

마코프 에러형태 하에서의 Blast 프로토콜의 수율 분석

Throughput Analysis for Blast Protocols under Markov Error Type

홍정식* · 홍정완** · 이창훈***

Jung Sik Hong* · Jung Wan Hong** · Chang Hoon Lie***

Abstract

In this paper, a Variation of Blast with Go-Back-N(V-BGBN) protocol is proposed, which differs from Blast with Go-Back-N(BGBN) and Blast with Full Retransmission on Error(BFRE) protocols in the retransmission strategy of packets. Performances of these three protocols under correlated packet errors are analyzed. Throughput efficiency of an arbitrary packet is obtained under the assumption that the round trip delay and the packet length are respectively constant. Recursive formula and difference equations are used as analytical tools. Correlation of packet errors is modelled by a two state Markov chain. The throughput efficiencies under these protocols are compared. V-BGBN protocol is shown to be superior to other two protocols in high speed network.

1. 서론

데이터 통신이 활성화된 이후, 다양한 에러제어 프로토콜들(error control protocols)이 제안되어 왔다.[1-5,8,12] 통신 네트워크의 환경(즉, 채널의 에러율(channel error rate), 전송용량(transmission capacity), 전파시간(propagation delay), 프로토콜 처리시간(protocol processing time) 등)이 변하면 에러제어 프로토콜의 성능도 변하게 된다. 따라서 통신 네트워크의 환경에 따라 적절한 에러 제어 프로토콜을 선택하여야 한다. 예를 들어, 에러 발생률이 낮고, 전송용량이 큰 최근의 고속 네트워크(high-speed network)에는 blast 프로토콜을 사용하는 것이 효율적이

다.[9,11] 이전의 프로토콜(예를 들면, go-back-N 등의 window-based 프로토콜)들은 blast 프로토콜에 비해 수율(throughput)면에서는 효과적이지만, control 패킷(ACK이나 NACK)과 연속전송에 따른 무의미한 패킷(에러가 난 패킷 이후의 패킷)이 많음으로 인해 네트워크에 큰 부하를 주었다. 그리고 프로토콜의 처리과정도 복잡하여 프로토콜 처리시간도 큰 단점이 있었다. 그러나 에러 발생률이 낮은 경우에는 blast 프로토콜을 사용함으로써, 유사한 수율내에서 네트워크의 부하도 줄일 수 있으므로 더 효율적이라고 할 수 있다.

Blast 프로토콜은 메시지 단위로 데이터를 전송하는 경우가 많이 사용되었다. 그러나 메시지의 크기가 큰 경우에

* 서울산업대학교 산업공학과

** 한성대학교 산업시스템공학부

*** 서울대학교 산업공학과

는 블록(block) 단위로 메시지를 쪼개어 처리한다.[9] 그리고 blast 프로토콜은 블록 단위로 에러를 처리하는 것이 특징이다. 즉, ACK(또는, NACK)가 패킷 단위로 보내지는 것이 윈도우(window) 프로토콜이라면, 블록을 모두 수신하고 난 뒤, 그 블록에 대한 ACK(또는, NACK)가 보내지는 것이 blast 프로토콜이다.

Blast 프로토콜은 재전송 방법 측면에서 BFRE(Blast with Full Retransmission on Error), BGBN(Blast with Go-Back-N), 그리고 BSR(Blast with Selective Repeat)의 3가지 세부 프로토콜로 나누어져 왔다. 이 세가지 프로토콜은 블록에 에러가 발생하지 않았을 경우에는 작동 방법이 동일하지만, 에러가 발생한 경우에는 서로 다르다. 에러가 발생한 경우 블록 전체가 재전송되는 것이 BFRE이다. 따라서 BFRE에서는 전송의 경우와 재전송의 경우 블록의 크기가 동일하다. 에러가 발생한 패킷부터 이후 패킷들을 재전송하는 것이 BGBN이다. 따라서 재전송의 경우 블록의 크기가 전송시보다 작아진다. 그리고 에러가 발생한 패킷만을 전송하는 것이 BSR이다.

Zwaenepoel[12]은 round-trip delay가 일정(constant)하고, 패킷들간의 에러 발생이 서로 독립이라는 가정하에서 Blast 프로토콜의 성능을 분석하였다. 여기에서 round-trip delay란 송신자가 블록의 전송을 끝낸 시점부터, 수신자로부터 ACK(또는, NACK) 신호를 수신할 때까지 소요되는 시간을 의미한다.(그림 1 참조) Mukherjee[9]는 round-trip delay를 분포를 가지는 확률변수로 두고 분석하여 round-trip delay가 일정하다는 가정을 해결하였으며, Pieris and Sasaki[11]는 에러 발생이 서로 독립이 아니라 마코프 형태의 에러 모형을 도입하여 위의 두 가정을 모두 해결하였다. 본 연구에서는 전송 채널의 특성상 마코프 형태의 에러 모형을 사용하고, round-trip delay는 일정하다고 가정하였다. 실제로 Mukherjee도 round-trip delay가 일정한 모형에 대해서 실험하였다.

