

IEEE 802.11 무선랜 시스템에서 PCF 프로토콜의 성능을 향상시키기 위한 최적의 폴링 방식

최우용[†] · 이상완

동아대학교 산업경영공학과

Optimal Polling Method for Improving PCF MAC Performance in IEEE 802.11 Wireless LANs

Woo-Yong Choi · Sang-Wan Lee

Department of Industrial and Management Systems Engineering, Dong-A University, Pusan, 604-714

A modified PCF(Point Coordination Function) protocol with the optimal polling sequence is defined in detail and shown to improve the efficiency of the conventional PCF protocol in IEEE 802.11 wireless LAN standard. The problem for the optimal polling sequence is formulated as TSP(Travelling Salesman Problem) with the distance values of 1's or 0's. Numerical examples show that the optimal polling sequence increases the capacity of the real-time service such as VoIP(Voice over Internet Protocol).

Keywords: wireless LAN, MAC, PCF, VoIP capacity

1. 서론

전 세계적으로 고속의 데이터 전송과 사용자의 이동성을 보장하는 무선랜 시스템에 대한 관심이 급속하게 증가하여 왔다 (Jeong and Lee, 2003). 이에 따라 IEEE 802.11 WG(Working Group)에서는 Ethernet 프로토콜에 기반한 기존의 유선랜 시스템을 무선 환경에 맞게 변경하기 위한 IEEE 802.11 무선 LAN 시스템의 통신 프로토콜 규격을 탄생시켰다.

IEEE 802.11 WG에서 발표한 무선랜 규격(IEEE Std 802.11 (1999))에 따르면 IEEE 802.11 무선랜 시스템에서 BSS(Basic Service Set) 내의 STA(Station)에게 전송 기회를 주기 위한 MAC(Medium Access Control) 프로토콜은 best effort 서비스를 위한 DCF(Distributed Coordination Function)와 VoIP(Voice over Internet Protocol) 서비스와 같은 실시간 서비스를 위한 PCF(Point Coordination Function)로 구성된다. DCF 프로토콜에서 각 STA은 WM(Wireless Media)의 채널 상태를 이용하여 분산 제어 방식으로 자신의 데이터 전송 시점을 결정한다. 그리고

PCF 프로토콜에서 AP(Access Point)는 폴링 방식을 사용하여 각 STA에게 전송 기회를 부여하는데 이 때 폴링 순서는 STA ID(Identification)를 기준으로 하는 라운드 로빈(round robin) 방식을 따른다. PCF 프로토콜의 비효율적인 면은 각 STA이 어떤 하나의 MPDU(MAC Protocol Data Unit)를 AP로 전송할 때마다 이를 위하여 AP로부터 하나의 폴링 프레임의 수신하여야 한다는 것인데 이러한 폴링 프레임의 전송 횟수를 줄인다면 PCF 프로토콜의 성능은 크게 향상될 수 있을 것이다.

폴링 프레임의 전송 횟수를 줄이기 위하여 Fischer(2001), Lo, Lee and Chen(2003) 그리고 Zhao and Fan(2002)에서는 여러 개의 STA를 하나의 멀티폴링 프레임의 사용하여 한 번에 폴링하는 방식을 사용한다. 하지만 기존의 멀티폴링 방식은 모두 다음과 같은 한계를 가진다. 먼저 Fischer(2001)의 멀티폴링 방식은 AP가 멀티폴링 프레임을 전송하는 시점에 폴링하고자 하는 각 STA에 대해 데이터 프레임 전송을 위한 채널 사용 시간을 미리 지정하여야 한다는 단점을 가진다. 일반적으로 STA의 MPDU는 길이가 가변적이기 때문에 AP가 STA이 MPDU의 전

[†] 연락저자 : 최우용 교수, 604-714 부산광역시 사하구 하단2동 840번지 동아대학교 산업경영공학과, Fax : 051-200-7697, E-mail : wychoi77@dau.ac.kr

