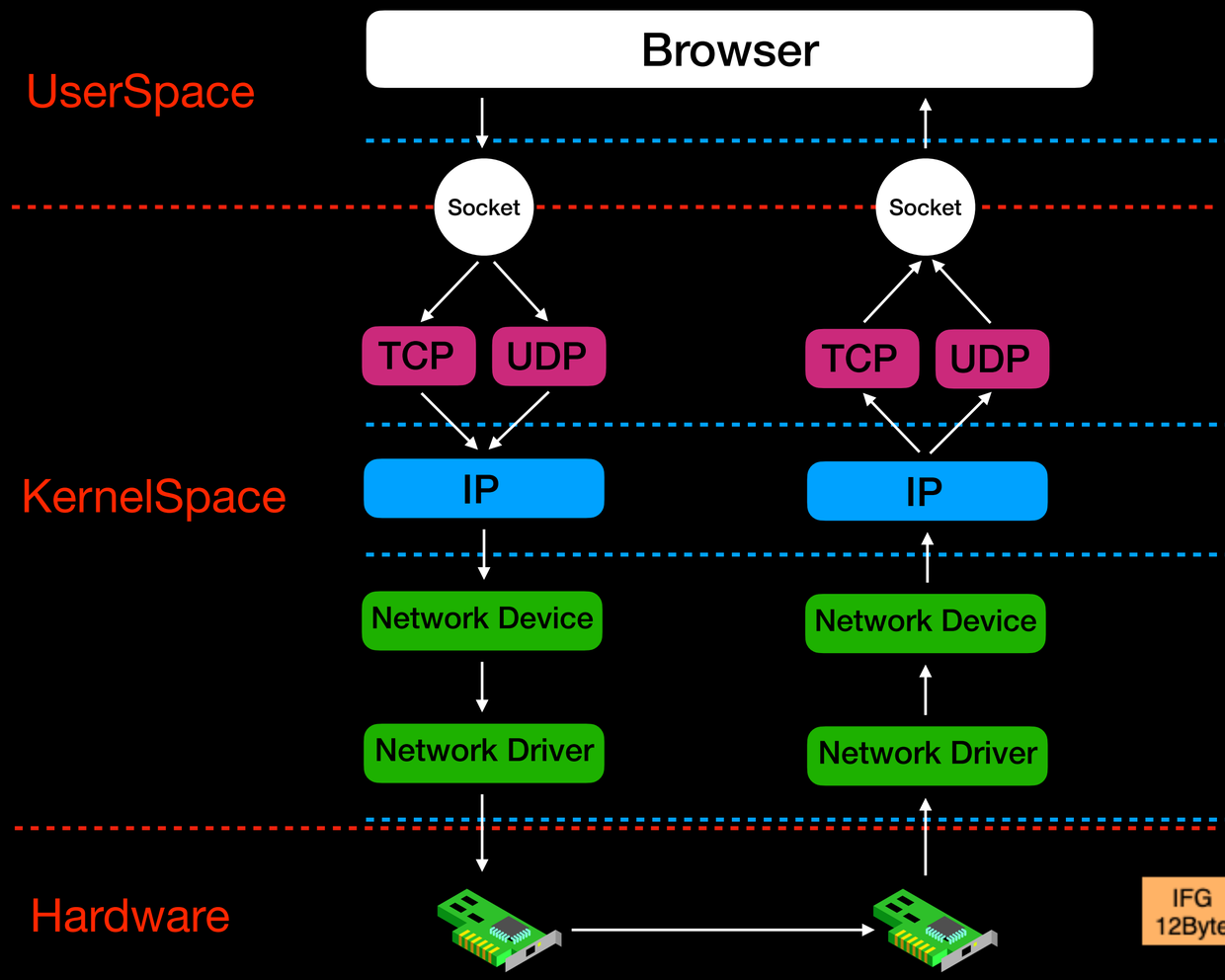
# 데이터의 흐름

( 브라우저에서 서버까지 )

홀스홀리커

# 가장 간단한 그림

# Client에서의 동작




OSI 7 Layer

Application ( L7 )
Presentation ( L6 )
Session ( L5 )

Data
1460Byte

Transport ( L4 )

TCP
20Byte
Data
1460Byte

Network ( L3 )

IP
20Byte
Segment
1480Byte

Data Link ( L2 )

Ethernet
14Byte
Packet
1500Byte

Physical ( L1 )

IFG
12Byte
Preamble/SFD
8Byte
Frame
1514Byte
FCS
4Byte

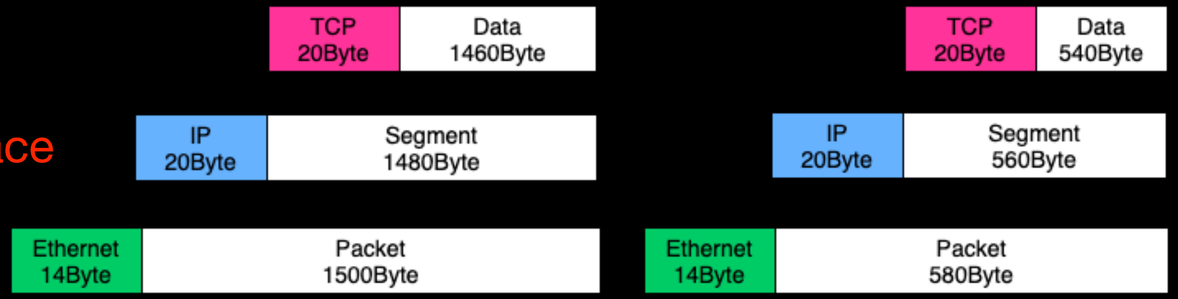
홀스홀리커

# Client에서의 동작

UserSpace

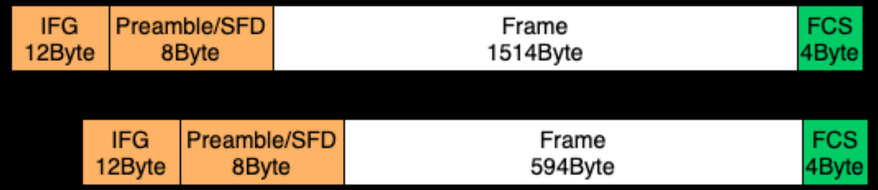

- MSS : 1460Byte
- MTU : 1500Byte

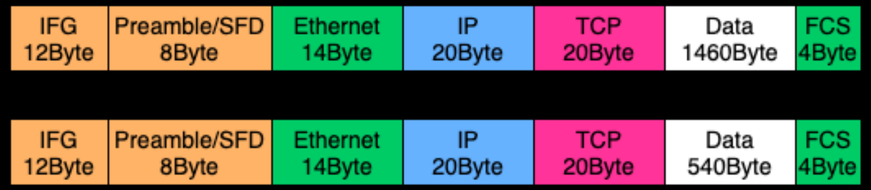
KernelSpace





Hardware




- TSO, GSO, LRO, GRO 등 미 적용
- TCP Segmentation, IP Fragmentation, Checksum 을 CPU에서 합니다

- TSO, GSO, LRO, GRO 등 적용
- TCP Segmentation, IP Fragmentation, Checksum을 NIC에서 합니다

