

Data Communication HW2

2019-12333 정윤서

1. Introduction

Layer 2의 주요 역할은 flow control&error control, addressing, framing, logical link management 등이다. 그중에서도 핵심적인 역할이 바로 sender와 receiver 간 frame 통신에서 over-run을 방지하는 flow control과, 에러가 난 frame에 대해서는 retransmission을 요구하는 error control이다. Flow control과 error control을 동시에 하는 방식으로는 Stop&Wait ARQ 방식이 있지만 이는 매우 비효율적이다. 따라서 보통은 flow control으로 sliding-window 방법을 사용하고, error control을 위해서는 go-back-N이나 selective repeat 방법을 사용하게 된다. 본 보고서에서는 후자의 세 가지 scheme에 대해 간략히 살펴본 뒤 두 가지 error control 방식에 따라 통신 성능이 어떻게 달라지는지 분석해 보고자 한다.

2. Scheme description

Sliding window는 말 그대로 sender와 receiver가 한 시점에 각각 최대 W개의 window를 open할 수 있는 flow control scheme이다. Sending window는 sender가 ack를 받지 않고도 추가적으로 보낼 수 있는 frame들을 나타내고, receiving window는 sender로부터 아직 받지 않았지만, 앞으로 받을 것으로 예상되는 frame들을 나타낸다. ack를 받지 않고 열릴 수 있는 window의 개수가 제한되어 있기 때문에 sender는 앞서 전송한 frame에 대한 ack를 받아야 새로운 window를 open할 수 있다. 그러나 이 scheme은 transmission error가 발생했을 때의 flow에 대해서는 설명하고 있지 않다. 이를 위해 필요한 것이 go-back-N과 selective repeat scheme이다.

Go-back-N은 비교적 단순한 방식으로, receiver는 error가 발생한 frame 이후부터 수신하는 모든 frame을 다 무시하고 error가 발생한 frame에 대해서 NAK을 보내거나 아무것도 전송하지 않는다. 따라서 sender는 timeout이 되거나 NAK를 받은 첫 frame부터 시작해 여태까지 보냈던(그러나 receiver에게 무시당했을) 모든 frame을 재전송한다. Go-back-N을 sliding window에 접목시켜 보면, sender 입장에서 이미 닫힌 window일지라도 ACK를 받지 못한 frame이라면 추후 재전송해야 할 가능성이 있기 때문에 이에 대한 copy를 buffer에 가지고 있어야 함을 알 수 있다. 다만 receiver 입장에서는 에러가 발생한 이후의 frame들을 buffer에 보관할 필요가 없기 때문에 보다 buffer 관리가 쉽다.

Selective repeat에서는 receiver가 error가 발생한 이후의 frame들을 무시하지 않고, 버퍼에 가지고 있다가 error가 발생했던 frame이 제대로 수신되면 그제서야 버퍼에 쌓인 frame들을 기존 frame number 순서에 따라 상위 layer로 보내게 된다. 따라서 버퍼 관리가 보다 어렵지만, sender 입장에서는 한 frame에서 error가 났을 때 모든 frame을 재전송

해야될 일은 생기지 않기 때문에 frame size가 큰 경우에는 유리하다. Sender는 timeout이 나거나 NAK를 수신한 frame에 대해서만 재전송을 한다.

3. Performance parameters

Go-back-N과 selective repeat 방식을 사용했을 때의 성능을 비교하기 위해 사용할 수 있는 performance parameter에는 W , a , λ , p_{data} , p_{ack} 등이 있다. 각 parameter의 의미는 다음과 같다.

- W : ACK를 받지 않고 전송할 수 있는 window의 최대 개수
- p_{data} : transmission error probability
- p_{ack} : ack error probability¹⁾
- a : propagation delay/transmission time
- λ : poisson rate of packet arrival

4. Performance comparison

$W < 2a+1$ 일 때, λ 를 조절함에 따라(packet arrival rate 조절하여 load condition을 다르게 설정하기 위함) packet delay와 channel utilization이 어떻게 달라지는지 확인해 보았다. Packet delay는 모든 성공한 packet의 delay를 합산한 것을 성공한 packet의 수로 나누어서 구하였고, channel utilization은 모든 성공한 packet의 transmission time을 합산한 것을 총 걸린 시간으로 나누어주었다(유효시간의 비율). Go-back-N, selective repeat 모두 input parameter를 동등하게 설정²⁾하고 각 load condition에 따라 λ 만 0.1, 0.3, 0.5로 설정해 주었을 때, 성능은 다음과 같았다.

- Performance tables³⁾

Load condition	Packet delay			Channel utilization		
	low	mid	high	low	mid	high
Go-back-N	10	25	48	0.06	0.23	0.41
Selective Repeat	4	10	21	0.05	0.2	0.38

- Discussion

Go-back-N과 Selective Repeat을 사용한 경우 모두 load condition이 높을수록

1) 본 보고서에서는 selective repeat에서만 ACK error가 발생하는 경우를 고려하고 있다.
 2) Selective repeat의 경우에만 p_{ack} 를 0.01로 추가적으로 설정함. 이외 input parameter의 경우, $N=10000$, $timeout_len=6$, $t_{pk}=0.5$, $t_{pro}=3$, $W=(2a+1)-3$, $p_{data}=0.05$ 로 통일함.
 3) packet delay는 소수점 첫째 자리에서 반올림, channel utilization은 소수점 셋째자리에서 반올림하여 계산.

delay는 늘어나고, utilization이 올라가는 것을 확인할 수 있었다. 이는 packet arrival rate가 높을수록 working queue에 쌓이는 packet의 수가 많아졌기 때문으로 보인다. 한 packet이 성공적으로 전송되기까지는 더 많은 시간이 걸리지만, 채널이 아무것도 하지 않고 있는 시간은 줄어들기 때문이다. 또한 selective repeat은 go-back-N보다 packet delay가 상대적으로 낮게 나타났는데, 이는 전송이 실패했을 경우 모든 packet을 다시 보낼 필요 없이 에러가 난 packet만 다시 보내면 되기 때문인 것으로 생각된다. 또 channel utilization이 보다 낮게 나온 것은 ack error를 고려했기 때문이라고도 생각해 볼 수 있다.