WiBro 시스템의 random access 기법의 충돌해결 방법
Conflict Resolution of the Random Access Algorithm in the WiBro System

국광호*, 임석구**, 이강원*, 권병천*, 김경수***
서울산업대학교 산업정보시스템공학과*
천안대학교 정보통신학부**,
한국전자통신연구원 이동통신연구단***

초 록

이동중인 단말에게 인터넷 서비스를 제공하고자 하는 WiBro 시스템은, 무선자원의 효율적인 사용을 위해서 전송할 데이터가 있는 단말들에게만 대역폭을 할당하는 demand-based 대역폭 할당 방식을 사용한다.

이때 여러 단말들이 동시에 데이터를 전송하고자 할 경우 충돌이 발생하므로 이들의 전송순서를 정해주는 것이 필요하다.

기존의 WiBro 시스템에서는 binary exponential backoff algorithm 에 기초한 random access 기법을 사용하는 데, 본 연구에서는 conflict resolution 기법에 기초한 새로운 random access 기법을 제안하고, 새로운 기법의 성능이 보다 우수함을 시뮬레이션을 통해 보였다.

1. 서 론

이동중인 단말들에게도 인터넷 서비스를 제공할 수 있는 WiBro 시스템이 내년도에 상용서비스 제공을 목표로 현재 개발 중인데, WiBro 시스템은 무선자원의 효율적인 사용을 위해서 전송할 데이터가 있는 단말들에게만 대역 폭을 할당하는 demand-based 대역폭 할당 방식을 사용한다.

즉 WiBro 시스템은 5ms 주기로 uplink 와 downlink 프레임을 TDD(Time Division Duplex) 방식으로 전송하는 데 기지국으로 전송할 데이터를 가진 단말들은 데이터 전송을 위한 uplink 슬롯을 할당 받기 위해서 uplink 상의 대역폭 요청 슬롯을 경쟁방식으로 액세스한다. 이때 여러 단말들이 동시에 대역폭 요청 슬롯을 액세스하는 경우 충돌이 발생하므로 이들의 전송을 제어해 주는 것이 필요하다.

여러 단말들이 공유매체를 통해서 메시지를 전송하고자 할 때 단말들의 메시지 전송을 제어하기 위해 conflict-resolution 알고리즘이 사용되어 왔는데([1]) conflict-resolution 알고리즘은 충돌을 해결하기 위해서 group-testing 기법을 사용한다([2]). 즉 conflict-resolution 알고리즘은 어떤 group 에 속한 단말들이 동일한 slot 에서 메시지를 전송하였을 때, 채널의 feedback 결과(idle, success, 또는 collision)를 이용한다. 만약에 feedback 이 idle 이거나 success 라면 그 group 멤버들의 메시지 전송은 다 해결되었음을 알 수 있다. 그러나 feedback 이 collision 이라면 group 의 적어도 두 멤버가 충돌했다는 것을 의미한다. 그러면 잠시 다른 단말들의 메시지 전송을 미루고 충돌된 메시지들이 해결되도록 한다. 예를 들어 충돌된 모든 단말들에게 동전을 던지도록 하여 "앞면"을 던진 단말들은 다음 슬롯에서 메시지를 재전송하게 하고 뒷면을 던진 단말들은 기다리게 한다. 다음 슬롯의 feedback 결과가 idle 이거나 success 가 되면 기다리고 있는 단말들에게 위 과정을 반복하게 하여 충돌을 해결토록 한다.

기존의 WiBro 시스템에서는 binary exponential backoff algorithm 에 기초한 random access 기법을 사용하는 데, 본 연구에서는 conflict resolution 기법에 기초한 새로운 random access 기법을 제안하고, 새로운 기법의 성능이 binary exponential backoff algorithm 보다 우수함을 시뮬레이션을 통해 보였다.

2. WiBro 시스템의 random access 알고리즘

본 절에서는 현재 Wibro 시스템에서 random access 방법으로 고려중인 binary exponential backoff algorithm 과, conflict resolution 알고리즘을 WiBro 시스템에 적용할 수 있도록 변형하는 방법에 대해 살펴보고자 한다.

