

# 20장 리다이렉션과 부하 균형

데이터인프라개발팀

조영일

# 목차

- 왜 리다이렉트인가?
- 리다이렉트할 곳
- 리다이렉션 프로토콜의 개요
- 일반적인 리다이렉션 방법
- 프락시 리다이렉션 방법
- 캐시 리다이렉션 방법
- 인터넷 캐시 프로토콜
- 캐시 배열 라우팅 프로토콜
- 하이퍼텍스트 캐싱 프로토콜

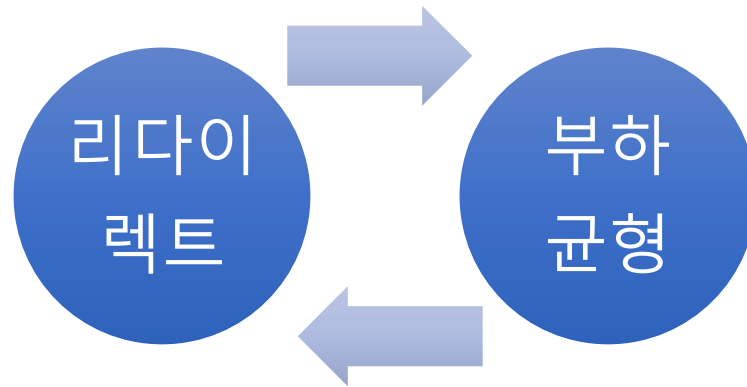
# 왜 리다이렉트인가?

- 리다이렉션?
  - 메시지를 여러 다른 경로로 보내거나 우회시키는 것



# 왜 리다이렉트인가?

- 필요성
  - 서비스 이중화
  - 지연 최소화
  - 네트워크 대역폭 절약



# 리다이렉트할 곳

- 인터넷 서비스 구성요소
  - 서버
  - 프락시
  - 캐시
  - 게이트웨이
- 공통점도 있으나 특수성도 존재하여 리다이렉션 기법이 달라질 수 있음
  - 서버는 IP 관점
  - 프락시(캐시)는 프로토콜 관점

# 리다이렉션 프로토콜의 개요

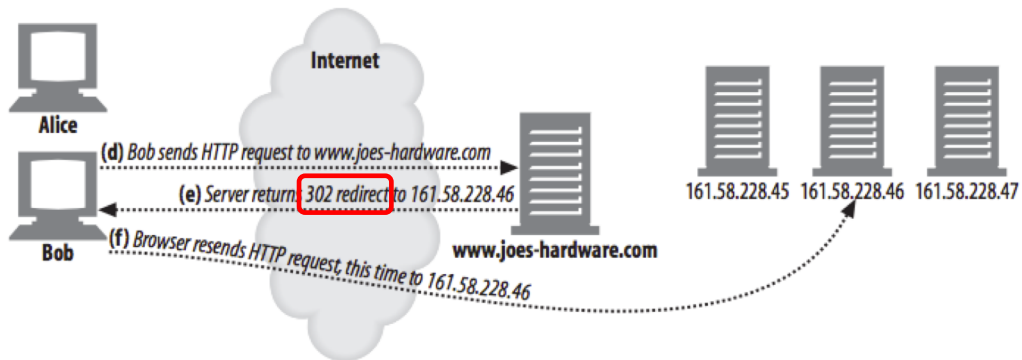
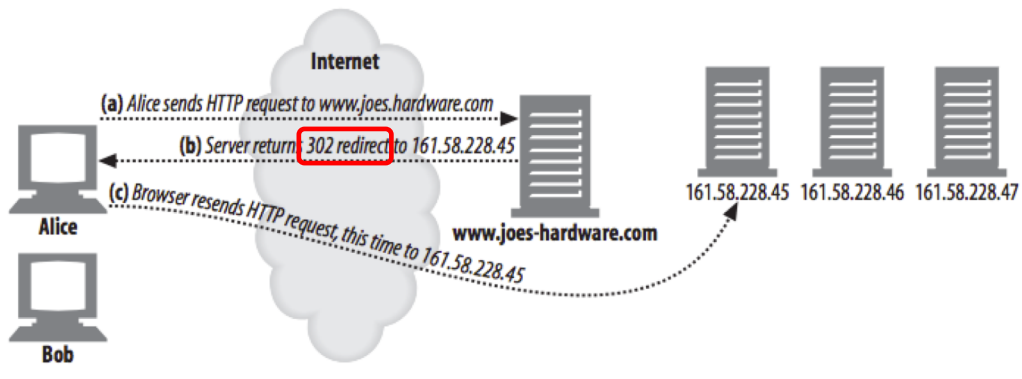
- 목표
  - 메시지를 서버로 가급적 빠르게 보내는 것
- 영향을 주는 요소들
  - 브라우저의 프락시 설정
  - DNS Resolution
  - 동일 메시지의 분할된 패킷의 라우팅
  - 웹 서버의 HTTP Redirect

