

## 22-1 데이터통신 과제 2

- Error and Flow Control Simulation on Point-to-Point Link

2017-19651 이혜민

### 1. Introduction

본 과제에서는 두 error control scheme 인 Go- Back-N 과 Selective-Repeat 의 성능을 평가하고 비교하고자 한다. 이를 위해 여러 가정 하에 시뮬레이션을 진행한다. 두 scheme 모두 양방향이 아닌 단방향 전송을 가정하며, 따라서 하나의 sender 와 하나의 receiver 만 존재하며 역할이 고정되어 바뀌지 않는다. 또한 ACK error 가 없고 processing time 도 없다고 가정한다. NAK 는 존재하지 않으며 ACK 와 timeout 만 존재한다고 가정한다.

위의 조건 하에서 시뮬레이션을 진행하되, 두 scheme 의 성능을 평가할 지표로 Packet transmission delay 와 Channel Utilization 을 사용한다. Packet transmission delay 는 패킷이 sender 에서 receiver 로 전송되고 receiver 가 이를 정상적으로 수신할 때까지 걸린 시간을 뜻한다. Channel Utilization 은 해당 채널에서 sender 가 패킷을 전송하는 데에 소요된 시간을 전체 통신 소요 시간을 나눈 값이며 얼마나 delay 없이 통신이 이루어지는지를 평가할 수 있는 값이다.

### 2. Scheme Description

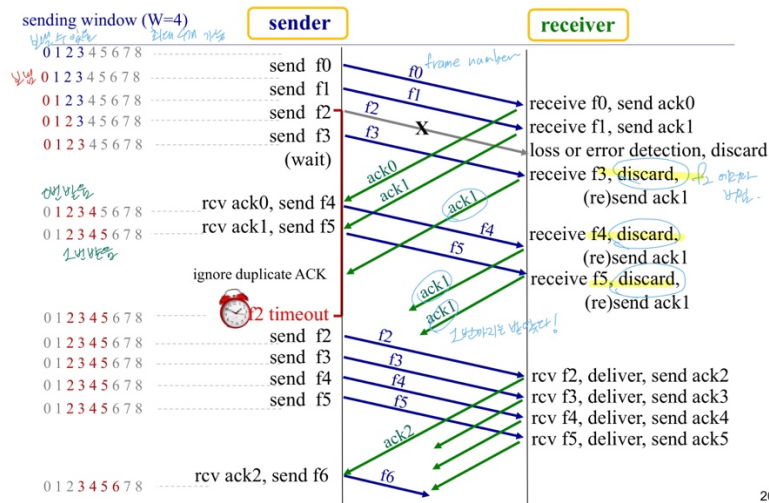
Internet protocol stack 의 Layer2 에 해당하는 link 에서는 flow control 과 error control 을 수행한다. 여기서 사용되는 대표적인 flow control scheme 인 sliding window 는 sender 가 프레임이 전달되었다는 신호인 ACK 를 receiver 에게 받아야 다음 패킷을 전송할 수 있는 stop & wait scheme 과는 다르게 window 너비인 W 를 정해두고 W 개의 프레임까지는 ACK 를 받지 않고도 전송할 수 있다. 연속적으로 frame 을 전송할 수 있기 때문에 stop&wait 에 비해 상대적으로 높은 utilization 을 보인다.

프레임 전송을 관리하는 flow control scheme 과 함께 사용되는 것이 error control scheme 이고, Go-Back-N 과 Selective-Repeat 은 error control scheme 의 대표적인 예시이다.

Go-Back-N ARQ 의 경우 에러가 발생하여 원하는 순서 번호의 프레임을 받지 못하면 receiver 에서 그 이후에 받은 프레임은 모두 버리면서 원하는 프레임만 기다리는 방식이다. 이를 그대로 원하는 N 으로 되돌아가서 진행하는 것이다. Go-Back-N 은 전송이 진행되면서 에러 없는 프레임을 순서대로 받게 된다. 전송 양상은 아래 Figure 1 과 같다.

반면 Selective-repeat 방식의 경우 에러가 발생하여 원하는 순서 번호의 프레임을 받지 못해도 리시버에서 원하는 순서 번호 이후의 프레임을 받아서 버퍼에 임시로 저장한 후 에러가 발생한 프레임을 재수신 했을 때에 순서를 조정한다. 즉, 순서와 상관없이 수신된 모든 프레임을 받은 후 순서를 조정하게 된다. 전송 양상은 아래 Figure 2 과 같다.

