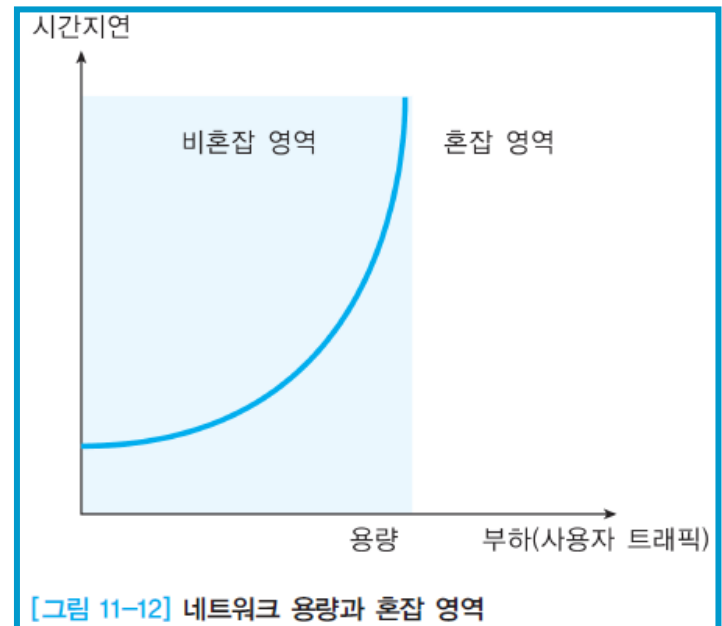


11장. TCP와 UDP

11-5 혼잡제어 기법

혼잡제어 (1)

- ▶ Congestion control
- ▶ 네트워크로 유입되는 데이터 트래픽의 양이 네트워크 용량을 초과하지 않도록 제어
- ▶ 데이터 총량이 네트워크가 처리할 수 있는 허용량을 초과한다면, 네트워크는 혼잡상태(congestion status)로 들어가게 됨
- ▶ 혼잡 영역
 - ▶ 가로축 : 사용자 트래픽
 - ▶ 세로축 : 네트워크에서의 지연시간
 - ▶ 사용자 트래픽이 네트워크 용량보다 충분히 적을 때는 시간지연이 그다지 크지 않지만, 네트워크 용량에 접근해가면 시간지연이 급증함

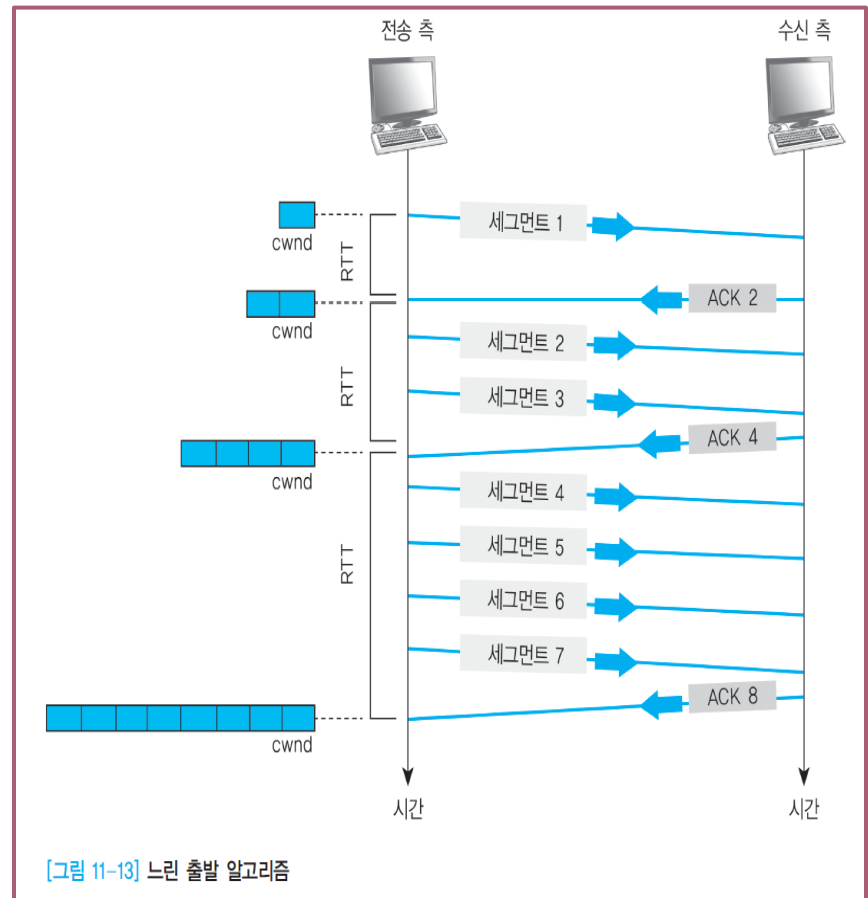


혼잡제어 (2)

