

## 블록 FEC알고리즘을 적용한 802.11e MAC 성능평가

윤종혁<sup>o</sup> 안중석  
 동국대학교 컴퓨터공학과  
 {ronaldo<sup>o</sup>, jahn}@dgu.edu

### The Performance Analysis of IEEE 802.11e MAC with a Block FEC Scheme

Jonghyuk Yoon<sup>o</sup>, Jongsuk Ahn  
 Department of Computer Engineering, Dongguk University

#### 요 약

본 논문에서는 블록 FEC(Forward Error Correction) 알고리즘을 적용한 IEEE 802.11e MAC(Medium Access Control)의 블록전송모드(Block Transfer Mode)의 성능을 분석한다. 성능 향상을 위해서 IEEE 802.11e MAC은 TXOP(Transmission Opportunity)시간 동안 특정 스테이션(station)에게 독점적인 전송 기회를 부여하며, 이 기간동안 하나의 블록 즉 연속적으로 여러 개의 데이터 프레임(frame)을 전송한다. 수신자는 블록내의 각 프레임에 수신 여부를 블록 ack (block Acknowledgment)방식, 즉 하나의 ack 패킷이 모든 수신 프레임의 일련번호를 모아 알려 준다. 그러나 오류가 빈번한 무선 채널 환경에서는 블록 ack 방식을 사용하더라도 빈번한 재전송으로 인해 채널의 성능이 현저히 감소한다. 이러한 문제를 해결하기 위해 심볼 단위로 오류를 복원하는 블록 FEC 알고리즘을 적용할 수 있는데, 본 논문에서는 802.11e에서 FEC 알고리즘을 적용하는 두 가지 방식의 성능을 해석적으로 분석한다. 즉 두 가지 방식은 프레임 당 블록 FEC를 적용하는 방법과 전체 블록 내의 프레임들 하나의 데이터로 간주하여 FEC를 적용하는 방식이다. 본 논문에서는 이 둘 두 가지 방식의 채널 성능 향상 정도와 블록 ack만을 사용한 경우의 성능을 수식적으로 계산한다. 실험에 의하면 BER이  $10^{-4}$ 인 채널 환경에서 블록 FEC를 적용한 방식이 블록 ack를 사용한 방식에 비해 약 1.5%의 성능향상을 보인다.

#### 1. 서 론

최근 표준화된 IEEE 802.11e MAC[4]은 Legacy MAC에 비해 QoS(Quality of Service)의 지원과 블록 ack 방식을 통해 54Mbps 이상의 고속 전송 속도에서의 MAC의 성능을 향상시키는 특징을 갖는다. 802.11 Legacy MAC은 빈번한 컨트롤 프레임 교환 및 DIFS(Distributed interframe space), backoff 시간, PHY header 등의 오버헤드로 인해 PHY의 전송 속도가 높아질수록 MAC의 성능은 오히려 떨어진다. 802.11e에서는 MAC의 성능을 향상시키기 위해 독점적으로 전송할 기회를 부여받은 노드가 프레임을 연속적으로 전송하고 수신 노드는 연속적으로 수신된 프레임에 대한 ack를 모아서 하나의 프레임으로 응답하는 블록 ack를 새롭게 정의하였다. 또한 한 노드가 연속적으로 프레임을 전송하는 시간을 TXOP로 정의하였으며 이 시간동안에는 다른 노드와 경쟁 없이 데이터 프레임을 SIFS(Short Interframe space)시간 간격을 두어 연속적으로 전송함으로써 Legacy MAC의 오버헤드를 줄일 수 있다. 802.11e는 2가지 모드를 제공하는데, 첫째, 경쟁을 통해 채널을 획득하는 EDCA(Enhanced Distributed Channel Access) 모드와 둘째, HC(Hybrid Coordinator)에 의해 채널을 할당받는 HCCA(Hybrid Coordinator Channel Access)모드로 구분된다. EDCA는 Legacy MAC의 DCF(Distributed Coordination Function)를 HCCA는 PCF를 확장한 방식이다.

