

고속 무선 PAN(Personal Area Network)에서의 위치기반 동적 스케줄링

곽동원* 박무성* 이승형* 정창모** 윤원용**
 광운대학교 무선네트워크 연구실, 삼성종합기술원
 {rednjs, parkms, shr}@kw.ac.kr {changmo.chung, wyoon}@sait.samsung.co.kr
 Location Dependent Scheduling in 802.15.3 High-rate WPAN

Dongwon Kwak Moosung Park Seung Hyong Rhee Changmo Chung Wonyong Yoon
 Wireless Networking Research Lab, Kwangwoon University*
 Samsung Advanced Institute of Technology**

요약

무선통신에서의 전송 매체인 air interface는 주위의 여러 환경적인 요인에 의해서 다양한 에러가 존재할 수 있다. 더욱이 ad hoc 환경인 802.15.3 High-rate WPAN에서는 이러한 에러에 대한 의존도가 더욱 클 수 있다. 본 논문에서는 802.15.3의 MAC enhancement를 목표로 위치에 따라서 달라지는 에러를 이용해서 할당하는 채널 시간의 크기를 동적으로 변화시켜서 전체 채널 사용 효율을 향상 시키려고 시도한다.

1. 서론

IEEE 802.15.3 표준은 10 m내의 개인 주변 영역 (POS: Personal Operating Space)에서 소비자 가전 또는 통신 디바이스들 간에 저복잡도, 저비용, 저소비전력, 고속의 무선 연결을 위한 PHY, MAC 규격을 제공한다. 802.15.3 High rate WPAN은 10m 이내의 무선 환경에서 실시간 비디오 및 고품질 오디오, 대용량 파일 전송을 가능케 하는 기술이다. 무선 환경아래 디지털 캠코더에서 TV 스크린으로 재생하거나 비디오 파일을 PC 등으로 저장하는 경우를 생각할 수 있다. 이 기술은 기본적으로 각 무선 노드들이 채널을 경쟁을 통해서 사용하는 Carrier Sensing Multiple Access/Collision Avoidance (CSMA/CA)와 주기적인 Contention Free Period(CFP)구간을 이용해서 중앙의 컨트롤러 역할을 하는 PicoNet Coordinator(PNC)가 타임 슬롯을 할당하며 할당 받은 타임 슬롯을 이용해서 데이터 전송을 하는 Time Division Multiple Access(TDMA)방식을 사용한다[1].

기존의 Intel사에서 제안한 queue size에 따라서 queue size가 작은 노드에게 더 많은 양을 할당하고 큰 노드에게는 적게 할당하는 방식은 중앙의 controller가 항상 각 노드들의 queue상황을 모니터링하고 있어야 하므로 전체적인 오버헤드가 상당히 늘어나는 단점이 있다[2].

현재의 채널시간 할당 방법은 다양한 종류의 stream을 제대로 지원하지 못하는 한계를 지닌다. PNC는 각 무선 노드들이 필요로 하는 채널시간을 아무런 여과 없이 그대로 반영한다[3]. 다시 말해서 PNC가 할당한 채널시간

이 할당받은 무선노드와 PNC사이에서 발생하는 error에 따라서 부족할 경우도 있고 적당한 경우도 있다. 특히 채널에 에러발생이 많아서 할당받은 채널 시간을 제대로 사용 못하는 경우, 무선노드가 요구한 양을 그대로 할당하는 것은 비효율적이다. 그러므로 High-rate WPAN에서 전송효율을 높이기 위해서는 각 무선 노드들이 요구하는 채널 시간을 각 노드들의 에러상황에 따라 가감하여 반영, PNC가 동적으로 할당을 해 준다면 전체적인 효율은 크게 향상될 것임을 예측할 수 있다.

2. IEEE802.15.3 High-rate WPAN

802.15.3 WPAN은 기본적으로 superframe이라는 주기적인 단위를 이용해서 각 무선 노드들이 데이터 전송을 가능하게 한다. 그림 1은 superframe의 구조를 나타낸다.

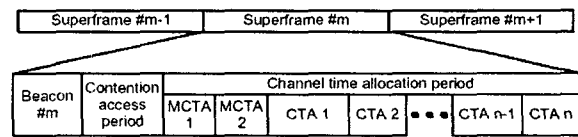


그림 1. superframe의 구조

전체적으로 channel의 상태정보, channel의 이용현황, 각 Device(DEV)들의 synchronization등을 위해서 필요한 Beacon부분과 CSMA/CA 방식으로 채널을 사용하기 위해서 각 DEV들끼리 경쟁을 하는 (Contention Access Period)CAP 구간과 TDMA 방식을 사용해서 각 DEV당 필

요한 타임 슬롯을 PNC가 할당해주는 구간인 Channel Time Access(allocation) Period(CTAP) 혹은 Contention Free Period(CFP)의 3가지 부분으로 나뉜다.