Blast 프로토콜들 중에는 BFRE와 BGBN이 프로토콜의 단순성(simplicity) 때문에 주로 사용되고 있다. 그러나 블록의 크기가 커질수록 BFRE의 성능은 급격히 나빠진다. 그리고 BGBN의 성능은 round-trip delay가 커질수록 급격히 나빠진다. 이 문제점을 해결하기 위해, 본 연구에서는 새로운 BGBN 프로토콜(V-BGBN)을 제안하였다. 그리고 BFRE, BGBN, V-BGBN 프로토콜의 수율을 순환식

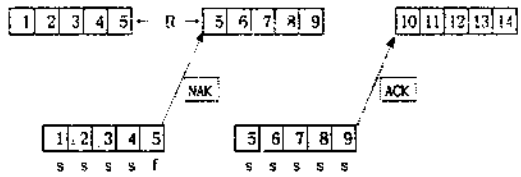


그림 1. 패킷의 상태에 따른 V-BGBN 프로토콜의 블록의 형태(N=5, R=3)

(recursive formula)과 차분 방정식(difference equation)을 이용하여 유도하였다.

2. V-BGBN 프로토콜의 성능분석

2.1 V-BGBN 프로토콜

본 연구에서는 에러 제어 프로토콜들의 작동과정을 쉽게 비교할 수 있도록 하기 위해서, 프로토콜들을 전송(transmission) 방법과 재전송(retransmission) 방법의 두 부분으로 나누어 살펴보았다.

BGBN 프로토콜의 전송 방법은 다음과 같다; 송신자(sender)는 N개의 패킷(전송단위인 블록이 N개의 패킷으로 구성)을 연속적으로(continuously) 전송한다. 수신자는 N개의 패킷을 다 받은 후에, 에러가 없는 경우에는 ACK 신호를 보내고, 에러가 있는 경우에는 첫 번째 에러가 발생한 패킷에 대한 NACK 신호를 보낸다. 그리고 재전송 방법은 다음과 같다; 에러가 난 패킷이 i번째 패킷인 경우, 송신자는 i번째 패킷과 블록내의 그 다음의 패킷들을 포함하여 (N-i+1)개의 패킷을 다시 전송한다.

본 연구에서 제안한 V-BGBN 프로토콜은 재전송 방법에서 BGBN 프로토콜과 다르다. V-BGBN 프로토콜에서는 에러가 난 패킷이 i번째 패킷인 경우, 송신자는 그 다음의 (N-i+1)+(i-1)개의 패킷을 다시 전송한다. 다시 말해서 항상 N개의 패킷을 전송한다.(그림 1 참조) 그림 1에서 s가 표시가 된 패킷은 정상적으로 전송된 패킷이며, f가 표시된 패킷은 에러가 발생한 패킷이다. 그리고 상수 R은 round-trip delay를 의미한다.

2.2 Markov 에러 형태

일반적으로 패킷의 길이는 일정하다. 따라서 패킷 전송

시간 T도 일정하다고 할 수 있다. 표현상의 편의를 위해서 본 연구에서는 T=1로 두었다. 그리고 패킷간의 종속적인 에러 발생을 반영하기 위하여, 패킷들의 상태(state)를 2-state Markov-chain으로 표현하였다. 따라서 i번째 패킷의 상태를 Y_i 라고 하고, 다음과 같이 상태에 대한 정의를 하였다.

$Y_i=0$, i번째 패킷이 제대로 전송된 경우

$Y_i=1$, i번째 패킷에 에러가 발생한 경우

그리고 상태전이확률(state transition probabilities)은 다음과 같다.

$$p = \Pr(Y_{i+1}=0 | Y_i=0)$$

$$q = \Pr(Y_{i+1}=1 | Y_i=0) = 1 - p$$

$$r = \Pr(Y_{i+1}=0 | Y_i=1)$$

$$s = \Pr(Y_{i+1}=1 | Y_i=1) = 1 - r$$

그리고 n슬롯(slot) 떨어져 있는 패킷들의 상태전이확률을 나타내는 n-단계(n-step) 상태전이확률은 생성함수기법(generating function techniques)을 이용하면 아래와 같이 표현된다.

$$\begin{bmatrix} p^{(n)} & q^{(n)} \\ r^{(n)} & s^{(n)} \end{bmatrix} = \frac{1}{r+q} \begin{bmatrix} r & q \\ r & q \end{bmatrix} + \frac{(1-r-q)^n}{r+q} \begin{bmatrix} q & -q \\ -r & r \end{bmatrix}$$

그리고 Markov-chain이 안정상태에서 시작한다고 가정하여, $\Pr(Y_i=0) = \pi_0$, $\Pr(Y_i=1) = \pi_1$ 이라고 표현하였다. 여기에서 안정상태확률 π_0, π_1 은 위 식으로부터 각각 $\pi_0 = \frac{r}{q+r}$, $\pi_1 = \frac{q}{q+r}$ 으로 표현된다.

2.3 순환식

본 연구에서는 성능평가 척도로 수율($= \eta$)을 사용하였다. 수율은 임의의 패킷이 에러없이 전송되는 시간(이하, 성공전송시간), τ 의 역수로 표현할 수 있다. 그리고 임의의 패킷의 성공전송시간은 그 패킷이 송신자로부터 맨 처음 전송이 시작된 시간부터 에러 없이 전송이 완료되는 시간까지를 의미한다.

블록내의 i번째 패킷의 성공전송시간의 기대치를 $\tau_i, i=1, \dots, N$ 라고 정의하고, 상태전이확률과 τ_i 와의 관계를 나

타낸 것이 그림 2이다. 그림 2의 (a)는 3번째 패킷이 첫 번째 전송과 첫 번째 재전송시 모두 에러가 발생한 경우이다. 그림 2의 (b)는 첫 번째 블록의 패킷들은 모두 첫 번째 전송시 성공적으로 전송되었고, 두 번째 블록의 마지막 패킷만이 첫 번째 전송시 에러가 발생하였고, 이 패킷은 첫 번째 재전송시 성공적으로 전송된 경우이다.