2.1 Binary exponential backoff algorithm

Binary exponential backoff 알고리즘에서의 충돌해결은 기지국에 의해 제어되는 initial backoff window 와 maximum backoff window 에 기초하여 다음과 같이 행해진다.

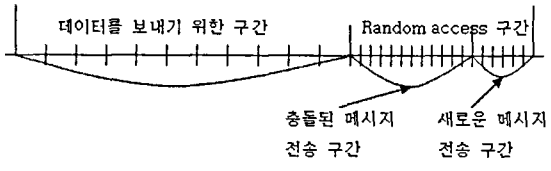
데이터를 전송하고자 대역폭을 요청하는 단말은 initial backoff window 와 maximum backoff window 의 값을 기지국으로부터 얻는다. 단말은 임의로 initial backoff window 내의 하나의 값을 선택한다. 이 숫자는 단말이 대역폭 요청 메시지를 전송하기 전에 그냥 보내야 할 대역폭 요청 슬롯의 수(random access 구간내의 슬롯)를 나타낸다. 단말은 대역폭 요청 메시지를 전송한 후 일정시간 내에 대역폭이 할당되면 대역폭 요청 절차를

끝내게 되고, 그렇지 않으면 대역폭 요청 절차가 실패한 것으로 간주하고 단말은 backoff window 를 maximum backoff window 를 초과하지 않는 한 두배로 증가시킨다. 위 과정을 최대 반복 횟수 만큼 시도하고 그래도 성공하지 못하면 해당 데이터는 폐기된다.

2.2 Conflict-resolution 알고리즘에 기반한 random access 알고리즘

앞 절에서 설명한 conflict-resolution 알고리즘을 random access 에 적용하기 위해서는 각 슬롯에서의 충돌 발생 여부를 기지국이 단말에게 알려줄 수 있는 feedback 절차가 필요하다. WiBro 시스템은 5ms 주기로 uplink 와 downlink 프레임을 전송하므로 프레임 단위로만 feedback 하는 것이 가능하다.

우선 uplink 채널이 다음 (그림 1)에서 보는 바와 같이 데이터를 보내기 위한 구간과 대역폭 요청을 위한 random access 구간으로 구성된다고 가정한다. 이때 random access 구간은 새로운 대역폭 요청 메시지가 전송되는 구간과, 대역폭 요청 메시지가 충돌되었을 경우 충돌된 대역폭 요청 메시지들만을 재전송하는 구간으로 구성된다고 가정한다.



(그림 1) Uplink 채널 구조

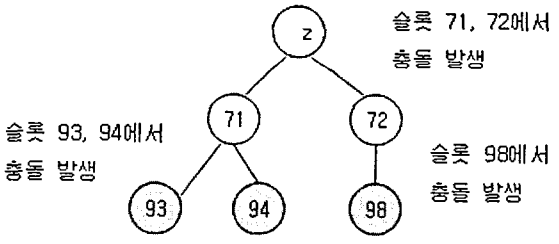
만약 WiBro 시스템의 기지국이 random access 구간내의 어떤 슬롯에서 대역폭 요청 메시지가 충돌되었을 경우, 이를 충돌이 발생한 단말들에게 알려줄 수 있다면 다음과 같이 변형된 conflict-resolution 알고리즘을 사용할 수 있다.

먼저 random access 구간내의 슬롯들이 일련번호를 갖는다고 가정한다. 새로운 대역폭 요청 메시지를 전송하는 슬롯 z 에서 충돌이 발생한 경우 기지국은 다음 (그림 2)와 같이 root node 에 z 를 갖는 새로운 tree 를 생성한다.

기지국은 z 에서 메시지를 전송한 단말들에게 대역폭 요청을 다시 할 수 있도록 충돌된 대역폭 요청 메시지들만을 재전송하는 구간내의 인접한 두 개의 슬롯을 할당한다. 그러면 z 에서 메시지를 전송한 단말들은 임의로 두 개의 슬롯 중 하나를 선택한 후 대역폭 요청 메시지를 재전송한다.

그 결과 하나의 대역폭 요청 메시지만 전송되어 대역폭 요청이 성공되는 슬롯에는 데이터 전송을 위한 대역폭이 할당되고 트리에 노드가 더 이상 더해지지 않는다.

