IEEE 802.16 SS에서의 Uplink Scheduler 구조 제안

이선돈[O†], 김우재[†], 서영주[†], 박윤상[‡]
[†] 포항공과대학교 컴퓨터공학과, [‡] 삼성전자 Telecommunication R&D Center
{sdonlee[O], hades15, yjsuh}@postech.ac.kr, yunsang.park@samsung.com

Uplink Scheduler Architecture of the SS in the IEEE 802.16

Seon-Don Lee[O†], Woo-Jae Kim[†], Young-Joo Suh[†], and Yun-Sang Park[‡]
[†] Pohang University of Science and Technology (POSTECH)
[‡] Samsung Electronics Co. Ltd., Telecommunication R&D Center

요 약

IEEE 802.16은 BWA (Broadband Wireless Access) 시스템의 표준으로 현재 많은 연구와 함께 상용 제품에 대한 연구가 진행중인 분야이다. IEEE 802.16에서는 QoS를 제공하기 위하여 BS (Base Station)와 SS (Subscriber Station)간의 QoS 협상 과정을 정의하고 있으며, BS 및 SS에서의 효율적인 QoS 보장을 위해 4가지의 서비스 클래스를 정의하고 있다. 이러한 서비스 클래스는 UGS, rtPS, nrtPS, 그리고 BE 이다. 하지만 표준에서는 이러한 서비스 클래스를 어떻게 이용할 것인지에 대한 언급이 없으며, 이에 따라 효율적인 packet scheduling에 관한 많은 연구가 진행되어 왔다. 기존의 연구에서는 주로 BS에서의 효율적인 scheduling에 초점을 맞추어 연구가 진행되었으며, SS에서의 scheduling에 대한 연구는 거의 되어 있지 않다. 하지만 BS에서 SS에게 대역폭을 할당할 때 GPSS (Grants per subscriber station) mode로 대역폭을 할당한다면 SS에서는 할당 받은 대역폭을 효율적으로 사용하기 위하여 scheduling이 필요하다. 본 논문에서는 SS에서의 scheduling architecture를 제안하고자 한다. 제안하는 scheduling architecture는 기존의 scheduling algorithm과는 달리 각 서비스 클래스에 대해서 효과적인 scheduling이 가능하도록 함으로써 시스템의 성능을 높이는데 기여할 것이다.

1. 서론

최근 광대역 무선 네트워크에 대한 연구가 활발히 진행되고 있으며, 특히 IEEE 802.16 [1] 시스템에 대한 표준이 많은 관심을 받고 있다. 광대역 무선 네트워크는 기존 이동통신망에서의 낮은 대역폭과 무선랜 시스템의 작은 전송영역을 보완하는 네트워크로서 이동통신 네트워크보다는 높은 대역폭을 제공하고 무선랜 시스템보다 넓은 전송영역을 가지고 있다. 현재 국내에서는 WiBro라는 시스템의 상용화에 대한 연구가 진행 중이며, 이 시스템은 IEEE 802.16 시스템에서 이동성 관리 기법을 추가한 것이다. 이렇듯 IEEE 802.16으로 대표되는 광대역 무선 네트워크가 많은 관심을 받는 이유 중의 하나는 바로 QoS 제공 능력이다.

광대역 무선 네트워크는 사용자에게 보다 많은 대역폭을 제공함으로써 향후 무선 네트워크의 핵심 응용프로그램인 멀티미디어 서비스에 적합하도록 설계되었다. 특히 응용프로그램마다 서비스 클래스를 다르게 할당함으로써 BS및 SS에서 차별화된 scheduling이 가능하도록 하였다. 이를 위하여 IEEE 802.16 표준에서는 UGS, rtPS, nrtPS, 그리고 BE의 네 가지 서비스 클래스를 정의하고, 각각에 대한 차별화된 정책을 적용하도록 하고 있다. 이 중 UGS는 일정한 시간마다 일정한 양의 대역폭이 필요한 응용 프로그램으로써 VoIP를 예로 들 수 있다. rtPS는 실시간 서비스를 위한 것으로 그 응용프로그램은 지연시간에 대한 제약조건과 함께 할당되는 대역폭에 대한 제약조건을 같이 가지고 있게 된다. nrtPS는 rtPS와 비슷하나 지연시간에 대한 제약 조건은 없고, 단지 할당되는 대역폭에 대한 제약조건만 존재한다. 마지막으로 BE는 일반적인 인터넷 응용 프로그램으로써 아무런 제약 조건이 없는 응용 프로그램을 의미한다.

