

NS-2를 이용한 IEEE 802.16d ARQ 알고리즘 성능분석

정회원 장재신*, 준회원 강민석**

Performance Evaluation of IEEE 802.16d ARQ Algorithm with NS-2 Simulator

Jaeshin Jang* *Regular Member*, Min-seok Kang** *Associate Member*

요 약

본 연구에서는 송신 측이 전송한 데이터를 오류 없이 수신하기 위해 IEEE 802.16d에서 채택하고 있는 4가지 ARQ 기법을 전송오류가 빈번하게 발생하는 무선채널환경에서 성능을 분석하였다. NS-2 네트워크 시뮬레이터를 사용하여 성능평가를 위한 시뮬레이터를 구현하였고, 성능평가의 척도로 시뮬레이션 수행시간 동안 수신 측이 송신 측으로부터 오류 없이 순서대로 수신한 총 ARQ 블록의 바이트 수와 본 연구에서 새롭게 정의한 ACK_Efficiency를 사용하였다. 성능평가 결과에 따르면 각 ARQ 기법이 다양한 무선채널환경에 따라 서로 상이한 성능을 제공함을 알 수 있었으며, 특정 채널환경에서 데이터 전송률을 높이기 위해서는 채널상태에 알맞은 ARQ 기법을 선택해야 하고, ARQ 기법의 적절한 ACK 전송주기를 선정하는 것이 중요함을 알 수 있었다.

Key Words : 휴대인터넷, IEEE 802.16d, ARQ, NS-2, WMAN

ABSTRACT

In this paper, we evaluated the performance of 4 types of ARQ algorithms that have been adopted to offer reliable transmission between MAC layers. First of all, we implemented IEEE 802.16 ARQ algorithms with NS-2 network simulator and then evaluated the performances in various channel states. We used two performance measures to evaluate those performances. First, we used the total number of ARQ block in bytes that have been received without error. Second, we used ACK_Efficiency that is newly defined in this paper. From the results of simulation, we could be aware that each ARQ algorithm has different performance in a various channel states. For improving the transmission data rate in a specific channel state, we have to choose the right ACK interval as well as a suitable ARQ algorithm.

1. 서론

현대 사회에서 인터넷은 우리 생활과 밀접한 관계를 맺고 있으며 인터넷과 떨어져 생활한다는 것은 이제 생각할 수도 없는 단계에 이르렀다. 이러한 욕구를 충족시키기 위해 이동 통신 시스템과 무선 LAN을 이용한 무선 인터넷이 발달하게 되었다. 그러나 이동 통신 시스템을 이용한 무선 데이터 통신

은 빠른 이동성을 제공하는 장점을 가지고 있지만 전송 속도가 낮고 통신요금이 비싼 단점을 가지고 있으며, 무선 LAN을 이용한 무선 데이터 통신은 전송 속도는 높지만 이동성을 제공하지 못하는 단점을 가지고 있다. WMAN 표준은 이러한 두 가지 서비스의 장점을 서로 결합한 기술로 2006년 하반기 상용화가 예상되는 WiBro 기술의 기반이 된다.

초기 광대역 무선통신 시스템 표준은 IEEE 802.16

* 인제대학교 정보통신공학과 조교수 (icjoseph@inje.ac.kr), ** 인제대학교 대학원 전자정보통신공학과 석사과정
논문번호 : KICS2006-02-072, 접수일자 : 2006년 2월 7일, 최종논문접수일자 : 2006년 7월 19일

으로 WMAN의 커버리지를 가지면서 PHY 계층으로는 10~66GHz 대역의 SC(Single Carrier)를 사용한다. 그 후 2~11 GHz 대역에서의 SCa, OFDM (Orthogonal Frequency Division Multiplexing), OFDMA (Orthogonal Frequency Division Multiple Access)를 추가한 IEEE 802.16a가 발표되었고 이 동성을 제공하는 IEEE 802.16e 표준과 호환성을 보장하기 위해 새롭게 MAC과 PHY를 수정한 IEEE 802.16d 표준이 제정되었다. WMAN 표준은 크게 두 가지 모드로 동작을 나눌 수 있다. 첫째는 한 개의 BS(Base Station)에 한 개 이상의 SS (Subscriber Station)들이 통신하는 PMP (Point-to-Multipoint) 모드이며 둘째는 SS들 간에 자유롭게 통신 할 수 있는 메쉬(옵션) 모드이다¹⁾. PMP 모드는 BS가 SS들에게 통신하는 순방향 링크와 여러 SS들이 BS에게 통신하는 역방향 링크로 나뉘는데 듀플렉싱 방법에 따라 TDD(Time Division Duplexing)와 FDD(Frequency Division Duplexing) 방식으로 나뉜다. 순방향링크의 데이터는 브로드캐스트 방식으로 모든 SS들에게 전달되며 데이터를 수신한 SS는 자신의 CID(Connection Identifier)로 전달된 데이터만 수신하는 방식을 채택하고 있으나 역방향은 이와 달리 On-demand 방식을 채택하고 있다. 역방향 링크를 통해 데이터를 전송하기 위해서는 먼저 BS로부터 전송 권한을 얻어야 한다. 전송 권한은 순방향 링크의 첫 번째 서브프레임에 UL-MAP 메시지를 통하여 전달된다. 권한을 얻는 방식은 SS의 서비스 방식에 따라 달라지는데, 예로 들면 rtPS 서비스에서는 먼저 BS와 SS가 PMP모드로 동작하기 위해서 초기 레인징(Initial ranging) 과정과 등록 과정 등의 시그널링을 통해 상호 PHY정보와 서비스 정보를 교환하고 협상하며 이 과정에서 BS는 SS에게 주기적인 폴링 인터벌을 설정하게 된다. 폴링 인터벌이란 해당 SS가 전송할 데이터를 보유하고 있는지 BS가 체크하는 주기를 나타내며, 폴링을 받은 SS는 전송할 데이터가 있을 경우 대역 할당을 요구하고 이를 성공적으로 처리한 BS는 다음 서브프레임의 순방향 링크 전송시간에 해당 SS가 데이터를 전송할 수 있게 대역을 할당한다. 현재 IEEE 802.16관련 연구는 꾸준히 진행 중이며 참고문헌⁴⁾에서는 IEEE 802.16과 IEEE 802.20의 표준화 동향과 기술방식에 대해 자세히 설명하고 있다. 참고문헌^{2, 3)}에서는 QoS(Quality of Service)를 보장하기 위한 연구에 관한 논문들로 IEEE 802.16d에서 규정하고 있는 4가지 서비스 방식 UGS(Unsolicited Grant Service),