송을 위해 필요한 채널 사용 시간을 예측하기는 매우 힘들다. 그리고 이러한 예측의 어려움으로 인하여 정확한 채널 사용 시간 대신 최대 채널 사용 시간을 사용한다면 채널이 낭비될 가능성이 매우 크다. Lo, Lee and Chen(2003)의 멀티폴링 방식에서는 AP가 멀티폴링 프레임 전송하는 시점에 폴링하고자 하는 각 STA에 대해 데이터 프레임 전송을 위한 채널 사용 시간을 미리 지정하지 않고 대신 각 STA에 대해 데이터 전송 순서를 지정해 준다. 멀티폴링 프레임 받은 각 STA는 자신보다 이전에 전송하도록 폴링된 다른 STA가 전송을 마친 후 자신의 데이터를 전송하게 되는데 이를 위해서는 각 STA는 자신보다 이전에 전송하도록 폴링된 다른 STA의 송신신호를 들을 수 있어야 한다. 하지만 hidden node 문제로 인하여 이는 정확하게 수행되기 힘들며 이를 해결하기 위한 효과적인 방법은 현재 제안되어 있지 않다. Zhao and Fan(2002)에서 제안된 방식은 Lo, Lee and Chen(2003)의 방식과 유사한 문제점을 가지는데, 어떤 STA가 AP로부터 폴링 프레임을 수신하지 않고 직전 전송이 끝난 이후에 데이터 프레임을 전송하기 위해서는 이전에 전송중인 STA의 전송신호를 수신할 수 있어야 한다. 그러나 무선랜 환경에서 발생하는 hidden node 문제로 인하여 이는 정확하게 수행되기 힘들다.

본 논문에서는 앞서 제시한 멀티폴링 방식의 문제점을 개선하여 상향 링크를 통한 각 MPDU(MAC Protocol Data Unit)의 전송을 위해서 AP가 해당 STA에게 매번 폴링 프레임을 전송해야 하는 PCF의 오버헤드를 줄이기 위한 최적의 폴링 방식을 제안한다. 그리고 시뮬레이션을 통하여 기존의 PCF 프로토콜의 VoIP(Voice over Internet Protocol) 서비스 용량과 제안한 방식의 VoIP 서비스 용량을 비교 분석함으로써 제안한 방식의 효율성을 입증한다. 본 논문은 다음과 같이 구성된다. 다음 장에서 PCF 프로토콜 그리고 hidden node 문제에 대해 설명한다. 그리고 3장에서 본 논문에서 제안하는 최적의 폴링 방식을 설명하고 4장에서 시뮬레이션 분석 결과를 제시한다. 마지막으로 5장에서 결론을 맺는다.

2. PCF 프로토콜과 Hidden Node 문제

PCF 프로토콜은 실시간 서비스를 제공하기 위해 제안된 MAC

프로토콜로서 AP 내에 PC(Point Coordinator)가 존재하여 각 STA는 PC의 폴링 프레임을 전송받았을 때에만 프레임을 전송할 권리를 가진다.

PCF에서 AP는 PIFS(PCF InterFrame Space) 시간을 이용하여 채널 액세스를 시도한다. CFP(Contention Free Period) 내에서 AP는 폴링 프레임을 STA에 전송하여 폴링 프레임 하나당 하나씩의 데이터 프레임을 전송할 기회를 해당되는 STA에 부여한다. AP는 PIFS를 이용하여 비콘 프레임을 전송하며 CFP 내에서 비콘 프레임을 전송한 후 SIFS(Short InterFrame Space) 시간을 이용하여 STA ID를 기준으로 라운드 로빈 방식으로 각 STA를 순서대로 폴링한다. 그리고 AP로부터 폴링 프레임을 수신한 STA는 SIFS 시간 이후에 데이터 프레임을 AP로 전송한다. 만약 AP의 폴링에 대해서 어떤 STA이 SIFS 시간 내에 응답하지 않으면 직전 폴링 프레임 전송 시간을 기준으로 PIFS 시간 이후에 다음 STA에 폴링 프레임을 전송한다. CFP 내에서 DCF 프로토콜에 의한 STA의 채널 액세스를 막기 위하여 AP는 비콘 프레임 전송 시 모든 STA의 NAV(Network Allocation Vector) 값을 CFP 시간으로 조정하도록 명령한다. 각 STA의 NAV는 DCF 프로토콜을 통하여 채널을 액세스할 수 없는 앞으로 남은 시간을 의미한다. 이러한 NAV 값의 조정으로 인하여 모든 STA는 CFP 내에서 AP의 폴링 없이는 데이터 프레임을 전송할 수 없게 된다(<Figure 1>).