IEEE 802.16 시스템은 이러한 네 가지의 서비스 클래스를 정의함으로써 BS 및 SS에서의 차별화된 scheduling이 가능하도록 하고 있다. 하지만 네 가지 서비스 클래스에 대한 정의 이외에는 구체적인 알고리즘이나 scheduling 구조에 대한 정의가 없기 때문에, 이에 대한 많은 연구가 진행되고 있다.

BS에서의 scheduling 알고리즘에 대해서는 기존의 이동통신망에서도 많은 연구가 되었으며, 이러한 연구 결과를 바탕으로 IEEE 802.16에서의 BS에 적합한 scheduling

본 논문은 삼성전자와 공동연구 결과이며 특허(10-2005-0036409)로 출원되었음.

알고리즘 및 구조에 대한 연구가 활발히 진행되고 있다 [2-4]. 하지만 IEEE 802.16에서 제시된 대역폭 할당 방법 중 하나인 GPSS (Grants per subscriber station) mode로 대역폭을 할당하는 경우 BS에서뿐만 아니라 SS에서도 scheduling 기법이 필요하다. BS가 GPC(Grants per connection) 방법으로 대역폭을 할당한다면 BS는 각 SS에 속한 connection에 대해서 각각 scheduling하여 그 결과를 SS에게 알려 주므로 SS는 할당 받은 대역폭에 대해서 해당 connection에 대한 데이터를 전송하면 되기 때문에, 특별한 scheduling이 필요 없다. 하지만 GPSS로 할당되는 경우, BS는 SS에서의 connection 종류나 개수에는 상관없이 SS가 필요로 하는 전체 대역폭을 할당하므로 SS는 할당 받은 대역폭을 효과적으로 사용하기 위하여 각 connection간에 전송 순서를 결정할 필요가 있다. 이렇게 SS에서의 scheduling 기법이 필요하지만 아직 관련 연구는 미미한 실정이다. 따라서 본 논문에서는 SS에서 효과적인 scheduling을 위한 구조를 제안하고 각각의 장단점에 대해서 분석하도록 한다.

2. SS에서의 Scheduling 구조

이장 에서는 앞장에서 언급하였듯이 SS에서의 scheduling 구조를 제안하고자 한다. SS에서의 scheduling은 기존 BS에서의 scheduling과 몇 가지 다른 사항이 존재한다. 첫 번째로 기존의 이동 통신망 등에서의 scheduling algorithm등은 무선 구간에서의 특수한 사항을 고려하여 제안되었다. 이는 BS에서 여러 SS에게 데이터를 전송하기 때문에 발생하는 것으로 point-to-multipoint 통신으로 발생하는 것이다. 하지만 SS에서의 scheduling은 BS와 통신하는 것이므로 point-to-point 통신이 된다. 따라서 기존 scheduling 알고리즘에서 고려하고 있는 개념들에 대해서는 본 논문에서는 고려하지 않도록 한다. 본 논문에서는 IEEE 802.16의 SS에서 동작 가능한 두 가지의 scheduler 구조를 제안하고자 한다. 그 중 하나는 one-level scheduler로서 하나의 scheduler가 모든 connection을 관리하는 기법이다. 다른 하나는 two-level scheduler로서 각 서비스 클래스마다 하나씩의 scheduler가 존재하고, 이들을 관리하는 또 다른 scheduler가 존재하는 형태이다. 다음 절에 각각에 대해서 자세히 살펴보도록 한다.

2.1 One-level scheduler

One-level scheduler는 그림 1에서 보여지는 것처럼 각 서비스 클래스에 해당하는 connection들이 오직 하나의 scheduler에 의해서 관리되는 형태이다. 먼저 그림에 대해서 간략히 설명하면 다음과 같다. SS의 application에서 생성되는 packet들은 connection ID를 가지고 있으며, 이에 따라 각각의 해당되는 queue에 삽입된다. 각각의 서비스 클래스들은 모두 같은 queue를 가지고 있으며, 데이터 전송을 위한 queue와 더불어 MAC에서의 control을 위한 control packet들을 위한 queue가 따로 존재한다. (Basic CH, Primary CH, Multicast CH) 이러한 control packet을 위한 queue는 데이터와 control packet간의 우선 순위에 따라서 전송 순서가 결정된다.