# DNS 통신

Client
한국

1. hosts File
2. Client DNS Cache

3. google.com 질의

11. X.X.X.X IP 응답

Local DNS

4. Local DNS Cache

5. google.com 질의

6. .com NS 정보 응답

Root DNS

7. google.com 질의

8. google.com NS 정보 응답

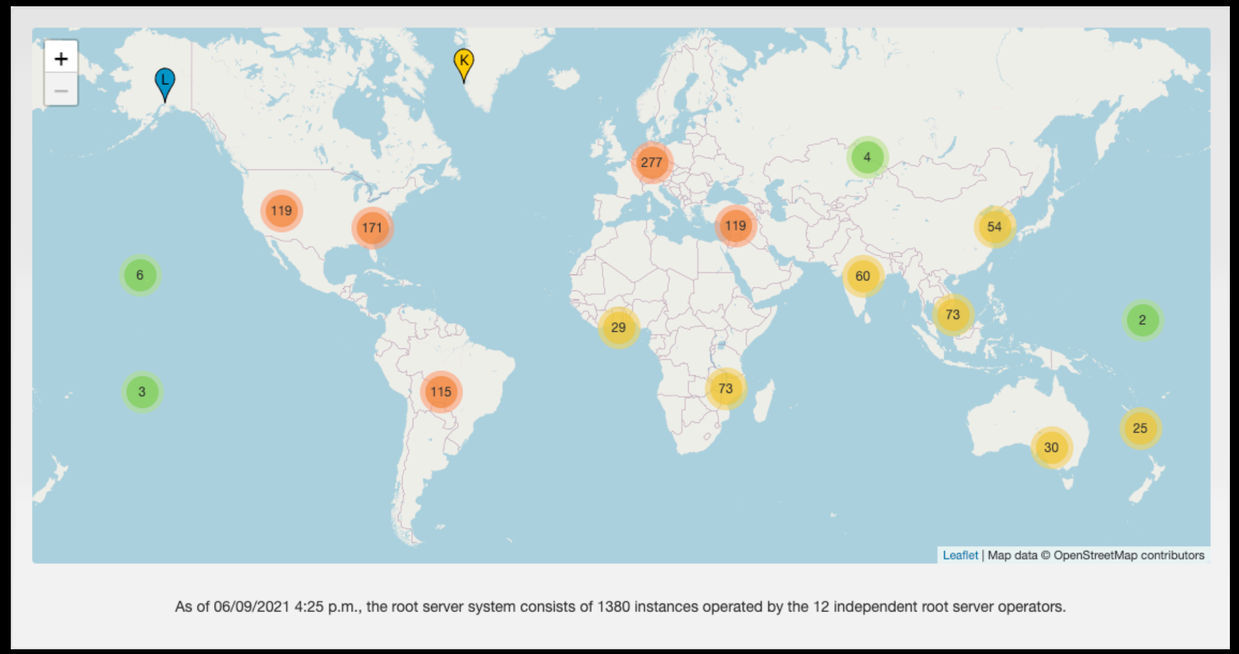
.com DNS

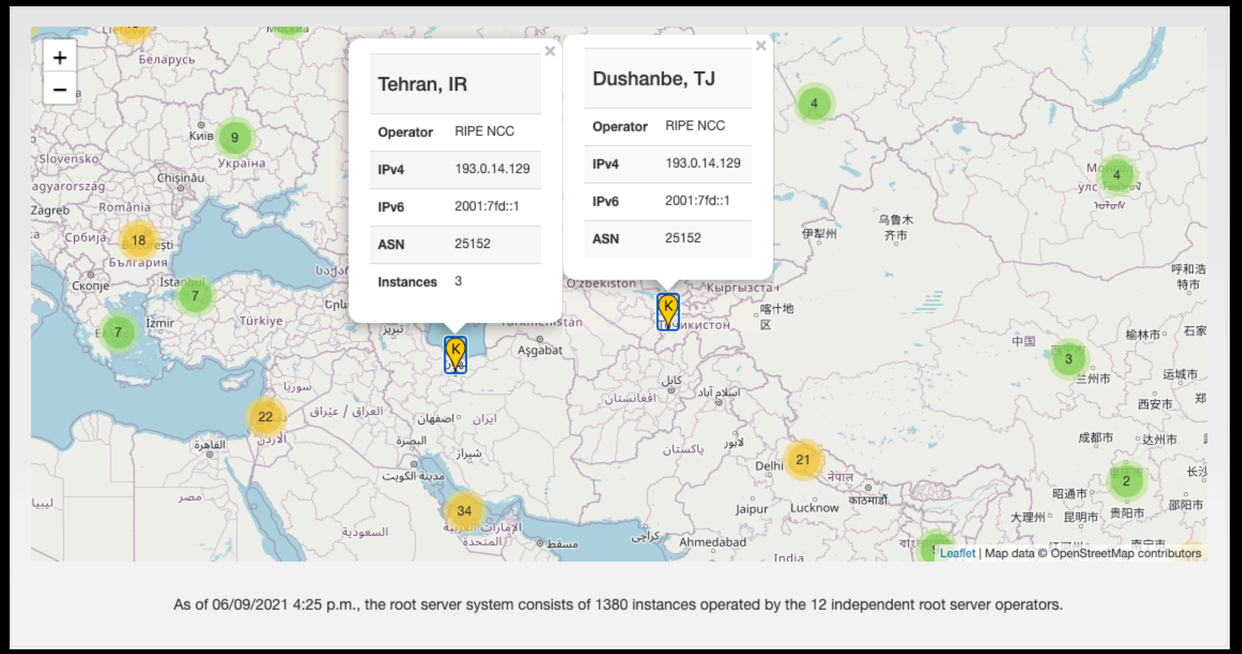
9. google.com 질의

10. X.X.X.X IP 응답

google.com DNS

# DNS 통신 ( Root DNS )




출처 : https://root-servers.org/

➢ 13개의 Root DNS로 구성되어 있으며, 12개의 조직이 전세계적으로 관리하고 있습니다.

➢ 빠른 응답, 부하 분산, 동일 IP 사용을 위해 BGP Anycast로 구성되어 있습니다

# DNS 통신 ( nslookup )

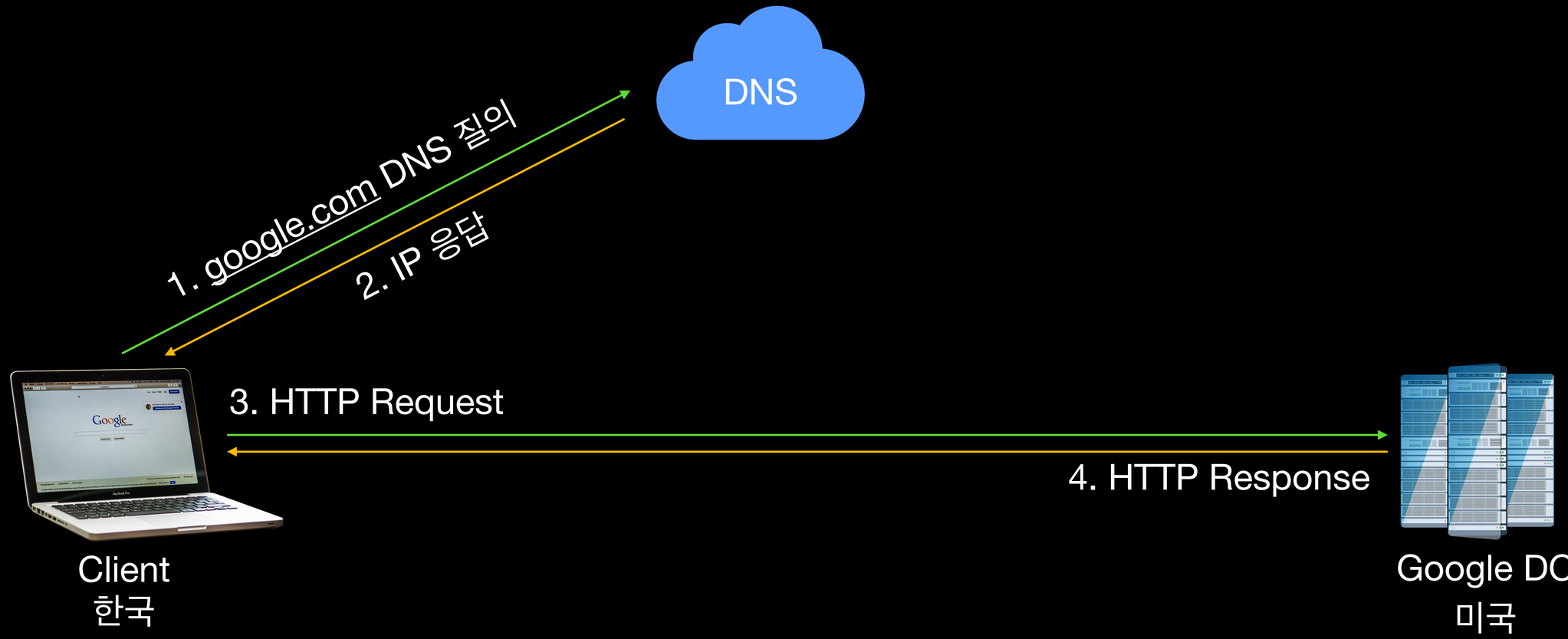
➢ 도메인에 대한 정보가 없을 경우
네임서버 혹은 캐시서버에서 응답해 줄 때는
권한 없는 응답(Non-Authoritative Answer)
이라고 나옵니다

➢ 캐시서버에서 DNS 정보에 대한 캐시 기간은
TTL 시간을 따라갑니다

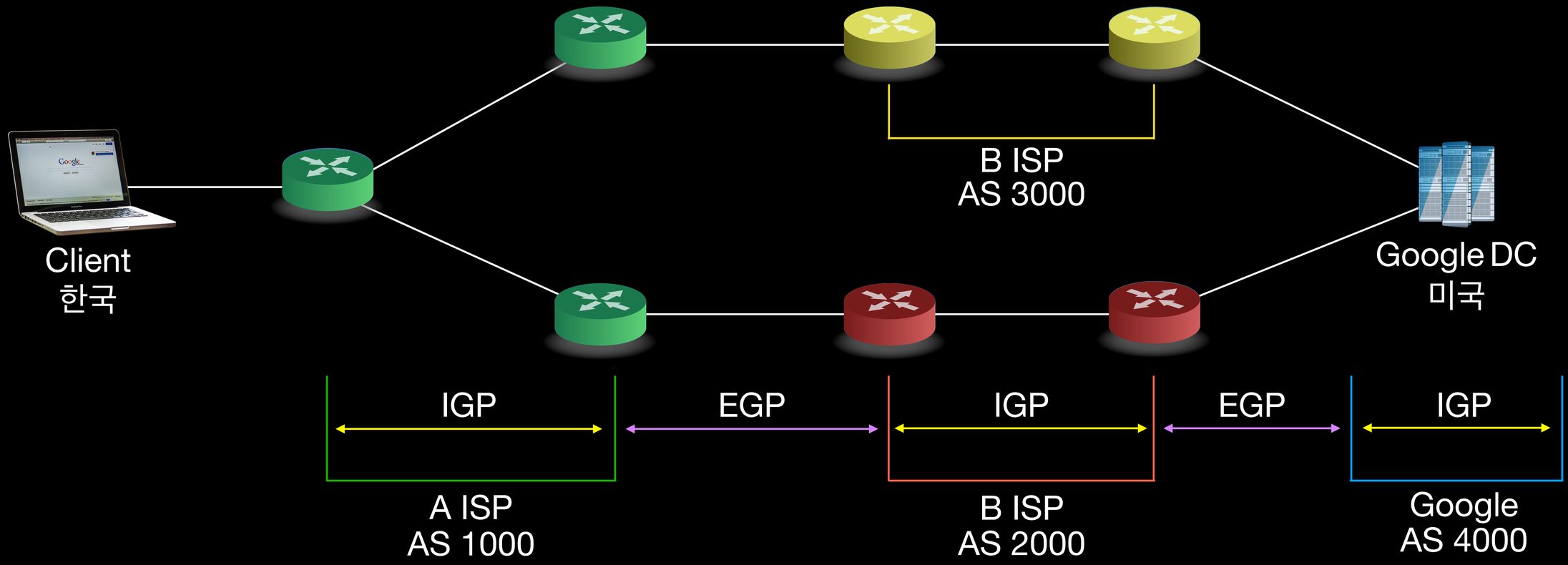
```
C:₩Users₩_     >nslookup -debug google.com
------------
Got answer:
    HEADER:
        opcode = QUERY, id = 1, rcode = NOERROR
        header flags:  response, want recursion, recursion avail.
        questions = 1,  answers = 1,  authority records = 0,  additional = 0

    QUESTIONS:
        8.8.8.8.in-addr.arpa, type = PTR, class = IN
    ANSWERS:
    ->  8.8.8.8.in-addr.arpa
        name = dns.google
        ttl = 19286 (5 hours 21 mins 26 secs)

------------
서버:    dns.google
Address:  8.8.8.8

------------
Got answer:
    HEADER:
        opcode = QUERY, id = 2, rcode = NOERROR
        header flags:  response, want recursion, recursion avail.
        questions = 1,  answers = 1,  authority records = 0,  additional = 0

    QUESTIONS:
        google.com, type = A, class = IN
    ANSWERS:
    ->  google.com
        internet address = 172.217.26.46
        ttl = 293 (4 mins 53 secs)

------------
권한 없는 응답:
------------
Got answer:
    HEADER:
        opcode = QUERY, id = 3, rcode = NOERROR
        header flags:  response, want recursion, recursion avail.
        questions = 1,  answers = 1,  authority records = 0,  additional = 0

    QUESTIONS:
        google.com, type = AAAA, class = IN
    ANSWERS:
    ->  google.com
        AAAA IPv6 address = 2404:6800:4004:80a::200e
        ttl = 223 (3 mins 43 secs)

------------
이름:    google.com
Addresses:  2404:6800:4004:80a::200e
          172.217.26.46
```

```
C:₩Users₩      >nslookup -debug google.com 216.239.38.10
------------
Got answer:
    HEADER:
        opcode = QUERY, id = 1, rcode = NOERROR
        header flags:  response, auth. answer, want recursion
        questions = 1,  answers = 1,  authority records = 0,  additional = 0

    QUESTIONS:
        10.38.239.216.in-addr.arpa, type = PTR, class = IN
    ANSWERS:
    ->  10.38.239.216.in-addr.arpa
        name = ns4.google.com
        ttl = 86400 (1 day)

------------
서버:    ns4.google.com
Address:  216.239.38.10

------------
Got answer:
    HEADER:
        opcode = QUERY, id = 2, rcode = NOERROR
        header flags:  response, auth. answer, want recursion
        questions = 1,  answers = 1,  authority records = 0,  additional = 0

    QUESTIONS:
        google.com, type = A, class = IN
    ANSWERS:
    ->  google.com
        internet address = 172.217.31.142
        ttl = 300 (5 mins)

------------
------------
Got answer:
    HEADER:
        opcode = QUERY, id = 3, rcode = NOERROR
        header flags:  response, auth. answer, want recursion
        questions = 1,  answers = 1,  authority records = 0,  additional = 0

    QUESTIONS:
        google.com, type = AAAA, class = IN
    ANSWERS:
    ->  google.com
        AAAA IPv6 address = 2404:6800:4004:808::200e
        ttl = 300 (5 mins)

------------
이름:    google.com
Addresses:  2404:6800:4004:808::200e
          172.217.31.142
```