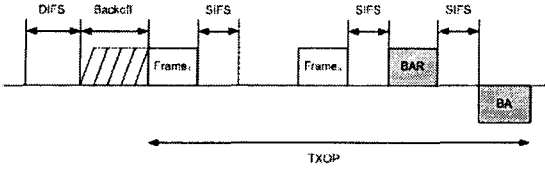
무선 채널의 평균 BER(Bit Error Rate)은  $10^{-6} \sim 10^{-3}$ 으로 평가되고 있어 전파 오류방지 또는 오류 복구 기법을 사용하지 않을 경우 손상된 패킷의 재전송으로 인해 전송 효율이 감소한다. 이러한 문제를 해결하기 위해 무선 네트워크 MAC 계층에서는 FEC를 사용하여 오류를 복구하는 방식을 이용한다. FEC는 전

송할 데이터에 추가적으로 정정 부호를 더하여 수신측에서 정정 부호를 이용하여 오류 패킷을 복구하는 방식이며, 근접적으로 오류가 발생하는 무선채널에서는 RS(Reed Solomon) 알고리즘 [3]과 같은 블록 FEC알고리즘을 이용하여 오류 복구율을 높이는 것이 효율적이다.

본 논문에서는 802.11e MAC의 블록전송모드에 RS 알고리즘을 적용한 2가지 방식에 대한 성능을 수식적으로 계산하고, 센서 노드에 적용하여 FEC알고리즘을 사용하지 않은 방식과의 성능을 비교한다. 2장에서는 EDCA하에서의 블록전송모드를 설명하고, 3장에서는 RS 알고리즘을 블록전송모드에 적용하는 방법에 대해 기술한다. 4장에서는 RS FEC알고리즘을 적용하였을 때 채널의 성능을 수식적으로 계산하며, 5장에서 FEC를 적용한 블록전송모드와 FEC를 적용하지 않은 블록전송모드의 성능을 비교 분석한다. 6장에서는 결론을 기술한다.

#### 2. 802.11e MAC에서 블록 전송 모드

[그림 1]은 802.11e EDCA 블록전송모드의 동작 방식을 보여준다. 백오프 시간 후 경쟁을 통해 채널을 할당받은 노드는 TXOP 기간 동안 전송한 프레임에 대한 ack를 기다리지 않고 SIFS 간격을 두고 연속적으로 데이터 프레임을 전송한다. 프레임들을 블록으로 전송 한 후에는 BAR(Block Ack Request)프레임을 보내어 수신측에 이전까지 보낸 모든 프레임에 대한 ack를 하나의 프레임에 모아서 전송하도록 요청한다. BAR를 받은 수신 노드는 SIFS시간 후 각 프레임의 재전송 여부를 비트맵을 이용하여 BA(Block Ack) 프레임으로 응답한다. BA를 받은 전송노드는 재 전송 해야 할 프레임들 선택하여 현재 TXOP 또는 다음 TXOP에 전송한다.



[그림 1] 802.11e EDCA의 블록전송모드

임을 전송하는 시간을  $T_{bar}$ 은 BAR프레임을 전송하는 시간을  $T_{ba}$ 는 BA프레임을 전송하는 시간을  $\delta$ 는 전파 지연 (Propagation Delay)을 의미한다.

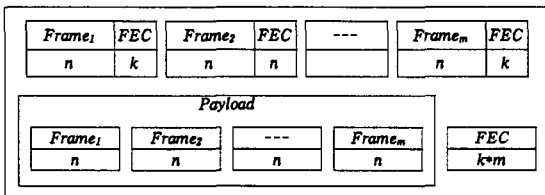
$$\begin{aligned}
 T_I &= \sigma \\
 T_S &= N_b(T_f + T_{SIFS}) + T_{DIFS} + (T_{bar} + T_{SIFS} + T_{ba}) + \\
 &\quad (N_b + 2)(T_{PHYhdr} + \delta) \\
 T_E &= T_S \\
 T_C &= N_b(T_f + T_{SIFS}) + T_{EIFS} + (T_{bar} + T_{SIFS} + T_{ba}) + \\
 &\quad (N_b + 1)(T_{PHYhdr} + \delta) \quad (2)
 \end{aligned}$$