Hidden node 문제는 어떤 임의의 BSS 내에서 어떤 STA의 전송신호를 다른 STA이 물리적으로 제대로 감지하지 못하는 경우에 발생한다. 예를 들어 <Figure 2>에서 어떤 STA A가 AP로 현재 데이터 프레임을 전송하는 중이고 STA B는 AP로 데이터 프레임을 전송하려고 시도한다고 하자. 그런데 STA B가 STA A의 전송신호를 제대로 수신하지 못하는 경우 STA B는 채널이 현재 idle하다고 판정하고 AP로 데이터를 전송할 수 있다. 결국 STA A와 STA B의 데이터 프레임은 서로 충돌하게 되고 STA A와 STA B는 모두 데이터 프레임을 제대로 전송할 수 없게 된다(<Figure 2>에서 원은 STA의 신호 수신 가능 영역을 나타낸다). PCF 프로토콜에서는 AP가 각 STA에 대해 별도의 폴링 프레임을 전송하기 때문에 BSS 내에서 hidden node 문제가 발생하지 않는다. 하지만 폴링 프레임의 전송 횟수를 줄이기 위한 기존의 방법들에서는 서론에서 설명한 것과 같은 문제점

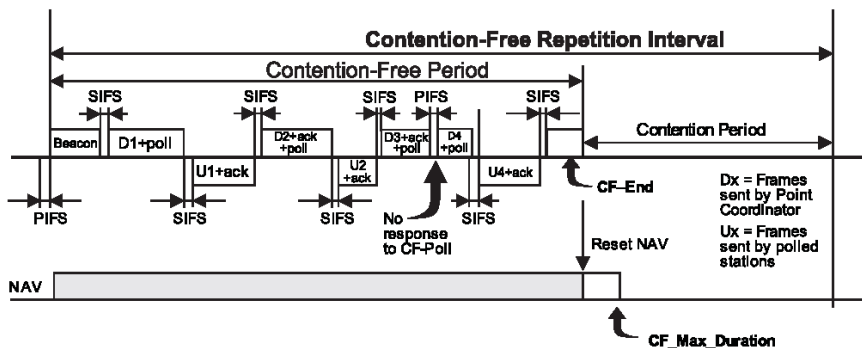
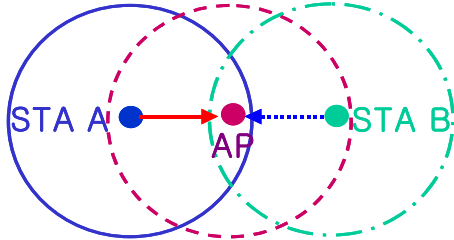


Figure 1. PCF polling scheme.

을 가진다. 특히, Lo, Lee and Chen(2003)과 Zhao and Fan(2002)에서는 멀티폴링 프레임에 의해 지정된 각 STA는 hidden node 문제로 인하여 자신보다 이전에 전송하도록 폴링된 다른 STA의 송신신호를 감지할 수 없는 경우가 발생하여 데이터 전송이 제대로 수행되지 못하는 문제점이 발생한다. 본 논문은 이러한 문제점을 해결하기 위한 방법을 다음 장에서 제안한다.



STA A: currently transmitting
STA B: attempting to transmit

Figure 2. Hidden node problem.

3. 최적의 폴링 방식

BSS 내의 STA와 AP 간의 association/reassociation/disassociation 과정을 통하여 AP는 CFP 내에서 PCF를 통하여 전송 기회를 부여해야 하는 STA의 리스트를 관리할 수 있다. BSS 내에서 AP가 PCF를 통하여 전송 기회를 부여해야 하는 STA의 개수를 N 이라 한다. BSS 내에서 AP는 이러한 N 개의 STA와 데이터를 직접 송·수신할 수 있다. 하지만 N 개의 STA는 서로 직접적으로 데이터를 송·수신할 수 없고 반드시 AP를 통하여 간접적으로 데이터를 송·수신해야 한다.