rtPS(Real-time Polling Service), nrtPS (Non-Real-Time Polling Service) 그리고 BE(Best Effort)에 따른 스케줄러의 요구조건을 규정하고 있다.

본 연구에서는 IEEE 802.16d의 MAC 규격에서 규정한 4가지 ARQ 알고리즘의 성능을 비교 분석하였다. ARQ 알고리즘 성능평가를 위해 NS-2 시뮬레이터⁵⁾를 사용하여 4가지 ARQ 알고리즘을 구현하였다. 2장에서는 IEEE 802.16d의 ARQ 기법에 대해 기술하고 3장에서는 ARQ 성능 분석을 위한 시뮬레이션 환경과 성능평가 결과에 대해서 기술한 뒤, 4장 결론으로 끝맺는다.

II. IEEE 802.16d ARQ 기법

IEEE 802.16d의 ARQ 알고리즘은 MAC 계층에서 오류 없이 데이터를 송수신하기 위해 채택된 규격으로 ARQ의 사용은 옵션으로 지정되어있다. ARQ는 연결 기반으로 동작하며 ARQ 파라미터들은 연결 설정 과정에서 결정된다. 주요 ARQ 파라미터로는 전송한 ARQ 블록의 재전송까지의 시간을 규정하는 ARQ_RETRY_TIMEOUT과 전송한 블록을 송신 측에서 처리할 수 있는 최대시간을 규정하는 ARQ_BLOCK_LIFETIME, 그리고 송신 측에서 ACK 없이 전송할 수 있는 최대 블록 수를 규정한 ARQ_WINDOW_SIZE 등이 있다. 한 개의 연결에는 ARQ와 non-ARQ 연결이 동시에 존재 할 수 없으며 각 ARQ 방식은 단 방향 연결에 대한 동작만을 규정하고 있다. ARQ기법이 활성화된 연결에 대해 데이터를 전송할 경우 Fragmentation은 옵션으로 지정되어 있다. Fragmentation을사용하는 경우 송신측은 상위 MAC SDU를 ARQ_BLOCK_SIZE의 단위로 나뉘어 전송하며, 이 값은 BS와의 연결 생성과정에 TLV(Type/Length/Value)값으로 규정하게 되어 있다. 그리고 Fragmentation을 사용하지 않는 경우의 동작은 Fragmentation 사용하는 것과 유사하게 동작하며, ARQ_BLOCK_SIZE와 무관하게 상위 MAC SDU를 모두 전송한다. 송신측이 전송한 ARQ 블록에 대해 수신측은 ARQ feedback IE를 사용하여 ACK 정보를 전송하며, ACK 정보는 독립된 MAC 프레임에 사용하거나(Standalone기법), Piggyback방식에 의해 데이터 MAC 프레임에 같이 포함되어 전송한다. ARQ feedback IE format의 주요 정보로는 현재 ARQ 알고리즘이 사용되는 연결 ID를 나타내는 CID, 현재 사용되는 ACK 알고리즘 나타내는 ACK type, 그리고 BSN(Block Sequence