# Domain Name System 개선



# Autonomous System

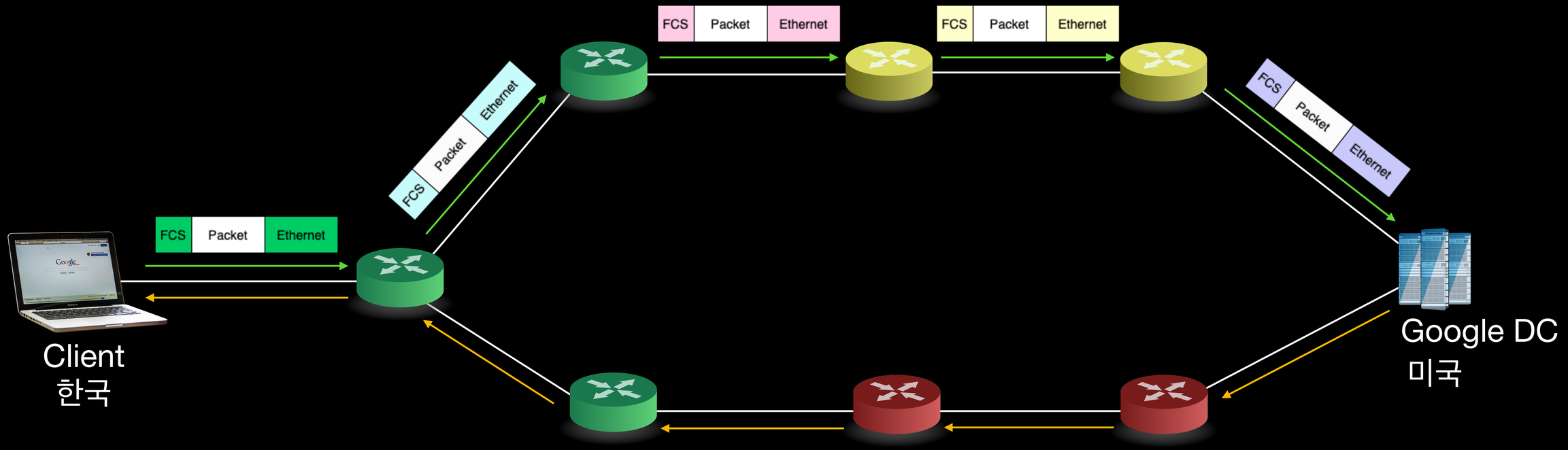



➢ AS ( Autonomous System / 자율시스템 ) : 동일한 라우팅 정책으로 하나의 관리자에 의하여 운영되는 네트워크
즉 한 회사나 단체에서 관리하는 라우터 집단

➢ IGP( Internal Gateway Protocol ) : AS 안에서 동작하는 Routing Protocol ( OSPF, IS-IS, EIGRP 등 )

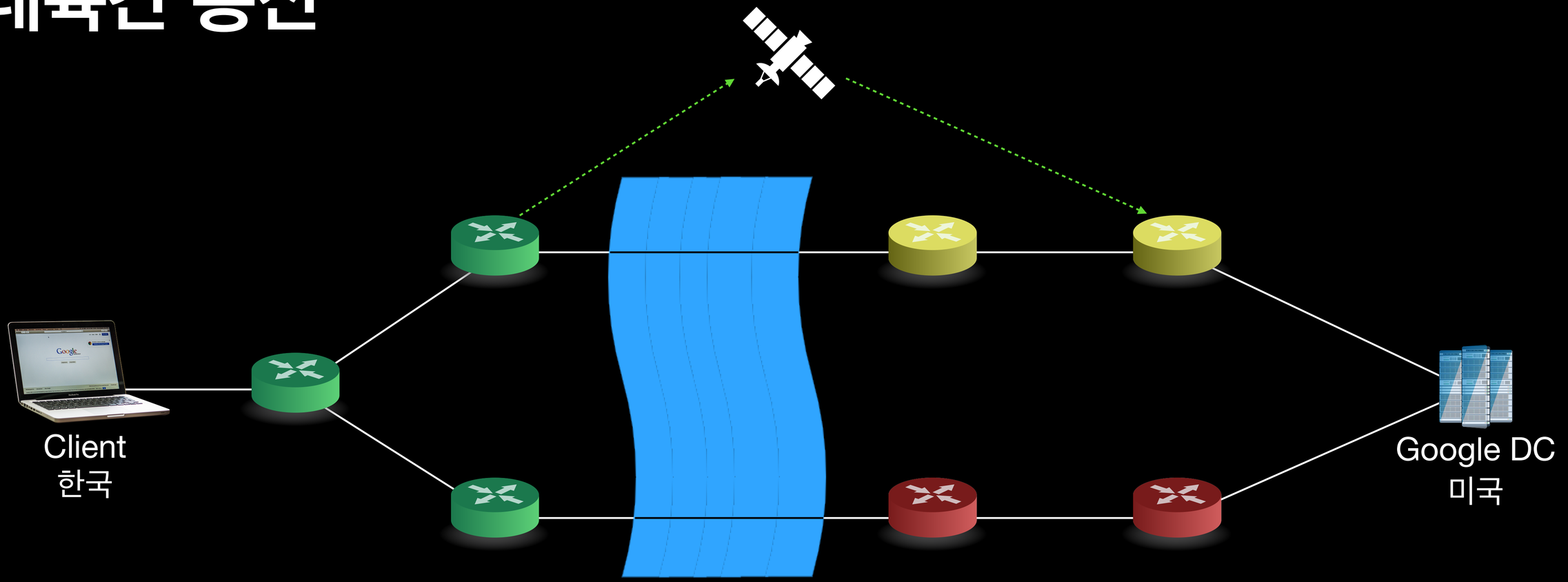
➢ EGP ( Exterior Gateway Protocol ) : AS 간에 동작하는 Routing Protocol ( BGP )

# Routing




- Ethernet Header는 Routing을 할 때 마다 변경됩니다
- 패킷을 보내고 받을 때 지나가는 경로는 동일한 경로가 아닌 경우가 대부분 입니다 ( Asymmetric Routing )
- 경로 상 문제점을 찾기 위해서는 양방향에서 Traceroute를 진행해야 합니다

