

3. Inter-Server Communication & Peer-to-Peer

Great Technology For Great Games



DK Moon

dkmoon@ifunfactory.com

Inter-Server Comm.



Recap: Database

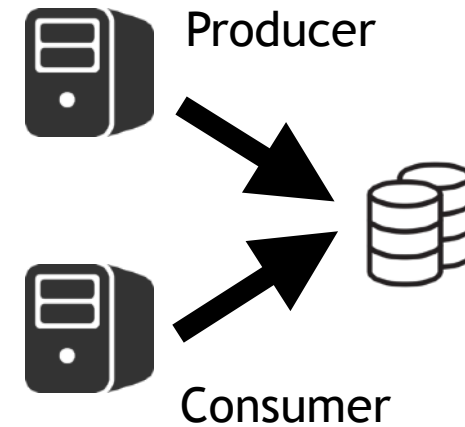
- ✓ CAP 이론?
- ✓ CA 혹은 CP 의 SQL, 그리고 ACID
- ✓ AP 혹은 CP 의 NoSQL, 그리고 BASE
- ✓ 게임서버에서 DB 호출하는 방식
 - RAW SQL
 - Stored Procedure
 - Object Relational Mapping (ORM)



Inter-Server Data Sharing

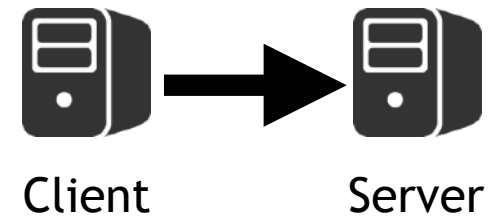
✓ Rendez-vous 방식

- 중간에 공유되는 저장소를 두는 방식
- E.g., shared memory, database



✓ Peer-to-Peer 방식

- 서버간 직접 전달
- E.g., RPC





In-Memory DB (IMDB)

✓ 사용 동기

- DB는 데이터 관리를 위해 여러번의 I/O 작업을 하게 됨
- Disk 작업 대신 훨씬 빠른 속도의 RAM 을 I/O에 활용

✓ Q) ACID 원칙 중 trade-off 하는 것은?



In-Memory DB (IMDB)

✓ 게임 서비스에서 용도

- 서버간 데이터 공유
- 로그인한 유저 데이터의 임시 저장소, 랭킹 임시 저장소

✓ 예시

- Memcached, Redis, H2, etc



Remote Procedure Call (RPC)

✓ 원리

- 좁은 의미: 원격 서버에 있는 Procedure 를 로컬 서버의 함수를 부르는 것처럼 호출
넓은 의미: 원격 서버의 기능 호출
- Client-Server model
- Request-Reply 방식 (즉, 동기 방식)

✓ Q) HTTP 는 RPC 일까?



Remote Procedure Call (RPC)

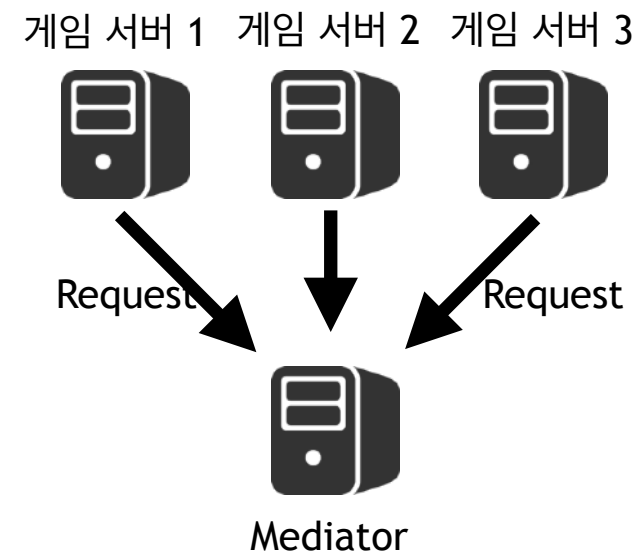
- ✓ Q) IMDB 에 비교했을 때 장단점?
- ✓ 구현 예시: XMLRPC, JSONRPC, RESTful, etc
- ✓ 게임 서비스에서의 용도
 - 다른 서버에 있는 유저에게 원격 작업 (쪽지, 선물 보내기, 길드원간 채팅, ...)