Hidden node 문제로 인하여 N 개의 STA는 현재의 채널 상태 (즉, N 개의 STA 중 어떤 STA이 무선 채널을 통하여 데이터를 전송하는 중인지 아닌지를 나타내는 상태)를 정확히 파악할 수 없고 AP만이 채널 상태를 정확히 파악할 수 있다. PCF 프로토콜에서 AP는 채널 상태를 파악하여 현재 채널 상태가 idle인 경우(즉, 어떤 STA도 무선 채널을 통하여 데이터를 전송하지 않는 상태) 어떤 STA에 폴링 프레임을 전송함으로써 해당 STA에 전송 기회를 부여한다. 그런데 각 STA는 BSS 내에 존재하는 다른 모든 STA의 데이터 전송 유무를 정확하게 파악하지 못할지라도 거리상으로 인접한 다른 STA의 송신 유무는 정확히 파악할 수 있다. 본 논문에서는 이러한 특성에 착안하여 AP가 N 개의 STA를 폴링할 때 가능하면 폴링 순서상의 각 STA가 직전에 폴링된 STA의 전송신호를 파악할 수 있도록 폴링 순서를 유도하고자 한다. 이렇게 함으로써 AP는 N 개의 모든 STA에 별도의 폴링 프레임을 전송할 필요가 없고 폴링 순서상에서 직전에 폴링된 STA의 전송신호를 제대로 파악할 수 없는 STA에게만 별도의 폴링 프레임을 전송할 수 있다. 본 장에서는 폴링 순서상에서 직전에 폴링된 STA와의 연결이 이루

어지지 않은 경우의 수가 최소가 되는 폴링 순서를 유도하는 문제를 TSP(Traveling Salesman Problem)로 변환한다.

본 논문에서는 BSS 내의 STA 간에 발생할 수 있는 hidden node 문제를 해결하기 위하여 BSS 내의 각 STA가 다른 STA의 신호를 제대로 수신할 수 있는지 없는지에 대한 STA 간의 연결정보를 AP가 알 수 있다고 가정한다. 그럼 CFP 내에서 AP가 STA 간의 연결 정보를 수집하는 방법을 설명한다. STA i 는 현재의 상태에서 자신이 신호를 수신할 수 있는 다른 STA의 MAC 주소의 집합, S_i 를 관리해야 하고 AP는 각 STA로부터 전달된 연결 정보를 바탕으로 N 개의 STA의 연결 정보 (S_1, S_2, \dots, S_N)를 관리해야 한다. AP는 자신의 BSS 내의 STA 간의 연결 정보를 알지 못하는 초기 상태에서 STA ID를 기준으로 각 STA를 순서대로 폴링한다. AP로부터 폴링 프레임을 전송받은 STA i 는 과거 자신이 폴링을 받은 시점부터 현재까지 전송한 STA 중 MAC 주소가 S_i 에 있지 않고 전송신호를 수신할 수 있는 STA가 있다면 해당되는 모든 STA의 MAC 주소를 S_i 에 추가한다. 그리고 MAC 주소가 S_i 에 있지만 과거 한 번의 폴링 주기 동안 전송신호를 수신할 수 없는 STA가 있다면 해당되는 모든 STA의 MAC 주소를 S_i 에서 삭제한다. 그리고 STA i 는 이러한 S_i 에 추가 혹은 삭제되는 STA의 MAC 주소를 AP로 전송하는 데이터 프레임에 piggyback하여 AP에 알려준다(만약 폴링 받은 시점에 STA i 가 전송할 데이터 프레임이 없으면 null 프레임을 전송하고 S_i 에 추가 혹은 삭제되는 STA의 MAC 주소는 null 프레임에 piggyback한다).

