**무선 네트워크 실습**

**10주차 실습 보고서**

1. **Slotted ALOHA 코드 설명**

Slotted ALOHA의 동작 방식은 ALOHA와 동일하나 한 slot들을 인식하고 있는 터미널들이 동기화 메커니즘에 따라 slot의 시작 경계에서 전송을 시작한다는 점에서 다르다. 즉, 새롭게 전송할 프레임이 생기면 다음 slot의 시작 경계까지 대기한 뒤 전송을 하게 되며, collision은 2개 이상의 프레임이 완전히 겹치는 slot에서 발생한다. **스크린샷이(가) 표시된 사진

자동 생성된 설명**

스크린샷이(가) 표시된 사진

자동 생성된 설명

먼저 전역 변수에 대해서 설명을 하면 다음과 같다. 터미널의 상태는 STANDBY, TRANSMIT, COLLISION 각각 대기, 전송, 충돌을 나타내고, Srate는 데이터 전송 속도, Plen은 패킷의 길이, Mnum은 단말의 개수, Mplen은 각 단말에서 패킷의 길이, Mstate는 현재 각 단말의 상태를 말한다. Tint는 패킷 생성 구간의 평균, Rint는 패킷 재전송 구간의 평균, Spnum은 전송이 성공한 패킷의 총 개수, Splen은 전송이 성공한 패킷들의 총 길이, Tplen은 전송 패킷의 길이, Wtime은 slot time-패킷 생성 시점을 의미한다.

다음 static variable로는 mgtime은 패킷 생성 시점, mtime은 패킷 전송 시점, s\_length는 슬랏의 길이를 말한다.

If now\_time<0의 부분은 터미널의 초기화 상태 설정을 말한다. 이 때, random table을 생성해주고, 슬랏 길이를 패킷 1개 전송 시간으로 설정한다. 이후 패킷 생성 시점 mgtime을 Tint를 이용해 계산하고, mgtime을 슬랏 길이로 나누고 ceil 함수를 적용하여 next\_slot이라는 패킷 생성 시점의 다음 슬랏을 설정한다. 패킷 전송 시점 mtime은 다음 슬랏과 슬랏 길이를 곱해서 생성한다. 그리고 각각 터미널의 상태를 0으로 초기화 하고 Mplen 각 단말의 패킷 길이는 모두 동일하게 Plen으로 두며, 다음 시점을 전송 시점 중 가장 빠른 것으로 리턴한다.

이 후 각 상태가 되었을 때에 대한 설정을 하였다. 먼저 전송이 성공한 터미널을 find을 통해 찾고 만약 탐지가 되었을 때, 전송된 총 패킷 수를 하나 증가 시키고, 패킷 길이 Splen와 Wtime도 업데이트를 해준다. 전송이 끝났기 때문에 단말의 상태를 STANDBY로 바꿔주고 다음 패킷 생성 시점과 다음 전송할 slot의 번호와 다음 패킷 전송할 시간을 각각 mgtime, next\_slot, mtime으로 설정해준다.

두번째, 단말의 상태가 COLLISION인 단말이 탐지가 되었을 때, 단말의 상태를 다시 대기 상태 STANDBY로 바꿔주고, 재전송할 시점 mtime을 rand를 사용하여 random time만큼 대기하여 설정한다. 그리고 TRANSMIT 상태와 동일하게 다음 슬랏을 next\_slot으로 설정하고 대기 시간 이후에 재전송하여 mtime을 설정한다.

마지막, 단말의 상태가 STANDBY, 즉 mtime이 현재 slot time과 동일한 단말이 있다면, 전송을 시작하므로 단말의 상태를 TRANSMIT로 바꿔준다. 이 후 전송이 끝나 시간을 mtime으로 설정해주고, 전송 시점을 구해서 이를 슬랏으로 바꿔 next\_slot을 설정한다. 마지막으로 패킷길이를 전송 패킷 길이에 합해주어 전송을 완료한다.

각각의 상태 확인 후 mtime의 최소값으로 다음 시점을 설정한다.

1. **ALOHA와 Slotted ALOHA 비교**

Slotted ALOHA는 Pure ALOHA를 보완한 무선 통신 프로토콜이다. Carrier sensing을 하지 않는 다는 점에서 Pure ALOHA와 공통점을 갖고 있지만, 동기화 기법을 적용하여 Pure ALOHA의 Throughput을 2배로 증가시켰다. 즉, Pure ALOHA는 COLLISION이 일어날 경우, random time만큼 대기를 하고 재전송을 하나, Slotted ALOHA는 random time만큼 대기하나 time slot를 이용하여 다음 time slot에서 동기화를 하여 보낸다. 이렇게 Slot을 사용하면 재전송 시 충돌이 일어날 확률이 줄어들고, 전체 처리량 또한 향상된다.

실제 Thoughput과 성능을 비교하면 다음과 같다. 이론적으로 채널의 트래픽을 G라고 설정했을 때, Pure ALOHA와 Slotted ALOHA의 thorughput은 다음과 같다.

Pure ALOHA = G \* exp( -2 \* G ) , Slotted ALOHA = G \* exp( -G )

스크린샷, 지도이(가) 표시된 사진

자동 생성된 설명스크린샷, 지도이(가) 표시된 사진

자동 생성된 설명이를 실제 Simulation을 하고 비교를 했을 때 다음과 같다. 점들이 simulation의 값이고, 선이 이론에 따른 값이다. 이론에 따르면 Pure ALOHA는 G=0.5일 때 최대, Slotted ALOHA는 G=1일 때 최대의 throughput을 갖는다. 그래프를 보면, G가 0.5 보다 작을 때 Pure ALOHA와 Slotted ALOHA의 처리량을 큰 차이가 없으나, 0.5보다 클 때 Slotted ALOHA가 throughput이 월등히 낫다는 것을 알 수 있다. 또한, 슬랏의 길이를 여기에서는 패킷 길이와 동일하게 맞춰주었으나 슬랏 길이가 좀 더 커진다면 Slotted ALOHA의 성능은 현재보다 낮아질 수 있다.

<Pure ALOHA 그래프> <Slotted ALOHA 그래프>