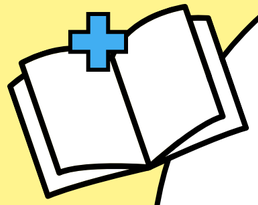
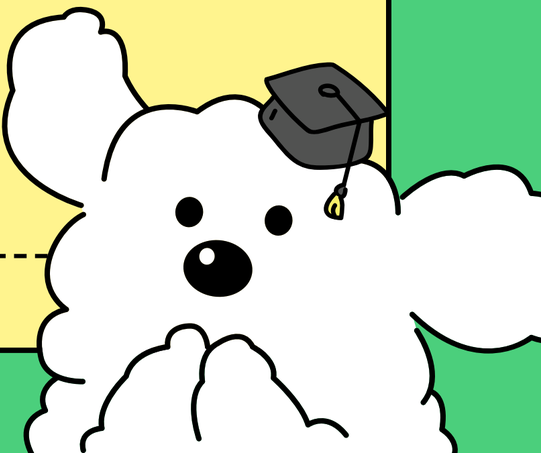
MIRI ACADEMY



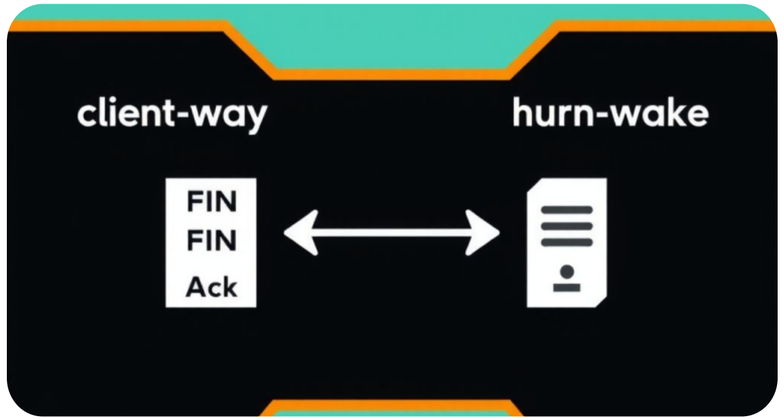
TCP 4-way Handshake

연결 종료의 원리와 과정



목차

4-way Handshake란?



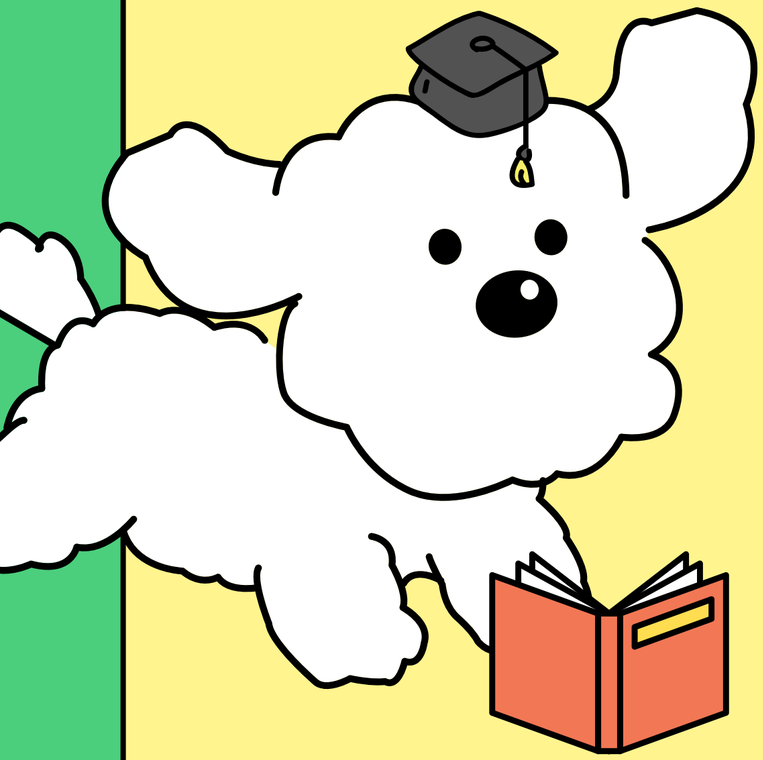
TCP 연결 종료 프로토콜

TCP 연결을 종료할 때 사용되는 프로토콜로, 양쪽 호스트가 서로 독립적으로 연결을 종료할 수 있도록 합니다. 이 과정에서 양측은 FIN 패킷을 주고받으며 데이터 전송이 완전히 끝났음을 확인합니다.