Number)이 있으며 해당 ACK 블록 전송 성공 여부를 비트맵으로 나타내는 Selective ACK Map 정보가 최대 4개까지 한 ARQ feedback IE에 포함될 수 있다. 규격에서 사용하는 ARQ 종류는 4가지가 있으며 각 ARQ 방식에 따라 수신측이 전송하는 Feedback IE 정보의 의미가 조금씩 달라진다. 첫째 Cumulative ACK방식은 송신 측에서 전송한 ARQ 블록 중 전송 오류 없이 순서대로 수신한 값의 최대를 ARQ feedback IE의 BSN값으로 알려주며 BSN값 이전의 블록은 모두 성공적으로 수신이 완료하였음을 의미한다. 둘째 Selective ACK 방식의 BSN값은 16비트 Selective ACK Map의 MSB(Most Significant Bit)의 값을 나타내며 ACK Map은 BSN값을 시작으로 차례대로 ARQ 블록의 전송 성공의 여부를 성공 시 '1' 실패 시 '0' 값으로 알려준다. 셋째 Cumulative with Selective ACK 방식은 앞의 두 방식은 결합한 것으로 BSN값은 Cumulative ACK 방식처럼 오류 없이 수신한 ARQ 블록의 최고 SN값을 의미하고 ACK Map은 그 이후의 ARQ 블록들의 전송 성공 여부를 나타내는 비트맵 정보를 나타낸다. 마지막으로 Cumulative ACK with block sequence ACK 방식은 Cumulative의 BSN의 의미와 ARQ 블록 수신에 대한 정보를 블록 시퀀스로 전달하는 방식이다. 블록 시퀀스 ACK Map은 해당 ARQ 블록들의 수신 성공여부를 성공 시 '1' 실패 시 '0' 값으로 나타내고, 시퀀스 길이는 동일한 상태의 ARQ블록 시퀀스의 길이를 나타낸다. 즉, 그림1에서 시퀀스 ACK MAP값 '01'의 의미는 전송오류에 의해 정상적으로 수신하지 못한 ARQ 블록을 먼저 표시하고, 성공적으로 수신한 ARQ 블록을 나중에 표시하겠다는 의미이며, 시퀀스 길이 값과 더불어 해석하면 수신 실패한 ARQ 블록이 연속해서 5개이고 성공적으로 수신한 ARQ 블록이 연속적으로 11개임을 의미한다. 블록 시퀀스 정보전달은 2개의 블록단위나 3개의 블록단위로 전달할 수 있으며 Sequence format 필드의 값을 '0'으로 했을 경우는 2개 블록 시퀀스, '1'로 했을 경우 3 블록 시퀀스를 사용한다. 그림 1에서는 2개의 블록단위를 전달하는 방식을 나타내었다. 위의 4가지 방식의 동작을 그림 1에 간단하게 나타내었다. 한 개의 ACK Map을 사용하여 송신측에 전송할 수 있는 ACK의 정보량은 4가지 기법마다 다를 수 있다. 오류 없이 순서대로 수신한 블록의 최고 BSN 값만을 전달하는 Cumulative ACK을 제외한 3가지 방식은 자신의 전달해야 하는 수신 블록 정보가 한

개의 ACK Map으로 나타내지 못할 경우 여러 개의 ACK Map를 사용하게 된다. 한 개의 ARQ feedback IE안에는 최대 4개의 ACK Map을 수용하기 때문에 4개 이상의 ACK Map을 사용할 경우는 새로운 ARQ feedback IE를 새롭게 만들어야 하며 새롭게 ARQ feedback IE를 만들 경우 블록의 수신여부를 전달하는 필드 이외 부가적인 정보(CID(16bit), LAST(1bit), BSN (11bit) 등)들도 추가된다. ACK Map를 많이 사용하게 되면 자신에게 할당되어 있는 대역폭에 부가적인 정보를 많이 전송하게 되므로 자원을 효율적으로 사용하지 못하게 된다. 즉, 그림 1에서 Selective ACK은 한 개의 ACK Map을 사용할 경우 자신이 수신한 19개의 ARQ 블록들에 대한 정보를 나타낼 수 없다. 따라서 16비트의 ACK Map과 16bit의 블록 시퀀스 필드를 한 개 사용하는 Cumulative with Selective ACK이나Cumulative ACK with block sequence 방식보다 효율성이 떨어질 수 있다.

ARQ 블록을 전송하는 송신 측에서 살펴보면 ARQ 블록은 그림 2와 같이 4가지 상태에 있게 된다. Not-sent, outstanding, discarded, 그리고 waiting-

