



NFS 분산 파일 시스템 접근관리 및 아키텍처



학습 목표

- NFS 서버 접근 관리
 - Mount 옵션을 이용한 클라이언트 접근권한 부여
 - ro
 - rw
 - root =
 - ro=access_list
 - rw= access_list
 - access=n
- NFS 아키텍처
 - 읽기 요구시
 - 쓰기 요구시
 - 재전송 요구시

NFS 서버 접근 관리

- 개념
 - NFS 서버가 공유 된 자원을 제공할 때 사용자나 그룹에게 적절한 접근 권한을 지정하고 관리하는 것을 의미한다.
- 특징
 - 모든 공유자원에 대한 기본 설정값은 읽고 쓸수 있는 권한이 부여되어 있으며
 - 단 CD-ROM 에 대한 기본 설정값은 읽기 권한이 부여되어 있다.
 - 접근 결정은 사용자나 소유권자의 UID 를 기반으로 결정한다.

NFS 서버 접근 관리

■ 접근 권한 종류

ro	클라이언트에게 서버의 공유자원에 대해 읽기 만을 승인
rw	클라이언트에게 서버의 공유자원에 대해 읽고쓰기 승인
root=client	공유 자원에 대해 슈퍼 유저 사용자 권한을 갖은 사람 지정시
ro= access_list	Access_list에 지정된 사용자에게 대해 읽기만 승인
rw= access_list	Access_list 에 지정된 사용자에게 대해 읽고 쓰기 승인
rw=사용자 계정1:사용자 계정 2	여러명의 사용자를 지정하기 위해 “:” 사용
rw=@network	/etc/network 파일에 등록된 네트워크 주소를 기준으로 접근권한이 결정 (예 : @192.168.0)
access_list=.domain	DNS 내 도메인 이름을 기준으로 접근승인 여부가 결정
access_list= netgroup_name	NIS 내 지정된 네트워크 그룹을 기준으로 승인 여부가 결정
access=n	일명 사용자에게 대한 EUID 지정함으로써 승인 여부가 결정 기본 설정은 EUID=60001 이며 만일 anon=-1 설정시에는 익명 사용자에게 대한 접근이 거부됨을 의미

NFS 서버 접근 관리 적용예제

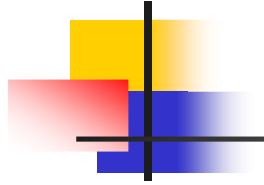
- 상황
 - 지정한 디렉토리에 대해 읽기 권한만을 명령 프롬프트상에서 부여 하고 싶을때.
- 예제

```
# share -F nfs -o ro 공유 디렉토리 지정
```
- 상황
 - 지정된 디렉토리에 대해 클라이언트 A에게는 읽고 쓰기권한을 나머지는 모두다 읽기 권한만을 부여 하는 예.
- 예제

```
# share -F nfs -o ro, rw=clientA 공유 디렉토리 지정
```
- 상황
 - 지정한 디렉토리에 대해 루트권한을 가지고 있는 사용자중 클라이언트 B에게만 디렉토리 사용권한을 부여하고 싶을때.
- 예제

```
# share -F nfs -o root=clientB 공유 디렉토리 지정
```
- 상황
 - 공유자원에 대해 EUID가 0 인것만 지정된 디렉토리에 접근을 허용하고 싶을때.
- 예제