안전하고 신뢰성 있는 연결 종료를 위한 절차

- 양방향 모두 독립적으로 연결 종료 가능
- 각 방향별로 FIN과 ACK 패킷 교환
- 한쪽이 먼저 종료해도 다른 쪽은 데이터 전송 계속 가능
- 모든 데이터가 안전하게 전송된 후 최종 종료

4-way Handshake 단계



1단계: 클라이언트 → 서버

- FIN 플래그 전송
- 데이터 전송 완료 신호
- 연결 종료 의사 표현

2단계: 서버 → 클라이언트

- ACK 플래그 전송
- 클라이언트의 FIN 확인
- 서버는 데이터 계속 전송 가능

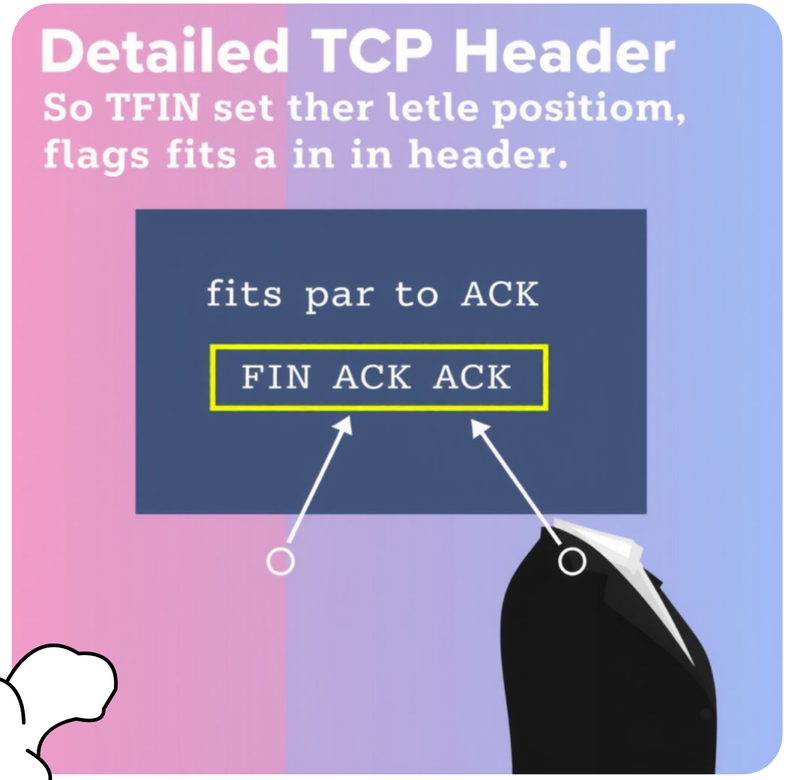
3단계: 서버 → 클라이언트

- FIN 플래그 전송
- 서버도 데이터 전송 완료
- 연결 종료 준비 완료

4단계: 클라이언트 → 서버

- ACK 플래그 전송
- 서버의 FIN 확인
- 최종 연결 종료

4-way Handshake 패킷 식별 방법



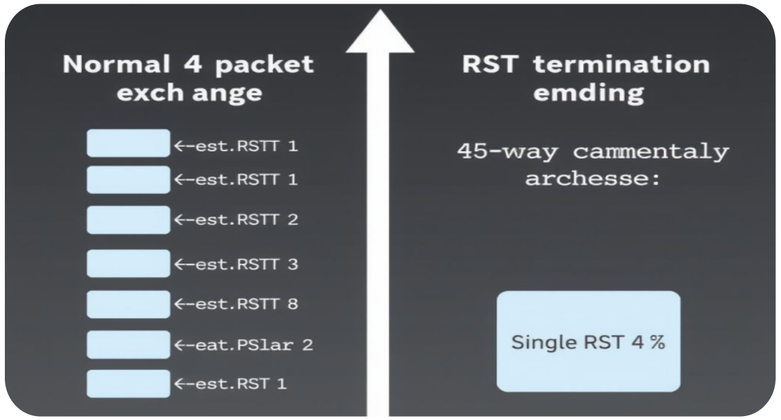
TCP 헤더 플래그 확인

- TCP 헤더의 플래그 필드 확인
- FIN 플래그: 연결 종료 의사 전달
- ACK 플래그: 상대방의 FIN 확인

연결 종료 패킷의 특징

- FIN 플래그가 설정된 패킷
- FIN+ACK 플래그가 함께 설정된 패킷
- 이러한 플래그가 있으면 4-way Handshake 과정임

빠른 연결 종료: RST 플래그



4-way Handshake 생략 방법

TCP는 정상적인 종료 과정에서 4-way Handshake를 사용하지만, 긴급하게 연결을 종료해야 하는 상황에서는 RST(Reset) 플래그를 사용할 수 있습니다. 이 방식은 정상적인 종료 과정을 생략하고 즉시 연결을 종료합니다.

