채널상황에 동적 대응하는 MAC의 구조에 대한 연구

김숙영, 김영성, 석정봉
연세대학교
iwilldo@empal.com, dolfeen@empal.com, jbsuk@dragon.yonsei.ac.kr

A Study of MAC Architecture Dynamic cope with channel status

Sookyoung Kim, YoungSung Kim, Jungbong Suk
Yonsei University

요 약

본 논문에서는 802.11e MAC의 EDCA 모드를 기반으로 한 무선랜 환경에서 realtime data인 voice traffic의 QoS를 향상하기 위한 방안을 연구하였다. 동적으로 채널의 상태를 보고, 네트워크 상태를 예측하여 채널 경쟁에 참여할지를 결정하는 방안으로, 경쟁이 치열할 때 voice traffic에 차별화된 가중치를 더 줄 수 있어 전반적인 voice traffic의 성능 향상에 기여한다.

본 연구에서 SU(slot utilization)을 계산하여 PT(Probability of Transmission)을 구할 때, 기존의 DCC 알고리즘을 그대로 802.11e에 적용하게 되면 4개의 AC에 똑같은 알고리즘이 적용되어 802.11e의 핵심인 차별화된 QoS를 지원하는데 무리가 있다. 기존 DCC 알고리즘에서 재시도 회수만 고려해 구하는 것을 802.11e 구조에 맞추어 4개의 AC별로 차별화하여 전송확률(PT)를 구할 수 있게 하였다. 뿐만 아니라, 재시도 회수를 고려하여 재시도 회수가 많은 packet에는 PT값이 높게 나올 수 있게 하고, 최대 가능한 재시도 회수에 도달할 때에는 상위 AC 영역의 값이 나오게 하여 AC의 upgrade가 되는 효과를 가져오게 하였다. 이 때 각 AC의 lower bound와 upper bound를 정하고, 이 때 802.11e의 최대 재시도 회수 parameter와의 상관관계를 정의하여 하위 AC가 상위 AC의 영역을 침범하지 못하도록 정의하고 있다. 추가적으로 SU의 값을 구할 때에도 현재의 SU값 대신 누적된 평균 SU값을 사용해 PT값을 구하여, utilization, latency, packet loss등 전반적인 부분의 성능개선을 확인할 수 있다.

1. 서 론

802.11 무선 랜은 data를 best-effort이라는 획일적인 방법으로 통신하는 메커니즘을 사용하고 있어 VoIP와 같은 realtime data를 서비스 하는 데에는 QoS를 지원에 무리가 있다. 하나의 queue에 realtime traffic과 non-realtime traffic을 넣어서 스케쥴링 하기에는 구조적 제한점이 많아 service의 차별화(differentiation)를 지원하기에 역부족이다. 이에 IEEE는 traffic을 여러 가지 AC(Access Category)로 나누고, 각 AC별로 queue를 두어서 차별화된 서비스를 가능하게 하기 위한 QoS frame을 마련하고 backward compatibility도 지원하는 802.11e 표준안 개발을 진행하고 있다. 802.11e는 MAC 계층에서 QoS를 지원하기 위한 QoS 프레임을 제공한다고 볼 수 있다.

IEEE 802.11e D8.0에 voice와 video를 위한 각 AC별 contention window의 최소값과 최대값의 간격이 줄어들어 backoff window의 adaptive하게 조절하는 제안된 방안[1][3]의 효력이 거의 없어졌다[4]. 최근에는 802.11기반인 동적인 네트워크의 상황을 반영하기 위해 기존 802.11에 제안되었던 DCC 알고리즘을 802.11e에 적용하는 것이 시도되고 있다[4]. 이에 802.11의 DCC 알고리즘을 기반으로 하여 802.11e의 QoS 프레임을 지원하면서도 voice와 같은 realtime data 서비스를 위해 더욱 개선된 성능을 가진 방안에 대한 연구가 진행되고 있다.