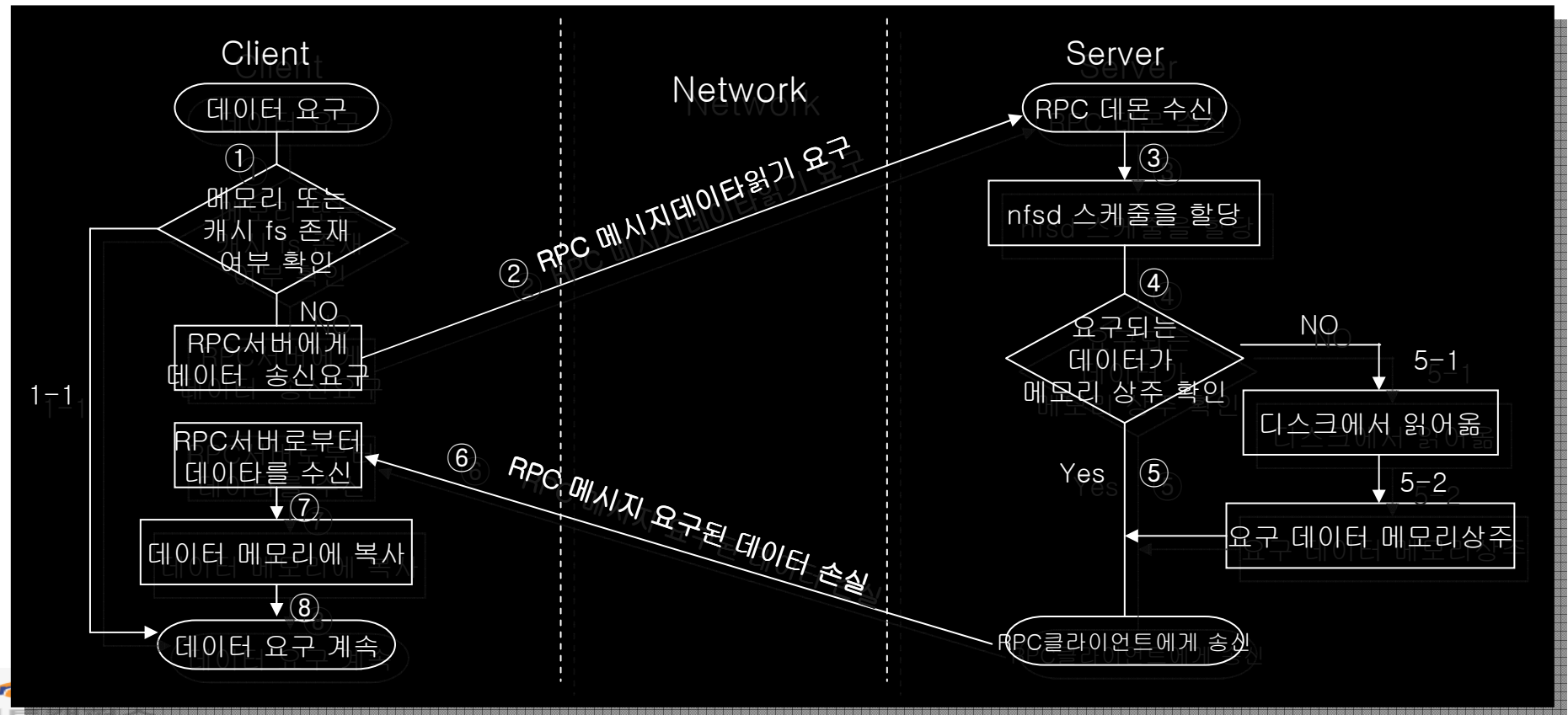
```
# share -F nfs -o anon=0 공유 디렉토리 지정
```



