

IEEE 802.11 무선 랜 환경에서 Idle time 기반의 링크 적응기술에 관한 연구

이유성, 김효곤, 최규영, 윤종원, 강인혜
고려대학교 컴퓨터정보통신대학원 컴퓨터공학과
e-mail : drakong@korea.ac.kr

An idle time-based link adaptation algorithm for IEEE 802.11 WLANs

You-Sung Lee, Hyogon Kim, Kyu-Young Choi, Jongwon Yoon, Inhye Kang
Dept. of Computer Science & Engineering, Korea University
Graduate School of Computer & Information Technology

요 약

본 논문에서는 관찰된 채널의 유휴 시간을 이용하여, ACK(Acknowledgement) 기반의 링크 적응 기술의 단점을 극복하는 새로운 알고리즘인 COLA 를 제안한다. COLA 는 광범위한 채널과 부하 상황에서 최적의 성능을 얻으며 RTS/CTS 메커니즘이나 하드웨어를 이용한 채널 SNR 측정과 같은 추가적인 옵션 혹은 별도의 프로토콜 메커니즘의 지원이 필요하지 않다. 또한 COLA 알고리즘은 단지 짧은 10 개의 critical 한 명령어를 사용함으로써 부하를 줄이며, 실제 동작 시 수반되는 heuristic 한 매개변수로부터 자유롭고, 따라서 실제 사용 시 장점을 가진다.

1. 서론

IEEE 802.11 무선 LAN (WLAN)에서 링크 적응 기법 [1] 은 SNR (Signal-to-noise ratio) 또는 신호 강도 측정 기반의 기술(signal strength monitoring-based scheme) [2]-[5]과 ACK (acknowledgement) 기반의 방법 [6]-[9]으로 분류된다. 전자는 후자에 비해 정밀하지만 별도의 하드웨어가 필요하거나 802.11 표준의 변경이 필요하다 [6]. ARF (Automatic Rate Fallback)와 같은 ACK 기반의 방법은 구현 비용이 저렴하여 상용 장비에 광범위하게 사용되지만, 채널에 의한 에러와 충돌의 구분이 어려운 단점이 있다 [6], [10]. 이 방법은 몇몇 연속적인 802.11 ACKs 수신 실패를 불안정한 채널 상태의 신호로 해석하여 링크 상에 충돌이 발생할 때 비정상적으로 동작될 수 있다. 이러한 판단 오류의 의한 수행 비용은 성능을 현저하게 저하시킨다 [6], [10].

본 논문에서는, 이러한 에러 구분 문제로부터 영향

을 받지 않는 채널 유휴 시간 측정에 근거한 새로운 링크 최적화 알고리즘인 COLA (Congestion-resilient Link Adaptation)을 제안한다. ACK 기반 기법에서의 처리량 감소 문제는 최근에야 제기되고 있으며, [6], [9], [10], 이러한 문제에 대한 연구는 RTS/CTS 기반의 방법을 제외하고 거의 논의되지 않고있다 [9]. RTS/CTS와 관련된 오버헤드는 802.11 링크 속도가 증가함에 따라 현저히 늘어나므로 우리의 제안은 RTS/CTS 기반의 방법에 의존하지 않는다 [11]. 또한 COLA는 SNR 이나 신호 강도 측정 하드웨어와도 관련이 없다. 그럼에도 불구하고 본 고를 통해 제안하는 주요한 10개 명령어 알고리즘은 광범위한 SNR 값에 걸쳐 최적의 성능을 발휘하고, 실제 운용 시 설정이 필요한 heuristic한 매개변수로부터 자유롭다. 본 논문에서는 802.11a 의 예 [12]를 사용했지만, 이 제안은 기타 802.11 네트워크에서 추가되는 overhead 없이 직접 적용이 가능하다.

2. COLA (Congestion-resilient Link Adaptation)

COLA의 전체적인 아이디어를 간단하게 표현하면, 실패 신호의 충돌 일부를 무시하는 것이다.

