 경우 특정 IP에 대한 제어가 실패하게 됩니다.

3. **성능 최적화:** 좁은 범위의 규칙이 먼저 적용되면, ACL이 더 적은 수의 규칙만 평가하고도 패킷의 처리를 결정할 수 있어 성능이 최적화됩니다. 패킷이 구체적인 규칙에 먼저 일치하면 나머지 넓은 범위의 규칙을 평가할 필요가 없어 ACL 처리 속도가 빨라질 수 있습니다.

## 요약

- **구체적인 규칙이 일반적인 규칙보다 먼저:** 좁은 범위의 규칙(예: 특정 IP 주소)은 일반적인 규칙(예: 넓은 서브넷 범위)보다 먼저 배치하여 세밀한 제어가 무시되지 않도록 해야 합니다.

- **효율적인 트래픽 필터링:** 좁은 범위의 규칙이 먼저 평가됨으로써 불필요한 규칙 평가를 줄이고, 네트워크 트래픽에 대한 정확한 제어를 제공할 수 있습니다.

이 원칙을 따르면 ACL을 통해 네트워크 접근을 보다 효과적이고 안전하게 관리할 수 있습니다. 추가적으로 더 알고 싶거나 질문이 있으시면 언제든지 말씀해 주세요!

## ACL Wildcard Masking

Matching criteria/matching rule has two elements:

- IPv4 address provides a reference against which IPv4 packet information is evaluated.
- Wildcard mask provides evaluation criteria:
  - 0 = this bit *must match* the value in the reference IPv4 address
  - 1 = this bit can have *whatever* value



- 와일드카드 마스크를 ACL에서 사용하는 이유는 홀수짜수 네트워크 설정을 위해서
- 10.1.1.1/30 (서브넷) → 와일드카드 마스크 (0.0.0.3)
- 10.1.1.1/28 (서브넷) → 0.0.0.15
- 0은 네트워크 대역이고 1의 숫자에 해당하는 것만 허용

## 와일드 마스크 사용법





## 1. 목표

`10.1.1.1/24` 대역에서 홀수 IP만 허용하려고 합니다. 이 대역은 `10.1.1.0` 부터 `10.1.1.255` 까지의 IP 주소를 포함합니다. 여기서 우리는 마지막 옥텟(즉, `10.1.1.x` 에서 `x`)이 홀수인 IP 주소만 허용하려고 합니다.

## 2. 와일드카드 마스크 설정

홀수 IP 주소는 마지막 비트가 `1` 로 끝나는 주소를 의미합니다. 따라서:

- **기준 IP 주소:** `10.1.1.1` (예: 마지막 비트가 `1` 인 주소를 사용)
- **와일드카드 마스크:** `0.0.0.254`

## 3. ACL 규칙 예시

```
permit 10.1.1.1 0.0.0.254
```

이 규칙을 사용하면 다음과 같은 동작을 하게 됩니다:

1. **기준 IP 주소** `10.1.1.1`: 이는 비교를 위한 기본 주소입니다.
2. **와일드카드 마스크** `0.0.0.254`: 이 마스크는 마지막 옥텟의 마지막 비트만 검사하도록 설정합니다. 나머지 비트는 무시합니다.

## 4. 예시의 동작

이 ACL 규칙은 `10.1.1.x` 의 모든 IP 중 마지막 옥텟이 홀수인 IP만 허용합니다. 예를 들어:

- `10.1.1.1`, `10.1.1.3`, `10.1.1.5`, ..., `10.1.1.253` 와 같은 홀수 IP는 허용됩니다.
- `10.1.1.2`, `10.1.1.4`, `10.1.1.6`, ..., `10.1.1.254` 와 같은 짝수 IP는 허용되지 않습니다.

## 5. 작동 방식

- \*와일드카드 마스크 `0.0.0.254` \*는 마지막 옥텟의 비트 중 마지막 비트만 검사하고 나머지 비트를 무시합니다.
  - `0` 비트는 참조 주소와 일치해야 함을 의미합니다.
  - `1` 비트는 어떤 값이든 허용한다는 의미로, 여기서는 `254` 를 사용하여 마지막 비트만 중요하다는 것을 의미합니다.

## 요약

- `10.1.1.1 0.0.0.254` 규칙을 사용하여 `10.1.1.0/24` 대역의 홀수 IP 주소만 허용할 수 있습니다.
- 와일드카드 마스크 `0.0.0.254` 는 마지막 비트만 검사하고, 이 비트가 `1` 인 IP 주소만 허용합니다.