RST 플래그를 이용한 즉시 연결 종료

- RST 패킷 전송 시 즉시 연결 종료됨
- 비정상 종료로 기록됨 (강제 종료)
- 상대방은 '연결이 강제로 끊겼다'고 인식
- 주요 사용 상황:
- 서버 다운 상황
- 심각한 에러 발생 시
- 공격 방어가 필요할 때
- 기타 긴급 상황에서 사용

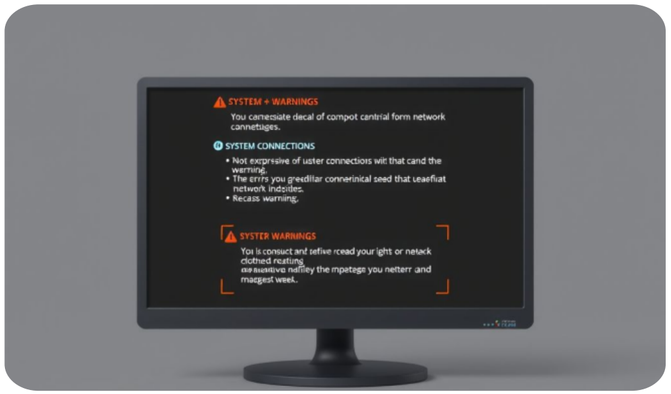
RST 플래그 사용 상황

1

서버 다운 상황

- 서버가 갑자기 종료되었을 때
- 하드웨어 장애 발생 시
- 시스템 충돌이 발생한 경우

2



에러 발생 시

- 메모리 할당 오류 발생
- 애플리케이션 충돌 시
- 프로토콜 오류 감지 시

3

공격 방어 필요할 때

- DDoS 공격 감지 시
- 비정상 연결 시도 차단
- 리소스 고갈 방지 필요 시

비정상 종료 시나리오

한쪽이 갑자기 연결 끊김

- 네트워크 장애로 한쪽이 응답 불가 상태
- FIN 전송 후 상대방이 응답하지 않음
- 연결이 비정상적으로 종료됨

ACK 타임아웃과 RST 패킷

- ACK 타임아웃: 일정 시간 응답 없으면 연결 종료
- RST 패킷 수신 시 즉시 연결 종료 처리
- 타임아웃 후 리소스 정리 및 해제

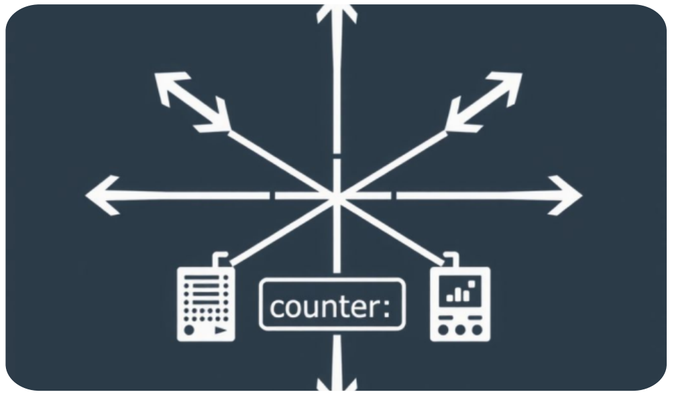
TCP의 비정상 종료 감지 방법

1

타임아웃 메커니즘

- 응답 대기 시간 초과 시 감지
- 일정 시간 후 연결 종료 처리
- 리소스 회수 및 정리

2



패킷 재전송

- 패킷 손실 시 재전송 시도
- 최대 재전송 횟수 초과 시 종료
- 재전송 실패 시 연결 해제

3

연결 상태 모니터링

- 소켓 상태 지속적 확인
- 비정상 패턴 감지
- 연결 상태 변화 추적

TIME_WAIT 상태의 목적 (1)