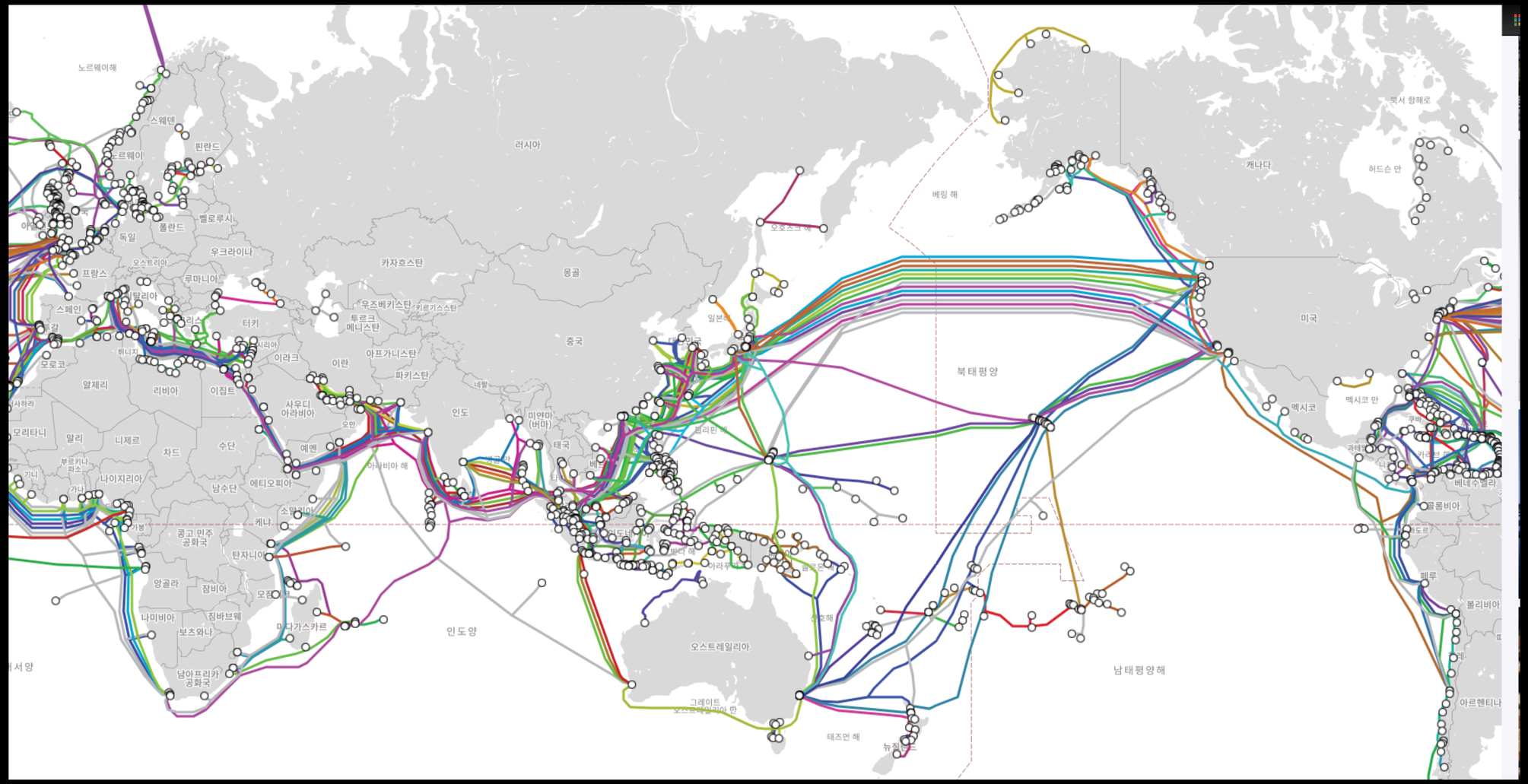
# 대륙간 통신




➢ 대륙간 통신을 하기 위한 방법으로는 지상, 인공위성, 해저케이블등이 있습니다.

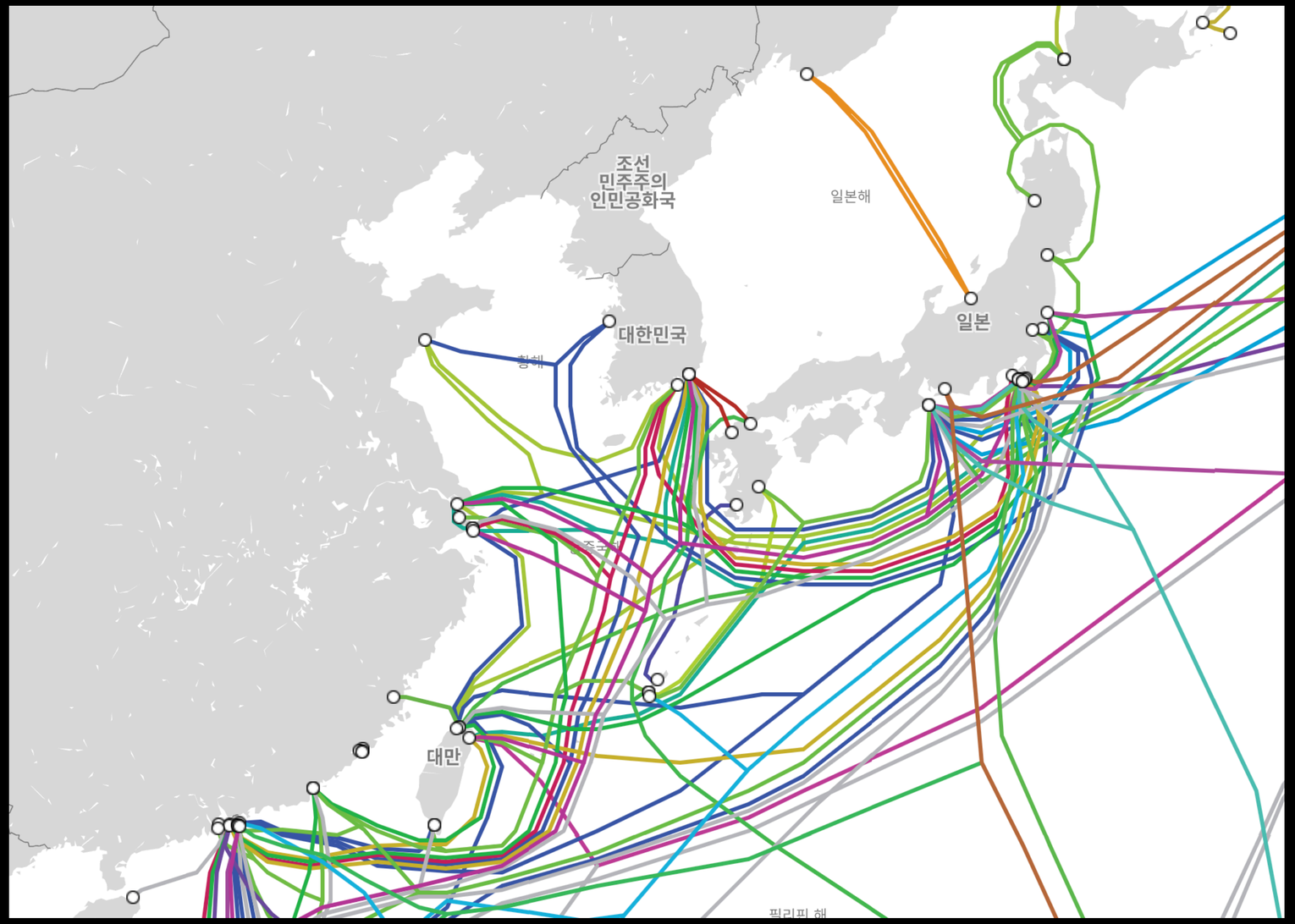
➢ 한국의 경우 지상은 북한으로 인해 불가능하며, 인공위성, 해저케이블을 이용하고 있습니다.