# 일반적인 리다이렉션 방법

- HTTP 리다이렉션
- DNS 리다이렉션
- 임의 캐스트 어드레싱
- 아이피 맥 포워딩
- 아이피 주소 포워딩
- 네트워크 구성요소 제어 프로토콜

# 일반적인 리다이렉션 방법

- HTTP 리다이렉션



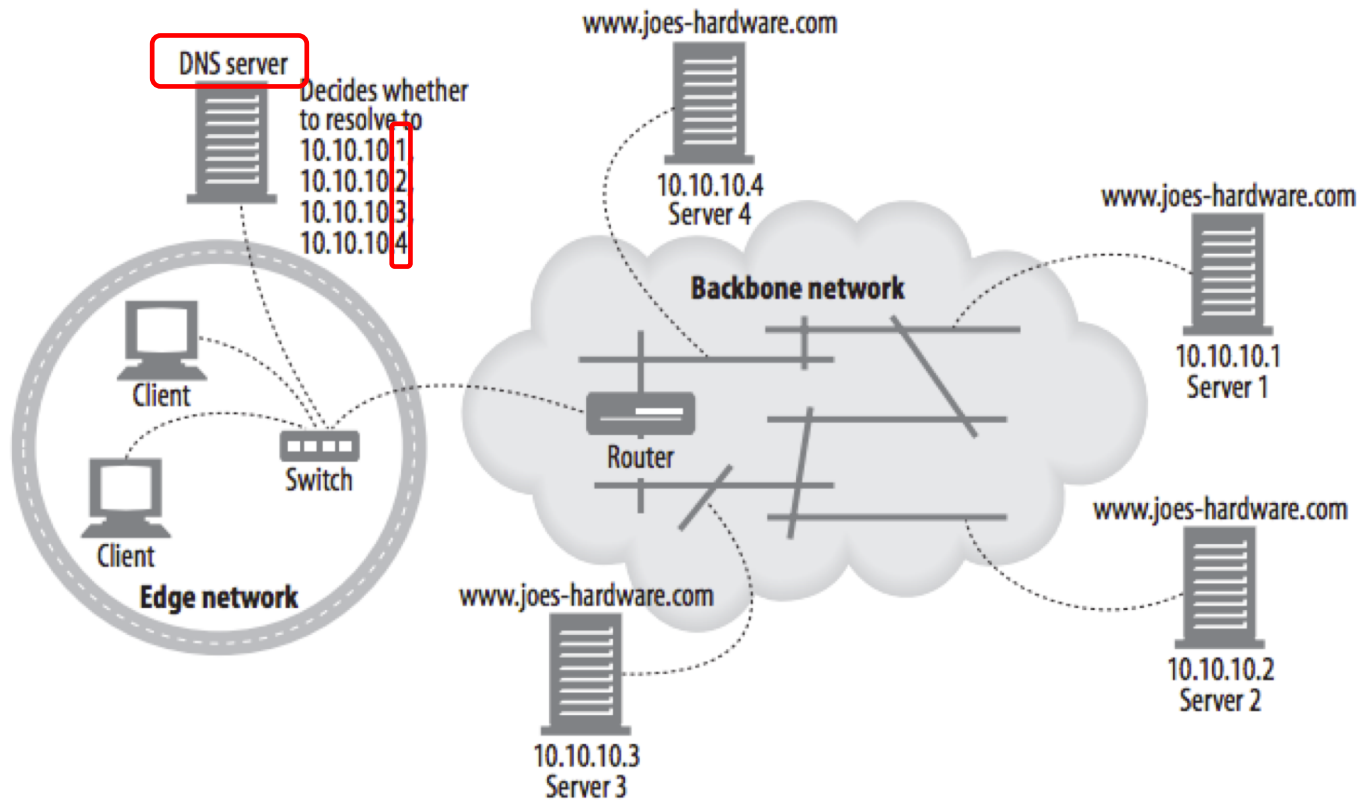


# 일반적인 리다이렉션 방법

- 단점
  - 예상 외로 큰 서버 부하
  - 사용자 대기 시간 증가
  - 대표 서버가 SPoF

# 일반적인 리다이렉션 방법

- DNS 리다이렉션



# 일반적인 리다이렉션 방법

- DNS 리다이렉션
  - 단순 round robin

• 다:

