

IEEE 802.11 WLAN에서의 DCF 성능개선 방안

Performance Improvement Method of IEEE 802.11 WLAN DCF

박두진, 고동엽, 임석구
백석대학교

Park doo-jin, Ko dong-yub, Lim seog-ku
Baekseok Univ.

요약

IEEE 802.11 WLAN(Wireless Local Area Network)의 MAC계층의 DCF(Distributed Coordination Function)는 각 스테이션이 동시에 매체 접속을 시도하여 발생하는 충돌을 해결하기 위하여 CSMA/CA(Carrier Sense Multiple Access with Collision Avoidance) 알고리즘을 이용한다. 하지만 DCF는 스테이션의 수가 증가함에 따라 성능이 저하되는 단점이 있다. 본 논문에서는 이러한 스테이션 수의 증가에 따른 혼잡상황에서 성능이 개선된 알고리즘을 제안하고, 기본 알고리즘과 DCF+ 알고리즘, 그리고 제안된 알고리즘을 시뮬레이션 도구인 NS-2(Network Simulator-2)를 이용하여 포화수율, 지연시간, 충돌율, 드랍율의 관점에서 비교분석하였다.

Abstract

DCF(Distributed Coordination Function) of IEEE 802.11 WLAN(Wireless Local Area Network) MAC Layer apply to CSMA/CA(Carrier Sense Multiple Access with Collision Avoidance) for solution of collision when the station try to access at the same time. But DCF happens falling performance as increasing a number of stations. In this paper, we suggest algorithm that improved performance in congestion. And we compare DCF, DCF+ and proposed algorithm respect to saturation throughput, delay, collision rate and drop rate using NS2(Network Simulator 2) simulation tool.

I. 서론

최근 IEEE 802.11은 인터넷 서비스를 무선이라는 편의성을 장점으로 수요가 많은 장소에 핫스팟(Hot Spot)을 설치하여 WLAN(Wireless Local Area Network)을 통해 제공한다. 이러한 WLAN을 구현하기 위한 규격인 IEEE 802.11이 1999년에 처음 발표되었고[1], 이후 IEEE 802.11b, IEEE 802.11a, IEEE 802.11g, IEEE 802.11n으로 발전되어 가고 있으며 앞으로 그 수요가 점점 더 증가될 전망이다.

IEEE 802.11 기반 WLAN의 MAC 계층은 PCF(Point Coordination Function)와 DCF(Distributed Coordination Function)기능을 지원한다. PCF는 중앙집중적인 폴링(Polling) 방식으로 폴링 오버헤드와 단말이 전송할 데이터가 없다는 것을 알리기 위하여 전송하는 널(Null) 패킷들로 인하여 무선자원을 비효율적으로 사용하는 문제점이 있다. [2]. 이러한 단점들로 인하여 PCF는 성능 및 유연성에 문제를 나타내고 있어서 대부분의 무선 랜 장비에서 거의 구현되지 않고 있다. 이러한 관점에서 본 논문에서는 DCF의 성능을 개선하는 방안에 대해서만 살펴보고자 한다. DCF는 CSMA/CA(Carrier Sense Multiple Access with Collision Avoidance)를 사용하는 IEEE 802.11의 기본 매체 접근 제어 방식이다.

DCF는 경쟁 단말의 수가 적은 상황에서는 우수한 성능을 보이지만 경쟁 단말의 수가 많아 트래픽의 부하가 생기는 경우에는 포화수율, 충돌율, 드랍율 측면에서 성능 저하되는 문제점을 갖고 있다. 현재까지 이러한 문제점을 보완하기 위한 연구가 진행되어 왔다.

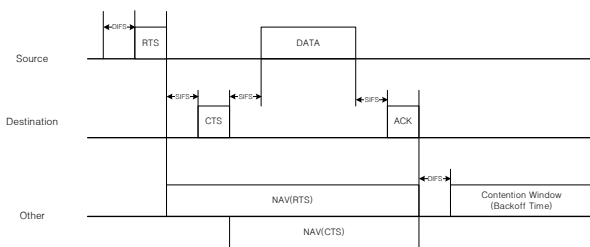
현재까지 진행되어온 연구결과로는 연속적으로 프레임의 전송이 성공하였을 때 백오프 스테이지(Backoff Stage)값이 1씩 감소하는 GDCAF(Gentle DCF)[3], 각 스테이지의 전송 성공 횟수와 충돌 횟수, 백오프 횟수의 평균을 계산하여 이 값을 기반으로 현재 채널에 적합한 경쟁 윈도우(Contention Window)의 크기를 찾는 ADCF(Adaptive DCF)[4], 경쟁 윈도우의 크기를 천천히 감소시킴으로 성공적으로 전송된 패킷에 대한 충돌확률을 다음 패킷에 일부 적용하기 위한 방법인 DCF+[5] 등이 있다.

