

무선 랜 환경에서의 비트맵을 이용한 경쟁 기반 다중 폴링 기법에 관한 연구

남용섭, 류지호, 최낙중, 권태경, 최양희
서울대학교
{ysnam, jhryu, fomula, tk, yhchoi}@mmlab.snu.ac.kr

Contention-based Multi-Polling Mechanism for Wireless LANs Using Bitmap

Yongsu Nam, Jiho Ryu, Nakjung Choi, Taekyoung Kwon and Yanghee Choi
School of Computer Science and Engineering
Seoul National University, Seoul, Korea

요약

IEEE 802.11 무선 랜의 DCF 기법은 backoff 로 인한 채널 유휴 시간과 충돌로 인해 실제 가능한 링크 대역폭에 못 미치는 효율을 보인다. 이의 대안으로 폴링 기법을 고려할 수 있으나, 기존의 폴링 기법은 개개의 단말을 폴링해야 하고, 또한 보낼 프레임이 없는 단말에게도 폴링을 수행하게 되어 채널을 낭비하게 된다. 본 논문은 무선 랜의 성능을 높이기 위해 비트맵을 이용한 경쟁 기반 다중 폴링 기법을 제안한다. 액세스 포인트가 주기적으로 전송 순서를 기록한 비트맵을 브로드캐스트하며, 이를 송신한 서비스 지역 내의 모든 단말이 자신의 전송 순서를 유추한다. 전송 순서가 결정되어 있으므로 충돌이나 backoff 로 인한 채널 낭비가 발생하지 않아 채널을 효율적으로 사용한다. 또한, 하나의 프레임으로 BSS 내의 모든 단말을 폴링하며, 청크 기반으로 폴링 목록을 전송하므로 폴링 오버헤드 역시 감소한다. 모의 실험을 통한 성능 평가 결과는 제안 기법의 우수한 성능을 입증한다.

I. 서론

IEEE 802.11 무선 랜 표준 [1]은 현재 무선 인터넷의 기반 기술로 널리 사용되는 표준의 하나로, CSMA/CA 방식의 채널 접근 기법 (DCF)을 채택하여 무선 환경에 적합하게 설계되었다. 그러나, 분석과 실험을 통한 많은 연구 결과들은 실제 대역에 비해 낮은 효율을 보이는 단점을 지적해 왔다. 이러한 성능 저하의 가장 큰 원인으로 backoff 와 충돌로 인한 대역의 낭비를 들 수 있다.

IEEE 802.11 표준은 별도의 폴링 기법 (PCF)을 정의하고 있으며, 이는 액세스 포인트 (AP)가 리스트 내의 각각의 단말들에게 차례로 채널 접근 권한을 주는 방법을 채택하였다. 그러나, PCF 사용 시, 단말에게 권한을 주기 위해 CF-Poll 메시지를 전송하는데, 폴링 리스트에 있는 모든 단말에게 매번 CF-Poll 메시지를 전송하므로 성능 저하를 유발하며, 또한 보낼 프레임이 없는 단말에게도 이 메시지를 전송하게 되어 채널 사용상의 낭비를 야기하게 된다.

TDMA 와 같은 슬롯 기반의 채널 접근 기법은 채널을 충분히 활용할 수 있는 장점이 있으나, 단말들에게 슬롯을 할당하는데 있어, 즉 채널 접근 순서를 결정하는데 있어, 고정적인 채널 낭비가 발생한다.

본 연구는 무선 랜 환경에서 채널 사용을 극대화하기 위해, DCF 와 PCF, 그리고 TDMA 방식의 채널 접근 기법의 단점을 보완하여, 경쟁 기반 다중 폴링 기법을 사용한 채널 접근 기법을 제안한다. 기본적으로 AP 가 하나의 폴링 메시지를 브로드캐스트하며, 메시지 내의 비트맵을 통해 서비스 영역 내의 모든 단말들이 자신의 전송 순서를 알아내어 전송을 수행한다.

본 논문은 다음과 같이 구성된다. 2 장에서 관련 연

구를 살펴보고, 3 장에서 제안한 기법을 소개한다. 4 장에서 성능 분석 및 모의 실험을 통해 제안 기법의 성능을 평가하고, 마지막으로 5 장에서 본 논문을 마무리 짓는다.