우선, Bianchi의 정리에 따르면,

$$n=1+\frac{\log(1-P_c)}{\log\left[1-\frac{2(1-2P_c)}{(1-2P_c)(W+1)+P_cW\sum_{k=0}^L(1-(2P_c))^k}\right]} \quad (1)$$

W 는 최소 경쟁 윈도우 사이즈 CW_{\min} 이며, n 은 경쟁하는 station의 수, L 은 backoff 재시도 횟수 최대값(본 논문에서는 4)이다. Station이 슬롯 안에서 전송을 시도할 가능성을 τ 라 한다면, 슬롯이 유향 상태일 가능성은 $P_i=(1-\tau)^n$ 다. t 는 슬롯 안에서 전송 시도 가능성으로 정의한다. 채널 유향 시간의 평균이 $t_i=(1-P_i)^{-1}-1$ 로 주어진다면, 우리는 다음과 같은 관계를 얻을 수 있다.

$$t_i=[1-(1-P_c)^{\frac{n}{n-1}}]^{-1}-1 \quad (2)$$

(1), (2) 수식을 정리하여 n 을 소거하면, t_i 의 함수인 P_c 값을 얻을 수 있다. 호스트가 전송을 시도하거나 다른 station의 전송을 감지하는 순간, 최종 경쟁이 시작되면서 경과된 시간을 고려하여 station을 요청함에 의해 전송 신호의 부재를 통해서 t_i 값을 파악할 수 있다 [14]. t_i 에 부응하는 P_c 값을 얻은 후, 다음과 같은 최적의 모드를 확실하게 얻을 수 있다. 각각의 전송 시도에 대해 다음과 같이 전송 실패를 N_f 전송 성공을 N_s 로 정의하고 값을 갱신한다.

$$\begin{aligned} \text{failure} &: N_f \leftarrow N_f + P_h; \quad \underline{N_s \leftarrow N_s + P_c.} \\ \text{success} &: N_s \leftarrow N_s + 1. \end{aligned} \quad (3)$$

편의상 P_c 값과 채널 에러 가능성 P_h 값이 공통의 원소를 갖지 않는다고 가정한다. 즉, $P_f \approx P_c + P_h$. 현재 모드 m 에서 전송시도 값 N_t 가 주어진다면, 채널에 의한 에러의 평균을 $\bar{H}=N_t \cdot P_h=N_f-N_t \cdot P_c$ 와 같이 추정할 수 있다. 현재 모드의 H 나 N_s 값이 mode-down과 mode-up 경계를 초과할 때, 각각 모드를 변경할 수 있다.

COLA의 첫번째 독창성은 (3): $\underline{N_s \leftarrow N_s + P_c}$ 수식에서 N_s 의 값이 전송 실패에서도 평가되는 것이다. 이

는 링크 최적화의 관점에서, 실패한 충돌 값 P_c 가 아직 “성공”인 경우이며, 이것은 채널의 영향이 아니기 때문이다. 이 부분에서의 개념은 채널에 의해 발생하는 값만이 링크 적응기술 알고리즘이 고려되어야만 하는 것이며 오직 위 값만이 실패 카운트에 반영되는 것이다. 본 논문의 두 번째 독창성은 mode-up/mode-down 경계의 선택에 관련한다. 모드 m (예를 들어, 802.11a에서는 모드 1은 6Mbps, 모드 8은 54Mbps)이 제공하는 비트 전송률을 r_m 라 하고, $P_h^{(m)}$ 값은 모드 m 하에서 현재 SNR에 의해 주어진 채널 에러에 의한 FER로 계산된다. 근접한 모드들 (예를 들어, 802.11a에서는 12Mbps의 크기) 사이에서 광범위한 차이에 기인하여, 모드 m 에서의 우발적 손실을 감수하는 것이 모드 $m-1$ 에서 에러 없이 전송하는 것보다 유용할 수 있다. 다시 말해서, 하위 모드로 변경하는 것보다 모드 m 에서 머무르는 것이 효과적이다.

```

N_t = N_t + 1
if (transmission_failure)
  if (up_test) call test_processing
  else /* not testing */
    N_f = N_f + 1
    N_s = N_s + P_c /* collisions are counted as "successes" */
    H = N_f - N_t * P_c /* how many channel-induced errors? */
    if (1 - H/N_t < r_{m-1}/r_m) /* go down */
      if (m != 1) m = m - 1 /* drop mode */
      N_s = 0; N_t = N_f = 0; u_m = 1 /* reset variables */
else /* success */
  if (up_test) {s_t = s_t + 1; call test_processing}
  else /* not testing */
    N_s = N_s + 1
    if (N_s >= u_m and m < 8)
      /* enough successes at current mode: start tests */
      a_t = s_t = 0; up_test = 1;
      m = m + 1 /* temporarily move to the tested mode */
    else if (m != 1)
      u_{m-1} = 1 /* backoff unnecessary at m - 1 */

