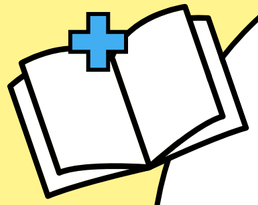
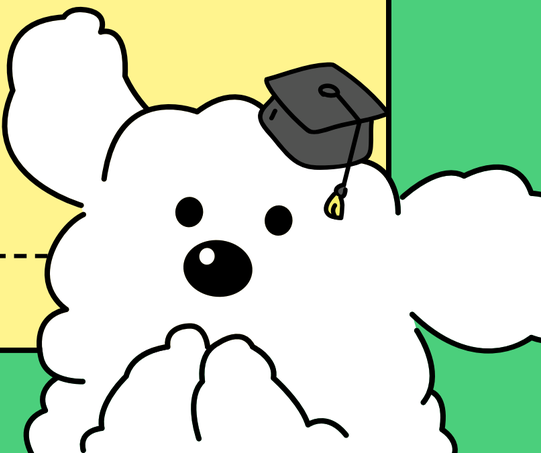
MIRI ACADEMY



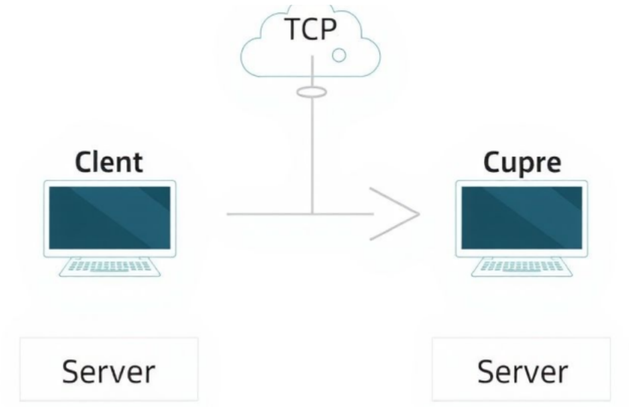
3-way Handshake

TCP 연결



목차

TCP 연결의 기본 개념



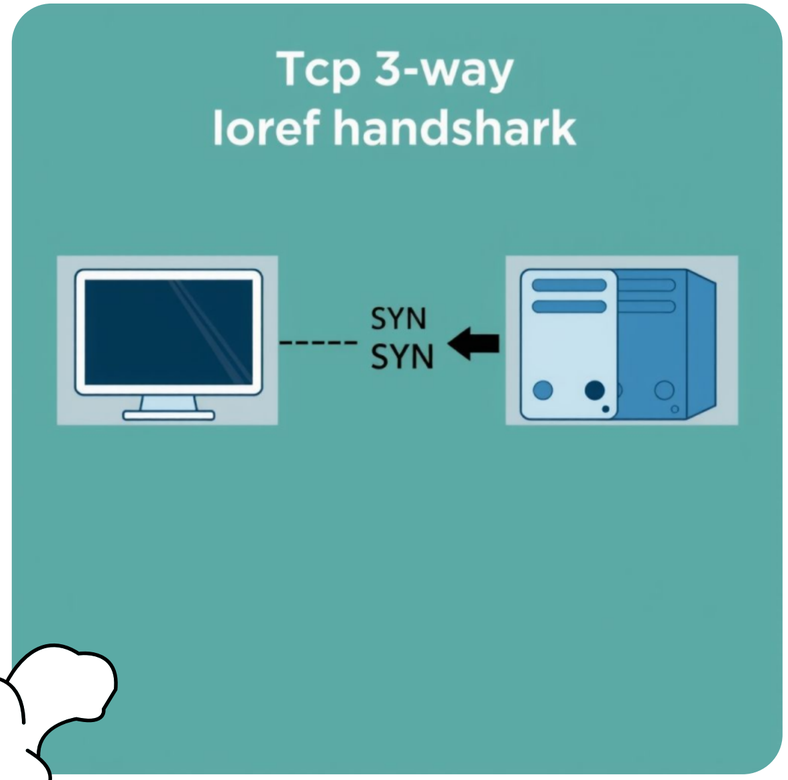
TCP(Transmission Control Protocol)

- TCP는 신뢰성 있는 데이터 전송을 위한 연결형 프로토콜입니다.
- 양방향 통신을 위해 연결을 먼저 수립해야 합니다.
- 데이터 전송 전 상호 간 '연결 가능 상태'임을 확인합니다.
- 초기 시퀀스 번호(ISN)를 교환하여 데이터 순서를 보장합니다.