# Submarin Cable




출처 : https://www.submarinecablemap.com/

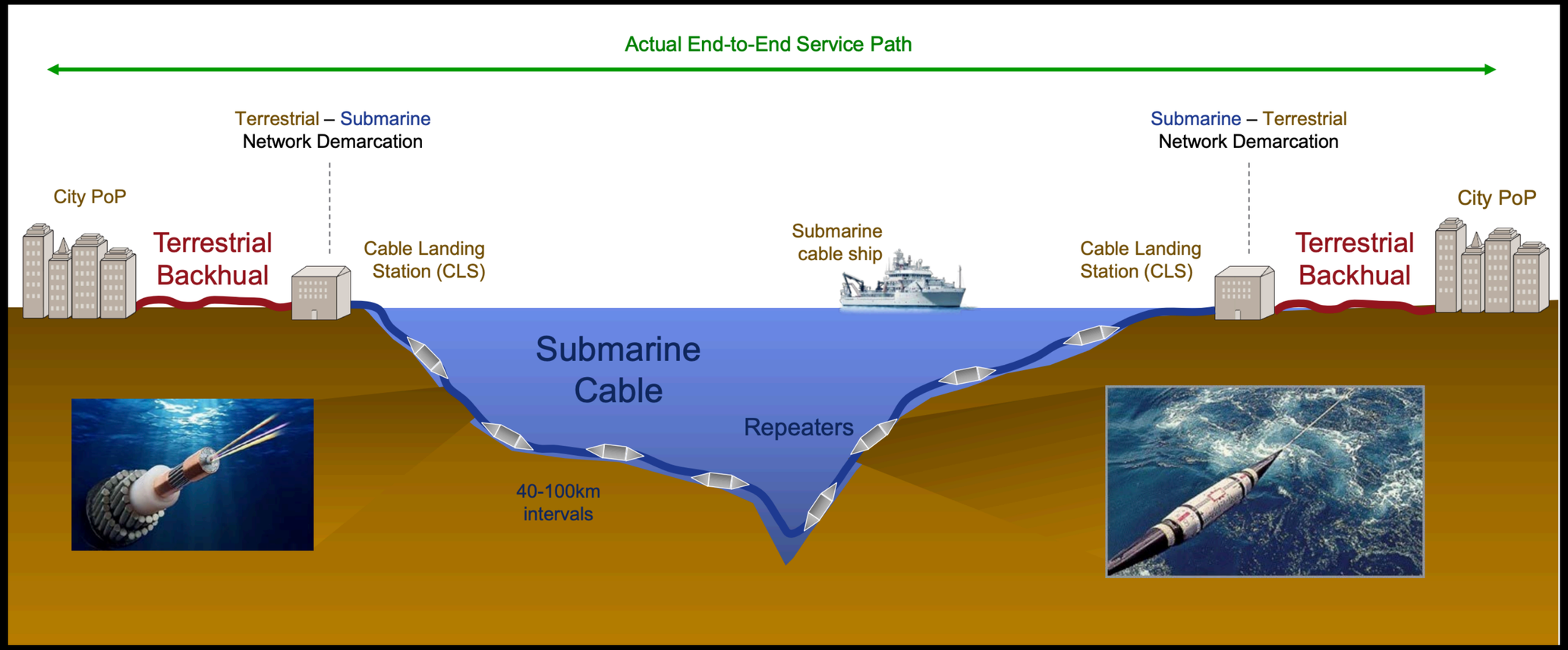
# Submarin Cable




출처 : https://www.submarinecablemap.com/

홀스홀리커

# Cable Landing Station



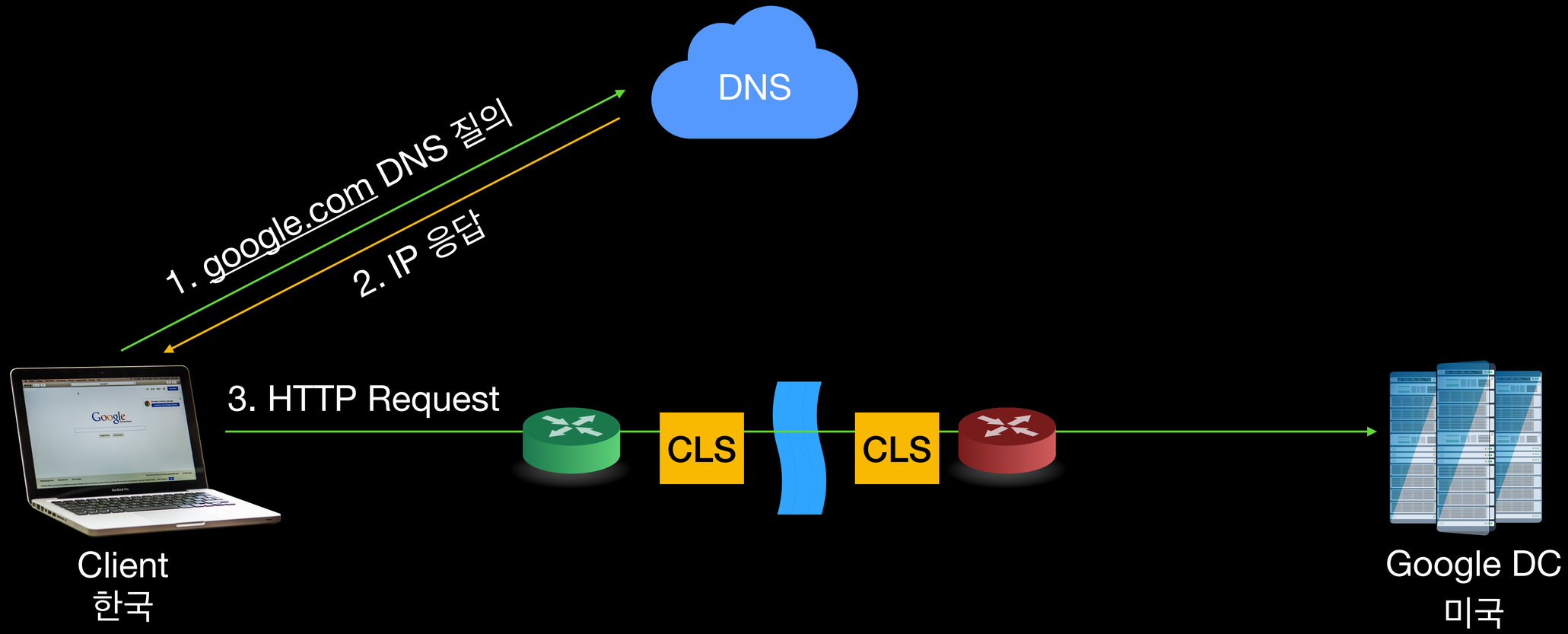

출처 : https://hknog.net/wp-content/uploads/2018/03/07_AnupChangaroth_TheSubmarineNetworkGame_and_the_Implications_of_ShannonsLimits.pdf