이 ACL 설정 방법이 이해되었는지 궁금합니다. 추가적인 질문이 있으시거나 더 자세히 알고 싶으신 내용이 있다면 언제든지 말씀해 주세요!



좋은 질문입니다! `10.1.1.1/28` 네트워크를 와일드카드 마스크로 표현하는 방법에 대해 설명드리겠습니다.

## 1. 네트워크와 서브넷 마스크 이해하기

- **네트워크:** `10.1.1.1/28` 은 실제로 `10.1.1.0/28` 네트워크를 의미하며, 이는 `10.1.1.0` 부터 `10.1.1.15` 까지의 IP 주소 범위를 포함합니다.
- **서브넷 마스크:** `/28` 은 서브넷 마스크가 `255.255.255.240` 임을 의미합니다. 이 서브넷 마스크는 네트워크 범위를 결정하는 데 사용됩니다.

## 2. 와일드카드 마스크 계산

와일드카드 마스크는 서브넷 마스크의 반대로(보수) 계산합니다. 와일드카드 마스크는 `1` 이 서브넷 마스크에서 `0` 으로, `0` 이 서브넷 마스크에서 `1` 로 변환됩니다.

### 서브넷 마스크 `255.255.255.240` 의 보수 계산:

- **서브넷 마스크:** `255.255.255.240`
- **이진수 표현:** `11111111.11111111.11111111.11110000`
- **와일드카드 마스크:** `00000000.00000000.00000000.00001111`

이진수 표현에서 각 옥텟을 10진수로 변환하면:

- **와일드카드 마스크:** `0.0.0.15`

## 3. 와일드카드 마스크 사용

`10.1.1.0/28` 네트워크를 ACL 규칙에서 와일드카드 마스크로 표현하려면 다음과 같이 할 수 있습니다:

```
10.1.1.0 0.0.0.15
```

이 와일드카드 마스크는 `10.1.1.0` 부터 `10.1.1.15` 까지의 IP 주소 범위를 지정합니다. 즉, ACL에서 이 범위에 있는 모든 IP 주소를 대상으로 규칙을 적용할 수 있습니다.

## 4. 예시: 특정 규칙 적용

- 예를 들어, `10.1.1.0/28` 범위의 IP 주소를 허용하는 ACL 규칙을 설정하려면:

```
permit 10.1.1.0 0.0.0.15
```

- 이 규칙은 10.1.1.0 부터 10.1.1.15 까지의 모든 IP 주소를 허용합니다.

## 요약

- 10.1.1.1/28 네트워크에 해당하는 와일드카드 마스크는 0.0.0.15 입니다.
- 이를 사용하여 ACL 규칙을 만들 때, 네트워크 주소와 와일드카드 마스크를 결합하여 특정 IP 범위를 지정할 수 있습니다.

이 설명이 도움이 되었나요? 추가적인 질문이 있거나 더 알고 싶은 부분이 있으시면 말씀해 주세요!

## ACL 번호부여 vs 이름지정



이미지에서는 **\*\*Standard IPv4 ACLs(Access Control Lists)\*\***의 설정 방법과 구성을 설명하고 있습니다. 주요 내용을 요약해 보겠습니다.

## 1. Standard IPv4 ACLs의 구조

Standard IPv4 ACL은 기본적으로 네트워크에서 특정 IP 주소를 기반으로 트래픽을 허용(permit)하거나 거부(deny)하는 규칙을 설정하는 데 사용됩니다. 이 ACL은 주로 소스 IP 주소에 기반하여 트래픽을 필터링합니다.

### 기본 명령 구조

```
access-list <access-list-number> <permit | deny> <source [source-wildcard] | host {address | name} | any>
```

- **access-list-number** : ACL의 번호를 나타냅니다. 일반적으로 1에서 99 사이의 숫자가 사용됩니다.
- **permit | deny** : 트래픽을 허용할지 거부할지 결정하는 액션입니다.
- **source** : 소스 IP 주소를 지정합니다. 와일드카드 마스크를 사용하여 특정 범위의 IP 주소를 지정할 수 있습니다.
- **host** : 특정 호스트 IP 주소를 지정할 때 사용됩니다. 예를 들어 **host 192.168.1.1** 은 특정 IP 주소 **192.168.1.1** 을 가리킵니다.
- **any** : 모든 IP 주소를 의미합니다.