그러나 또 다시 충돌이 발생하는 경우에는 (그림 2)에서 보는 바와 같이 새로운 노드를 tree 에 더한 후 충돌이 발생한 슬롯 번호를 그 노드에 기입한다. 두 개의 슬롯 모두에서 충돌이 발생한 경우에는 두 개의 노드가 tree 에 더해지고 슬롯번호가 각각 노드에 기록된다. 위 과정을 preorder tree-traversal 알고리즘을 사용하여 tree 의 모든 노드가 조사될 때까지 반복한다.



(그림 2) Conflict resolution 알고리즘의 conflict resolution tree

(그림 1)에서 충돌된 메시지를 전송하는 구간의 슬롯 수가 n 이라 하면, n/2 개의 슬롯에서 충돌이 발생한 단말들에게 인접한 2 개의 슬롯을 할당해 줄 수 있다. 현재 tree 가 하나뿐이라면 트리내의 아직까지 방문되지 않은 노드들에게 각각 2 개의 인접한 슬롯을 할당해 줄 수 있다. 즉, (그림 2)에서 현재까지 방문한 노드들 밑에 있는 노드들(회색으로 표현된 노드들) 각각에게 인접한 슬롯들을 할당해 줄 수 있다.

한편 새로운 대역폭 요청 메시지를 전송하는 구간내의 슬롯에서 충돌이 발생할 때마다 새로운 트리가 만들어 지므로 임의의 순간에 여러 개의 tree 가 존재할 수 있다. 이때에는 round-robin 방식 등의 방법에 의해 각 tree 에게 2 개의 인접한 슬롯을 순차적으로 할당해 줄 수 있다.

위의 과정에 의해 conflict-resolution 이 행해지려면 기지국은 특정 슬롯에서 충돌이 발생한 단말들에게, 충돌된 대역폭 요청 메시지들만을 재전송하는 구간내의 어떤 인접한 두 개의 슬롯에서 데이터를 전송해야 하는지를 알려주어야 한다. 이는 기지국이 단말로 보내는 UL-MAP 메시지가 다음 <표 1>에서 보는 바와 같이 random access 구간내의 슬롯 번호와 이전에 충돌이 발생한 슬롯 번호를 포함하고 있으면 가능하다. 즉 <표 1>은 이전에 슬롯 46 에서 충돌이 발생한 단말들은 현재 프레임의 슬롯 75 와 76 을 이용하여 전송하도록 하는 것을 나타낸다.

<표 1> random access 구간을 위한 UL-MAP 메시지 형식

71	21
72	21
73	41
74	41
75	46
76	46
77	
	새로운 대역폭 요청 메시지
	:

이때 random access 구간 내의 슬롯 번호의 주기가 충분히 커서 충돌이 발생한 슬롯 번호와 overlap 이 발생하지 않아야 한다. 이를 위해서는 random access 구간 내의 슬롯 번호를 16 비트

정도로 표현해 주는 것이 필요하다. 하지만 random access 구간내의 매 슬롯에 대해 4 바이트의 데이터를 전송해야 하므로 UL-MAP 메시지의 양이 너무 많아지게 되어 실용성이 없어진다. 따라서 UL-MAP 의 메시지 양을 크게 하지 않으면서 위의 방법을 근사적으로 실현할 수 있는 다음과 같은 방법을 고려할 수 있다.

위 conflict-resolution 알고리즘의 기본 개념은 충돌 발생시, 다음 번에 메시지를 재전송할 때에는 충돌 발생 가능성을 줄이기 위해서 특정 2 개의 슬롯에는 특정한 하나의 슬롯에서 충돌이 발생한 단말들만 메시지를 전송하도록 하는 방식을 택한다.

UL-MAP 메시지 양을 줄이기 위해 위 개념을 적용하여, 각 프레임에서 향후 해결해야 할 충돌된 슬롯의 수가 n 으로 주어졌을 때 각 충돌된 슬롯들에게 2 개의 슬롯들을 할당해 주는 대신 전체 n 개의 충돌된 슬롯들에게 2n 개의 슬롯을 할당해 주고 이들 중 임의로 하나를 선택하여 전송하도록 할 수 있을 것이다.