멀티 서버 구축 방법 #1

IMDB 활용



- ✓ IMDB 를 활용한 Data sharing
- ✓ 각 서버는 IMDB 에 polling 작업을 하며, 정보에 기반해 독립적인 의사 결정을 함
- ✓ 장점: 구현의 단순함
- ✓ 단점:

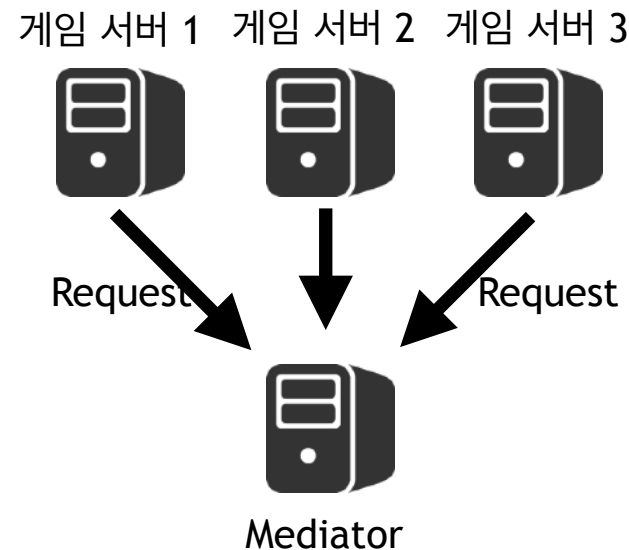


멀티 서버 구축 방법 #1

IMDB 활용



- ✓ IMDB 를 활용한 Data sharing
- ✓ 각 서버는 IMDB 에 polling 작업을 하며, 정보에 기반해 독립적인 의사 결정을 함
- ✓ 장점: 구현의 단순함
- ✓ 단점: Single point of failure
polling overhead

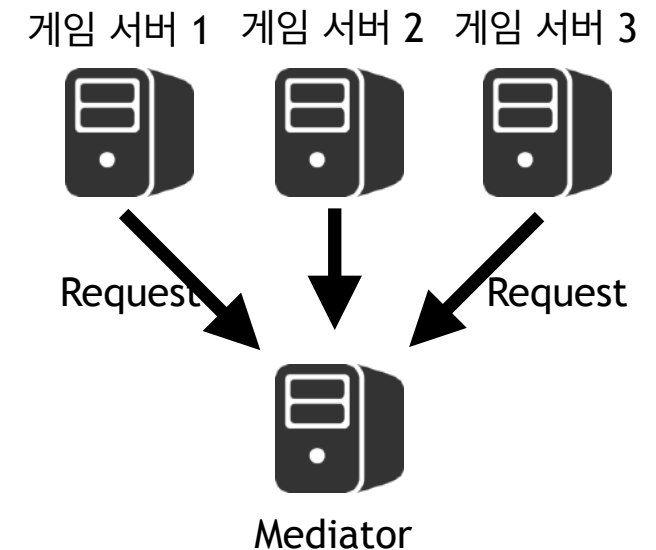


멀티 서버 구축 방법 #2

Centralized Approach



- ✓ 중앙에 의사 결정을 하는 서버 (mediator) 를 운영
- ✓ 각 서버는 mediator 에게 상태 변화에 대한 요청을 보냄
- ✓ Mediator 가 의사 결정 후 다른 서버들에게 통지
- ✓ 장점: 구현의 단순함 (Q: IMDB 에 비해 장점은?)
- ✓ 단점: Single point of failure