- ▶ 혼잡상태는 네트워크의 성능을 현저히 떨어뜨리고, 결국 네트워크 기능을 마비시킴
 - ▶ 네트워크의 혼잡상태를 방지하기 위해서는 혼잡제어 기능이 요구됨
- ▶ TCP에서 사용하는 혼잡제어의 동작
 - ▶ TCP는 느린 출발 알고리즘과 혼잡회피 알고리즘을 기반으로 하여 혼잡제어를 수행
 - ▶ 느린 출발 단계로부터 시작하여 혼잡회피 단계, 혼잡감지 단계로 구분하면서 각 단계 별로 혼잡제어가 수행됨

TCP 혼잡제어 (3)

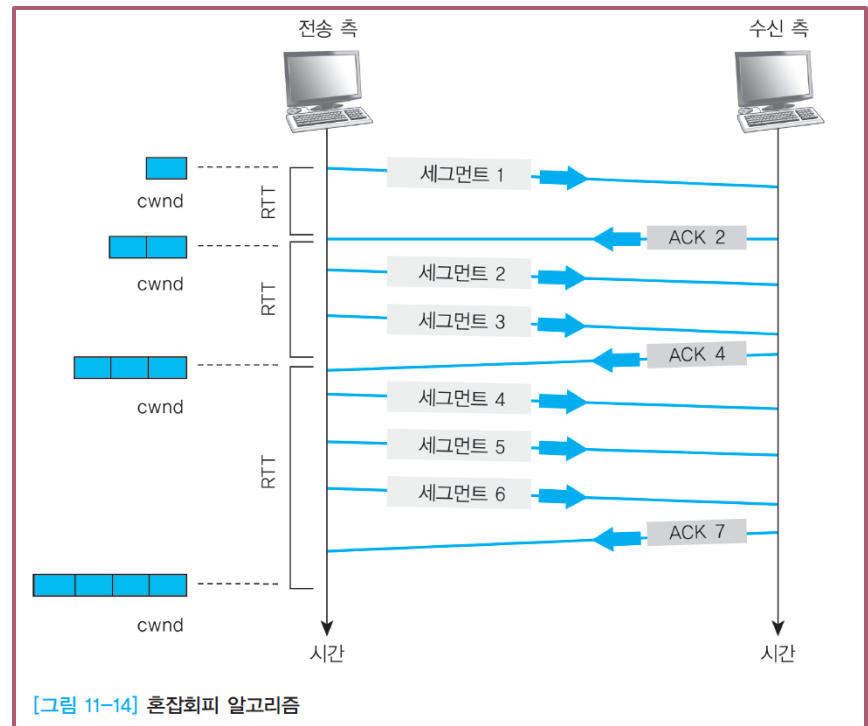
- ▶ 느린 출발 알고리즘(slow start algorithm)
 - ▶ 혼잡제어를 위한 윈도우의 크기(congestion window)를 한계치에 도달할 때까지 지수적으로 증가시키는 방법
 - ▶ 처음에 혼잡 윈도우(cwnd) 값이 1로 시작해 2, 4, 8...로 증가됨
 - ▶ 느리게 출발하여 점점 증가속도가 빨라짐



TCP 혼잡제어 (4)