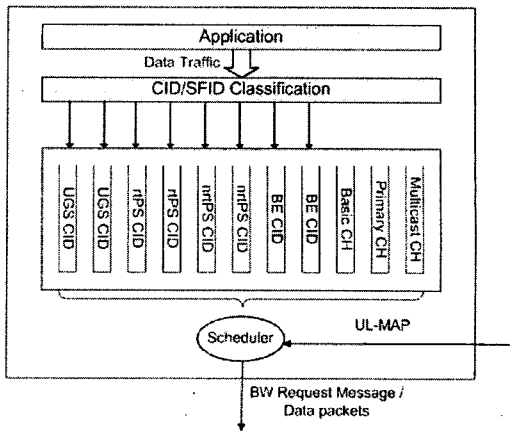

그림 1. One-level scheduler (Flow queue)

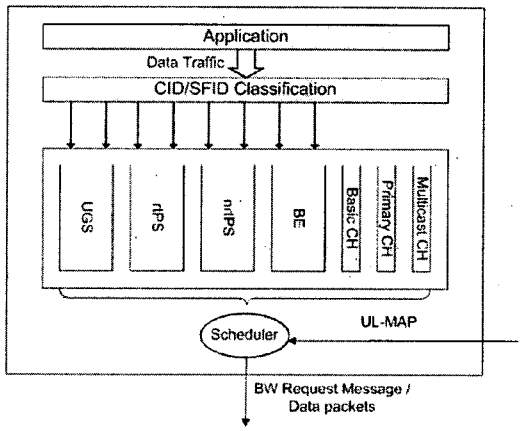

그림 2. One-level scheduler (Class queue)

먼저 one-level scheduler에서는 BS로부터 받은 frame의 UL-MAP 부분을 해석하여 SS가 언제부터 언제까지 데이터를 전송할 수 있는지를 판단한다. UL-MAP은 매 frame마다 BS에서 SS들에게 broadcast되는 부분으로써 각각의 SS가 BS에게 데이터를 전송할 때 그 시간을 명시해 주고 있다. 따라서 SS의 scheduler는 이 UL-MAP을 해석함으로써 자신이 전송할 시간을 알 수 있다. 또한 SS는 BS로 데이터를 전송할 때 자신의 signal strength를 측정함으로써 전송 속도에 대한 정보도 획득할 수 있으므로 이를 이용하면 자신이 전송할 수 있는 시간 동안 얼마만큼의 데이터를 전송할 수 있는지 알 수 있게 된다. 따라서 이 정보를 바탕으로 scheduler는 여러 queue에서 선택 기준에 따라서 packet을 선택한 후 그 순서에 따라 packet을 전송하게 된다.

여기서 한 가지 고려해볼 사항은 one-level scheduler에서는 하나의 scheduler가 존재하게 되므로 모든 queue가 하나의 선택 기준에 따라서 우선 순위가 정해진다는 것이다. 이는 각 서비스 클래스에 속하는 connection들이 요구하는 QoS가 각각 다른 상황에서 모든 요구 조건을 만족시키기는 어렵다. 즉 모든 queue에 대해서 그 queue가 UGS, rtPS, nrtPS, BE에 속하는지를 판단하고 이에 따라

scheduling을 진행하는 것은 너무 큰 overhead를 유발한다. 따라서 본 논문에서는 queue의 구조를 달리하여 보다 효율적인 scheduling이 가능하도록 두 가지 queue 구조를 제안한다.

그림 1에 보이는 것은 모든 connection에 대해서 하나씩 queue가 할당되는 것으로서 이를 flow queue라고 부른다. Flow queue에서는 앞서 언급하였듯이 모든 connection이 자신의 queue를 가지고 있으므로 scheduling시에 많은 overhead를 유발할 수 있다. 하지만 scheduling 알고리즘은 간단해지는 장점이 있다. 반면 그림 2에 보이는 것은 각 서비스 클래스마다 하나의 queue를 할당하고 모든 connection은 해당하는 서비스 클래스의 queue에 삽입되는 것이다. 이를 class queue 구조라고 부른다. 이 경우 scheduling 알고리즘은 flow queue를 사용하는 경우와 비교하여 queue 개수가 적고 각 queue는 다른 특징을 가지는 flow들을 가지고 있으므로 보다 scheduling이 단순해지는 장점이 있을 수 있으나, 같은 서비스 클래스에 속하는 packet들은 모두 하나의 queue에 삽입됨으로써 각 flow간의 fairness를 보장해 주기에는 무리가 있다.

2.2 Two-level scheduler

그림 3은 two-level scheduler의 구조를 보여주고 있다. One-level scheduler와 다른 점은 scheduler가 2계층으로 되어 있다는 것이다. 먼저 SS에는 각 서비스 클래스를 담당하는 4개의 scheduler가 존재한다. 이 scheduler는 같은 특성을 가지는 traffic들에 대해서 scheduling을 수행하기 때문에, 보다 효과적인 scheduling이 가능하며 QoS를 보다 더 잘 보장해 줄 수 있다. 각 서비스 클래스를 담당하는 scheduler는 aggregate scheduler와 내부적으로 통신하여, 요구하는 대역폭 정보를 전달한다. 이 정보를 취합하여 aggregate scheduler는 BS에게 대역폭을 요청한다. BS는 앞 절에서 설명하였듯이 UL-MAP을 이용하여 각각의 SS에게 대역폭을 할당하고 aggregate scheduler는 UL-MAP으로 할당 받은 대역폭을 각각의 서비스 클래스를 담당하는 scheduler에게 재할당하게 된다.