### 3. RS code를 적용한 802.11e MAC

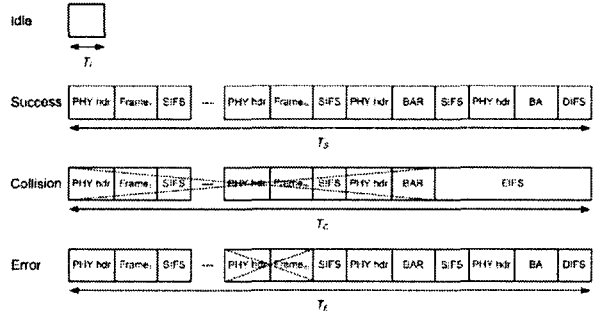
본 절에서는 RS FEC 알고리즘을 블록전송모드에 적용하는 2가지 방법에 대해 기술한다. 2가지 방식 모두 FEC 코드 사이즈는 8비트를 적용하며, 블록전송모드동안 사용하는 전체 FEC 코드량은 같다. [그림 2]는 FEC 적용 방식에 따른 프레임의 모습을 보여주고 있다.

첫 번째 방식은 한 TXOP 동안에 할당된 전체 FEC 코드량  $T$ 비트를 프레임 개수  $m$ 으로 나누어 각 프레임별로 RS FEC 알고리즘을 적용하고, 프레임에  $k$ 바이트의 FEC 코드를 추가하여 전송하는 방식이다. 이 때 프레임 별로 정정 가능한 바이트 수는  $k/2$ 개 이다.

두 번째 방식은 TXOP 동안에 전송할 모든 프레임을 하나의 데이터로 간주하여 FEC 알고리즘을 적용하고, FEC 코드는 모든 프레임 전송 후 별도의 프레임으로 전송한다. 모든 프레임들 중에서 정정 가능한 바이트 수는  $(k*m)/2$  이다.



[그림 2] RS FEC알고리즘을 적용한 블록전송모드의 프레임 형태



[그림 3] 802.11e 블록전송모드의 시간간격

슬롯 시간의 확률을 구하기 위해서 한 슬롯에서 노드가 전송할 확률을  $\tau$ 로, 노드의 수를  $n$ 으로 나타낸다.  $\tau$ 는 경쟁 윈도우 사이즈가 2배로 되는 확률  $p$ 로 기술되는데, 본 논문에서는  $\tau$ 값을 제한하기 위하여 [2]에서 적용한 마코프 체인(Markov Chain)을 이용한다.

식(3-6)은 [그림 3]의 4가지 경우의 확률을 보여준다. Idle 슬롯에서 한 슬롯이 전송을 시도하지 않을 확률은  $(1 - \tau)$ 이므로  $n$ 개 노드가 전송을 하지 않을 확률  $P_I$ 는 식 (3)과 같다. 성공적으로 전체 프레임을 전송할 확률은 식 (4)과 같다. 이 때  $p_e^{bit}$ 는 전체 블록에서 한 노드의 오류 확률을 의미하는데, 식 (7)과 같이 계산된다. 한 개 프레임 이상의 오류 확률은 식 (5)와 같다. 충돌 확률은 식(6)과 같다.

$$P_I = (1 - \tau)^n \quad (3)$$

$$P_S = n(\tau(1 - \tau)^{n-1})(1 - p_e^{bit}) \quad (4)$$

$$P_E = n(\tau(1 - \tau)^{n-1})p_e^{bit} \quad (5)$$

$$P_C = 1 - P_I - P_S - P_E \quad (6)$$

식 (7)은 블록의 에러 확률을 의미하며  $p_e$ 는 패킷 손실 확률(PER)을  $N_b$ 는 한 블록내의 프레임수를 의미한다.

$$p_e^{bit} = 1 - (1 - p_e)^{N_b} \quad (7)$$

FEC코드 크기가  $T$ 비트이고, FEC 코드를 포함한 전체 전송 데이터 크기가  $N$ 비트이며, FEC의 심볼 크기가  $m$ 비트일 때  $n = N/m$ ,  $t = T/m$ 이며 이 FEC코드로 복원할 수 있는 심볼 개수는  $t/2$ 이다. 또한 전송 비트 수를  $c$ , BER을  $p_b$ 라 할 때 FEC 심볼 오류확률  $p_f$ 는 식 (8)과 같다. BER이  $p_b$ 일 때 하나의 FEC