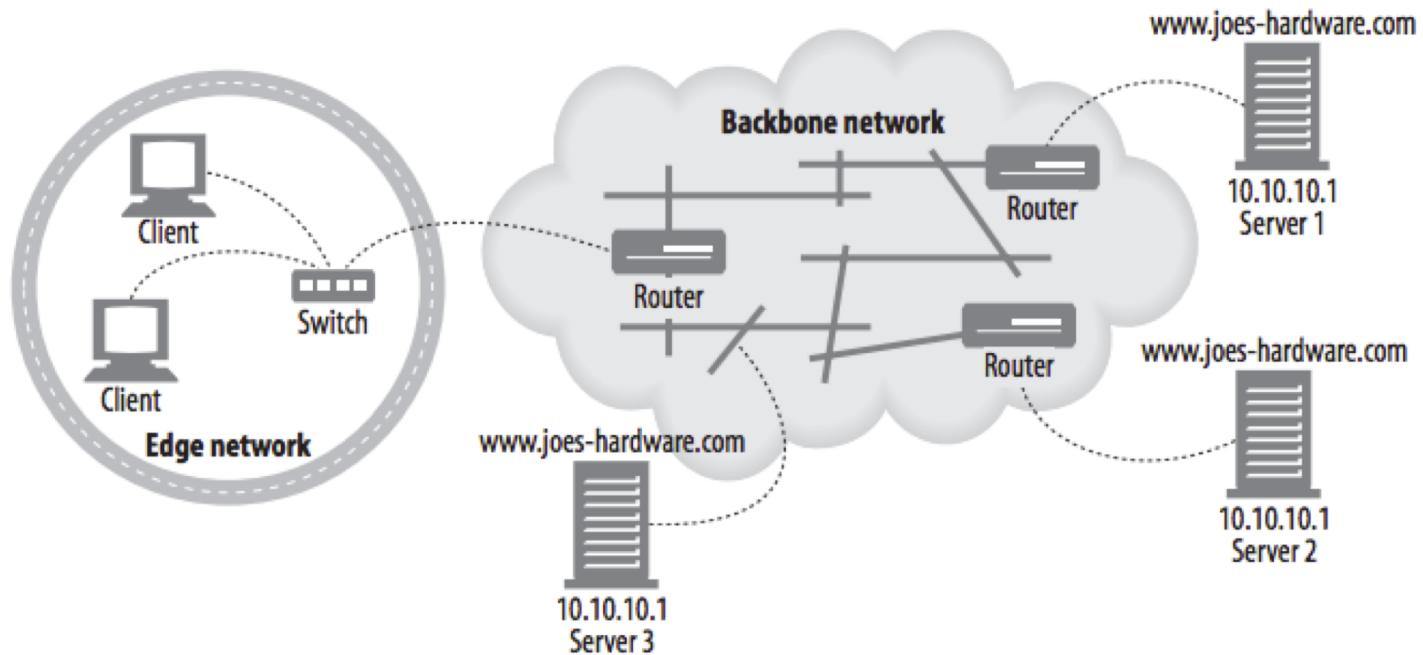
```
% nslookup www.cnn.com
Name:      cnn.com
Addresses: 207.25.71.5, 207.25.71.6, 207.25.71.7, 207.25.71.8
           207.25.71.9, 207.25.71.12, 207.25.71.20, 207.25.71.22, 207.25.71.23
           207.25.71.24, 207.25.71.25, 207.25.71.26, 207.25.71.27, 207.25.71.28
           207.25.71.29, 207.25.71.30, 207.25.71.82, 207.25.71.199, 207.25.71.245
           207.25.71.246
Aliases:   www.cnn.com

% nslookup www.cnn.com
Name:      cnn.com
Addresses: 207.25.71.5, 207.25.71.6, 207.25.71.7, 207.25.71.8
           207.25.71.9, 207.25.71.12, 207.25.71.20, 207.25.71.22, 207.25.71.23
           207.25.71.24, 207.25.71.25, 207.25.71.26, 207.25.71.27, 207.25.71.28
           207.25.71.29, 207.25.71.30, 207.25.71.82, 207.25.71.199, 207.25.71.245
           207.25.71.246

% nslookup www.cnn.com
Name:      cnn.com
Addresses: 207.25.71.6, 207.25.71.7, 207.25.71.8, 207.25.71.9
           207.25.71.12, 207.25.71.20, 207.25.71.22, 207.25.71.23, 207.25.71.24
           207.25.71.25, 207.25.71.26, 207.25.71.27, 207.25.71.28, 207.25.71.29
           207.25.71.30, 207.25.71.82, 207.25.71.199, 207.25.71.245, 207.25.71.246
           207.25.71.5
```