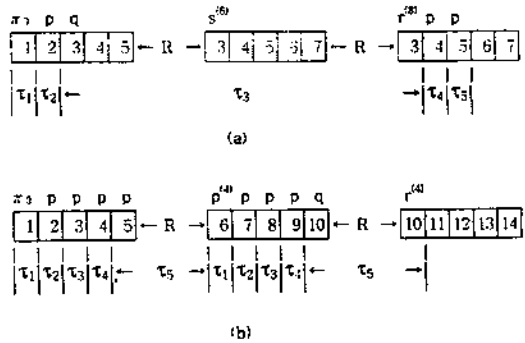


그림 2. 패킷 상태확률에 따른 블록의 형태(N=5, R=3)

그런데, 패킷들이 연속적으로 전송되기 때문에 동일한 블록내의 패킷들끼리 성공전송시간이 서로 중복된다. 따라서 블록내에서 이전 패킷에 에러가 없는 경우에만 전송이 시작되었다고 생각하고, 블록내의 마지막 패킷(N번째 패킷)의 성공전송시간에만 round-trip delay를 포함시키면 패킷간의 중복이 없어지게 된다. 즉, 다른 패킷(첫 번째에서 (N-1)번째 패킷까지)의 성공전송시간에는 round-trip delay를 포함시키지 않는다.(그림 2 참조) 블록내의 첫 번째 패킷의 재전송 회수는 자신의 상태에만 좌우된다. 그러나 두 번째 패킷의 재전송 회수는 자기 자신 뿐만 아니라 첫 번째 패킷의 상태에도 좌우된다. 따라서 블록내의 i번째 패킷의 성공전송시간 τ_i 를 계산하기 위해서는 i개 패킷의 상태를 모두 고려하여야 한다. 이 τ_i 와 i개 패킷의 상태와의 관계를 순환식으로 표현하면 아래와 같다.

$$\tau_i = \pi_0 + \pi_1(m_0 + \tau_i) \tag{1}$$

$$\tau_i = \pi_0 p + \pi_0 q(m_1 + \tau_i) + \pi_1 \tau_i^2 \tag{2}$$

$$\tau_i = \pi_0 p^{i-1} + \pi_0 p^{i-2} q(m_{i-1} + \tau_i)$$

$$+ \sum_{k=3}^i \pi q^{i-k} q \tau_{k-1}^i + \pi_i \tau_i^i, \quad 3 \leq i \leq N-1 \quad (3)$$

$$\tau_N = \pi q^{N-1} (R+1) + \pi q^{N-2} q (m_{N-1} + \tau_1^N) + \sum_{k=3}^N \pi q^{N-k} q \tau_{k-1}^N + \pi_N \tau_N^N \quad (4)$$

여기에서, $m_i = N + R - i$

그리고 위의 식들에서 $\tau_j^i, i \geq j$, 는 이전 전송시에 i 번째 슬롯에 위치했던 패킷이 다음 재전송시에는 j 번째 슬롯에 위치한 경우의 기대 성공전송시간이다.

블록 내의 첫 번째 패킷이 성공적으로 전송될 확률이 서로 다르기 때문에 τ_j^i 와 τ_j^j 를 표현하는 식은 서로 다르다. τ_j^i 들을 표현하는 순환식은 위와 유사한 방법을 사용하여 아래와 같이 나타낼 수 있다.

$$\tau_1^i = r^{(m_i-1)} + s^{(m_i-1)} (m_0 + \tau_1^i), \quad 1 \leq i \leq N \quad (5)$$

$$\tau_2^i = r^{(m_i-2)} p + r^{(m_i-2)} q (m_1 + \tau_1^i) + s^{(m_i-2)} \tau_1^i, \quad 2 \leq i \leq N-1 \quad (6)$$

$$\tau_j^i = r^{(m_i-j)} p^{j-1} + r^{(m_i-j)} p^{j-2} q (m_{j-1} + \tau_1^i) + \sum_{k=3}^j r^{(m_i-j+k)} p^{j-k} q \tau_{k-1}^j + s^{(m_i-j)} \tau_j^j, \quad 3 \leq j \leq i \leq N-1 \quad (7)$$

$$\tau_N^i = r^{(m_0)} p^{N-1} (R+1) + r^{(m_0)} p^{N-2} q (m_{N-1} + \tau_1^N) + \sum_{k=3}^N r^{(m_0)} p^{N-k} q \tau_{k-1}^N + s^{(m_0)} \tau_N^N \quad (8)$$

τ_i 과 τ_1^i 는 식(1)과 (5)로부터 쉽게 유도할 수 있다.

$$\tau_i = \frac{r^{(m_0)} + m_0 \pi_i}{r^{(m_0)}} \quad (9)$$

$$\tau_1^i = r^{(m_i-1)} + s^{(m_i-1)} (m_0 + \frac{r^{(m_0)} + m_0 s^{(m_0)}}{r^{(m_0)}}) \quad (10)$$

만약 m_0 가 round trip delay라면, 식(9)의 역수는 지금까지 알려져 있는 예러가 종속인 상황하에서의 Go-Back- ∞ 프로토콜의 수율과 동일함을 알 수 있다.[6, 7]

아래 [보조정리 1]에서는 τ_j^m 과 τ_j^n 의 관계를 보여주고 있다.