II. 관련 연구

PCF 의 오버헤드를 줄이기 위해 많은 연구가 이루어져 왔다. [3]에서는 보낼 프레임이 없는 단말에게 폴링하여 낭비되는 비용을 줄이기 위해 동시 폴링 기법을 제안한다. 동시 폴링 기법은 활성/유휴 상태의 단말을 각각의 링으로 관리하며, 활성 링 내의 단말에게 차례로 폴링을 수행한다. 활성 링 내의 단말은 보낼 프레임이 고갈된 경우 유휴 링으로 이동하며 다시 보낼 프레임이 발생하는 경우 활성 링으로 이동한다. 보낼 프레임이 없는 단말에게 폴링하는 오버헤드를 줄이나, 각각의 단말에게 폴링하는 비용은 여전히 남게 된다.

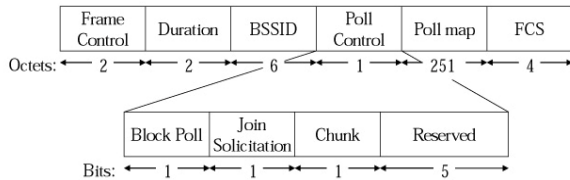
[4][5][6]은 IEEE 802.11e [2] 환경에서 다중 사용자 폴링을 위해 제안되었다. [4]에서 각각의 단말은 자신이 전송하는 프레임의 뒤에 다음에 폴링될 단말의 주소를 추가하여 차례로 폴링을 수행한다. 이러한 폴링 방식은 채널 에러에 취약하며, 또한, 채널 사용상의 낭비는 여전히 남게 된다. [5][6]은 폴링 목록을 담은 하나의 폴링 프레임을 통해 경쟁 기반 다중 폴링을 수행한다. 폴링 프레임을 받은 단말들은 리스트 상의 순서에 따라 backoff 타이머를 결정하여 폴링을 수행하여 채널 낭비를 줄인다. 그러나, 폴링 목록은 단말 수가 늘어남에 따라 목록의 크기도 증가하여 채널 사용을 극대화하지 못한다. 또한, IEEE 802.11 에 기반한 다중 폴링의 경우, TXOP 과 같은 부가적인 오버헤드를 수반한다.

III. 제안 기법

본 연구는 폴링으로 인한 오버헤드를 줄여 궁극적으로 모든 슬롯이 단말에게 할당되는 TDMA의 기법과 유사한 채널 액세스를 수행하고자 한다. 이를 위해, 무선 랜 사용자의 트래픽 패턴의 특성에 시간적 지역성이 있다고 가정한다. 즉, FTP나 World Wide Web 사용과 같이 무선 랜을 일정 시간 사용하고 채널 접근을 일정 시간 동안 중단하며 다시 일정 시간 채널 접근을 수행하는 패턴을 보인다고 가정한다.

A. 제안 기법

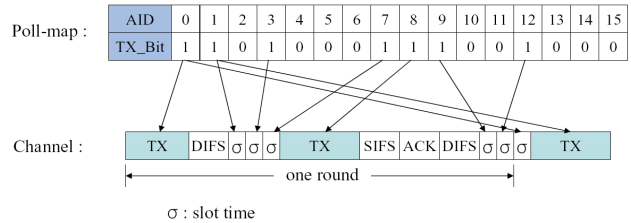
AP는 주기적으로 Block-Poll 프레임을 브로드캐스트하여 서비스 지역 내의 모든 단말들에게 폴링 순서를 알린다. Block-Poll 프레임은 비트맵(Poll-map)을 포함하고 있으며, 비트맵의 각 비트는 각 단말의 AID에 대응한다. 비트맵은 251 바이트로서, 총 2007개의 AID를 관리할 수 있으며, AID 0은 AP에 해당한다. 비트맵의 비트가 1로 설정된 경우, 해당 AID에 대응하는 단말은 전송할 기회를 얻으며, 0으로 설정된 경우 전송할 수 없다. 즉, 단말이 보낼 프레임 가지고 있고, 비트맵 상의 대응 비트가 1로 설정된 경우에만 프레임이 전송된다. Block-Poll 프레임의 구조는 그림 1과 같다. 기본적인 IEEE 802.11의 MAC 프레임 헤더와 유사하며, Block-Poll 메시지를 구분하기 위한 Poll-Control 필드와 Poll-map이 추가되었다. Poll-Control 필드에 대한 자세한 내용은 이후에 소개한다.