연결 수립의 필요성

- 연결을 초기화하는 프로세스를 클라이언트라고 합니다.
- 연결 요청을 받는 프로세스를 서버라고 합니다.
- 클라이언트는 connect() 함수로 연결을 시도합니다.
- 3-way Handshake를 통해 양방향 통신 채널이 수립됩니다.

3-way Handshake 과정 (1)



1단계: SYN - 클라이언트 → 서버

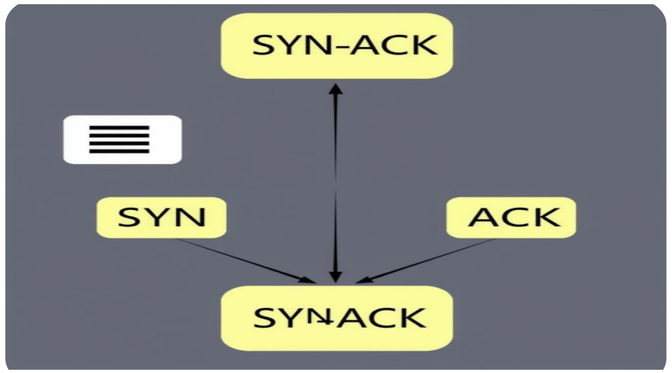
- 클라이언트가 연결을 요청하며 SYN 플래그를 설정합니다.
- 자신의 초기 시퀀스 번호(ISN)를 패킷에 포함시킵니다.
- clientSocket.connect((serverName, serverPort)) 호출로 시작됩니다.

SYN 패킷의 특징

- 애플리케이션 계층의 데이터는 포함되지 않습니다.
- 단순히 '연결하자'는 신호만 전달합니다.
- TCP 헤더의 SYN 플래그 비트가 1로 설정됩니다.

3-way Handshake 과정 (2)