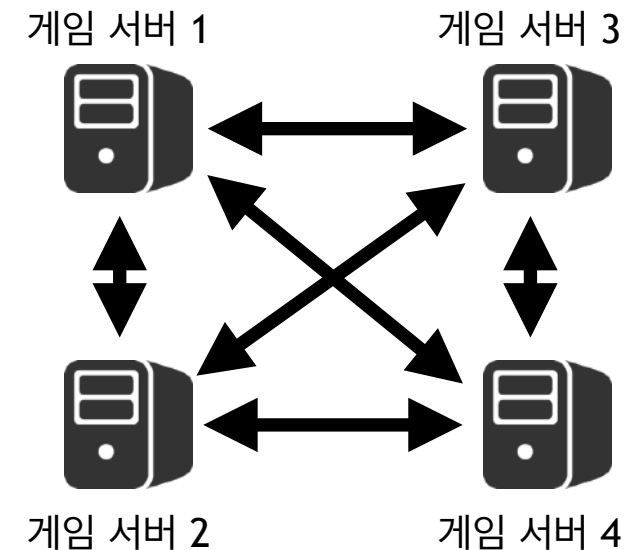


멀티 서버 구축 방법 #3

Distributed Approach



- ✓ 서버는 다른 서버들과 직접 연결을 맺어 통신을 함
- ✓ 서버는 영향 받는 모든 서버에게 상태 변화에 대한 통지를 보냄
- ✓ 장점: Single point of failure 없음
(서버 장애에 대한 영향 최소화)
- ✓ 단점: 설계의 어려움
각 서버들간의 discovery 문제

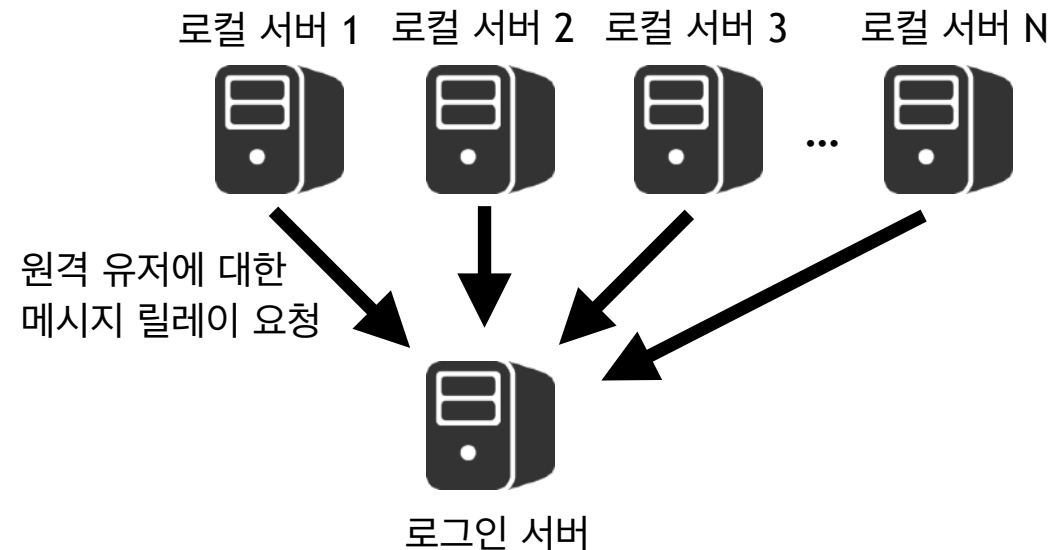


멀티 서버 구축 방법

Case study: 넥슨 바람의 나라



- ✓ Login 서버 - Local 서버 형태로 Centralized Approach 활용
- ✓ Login 서버는 “유저 - 로컬 서버” 맵핑 관리
- ✓ Local 서버는 Zone 을 나눠 유저를 호스팅 하는 형태
- ✓ 단일 채널을 유지 하며 scale-out 하는 구조
- ✓ Q) 장단점?