[그림 1] Block-Poll 프레임 구조

Block-Poll을 받은 단말은 Poll-map에서 자신의 AID에 대응하는 비트 이전의 비트 중 1의 개수만큼 backoff 타이머를 설정한다. 즉, 자신의 비트 이전의 비트 중 3개의 비트가 1로 설정되어 있다면 backoff 타이머를 3으로 설정한다. 이후의 과정은 DCF의 경우와 마찬가지로 채널이 유휴 상태이면 슬롯 하나마다 backoff 타이머를 하나씩 줄이고, 0이 되는 시점에서 프레임 전송을 시도한다. (채널이 사용 중이었다가 유휴 상태가 되는 경우, DCF의 경우와 마찬가지로 DIFS 후에 backoff 타이머 감소 과정을 수행한다.) 만약 해당 단말에 대응하는 비트가 0인 경우 backoff 타이머를 설정하지 않는다. 이러한 방식으로 프레임 전송이 TDMA와 유사하게 슬롯 단위로 이루어진다. 만약 해당 비트가 1로 설정되어 있거나 보낼 프레임이 없는 경우 자신의 backoff 타이머를 설정하지 않는다. 이로 인해 채널 낭비가 발생할 수 있으나, 하나의 슬롯만 낭비되므로 채널 효율에 거의 영향을 미치지 않는다. 또한, 미리 지정된 M개의 라운드 동안 프레임을 전송하지 않은 단말은 이후 Poll-map에서 0으로 설정되어 채널 낭비를 막는다.

뿐만 아니라, 제안 방식에서는 사용자 트래픽의 시간적 지역성을 가정하고 있으므로, 비트맵은 시간에 따라 자주 변하지 않는다. 따라서, Block-poll 프레임을 매 라운드마다 보낼 필요가 없으며, 결국 Poll-map의 오버헤드는 급격히 줄게 된다. 마찬가지로 단말이 보낼 프레임을 가지고 있지 않으나 비트맵의 비트가 1로 설정되어 낭비되는 슬롯 또한 트래픽의 시간적 지역성에 의해 성능에 거의 영향을 미치지 않게 된다.



[그림 2] Block-Poll 동작 예

B. 청크 (Chunk) 기반 폴링

Block-Poll 프레임의 Poll-map은 크기가 고정되어 있으므로 단말 수가 적은 경우에 상대적으로 오버헤드가 커지게 된다. 이러한 단점을 보완하고자 청크 기반 폴링 기법을 제안한다. Poll-map을 일정 크기의 청크로 나누어 관리하며, Block-Poll 프레임을 보내는 경우에도 변경된 청크만 전송한다. 각 청크는 그림 3에 나타난 것처럼 청크 번호와 해당 청크에 대응하는 비트의 집합으로 구성된다. 사용자 트래픽의 시간적 지역성을 고려하면 한 시점에 변경되는 비트는 극히 드물므로 전송되는 청크의 수 역시 급격히 줄어든다.

Chunk #	AID	0	1	2	3
0	TX_Bit	1	1	0	0

Chunk #	AID	2004	2005	2006	2007
251	TX_Bit	1	0	0	0

[그림 3] 청크 구조 예

이러한 방법으로 Poll-map의 오버헤드를 줄일 수 있게 된다. 청크로 구분하여 전송하는 경우, Block-Poll 프레임의 Poll-Control 필드에서 Chunk 비트를 1로 설정한다. 반면, Poll-map을 전송하는 경우에는 Chunk 비트를 0으로 설정한다.

C. Join/Leave

다른 폴링 프로토콜과 달리 제안된 방식은 별도의 경쟁 구간을 두지 않는다. 따라서, 새 단말이 폴링에 참여하고 싶은 경우 이를 알리기 위한 방법이 필요하다. 따라서, AP는 매 라운드 혹은 몇 개의 라운드마다 Join-Solicitation 프레임을 전송한다. 프레임 구조는 Block-Poll과 동일하며, Poll-Control 필드의 Join-Solicitation 비트를 1로 설정한다. 그리고, Poll-map의 경우에도 일반 Poll-map을 역으로 뒤집은 비트맵을 전송한다. 즉, Poll-map에서 0으로 설정되어 있던 단말은 Join-Solicitation 시 1이 되고 1로 설정되어 있던 단말은 0으로 설정된다.