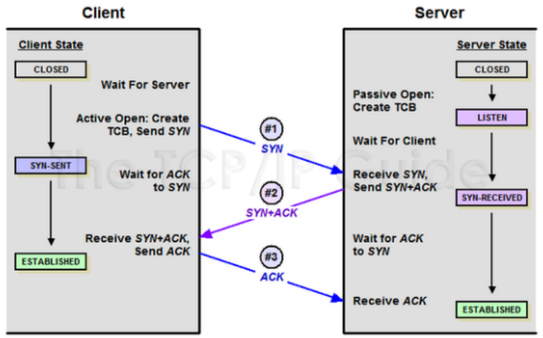
1



SYN + ACK 패킷

- 서버가 클라이언트의 SYN을 수신
- 자신의 ISN을 담은 SYN 전송
- 동시에 클라이언트 ISN+1로 ACK 응답

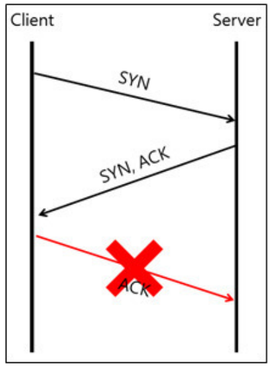
2



서버 측 자원 할당

- TCP 버퍼 할당
- 연결 관련 변수 초기화
- 연결 대기 큐(backlog)에 등록

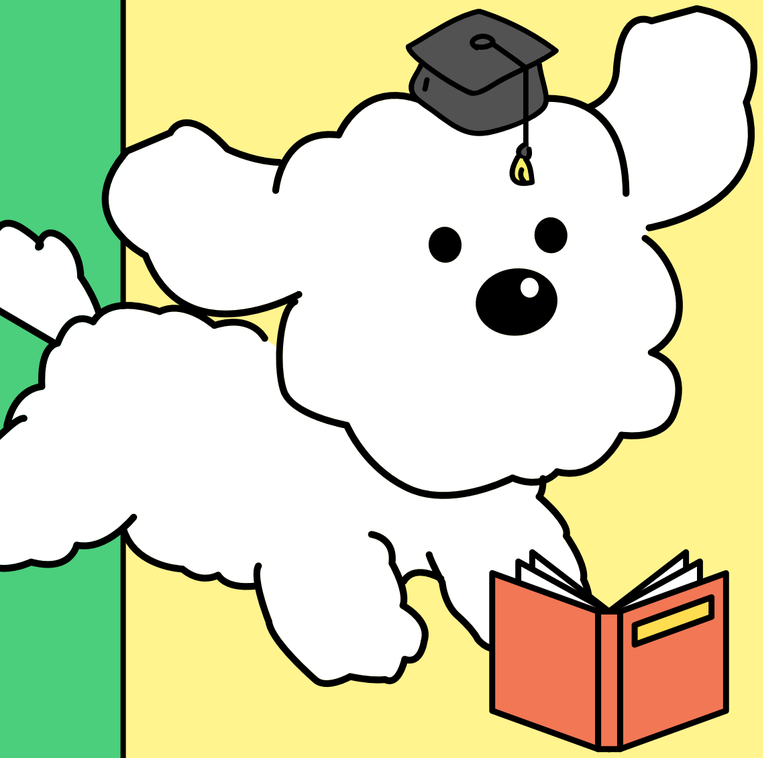
3



Half-open 연결 상태

- 아직 완전한 연결이 아님
- SYN Flooding 공격에 취약
- 애플리케이션 데이터는 없음

3-way Handshake 과정 (3)



ACK 패킷 전송

- 클라이언트가 서버의 SYN을 수신
- 서버의 ISN+1 값으로 ACK 응답
- 이로써 양방향 연결 확인 완료

연결 확립 완료

- 양쪽 모두 상대방의 수신 능력 확인
- 양쪽 모두 초기 시퀀스 번호 확인
- 데이터 전송 준비 완료

페이로드 포함 가능

- 3단계 ACK에는 데이터 포함 가능
- 연결 수립과 동시에 첫 데이터 전송
- 효율적인 통신 시작 가능

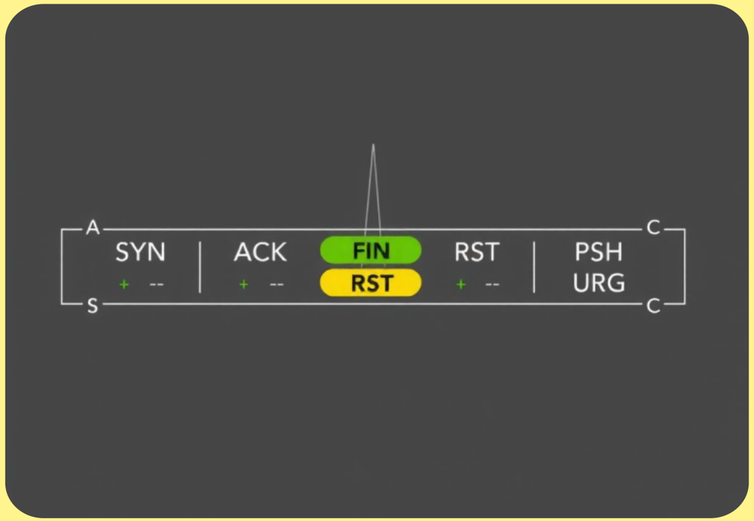
헤더와 페이로드 비유

- 헤더: 편지 봉투의 주소 정보
- 페이로드: 봉투 속 편지 내용
- 1,2단계는 봉투만, 3단계는 내용 포함

TCP 헤더와 플래그

TCP 헤더의 Control Flag

- TCP 헤더에는 6비트의 플래그가 존재
- 각 패킷이 전달될 때 비트 조합으로 상태 표시
- 패킷의 목적과 특성을 나타내는 중요 정보
- 연결 설정, 데이터 확인, 연결 종료에 활용



ACK 플래그

- 확인 응답 필드의 값이 유효함을 표시
- 이전에 받은 패킷을 확인했다는 의미
- 연결 설정 과정에서 중요한 역할 수행
- 데이터 수신 확인에 필수적인 플래그

연결 관련 플래그

- SYN: 연결 설정 요청
- FIN: 연결 종료 요청
- RST: 연결 강제 초기화
- 연결의 시작과 종료를 제어하는 플래그들

데이터 전송 관련 플래그

- PSH: 데이터를 즉시 상위계층에 전달
- URG: '긴급' 데이터임을 표시
- 데이터 처리 우선순위와 방식을 결정
- 특수한 상황에서 데이터 전송 제어

플래그 전송 방식

- IP 패킷 내부의 TCP 세그먼트에 포함
- 플래그 비트를 세팅하여 네트워크 통해 전송
- 별도 메시지가 아닌 TCP 헤더의 일부
- 효율적인 통신 제어 메커니즘 제공

왜 2-way가 아닌 3-way Handshake인가?