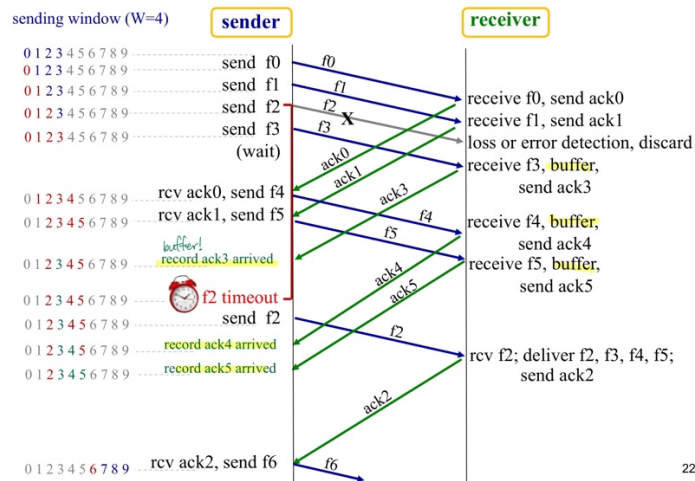
## Go-back-N Example



20

Figure 1. Go-Back-N ARQ

## Selective-repeat Example



22

Figure 2. Selective-repeat ARQ

### 3. Simulation Method (Performance Parameters)

성능 측정 시뮬레이션에서 사용된 Performance parameters 는 다음과 같다.

- W : Sliding-window flow control 에서 사용할 window size
- p : frame transmission 에서 error 가 발생할 확률
- a : link propagation time 과 packet transmission time 의 비
- $\lambda$  : packet arrival time 을 결정하는 poisson rate

Load condition 에 따른 Packet delay 와 Channel utilization 을 평가하기 위해  $a$  값을 변경하여 load condition 을 조절하였다.  $p$  와  $w$ ,  $\lambda$  값을 고정하여 두 가지 케이스를 선정했으며 첫 번째 케이스는  $p$  를 0.05,  $w$  를 8,  $\lambda$  를 10 으로 두었고 두 번째 케이스는  $p$  를 0.005,  $w$  를 20,  $\lambda$  를 10 으로 두었다.

두 가지 케이스 모두에 대해  $a$  를 1, 3, 5, 7, 10, 20, 30, 50, 70, 100, 200, 700, 1000 로 변경하며 channel utilization 과 packet delay 를 측정하였다. 시뮬레이션에서 전달 패킷의 개수  $N$  을 10,000 개로 설정하고 모든 경우에 10 번씩 시뮬레이션한 후 결과의 평균치를 사용하였다.

#### 4. Simulation Result (Performance Comparison)

Figure 3 은  $a$  값을 변화시켜가며 얻은 utilization 값 분포를 나타낸 결과이다. Go-Back-N 과 Selective-Repeat scheme 각각에 대한 case 1, case 2 를 나누어 표시하였다. 앞에서 언급했듯이 case 1 은  $p = 0.05$ ,  $w = 8$ ,  $\lambda = 10$  의 조건, case 2 는  $p = 0.005$ ,  $w = 20$ ,  $\lambda = 10$  의 조건 하에서 수행되었다.

같은 조건에서  $a$  값이 작을수록 높은 utilization 을 가지며 Go-back-N 보다는 Selective-Repeat 를 사용하였을 때에 조금 더 높은 utilization 값을 얻었다.