이러한 방법을 이용하면 STA 간의 연결 상태가 대칭적이고 (즉, STA i 가 STA j 의 신호를 수신할 수 있으면 STA j 가 STA i 의 신호를 수신할 수 있는 경우) STA와 AP 간에 전송 에러가 발생하지 않는 경우 AP는 한 번의 폴링 주기만으로 현재의 STA 간의 연결 정보를 수집할 수 있다(각 STA는 AP로부터 폴링 받은 시점에서 폴링 주기 내에서 이전에 전송한 STA와 자신의 연결 정보를 AP에 보고하기 때문에 STA 간의 연결 상태가 대칭적인 경우 한 번의 폴링 주기만으로 AP는 STA 간의 연결 정보를 모두 수집할 수 있다). 그리고 STA 간의 연결 정보가 변하면 AP는 그 연결 정보의 변화를 다음 폴링 주기에서 감지할 수 있다. 만약 STA 간의 연결 상태가 비대칭적이라 하더라도 STA와 AP 간에 전송 에러가 발생하지 않는 경우 AP는 두 번의 폴링 주기만으로 현재의 STA 간 연결 정보를 수집할 수 있다.

예를 들어 AP가 자신에게 보고된 각 STA의 신호 수신 상태 정보를 바탕으로 순환적으로 서로 연결된 STA의 순서를 (STA 1, STA 2, ..., STA N)와 같이 유도할 수 있었다고 하자(여기서 순환적인 연결의 의미는 폴링 순서상의 서로 인접한 STA이 서로 연결되어 있고 폴링 순서상의 마지막 STA와 폴링 순서상의 첫 번째 STA이 서로 연결된 것을 의미한다). 그러면 AP는 이러한 N 개의 STA를 하나의 멀티폴링 프레임을 사용하여 폴링할 수 있다. 왜냐하면 각 STA는 순서상의 직전의 STA의 전송이 끝난 후 SIFS 이후에 자신의 데이터를 전송할 수 있기 때문

이다. 각 STA는 직전의 STA의 신호를 수신할 수 있기 때문에 AP로부터 별도의 폴링이 필요 없다. 이러한 경우는 폴링 프레임의 전송 횟수가 최소화되는 경우로서 일반적으로 BSS 영역 내에 STA가 균일하게 분포하는 경우에 해당할 것이다. 일반적으로 N개의 STA가 M개의 그룹으로 나누어지고 각 그룹 내에서는 STA 간에 순차적으로 연결된 STA의 순서가 존재할 것이다(가장 최악의 경우는 각 그룹이 하나의 STA만으로 구성되는 경우이다). 이 경우 AP는 각 그룹을 하나의 멀티폴링 프레임을 사용하여 한꺼번에 폴링할 수 있다. 그리고 M개의 멀티폴링 프레임을 전송함으로써 N개의 STA를 폴링할 수 있다.

본 논문의 목적은 PCF 프로토콜에서 N개의 STA에게 전송 기회를 부여하기 위하여 사용하는 폴링 프레임의 전송 횟수를 최소화함으로써 PCF의 성능을 향상시키는 것이다. 본 논문에 의해 구해진 최적 폴링 순서에 의해 AP는 N개의 STA를 순서대로 폴링한다. 만약 하나의 CFP 내에서 한 번의 폴링 주기를 통하여 N개의 STA를 모두 한 번씩 폴링하였다면 CFP의 나머지 잔여시간 동안 N개의 STA를 순서대로 계속적으로 폴링한다고 가정한다. 이 때 CFP의 잔여시간이 부족하여 AP가 어떤 하나의 폴링 주기에서 N개의 STA 중 일부 STA만을 폴링하였다면 AP는 순서상의 나머지 STA를 다음 CFP에서 계속적으로 폴링한다고 가정한다. 하지만 본 논문의 최적 폴링 방식에서는 STA 간의 연결 상태를 고려하여 최적 폴링 순서는 폴링 주기 단위로 변환할 수 있고 AP는 이를 반영하여 N개의 STA를 폴링하게 된다(본 논문에서는 상향 트래픽만을 고려하였는데 하향 트래픽은 CFP 내에서 상향 트래픽이 전송되기 전에 서비스되도록 스케줄링할 수 있을 것이다).