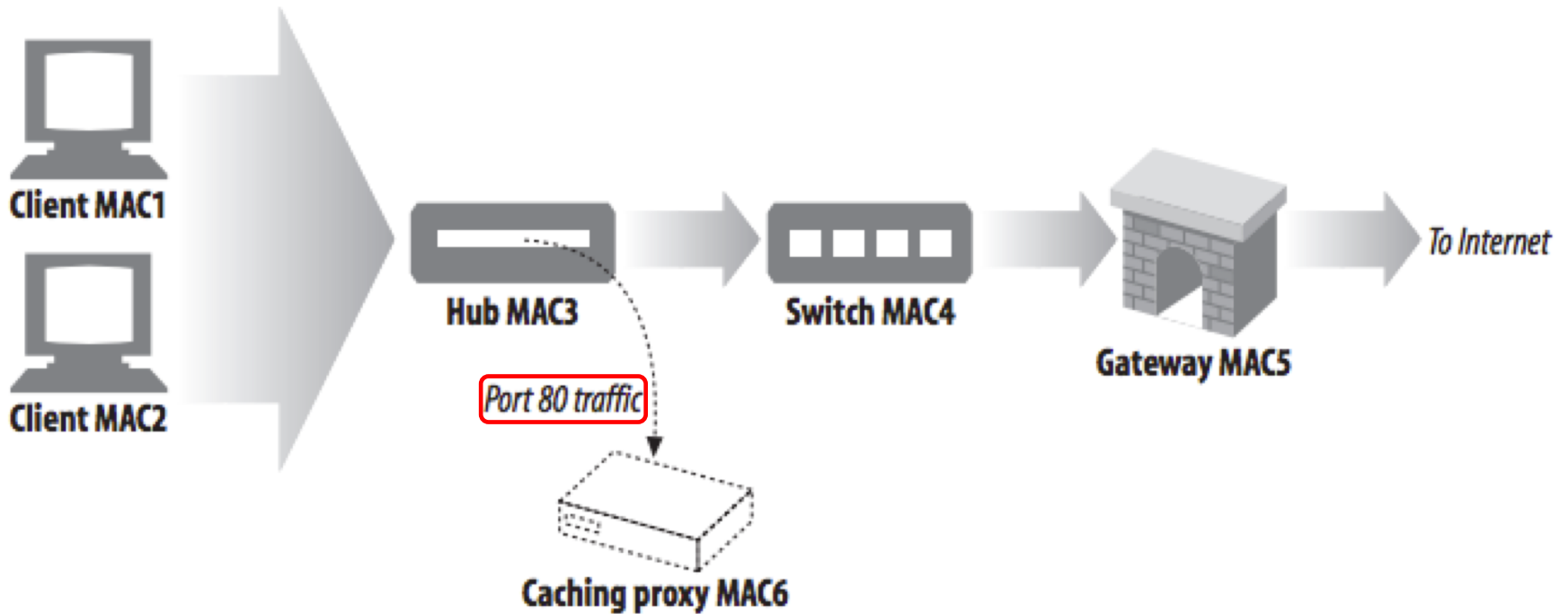
# 일반적인 리다이렉션 방법

- 임의 캐스트 어드레싱



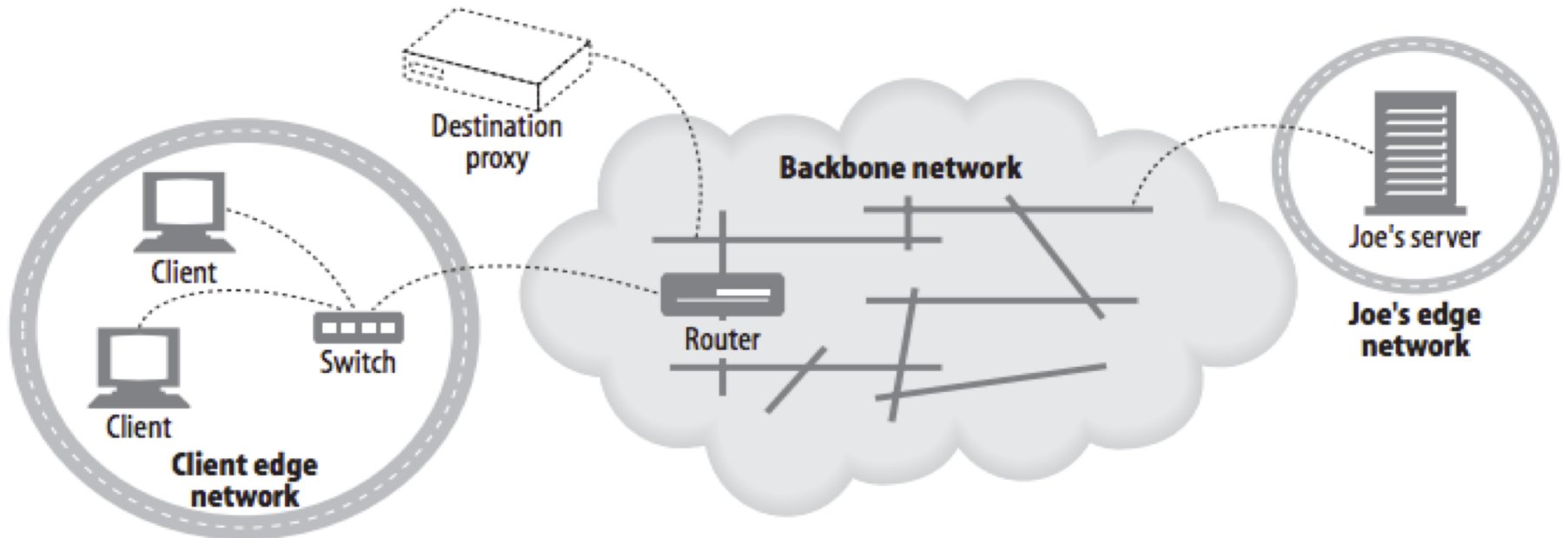
# 일반적인 리다이렉션 방법

- 아이피 맥 포워딩



# 일반적인 리다이렉션 방법

- 아이피 주소 포워딩
  - NAT
  - 라우팅 대칭성 문제 발생



# 일반적인 리다이렉션 방법

- 네트워크 구성요소 제어 프로토콜(NECP)
  - Skip!
  - 표준이 아닌 드래프트 단계

# 목차 (REVISITED)

- 왜 리다이렉트인가?
- 리다이렉트할 곳
- 리다이렉션 프로토콜의 개요
- 일반적인 리다이렉션 방법
- 프락시 리다이렉션 방법
- 캐시 리다이렉션 방법
- 인터넷 캐시 프로토콜
- 캐시 배열 라우팅 프로토콜
- 하이퍼텍스트 캐싱 프로토콜