ISN 동기화

- '요청-응답'만으로는 초기 시퀀스 번호(ISN) 동기화 문제 발생
- 서버는 자신이 보낸 ISN을 클라이언트가 받았는지 확신할 수 없음
- 3-way를 통해 양쪽 모두 상대방의 ISN을 확인 가능
- 데이터 전송 시작점을 명확히 설정 가능
- 패킷 순서 추적과 관리에 필수적
- 데이터 무결성 보장을 위한 기본 요소
- 안정적인 통신의 기반이 됨

Half-Open 상태

- 2-way만 사용하면 중간에 패킷 손실이나 지연 발생 시 문제
- 한쪽은 연결되었다고 생각하고 다른 쪽은 모르는 상황 발생
- Half-Open 상태로 인한 리소스 낭비 가능성
- 서버 측에서 불필요한 연결 대기 상태 유지
- 클라이언트는 응답을 기다리며 타임아웃 발생
- 네트워크 자원 비효율적 사용
- 연결 상태에 대한 불일치 문제
- 정상적인 데이터 교환 불가능

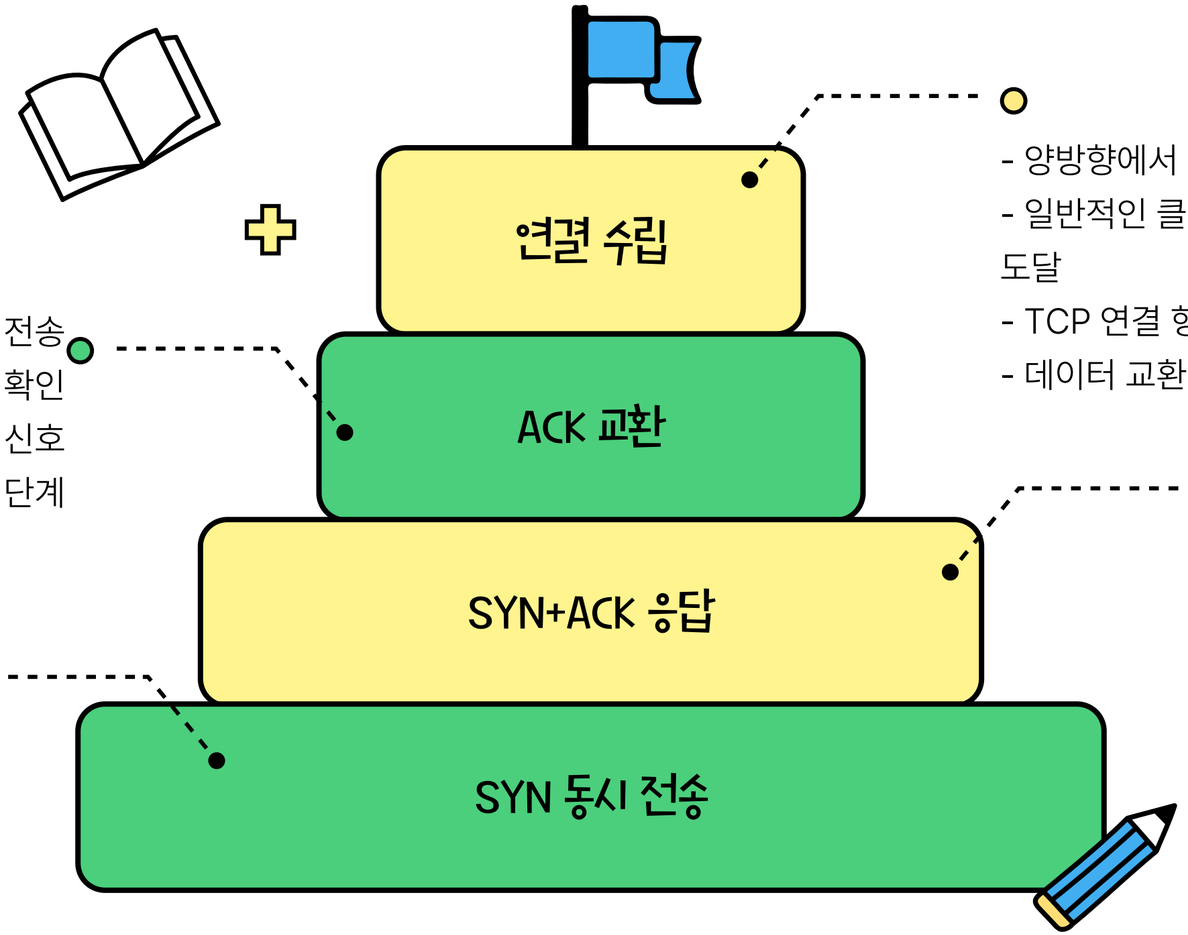
신뢰성 보장

- 3-way Handshake는 양방향 통신의 신뢰성 보장
- 양쪽 모두 연결 준비가 완료되었음을 확인
- 네트워크 상태와 연결 가능성 검증

- 연결 설정 과정의 완전성 확보

- 데이터 전송 전 안정적인 채널 구축
- 통신 오류 가능성 최소화
- 효율적인 리소스 관리 가능
- TCP의 신뢰성 있는 통신 보장

동시 연결 시도 (Simultaneous Open)