예를 들어, 향후 해결해야 할 충돌된 슬롯의 수가 3 개라 하자. 그러면 첫 번째 충돌된 슬롯에서 전송한 단말들은 슬롯 1, 2 에서, 두 번째 충돌된 슬롯에서 전송한 단말들은 슬롯 3, 4 에서, 세 번째 충돌된 슬롯에서 전송한 단말들은 슬롯 5, 6 에서 전송하도록 하는 대신 충돌이 발생한 모든 단말들이 슬롯 1~6 에서 임의로 전송하도록 할 수 있을 것이다.

향후 해결해야 할 충돌한 슬롯의 수는 기지국이 하나의 프레임에서 충돌이 발생한 슬롯의 수를 알 수 있다면 다음과 같이 구해질 수 있다. n_i 를 i 번째 프레임 시작 시 앞으로 해결해야 할 충돌된 슬롯의 수라 하고 a_i 를 i 번째 프레임에서 해결이 시도된 충돌된 슬롯의 수, c_i 를 i 번째 프레임에서 새로이 발생한 충돌된 슬롯의 수라 하면 i+1 번째 프레임부터 향후 해결해야 할 충돌된 슬롯의 수 n_{i+1} 은 다음과 같이 구해질 수 있다.

$$n_{i+1} = n_i - a_i + c_i$$

한편 하나의 프레임이 r 개의 random access 슬롯으로 이루어지고 이중 최소한 r_n 개는 새로운 호의 대역폭 요청 메시지를 전송하기 위해서 사용된다고 하면 a_i 는 $0 \sim (r-r_n)/2$ 개의 값을 가질 수 있다. 따라서 i 번째 프레임에서 해결해야 할 충돌의 수 n_i 가 $(r-r_n)/2$ 를 초과한다면 $(r-r_n)/2/n_i$ 만큼의 충돌만 i 번째 프레임에서 해결하도록 한다. 이는 전체 충돌이 발생한 단말 중 $(r-r_n)/2/n_i$ 만큼의 단말만 대역폭 요청 메시지를 재전송하도록 하는 것과 동일하다고 볼 수 있으므로, 충돌이 발생한 모든 단말들이 각각 [0, 1] 사이의 난수를 뽑아 $(r-r_n)/2/n_i$ 보다 적으면 재전송하도록 한다.

그리고 새로운 호의 대역폭 요청 메시지들은 위와 같이 충돌된 메시지들을 재전송하는 구간이 결정되면 random access 구간내의 나머지 슬롯에서 전송하도록 한다.

3. Random access 알고리즘의 성능분석 결과

본 절에서는 2 절에서 설명한 random access 알고리즘들의 성능을 시뮬레이션에 의해 분석한다. 이때 대역폭 요청 메시지들은 포아송 분포에 따라 발생한다고 가정한다.

3.1 Binary exponential backoff 알고리즘의 성능 분석 결과

Binary exponential backoff 알고리즘의 성능은 initial backoff window, maximum backoff window, 최대 반복회수 등의 값에 따라 약간 다를 것이나 본 연구에서는 maximum backoff window 의 값은 1024 로, 최대반복횟수는 16 으로 가정한다. 즉 충돌이 발생하면 최대 16 번까지 재전송을 시도한다고 가정한다.

<표 2>는 한 프레임에서 random access 슬롯의 수가 50 개이고 initial backoff window 가 32 일 때의 성능을 나타내며, <표 3>은 한 프레임에서 random access 슬롯의 수가 20 개이고 initial backoff window 가 8 일 때의 성능을 나타낸다.

<표 2> Binary exponential backoff 알고리즘의 성능
(Random access 슬롯의 수가 50 개 일 때)

부하	평균 접속시간	총충돌수	총요청수	충돌비율
5%	52.41	1108	9970	0.11
10%	55.19	4696	19977	0.24
15%	59.20	11903	29981	0.40
20%	65.77	25612	40031	0.64
25%	76.17	48698	50029	0.97
30%	102.06	92856	59903	1.55

<표 3> Binary exponential backoff 알고리즘의 성능
(Random access 슬롯의 수가 20 개 일 때)

부하	평균 접속시간	총충돌수	총요청수	충돌비율
5%	20.87	1134	9967	0.11
10%	22.09	5139	20002	0.26
15%	23.62	12883	30057	0.43
20%	26.36	28694	39958	0.72
25%	30.95	56261	50086	1.12
30%	43.73	116504	60083	1.94