본 논문에서는 혼잡상황에서 기본 DCF와 위에 이미 제안된 방법 중 하나인 DCF+를 비교 대상으로 하여 우수한 성능을 갖는 알고리즘을 제안한다. 제안하는 알고리즘의 효율성을 입증하기 위하여 시뮬레이션 도구인 NS-2(Network Simulator-2)[6]를 이용하여 포화수율과 지연시간, 충돌율, 드랍율 관점에서 비교 분석한다.

서론에 이어 II장에서는 WLAN에서의 매체 제어 접근 방식으로 기본 DCF, DCF+, 제안하는 알고리즘에 대해서 설명하며, III장에서는 세 가지 알고리즘에 대하여 시뮬레이션을 통해 성능을 비교분석하며, 마지막으로 IV장에서는 결론을 맺는다.

II. WLAN에서의 매체 접근 제어 방식

IEEE 802.11에서 매체 접근 제어는 CSMA/CA방식을 따르며 충돌을 줄이기 위해 Random Backoff Time을 사용한다 [7][8][9]. 그림 1은 DCF 환경에서 각 단말의 동작을 보여준다.



▶▶ 그림 1. DCF의 매체 접근 제어 방식

먼저 Source 스테이션에서 Destination 스테이션으로 보낼 데이터가 있으면 매체가 사용 중인지 감지하여 사용 중이 아니면 DIFS(Distribute Interframe Space)만큼 기다린 후에 데이터를 보내겠다는 신호인 RTS(Request to Send) 제어 프레임을 Destination 스테이션으로 보낸다. 이때 다른 스테이션들은 NAV(No Carrier Sensing)상태로 들어가 매체를 감지하지 않아 매체를 사용하지 않게 되며, Destination 스테이션에서 RTS를 받으면 SIFS(Short Interframe Space)만큼 기다린 후에 Source 스테이션으로 데이터를 받을 준비가 되었다는 것을 알려주기 위해 CTS(Clear to Send) 제어 프레임을 Source 스테이션으로 보내고 다시 SIFS만큼 기다린 후에 Source 스테이션에서 Destination 스테이션으로 데이터를 전송한다. Hidden 스테이션의 문제로 RTS와 CTS를 사용하지 않[10] 본 논문에서는 Hidden 스테이션의 문제는 없다는 가정 하에서 다루지 않는다. 데이터 전송이 완료되면 SIFS만큼 기다린 후에 Destination 스테이션은 데이터를 잘 받았다는 ACK 메시지를 Source 스테이션으로 전송하여 데이터 전송을 마치고 모든 스테이션은 DIFS만큼 기다리고 매체에 접근하기 위해 CW(Contention Window)에 의해 생성된 Random한 Backoff Time이 지난 후에 매체 접근을 시도하게 된다. 각 스테이션별로 다른 Backoff Time을 갖게 되어 충돌을 피하게 된다.

이와 같이 충돌을 피하게 해주는 Rndom한 Backoff Time을 결정하기 위해서 각 스테이션들은 CW_{min} (CW 최소값 = 31)과 CW_{max} (CW 최대값 = 1023)사이

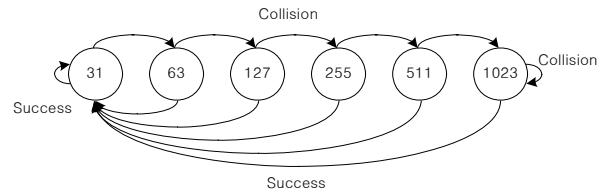
값을 갖게 되어 0에서 현재 CW 값 사이에서 Random값을 Uniform하게 추출하여 $aSlotTime$ 을 곱함으로 Rndom한 Backoff Time을 결정하게 되고 식(1)과 같이 표현 할 수 있다.

$$Backoff\ Time = rand(0 \sim CW) \times aSlotTime \quad (1)$$

그러나 스테이션의 수가 많아지게 되면 Backoff Time이 동시에 0이 되는 스테이션 수가 증가하게 되어 동시에 매체 접근을 시도하는 경우 충돌이 발생할 확률이 높아진다.