```

```

test_processing:
  a_t = a_t + 1
  if (a_t >= T) /* tests completed */
    if (m != 1) m = m - 1 /* restore mode */
    if (s_t / a_t <= r_m / r_{m+1}) /* cannot go up */
      u_m = 2 * u_m; /* next time be more conservative */
      N_s = N_f = N_t = 0 /* reset variables */
    else /* high enough success rate, go up */
      u_{m-1} = 1; m = m + 1; u_m = 1; N_s = N_f = N_t = 1
      up_test = 0 /* out of the test */
return

```

그림 1. COLA 알고리즘

$$(1-p_h^{(m+1)}) \cdot r_m > r_{m-1}, p_h^{(m)} = \bar{H}^{(m)} / N_t^{(m)} \quad (4)$$

마찬가지로, 모드 증가에 관해서 아래와 같은 조건을 시행한다.

$$(1-p_h^{(m+1)}) \cdot r_{m+1} > r_m. \quad (5)$$

mode-up 상황에서의 요청은 m 모드에 머무르는 동안 $P_h^{(m+1)}$ 의 값을 측정하지만, $m+1$ 의 충분한 통계를 수집하기 전까지 모드의 증가는 발생하지 않는다. m 모드에 머무르는 동안, $m+1$ 의 $a_i \geq T$ 테스트를 수행하며, 성공한 횟수 s_i 를 기록한다. 이에 따라 $P_h^{(m+1)} = 1 - s_i / a_i$ 를 계산할 수 있고, (5) 수식을 검사한다. T 매개변수에 값을 설정하여 s_i / a_i 는 r_m / r_{m+1} 값과 유사한 결과를 갖는다. 본 논문에서는 4로 설정한다.

COLA 알고리즘은 그림1.과 같이 구성된다. Mode-up 테스트를 마치면, (5)식과 같은 결론을 얻을 수 있다. 만약 식을 만족하지 못한다면, 다음 $m+1$ 으로의 mode-up 시도가 신중하게 이루어진다. 이러한 상황에서 우리는 급격한 backoff 개념을 사용하여, [6]의 논문에서와 같이 요구되는 성공횟수 u_m 를 배가하여 mode-up 테스트에 적합하도록 만들 수 있다. 만약 수식 (5)를 만족하면, $m+1$ 모드로 이동한다. 이러한 경우, 모드 m 에서 최소 한번 이상의 성공 이후로 u_{m-1} 값은 1로 리셋된다. Mode-up 테스트를 실행하지 않는 한은 각각의 실패에 대해 수식 (4)를 테스트한다. 결과를 만족하면, m 모드에 머무르지만, 그렇지 않다면 $m-1$ 모드로 이동한다. 이 알고리즘은 각각의 전송시도에 대해 수행하기 복잡해 보이지만, 모드가 하향 되기 전 전송실패에 대해 즉각 수행하는 10개 항목의 critical 경로의 명령문을 갖고있다. 이외의 사항은 이보다 더 단순하며, 최악의 수렴시간은 대략 $7T$ 로 볼 수 있다. 이 알고리즘은 m 의 값이 1부터 시작할 수 있도록 설정이 가능하며, 최적 모드가 8이라면 T 의 값은 4가 되며 28번의 전송시도를 갖는다.