- 양방향에서 모두 연결이 정상적으로 수립
- 일반적인 클라이언트-서버 연결과 동일한 상태 도달
- TCP 연결 형성 완료
- 데이터 교환 준비 완료

- 마지막으로 서로 ACK를 전송
- 상대방의 SYN+ACK 수신 확인
- 양쪽 모두 연결 준비 완료 신호
- 3-way Handshake의 마지막 단계

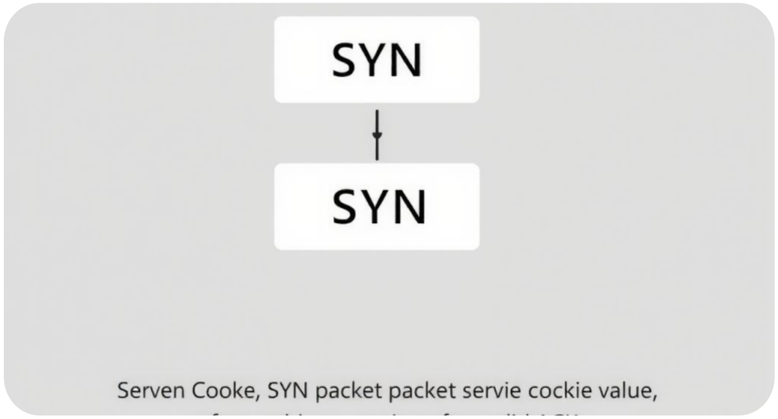
- 각각 SYN을 받으면 SYN+ACK 응답
- 상대방의 ISN 확인 및 자신의 ISN 전송
- 양방향 모두 연결 요청 승인
- 서로의 연결 요청에 대한 응답

- 두 호스트가 동시에 상대에게 SYN 전송
- 각자 자신의 ISN 정보 포함
- 일반적인 경우보다 드문 상황

SYN Flooding 공격

DoS 공격 유형	- 서비스 거부(DoS) 공격의 대표적 유형
공격 메커니즘	- 대량의 SYN 패킷 전송 후 ACK 미응답
Half-Open 상태	- 서버가 미완료 연결 상태 유지
서버 리소스 고갈	- 연결 큐, 메모리 등 자원 지속 점유
서비스 거부 상황	- 정상 사용자의 연결 요청 거부 발생

SYN Flooding 방어 기법

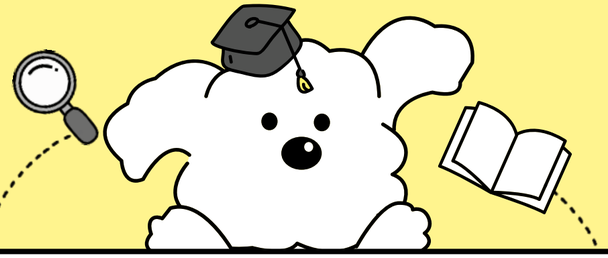


SYN 쿠키 기술

- 서버가 SYN 패킷 수신 시 연결 상태를 저장하지 않음
- 해시함수를 통해 특별한 초기 TCP 순서 번호('쿠키') 생성
- 클라이언트의 IP, 포트, 비밀값 등을 조합하여 생성
- 정상적인 ACK 응답이 올 때만 연결 자원 할당

추가 방어 기법: 백로그 큐 증가, 방화벽 필터링, 타임아웃 설정

- 백로그 큐 증가: 더 많은 Half-Open 연결 수용 가능
- 방화벽 필터링: 비정상적인 SYN 패킷 패턴 차단
- 타임아웃 설정: Half-Open 연결의 대기 시간 단축
- SYN Proxy: 방화벽이 초기 연결 요청 검증 후 서버에 전달



0-RTT 기법과 최신 기술

3-way Handshake는 1.5 RTT의 지연을 발생시키는데, 이를 줄이기 위한 최신 기술들이 개발되고 있습니다.

- TLS 1.3의 0-RTT: 이전 세션 키를 캐시하여 첫 패킷에 데이터 포함 가능
- QUIC 프로토콜: UDP 기반으로 설계되어 1-RTT 또는 0-RTT로 연결 수립
- HTTP/3: QUIC 기반으로 연결 설정 시간 단축

단, 0-RTT는 재전송·재연 공격(replay attack) 가능성이 있어 멱등(idempotent) 요청에만 사용이 권장됩니다.