➢ Cable Landing Station ( CLS ) : 해저케이블이 지상의 네트워크 망과 연결되기 위한 장소

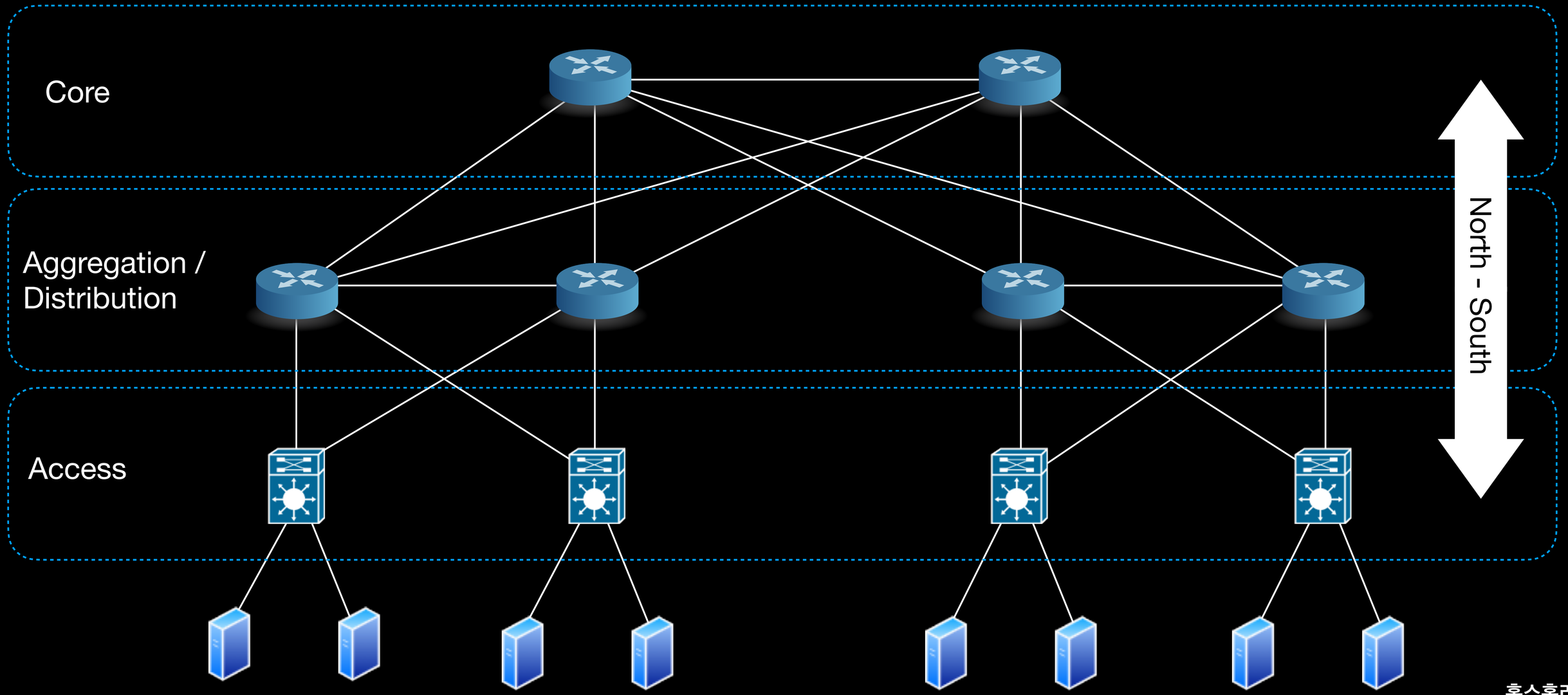
➢ 한국 CLS : 태안, 거제, 부산

홀스홀리커

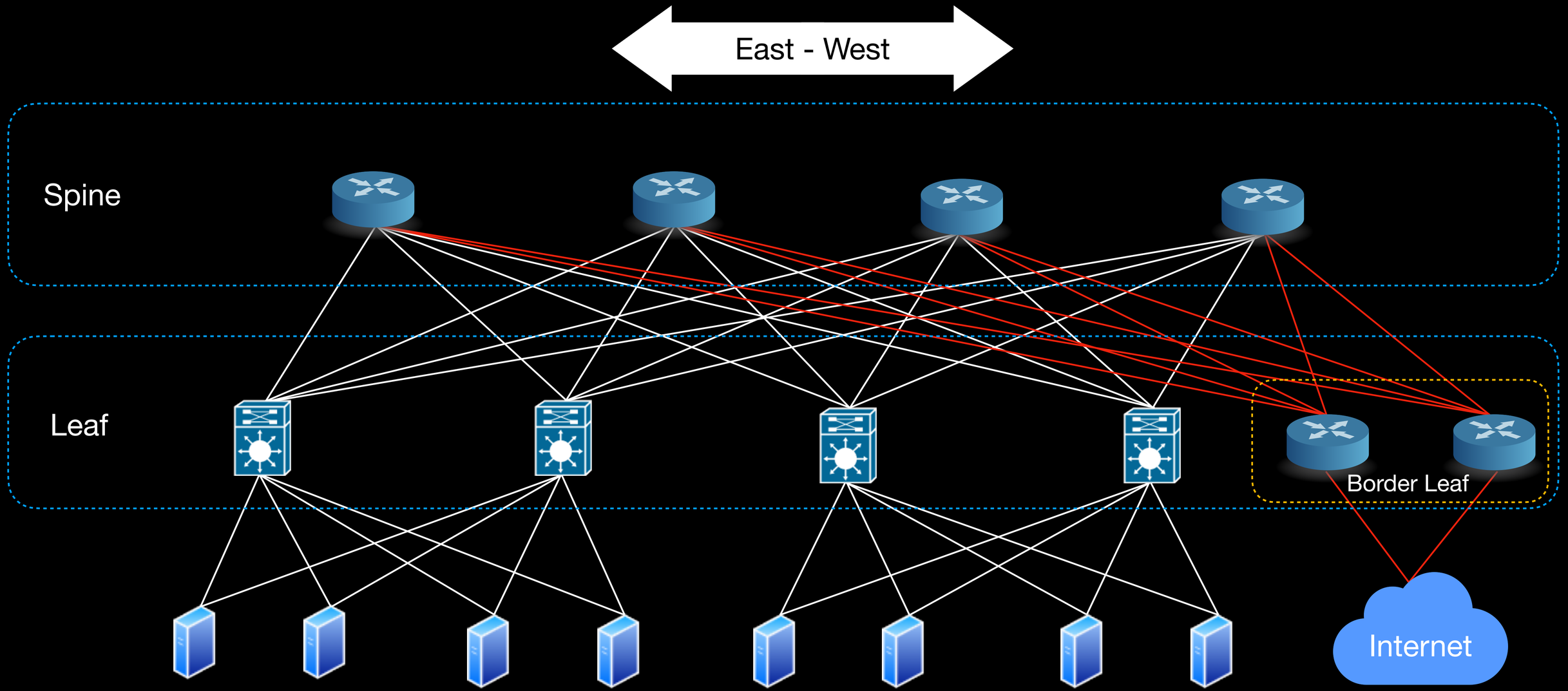
# 물리적 환경 개선


DNS
1. google.com DNS 질의
2. IP 응답
3. HTTP Request
CLS
CLS
Client
한국
Google DC
미국

# Datacenter ( Tranditional 3 Tier )

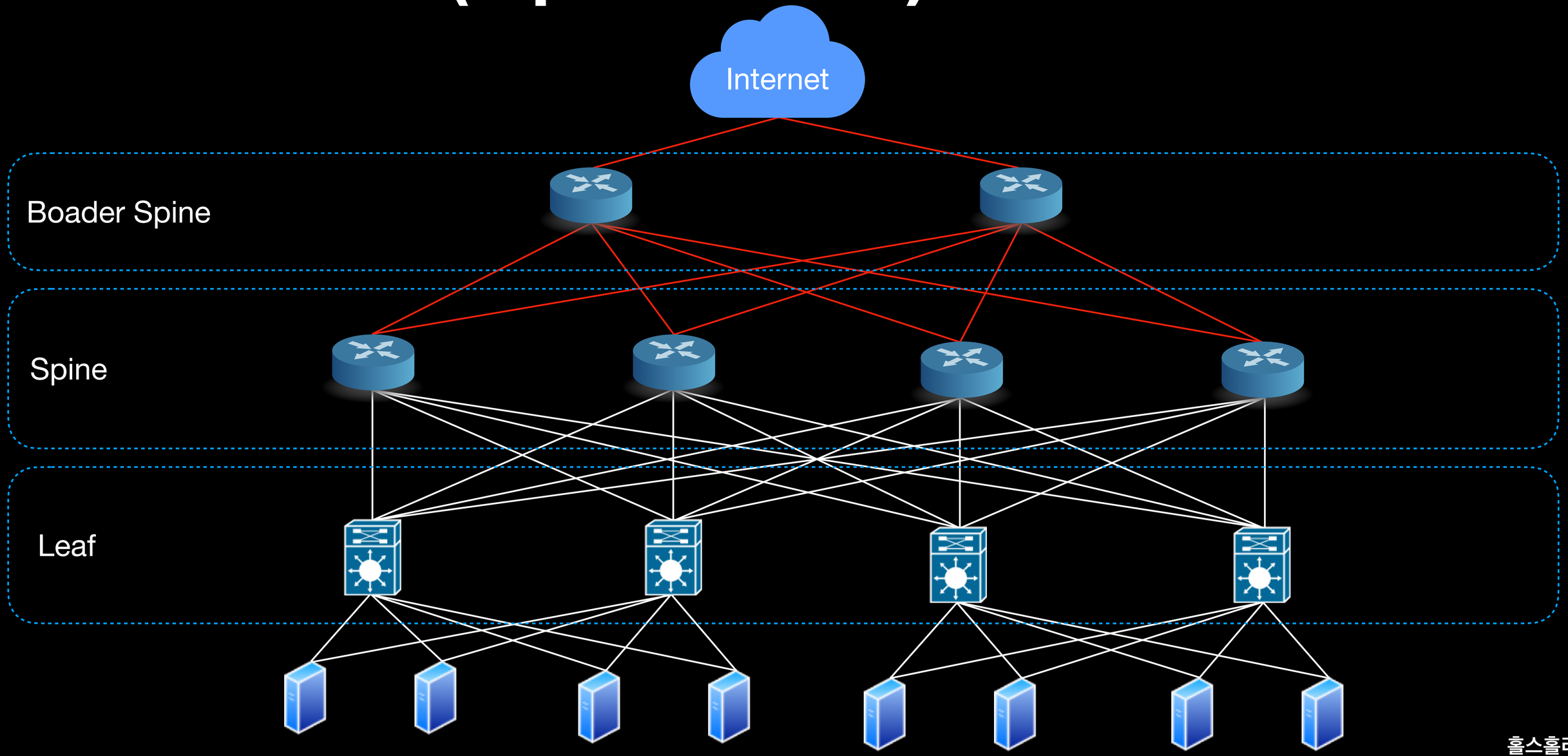


# Datacenter ( Spine - Leaf )



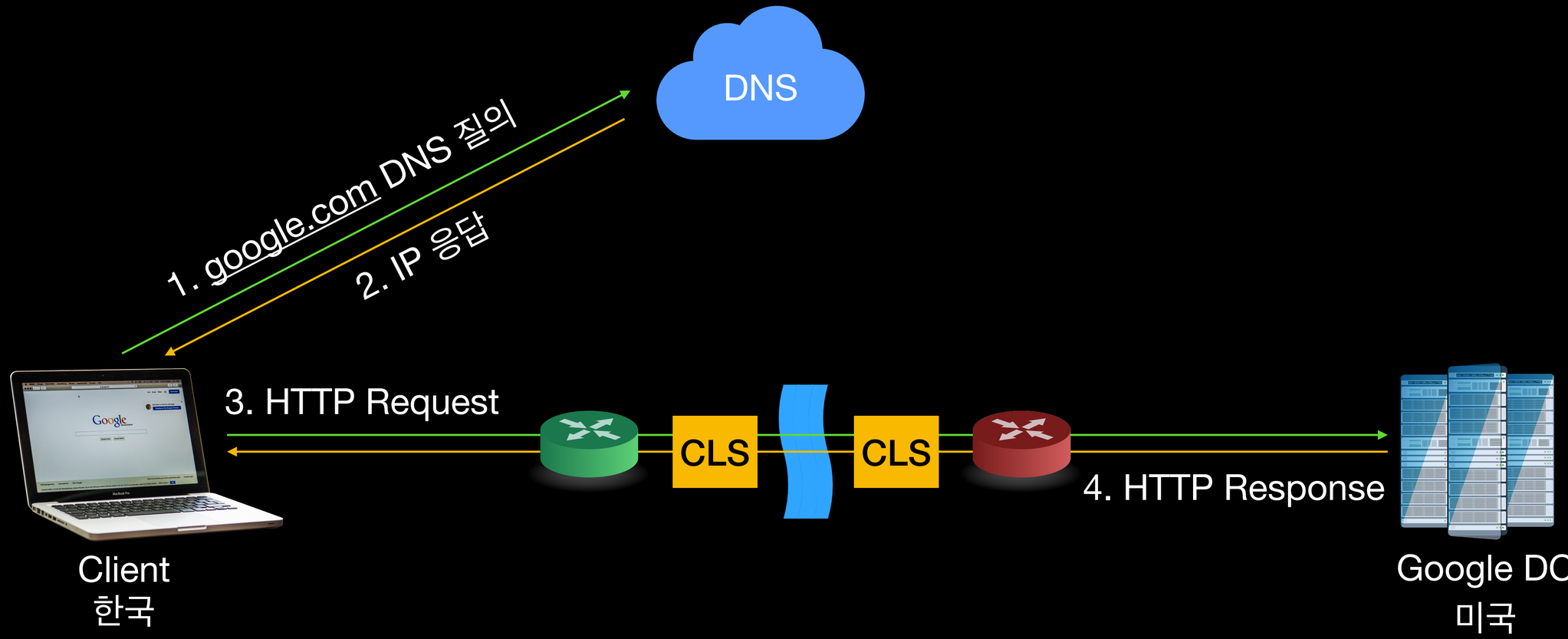

홀스홀리커

# Datacenter ( Spine - Leaf )



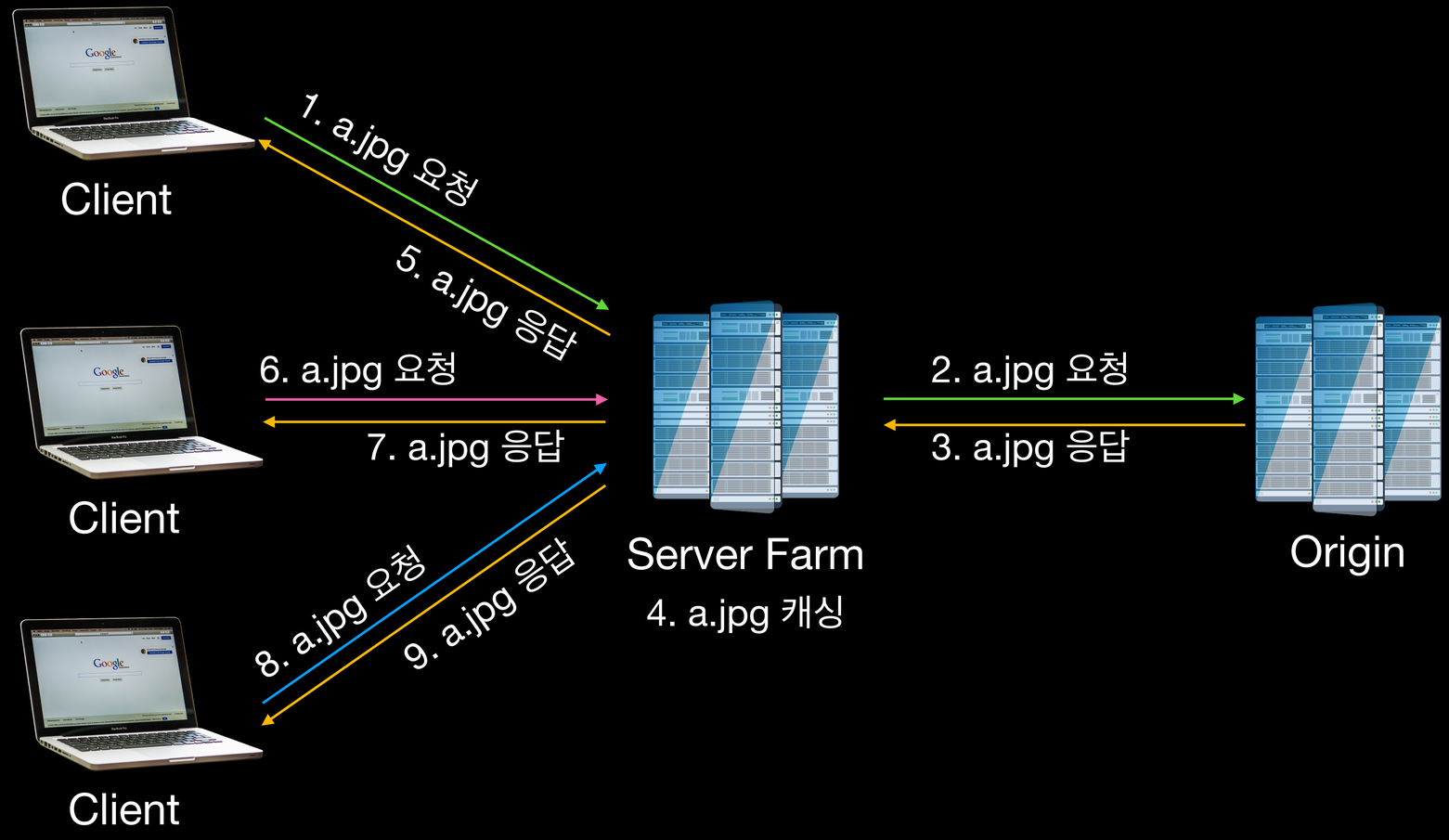

홀스홀리커

# 물리적 환경 개선


DNS
1. google.com DNS 질의
2. IP 응답
3. HTTP Request
CLS
CLS
4. HTTP Response
Client
한국
Google DC
미국