## 2. Matching Criteria for Source IPv4 Address

- **Option 1: Reference IPv4 Address and a Wildcard Mask**: 특정 IP 주소와 와일드카드 마스크를 사용하여 매칭 기준을 설정합니다.
- **Option 2: Keyword **host** and a Reference IPv4 Address**: **host** 키워드를 사용하여 특정 단일 IP 주소를 지정합니다.
- **Option 3: Keyword **any**** : 모든 IP 주소에 대해 매칭합니다.

## 3. Example Configurations

이미지에는 두 가지 설정 방법이 설명되어 있습니다: **Numbered Configuration Method**와 **Named Configuration Method**.

### Numbered Configuration Method

- 이 방식은 ACL을 번호로 식별합니다.
- 예시:

```
RouterX(config)# access-list 1 deny host 172.16.3.3
RouterX(config)# access-list 1 permit 172.16.0.0 0.0.255.255
```

- `access-list 1 deny host 172.16.3.3` : ACL 번호 1을 사용하여 `172.16.3.3` IP 주소를 거부합니다.
- `access-list 1 permit 172.16.0.0 0.0.255.255` : ACL 번호 1을 사용하여 `172.16.0.0` 에서 `172.16.255.255` 범위의 IP 주소를 허용합니다.

## Named Configuration Method

- 이 방식은 ACL에 이름을 붙여 식별합니다.
- 예시:

```
RouterX(config)# ip access-list standard acl2
RouterX(config-std-nacl)# deny host 172.16.3.3
RouterX(config-std-nacl)# permit 172.16.0.0 0.0.255.255
```

- `ip access-list standard acl2` : `acl2` 라는 이름으로 ACL을 정의합니다.
- `deny host 172.16.3.3` : 특정 호스트 `172.16.3.3` 을 거부합니다.
- `permit 172.16.0.0 0.0.255.255` : `172.16.0.0` 부터 `172.16.255.255` 까지의 범위를 허용합니다.

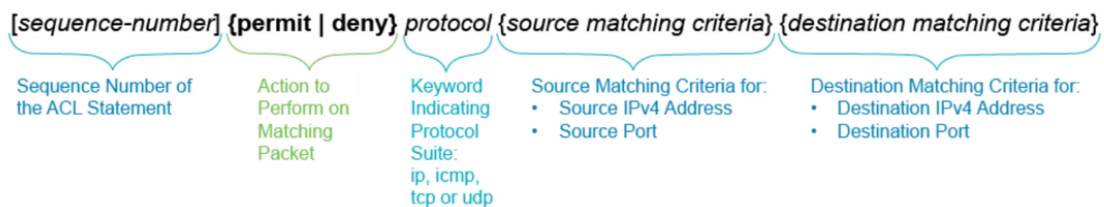
## 요약

- **Standard IPv4 ACL**은 주로 소스 IP 주소 기반의 트래픽 필터링을 위해 사용되며, 특정 IP 주소나 IP 범위를 허용하거나 거부할 수 있습니다.
- **Numbered 방식**과 **Named 방식** 두 가지로 구성할 수 있으며, 각각 ACL 번호나 이름으로 식별합니다.
- 특정 IP 주소를 지정하거나 와일드카드 마스크를 사용해 IP 범위를 지정할 수 있습니다.

이 요약이 도움이 되었길 바랍니다. 추가적인 질문이 있거나 더 알고 싶은 사항이 있다면 언제든지 말씀해 주세요!

## Extended ACL 문법

### Configuring Extended IPv4 ACLs



Example of an Extended ACL Statement:

- Allows TCP Connections from Ports 56000 to 60000 on the Host 172.16.3.3 to Port 80 on Host 203.0.113.30:

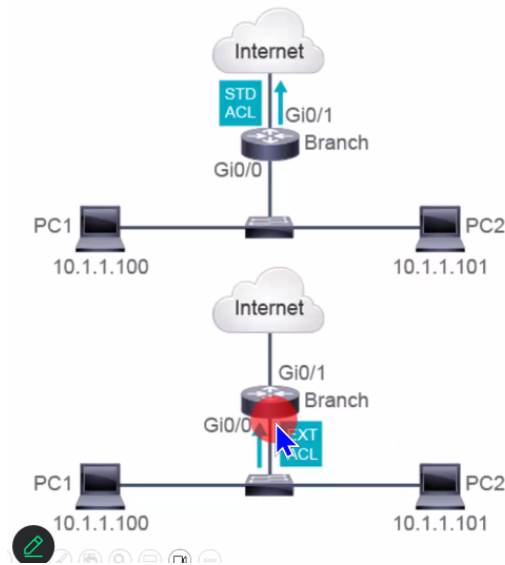
Router(config)# **ip access-list extended 101** *NO-Permit 3*  
 Router(config-ext-nacl)# **permit tcp host 172.16.3.3 range 56000 60000 host 203.0.113.30 eq 80**