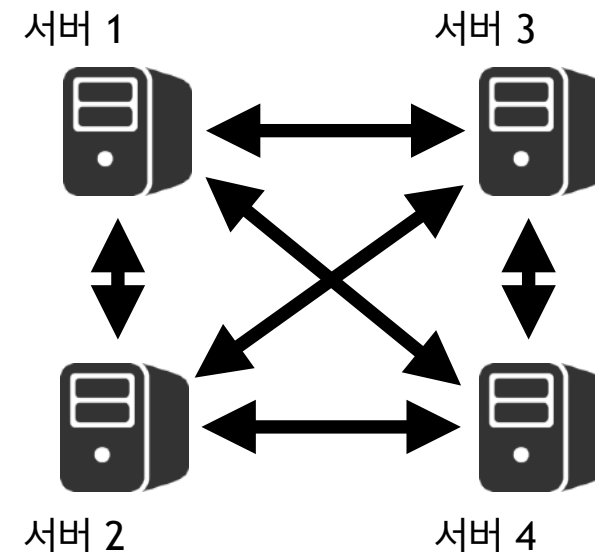


Peer-to-Peer Comm.



Peer-to-Peer (P2P) networking

- ✓ 참여자의 컴퓨팅 리소스 제공
- ✓ 참여자들간 작업의 분배 (이는 참여자들간 상호 일부 작업에 대한 요청을 의미)
- ✓ 각 참여자는 수평적이며 동등한 권리를 가짐

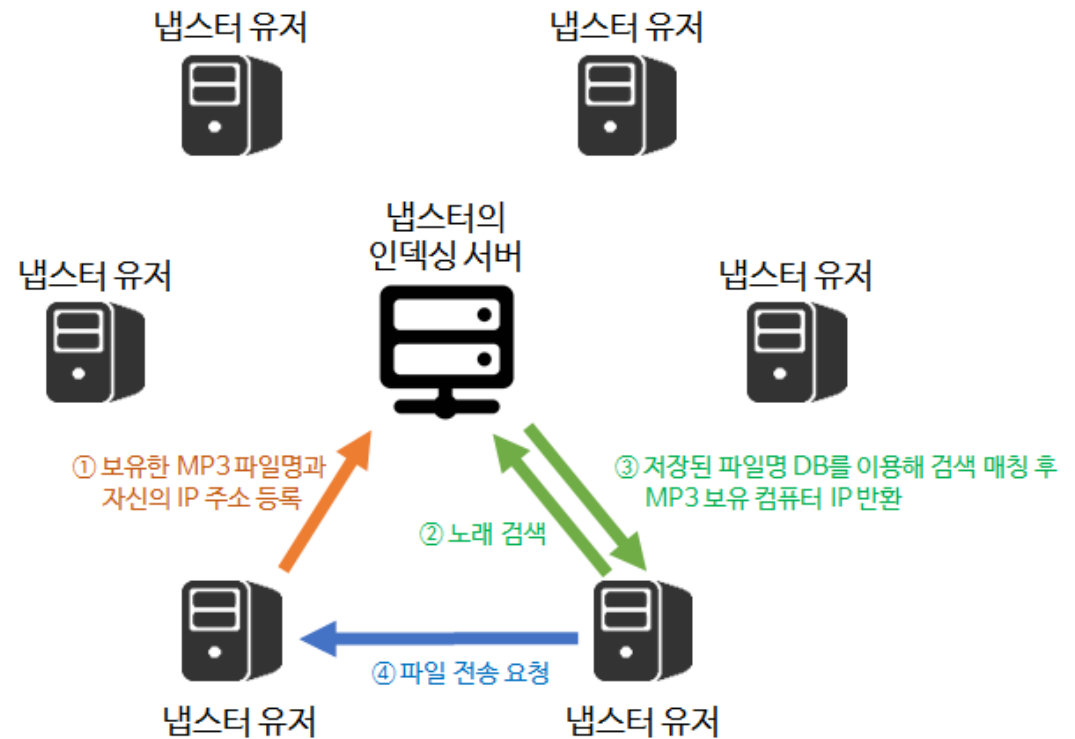


Peer-to-Peer (P2P) networking

Napster



- ✓ P2P 는 90년대 말, 2000년대 초에 파일 공유 영역에서 폭발적 영향력
- ✓ Napster 는 파일 공유 역사에 있어서 중요한 위치를 차지함
- ✓ 중앙의 인덱싱 서버 운영