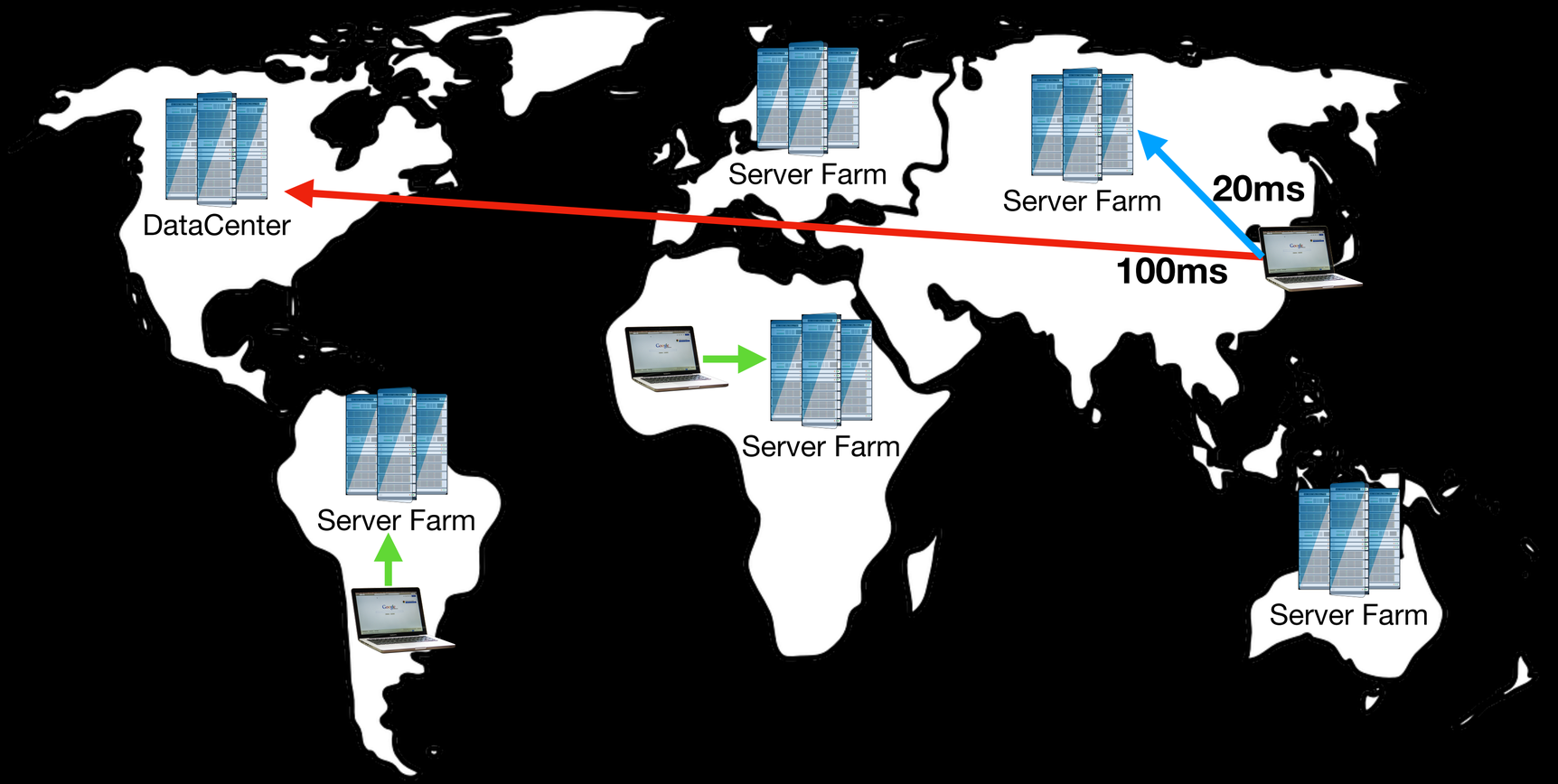
# Content Delivery Network ( CDN )

➢ **Origin Server 의 부하 감소**



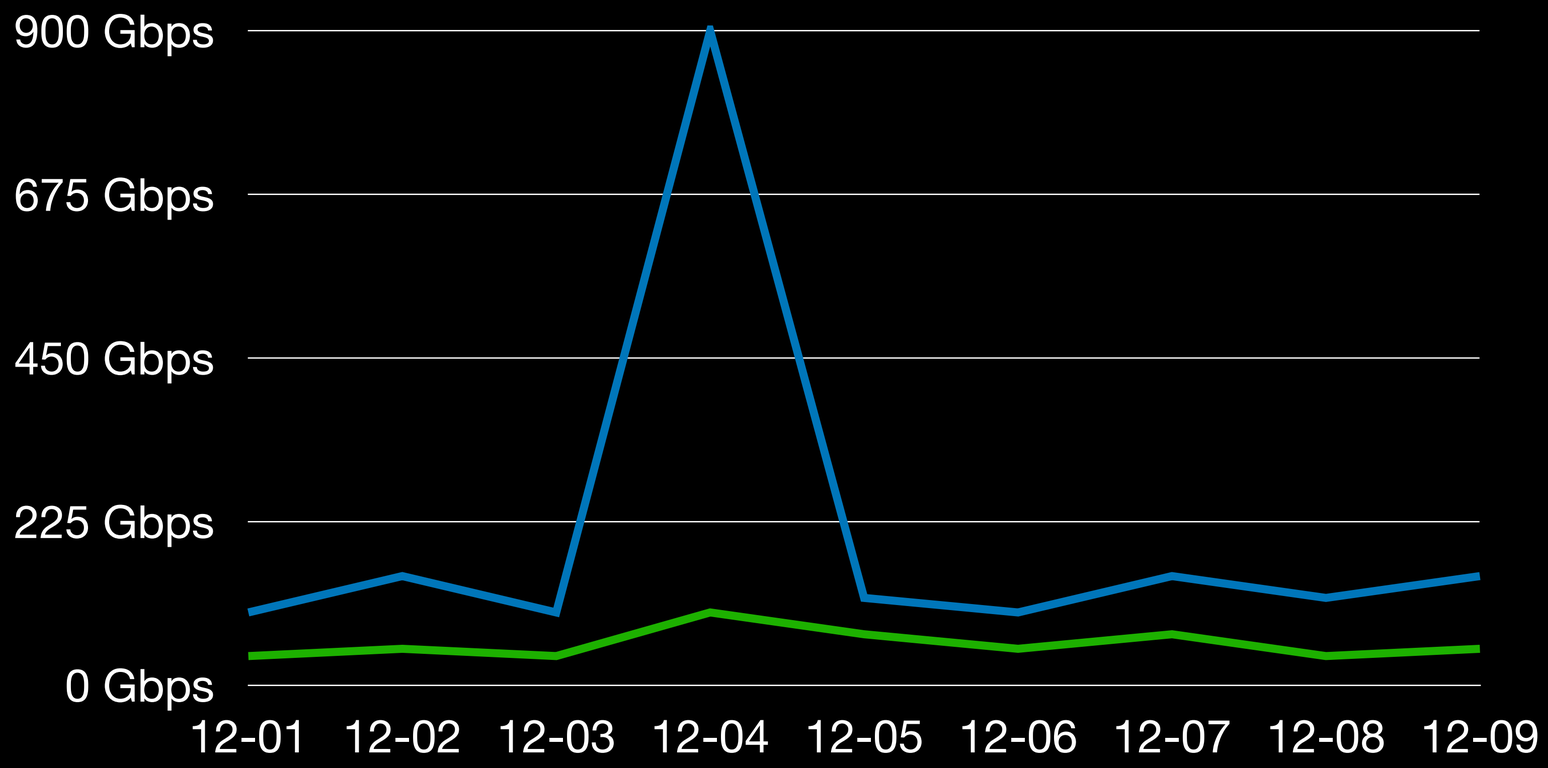
# Content Delivery Network ( CDN )

- Origin Server 의 부하 감소
- **물리적 거리를 줄여서 Latency 감소**



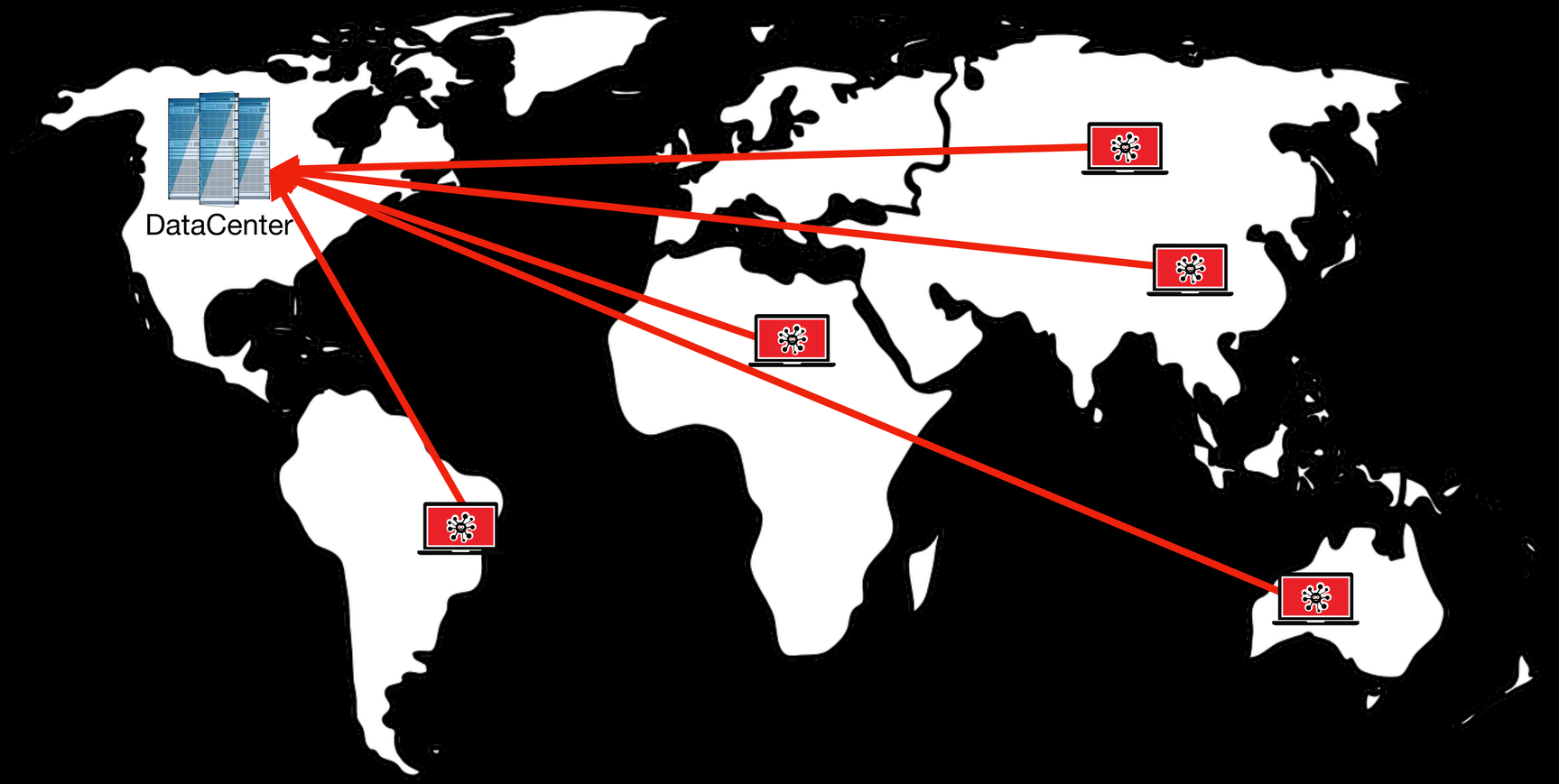
# Content Delivery Network ( CDN )