Labels for the configuration line:

- permit**: Action
- tcp**: Protocol
- host 172.16.3.3 range 56000 60000**: Source Matching Criteria (Source IPv4 Address, Source Port)
- host 203.0.113.30 eq 80**: Destination Matching Criteria (Destination IPv4 Address and Port)

- 시퀀스 넘버는 옵션 디폴트로 알아서 넘버부여
- [허용, 비허용] / 프로토콜 / 소스IP / 목적지 IP, PORT

## Applying IPv4 ACLs to Filter Network Traffic (Cont.)



Security Policy  
• Deny Internet Access for PC2

```
BRANCH# show access-lists
Standard IP access list 15
    10 deny 10.1.1.101
    20 permit 10.1.1.0 0.0.0.255
BRANCH(config)# interface GigabitEthernet 0/1
BRANCH(config-if)# ip access-group 15 out
```

```
BRANCH# show access-lists
Extended IP access list NOINTERNET_PC2
    10 deny ip host 10.1.1.101 any
    20 permit ip 10.1.1.0 0.0.0.255 any
BRANCH(config)# interface GigabitEthernet 0/0
BRANCH(config-if)# ip access-group
NOINTERNET_PC2 in
```

### 실습

- acl 설정
  - 설정 라우터 선택 : R2
  - 설정 인터페이스 선택 : f0/1
  - 방향성 결정 : inbound 선택

### 인바운드로 ACL 걸기

```
** R2
en
conf t
ip access-list exte NO_ICMP
    deny icmp host 1.1.1.1 host 2.2.2.1
    permit ip any any
!
int f0/1
ip access-group NO_ICMP in
```



## 아웃바운드로 ACL 걸기

```

** R2
en
conf t
ip access-list exte NO_ICMP
    deny icmp host 2.2.2.1 host 1.1.1.1
    permit tcp host 2.2.2.1 eq 80 host 1.1.1.1
!
int f0/1
ip access-group NO_ICMP out

```

---

## 라우팅 프로토콜

- 다이나믹 라우팅 프로토콜 종류



이해를 돕기 위해 IGP(Interior Gateway Protocol)와 EGP(Exterior Gateway Protocol)에 대해 설명드리겠습니다. 이 두 프로토콜은 인터넷을 통해 데이터가 어떻게 이동하는지를 이해하는 데 중요한 역할을 합니다. 각 프로토콜을 쉽게 이해할 수 있도록 일상적인 비유를 사용하겠습니다.

## IGP (Interior Gateway Protocol)란?

### 1. 기본 개념:

- IGP는 네트워크 내부에서 데이터가 어떻게 이동하는지를 결정하는 프로토콜입니다. 이를 이해하기 위해, 하나의 큰 회사 안에 여러 부서가 있다고 생각해 봅시다. 이 회사의 각 부서에서는 회사 내부의 업무 흐름을 효율적으로 관리하기 위한 규칙이 필요합니다. IGP는 이러한 규칙에 해당하며, 회사 내부의 네트워크(하나의 자율 시스템, AS) 안에서 가장 효율적인 경로를 찾는 역할을 합니다.

### 2. 어디에서 사용되나요?:

- IGP는 주로 하나의 조직이나 기업 내부에서, 즉 단일 자율 시스템(AS, Autonomous System) 내에서 사용됩니다. 예를 들어, 한 대학교의 모든 네트워크가 IGP를 사용하여 데이터를 전달하는 방식입니다.

### 3. 주요 프로토콜:

- **OSPF (Open Shortest Path First):** 앞서 설명한 것처럼 OSPF는 IGP의 한 예로, 링크 상태 방식을 사용하여 네트워크 전체의 최단 경로를 계산합니다.
- **IS-IS (Intermediate System to Intermediate System):** 이 프로토콜도 IGP의 일종으로, 대규모 네트워크에서 확장성과 효율성을 제공하는 데 사용됩니다.
- **RIP (Routing Information Protocol):** 비교적 오래된 프로토콜로, 거리 벡터 방식을 사용합니다. RIP는 간단하지만, 확장성 및 빠른 수렴성 면에서는 OSPF나 IS-IS보다 덜 효율적입니다.

### 4. IGP의 특징:

- **빠른 수렴성:** 네트워크 내부에서 문제가 발생하면, IGP는 이를 신속하게 감지하고 대체 경로를 찾습니다.
- **네트워크 효율성:** IGP는 네트워크 자원을 효과적으로 사용하여 데이터를 전달하는 최적의 경로를 찾습니다.