Peer-to-Peer (P2P) networking

Napster 이후



- ✓ Napster 는 중앙의 인덱싱 서버가 병목이며 법적 분쟁의 이유가 됨
- ✓ gnutella, eDonkey, eMule, Overnet, BitTorrent, Vuze 등은 인덱싱 서버를 계층화하거나 인덱싱 서버 없이 file discovery 를 할 수 있도록 개선
- ✓ 파일 공유에서 시작된 P2P 는 이후 2000년대 초중반 컴퓨터 네트워킹 연구에서 중요한 위치를 차지하게 됨 (e.g., DHT, overlay network)



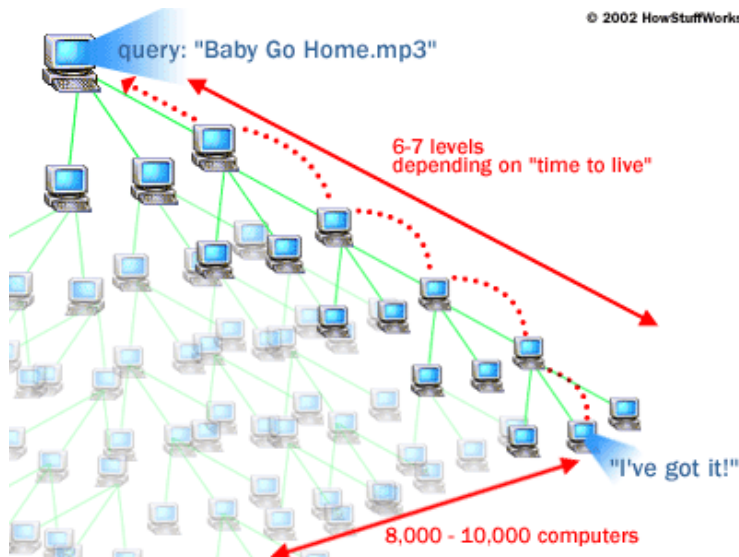
- ✓ Resource discovery
- ✓ Churn handling
- ✓ Fairness enforcement
- ✓ NAT traversal

Peer-to-Peer (P2P) networking

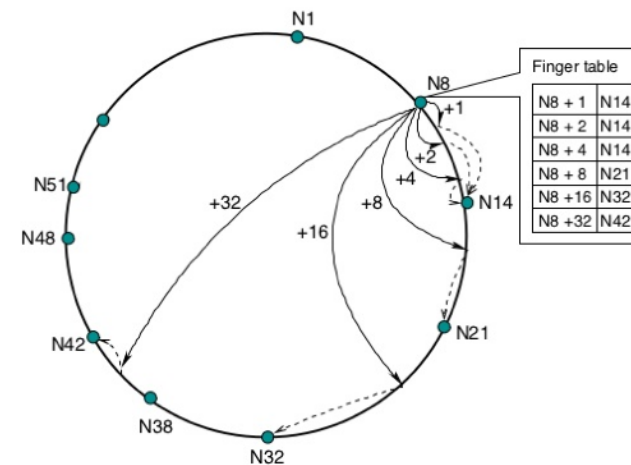
도전 과제 1 - Resource Discovery



- ✓ Centralized approach (e.g., Napster)
- ✓ De-centralized approach
 - Broadcasting (gnutella)
 - Distributed Hash Table (BitTorrent)