Two-level scheduler의 가장 큰 장점은 하나의 scheduler가 같은 특성을 가지는 connection들에 대해서 scheduling을 수행함으로써 각 scheduler의 구조가 간단해지고, 각 connection의 QoS를 보다 더 잘 보장해 줄 수 있다는 점이다. 즉 각각의 서비스 클래스는 서로 다른 QoS 요구 조건을 가지고 있으므로 one-level scheduler에서 모든 요구 조건을 보장해 주기 위해서는 알고리즘이 복잡해 지기 마련이다. 하지만 two-level scheduler에서는 하나의 scheduler가 보장해야 하는 QoS 조건이 scheduling 해야 하는 모든 connection에 대해서 동일하므로 알고리즘을 단순하게 디자인 할 수 있다. 반면 SS내에 각 서비스 클래스를 담당하는 scheduler와 이들을 관리하는 aggregate scheduler가 존재함으로 인해서 내부 구조가 복잡해지는 단점이 존재한다. 이는 scheduling을 통해서 다음 packet을 선택하는 과정에서 시간이 더 소요되는 요인이 되며, 이는 결국 전체 구조를 복잡하게 만드는 원인이 된다.

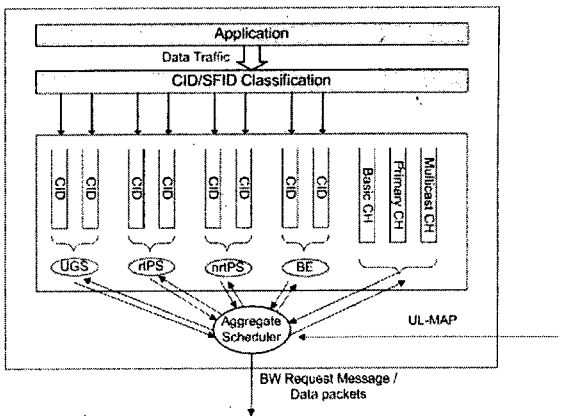

그림 3. Two-level schedulers

One-level에서 제안하였던 두 가지 queue 구조 중에서 two-level에서는 flow queue만을 사용하게 된다. Class queue는 하나의 class에 속하는 모든 packet을 가지고 있게 되는 하나의 queue를 가지게 되는데, 이는 two-level 구조에서는 각 서비스 클래스의 scheduler가 단지 하나의 queue만을 가지는 경우가 되므로 scheduling의 의미가 없어지게 된다. 따라서 two-level에서는 flow queue만을 사용하도록 한다.

3. 결론

본 논문에서는 IEEE 802.16에서 SS에서의 보다 효율적인 대역폭 사용을 위해서 scheduler 구조를 제안하였다. One-level scheduler는 SS에서 생성되는 모든 traffic에 대해서 하나의 scheduler를 사용함으로써 QoS를 보장해 주고 있으며, 이에 따라 scheduling 알고리즘이 복잡해지는 단점이 존재한다. 반면 two-level scheduler에서는 각 서비스 클래스마다 독립된 scheduling 알고리즘을 사용함으로써 보다 더 QoS를 잘 보장해 줄 수 있지만, scheduler 구조가 복잡해지는 단점이 존재한다. 향후 연구에서는 제안한 scheduler 구조를 이용하여 각각에 적합한 scheduling 알고리즘을 제안하고 이를 성능 평가 함으로써 각 scheduler 구조에 맞고 대역폭을 효율적으로 사용하며, 각 connection의 QoS를 만족시켜줄 수 있는 scheduling 알고리즘에 대한 연구를 수행하여야 한다.

참고문헌

[1] IEEE 802.16-2004, *Air Interface for Fixed Broadband Wireless Access Systems*.

[2] Mohammed Hawa and David W.Petr, *Quality of Service Scheduling in Cable and Broadband Wireless Access System*, IEEE International Workshop on Quality of Service, pp. 247-255, 2002.

[3] G.Nair, J.Chou, T.Madejski, K.Perycz, D.Putzolu and J.Sydir, *IEEE 802.16 medium access control and service provisioning*, Intel Technology Journal, vol. 8, no. 3, pp. 213-228, 2004

[4] Dong-Hoon Cho, Jung-Hoon Song, Min-Su Kim, and Ki-Jun Han, *Performance Analysis of the IEEE 802.16 Wireless Metropolitan Area Network*, IEEE International Conference on Distributed Frameworks for Multimedia Applications, 2005.