이제 N개의 STA 간의 어떤 주어진 연결 상태에 대해서 N개의 STA를 폴링하기 위한 최적 순서를 분석한다. 먼저 각 STA의 다른 STA의 신호 수신 상태를 나타내기 위한 indicator 상수, $C_{i,j}, i \neq j$ 를 다음과 같이 정의한다.

$$C_{i,j} = \begin{cases} 0 & (\text{STA } j \text{가 STA } i \text{의 신호를 수신할 수 있는 경우}) \\ 1 & (\text{STA } j \text{가 STA } i \text{의 신호를 수신할 수 없는 경우}) \end{cases} \quad (1)$$

그러면 BSS 내 N개의 STA를 폴링할 때 폴링 횟수를 최소화하는 문제는 다음의 TSP로 변환 가능하다.

$$\text{Minimize } Z = \sum_{i=1}^N \sum_{j=1}^N C_{i,j} X_{i,j} \quad (2)$$

subject to

$$\sum_{j:j \neq i} X_{i,j} = 1, i = 1, 2, \dots, N \quad (3)$$

$$\sum_{i:i \neq j} X_{i,j} = 1, j = 1, 2, \dots, N \quad (4)$$

$$\sum_{i \in U_j} \sum_{j \in U_i} X_{i,j} \geq 1, U \subset T, |U| \geq 2 \quad (5)$$

$$X_{i,j} \in \{0, 1\}, i = 1, 2, \dots, N, j = 1, 2, \dots, N, i \neq j \quad (6)$$

$$T = \{1, 2, \dots, N\} \quad (7)$$

앞의 수식에서 $X_{i,j}$ 는 0 또는 1의 값을 가지는 변수로서 $X_{i,j} = 1$ 은 STA i 를 폴링한 다음 STA j 를 폴링함을 의미하며 $X_{i,j} = 0$ 은 STA i 를 폴링한 다음 STA j 를 폴링하지 않음을 의미한다. 수식 (2)는 N개의 STA를 모두 폴링하기 위해 필요한 폴링 프레임 전송 횟수를 최소화하기 위한 목적함수이다. STA i 를 폴링한 다음 STA j 를 폴링할 때 만약 $C_{i,j} = 0$ 이면(즉, STA j 가 STA i 의 신호를 수신할 수 있는 경우) 목적함수의 $C_{i,j} X_{i,j} = 0$ 이다. 그런데 STA i 를 폴링한 다음 STA j 를 폴링할 때 만약 $C_{i,j} = 1$ 이면(즉, STA j 가 STA i 의 신호를 수신할 수 없는 경우) 목적함수의 $C_{i,j} X_{i,j} = 1$ 이다. 따라서 수식 (2)의 목적함수를 최소화하는 것은 곧 직전 STA의 전송신호를 듣지 못하는 다른 STA를 다음에 폴링하는 경우의 수를 줄이는 것과 같으며 이 것은 멀티폴링 프레임의 전송 횟수를 줄이는 것과 같다. 다시 말하면 N개의 STA를 위한 폴링 순서를 유도할 때 인접한 STA 간의 연결이 존재하지 않는 경우의 수를 최소화하는 것이다. 수식 (3)은 STA i 를 폴링한 다음 어떤 하나의 STA j 를 반드시 폴링하여야 함을 의미하며 수식 (4)는 STA j 를 폴링하기 전에 다른 어떤 하나의 STA i 를 폴링하여야 함을 의미한다(수식 (3)에서 만약 STA i 가 폴링 순서상의 마지막이라면 AP는 다시 처음으로 돌아가 순서상의 첫번째 STA를 다음에 폴링한다고 가정하였다. 그리고 수식 (4)에서 만약 STA j 가 순서상의 첫번째 STA이라면 AP는 순서상의 마지막 STA를 STA j 직전에 폴링한다고 가정하였다). 수식 (5)는 서로 독립된 여러 개의 폴링 순서가 만들어지는 경우를 배제하기 위해 추가된 제한식으로서 임의의 STA의 부분집합 U 와 U 의 여집합에 있는 어떤 두 개의 STA는 반드시 폴링 순서로 연결되어야 함을 의미한다. 예를 들어 $N=4$ 인 경우($X_{1,2} = 1, X_{2,1} = 1, X_{3,4} = 1, X_{4,3} = 1$)는 수식 (3)과 (4)를 만족하지만 이로부터 유도되는 폴링 순서는 (STA 1, STA 2)와 (STA 3, STA 4)로서 4개의 STA를 모두 포함하는 하나의 폴링 순서가 아니다. 이러한 경우를 배제하기 위해 수식(5)가 추가되었다. 수식 (7)에서 T 는 N개의 STA의 집합을 의미한다.