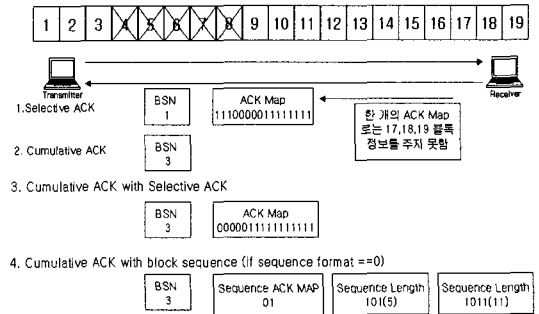


그림 1. 한 개의 ARQ Feedback IE를 사용하는 경우의 ACK 사용 예

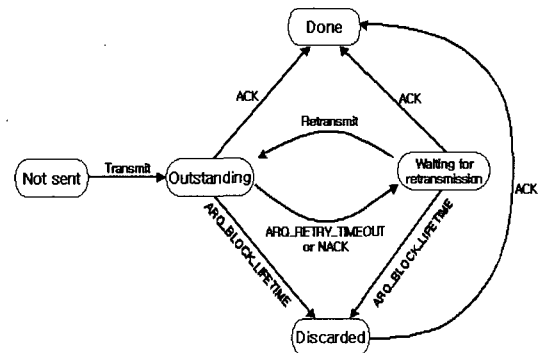


그림 2. 프레임 송신을 위한 송신 측 상태도

for-retransmission이 그것이며 모든 ARQ 블록은 Not-sent 상태에서 시작하여 전송된 이후에는 Outstanding 상태에 위치하며 ARQ_RETRY_TIMEOUT 시간동안 머물게 된다. 블록이 Outstanding 상태에 머무는 동안 해당 블록은 ACK을 수신하거나 Discarded 상태로 천이하며 ARQ_RETRY_TIMEOUT 또는 NACK에 의해서 재전송을 기다리는 상태로 천이하게 된다. 송신 측의 전송 순서는 재전송을 해야 하는 ARQ 블록이 있을 경우 먼저 전송을 하고 BSN 값이 작은 블록부터 차례로 ARQ_WINDOW_SIZE가 허락하는 범위에서 전송한다.

송신측과 수신측은 자신이 전송한 또는 수신한 ARQ 블록의 정보를 다음과 같은 변수에 저장한다. 송신측은 ARQ_TX_WINDOW_START, ARQ_TX_NEXT_BSN 변수를 이용하여 자신이 오류 없이 연속적으로 전송한 ARQ 블록의 최고 BSN값과 자신이 전송한 ARQ 블록의 최고 BSN값을 각각 저장하며 수신측은 ARQ_RX_WINDOW_START, ARQ_RX_HIGHEST_BSN을 이용하여 순서대로 오류 없이 수신한 최고 BSN값과 수신한 최고 블록의 BSN값을 저장하게 된다.

- ARQ_TX_WINDOW_START
: (ARQ_TX_WINDOW_START-1)까지의 ARQ 블록은 ACK을 수신한 BSN값.
- ARQ_TX_NEXT_BSN
: 송신측이 다음으로 전송할 블록의 BSN값이며 이 값은 ARQ_TX_WINDOW_START 보다 크고 ARQ_TX_WINDOW_START + ARQ_WINDOW_SIZE보다 작음.
- ARQ_RX_WINDOW_START
: (ARQ_RX_WINDOW_START-1)까지의 블록은 순서대로 오류 없이 수신한 BSN 값.
- ARQ_RX_HIGHEST_BSN
: 수신 측이 수신한 최고 블록의 BSN값에 1을 더한 BSN값으로 ARQ_RX_WINDOW_START 보다 크고 (ARQ_RX_WINDOW_START + ARQ_WINDOW_SIZE) 보다 작음.

위와 같은 변수들은 ARQ 연결 설정과정에서 '0'으로 초기화 된다.

III. 시뮬레이션 환경

본 연구에서는 위에서 설명한 4가지 ARQ방식을

표 1. 시스템 및 시뮬레이션 파라미터

| 시뮬레이션 파라미터 | 값 |
|---------------------|---------------------------------|
| 시뮬레이션 수행시간 | 10 s |
| 프레임 전체 길이 | 1 ms |
| 역방향 프레임 길이 | 0.5 ms |
| 순방향 프레임 길이 | 0.5 ms |
| Duplex Mode | TDD |
| Application profile | CBR (크기 : 100byte, 간격 : 0.2 ms) |
| Agent profile | UDP |
| ARQ_BSN_MODULUS | 1024 |
| ARQ_WINDOW_SIZE | 500 |
| PHY rate | 1 Mbps |

NS-2로 구현하여 성능을 분석하였다. NS-2로 구현하기 위해서 그림 3과 같은 모바일 노드에서 기존 WLAN MAC을 제거하고 IEEE 802.16d MAC을 새롭게 구현하였다. MAC을 제외한 나머지 부분은 기존 모바일 노드의 각 모듈을 그대로 사용하였다. ARQ 기법 성능 분석을 위해서 사용한 네트워크 모델은 PMP모드 WMAN 구조로 본 연구에서는 데이터를 전송하는 한 개의 BS와 수신하는 한 개의 SS가 있는 간단한 모델을 사용하였다. 이는 시뮬레이션을 간단히 하면서 본 연구의 목적이 ARQ기법에 대한 성능 분석이므로 다른 요인은 배제하기 위함이다. 응용프로토콜은 UDP위에 CBR 트래픽 모듈을 사용하였으며 순방향 링크 전송동안 SS는 평균 5개의 ARQ 블록을 수신 할 수 있도록 CBR 패킷 발생률을 조정하였다. 시뮬레이션을 시작하면 IEEE 802.16d에 명시된 초기 레인징 절차를 수행한 후 BS에서 SS로 데이터를 전송하는 순방향 링크의 ARQ성능을 측정할 수 있도록 구현하였다. ARQ의 Fragmentation 기능은 비활성화 시켰으며 수신 측에서 ARQ feedback IE는 Standalone에 따라 전송한다. 그리고 SS로부터 Feedback IE 정보를 받기 위해서 역방향 링크의 전송 기간 동안 SS가 Feedback IE 정보를 전송할 수 있도록 Grant를 부여하며 (예를 들면 2개 프레임 당 1번 Grant를 부여하는 것을 Grant 2라 칭함) 시뮬레이션에 사용한 자세한 파라미터는 표 1에 나타내었다.