- Origin Server 의 부하 감소
- 물리적 거리를 줄어서 Latency 감소
- **물리적 투자 비용 감소**



# Content Delivery Network ( CDN )

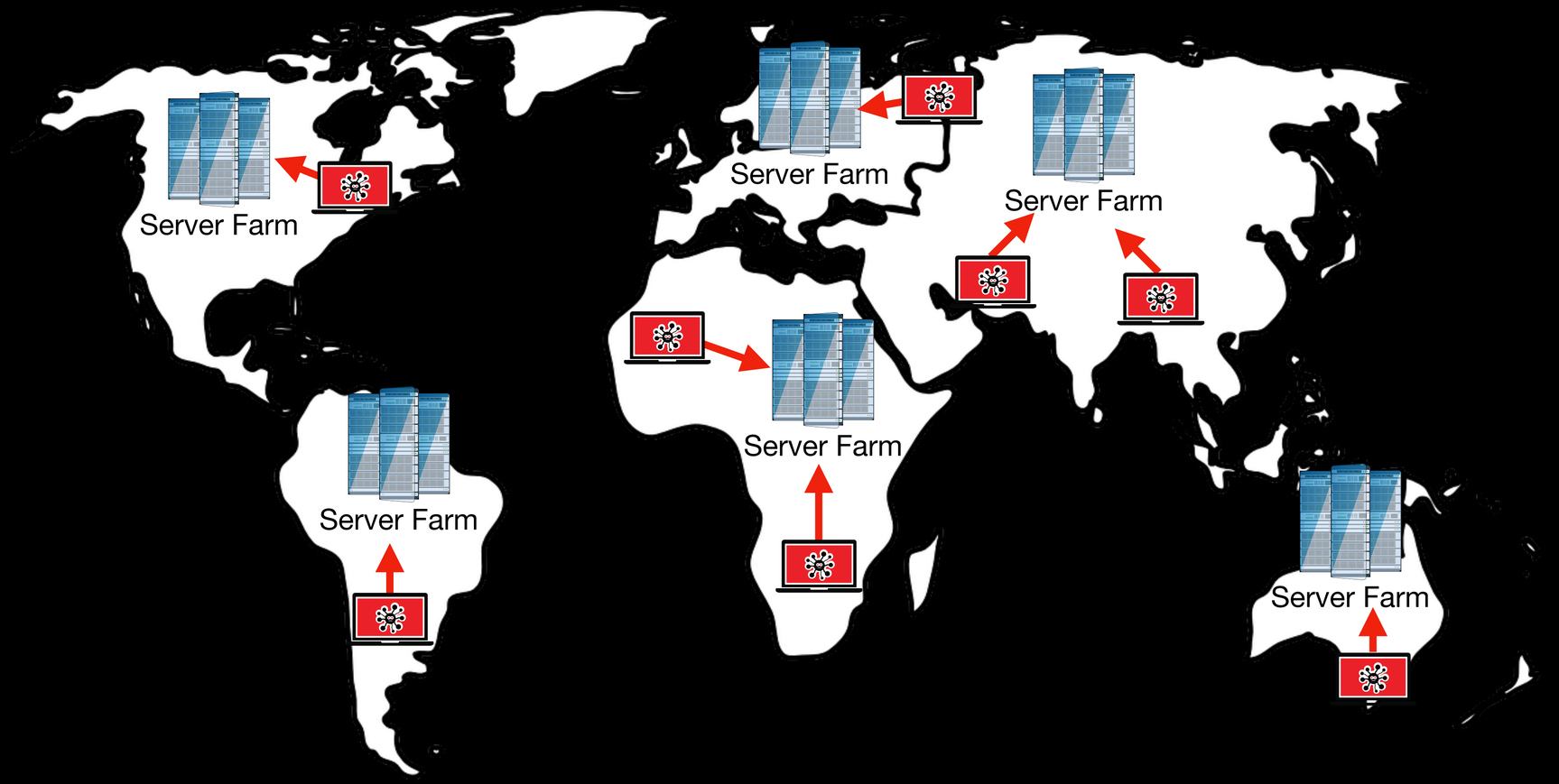
- Origin Server 의 부하 감소
- 물리적 거리를 줄어서 Latency 감소
- 물리적 투자 비용 감소



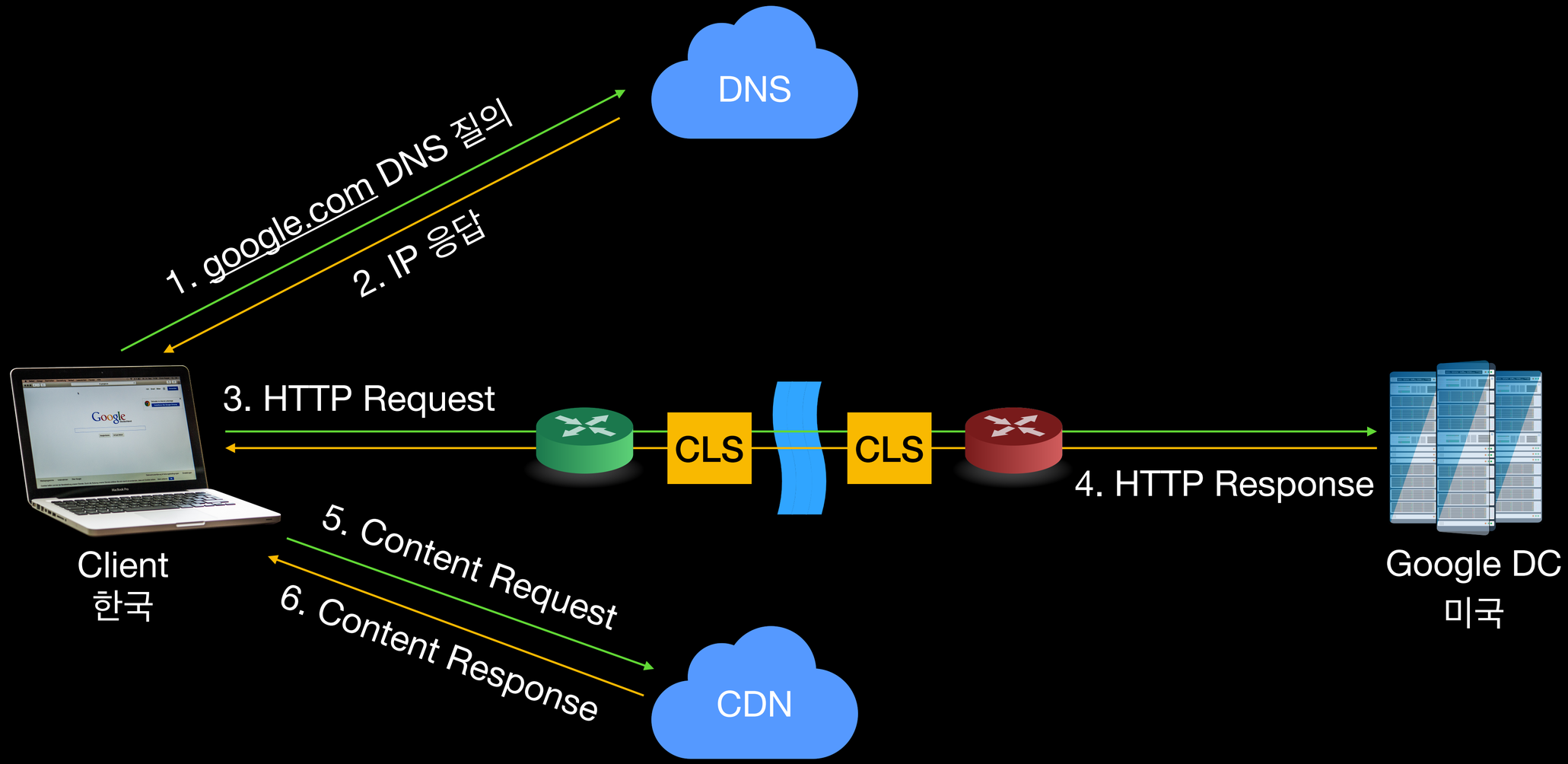

홀스홀리커

# Content Delivery Network ( CDN )

- Origin Server 의 부하 감소
- 물리적 거리를 줄어서 Latency 감소
- 물리적 투자 비용 감소
- **DDoS 방어 효과**



# CDN 개선


DNS
1. google.com DNS 질의
2. IP 응답
3. HTTP Request
CLS
CLS
4. HTTP Response
Client
한국
Google DC
미국
5. Content Request
6. Content Response
CDN

# Q & A