마지막 ACK 전송 후 대기 상태

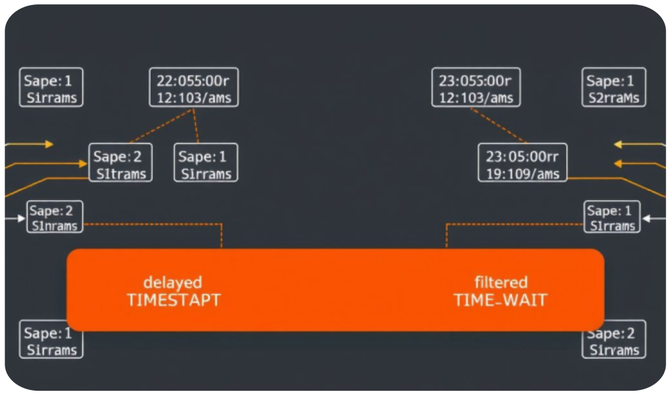
TIME_WAIT 상태는 TCP 연결 종료 과정에서 마지막 ACK을 보낸 후 일정 시간 동안 유지되는 상태입니다. 이 상태는 보통 2*MSL(최대 세그먼트 생존 시간) 동안 지속되며, 안전한 연결 종료를 보장하기 위한 중요한 메커니즘입니다.

ACK 유실 대비 메커니즘

- 마지막 ACK이 유실될 경우 상대방은 FIN을 재전송함
- TIME_WAIT 상태에서는 이러한 재전송된 FIN에 다시 ACK으로 응답 가능
- 이를 통해 상대방이 연결이 제대로 종료되었음을 확인할 수 있음
- 양쪽 모두 안전하게 연결 종료 상태로 전환 가능

TIME_WAIT 상태의 목적 (2)

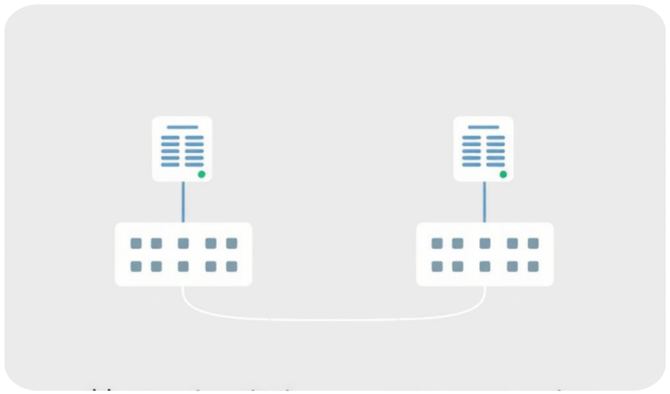
1



지연된 중복 패킷 제거

- 네트워크에 지연된 패킷 존재 가능
- 오래된 패킷이 나중에 도착할 수 있음
- TIME_WAIT로 이런 패킷 처리

2



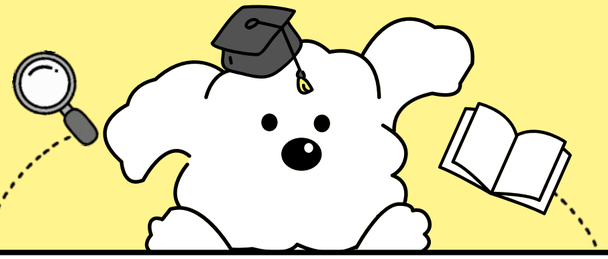
새 연결과의 혼선 방지

- 동일 포트로 새 연결 생성 가능
- 이전 연결의 패킷이 새 연결에 섞일 위험
- TIME_WAIT로 이런 혼선 방지

3

소켓 자원의 안전한 정리

- 모든 패킷이 처리될 시간 확보
- 리소스 안전하게 해제
- 메모리 누수 방지



4-way Handshake 요약

TCP 4-way Handshake는 연결을 안전하게 종료하기 위한 필수적인 메커니즘입니다.

- 양방향 종료 확인: 클라이언트와 서버 모두 독립적으로 FIN과 ACK을 주고받음
- 단계적 종료: FIN → ACK → FIN → ACK 순서로 진행
- TIME_WAIT 상태: 마지막 ACK 유실 대비 및 지연 패킷 처리
- 비정상 종료 대응: RST 플래그, 타임아웃, 재전송 등의 메커니즘 제공

이러한 과정을 통해 TCP는 신뢰성 있는 연결 종료를 보장합니다.