본 논문의 내용은 관련연구 및 차별화된 QoS 지원을 위해 802.11e를 기반으로 VoIP QoS 향상을 개선하기 위한 새로운 방안들의 제시와 모의실험을 통한 성능평가와 결론으로 구성된다.

2. 관련 연구

2.1 AEDCF[1]

이 논문에서는 IEEE 802.11e의 기본적인 EDCF 모드를 기반으로 성공적인 전송이나 collision 발생 후 contention window의 크기를 적응적(adaptive)으로 조절하여 throughput, delay, packet loss 등의 성능 평가에서 EDCF보다 나은 성능을 보여주고 있다.

기본 EDCF scheme에서, CWmin[i]와 CWmax[i]의 값은 각 priority level에 맞게 정의되어 있다. 매 성공적인 전송 이후에 CW[i]의 값은 획일적으로 CWmin[i]의 값으로 설정되고, collision이 발생할 때 contention window의 값 CW[i]를 현재 CW[i] 값의 2배로 증가시키는 방법을 사용하고 있다.

2.2 DCC(Distributed Contention Control)[2]

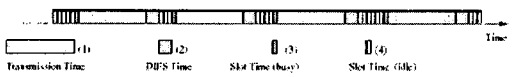
실제 채널 혼잡 수준에 대한 초기값과 의미 있는 정보를 개발함으로 CSMA/CA access scheme의 성능향상을 위한 가능성에 대한 연구로 slot utilization(SU)을 측정하여 채널의 혼잡 수준의 측정값을 구한다. SU는 다음 2가지 조건을 필요로 한다.

- SU은 0과 1사이임. 0은 slot overhead가 없음을 의미.(busy하지 않음). 반면에 1은 전송을 위해 가용한 매 slot이 busy한 경우임.
- SU 값은 확률적으로 네트워크의 경쟁 수준에 대하여 내부적으로 [0..1]사이의 분산된 값이어야 함.

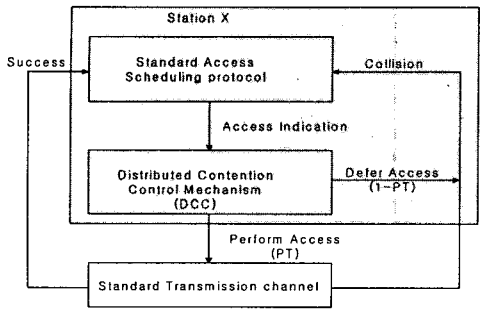
자체의 backoff interval동안, 매 스테이션은 채널상에 발생하는 전송 시도의 수(NumBusySlots)를 세고, 채널상에서 관찰되는 전송을 위한 가용 슬랏의 전체수를 나눈다(NumAvailableSlots). 따라서, SU의 정의는 다음과 같다.

$$SlotUtilization(SU) = \frac{NumBusySlots}{NumAvailableSlots}$$

DCC에 따라 각 스테이션은 채널의 경쟁 수준에 의존하는 PT의 값에 따라 전송시도를 필터 해야 한다. 특히, SU 평가 이후에 각 스테이션은 PT 값을 계산하여 공유 채널에 접근하기 위한 기회를 평가하는데 적용한다. 스테이션이 전송을 지연하기로 결정하면, 새로운 시도를 재스케쥴하고, 충돌이 발생할 때에도 같은 방법으로 재스케쥴한다. 따라서, 이 방법은 경쟁 감소를 통계에 기반으로 현실화하고, 표준 프로토콜



[그림 1] 각 station에서 관찰된 channel state의 예



[그림 2] DCC의 구조적 배치

에 따라 충돌 확률을 줄인다.