### 4. RS 알고리즘을 적용한 802.11e 성능 계산

본 절에서는 [1]를 기반으로 하여 FEC를 적용한 블록전송모드의 성능  $S$ 를 수식적으로 계산한다. 식 (1)은 성능 계산식을 보여 주는데, 성능  $S$ 는 예상 슬롯 시간  $E[T]$  동안 성공적으로 전송되어진 프레임의 payload 크기  $E[L_{pld}]$  로 계산된다.

$$S = \frac{E[L_{pld}]}{E[T]} \quad (1)$$

예상 슬롯 시간  $E[T]$ 을 구하기 위해서 [그림 3]과 같은 4가지 경우를 고려한다. [그림 3]에서  $T_I$ 는 idle 슬롯 시간을 의미하며 한 슬롯 간격  $\sigma$ 값을 갖는다.  $T_S$ 는 성공적으로 블록을 전송한 시간을,  $T_C$ 는 채널 충돌로 인해 전체 프레임이 오류가 발생한 시간을,  $T_E$ 는 한 프레임 이상이 오류로 인해 재전송해야 하는 시간을 나타낸다. 한 블록의 프레임수를  $N_b$ , 한 프레임을 전송하는 시간을  $T_f$  할 때, [그림 3]의 4가지 경우에 따른 슬롯 시간은 식 (2)와 같다.  $N_b$ 는 한 블록의 프레임수를  $T_f$ 는 함 프레

심볼에 오류가 발생할 확률  $p$ 는  $c$ 개의 비트 중에 어느 한 개 비트에 오류가 발생할 확률이다.

$$p_f = 1 - (1 - p_b)^c \quad (8)$$

식 (9)는 패킷 오류확률을 보여주는데 RS FEC 알고리즘을 사용하는 경우  $t$ 개의 FEC 코드 심볼을 포함한  $n$ 개의 FEC 심볼을 포함하는 패킷 오류 확률로 결정되고, RS FEC 알고리즘을 사용하지 않는 경우는  $p_b$ 에 의해서 패킷 오류 확률이 결정된다.

$$p_e = \begin{cases} 1 - \sum_{k=0}^{t/2} \binom{n}{k} (p_f)^k (1 - p_f)^{n-k} & (\text{FEC사용}) \\ 1 - (1 - p_b)^c & (\text{FEC미사용}) \end{cases} \quad (9)$$

최종적으로 성능  $S$ 는 식 (10)와 같이 계산된다. 식 (10)에서  $E[L]$ 은 오류가 발생한 경우에 성공적으로 전송된 프레임 크기를 나타내며 식 (11)과 같이 계산된다.

$$S = \frac{P_s N_s c + P_e E[L]}{P_f T_f + P_s T_s + P_e T_e + P_c T_c} \quad (10)$$

$$E[L] = \sum_{i=0}^{N_b} \binom{N_b}{i} (p_e)^i (1 - p_e)^{N_b - i} c \quad (11)$$

### 5. RS 알고리즘을 적용한 802.11e 성능 분석 비교

RS FEC 알고리즘을 적용하기 위해 2가지 방식 모두 FEC 심볼 사이즈는 8비트를 전체 FEC 코드량은 16바이트를 사용한다. 한 프레임의 크기는 MAC 헤더를 포함하여 30 바이트, TXOP동안 연속적으로 보내는 프레임 수는 4개로 가정한다. 이에 따라 개별 프레임마다 FEC 알고리즘을 적용하는 경우는 프레임 당 4(=16/4)바이트의 FEC 코드를 사용하고, 블록의 전체 프레임에 FEC 알고리즘을 적용하는 경우는 모든 프레임 전송 후 16바이트의 FEC코드를 전송한다. 또한 성능비교를 위해 FEC를 적용하지 않은 프레임 크기는 34바이트를 사용한다.