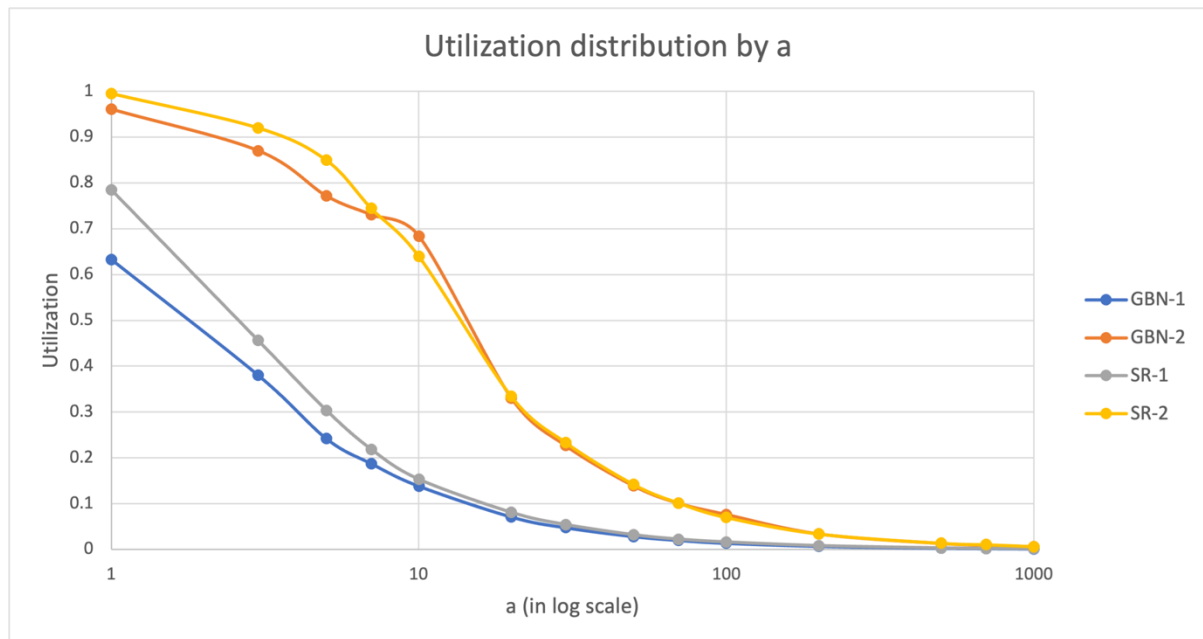


Figure 3.  $a$  값에 따른 utilization 값 분포

Figure 4 는  $a$  값을 변화시켜가며 얻은 평균 delay 값의 분포를 나타낸 결과이다. 두 scheme 모두  $a$  값이 1 인 경우에만 특별히 높은 delay 값을 가지며, 이후  $a$  를 증가시키기에 따라 delay 값은 크게 영향을 받지 않는다. Case 2 에 대해서는 Go-Back-N 과 Selective-Repeat 에서 거의 비슷한 delay 값이 나타났으며 case 1 의 경우에는 Go-Back-N scheme 에서 조금 더 높은 delay 값이 나타났다. 이는 case 1 이 case 2 보다 더 큰 window size 를 가지기 때문에 두 scheme 간의 차이가 더 두드러지는 것으로 보인다.