PT를 평가하기 위한 제안된 메커니즘에 적용된 식을 분석해보면, 혼잡한 시스템을 제어하기 위하여 access의 수(즉, collision의 확률)를 줄이는 것이 적당하다고 결론짓고 있다. 이는 SU의 평가에 의한 PT의 정의로 쉽게 구할 수 있다.

$$PT(SU) = 1 - SU$$

정의에 따라, SU은 [0..1] 사이의 값으로 가정한다. SU은 채널 경쟁 수준을 표현하고, 그 값은 DCC 방법에 적용된다.

$$PT(SU, NumAtt) = 1 - SU^{NumAtt}$$

SU 에 따른 전송확률(PT)은 전송시도회수(NumAtt)가 증가함에 따라 높아지는 것을 확인할 수 있다.

DCC 알고리즘을 정리하면 backoff_counter가 0이면(전송할 때가 되면), SU을 계산하고, 이에 따른 PT 값을 구한다. 이 때 구한 PT값이 임의의 값보다 작을 때에는 전송을 지연하고, 임의의 값보다 클 때에는 전송을 한다.

2.3 DCC_ICU[4]

본 논문에서는 802.11에 제시되었던 SU를 통한 DCC 방법을 802.11e EDCA에 적용하였다. DCC를 그대로 EDCA 에 적용할 경우 모든 AC에 같은 알고리즘이 적용된다. Contention이 증가함에 따라 DCC가 커버할 수 있는 범위를 넘어서 voice나 video가 delay bound를 만족시키지 못하고 drop하는 비율이 증가한다. 이를 해결하기 위해 contention이 증가함에 따라 전송확률을 떨어뜨리는 파라미터 collision probability를 추가한다.

$$f^{j}_{avg} = (1-\alpha)\cdot f^{j}_{curr} + \alpha \cdot f^{j-1}_{avg}$$

f^{j}_{curr}는 현재의 collision rate이며, α는 smoothing factor이다. Period j는 5000 time-slots 마다 update되며, α는 0.8이다.

$$SU[i] = \frac{NumBusySlots[i]}{InitBackoff[i]+1}$$

$$PT[i] = 1 - min\left(1, \frac{SU[i]}{1.2 - (N_{AC[i]}+1)\cdot f_{avg}}\right)^{NumAtt[i]}$$

$$where\, N = i \quad for\ i \in \{0,1,2,3\}$$

3. 제안방안

3.1 NEW1

이 방안은 SU와 AC의 값을 고려하여 PT를 구하는 방안이다. 기존의 DCC는 NumAtt를 고려하였으나, 이는 경쟁이 많아 재시도 횟수가 증가할 때에만 영향을 미치게 된다. NEW1은 AC의 값을 재시도 횟수에 상관없이 적용시켜, 경쟁이 치열하지 않을 때에도 AC에 의한 가중치를 전송확률 PT에 적용하자는 것이다. 이 때 AC의 값은 voice를 위한 AC0이 최대 값을 가지고, background traffic인 AC3을 최소 값을 가지게 지정하였으며 이 값이 제안방안의 AC값으로 사용되었다.

$$PT(SU, AC) = 1 - SU^{AC},\ AC \in \{1,2,3,4\}$$

3.2 NEW2

DCC[2]에서 SU을 구해 PT를 구할 때 NumAtt를 사용하였다. NumAtt는 전송 시도 회수를 나타내는데, collision이 많이 발생하지 않는 채널 상황에서는 전송 시도 회수는 거의 1 값을 가지며, 다른 영향을 주지 않는다.

NEW1의 AC만 고려한 것이 아니라, NumAtt도 고려한 PT를 제안한다. 802.11e MAC의 parameter에는 Retry Count 와 관련된 parameter가 2가지가 있다. ShortRetryLimit와 LongRetryLimit가 그것인데, 둘 다 collision 발생시 재 전송 시도회수가 증가하는데, 이 때 이 parameter의 upper bound를 정의하여 그 시도 회수를 제한하고 있다. LongRetryLimit은 packet의 size가 RTSThreshold 보다 클 때 사용하는 값이며 본 모의 실험 환경에서는 다음과 같이 정의되어 있다.