만약 보낼 프레임이 있는 단말은 이에 응답하여 프레임을 전송하면 되고, 보낼 것이 없는 단말은 해당 슬

롯을 유휴 상태로 두면 된다. 폴링 목록에서 빠지기 위해서는 일정 라운드 동안 프레임이 전송하지 않으면 되며, 이 경우도 트래픽의 시간적 지역성에 기인한다. 미리 지정된 M 개의 라운드 동안 프레임이 전송하지 않은 단말은 이후의 Block-Poll 전송 시 Poll-map 에서 0 으로 설정되게 된다.

IV. 성능 평가

수학적 분석과 모의 실험을 통해 제안 기법과 기존 DCF 의 성능을 평가하였으며, 각각의 성능 평가 결과를 비교하였다. 포화 트래픽과 버스트 트래픽의 경우를 구분하여 성능 평가를 수행하였다.

A. 성능 분석

제안 기법의 성능을 분석하기 위해 특정 시간, 즉 M 라운드, 동안의 채널 사용 형태를 추정하였다. 포화 트래픽의 경우 모든 단말이 보낼 프레임을 가지고 있으므로, 수월하게 추정이 가능하다. 즉, BSS 내에 N 개의 단말이 있는 경우, 효율 T_p 는 아래 식과 같이 계산 가능하다.

$$T_p = \frac{T_T}{T_T + T_{CO}}$$

$$T_T = E[P] \cdot M \cdot N$$

$$T_{CO} = (H_d + DIFS + SIFS + ACK) \cdot M \cdot N + H_b$$

위 식에서 T_T , T_{CO} 는 각각 프레임 전송에 사용된 시간과 제어 오버헤드를 의미한다. 또한, $E[P]$, H_d , H_b 는 각각 프레임의 평균 크기, 데이터 프레임의 헤더 크기, Block-poll 의 헤더 크기를 의미한다. 즉, 모든 단말이 보낼 프레임을 가지고 있으므로, backoff 로 인해 낭비되는 슬롯은 없으며, 추가되는 오버헤드는 헤더와 IFS, ACK 뿐이다. 버스트 트래픽 경우의 성능 분석은 포화 트래픽의 경우보다 다소 복잡하나, 유사한 방법을 통해 추정이 가능하다. 즉, 각 단말이 일정 시간 τ_A 동안 프레임을 보내고 τ_I 동안 유휴 상태에 있는 과정을 반복한다고 가정하면, 버스트 트래픽 상황에서의 제안 기법의 효율은 아래 식과 같이 계산이 가능하다.

$$T_p = \frac{T_T}{T_T + T_I + T_{BO} + T_{CO}}$$

$$T_T = E[P] \cdot M \cdot N \cdot \frac{t_A}{t_A + t_I}$$

$$T_I = s \cdot M \cdot N \cdot \frac{t_I}{t_A + t_I}$$

$$T_{BO} = \frac{2 \cdot E[R]}{t_A + t_I} \cdot \left(\frac{N}{K} \cdot T_C \right) \cdot M$$

$$T_{CO} = (H_d + DIFS + SIFS + ACK) \cdot \frac{M \cdot N \cdot t_A}{t_A + t_I} + H_b$$

$$T_C = \left(E[P] \cdot \frac{t_A}{t_A + t_I} + s \cdot \frac{t_I}{t_A + t_I} + \frac{T_{CO}}{M \cdot N} \right) \cdot N$$

위 식에서 $E[R]$ 과 T_C , K 는 각각 각 라운드의 평균 시간, 청크 하나의 전송 시간, 그리고 하나의 청크 내의 단말의 수를 의미한다.

DCF 의 경우, [7]에 다양한 트래픽 환경에서의 DCF 의 성능에 대한 분석이 소개되었다. 따라서, 포화 트래픽 경우의 DCF 의 성능은 [7]에서 도입한 모델을 사용하였다. 버스트 트래픽 경우의 성능은 [7]의 모델에 앞서 제안 기법의 성능 분석에서 사용한 버스트 트래픽 모델을 적용하였다.