1. IEEE 802.11 기본 DCF 알고리즘

IEEE 802.11에서의 기본 DCF는 그림 2에서처럼 충돌 발생 시 CW 값을 두 배로 하여 더 큰 범위에서 Random 값을 추출하여 Backoff Time을 생성하여 충돌 발생 확률을 줄여주게 되고 패킷전송이 성공하게 되면 CW 값을 CW_{min} 값으로 낮춰주게 되는 알고리즘으로 식(2)와 같이 나타낼 수 있다. 이때 CW 의 초기값은 31이고 CW_{n-1} 은 현재 CW 값이고 CW_n 은 다음 전송에 적용될 CW 값이다. 그러나 혼잡상황에서는 CW 값을 두 배로 증가시키는 것으로는 부족하여 연속적인 충돌이 일어나 충돌이 일어날 때마다 DIFS의 시간만큼 기다리고 또 다시 Backoff Time을 기다려야 하기 때문에 효율이 떨어질 수 있다.



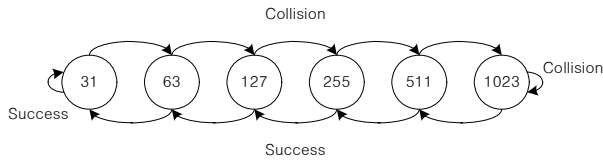
▶▶ 그림 2. IEEE 802.11 기본 알고리즘의 CW상태 천이도

$$CW_n = \begin{cases} \min((CW_{n-1} \times 2) + 1, CW_{max}) & \text{if collision} \\ CW_{min} & \text{if success} \end{cases} \quad (2)$$

충돌을 회피하기 위한 Backoff Time은 CW 값에 의해 결정이 됨으로 CW 값을 결정하는 알고리즘이 WLAN의 성능에 중요한 역할을 한다고 할 수 있다.

2. IEEE 802.11 DCF+ 알고리즘

IEEE 802.11 DCF+는 그림 3과 같이 동작하며, 기본 알고리즘에서 전송 성공 시 CW 값을 CW_{min} 값으로 급격하게 낮추게 되면 잠재적 충돌이 발생 할 수 있으므로 CW 값을 CW_{min} 값보다 낮아지지 않는 조건으로 CW 값을 반으로 낮추는 알고리즘을 제안했다. DCF+ 알고리즘에서 CW 값의 산출은 식(3)과 같이 나타낼 수 있다.



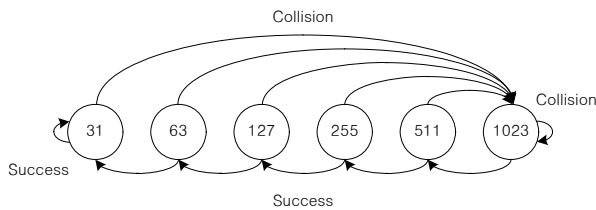
▶▶ 그림 3. IEEE 802.11 DCF+ 알고리즘의 CW상태 천이도

$$CW_n = \begin{cases} \min((CW_{n-1} \times 2) + 1, CW_{max}) & \text{if collision} \\ \max(int(CW_{n-1} \times 0.5), CW_{min}) & \text{if success} \end{cases} \quad (3)$$

3. 제안하는 알고리즘

두 가지 방법 모두 충돌이 발생했을 경우 CW값을 두 배로 늘려주게 된다. 하지만 혼잡상황에서는 CW값을 두 배로 늘려주는 것으로는 다음에 일어날 충돌을 막기 힘들다. 충돌이 일어나게 되면 충돌이 일어날 때 마다 DIFS값만큼 기다리고 또 다시 Contention Window크기(Backoff Time)만큼 기다려야 함으로 성능저하를 수반하게 된다.