- *#define ShortRetryLimit 7*
- *#define LongRetryLimit 4*
- *#define RTSThreshold 3000*

이 때 RetryLimit(RL)은 packet의 size가 RTSThreshold보다 크면 LongRetryLimit을 가져오고, 작으면 ShortRetryLimit을 가져오게 한다. 재시도 회수가 RetryLimit의 최대값에 도달했을 때에는 packet은 MAC_RETRY_COUNT_EXCEEDED이라는 사유로 drop된다.

$$PT(SU, AC, i) = 1 - SU^{AC*(1+(i-1)/(j-2))}$$

$$where\, i = NumAtt,\ j = RetryLimit,\ AC0\,(Voice)$$

$$PT(SU, AC, i) = 1 - SU^{AC*(1+(i-1)/((j-2)*AC)}$$

$$where\, i = NumAtt,\ j = RetryLimit\, AC1/2/3$$

3.3 NEW3

이 방안은 DCC_ICU[4]에서 제안한 방안을 기본적으로 사용하면서, 재시도 회수(NumAtt)에 의한 거듭제곱을 NEW2에서 제안한 방안의 거듭제곱으로 바꾼 것이다. [4]는 충돌비율(collision rate)를 5000 slot time마다 측정하여 이전에 측정된 평균 충돌 비율과 현재 측정된 충돌 비율을 smoothing factor α를 0.8의 값으로 정하였다.

$$PT[i] = 1 - min\left(1, \frac{SU[i]}{1.2 - (N_{AC[i]}+1)\cdot f_{avg}}\right)^{AC*(1+(NA-1)/(RL-2))}$$

$$where\, N = i \ \text{ for } i \in \{0\},\ NA \in \{1,2,3,4\ldots\}$$

$$PT[i] = 1 - min\left(1, \frac{SU[i]}{1.2 - (N_{AC[i]}+1)\cdot f_{avg}}\right)^{AC*(1+(NA-1)/((RL-2)*AC))}$$

$$where\, N = i \ \text{ for } i \in \{1,2,3\},\ NA \in \{1,2,3,4\ldots\}$$

3.4 NEW4

NEW4는 SU 값을 현재 측정된 SU 값과 지난번까지의 평균 SU 값에 smoothing factor α를 적용해 현재의 평균 SU의 값을 구하는 방안으로 매 backoff time마다 구한다.

$$SU^{j}_{avg} = (1-\alpha)\cdot SU^{j}_{curr} + \alpha \cdot SU^{j-1}_{avg}$$

$$PT[i] = 1 - min\left(1, \frac{SU_{avg}[i]}{1.2 - (N_{AC[i]}+1)\cdot f_{avg}}\right)^{AC*(1+(NA-1)/((RL-2)*AC))}$$

$$where\ N = i \quad for\ i \in \{0,1,2,3\}$$

SU^{j}_{avg}는 현재(j번째)의 평균 SU 값이며, α는 실험결과에 의해 0.9일 때 성능이 제일 좋았다. SU^{j}_{curr}은 현재 측정된 SU의 값이며, SU^{j-1}_{avg}은 j-1번째까지의 평균 SU 값이다. 이 때 SU의 값은 [4]의 제안대로 초기 backoff 값 들 중의 busy slot의 개수이며, 이 때 초기 backoff 값이 0이 나올 수 있으므로 1을 더해주고, AC값은 voice의 AC0가 4의 값을 갖고, background AC3이 1의 값을 갖는다.

3.5 NEW5

NEW5는 기존의 DCC에 NEW4에 제안한 평균 SU 값을 사용하는 간단한 방법이다. 즉, 평균 SU의 값에 NumAtt만큼 거듭제곱 한 값을 1에서 빼 PT 값을 구하는 방법이다. DCC의 방안보다는 누적된 SU값을 사용하여 전반적인 channel utilization에 따른 PT 값을 구할 수 있다.

$$PT[i] = 1 - SU_{avg}[i]^{NA}$$
$$where\, NA \in \{1, 2, 3, 4 \ldots\}$$

3.5 NEW6

NEW6는 NEW4에 제안한 평균 SU 값을 사용하고, 거듭제곱 값으로 NEW2에 제안된 방안을 채택하였다. 다음과 같은 식을 사용하여 PT 값을 구한다.