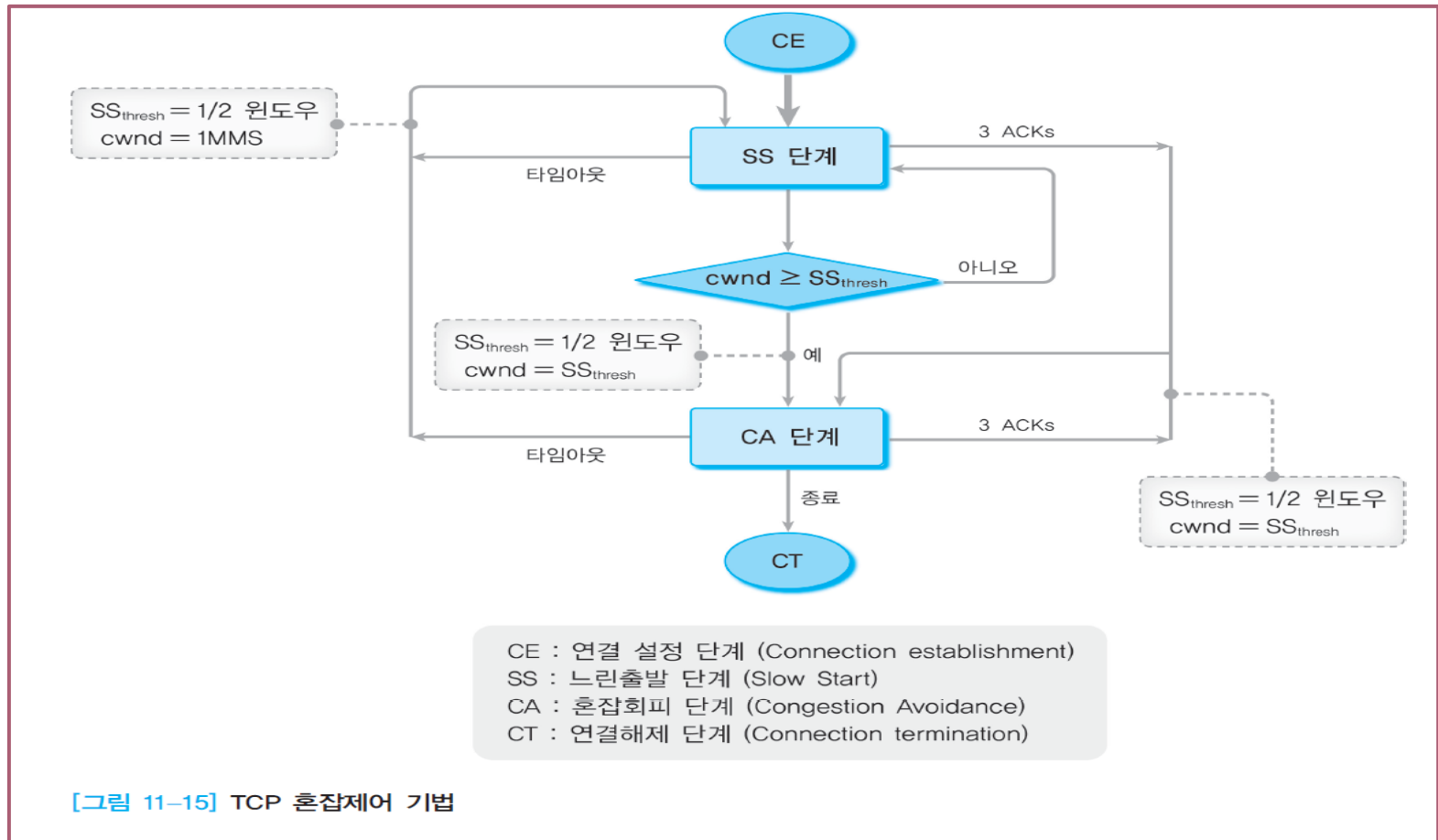
▶ 혼잡회피 알고리즘 (congestion avoidance algorithm)

- ▶ 혼잡 윈도우의 크기를 혼잡상태가 감지될 때까지 1씩 (1, 2, 3, 4, ...) 증가시키는 방법
- ▶ 느린 출발 알고리즘과는 달리 비교적 큰 값에서 시작하지만 증가속도는 일정하게 1만큼씩 증가함
- ▶ 혼잡감지(congestion detection) 단계에서 상호 간에 연계하여 동작되도록 함



TCP 혼잡제어 (5)

▶ 혼잡제어 - 혼잡감지 단계에서의 동작



TCP 혼잡제어 (6)

- ▶ 혼잡제어 - 혼잡감지 단계에서의 동작 (계속)
 - ▶ TCP에서 혼잡제어는 느린 출발 알고리즘과 혼잡회피 알고리즘이 서로 연동되어 수행
 - ▶ 타임아웃이 발생하거나 3번의 ACK 세그먼트가 수신되는 경우 → 혼잡이 감지된 것으로 판단
 - ▶ 만일 타임아웃이 발생하여 혼잡상태라고 판단되면 → 느린 출발단계로 다시 돌아가서 느린 출발 알고리즘이 수행
 - ▶ 3번의 ACK가 수신되어 혼잡상태로 판단되면 → 혼잡회피 단계로 되돌아가도록 하여 혼잡회피 알고리즘이 수행

혼잡제어(congestion control) – 예제

예제 11-5

혼잡 윈도우의 한계값이 16이고, $t = 8$ MSSs에서 타이머(RTO)의 값이 타임아웃이 된다고 가정했을 때, 혼잡제어 과정이 어떻게 진행되는지 설명하라.

풀이

초깃값으로 혼잡 윈도우값을 1로 설정하고 혼잡제어를 수행한다. 이때 느린 출발 알고리즘이 적용되어 2, 4, 8, ...로 증가하다가, $t = 4$ MSSs에서 윈도우 한계값인 16을 갖게 되면 이 시간 이후에는 혼잡회피 알고리즘이 적용된다. 즉 윈도우값은 16부터는 17, 18, 19, 20 처럼 1씩 증가된다. $t = 8$ MSSs에서 타이머값이 타임아웃되면, 이때 혼잡감지 단계에서의 동작으로 넘어간다. 즉 $t = 8$ MSSs에서 혼잡 윈도우값은 다시 1로 리셋되고, 이때 혼잡 윈도우의 한계값은 $\frac{20}{2} = 10$ 으로 변경된다. 이때 느린 출발 알고리즘이 적용되어 혼잡 윈도우값은 1, 2, 4, 8과 같이 지수적으로 증가된다. 이후 계속해서 이러한 과정이 반복된다.