[보조정리 1] $\tau_j^m = \tau_j^n = \tau_j, j \leq m, n \leq N, j \geq 2$

증명) Appendix 1 참조

[보조정리 2]에서는 $\tau_j, 2 \leq j \leq N$ 의 닫힌 형태의 해 (closed form solution)를 보여주고 있다.

[보조정리 2] τ_j 의 해는 아래와 같다.

$$\tau_i = A_i \quad (11)$$

$$\tau_i = A_i + q \left(\sum_{k=2}^{i-1} A_k \right), \quad 3 \leq i \leq N \quad (12)$$

여기에서, $A_i = p^{i-1} + p^{i-2} q (m_{i-1} + \tau_1^i), 2 \leq i \leq N-1$

$$A_N = (R+1) p^{N-1} + p^{N-2} q (m_{N-1} + \tau_1^N)$$

증명) Appendix 2 참조

2.4 V-BGBN의 수율

수율은 성공전송시간 τ 의 역수로 표현할 수 있다. 임의의 패킷의 성공전송시간을 구하기 위해서는, 임의의 패킷이 전송을 위해서 처음으로 블록 내부의 슬롯에 위치할 때, 각 슬롯별 위치확률(location probability)을 알아야 한다. 임의의 패킷이 i 번째 슬롯에 위치할 확률을 f_i 라고 하면 임의의 패킷의 성공전송시간의 기대값은 아래와 같다.

$$\tau = \sum_{i=1}^N f_i \tau_i \quad (13)$$

그리고 $f_{n,i}, n)N$ 를 n 번째 패킷이 전송을 위해 블록 안에 처음 위치할 때, 블록 내의 i 번째 슬롯에 위치할 확률이라고 하면, f_i 는 $f_{n,i}$ 의 n 에 대한 극한확률(limiting probability)이라고 할 수 있다. 그리고 f_i 와 $f_{n,i}$ 는 아래의 [보조정리 3]과 같이 유도할 수 있다.

[보조정리 3] f_i 와 $f_{n,i}$ 는 아래와 같이 표현할 수 있다.

$$\begin{bmatrix} f_{n,1} \\ f_{n,2} \\ f_{n,3} \\ \vdots \\ f_{n,i} \\ \vdots \\ f_{n,N} \end{bmatrix} = \begin{bmatrix} 0 & 0 & \cdots & 0 & p^{N-1} \\ & 1 & 0 & \cdots & 0 & p^{N-2} q \\ & & 0 & 1 & \cdots & 0 & p^{N-3} q \\ & & & \vdots & \vdots & \vdots \\ & & & & 0 & 0 & \cdots & 0 & p^{N-i} q \\ & & & & & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & q \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad (14)$$

$$f_i = \frac{p^{N-i}}{\sum_{k=0}^{N-1} p^k} \quad (15)$$

증명) Appendix 3 참조

따라서 V-BGBN 프로토콜의 성공전송시간은 식(9), (11),(12),(13),(15)를 이용하여 유도할 수 있다.

3. BFRE와 BGBN의 성능분석

2장에서 사용한 방법과 유사한 방법을 사용하면 BFRE와 BGBN의 수율을 쉽게 유도할 수 있다. 또한, 이 두 프로토콜에서는 블록 안에 패킷이 처음 위치할 때, 패킷의 순서에 따라 위치가 이미 정해져 있기 때문에, 이 두 프로토콜에 적용되는 해법은 V-BGBN의 경우보다 훨씬 간단하다.

3.1 BFRE

t 를 N 개의 패킷을 성공적으로 전송하는 데 소요되는 기대 시간이라고 정의하면, BFRE의 수율 η 는 N/t 으로 표현할 수 있다. 패킷들의 상태확률에 따라 t 는 아래와 같이 표현된다.

$$t = (N+R) + \sum_{i=0}^{N-2} \pi_i q^i t_{N-i-1} + \pi_N t_N \quad (16)$$

위 식에서 t_i 는 $(N-i+1)$ 번째 패킷이 에러일 경우의 N 개의 패킷을 성공적으로 전송하는 데 소요되는 기대 시간이다. 따라서 t_i 는 t 의 경우와 유사하게 아래와 같이 표현된다.

$$t_i = (N+R) + r^{(R+i)} \sum_{k=0}^{N-2} p^k q t_{N-k-i} + s^{(R+i)} t_N, \quad i=1, \dots, N \quad (17)$$

식(16)과 (17)을 결합하면, t 를 유도할 수 있다.

$$t = \left(\frac{\pi_1}{r^{(R+N)}} + \frac{r^{(R+N)} + \sum_{i=1}^{N-1} p^{N-i-1} q s^{(R+i)}}{p^{N-1} r^{(R+N)}} \right) (N+R) \quad (18)$$

3.2 BGBN

BGBN 프로토콜의 경우, N 개의 패킷을 성공적으로 전송하는 데 소요되는 기대 시간 t 는 식(16)과 동일하다. 그

러나 t_i 는 재전송시 전송되는 패킷의 수가 다르기 때문에 BFRE의 경우와 다르다. BGBN 프로토콜의 t_i 는 아래의 순환식의 형태로 표현된다.

$$t_i = (R+1) + s^{(R+1)} t_i \quad (19)$$

$$t_i = (R+i) + r^{(R+i)} \sum_{k=0}^{i-2} p^k q t_{i-k-1} + s^{(R+i)} t_i, \quad i=2, \dots, N \quad (20)$$

따라서 t 는 식(16),(19),(20)을 결합하여 유도할 수 있다.

$$t = \left(1 + \frac{\pi_1}{r^{(N+R)}} \right) (N+R) + q \sum_{i=1}^{N-1} \frac{R+i}{r^{(R+i)}} \quad (21)$$

4. 성능 비교

본 연구에서는 BFRE, BGBN 그리고 V-BGBN 프로토콜에 대해서 여러 가지 환경하에서 수율을 비교해 보았다.