Gnutella



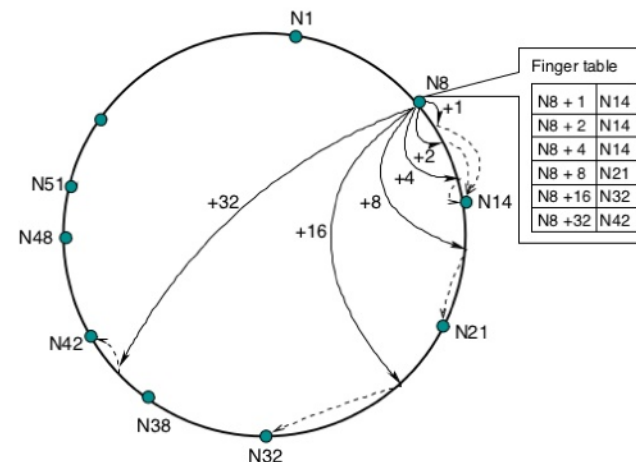
DHT

Peer-to-Peer (P2P) networking

도전 과제 2 - Churn Handling



- ✓ P2P 는 참여자들간 작업 분배에 기초함
- ✓ 따라서 참여자들간 입출입(churn) 이 있을 경우 작업을 재분배해야 함
- ✓ 이 때 모든 참여자들이 동일한 state 로 수렴할 수 있어야 하는 것이 어려움
- ✓ 앞의 DHT 의 경우 Finger Table (이웃 테이블) 을 얼마나 적게 바꾸는지가 관건



Peer-to-Peer (P2P) networking

도전 과제 3 - Fairness Enforcement



- ✓ 참여자는 이기적 선택을 통해 이익을 추구함 (e.g. 파일 받고 바로 공유 해제)
- ✓ 참여자가 지속적으로 정정당한 행동을 하게 강제하는 구조여야 함
- ✓ BitTorrent 의 Tit-for-Tat
 - 더 많은 파일을 공유하면 더 많은 파일을 받을 수 있음



✓ NAT (Network Address Translation)

- IP 주소의 전환 (공인 IP <-> 사설 IP, 사설 IP <-> 사설 IP)
- NAPT (Network Address Port Translation) 도 일반적으로 NAT 로 호칭

✓ NAT traversal

- NAT 을 사이에 두고 통신하는 것
- 대개는 양측 통신 대상 모두 각각 NAT 뒤에 있을 때 문제가 됨



도전 과제 4 - NAT Traversal

✓ NAT traversal 의 주요 문제는 port prediction

✓ TCP hole punching

- NAT 장비의 TCP port preservation 활용 ($ePort == iPort$)
- 두 호스트가 동시에 상대방 ePort 로 연결 시도하고 잘 될 거라고 기도
- 문제: 그런데 이미 다른 내부 호스트가 ePort 를 선점한 경우 $ePort \neq iPort$

✓ UDP hole punching

- UDP 는 연결 개념이 없다는 점을 활용 (대화 상대를 바꿔치기함)
- 외부 STUN (Session Traversal Utilities for NAT) 서버에 패킷을 보내고, 상대방은 STUN 이 받은 ePort 를 추출해서
이후 STUN 대신 자신이 ePort 로 패킷을 보내기 시작



도전 과제 4 - NAT Traversal

✓ NAT 는 $(iAddr, iPort) \rightarrow (eAddr, ePort) \rightarrow (hAddr, hPort)$ 형태로 통신

✓ NAT 구현 방식과 traversal 가능 여부

- Full-cone: $(iAddr, iPort) \rightarrow (eAddr, ePort)$ 는 고정
hAddr 외 아무 외부 기계나 $(eAddr, ePort)$ 을 통해 통신 가능
- Address-restricted-cone: $(iAddr, iPort) \rightarrow (eAddr, ePort)$ 는 고정
hAddr 은 $(eAddr, ePort)$ 를 통해 통신 가능
- Port-restricted-cone: $(iAddr, iPort) \rightarrow (eAddr, ePort)$ 는 고정
 $(hAddr, hPort)$ 만이 $(eAddr, ePort)$ 로 통신 가능
- Symmetric: $(hAddr, hPort)$ 별로 $(iAddr, iPort) \rightarrow (eAddr, ePort)$ 변경
따라서 $(hAddr, hPort)$ 만이 $(eAddr, ePort)$ 로 통신 가능



도전 과제 4 - NAT Traversal

✓ NAT 은 구현의 표준이 없음.