만약 최소 목적함수 값 $Z^* = 0$ 이면 N개의 STA를 포함하는 서로 순환적으로 연결된 하나의 폴링 순서가 존재한다. 이 경우 한 개의 폴링 프레임의 전송만으로 N개의 STA에 대한 폴링이 가능하다. 그리고 $Z^* \geq 1$ 이면 N개의 STA는 서로 순차적으로 연결된 Z^* 개의 그룹으로 구성된다. 이 경우 N개의 STA를 폴링하기 위해서는 최소한 Z^* 개의 폴링 프레임의 전송이 필요하다. 그리고 이에 해당하는 최적의 폴링 순서는 $X_{i,j}^*, i \neq j$ 와 $C_{i,j}, i \neq j$ 를 사용하여 구할 수 있다.

4. 시뮬레이션 분석 결과

<Figure 3>은 IEEE 802.11a 무선랜의 4가지 경우($N=5, N=10,$

N=20, N=30)를 보여준다. <Figure 3>에서 STA 간의 선은 STA가 다른 STA의 전송신호를 수신할 수 있음을 의미한다 <Figure 3>에서 어떤 두 개의 STA의 전송신호 수신 가능 여부는 서로 대칭적이라고 가정한다. 4가지 경우의 BSS 내의 STA가 hidden node 문제로 인하여 서로 완전히 연결되어 있지 않지만 Wolsey(1998, Figure 7.2)와 Nilsson(1980, Figure 1.6)의 enumeration tree를 효율적으로 탐색하여 최적해를 찾기 위해 다음과 같은 탐색 알고리즘을 사용하여 분석한 결과 BSS 내의 모든 STA는 순환적으로 연결 가능하다는 것을 알 수 있었다(다음의 알고리즘에서 tour는 최적해를 찾기 위하여 계속적으로 변경되며 구성중인 STA의 폴링 순서로서 초기에는 어떤 STA도 포함하지 않는 순서로 시작한다. 그리고 C는 탐색중에 발견된 완전한 tour 중에서 최소의 비용을 가지는 폴링 순서의 비용이다. C는 알고리즘의 초기에 무한대의 값을 가진다. 본 논문에서는 알고리즘의 구현의 편의를 위하여 enumeration 기반의 알고리즘을 사용하였으나 Blaser(2004)와 Lewenstein and Sviridenko(2003)와 같은 asymmetric TSP를 위한 알고리즘을 사용할 수도 있을 것이다).

Enumeration Tree의 효율적인 탐색 알고리즘

STEP 0

한 개의 STA도 포함하지 않는 tour로 시작하며 C는 무한대의 값을 가지도록 한다.

STEP 1

Enumeration tree의 모든 경우를 탐색하였으면 알고리즘을 종료하고 C에 해당되는 폴링 순서를 최적해로 보고하고 알고리즘을 종료한다. Enumeration tree의 모든 경우를 탐색하지 않았으면 Nilsson(1980, Figure 2.6)의 depth-first 방식을 사용하여 현재의 tour를 다음의 탐색 순서에 해당하는 tour로 변경한다.

STEP 2

STEP 2.1: 현재의 tour가 완전한 경우 tour의 비용을 계산하여 C와 비교한다. 만약 현재의 tour 비용이 C보다 작다면 C를 현재의 tour 비용으로 바꾼다. 만약 tour 비용이 C보다 크거나 같다면 STEP 1으로 돌아간다.

STEP 2.2: 현재의 tour가 불완전한 경우 현재의 tour에 의하여 야기되는 최소 비용을 계산한다(이 때, 최소 비용은 현재의 tour를 완전하게 하기 위해 추가적으로 소요되는 비용을 0으로 가정하여 계산되는 비용을 의미한다). 만약 tour의 최소 비용이 C보다 크거나 같다면 현재 tour의 STA 순서를 변경하지 않으며 tour를 완전하게 할 수 있는 나머지 모든 경우를 이후의 탐색에서 생략하고 STEP 1으로 돌아간다. 만약 tour의 최소 비용이 C보다 작다면 tour에 STA를 추가하기 위하여 STEP 1으로 돌아간다.