그림 3과 표 1은 $N=256, R=64, r=0.25$ 인 환경하에서 에러발생확률 q 에 대한 위의 세가지 프로토콜들의 수율을 나타낸 것이다. $q < 10^{-5}$ 인 경우에는 세가지 프로토콜

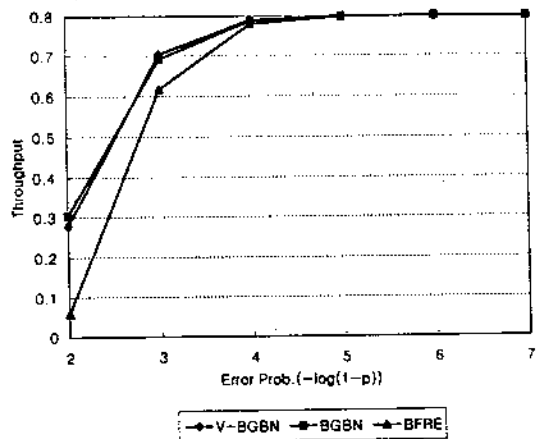


그림 3. 패킷에러확률에 대한 수율의 변화($N=256, R=64, r=0.25$)

의 수율은 $\eta \approx 256/(256+64)=0.8$ 로서 에러가 없는 경우의 수율과 거의 동일하다고 볼 수 있다. 그러나 BFRE의 경

표 1. 패킷에러확률에 대한 수율의 변화
(N=256, R=64, r=0.25)

프로토콜 \ 패킷에러확률	V-BGBN	BGBN	BFRE
10^{-2}	0.277550	0.304044	0.059297
10^{-3}	0.703303	0.691087	0.617385
10^{-4}	0.789577	0.787629	0.779545
10^{-5}	0.798949	0.798746	0.797931
10^{-6}	0.799895	0.799874	0.799793
10^{-7}	0.799989	0.799987	0.799979

표 2. 평균에러지속시간에 대한 수율의 변화
(N=256, R=64, q=10⁻³)

프로토콜 \ 평균에러지속시간	V-BGBN	BGBN	BFRE
2	0.704707	0.692457	0.618617
4	0.703303	0.691078	0.617385
8	0.700511	0.688335	0.614934
16	0.694877	0.682883	0.609988
32	0.682275	0.671656	0.598925
64	0.653354	0.647012	0.573538
128	0.593354	0.592641	0.522080
256	0.491128	0.487972	0.431130
512	0.344957	0.337226	0.302816

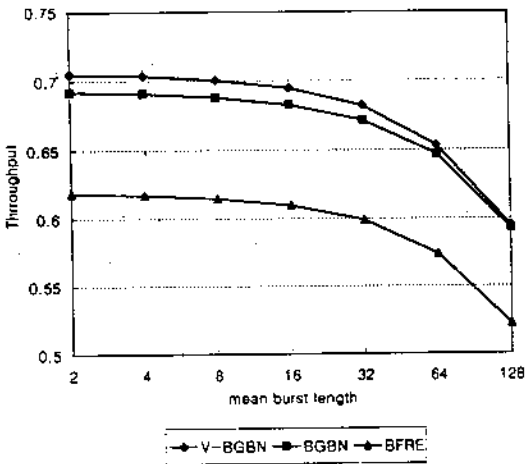


그림 4. 평균에러지속시간에 대한 수율의 변화(N=256, R=64, q=10⁻³)

우에는 $q > 10^{-4}$, 그리고 BGBN과 V-BGBN의 경우에는 $q > 10^{-3}$ 일 때부터 수율이 급격하게 낮아짐을 알 수 있다. 그리고 $q < 10^{-4}$ 인 비교적 에러가 적은 채널(low error channel)인 경우에는 V-BGBN이 BGBN보다 성능이 뛰어난 것을 알 수 있으며, $q > 10^{-3}$ 인 에러가 많은 채널(high error channel)인 경우에는 BGBN이 V-BGBN보다 성능이 뛰어난 것을 알 수 있다.

그림 4와 표 2는 $q=10^{-3}$ 인 경우에, 평균에러지속시간(mean burst error length(=1/r))에 대한 수율을 나타낸 것이다. 평균에러지속시간이 짧은 경우에는 V-BGBN이 가

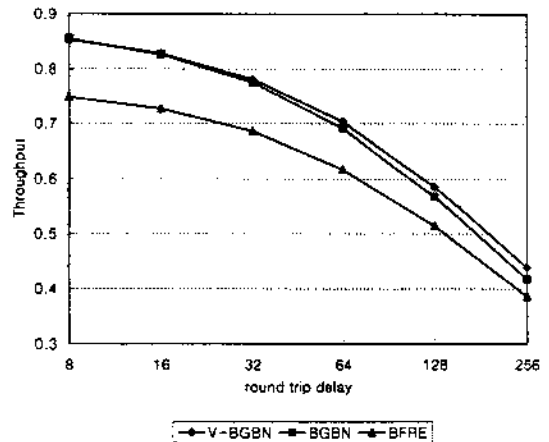


그림 5. Round trip delay에 대한 수율의 변화(N=256, r=0.25, q=10⁻³)

장 좋은 성능을 보이고 있으며, 평균에러지속시간이 긴 경우에는 V-BGBN과 BGBN이 유사한 성능을 보이고 있다.