3. 성능 검증

이 장에서는 n 개의 스테이션을 갖는 802.11a BSS 무선 랜 환경에서 모든 상호 수신 범위에서 링크 최적화 알고리즘을 실험한다. 이 n 개의 노드들은 암호화된 UDP/IP 하에서 최대 전송속도로 CBR 트래픽을 전송한다. Shadowing 모델인 ns-2 [15] 시뮬레이터를 사용하여 802.11a PHY를 지원하도록 802.11 모듈을 확장하고, 다른 모드에서 신호를 받아들이는 감도를 변화하는 모델링 작업을 수행한다. 또한 우리는 Heiskala 와 Terry [16]가 제공하는 FER을 평가하는 실험적인 BER vs. SNR 곡선을 사용한다.

그림 2.는 채널 SNR의 함수로서 COLA의 성능 효과를 보여준다. 또한 비교를 위해 ARF의 최대 성능과 비교하였다. ARF 항목은, 모드 증가에 따른 연속된 성공 횟수에 따른 두 가지 변수를 사용하였다. 하나는 3, 다른 하나는 10으로 설정하였고, 전자를 "ARF 3", 후자를 "ARF 10"로 명명한다. 모드 감소 시에는 연속되는 두 실패 건을 두 변수에 따라 구

정한다. 참고로, "OPT"의 성능을 먼저 실험한 최상의 결과를 보이는 모드로 가정한다. 모드 1부터 8까지의 지정된 값을 통해 결과를 얻어, 그 중 가장 좋은 결과를 보이는 SNR 값을 선택하였다. 주지할 것은 OPT는 충돌과 채널에 의한 에러를 구분하지 않는다.

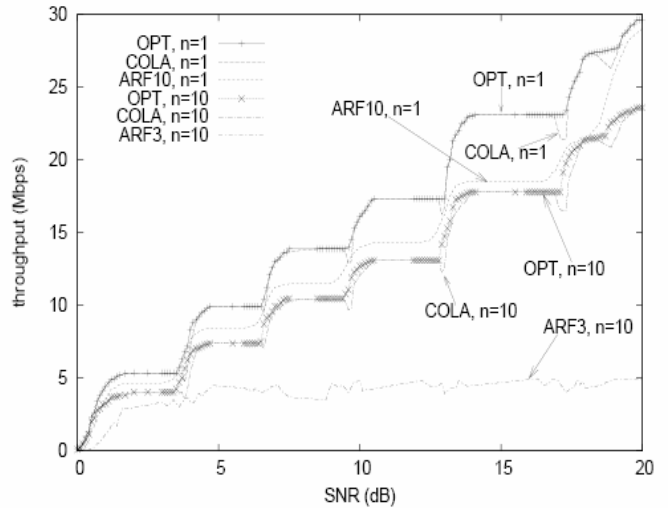


그림 2. COLA vs. ARF. Throughput 성능 비교

예상과 같이, ARF의 결과치는 충돌에 심각한 영향을 받는다. N 값이 10일 경우, ARF의 최상 결과치¹는 채널의 상태에 상관없이 5Mbps 대에 고정됨을 볼 수 있지만, COLA의 경우는 OPT에 근접한다. 모드 변경 구간을 제외하고, 결과값 곡선은 사실상 각각 최상에 위치한다. COLA와 ARF의 결과값 차이는 채널이 상충함에 따라 충돌이 발생하고 전송 실패가 일어나는 경우 더욱 명확하게 나타난다. 예를 들어, 20dB 구간에서 n 값이 10일 때, 그 차이는 20M bps에 가깝다.