IV. 성능 평가

본 연구에서는 ARQ 기법의 성능 평가를 위해 무선채널 환경에서 나타날 수 있는 다양한 환경을 가정하였다. 먼저 LOS(Line of Sight) 환경처럼 무선채널 상태가 양호한 경우를 가정하여 오류가 없

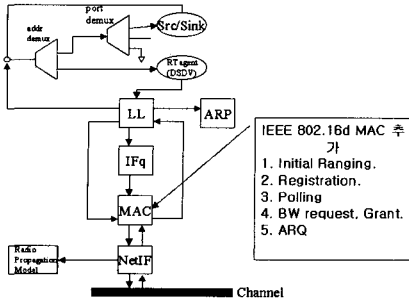


그림 3. NS-2 Mobile node 구조

는 환경을 가정하였으며, 전송 오류가 발생하는 무선채널 환경에서 ARQ 기법의 성능을 평가하기 위해서 랜덤 전송오류가 발생하는 환경 그리고 고층 빌딩이나 전파의 음영지역에 단말이 있을 경우 나타날 수 있는 채널환경을 가정하여 버스트 오류가 발생하는 환경으로 분석하였다. 성능 평가의 척도로는 각 ARQ기법의 성능과 효율성을 비교하기 위해 다음과 같은 두 가지를 정의하여 사용하였다.

- 시뮬레이션 시간동안 수신 측에서 오류 없이 수신한 총 ARQ 블록 수(수치계산 결과 그래프에서는 “블록 수”라 칭함)

- $ACK_Efficiency =$

$$\frac{\text{오류없이 순서대로 수신한 총 ARQ 블록 크기 [bytes]}}{\text{시뮬레이션 동안 사용한 총 ARQ_Feedback_IE [bits]}}$$

4.1 오류가 없는 채널 환경

먼저 오류가 발생하지 않는 채널 환경에서 총 시뮬레이션 시간동안 수신 측에서 오류 없이 순서대로 수신한 ARQ 블록 수를 비교하였다(표 2 참조). 표에 나타난 바와 같이 4가지 기법이 동일한 블록 수를 가지는 것을 알 수 있다. 그러나 $ACK_Efficiency$ 를 이용하여 비교하면 Cumulative ACK이 가장 우수하고 나머지 세 가지 방식은 성능이 동일함을 알 수 있다(표 2 참조). 이 결과는 Cumulative ACK 방식이 나머지 방식에 비해 데이터 전송 과정에서 사용한 ARQ feedback IE의 양이 상대적으로 작기 때문이다(2개 프레임마다 1번 Grant를 부여하는 경우임).

표 2. 오류가 없는 환경에서 성능결과 (Grant 2)

| 성능 평가 척도 | Sel. ACK | Cum. ACK | Cum. with Sel. ACK | Cum. with Block ACK |
|----------------|----------|----------|--------------------|---------------------|
| 블록 수 | 45391 | 45391 | 45391 | 45391 |
| ACK_Efficiency | 20.834 | 31.251 | 20.834 | 20.834 |

위와 같은 결과는 수신측이 ACK 정보로 보내는 ARQ Feedback IE의 필드의 크기를 살펴보면 알 수 있다. Cumulative ACK은 공통적인 정보(CID, LAST, ACK type 등)를 제외하고는 11bit의 BSN 필드만을 이용하지만 나머지 방식은 BSN 필드와 더불어 16비트의 추가적인 정보를 사용하기 때문이다.

4.2 랜덤 채널 오류 환경

랜덤 채널 환경에서의 4가지 ARQ기법의 성능을 비교하기 위해 랜덤 오류는 순방향 링크에만 발생하며 역방향에 대한 채널 오류는 발생하지 않는다고 가정하였고 랜덤 오류확률을 0~2.5%로 증가시키면서 4가지 기법의 성능을 분석하였다. 먼저 각 ARQ 기법 별 수신 측에서 오류 없이 순서대로 수신한 ARQ 블록 수를 Grant 2인 환경에서 (2개 프레임 당 1번의 Grant 부여) 그림 4에 나타내었다.

그림 4에서 볼 수 있듯이 Cumulative ACK을 사용하였을 경우를 제외하고는 첫 번째 성능평가 척도가 동일하게 나타났다. 이는 Cumulative ACK을 사용하는 경우, 수신 측은 오류 없이 순서대로 수신한 블록의 BSN 값만을 ARQ feedback IE에 전송하며 송신측은 수신측이 보낸 ARQ feedback IE의 BSN이후의 블록을 수신 측의 수신여부에 관계 없이 보내게 된다. 따라서 동일한 블록의 수신이 반복해서 일어날 수 있기 때문에 성능이 저하되지만, 나머지 ARQ 방식들은 자신이 수신한 모든 블록의 정보를 ARQ feedback IE의 필드를 이용하여 송신 측에 알려주기 때문에 동일한 블록의 송신이 일어나지 않는다. 그리고 Cumulative ACK를 제외한 기법의 블록수가 동일한 값을 가지는 이유는 오류 모델이 동일하고 전송한 ARQ 블록의 수신여부를 ARQ feedback IE를 통해 모두 전달 받을 수 있으며 총 시뮬레이션 동안 전송할 수 있는 ARQ 블록