이를 보완하기 위해서 본 논문에서는 그림 4와 같이 충돌 발생시 CW_{max} 값으로 높여주고 전송 성공 시 CW값을 반으로 줄여 주는 방법을 제안한다. 제안한 방법에서 CW 값의 산출은 식(4)와 같이 나타낼 수 있다.



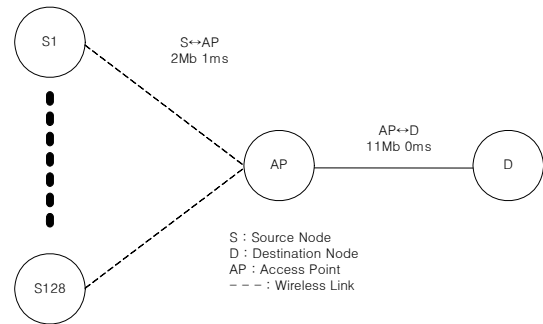
▶▶ 그림 4. 제안하는 알고리즘의 CW상태 천이도

$$CW_n = \begin{cases} CW_{max} & \text{if collision} \\ \max(int(CW_{n-1} \times 0.5), CW_{min}) & \text{if success} \end{cases} \quad (4)$$

III. 시뮬레이션을 통한 성능 분석

1. 시뮬레이션 모델

시뮬레이션 모델은 그림 5와 같이 구성하였고 소스 스테이션은 2개에서 128개까지 트래픽을 증가시켰으며, 각 소스 스테이션은 항상 전송할 패킷이 대기하고 있는 포화상태에서 시뮬레이션을 수행하였다. 각 소스 스테이션에서 발생하는 패킷은 1024 byte 크기의 일정한 패킷을 전송하며 RTS와 CTS는 사용하지 않았으며, 총 시뮬레이션 시간은 250sec이다. 또한 시뮬레이션에 사용한 소프트웨어는 Linux 환경 기반에 ns2-2.31 버전을 사용하였다.



▶▶ 그림 5. 시뮬레이션 모델

2. 시뮬레이션 파라미터

제안하는 알고리즘의 성능을 평가하기 위하여 표 1과 같은 파라미터를 사용하였고 $CW_{min} = 31$, $CW_{max} = 1023$ 로 설정하였다. 동일 패킷에 대한 최대 허용 가능한 재전송 회수는 7로 가정하였으며, 무선링크의 전송속도는 2Mbps이고 무선변조 방식은 DSSS임을 가정하였다.

[표 1] 파라미터

항목	값
Frame Payload	1024byte
Header	20byte
ACK	40byte
Channel Bit Rate	2Mbps
Slot Time	20µsec
SIFS	10µsec
DIFS	50µsec

3. 시뮬레이션 결과 분석

제안하는 방식의 효율성을 입증하기 위하여 성능 파라미터로는 포화수율, 전송지연, 충돌율, 드랍율을 이용하였다.

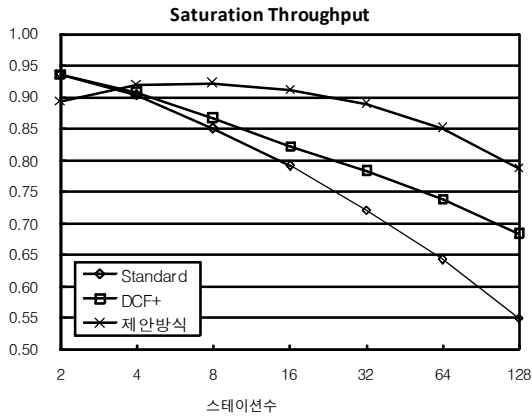
그림 6은 각각의 알고리즘에 대한 포화수율 (S : Saturation Throughput)를 나타낸다. 포화수율은 전체 전송 시간 중 순수하게 데이터를 전송하는 데 소요된 시간의 비율로 정의하며, 이를 식으로 나타내면 식(6)과 같다.