그림 5는 round trip delay(R)에 대한 수율을 나타낸 것이다. round trip delay가 긴 경우 V-BGBN이 가장 좋은 성능을 보이고 있다.

그림 6은 R=32인 경우의 블록의 크기 N에 대한 수율

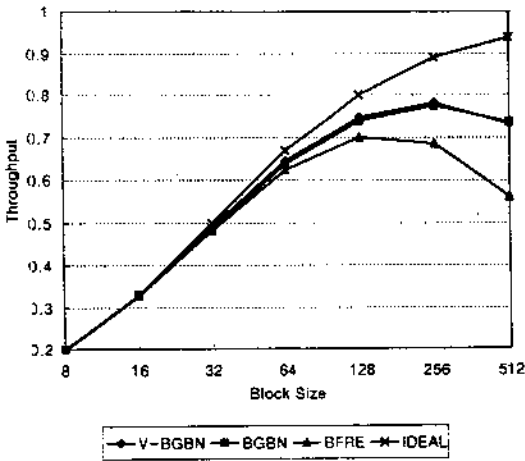


그림 6. 블록 크기에 대한 수율의 변화($r=32$, $\tau=0.25$, $q=10^{-3}$)

을 나타내고 있다. 그림 6에서 IDEAL은 주어진 조건하에서의 에러가 발생하지 않는 경우의 최대 수율을 나타낸 것이다. 그림 6에서 본 바와 같이, 주어진 조건하에서의 최적 블록 크기가 존재함을 알 수 있다.

5. 결론

본 연구에서는 V-BGBN이라고 불리우는 변형된 형태의 blast 프로토콜을 제안하였다. 그리고 blast 프로토콜들을 분석하는데 용이한 순환식을 이용한 새로운 분석 방법을 제시하였다. 그리고 에러의 발생이 서로 종속적이라는 것을 반영하기 위해 패킷의 상태를 2-state Markov-chain으로 모델링하였다.

그리고 BFRE, BGBN and V-BGBN의 세가지 프로토콜에 대해서 패킷에러확률, 에러지속시간, round trip delay, 그리고 블록의 크기에 따른 수율의 영향을 살펴보았다. 패킷에러확률이 아주 작을 경우에는 3 프로토콜의 성능이 유사하였으나, 에러확률이 커짐에 따라 BFRE의 성능은 급속히 나빠짐을 알 수 있었다. 그리고 에러확률이 적고, 채널이 버스티(bursty)하지 않으며, round trip delay가 긴 경우에는 V-BGBN의 성능이 가장 뛰어난을 실험을 통하여 알 수 있었다. 따라서 위성통신과 같은 원거리 통신에 V-BGBN이 효과적임을 알 수 있었다.

본 연구에서 사용한 분석 도구는 3가지 프로토콜에 대해 약간의 변형을 통하여 모두 사용될 수 있었다. 또한 blast 프로토콜 뿐만아니라 다른 에러 제어 프로토콜에도 이 방법을 적용할 수 있을 것이다.

참고문헌

- [1] Akyildiz, I.F. and W. Liu, "A General Analysis Technique for ARQ Protocol Performance in High Speed Networks," *IEEE INFOCOM '91*, pp. 499-507, 1991
- [2] De Lima, H.M. and O.C.M.B. Duarte, "A Go-Back-N Protocol with Multicopy Retransmission for High Speed Satellite Communications," *SUPERCOMM/ICC '94*, pp.859-863, 1994
- [3] Dravida, S., "Error Control Aspects of High Speed Networks," *IEEE INFOCOM '92*, pp.272-281, 1992
- [4] Fantacci, R., "Performance Evaluation of Continuous ARQ Protocols," *IEEE Trans. Commun.*, vol. 38, pp. 773-781, Jun., 1990
- [5] Lee, T.S., "The Throughput Efficiency of Go-Back-N ARQ Schemes for Burst-Error Channels," *IEEE INFOCOM '91*, pp.773-779, 1991
- [6] Lee, I.W., T.G. Kim, D.H. Cho, C.S. Sone and H. Lee, "A Study on the Performance of Error Control Algorithms in Digital Cellular DS/CDMA Systems," *SUPERCOMM/ICC '94*, pp.908-912, 1994
- [7] Leung, C.H.C., Y. Kikumoto and S.A. Sorenson, "The Throughput Efficiency of the Go-Back-N ARQ Scheme under Markov and Related Error Structure," *IEEE Trans. Commun.*, vol. 36, pp.231-234, Feb., 1988
- [8] Morris, J.M., "On Another Go-Back-N ARQ Technique for High Error Rate Condition," *IEEE Trans. Commun.*, vol. 26, pp.187-189, Jan., 1978
- [9] Mukherjee, A., L.H. Landweber and J.C. Strikwerda, "Evaluation of Retransmission Strategies in a Local Area Network Environment," *ACM Sigmetrics 89 / Performance 89*, pp.98-107, 1989

[10] Mukherjee, A., L.H. Landweber and J.C. Strikwerda, "Simultaneous Analysis of Flow and Error Control Strategies with Congestion-Dependent Errors," *ACM Sigmetrics 90 / Performance 90*, pp.86-91, 1990

[11] Pieris, G.R. and G.H. Sasaki, "The Performance of Simple Error Control Protocols under Correlated Packet Losses," *IEEE INFOCOM '91*, pp.764-772,

1991

[12] Zwaenepoel, W., "Protocols for Large Data Transfers on Local Networks," *ACM Sigcomm. '85*, pp.22-32, 1985

97년 4월 최초 접수, 97년 8월 최종 수정

Appendix 1) [보조정리 1]의 증명

[보조정리 1]은 아래와 같은 간접증명법으로 증명할 수 있다.