$$PT[i] = 1 - SU_{avg}[i]^{AC*(1+(NA-1)/((RL-2)*AC))}$$
$$where\, NA \in \{1, 2, 3, 4 \ldots\}$$

4. 성능평가 및 결론

4.1. 모의 실험 환경

이 논문을 위한 실험 환경으로는 Linux Redhat 9.0을 기반으로 NS-2 simulator 2.26 version[6]을 사용하였으며, 802.11e module로는 TKN_Berlin IEEE 802_11e EDCF module[5]을 사용하였다.

4.2. 모의 실험 시나리오

n개의 노드를 만들고 node i에서 node i+1으로 voice, video, background 세 가지 유형의 traffic을 송신하고, node i+1은 node i에서 송신한 세 가지 유형의 traffic을 수신한다[1].

Physical layer는 802.11b를 사용하였으며, 해당하는 parameter와 4개의 AC를 위한 MAC parameter는 [5]을 참조하여 정의하였다. 4개의 AC를 위한 Parameter로 CW_{min}, CW_{max}, AIFS, PF, TxOPLimit을 사용하고 있다. AC0(voice)을 위해 contention window의 값을 3과 7을 정의하고, AC1(video)을 위해 7과 15의 값을 사용하고 있다. 이는 최근 정의하고 있는 802.11e D8.0[4]을 반영한 결과로 contention window의 range가 줄어들어 AEDCF[1]과 같이 contention window 조절하여 채널의 idle slot을 줄여 성능개선을 하는 알고리즘은 거의 영향력이 없어짐을 볼 수 있다.[4]

Traffic model로 AEDCF[1]에 사용한 voice, video, background traffic을 위한 3가지를 정의하였다. AC0(voice)는 160bytes의 packet size로 20ms interval로 보내는 것은 64Kbps 속도인 PCM codec을 사용하는 것으로 생각할 수 있다. AC1(video)은 1280bytes의 packet size를 10ms interval마다 보내는 것으로 하였는데, 이는 1024Kbps의 전송속도를 가지며, MPEG4 medium quality를 내는 VOD 서비스로 간주할 수 있다. background traffic(bt)은 AC2로 정의하고, 200bytes의 packet size로 12.5ms interval로 보낸다.

NEW2와 NEW3에서 사용되는 ShortRetryLimit은 7의 값으로, LongRetryLimit는 4, RTSThreshold의 값은 3000으로 정의되어 있으며, 평균 SU를 위한 alpha 값으로는 0.9를 사용하였다.

4.3 모의 실험 결과

Station의 수가 늘어남에 따라 channel utilization도 증가하는데, station의 수가 5와 10일 때, new4와 new5, new6는 dcc에 비해 10% 이상의 성능 개선 효과가 나타난다.

Goodput[1]은 station의 수가 증가함에 따라 전체적으로 감소하는데, station의 수가 5일 때 new4의 경우 dcc보다 10.7%의 성능 개선 효과가 나타난다.

Station의 수가 증가함에 따라 latency는 증가하는데, station의 수가 5일 경우, new4가 dcc에 비해 7% 성능 개선 효과가 나타나고 station의 수가 5와 10일 때, new4와 new6는 dcc에 비해 평균 30% 정도의 packet loss를 줄이는 효과가 나타났다.