- 가정용 공유기 => iPort-ePort 맵핑 알고리즘이 단순
- Carrier Grade NAT (대부분의 통신사 무선 망) => 맵핑 알고리즘이 복잡

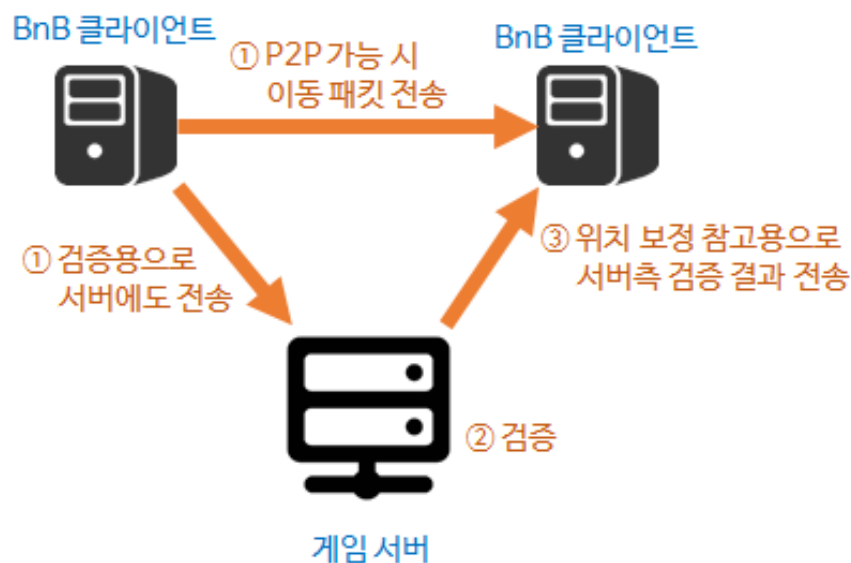
✓ 따라서 어떤 경우든 동작하는 NAT traversal 은 존재하지 않음

Peer-to-Peer (P2P) networking

게임에서의 도입 배경



- ✓ CA BnB : 상업적으로 성공을 거둔 P2P 기반 게임
- ✓ 도입 이유: 사용자간 움직임의 빠른 동기화
Client-Server 보다 Client-Client 가 더 빠를 수 있다는 개연성 활용
- ✓ 반드시 더 빠르다는 보장이 없으므로 BnB 는 C2S 채널과 P2P 채널을 모두 유지



Peer-to-Peer (P2P) networking

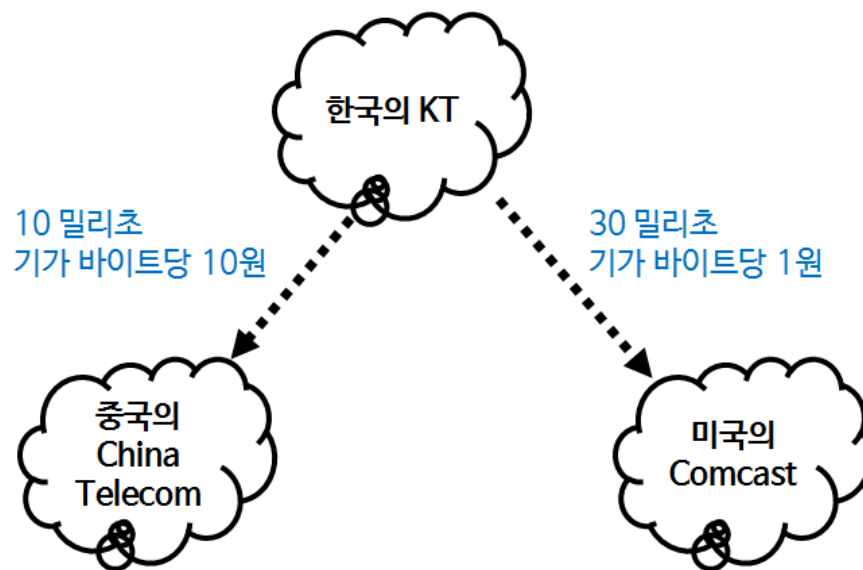


P2P 가 왜 더 빠를 수 있는가?