이들 표에서 부하는 하나의 프레임에서 하나의 random access 슬롯당 대역폭 요청이 몇 번 행해지는 지를 나타낸다. 예를 들어 random access 슬롯의 수가 50 개 일 때 부하가 10%라는 것은 평균적으로 하나의 프레임마다 5 개의 대역폭 요청 시도가 발생하는 경우를 나타낸다. 평균접속시간은 대역폭 요청을 시작한 순간부터 대역폭 요청이 완료될 때까지의 시간을 random access 슬롯의 수로 나타낸 결과이다. 그리고 총요청수는 시뮬레이션 기간 동안 대역폭 요청이 발생한 총 횟수를, 총충돌수는 충돌이 발생한 총 횟수를, 충돌비율은 대역폭 요청당 평균적으로 몇 번의 충돌이 발생하는 지를 나타내며 (총충돌수) /(총요청수)로 얻어진다.

3.2 Conflict-resolution 에 기반한 random access 알고리즘의 성능 분석 결과

2.2 절에서 설명한 conflict resolution 에 기반한 random access 알고리즘의 성능을 random access 구간이 각각 50 개와 20 개의 슬롯으로 이루어질 때 분석하면 다음 <표 4>와 <표 5>에서 보는 바와 같다. 이때 새로운 호의 대역폭 요청을 위한 최소한의 슬롯 개수인 r_n 은 random access 구간의 10%의 값을 갖는다고 가정한다.

<표 4> Conflict-resolution 알고리즘의 성능
(Random access 슬롯의 수가 50 개 일 때)

부하	평균 접속시간	총충돌수	총요청수	충돌비율
5%	52.46	694	9991	0.07
10%	55.70	3018	19947	0.15
15%	60.00	7511	30010	0.25
20%	65.08	14397	39944	0.36
25%	73.31	26051	49995	0.52
30%	82.96	42182	59975	0.70

<표 5> Conflict-resolution 알고리즘의 성능
(Random access 슬롯의 수가 20 개 일 때)

부하	평균 접속시간	총충돌수	총요청수	충돌비율
5%	20.96	643	10003	0.06
10%	22.31	3019	20010	0.15
15%	23.84	7301	30011	0.24
20%	25.86	14203	40036	0.35
25%	28.50	24661	50030	0.49
30%	33.68	44728	59987	0.75

<표 2>와 <표 4>로부터 새로운 conflict resolution 알고리즘의 성능은 부하가 25% 이하일 때에는 평균접속시간이 exponential backoff 기법과 별 차이가 없지만 부하가 25% 이상이 될 때에는 훨씬 좋아짐을 알 수 있다. 그리고 충돌비율은 모든 부하 하에서 거의 2 배 정도 줄어 듬을 알 수 있다. 즉 uplink 대역폭의 낭비를 상당히 줄일 수 있음을 알 수 있다. 따라서 새로이 제안하는 conflict-resolution 알고리즘의 성능이 exponential backoff 기법보다 월등히 뛰어남을 알 수 있다. <표 3>과 <표 5>로부터 random access 슬롯의 수가 20 개 일 때에도 유사한 결과를 낳음을 알 수 있다.

참고문헌

[1] M. Molle, "Conflict-Resolution Algorithms," Encyclopedia of Telecommunications, Vol. 4, Communications Human Factors to Cryptology, pp. 381-398.

[2] T. Berger, N. Mehravri, D. Towsley, and J. Wolf, "Random Multiple-Access Communication and Group Testing," IEEE Trans. Commun., COM-32(7), pp. 769-779, July 1984.

[3] J. I. Capetanakis, "Tree Algorithms for Packet Broadcast Channels," IEEE Trans. Inform. Theory, IT-25, pp 505-515, September 1979.

[4] R. G. Gallager, "Conflict Resolution in Random Access Broadcast Networks,", Proc. AFOSR Workshop in Commun. Theory and Application, pp. 74-76, September 1978.

[5] J. Mosley and P. A. Humblet, "A Class of Efficient Contention Resolution Algorithms for Multiple Access," IEEE Trans. Commun., COM-33, pp 145-151, February 1985