3. Location Dependent Scheduling(LDS)

LDS에서는 노드가 요구한 채널 시간을 그대로 할당하는 것이 아니라 그 노드의 채널상태에 따라서 차등적으로 적용한다. Piconet은 channel status를 계산하는 device와 계산된 값을 받아서 동적으로 채널 시간을 할당하는 PNC로 구성된다.

3.1 Device

각 device는 기본적으로 각 channel status를 계산하며 그것에는 두 가지 방법이 있다. 첫 번째로 moving average를 이용하는 방법으로 moving average란 그림 2에 나타나 있다. Moving average를 20으로 설정하고 설명을 하면 가장 최근의 20개의 채널 상태의 평균을 다음 번 channel status value로 사용한다. 처음 시작할 때의 20개의 값은 전송되는 프레임이 증가함에 따라서 그림 2의 channel status value 처럼 값이 선택된다. 그 이후 21번째 값부터는 moving average를 이용해서 선택된다. channel status는 보낸 프레임의 수와 각 프레임에 대한 ACK수의 비로 정한다. 이렇게 정해진 값은 PNC의 channel status request에 대한 응답으로 PNC에게 보내지며 PNC는 그림 3처럼 할당할 채널 시간을 계산하며 beacon을 통해서 piconet에 속한 모든 무선 노드들에게 알린다.

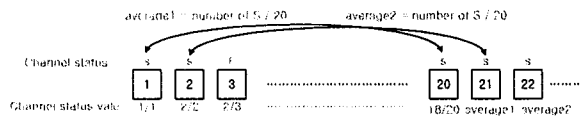


그림 2. moving average 계산

두 번째는 2-state Markov chain을 이용하는 방법으로 각 노드들은 자신들이 보낸 프레임이 성공적으로 전송될지 그렇지 못할지를 결정하고 프레임들이 보내짐에 따라서 2번 연속으로 성공적으로 성공할 확률과 2번 연속으로 실패할 확률을 구한다. 이렇게 구해진 값은 가장 최근에 보낸 프레임이 성공적으로 전송되었다면 전송이 성공적이었다는 정보와 2번 연속적으로 성공할 확률을 PNC의 channel status request에 대한 응답으로 넘겨주며 실패했다면 전송에 실패했다는 정보와 2번 연속으로 실패할 확률을 PNC의 channel status request에 대한 응답을 통해서 넘겨준다.

3.2 PNC

첫 번째 방법인 moving average를 이용한 방법에서 각 device들의 channel status를 넘겨받은 PNC는 DEV가 요구한 CTA와 각 DEV의 channel 상태의 곱의 평균을 구한다. 그 후 각 DEV가 요구한 CTA의 양과 channel 상태를 곱한 후 이전에 구했던 평균과의 비율로서 할당할 채널 시간의 양을 구한다. CTA를 요구한 DEV에게 PNC가 할당할 채널 시간의 양은 다음과 같이 구해진다.

$$\text{Assigned CTA} = \text{requested CTA} * \text{channel state} / \text{average channel state}$$

$$\text{Channel state} = \text{requested CTA} * \text{channel status}$$

$$\text{Average channel state} = \text{all DEV's channel state} / \text{all DEV number}$$

표 1. CTA 할당을 위한 예제

	status	requested CTA
Device1	0.3	3000 μ s
Device2	0.5	5000 μ s
Device3	0.7	7000 μ s

두 번째 방법인 2-state Markov chain에서는 넘겨받은 정보인 전송에 성공했는지 혹은 실패했는지와 성공했다면 연속적으로 2번 성공할 확률 값 실패했다면 연속적으로 2번 실패할 확률 값 등 2가지 값을 바탕으로 전송된 확률 값과 PNC가 0~1사이에서 선택한 random 변수 값의 비교를 통해서 다음 프레임의 전송이 실패할 것이라고 예측되면 요구한 채널 시간을 줄일 것이며 성공할 것이라고 예측되면 늘릴 것이다. 우선 실패할 것으로 예측된 device들에게 회수할 채널 시간의 양을 그림 4처럼 모두 더하고(save) 다음에 성공이 예상되는 노드의 숫자(s_node)를 구한다. Piconet에 속한 모든 노드들의 다음 전송이 성공으로 예상되거나 실패로 예상되면 모든 노드들에게는 요구한 채널시간을 그대로 할당하며 그 외의 경우 구해진 양은 그림 5처럼 성공이 예상되는 노드의 성공확률에 비례적으로 할당해준다.

4. 시뮬레이션

표 1은 시뮬레이션 환경을 나타내고 있다. 시뮬레이터는 ns2-2.1b9a버전을 사용했으며 [4] 여기에 802.15.3 MAC을 구현한 기존의 소스코드 [3]에 수정을 통해서 수행하였다. 802.15.3에서 지원 가능한 최대의 대역폭이 55Mbps이므로 이를 최대의 대역폭으로 설정했고, High Definition(HD)급 화질을 요구하는 media streaming의 경우 최소 18(Mbps)의 traffic rate가 필요하므로 CBR rate는 대략 20(Mbps)으로 선정하였다.