1) $j=2$ 인 경우

식(6)을 다시 쓰면 아래와 같다.

$$\tau_2^k = r^{(m_{k-2})} p + r^{(m_{k-2})} q(m_{k-1} + \tau_1^k) + s^{(m_{k-2})} \tau_2^k \tag{A.1}$$

(A.1)식으로부터 τ_2^k 는 쉽게 유도할 수 있다.

$$\tau_2^k = p + q(m_{k-1} + \tau_1^k) \tag{A.2}$$

그리고 (A.1)과 (A.2)로 부터

$$\tau_2^k = p + q(m_{k-1} + \tau_1^k) \tag{A.3}$$

임을 알 수 있다. 식(A.3)로부터 τ_2^k 가 k 와는 무관함을 알 수 있다. 따라서, $\tau_2^k = \tau_2^l, 2 \leq k, l \leq N$ 이다.

식(2)을 다시 쓰면 아래와 같다.

$$\tau_2 = \pi_0 p + \pi_0 q(m_{k-1} + \tau_1^k) + \pi_1 \tau_2^k \tag{A.4}$$

식(A.4)에 식(A.2)를 대입하면,

$$\tau_2 = p + q(m_{k-1} + \tau_1^k) \tag{A.5}$$

ㄱ- 된다. 위의 식(A.5)는 식(A.3)과 동일하다. 따라서, $\tau_2^k = \tau_2^l = \tau_2, 2 \leq k, l \leq N$ 라고 할 수 있다.

2) [보조정리 1]이 $j=n-1$ 인 경우에 맞다고 가정하면, 식(3)으로부터 아래와 같이 표현할 수 있다.

$$\tau_{n-1}^k = \tau_{n-1}^l = \tau_{n-1} = p^{n-2} + p^{n-3} q(m_{n-2} + \tau_1^{n-1}) + \sum_{i=3}^{n-1} p^{n-1-i} q \tau_{i-1} \tag{A.6}$$

3) $j=n$ 인 경우

식(A.6)의 결과를 식(7)에 대입하면 아래와 같이 표현된다.

$$\tau_n^k = r^{(m_{k-n})} p^{n-1} + r^{(m_{k-n})} p^{n-2} q(m_{n-1} + \tau_1^n) + \sum_{i=3}^n r^{(m_{k-n})} p^{n-i} q \tau_{i-1} + s^{(m_{k-n})} \tau_n^k \tag{A.7}$$

$k=n$ 으로 두고 식(A.7)을 정리하면,

$$\tau_n^n = p^{n-1} + p^{n-2}q(m_{n-1} + \tau_1^n) + \sum_{i=3}^n p^{n-i}q \tau_{i-1} \tag{A.8}$$

이 된다. 그리고 식(A.8)을 식(A.7)에 대입하여 정리하면,

$$\tau_n^k = p^{n-1} + p^{n-2}q(m_{n-1} + \tau_1^n) + \sum_{i=3}^n p^{n-i}q \tau_{i-1} \tag{A.9}$$

이 된다. 그런데 위의 식(A.9)은 식(A.8)과 동일하다. 따라서, $\tau_n^k = \tau_n^l$, $2 \leq k, l \leq N$ 라고 할 수 있다. 그리고, 식(A.6), (A.8)의 결과를 식(3)에 대입하면 아래와 같이 표현된다.

$$\tau_n = p^{n-1} + p^{n-2}q(m_{n-1} + \tau_1^n) + \sum_{i=3}^n p^{n-i}q \tau_{i-1} \tag{A.10}$$

위의 식(A.10)은 식(A.9)과 동일하다. 따라서, $\tau_n^k = \tau_n^l = \tau_n$, $2 \leq k, l \leq N$ 라고 할 수 있다.
(증명 끝)

Appendix 2) [보조정리 2]의 증명

[보조정리 1]의 결과를 식(2)에 대입하면,

$$\tau_2 = p + q(m_1 + \tau_1^2) \tag{A.11}$$

와 같이 표현된다. 따라서 $\tau_2 = A_2$ 임을 알 수 있다.
식(12)는 아래와 같은 간접증명법으로 증명할 수 있다.

1) $j=3$ 인 경우
식(A.10)으로부터,

$$\tau_3 = p^2 + pq(m_2 + \tau_1^3) + q \tau_2 \tag{A.12}$$

을 유도할 수 있다. 따라서, $\tau_3 = A_3 + qA_2$ 라고 할 수 있다.