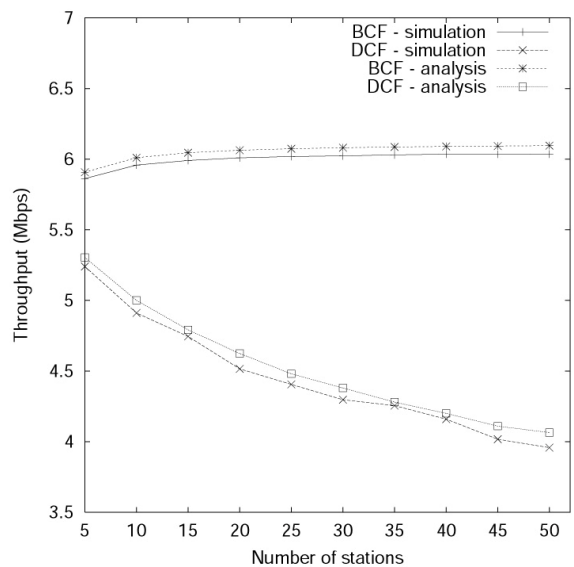
B. 모의 실험 환경

NS-2 를 사용하여 제안한 기법의 성능을 평가하였다. 포화 트래픽과 버스트 트래픽의 경우에 각각 모의 실험을 수행하였으며, 대조군으로 DCF 를 두어 성능을 비교하였다. 실험에 있어 단말의 수는 5 - 50 사이에서 변경하였으며, 프레임의 크기는 1000 바이트로 고정하였고, Block-Poll 과 Join-Solicitation 메시지는 매 10 라운드마다 전송하도록 하였다. 청크 기반 폴링을 기본으로 사용하였으며, 하나의 청크는 8 개의 단말 정보를 포함하도록 하였다. 버스트 트래픽의 경우, 각 단말의 활성 시간과 비활성 시간을 같게 두었다.

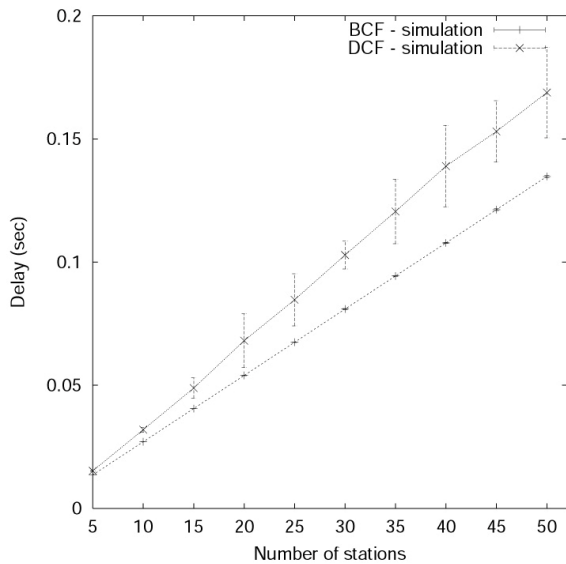
C. 모의 실험 결과

모의 실험과 수학적 분석을 통해 얻은 포화 트래픽 상황에서의 효율과 딜레이는 각각 그림 4, 5 와 같다. 그림 4 에서 제안 기법은 단말 수에 관계없이 거의 6 Mbps 의 효율을 유지하는 반면, DCF 는 단말 수가 증가함에 따라 급격히 효율이 감소하는 것을 확인할 수 있다. 단말의 수가 50 개인 경우, 제안 기법은 6 Mbps 의 효율을 보이며, DCF 는 4 Mbps 의 효율을 보인다. 즉, 약 1.5 배로 성능이 향상됨을 확인할 수 있다. 이는 채널 낭비 원인을 상당수 줄였기 때문으로 해석할 수 있다.

그림 5 에서 제안 기법은 DCF 보다 작은 지연 시간을 보인다. 또한, 제안 기법의 지연 시간이 편차가 없이 고르게 분포하는 반면, DCF 는 편차가 큰 것을 확인할 수 있다. 이는 폴링을 통해 전송 순서가 거의 일정하게 유지되기 때문에 지연 시간 역시 일정하게 유지되는 것으로 해석할 수 있다.