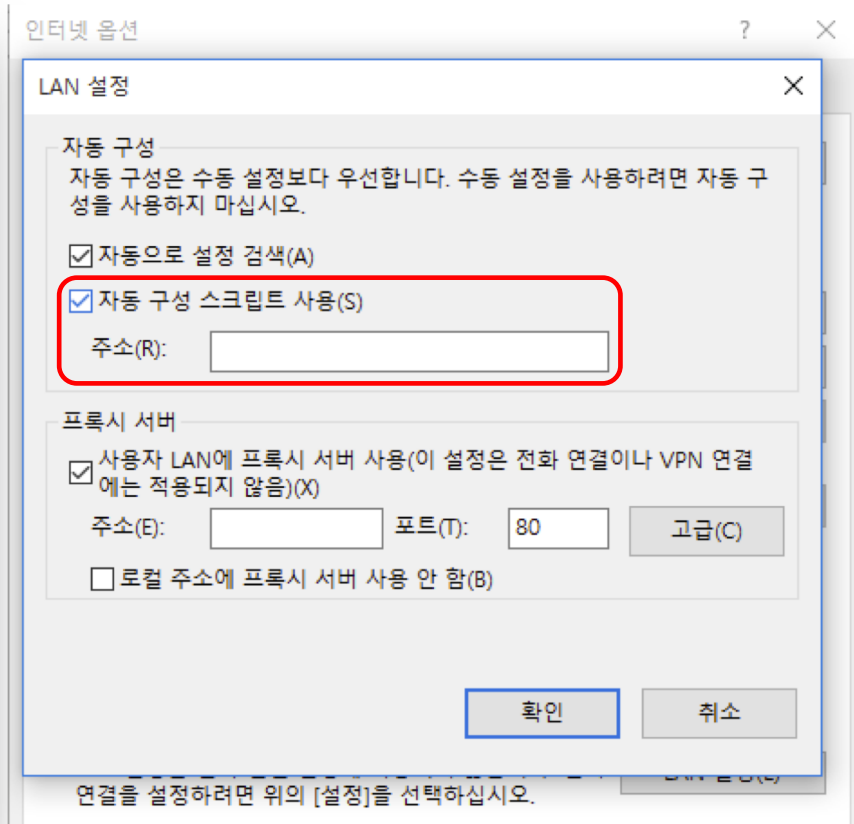
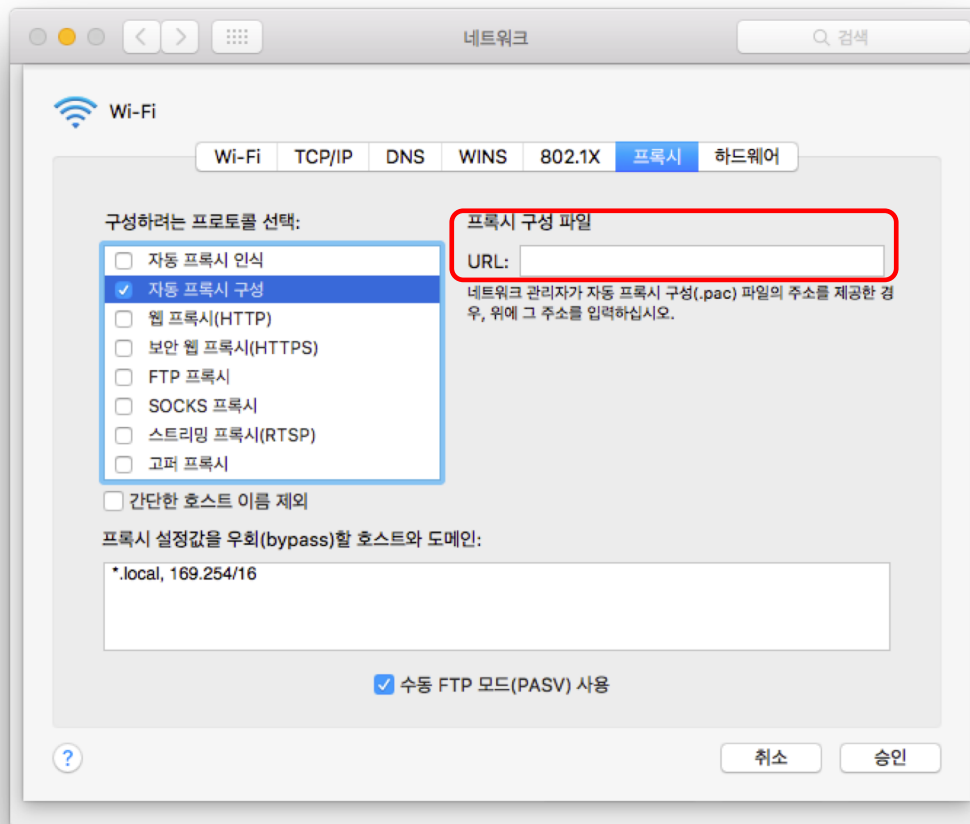


# 프락시 리다이렉션 방법

- 명시적 브라우저 설정
- 프락시 자동 설정(PAC)
- 웹 프락시 자동발견 프로토콜(WPAD)