$$S = \frac{N_{TX} \times Frame\ Payload}{T_{total}} \quad (6)$$

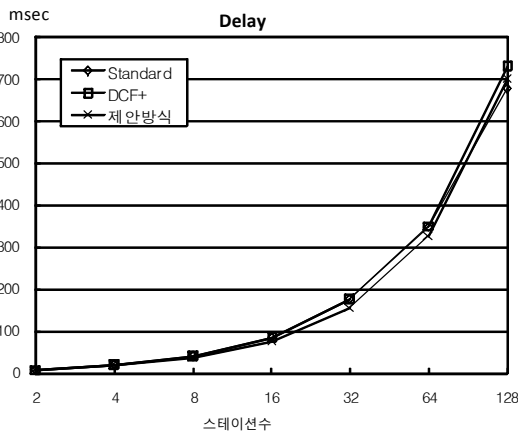
T_{total} 은 총 시뮬레이션 시간이며 N_{TX} 는 총 전송 성공회수이다. 그림 6에서 스테이션의 수가 2개일 때는 제안한 알고리즘보다 기존 알고리즘의 수율이 좋지만 스테이션의 수가 3개 이상 부터는 제안한 알고리즘의 수율이 좋아지기 시작하여 스테이션의 수가 128개가 될 때까지 성능이 좋음을 알 수 있다.

그림 7은 각각의 알고리즘에 대한 지연시간을 나타낸 것으로 트래픽이 증가할수록 지연시간이 길어짐을 확인할 수 있음

며, 각각의 알고리즘의 지연시간은 거의 동일함을 알 수 있다.

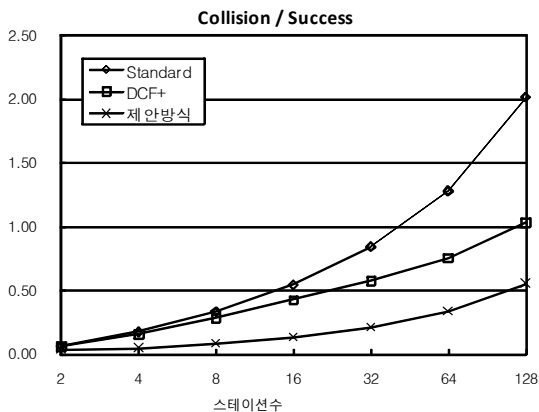


▶▶ 그림 6. 트래픽 증가에 따른 포화수율



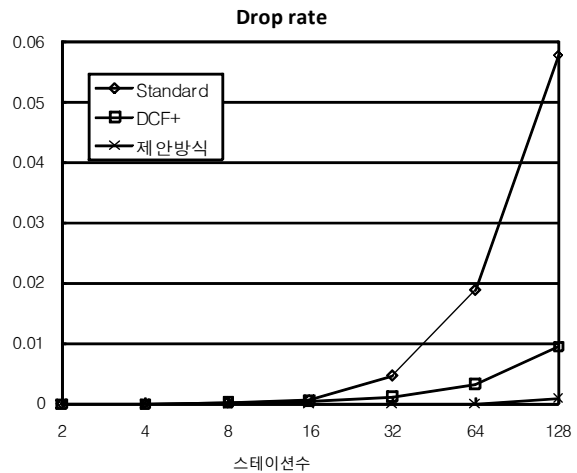
▶▶ 그림 7. 트래픽 증가에 따른 지연시간

그림 8은 트래픽의 증가에 따른 충돌율을 나타낸 것으로 충돌율은 충돌이 발생한 횟수에서 전송이 성공한 횟수로 나누어 구할 수 있고 트래픽이 증가할수록 패킷전송에 성공한 횟수에 비해 충돌이 발생한 횟수가 증가하는 것을 알 수 있다. 스테이션의 수가 128개 일 때는 기본 알고리즘보다 제안한 알고리즘이 거의 4배 가까이 성능이 우수함을 확인할 수 있다.



▶▶ 그림 8. 트래픽 증가에 따른 충돌률

DCF는 충돌이 7번 일어난 후에는 패킷을 폐기 시키는데 그림 9의 드랍율은 전송하는 패킷에서 어느 정도의 패킷이 드랍되는 지에 대해 나타낸 것으로 기본 알고리즘은 16개의 스테이션부터 드랍이 생기기 시작하며 트래픽이 증가함에 따라 드랍이 점점 많아지는 것을 확인 할 수 있고 DCF+ 또한 32개의 스테이션부터 발생하여 트래픽이 증가할수록 드랍이 증가하는 것을 알 수 있다. 하지만 제시한 알고리즘은 128개의 스테이션이 되어야 약간의 드랍이 발생하는 것으로 보아 제시한 알고리즘의 성능이 뛰어남을 확인할 수 있다.