## EGP (Exterior Gateway Protocol)란?

## 1. 기본 개념:

- EGP는 네트워크 간, 즉 서로 다른 자율 시스템(AS) 간의 데이터 이동을 관리하는 프로토콜입니다. 다시 비유를 들어보면, 여러 회사가 모여서 큰 네트워크(인터넷)를 형성하고 있다고 가정해 보세요. 이 회사들 간에는 서로 다른 규칙이 있을 수 있으며, 각 회사가 외부와 어떻게 소통할지 정하는 규칙이 필요합니다. EGP는 이러한 회사 간의 규칙을 정의합니다.

## 2. 어디에서 사용되나요?:

- EGP는 하나의 조직이 아닌, ISP(Internet Service Provider) 간 또는 큰 네트워크 간에 사용됩니다. 예를 들어, Google, Amazon, Facebook과 같은 대기업의 네트워크가 서로 소통할 때 EGP를 사용합니다.

## 3. 주요 프로토콜:

- **BGP (Border Gateway Protocol):** EGP의 대표적인 프로토콜로, 인터넷에서 가장 널리 사용됩니다. BGP는 AS 간의 경로 정보를 교환하며, 인터넷 상의 모든 네트워크 간 연결성을 유지하는 핵심 역할을 합니다. BGP는 정책 기반 라우팅을 지원하여, 단순히 최단 경로뿐만 아니라 다양한 비즈니스 정책을 반영하여 경로를 선택할 수 있습니다.

## 4. EGP의 특징:

- **확장성:** EGP는 전 세계적으로 매우 큰 규모의 네트워크 간 연결을 관리할 수 있도록 설계되었습니다.
- **정책 기반 라우팅:** EGP, 특히 BGP는 각 네트워크가 자체 정책을 설정하여 경로 선택에 영향을 줄 수 있게 합니다. 이는 비즈니스 요구 사항이나 보안 규정 등을 반영할 수 있는 유연성을 제공합니다.

## IGP와 EGP의 비교

### 1. 사용 범위:

- **IGP**는 하나의 자율 시스템(AS) 내에서 사용되며, 내부 네트워크 간의 최적의 경로를 찾는 데 초점을 맞춥니다.
- **EGP**는 여러 자율 시스템(AS) 간에 사용되며, 네트워크 간의 경로 선택과 정책을 관리하는 데 중점을 둡니다.

### 2. 프로토콜 종류:

- **IGP:** OSPF, IS-IS, RIP 등이 있습니다. 이들은 내부 네트워크 최적화에 중점을 둡니다.

- **EGP:** BGP가 주요 프로토콜로, 인터넷 상의 AS 간 연결을 관리합니다.

### 3. 경로 선택 기준:

- **IGP:** 주로 최단 경로를 기준으로 하여 내부 네트워크의 빠른 데이터 전달을 목표로 합니다.
- **EGP:** 경로 선택 시 정책과 규칙이 중요하며, 비즈니스 요구나 보안 정책을 반영할 수 있습니다.

### 요약

- **IGP**는 회사 내부의 네트워크 흐름을 관리하는 규칙으로, 네트워크 안에서 최적의 경로를 찾기 위한 프로토콜입니다. (예: OSPF, IS-IS)
- **EGP**는 회사 간 또는 큰 네트워크 간의 연결을 관리하는 규칙으로, 서로 다른 네트워크 간의 데이터 이동을 관리하는 프로토콜입니다. (예: BGP)

이해를 돕기 위해, 라우팅과 관련된 기본 개념(예: 라우터의 역할, 자율 시스템의 개념)에 대해 추가 설명이 필요하신지요? 더 궁금한 점이 있다면 언제든지 질문해주세요!

## 라우팅 경로 결정 기준

- 매트릭과 AD

## OSPF의 테이블구조

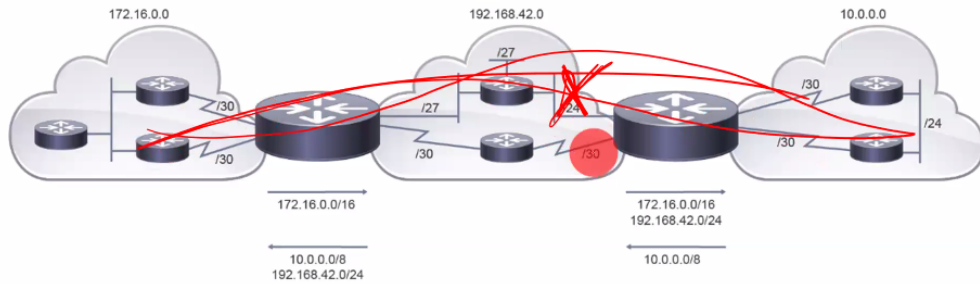
### Link-State Routing Protocol Overview

Link-state routing protocols such as OSPF have several advantages when compared to traditional distance vector routing protocols.