4.4. 결론

모의실험 결과, station의 수가 늘어남에 따라 경쟁이 치열해져 collision이 증가되고, 이에 따른 goodput이 감소하며, delay와 packet loss가 증가하는 것을 확인할 수 있다. Channel utilization의 결과는 station의 수가 5와 10일 때 new4와 new5는 dcc에 비해 10%이상의 성능개선 효과가 나타난다. 결과적으로 dcc_icu에서 제안한 방안에 누적된 평균 SU의 값(SU_{avg})을 사용하는 *new4*의 제안방안이 성능 개선 효과가 큰 것을 확인할 수 있다. 거듭제곱으로 AC와 RetryLimit을 고려한 제안방안의 결과로 packet loss에 대한 성능이 크게 개선된 것을 확인할 수 있다. 이는 재시도 회수가 최고 값에 이르면, 재시도 회수가 제한되어 drop되는데, 이 때 drop되기 전에 AC의 상승효과를 일으켜 높은 PT 값으로 전송하게 하여 drop되는 것을 미리 막아주기 때문이다.

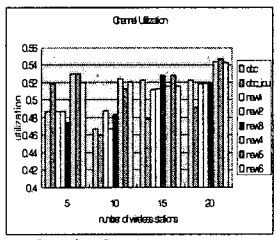
제안된 방안 모두 VoIP QoS 요구사항[3]의 delay나 packet loss에서 good quality 이상의 수준을 제공하고 있어, VoIP를 위한 intelligent한 MAC으로서의 기능을 제공한다고 볼 수 있다. 그러나 모의실험 결과 제안 방안이 station의 수와 관련하여 일관적으로 개선된 성능을 보여준다고 말할 수는 없으므로 추후 복잡도나 구현 가능성을 고려하여 연구할 필요가 있다.

[표1] 802.11b PHY/MAC

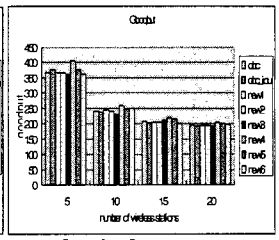
Parameters	Value
Slot Time	20μs
CCA Time	15μs
RxTxTurnaround	5μs
SIFS Time	10μs
Preamble Length	144bits
PLCP Hdr. Leng.	48bits
PLCP Data Rate	1.0e6

[표2] MAC 파라미터

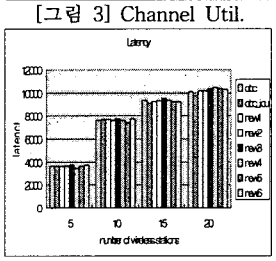
Param.	AC0	AC1	AC2	AC3
CW_{min}	3	7	15	31
CW_{max}	7	15	1023	1023
AIFS (μs)	34	34	43	43
PF	2	2	2	2
TxOP Limit	0.003008	0.006016	0	0



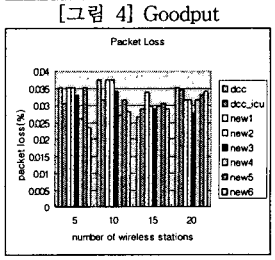
[그림 3] Channel Util.



[그림 4] Goodput



[그림 5] Latency



[그림 6] Packet Loss

5. 참고문헌

[1] Lamia. Romdhani, Qiang Ni, and Thierry Turletti, "Adaptive EDCF:Enhanced Service Differentiation for IEEE 802.11 Wireless Ad-Hoc Networks", IEEE WCNC 2003

[2] L. Bononi, M. Conti, E. Gregori "Design and Performance Evaluation of a Distributed Contention Control(DCC) Mechanism for IEEE 802.11 Wireless local Area Networks", J.parallel and distributed computing, 2000

[3] C. Casetti and C.-F. Chiasserini, "IMPROVING FAIRNESS AND THROUGHPUT FOR VOICE TRAFFIC IN 802.11E EDCA"

[4] Hyunmin Kyung, Snagho Seo and Sin-Chong Park, "QoS(Quality-of-Service) Improvement in IEEE 802.11e Enhanced Distributed Channel Access(EDCA)", KICS, 2004

[5] Technical Report TKN-03-019, Sven Wiethlter, Christian Hoene, "Design and Verification of an IEEE 802.11e EDCF Simulation Model in ns-2.26", Telecommunication Networks Group, Technische Universitt Berlin, November 2003

[6] "The Network Simulator ns-2", http://www.isi.edu/nsnam/ns/