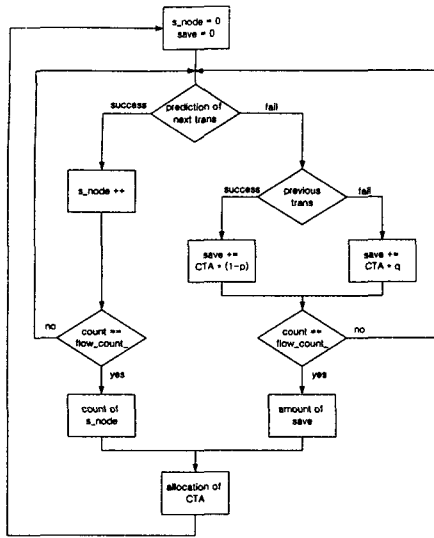


그림 4. s_node와 save의 계산

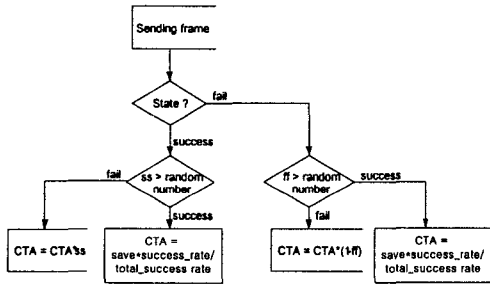


그림 5. scheme2 에서의 CTA 할당량 계산

표 2. 시뮬레이션 환경

Attribute	Value
bandwidth	30,40,55Mbps
number of flows	3
moving average	20
run time	10(simulation time)
CBR rate	20Mbps
CBR packet size	1024
CTA size	4000μs
error rate	0, 0.3, 0.5

그림 6은 3개 flow의 error rate가 0:30(%):50(%)일 경우 대역폭에 따른 aggregate throughput의 변화를 보여 준다. 전체적으로 scheme1의 성능이 가장 높았으며 scheme2도 scheme1보다는 약간 성능이 낮지만 대체적으로 standard 방식보다는 높게 나타나는데 이것은 그림 7처럼 error가 없는 flow1의 throughput이 scheme1,2에

서 standard보다 상당히 높기 때문이다.

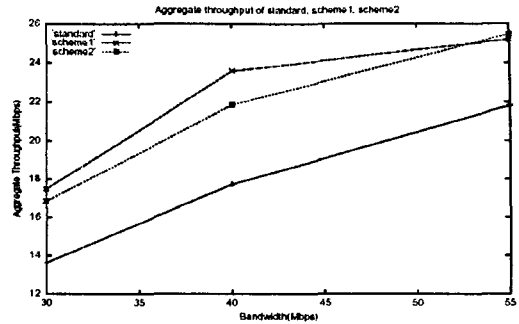


그림 6. 대역폭에 따른 aggregate throughput의 변화

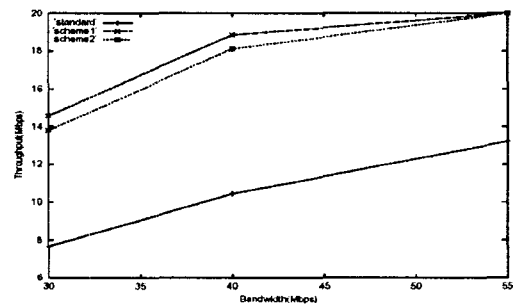


그림 7. flow1의 throughput 변화

5. 결론

무선 환경에서는 주위환경 및 위치변화 의한 에러 가능성이 유선환경보다 월등히 높으므로 throughput을 유지하는데 에러의 영향은 상당히 중요하다. 사용자가 요구하는 traffic rate가 점점 높아지고 있는 상황에서 에러가 발생해도 요구한 채널 시간을 그대로 할당하는 standard 방식보다는 에러 상황에 따라서 요구한 채널 시간에 가감을 하여서 할당해주는 LDS의 성능이 높을 것이라는 것은 쉽게 예상할 수 있으며 그 결과는 시뮬레이션을 통해서 확인할 수 있었다.

참고문헌

- [1] Wireless Medium Access Control (MAC) and Physical Layer(PHY) Specifications for High Rate Wireless personal Area Networks(WPAN), IEEE, Draft P802.15.3/D15, Oct. 2002
- [2] http://grouper.ieee.org/groups/802/15/pub/2002/Jul02/02297r1P802-15_TG3_Performance-Analysis-of-MAC-and-QoS.ppt
- [3] <http://www.winlab.rutgers.edu/~demirhan/research>
- [4] The CMU Monsrch Project, "Wireless and mobile extension to ns." Snapshot Release 1.1.1, Carnegie Mellon University, Aug. 5, 1999