- Link-state protocols are more scalable.
- Each router has a full map of the topology.
- Updates are sent when a topology change occurs and are reflooded periodically.
- Link-state protocols respond quickly to topology changes.
- More information is communicated between the routers.

12. 11

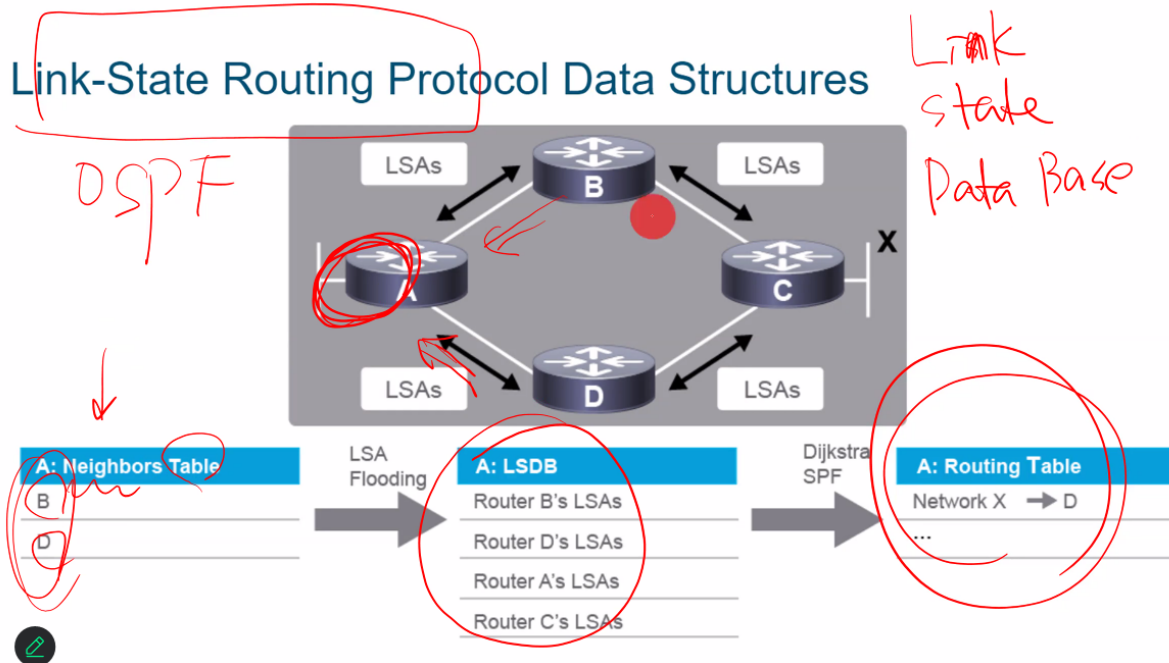
## Dynamic Routing Protocols (Cont.)



Different protocols behave differently.

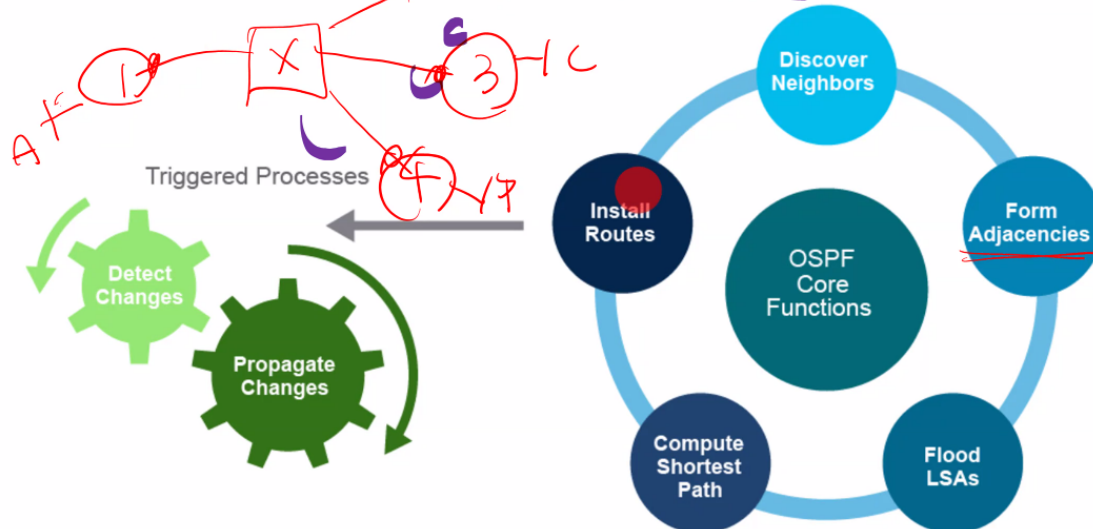
- IGP versus EGP
- Distance vector versus link state
- Classless versus classful