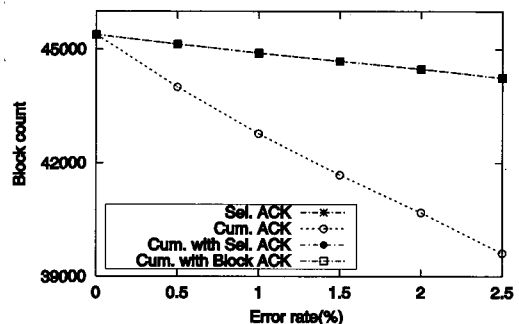


그림 4. 랜덤 오류 환경에서 블록 수 비교

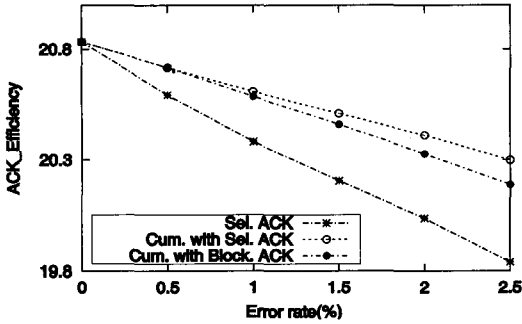


그림 5. 랜덤 오류 환경에서 ACK_Efficiency 비교(Grant 2)

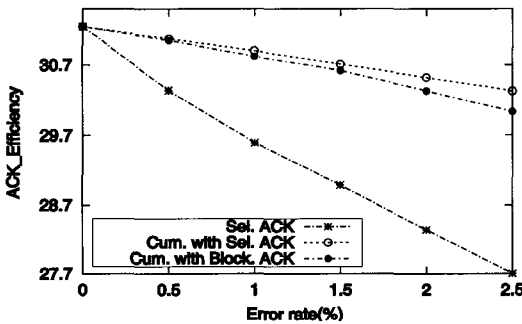


그림 6. 랜덤 오류 환경에서 ACK_Efficiency 비교(Grant 3)

개수는 동일하기 때문이다. ACK_Efficiency 측면에서 Cumulative ACK 을 제외한 나머지 세 가지 ARQ 방식간의 효율성을 비교한 결과를 그림 5에 나타내었으며, Cumulative with Selective ACK 방식이 가장 효율성이 우수함을 알 수 있다. 왜냐하면 ACK_Efficiency 값이 가장 작기 때문에 자신에게 할당된 대역폭을 가장 효과적으로 사용하였다.

그림 6과 표 3은 동일한 환경에서 수신 측에서 전송하는 ARQ feedback IE 주기를 세 프레임 당 1번으로 수정했을 경우의 성능분석 결과이다(Grant 3). 본 결과에서 특이한 사항은 Selective ACK 방식의 성능이 급격하게 감소하는 것이다. 이는 다른 방식에 비해 ACK Map개수를 사용하는 양이 많기 때문이다. 즉 Selective ACK 방식은 Grant 주기를 3으로 늘리게 되며 오류가 발생하지 않은 경우는 평균 15개의 블록에 대한 수신 정보를 한 개의 ACK Map을 사용하여 전송할 수 있지만 블록 오류가 발생한 경우는 한 개의 ACK Map으로 충분하지 않게 된다. 즉 2개를 사용해야만 전송할 수 있다. 하지만 나머지 Cumulative with Selective ACK과 Cumulative with Block sequence ACK은 Cumulative의 기능을 가지고 있기 때문에 오류확률

표 3. 랜덤 오류 환경에서 성능결과 (Grant 3)

| 버스트 오류(%) | Sel. ACK | Cum. ACK | Cum. with Sel ACK | Cum. with Block ACK |
|-----------|----------|----------|-------------------|---------------------|
| 0 | 45391 | 45391 | 45391 | 45391 |
| 0.5 | 45115 | 44909 | 45115 | 45115 |
| 1 | 44804 | 44369 | 44804 | 44804 |
| 1.5 | 44499 | 43753 | 44499 | 44499 |
| 2 | 44219 | 43278 | 44219 | 44219 |
| 2.5 | 43941 | 42830 | 43941 | 43941 |

이 낮은 경우는 1개의 ACK Map으로 충분히 전송할 수 있게 된다.