▶▶ 그림 9. 트래픽 증가에 따른 드랍률

시뮬레이션의 결과를 토대로 분석한 결과 세 가지의 알고리즘의 지연시간은 거의 동일하지만, 본 논문에서 제안한 알고리즘은 포화수율, 충돌율, 드랍율 측면에서 성능이 우수함을 알 수 있었다.

IV. 결론

본 논문에서는 기존의 IEEE 802.11 DCF에서 성공적인 패킷 전송 후에 $CW=CW_{min}$ 으로 급격히 감소시킴으로 인해 성능이 저하되는 문제점을 해결하기 위해 스테이션간의 충돌확률을 낮게 하여 무선자원의 이용률을 높일 수 있는 새로운 알고리즘을 제안하였다.

혼잡상황에서 기본 DCF 알고리즘은 전송 성공 시 CW 값을 CW_{min} 값으로 줄여주는 것보다 CW 값을 반으로 줄여주는 것이 다음에 일어날 충돌인 잠재적 충돌이 일어날 확률을 줄여주게 되어 성능이 좋아진다는 것을 기본 DCF 알고리즘과 DCF+ 알고리즘을 비교함으로써 알 수 있었고, 충돌 발생 시 CW 값을 두 배씩 증가하는 것보다 CW_{max} 값으로 증가하는 것이 성능이 좋아진다는 것을 DCF+와 제안한 알고리즘을 비교함으로써 알 수 있었는데, 성능이 좋아지는 이유로는 혼잡상

황에서는 충돌이 많이 발생되게 되어 CW 값이 CW_{max} 값까지 증가하는 경우가 많아지게 되면서 여러 단계에 거쳐 CW_{max} 값에 도달하는 것보다 CW_{max} 값에 한 번에 도달하는 것이 DIFS와 Backoff Time으로 인한 자원의 낭비가 적으므로 제안한 알고리즘의 성능이 좋은 것으로 판단된다.

본 연구결과를 바탕으로 진행할 향후 연구과제로는 제안하는 알고리즘에 대한 수학적 분석이 수행될 예정이다. 또한 멀티미디어 트래픽 환경을 고려한 시뮬레이션 수행과 이를 위한 각 클래스별 파라미터를 예측하기 위해 시뮬레이션도 아울러 수행할 예정이다.

■ 참고 문헌 ■

- [1] The Editors of IEEE 802.11 IEEE Standard for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications, Nov. 1997.
- [2] Y. Xiao, "An Analysis for Differentiated services LANs," Proc. in ICFCS'04, 2004. pp. 32-39
- [3] Y. Peng, H. Wu, S. Cheng and K. Long, "A New Self-Adapt DCF Algorithm," Proc. in IEEE GLOBECOM '02, IEEE, Vol. 1, Nov. 2002. pp. 87-91
- [4] Y. Xiao, J. Rosdahl, "Throughput and Delay Limits of IEEE 802.11," IEEE Commun. Lett., Vol. 6, No. 8, Aug. 1996. pp. 355-357
- [5] 황안규, 이재용, 김병철 "IEEE 802.11 DCF 성능 개선을 위한 매체접근제어 알고리즘의 설계 및 성능 분석", 전자공학회 논문지 제 42권 TC편 제 10 호, pp.709-720, 2005.
- [6] The network simulator (ns-2), www.isi.edu/nsnam/ns/
- [7] H. S. Chhaya and S. Gupta, "Performance modeling of asynchronous data transfer methods of IEEE 802.11 MAC protocol", Wireless Networks, vol. 3 (1997), pp. 217 - 234, 1997.
- [8] T. S. Ho and K. C. Chen, "Performance evaluation and enhancement of the CSMA/CA MAC protocol for 802.11 wireless LAN's," in Proc. IEEE PIMRC, Taipei, Taiwan, pp. 392-296
- [9] F. Cali, M. Conti, and E. Gregori, "IEEE 802.11 wireless LAN: Capacity analysis and protocol enhancement", presented at the INFOCOM'98, San Francisco, CA, Mar. 1998.
- [10] http://aqua.comptek.ru/test/HiddenNode/hidden_node_en.html