## Link-State Routing Protocol Data Structures



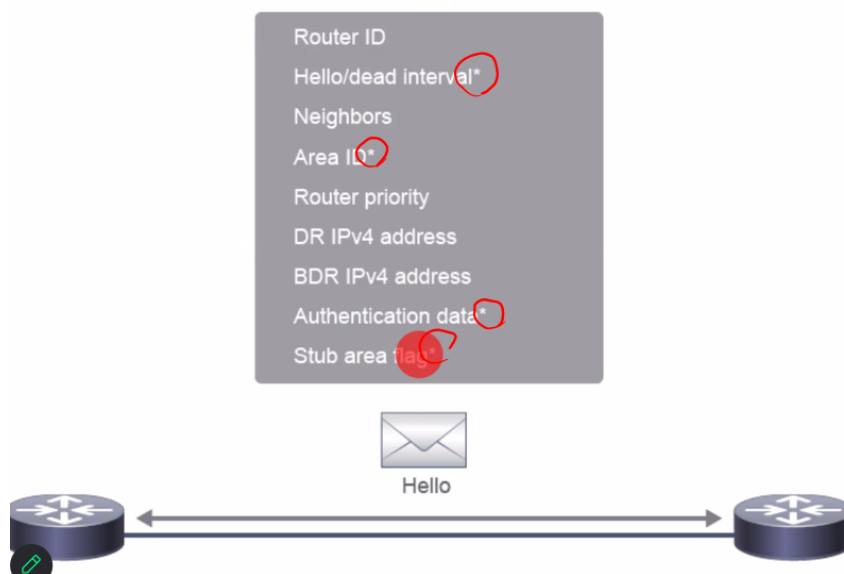
9

**A, C, D**



- 1

## Establishing OSPF Neighbor Adjacencies



- OSPF routers first establish neighbor adjacencies.
- Hello packets are periodically sent to the all OSPF routers IPv4 address 224.0.0.5.
- Routers must agree on certain information (\*) inside the hello packet before adjacency can be established.

- \* 표시가된 위의 4가지가 같으면 네이버 관계를 맺는다.

## 네이버 관계를 맺는 형태

### Establishing OSPF Neighbor Adjacencies (Cont.)

OSPF routers establish a neighbor relationship over point-to-point links.

- Commonly a serial interface running either PPP or High-Level Data Link Control (HDLC)
- May also be a point-to-point subinterface running Frame Relay or ATM
- Does not require DR or BDR election
- May be optionally configured on Ethernet links for directly connected routers as a non-default media type: point-to-point. In this case, it should be configured on both ends of the link (both router interfaces as point-to-point)