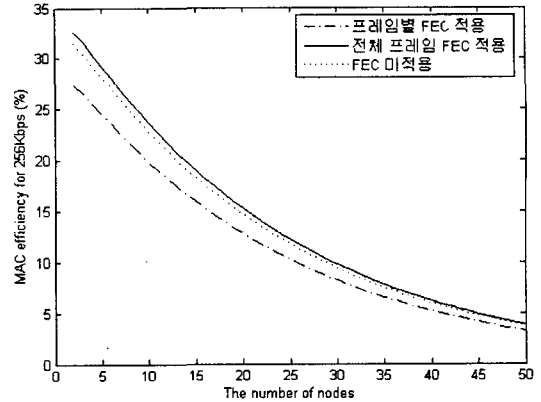
[표 1]은 성능분석에 사용된 802.15.4프로토콜을 사용하는 TIP50CM[5] 센서 노드의 Physical layer 및 MAC layer 패러미터 값[6]을 보인다.

Frame Size	30바이트
PHY header	125 $\mu$ s
BAR	24바이트
BA	152바이트
PHY rate	256Kbps
Propagation delay	1 $\mu$ s
Slot time	60 $\mu$ s
SIFS	93 $\mu$ s
DIFS	186 $\mu$ s

[표 1] TIP50CM의 PHY와 MAC 레이어 패러미터

[그림 4]는 BER이  $10^{-4}$ 인 환경에서 전송 노드 수 증가에 따라 RS FEC 알고리즘을 적용한 2가지 방식과 적용하지 않은 MAC의 성능차이를 보여준다. [그림 4]에 의하면 전송 노드 수 증가에 따라 MAC의 성능이 급격히 감소함을 볼 수 있다. 또한

RS FEC 알고리즘을 블록 내 전체 프레임에 적용한 방식이 가장 높은 성능을 보이며, 프레임별로 RS FEC알고리즘을 적용한 방식이 가장 낮은 성능을 보인다.



[그림 4] RS FEC알고리즘을 적용한 블록전송모드와 FEC를 적용하지 않은 블록전송모드의 성능 비교

### 6. 결론 및 향후 연구과제

본 논문에서는 RS FEC알고리즘을 적용한 802.11e MAC의 블록전송모드의 성능을 수식적으로 계산하고, 이를 저속의 센서 네트워크에 적용할 때의 성능 차이를 비교하였다. 실험에 의하면 TXOP 시간 동안 전송할 모든 프레임은 하나의 데이터로 간주하여 FEC를 적용한 방식이 FEC를 적용하지 않은 방식에 비해 약 1.5%의 성능향상을 보였으며, 개별 프레임에 FEC를 적용한 방식은 FEC를 적용하지 않은 방식에 비해 약 4.8%의 성능감소를 보였다.

향후과제로는 FEC 심볼 오류확률에 2-state 마코프 체인을 기반으로 한 길버트 모델(Gilbert model)을 적용하여 FEC심볼의 상태에 따른 전이확률을 추가하여 실제 네트워크의 특성이 가깝게 분석할 것이다.

### 참 고 문 헌

- [1] Tianji Li, Qiang Ni, Thierry Turletti and Yang Xiao, "Performance Analysis of the IEEE 802.11e Block ACK Scheme in a Noisy Channel", IEEE BroadNets 2005, Wireless Networking Symposium, Boston, MA, USA, October 3-7, 2005.
- [2] Haitao Wu, Yong Peng, Keping Long, Shiduan Cheng and Jian Ma, "Performance of Reliable Transport Protocol over IEEE 802.11 Wireless LAN: Analysis and Enhancement", IEEE INFOCOM 2002.
- [3] I. S. Reed and G. Solomon, "Polynomial codes over Certain Fields", J. Soc. Ind. Appl. Math., 8: 300-304, June 1960
- [4] IEEE Std 802.11.e-2005 Amendment 8: Medium Access Control (MAC) Quality of Service Enhancements
- [5] <http://www.maxfor.co.kr/maxfor/gnu3/files/TIP50C.alz>
- [6] IEEE Std 802.15.4-2003