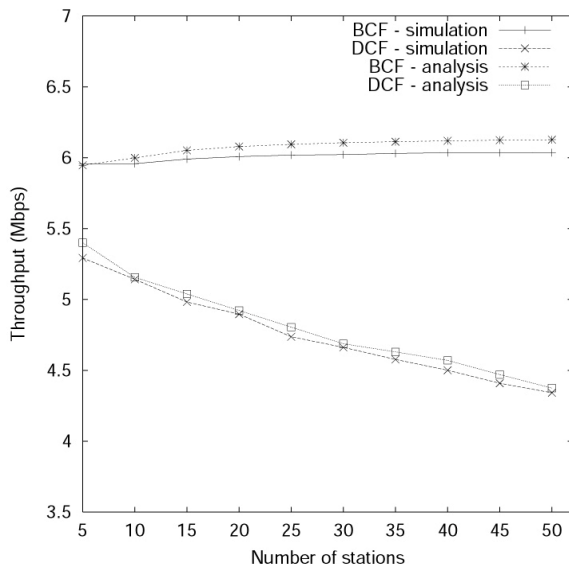


[그림 4] 포화 트래픽에서의 효율 비교



[그림 5] 포화 트래픽에서의 지연 시간 비교

그림 6은 버스트 트래픽 경우의 DCF와 제안 기법의 효율을 비교한다. 모의 실험에 있어서 활성/비활성 시간은 각각 1 초씩으로 두었다. 즉, 각 단말은 1 초 간격으로 포화 트래픽 상태와 유휴 상태를 반복한다. 이 경우, 제안 기법은 6 Mbps의 효율을 유지하는 것을 확인할 수 있으며, DCF의 경우 단말 수에 따라 감소하는 것을 확인할 수 있다. 그러나, DCF의 경우, 포화 트래픽 경우에 비해 좋은 성능을 보이는데, 이는 버스트 트래픽의 특성 상 한 시점에 경쟁에 참여하는 단말의 수가 포화 트래픽의 경우보다 작기 때문에 충돌로 인한 성능 감소가 줄었기 때문이다.



[그림 6] 버스트 트래픽에서의 효율 비교

버스트 트래픽의 성능 평가에 있어 활성/비활성 시간을 동일하게 1 초로 두었으나 시간이 서로 다른 경우에도 유사한 성능을 보일 것으로 예상된다. 왜냐하면, 성능 분석 결과 식에서 볼 때, 성능에 직접적으로 영향을 미치는 것은 전체 시간 중 활성 시간의 비율이기 때문이다.

V. 결론 및 향후 연구 방향

IEEE 802.11 무선 랜 환경의 성능을 높이기 위해 비트맵을 이용한 경쟁 기반 다중 폴링 기법을 제안하였다. 우리가 제안한 기법은 비트맵을 통해 서비스 지역 내의 모든 단말이 자신의 전송 순서를 얻게 되어 채널을 효율적으로 사용하는 방식이다. 수학적 분석과 모의 실험을 통한 성능 평가 결과는 제안 방식의 우수한 성능을 입증하였다. 향후에는 QoS 제공을 위해 스케줄링 기법을 도입하여 라운드 로빈 방식이 아닌 우선순위에 따른 서비스 제공을 할 수 있도록 할 예정이다.

VI. 참고문헌

- [1] IEEE Computer Society. 802.11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, June 1997.
- [2] IEEE Computer Society. 802.11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications Amendment 8: Medium Access Control (MAC) Quality of Service Enhancements (802.11e), Nov. 2005.
- [3] O. Sharon and E. Altman, "An Efficient Polling MAC for Wireless LANs," IEEE/ACM Transactions on Networking, vol. 9, no. 4, pp. 439-451, Aug. 2001.
- [4] A. Ganz and A. Phonphoem, "Robust SuperPoll with Chaining Protocol for IEEE 802.11 Wireless LANs in Support of Multimedia Applications," Kluwer Wireless Networks, vol. 7, no. 1, pp. 65-73, Jan. 2001.
- [5] S. Lo, G. Lee and W. Chen, "An Efficient Multipolling Mechanism for IEEE 802.11 Wireless LANs," IEEE Transactions on Computers, vol. 52, no. 6, June 2003.
- [6] S. Kim, Y. Kim, S. Choi, K. Jang and J. Chang, "A High-Throughput MAC Strategy for Next-Generation WLANs," Proc. IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks, June 2005.
- [7] M. Garetto and C. Chiasserini, "Performance Analysis of the 802.11 Distributed Coordination Function under Sporadic Traffic," Proc. IFIP Networking Conference 2005.