4.3 버스트 채널 오류 환경

무선 환경의 채널의 특징은 버스트 채널 오류가 발생하는 것이다. IEEE 802.16d에서 규정하는 4가지 기법 중 Cumulative with Block Sequence ACK 방식의 ARQ feedback IE의 필드를 살펴보면 버스트 에러에 대비하여 제정된 기법임을 쉽게 알 수 있다. 왜냐하면 한 개의 ACK Map만 사용할 경우 다른 방식들은 16비트의 ACK Map 필드를 사용하여 최대 16개의 부가적인 정보를 송신 측에 알려 줄 수 있지만 Cumulative with Block Sequence ACK은 16개 비트를 그림 1과 같이 2개 또는 3개의 블록으로 나누어 사용하며 3개를 사용할 경우 한 블록 당 15개의 정보를 담을 수 있으며 2개의 블록을 사용할 경우 63개의 블록정보를 담을 수 있기 때문이다. 버스트 채널 오류 환경은 랜덤 오류모델과 같이 순방향 링크에만 발생한다고 가정하였고 오류 확률을 0~2.5% 사이에서 0.5%단위 마다 성능을 평가 했다. 버스트 오류 모델은 랜덤 오류 방식에 따라 오류가 발생하며, 랜덤 오류가 발생하면 오류 블록의 크기가 0과 10사이에서 균일 분포로 발생하도록 하여 평균 5개의 블록이 연속적으로 오류가 나게 하였다. 표 4에서는 Grant 2 환경에서 4가지 ARQ 방식의 성능을 성능평가 척도 1관점에서 나타내었고, 그림 7에서는 ACK_Efficiency 측면에서 성능비교 결과를 나타내었다. 그림 7에서 Cumulative ACK을 비교대상에서 제외한 이유는 오류 없이 수신한 블록의 BSN값이 나머지 방식에 비해 상대적으로 작기 때문에 ACK_Efficiency를 가지고 비교할 수 없기 때문이다.

그림 7의 결과를 살펴보면 예상했던 결과와는 달리 랜덤 오류에서의 성능이 우수했던 Cumulative with Selective ACK 기법의 성능이 Cumulative with Block sequence ACK보다 여전히 좋게 나타났다.

표 4. 버스트 오류 환경에서 성능결과 (Grant 2)

| 버스트 오류(%) | Sel. ACK | Cum. ACK | Cum. with Sel ACK | Cum. with Block ACK |
|-----------|----------|----------|-------------------|---------------------|
| 0 | 45391 | 45391 | 45391 | 45391 |
| 0.5 | 45136 | 43537 | 45136 | 45136 |
| 1 | 44902 | 40212 | 44902 | 44902 |
| 1.5 | 44690 | 40212 | 44690 | 44690 |
| 2 | 44450 | 38629 | 44450 | 44450 |
| 2.5 | 44203 | 36978 | 44203 | 44203 |

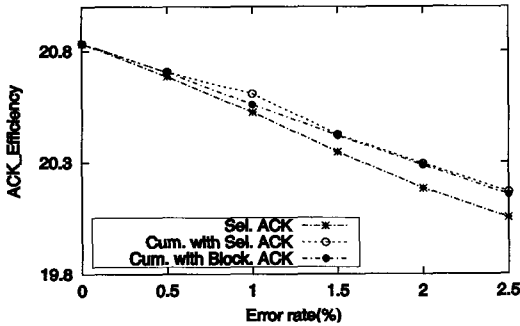


그림 7. 버스트 오류 환경에서 ACK_Efficiency 비교 (Grant 2)

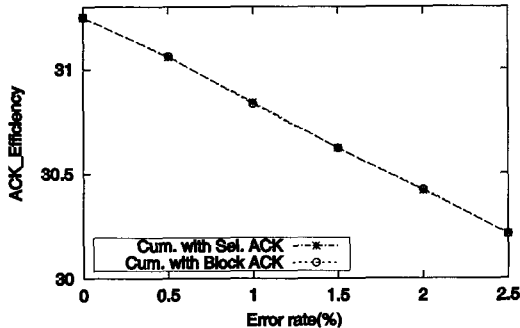


그림 8. 버스트 오류 환경에서 ACK_Efficiency 비교 (Grant 3)

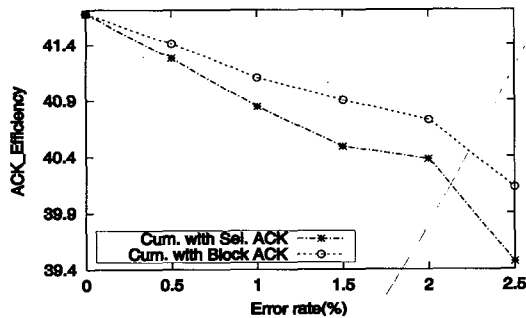


그림 9. 버스트 오류 환경에서 ACK_Efficiency 비교 (Grant 4)

이는 ACK 정보를 받기 위해 부여한 Grant 2 때문에 ARQ feedback IE가 상대적으로 빈번하게 전송되었기 때문이다. 즉 Cumulative with Block Sequence

ACK의 필드 값을 효과적으로 사용하지 못했기 때문에 판단되어 grant 주기를 3, 4 프레임 당 1번으로 늘려가면서 동일한 버스트 오류 채널 환경의 효율성 평가 결과를 그림 8과 9에 나타내었다.