# 프락시 리다이렉션 방법

- PAC file



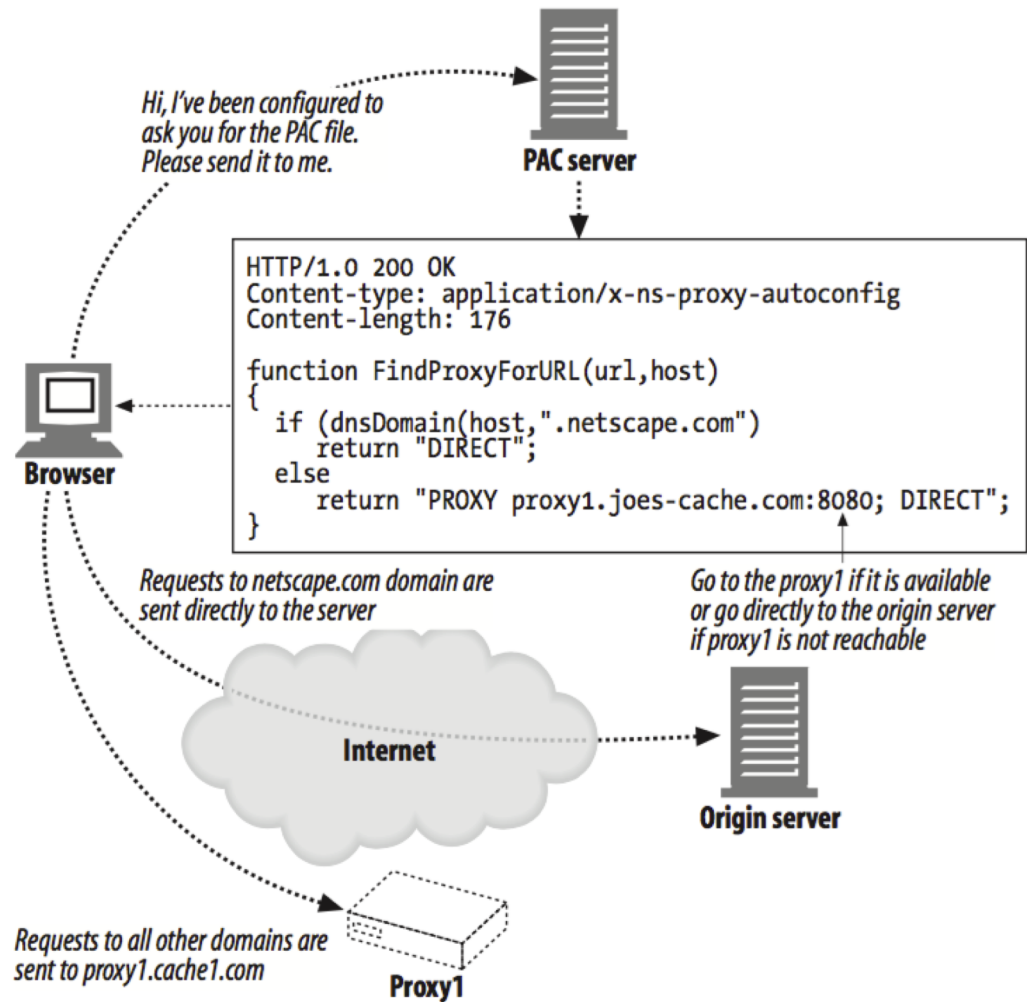
# 프락시 리다이렉션 방법

- PAC file

```
function FindProxyForURL(url, host) {  
    // our local URLs from the domains below example.com don't need a proxy:  
    if (shExpMatch(host, "*.example.com"))  
    {  
        return "DIRECT";  
    }  
  
    // URLs within this network are accessed through  
    // port 8080 on fastproxy.example.com:  
    if (isInNet(host, "10.0.0.0", "255.255.248.0"))  
    {  
        return "PROXY fastproxy.example.com:8080";  
    }  
  
    // All other requests go through port 8080 of proxy.example.com.  
    // should that fail to respond, go directly to the WWW:  
    return "PROXY proxy.example.com:8080; DIRECT";  
}
```