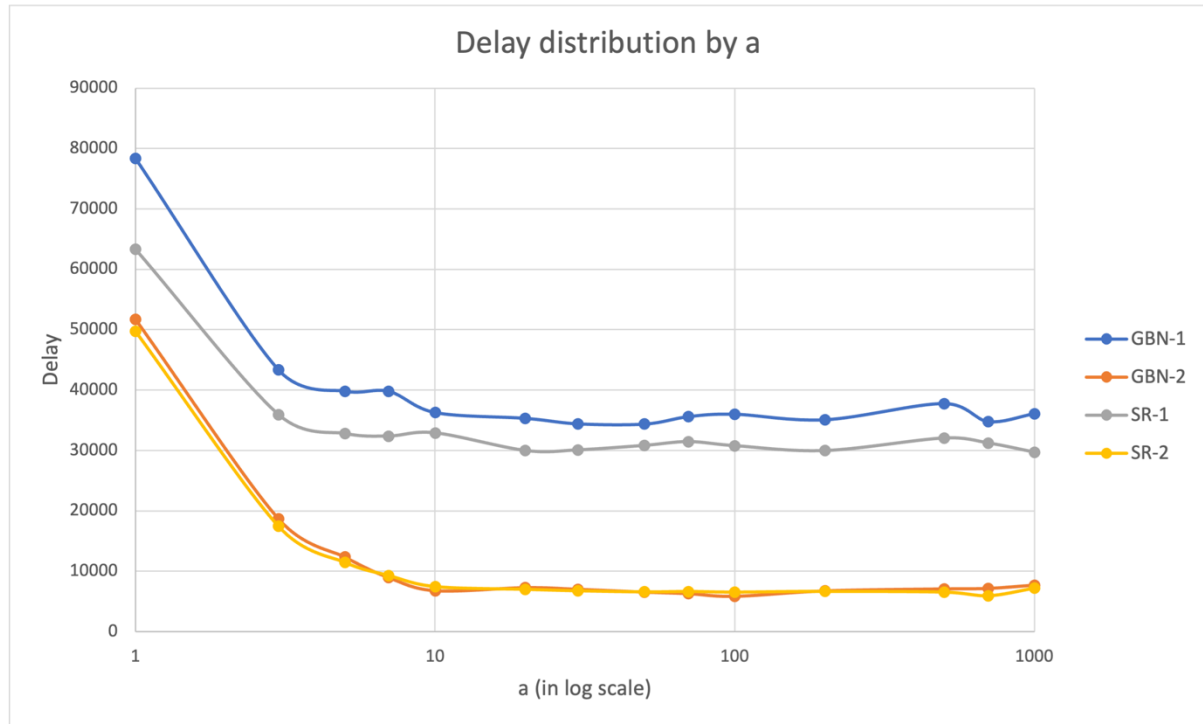


Figure 4. a 값에 따른 delay 값 분포

a 값으로 load condition 을 조절하였을 때 delay 에서 큰 변화를 찾아볼 수 없었기 때문에 lambda 값을 조절하여 추가로 시뮬레이션을 진행하였다. p 와 w, a 값을 고정하여 두 가지 케이스를 선정했으며 첫 번째 케이스는 p 를 0.05, w 를 8, a 를 10 으로 두었고 두 번째 케이스는 p 를 0.005, w 를 20, a 를 10 으로 두었다.

Figure 5, 6 은 각각 lambda 값을 변화 시켜 가며 얻은 utilization 값의 분포와 delay 값의 분포를 나타낸 그래프이다. Utilization 의 경우 모든 케이스에서 lambda 를 증가시켰을 때 utilization 도 일정 수준까지 따라 증가하지만 이후에는 더 증가하지 않고 값이 유지된다. Delay 의 경우 두 scheme 모두 case 1 이 case 2 에 비해 높은 delay 값을 가진다. 또한 lambda 값이 증가할수록 delay 값도 증가하며, Go-Back-N scheme 이 Selective-Repeat 보다 높은 delay 를 보인다.