✓ 인터넷상에서의 라우팅은 “정책 기반임”

- 망 운영자 입장에서 정부 규제나 가격 등의 이유로 경로를 결정함

✓ Q) 아래 상황에서 어떤 경로를 택할 것인가?



Peer-to-Peer (P2P) networking



유선 망과 모바일 망의 차이

✓ 유선망: 안정적인 연결에 따라 불안정한 상황을 예외처리 할 수 있었음.

- Resource Discovery: 게임 서버가 indexing 서버 역할
- Churn handling: 연결이 불안정하면 바로 게임 중단
- Fairness enforcement: 연결이 불안정하면 cheating 으로 의심 가능
- NAT traversal: 가정용 공유기는 대부분 간단한 알고리즘 사용

Peer-to-Peer (P2P) networking



유선 망과 모바일 망의 차이

✓ 무선망: 불안정한 jitter 가 기본 특성이므로 예외 처리 불가능

- Resource Discovery: 게임 서버가 indexing 서버 역할
- Churn handling: 연결이 불안정하다고 바로 중단 불가능
- Fairness enforcement: ???
- NAT traversal: 통신사 장비는 NAT traversal 이 되지 않음 (3G, LTE)

Peer-to-Peer (P2P) networking

무선 통신사가 NAT traversal을 허용할까?

- ✓ 무선 통신을 위한 주파수는 희귀 자원임
- ✓ 그 때문에 통신사가 네트워크 안에 서버가 동작하는 것을 허용할 가능성은 낮음.

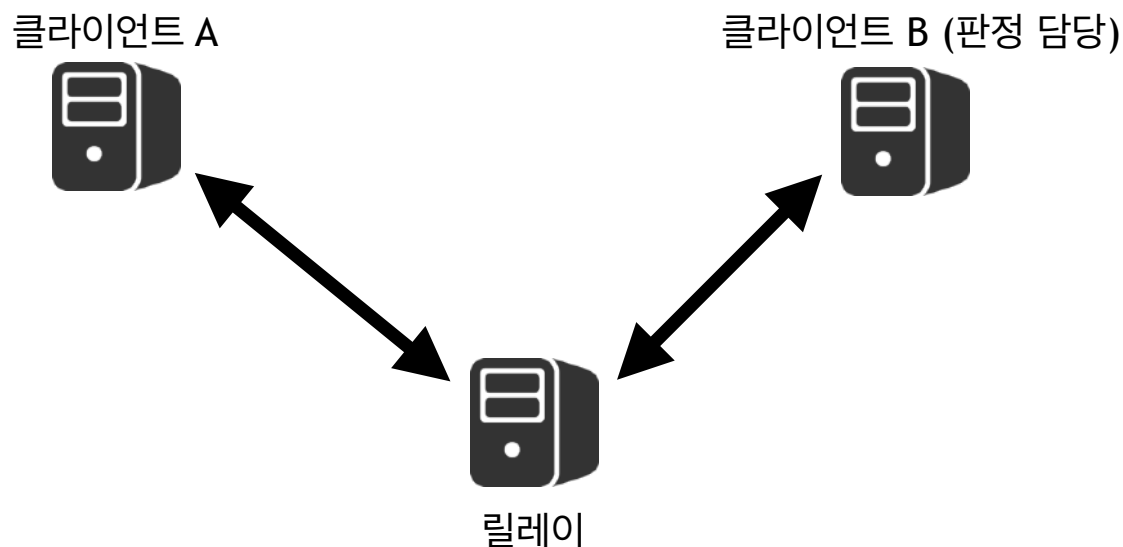


Peer-to-Peer (P2P) networking

Relay 를 이용한 방식



- ✓ Latency 에서 P2P 의 이득이 사라짐
- ✓ Q) 아래 그림에서 A 와 B 의 동작이 판정을 받을때까지 hop 차이는?
B 가 WiFi 에 연결되어있고 AP 의 랜선을 뽑는다면?



THANKS!

Great Technology For Great Games, **iFunFactory**



DK Moon



dkmoon@ifunfactory.com



www.ifunfactory.com