Grant의 주기가 2에서 4로 증가할수록 Cumulative with Block Sequence ACK의 효율성의 저하 정도는 미비하지만 Cumulative with Selective ACK의 효율성은 큰 폭으로 저하되는 것을 볼 수 있다. 이는 Cumulative with Selective ACK은 버스트에러 정보를 주기 위해 사용하는 ACK Map 한 개로 모자라서 2개 혹은 3개를 사용하기 때문이다. 그러나 버스트 오류가 발생하는 환경처럼 오류가 군집하여 발생하는 경우 Cumulative with Block Sequence ACK을 사용하게 되면 블록의 수신정보를 앞에서 설명한 것처럼 2 블록단위 또는 3 블록단위로 나누어서 전송하게 되면 상대적으로 많은 양의 정보를 나타낼 수 있다. 즉 버스트 오류가 발생하는 채널 환경에서는 Cumulative with Block Sequence ACK이 다른 방식에 비해 효율성이 좋다.

V. 결론

IEEE 802.16d의 ARQ 기법은 송수신 MAC 계층 간 신뢰성 있는 데이터 송수신을 위해 채택된 기법으로 무선채널 환경과 다양한 제약사항에 따라 알맞은 기법을 채택하여 사용할 수 있도록 4가지 기법을 정의하고 있다. 본 연구는 이러한 4가지 ARQ기법의 성능을 다양한 무선채널 환경에 따라 앞에서 정의한 두 가지 성능평가 척도를 이용하여 특정 채널환경에서 우수한 ARQ기법을 찾고자 하였다. 성능분석 결과로부터 알 수 있듯이 4가지 ARQ 기법은 각각의 장점을 가지고 있음을 알 수 있었다. 먼저 채널 오류가 없는 환경에서는 송신측이 전송한 데이터를 수신 측에서 오류 없이 모두 수신할 수 있기 때문에 4가지 ARQ기법에 대한 첫 번째 성능평가 파라미터(수신 측에서 오류 없이 순서대로 수신한 블록 수)는 모두 동일하였지만 두 번째 파라미터(ACK_Efficiency)를 고려하면 Cumulative ACK 방식의 ARQ feedback IE 크기가 다른 기법에 비해 상대적으로 작기 때문에 가장 효과적임을 알 수 있었다. 그러나 전송 오류가 발생하는 랜덤 채널 오류 환경이나 버스트 채널 오류 환경에서는 각 ARQ 기법의 성능은 채널환경 그리고 ARQ feedback IE 전송주기에 따라서도 크게 변화하는 것을 알 수 있었다. 따라서 채널 오류가 있는 환경에서 효과적인

ARQ 기법을 선택하기 위해서는 현재 채널상태 이외에 ARQ feedback IE 전송 주기를 같이 고려해야 한다는 것을 알 수 있었다. 무선채널 환경은 유선채널 환경에 비해 전송 오류가 발생할 확률이 상대적으로 높아 본 연구에서 고려한 채널환경 중 랜덤 오류 환경이나 버스트 오류환경을 고려해야 한다. 따라서 ARQ 기법을 효과적으로 사용하기 위해서는 현재의 채널 상태만 뿐만 아니라 ARQ feedback IE의 전송 주기도 함께 고려하여 운영해야 최적의 ARQ기법을 운영할 수 있다. IEEE 802.16d의 ARQ 방식은 초기 연결 설정 시 4가지 방법 중 한 가지 방식이 선택 되지만, ARQ Feedback IE payload의 전송주기는 특별히 정해져 있지 않기 때문에 자신이 사용하는 ARQ 방식에 따라 전송효율을 높이기 위해서는 채널환경과 더불어 ARQ feedback IE주기를 고려하여 ARQ기법을 채택하는 것이 IEEE 802.16d의 전송 효율을 더욱 높일 수 있음을 본 연구결과를 통해서 확인할 수 있었다.

참 고 문 헌

[1] IEEE std 802.16-2004, part 16 : *Air Interface for Fixed Broadband Wireless Access System*, Oct 2004.

[2] Guo Song Shu, Deng Wang and Shunliang Mei, "A QoS architecture for the MAC protocol of IEEE 802.16 BWA System," *Communication Circuits and System and West Sino Exposition, IEEE 2002*.

[3] Howon Lee, Tae Soo Kwon and Dong-Ho Cho, "An Efficient Uplink Scheduling Algorithm for VoIP Service in IEEE 802.16 BWA System"

[4] 주판유, 구창희, 이현우, 김영균, "이동 광대역 무선 접속 시스템의 기술정리 및 표준화 동향," *Telecommunication Review*, 특별부록 2003.

[5] S. McCanne and S. Floyd, "NS network simulator," version 2.27, <http://www.isi.edu/nsnam/ns>.

장 재 신 (Jaeshin Jang)

정회원



1990년 2월 동아대학교 전자공학
학과 졸업
1992년 2월 한국과학기술원 전
기및전자공학과 석사
1998년 2월 한국과학기술원 전
기및전자공학과 박사
1998년 7월~2002년 2월 (주)삼

성전자 정보통신총괄 책임연구원

2002년 3월~현재 인제대학교 공과대학 정보통신공학과
교조교수

<관심분야> WMAN, 무선 메쉬네트워크, 센서네트워크, 무선인터넷, QoS 보장

강 민 석 (Min-seok Kang)

준회원



2005년 2월 인제대학교 정보통신
공학과 졸업
2005년 3월~현재 인제대학교 대
학원 전자정보통신공학과 석
사과정

<관심분야> WMAN(IEEE 802.16),
WiBro, QoS, Scheduling