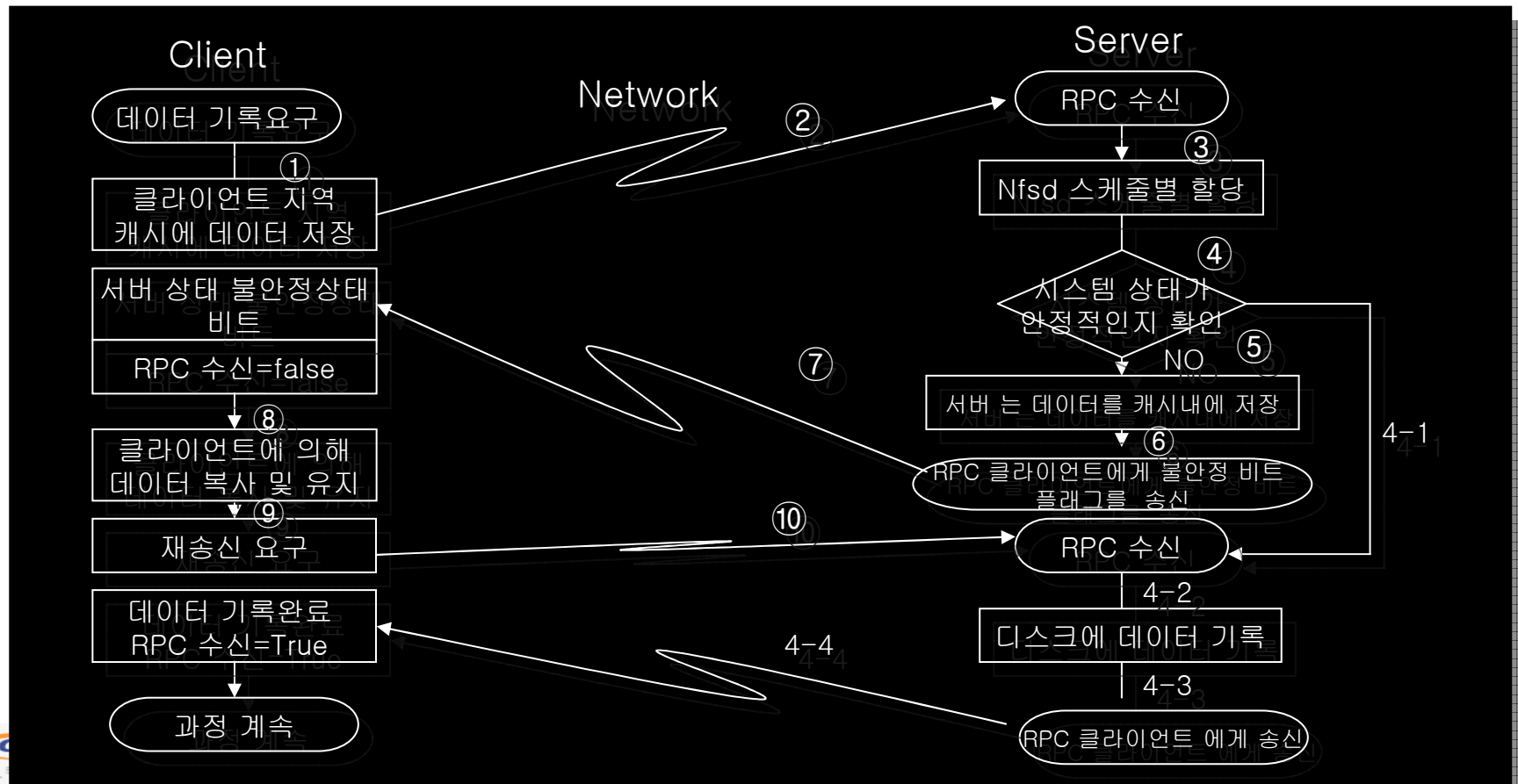
NFS 아키텍처

- nfs 서버에 클라이언트가 읽기 요구를 하였을 경우 호출절차
- nfs 서버에 클라이언트가 쓰기 요구를 하였을 경우 호출절차
- nfs 서버에 클라이언트가 재전송 요구를 하였을 경우 호출절차

NFS 아키텍처 – 읽기 요구

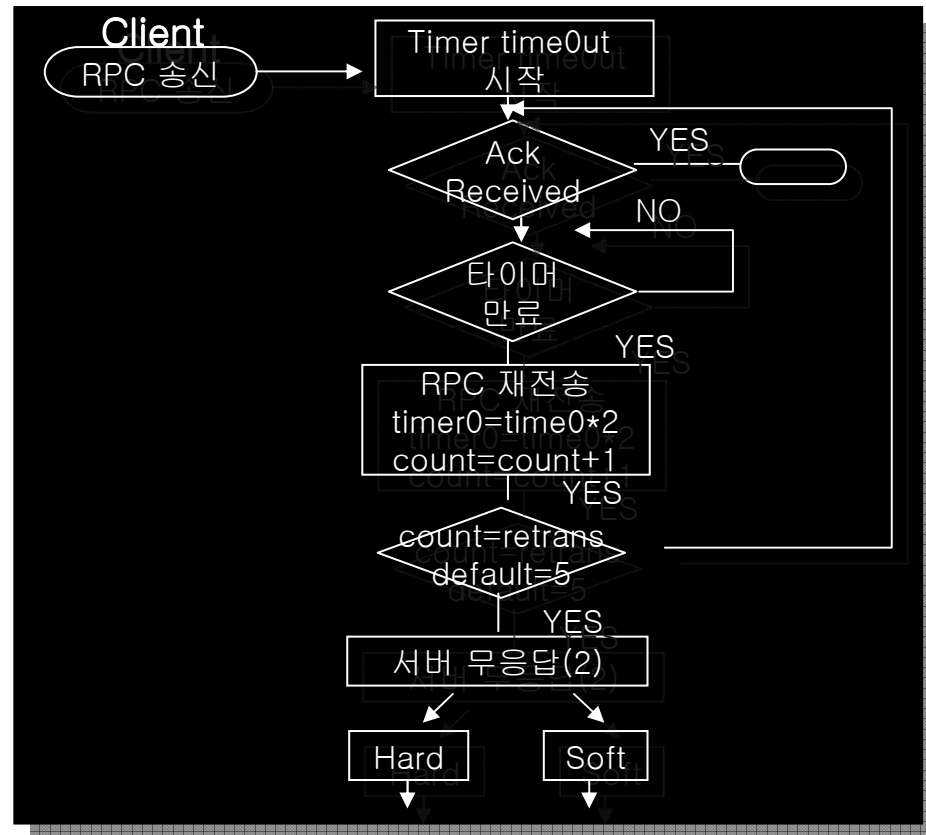


NFS 아키텍처 – 쓰기 요구



NFS 아키텍처 – NFS Retransmission

- 서버에게 메시지 송신후 클라이언트는 time out값을 설정한 시간만큼 응답(Acknowledgement)을 기다림.
- Ack 응답이 오지 않으면 클라이언트는 기존에 설정한 time out 값을 2 배($time0 \times 2$)와 counter값을 1씩 증가 시킨후 재전송을 요구한다.
- 재 전송 패킷수가 retrans 에서 규정한 값과 일치하고 마운트 옵션을 soft 로 설정 하였다면 error 메시지 보여준후 종료 할것이며 마운트 옵션을 Hard 로 설정시 경고후 계속하여 재송신 요구를 할것이다.





요점 정리

- NFS 접근권리는 보안상에 치명적인 영향을 미칠 수 있으므로 신중히 고려해야 한다.
 - ro, rw, root=client etc 등등
- NFS 아키텍처의 이해는 곧바로 NFS 서버의 취약점을 이해하는데 도움이 된다.