2) [보조정리 2]가 $j=n-1$ 인 경우에 맞다고 가정하면, 아래와 같이 표현할 수 있다.

$$\tau_{n-1} = A_{n-1} + q \sum_{i=2}^{n-2} A_i \tag{A.13}$$

3) $j=n$ 인 경우
식(A.10)과 (A.13)으로부터 아래의 식을 유도할 수 있다.

$$\bar{\tau}_n = A_n + p^{n-3} q A_2 + \sum_{k=4}^n p^{n-k} q [A_{k-1} + q \sum_{i=2}^{k-2} A_i] \tag{A.14}$$

식(A.14)를 다시 정리하면,

$$\bar{\tau}_n = A_n + q \sum_{k=3}^n A_{k-1} [p^{n-k} + q \sum_{i=k+1}^n p^{n-i}] \tag{A.15}$$

로 표현할 수 있다. 따라서 $\tau_i = A_i + q \sum_{k=2}^{i-1} A_k$ 로 나타낼 수 있다.
(증명 끝).

Appendix 3) [보조정리 3]의 증명

먼저, (N+1)번째 패킷의 위치확률 $f_{N+1,i}$ 에 대한 순환식을 세워보았다.

$$\begin{aligned} f_{N+1,1} &= \pi(f_{N+1,1} | Y_1=1) + \pi q^{N-1} \\ f_{N+1,2} &= \pi(f_{N+1,2} | Y_1=1) + \pi q^{N-2} q \\ &\vdots \\ f_{N+1,i} &= \pi(f_{N+1,i} | Y_i=1) + \pi q^{N-i} q \\ &\vdots \\ f_{N+1,N} &= \pi(f_{N+1,N} | Y_N=1) + \pi q \end{aligned} \tag{A.16}$$

위의 식에서 $(f_{N+1,i} | Y_i=1)$ 는 첫 번째 패킷에서 에러가 발생하였을 경우, (N+1)번째 패킷이 블록내의 i번째 슬롯에 위치할 확률을 나타낸다. 따라서 아래와 같이 표현할 수 있다.

$$\begin{aligned} (f_{N+1,1} | Y_1=1) &= s^{(m_0)} (f_{N+1,1} | Y_1=1) + r^{(m_0)} p^{N-1} \\ (f_{N+1,i} | Y_i=1) &= s^{(m_0)} (f_{N+1,i} | Y_i=1) + r^{(m_0)} p^{N-i} q, \quad i=2, \dots, N \end{aligned} \tag{A.17}$$

식(A.16)과 (A.17)로 부터 (N+1)번째 패킷의 위치확률을 구할 수 있다.

$$\begin{aligned} \hat{f}_{N+1,1} &= p^{N-1} \\ \hat{f}_{N+1,i} &= p^{N-i} q, \quad i=2, \dots, N \end{aligned} \tag{A.18}$$

그리고 n번째 패킷의 위치확률 $f_{n,i}$ 는 (n-1)번째 패킷의 위치확률 $f_{n-1,i}$ 의 조건부확률(conditional probability)로 나타낼 수 있다.

$$\begin{aligned} f_{n,i} &= f_{N+1,i} + f_{n-1,N} \\ f_{n,i} &= f_{n-1,i-1} + f_{N+1,i} + f_{n-1,N}, \quad i=2, \dots, N \end{aligned} \tag{A.19}$$

위의 식(A.19)는 $f_{n,i}$ 의 n에 대한 차분방정식(difference equation)의 형태를 띠고 있다. 식(A.19)를 행렬 형태로 나타내면 아래와 같다.

$$\begin{bmatrix} f_{n,1} \\ f_{n,2} \\ f_{n,3} \\ \vdots \\ f_{n,i} \\ \vdots \\ f_{n,N} \end{bmatrix} = \begin{bmatrix} 0 & 0 & \cdots & 0 & p^{N-1} \\ & 1 & 0 & \cdots & 0 & p^{N-2}q \\ & & 0 & 1 & \cdots & 0 & p^{N-3}q \\ & & & \vdots & \vdots & \vdots \\ & & & & 0 & 0 & \cdots & 0 & p^{N-i}q \\ & & & & & \vdots & \vdots & \vdots \\ & & & & & & 0 & 0 & \cdots & 1 & q \end{bmatrix} \begin{bmatrix} f_{n-1,1} \\ f_{n-1,2} \\ f_{n-1,3} \\ \vdots \\ f_{n-1,i} \\ \vdots \\ f_{n-1,N} \end{bmatrix} \tag{A.20}$$

식(A.20)의 일반해(general solution)는 식(14)의 형태로 표현된다.

식(14)의 정방행렬(square matrix)을 A라고 하자. 행렬 A의 모든 열의 합은 1이므로, 행렬 A의 주 고유값(dominant eigenvalue)은 1이다. 그리고 E_r 과 E_l 을 각각 행렬 A의 우측 고유벡터(right eigenvector)와 좌측 고유벡터(left eigenvector)라고 하면, 아래와 같이 표현할 수 있다.

$$\lim_{n \rightarrow \infty} A^n = \frac{1}{E_r \cdot E_l} [E_r \ E_l] \tag{A.21}$$

$$\begin{aligned}
 \text{여기에서 } E_r &= (p^2 \ p \ 1)^T \\
 E_l &= (1 \ 1 \ 1)
 \end{aligned}$$

그리고 극한확률 f_i 는 아래와 같이 정의할 수 있다.

$$f_i = \lim_{n \rightarrow \infty} A^n \tag{A.22}$$

따라서, 식(15)는 식(A.21)과 (A.22)를 통해서 유도된다.
(증명 끝)