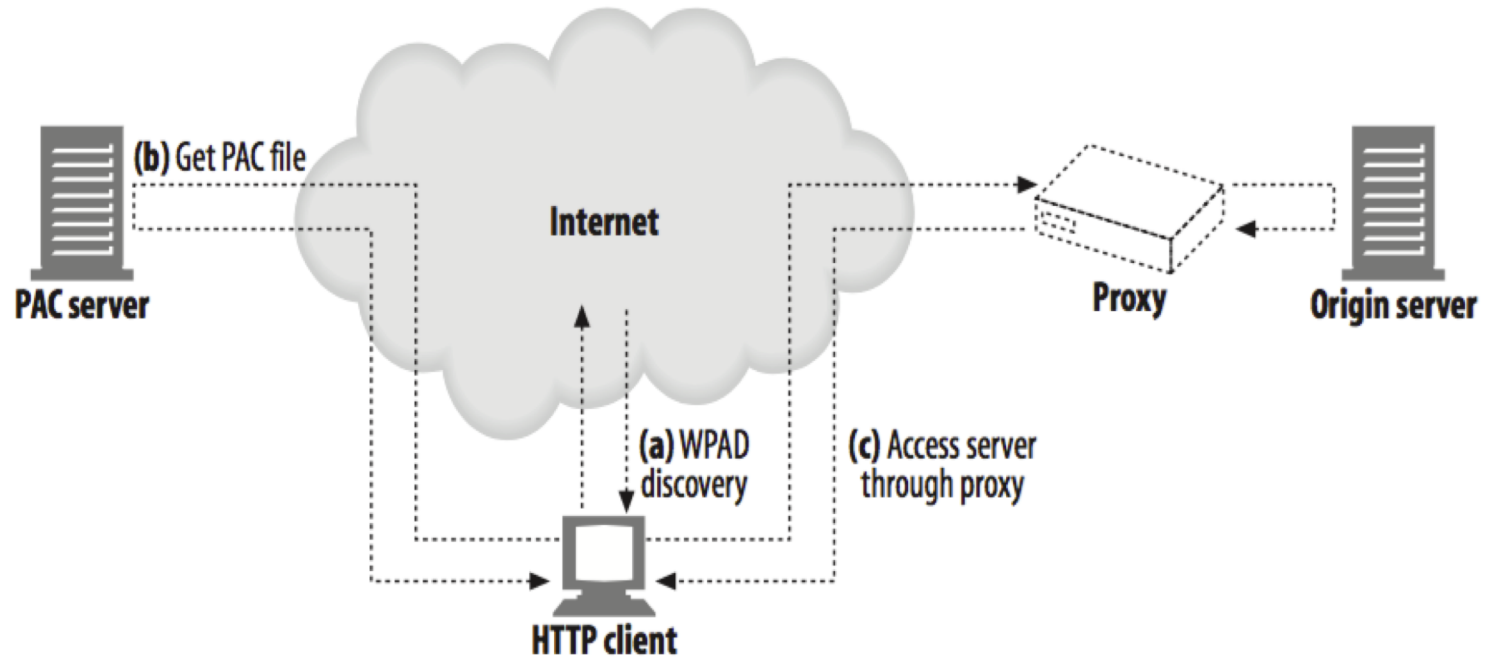
# 프락시 리다이렉션 방법

- PAC 동작 원리



# 프락시 리다이렉션 방법

- WPAD



# 캐시 리다이렉션 방법

- WCCP(Web Cache Comm.. Prot..) 리다이렉션
  - 시스코에서 제안
  - 라우터가 캐시들을 관리
  - Skip!

# 인터넷 캐시 프로토콜

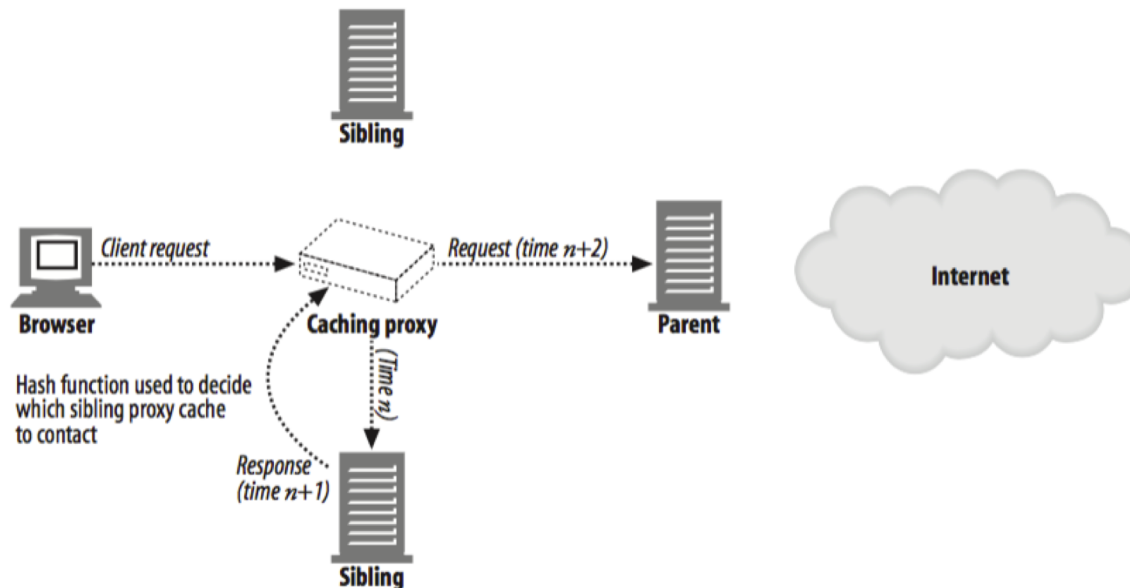
## • ICP

- 일종의 캐시 클러스터링 프로토콜 (in UDP)
- 캐시가 자신이 요청받은 컨텐츠가 없으면 형제 캐시에서 찾아서 반환
  - 중복 / 누락?
- URL을 broadcast
- HIT/MISS로 회신

# 캐시 배열 라우팅 프로토콜

- CARP

- disjoint한 캐시프락시 서버군
- 클라이언트가 hash function 제공





# 하이퍼텍스트 캐싱 프로토콜

- 기존 캐싱 프로토콜의 문제점
  - URL만으로 판단 - 헤더가 고려되지 않음
- HTCP
  - ICP를 보완하는 프로토콜
  - 형제들에게 URL+헤더로 캐시의 존재여부를 물어보는 통신 프로토콜

끝