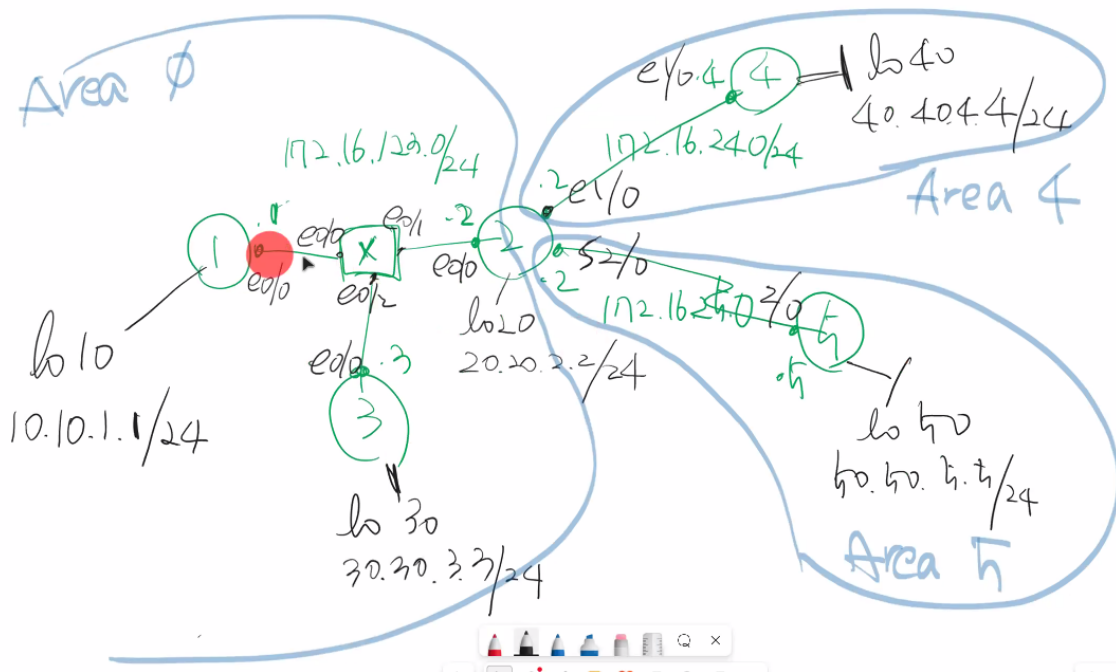


- 유일하게 HDLC를 이용해서 네이버 관계를 맺는다.

## 정리

- OSPF는 내부 네트워크에서 라우터들끼리 빠른 라우팅 경로를 찾아주는 프로토콜이다.
- 링크 스테이트 라우팅 프로토콜이라고도 불리며 네이버 테이블 구조를 사용한다.
- Area 디자인을 꼭 해야한다. Area끼리 묶는이유는 리소스 최대활용 목적
- Hello 패킷 주고받으면서 네이버 관계를 맺고 Area 내부의 모든 라우터들에게 네트워크 정보를 뿌린다.
- LSDB를 만들고 알고리즘을 계산해서 빠른 경로를 만들어낸다.

## OSPF 실습



## 프롬프트

OSPF 설정을 위한 라우터 인터페이스 설정 정보입니다

\*\*\* R1

```
conf t
int lo 10
ip add 10.10.1.1 255.255.255.0
no sh
int e0/0
ip add 172.16.123.1 255.255.255.0
no sh
line vty 0 4
pass cisco
```

\*\*\* R3

```
conf t
int lo 30
ip add 30.30.3.3 255.255.255.0
no sh
int e0/0
```



```
ip add 172.16.123.3 255.255.255.0
no sh
line vty 0 4
pass cisco
```

\*\*\* R2

```
conf t
int lo 20
ip add 20.20.2.2 255.255.255.0
no sh
int e0/0
ip add 172.16.123.2 255.255.255.0
no sh
int e1/0
ip add 172.16.24.2 255.255.255.0
no sh
int s2/0
ip add 172.16.25.2 255.255.255.0
no sh
line vty 0 4
pass cisco
```

\*\*\* R4

```
conf t
int lo 40
ip add 40.40.4.4 255.255.255.0
int e1/0
ip add 172.16.24.4 255.255.255.0
no sh
line vty 0 4
pass cisco
```

```

*** R5
conf t
int lo 50
ip add 50.50.5.5 255.255.255.0
int s2/0
ip add 172.16.25.5 255.255.255.0
no sh
line vty 0 4
pass cisco

```

\*\*\*\*\* ping 테스트 \*\*\*\*\*

```

R2에서
ping 172.16.123.1
ping 172.16.123.3
ping 172.16.24.4
ping 172.16.25.5

```

\*\*\*\*\* R1 - R5 설정 저장 \*\*\*\*\*

```

end
wr

```

\*\*\*\*\* R2 OSPF 설정 \*\*\*\*\*

```

conf t
router ospf 2
int lo 20
ip ospf 2 area 0
int e0/0
ip ospf 2 area 0

```

\*\*\*\*\* R3 \*\*\*\*\*

```

conf t
router ospf 3

```

```
int lo 30
  ip ospf 3 area 0
int e0/0
  ip ospf 3 area 0
```

```
conf t
router ospf 1
int lo 10
  ip ospf 1 area 0
int e0/0
  ip ospf 1 area 0
```

```
*** R1
conf t
int lo 10
  ip ospf network point-to-point

*** R2
conf t
int lo 20
  ip ospf network point-to-point

*** R3
conf t
int lo 30
  ip ospf network point-to-point
```