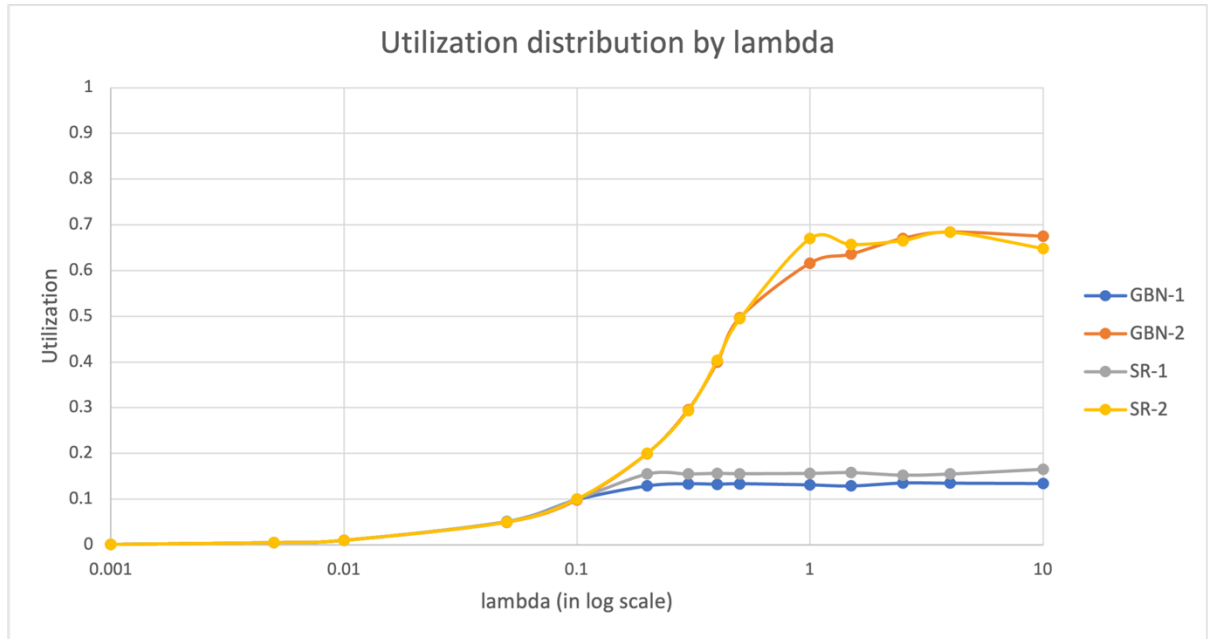


Figure 5. lambda 값에 따른 utilization 값 분포

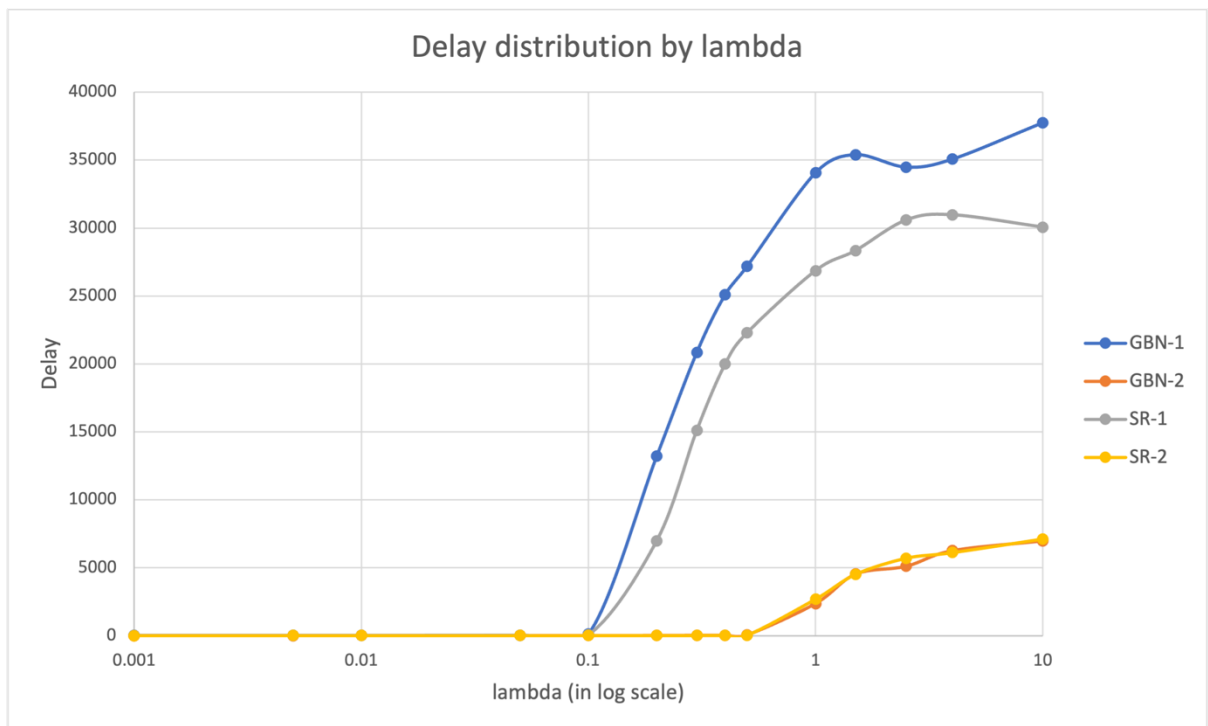


Figure 6. lambda 값에 따른 delay 값 분포

## 5. Discussion

전체적으로  $a$  값이 작을수록 Utilization 값은 증가하는 양상을 보였다.  $a$ 는 패킷이 이동하는 시간인 propagation delay를 패킷이 전송되는 데에 걸린 시간인 transmission delay로 나눈 값으로  $a$ 가 작다는 것은 같은 패킷 전송 시간에 대해 이동하는 데에 시간이 더 적게 소요된다는 것을 의미한다. 따라서 전체 소요 시간이 상대적으로 줄어들게 된다.

두 error control scheme 을 비교하자면 Go-back-N 의 경우 에러가 발생한 패킷 이후에 도달하는 패킷들은 제대로 전송되어도 추후에 재전송되어야 하므로 재전송 하는 시간이 추가로 소요된다. 시뮬레이션에서 Utilization 은 Go-back-N 을 사용할 때 보다 Selective- Repeat 를 사용할 때에 더 크고, delay 는 Go-Back-N 에서 더 크게 나타났다.

$\lambda$  는 단위 시간에서 랜덤하게 발생하는 사건의 평균 횟수를 의미한다. 시뮬레이션에서  $\lambda$  값에 따라 단위 시간 내에 발생하는 패킷 수가 결정된다.  $\lambda$  가 작으면 패킷이 발생 되는 시간 간격이 크기 때문에 각 패킷이 발생되고 얼마 기다리지 않고 전송될 수 있다. 반대로  $\lambda$  가 크면 패킷 간 발생되는 시간 간격은 작지만 window size 는 고정되어 있기 때문에 패킷이 전송되고 받을 때까지 소요되는 시간이 증가한다.

## Reference

1. Data and Computer Communications class materials, 2022.