4. 결론

본 논문에서는 IEEE 802.11의 ACK기반의 링크 적응 기술 알고리즘인 COLA를 제안하였다. 이 알고리즘은 특정 하드웨어를 사용한 RTS/CTS 메커니즘, 채널 SNR 및 신호 강도 측정이 필요 없으며, 실제 운용 시 사용되는 발견적 매개변수 설정이 필요하지 않다. 이것은 802.11 링크(링크 최적화 역시 그래야 하듯이)에서의 명시적인 혼잡 레벨의 정보가 필요 없는 상황에서 운용됨에도 불구하고, 광범위한 SNR 값에 걸쳐 이상적인 결과를 얻을 수 있다. 실행 코드는 짧은 critical 경로를 갖고있어, 전통적인 802.11 MAC 기능에 최소한의 부하를 주며 실행된다. 우리는 본 논문에서 802.11a의 경우에 대해 논하였지만, COLA 알고리즘은 기타 다른 802.11 링크들에서도 적용할 수 있는 장점을 가지고 있다.

참고문헌

- [1] IEEE, Part II: *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*. IEEE Std 802.11-1999, Aug. 1999.
- [2] G. Holland, N. Vaidya, and P. Bahl, "A Rate-Adaptive MAC Protocol for Multi-Hop Wireless Networks," in proceedings of ACM Mobicom, 2001.
- [3] D. Qiao, S. Choi, and K. G. Shin, "Goodput Analysis and Link Adaptation for IEEE 802.11a Wireless LANs," *IEEE Transactions on Mobile Computing*, 1 (4), pp. 278-292, Oct.-Dec. 2002.
- [4] D. Qiao, S. Choi, "Goodput Enhancement of IEEE 802.11a Wireless LAN via Link Adaptation," in proceedings of IEEE ICC, 2001.
- [5] J. del P. Pavon and S. Choi, "Link Adaptation Strategy for IEEE 802.11 WLAN via Received Signal Strength Measurement," in proceedings of IEEE ICC, 2003
- [6] S. Choi and D. Qiao, "Fast-responsive Link Adaptation for IEEE 802.11 WLANs," in proceedings of IEEE ICC, 2005.
- [7] A. Karmar and L. Monetban, "WAVELAN II: a High Performance Wireless LAN for Unlicensed Band," *Bell Labs Technical Journal*, pp. 118-133, Summer 1997.
- [8] P. Chevillat, J. Jelitto, A. N. Barreto, and H. Truong, "A Dynamic Link Adaptation Algorithm for IEEE 802.11a Wireless LANs," in proceedings of IEEE ICC, 2003.
- [9] J. Kim, S. Kim, S. Choi, and D. Qiao, "CARA: Collision-Aware Rate Adaptation for IEEE 802.11 WLANs," to appear in proceedings of IEEE Infocom 2006.
- [10] S. Choi, K. Park and C. Kim, "On the Performance Characteristics of WANS: Revisited", in proceedings of the ACM SIGMETRICS, 2005.
- [11] I. Tinnirello, S. Choi, and Y. Kim, "Revisit of RTS/CTS Exchange in High-speed 802.11 Networks," in proceedings of IEEE WoWMoM, 2005.
- [12] IEEE 802.11a, Part II: *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: High-speed Physical Layer in the 5 GHz Band*. Supplement to IEEE 802.11 Standard, Sept. 1999.
- [13] G. Bianchi and I. Tinnirello, "Kalman Filter Estimation of the Number of Competing Terminals in an IEEE 802.11 Network," in proceedings of IEEE Infocom, 2003.
- [14] M. Heusse, F. Rousseau, R. Guillier, and A. Duda, "Idle Sense: An Optimal Access Method for High Throughput and Fairness in Rate Diverse Wireless LANs," in proceedings of ACM SIGCOMM, 2005.
- [15] The ns-2 simulator. Available at <http://www.isi.edu/nsnam/ns/>.
- [16] J. Heiskala and J. Terry, *FDM wireless LANs: a theoretical and practical guide*, SAMS Publishing, 2002.

ⁱ n 값이 10 인 ARF 3 에 반하여, n 값이 1 인 ARF 10 이 더 우월하다. 만약 비을 하락 시 충돌 발생이 주요한 경우, ARF3 는 더 빨리 상위 모드로의 변경을 시도할 것이고, 이것은 적합한 모드로 빠르게 회귀하는 데 도움이 된다.