앞에서 설명한 탐색 알고리즘을 사용하여 얻어진 각 경우의 최적의 폴링 순서는 하나의 멀티폴링만으로 N개의 모든 STA를 폴링할 수 있는 경우로서 다음과 같다(아래의 결과는 3.0

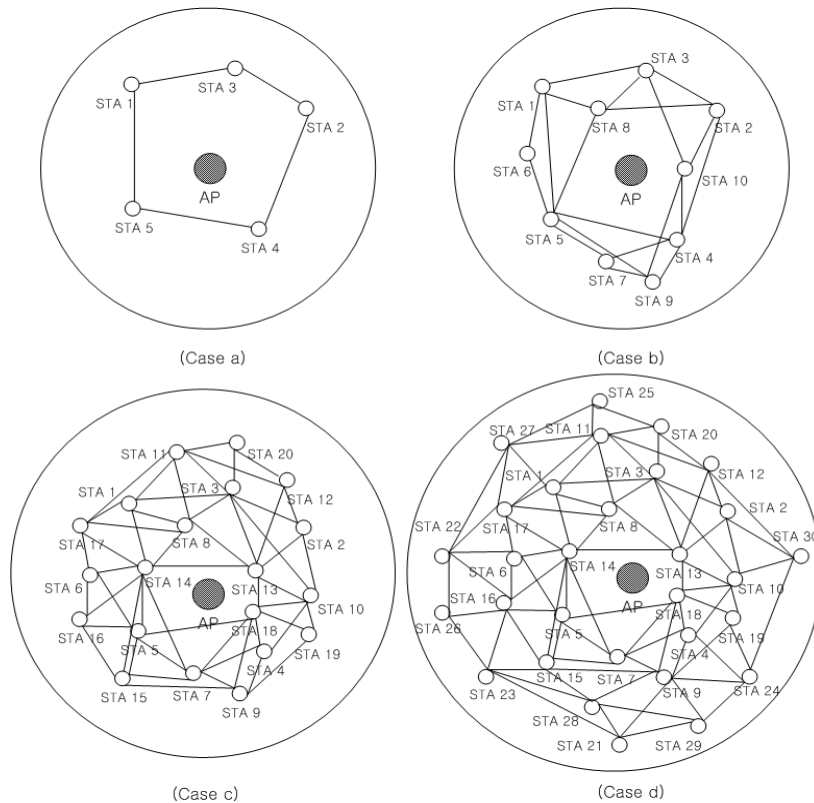


Figure 3. Four Cases of IEEE 802.11a wireless LAN.

GHz Intel CPU를 탑재한 PC에서 앞의 탐색 알고리즘을 C 언어로 구현하여 얻어진 결과로서 Case a, Case b, Case c의 경우 0.5초 이내에 결과가 얻어졌으며 Case d의 경우 10초 이내에 결과가 얻어졌다).

Case a: (STA 1, STA 3, STA 2, STA 4, STA 5)

Case b: (STA 1, STA 3, STA 8, STA 2, STA 4, STA 10, STA 9, STA 7, STA 5, STA 6)

Case c: (STA 1, STA 3, STA 2, STA 10, STA 4, STA 7, STA 5, STA 6, STA 14, STA 16, STA 15, STA 9, STA 19, STA 18, STA 13, STA 12, STA 20, STA 11, STA 8, STA 17)

Case d: (STA 1, STA 3, STA 2, STA 10, STA 4, STA 7, STA 5, STA 6, STA 14, STA 8, STA 13, STA 12, STA 30, STA 24, STA 19, STA 18, STA 9, STA 21, STA 29, STA 28, STA 15, STA 16, STA 23, STA 26, STA 22, STA 17, STA 11, STA 20, STA 25, STA 27)

N개의 STA가 순환적으로 연결되지 않는 경우를 생성하기 위하여 <Figure 3>의 Case c에서 STA 15가 고립되었을 경우(즉, STA 15와 다른 STA 간의 연결이 이루어지지 않았을 경우로 Case c'으로 표시)와 Case c에서 STA 15와 STA 9가 모두 고립되었을 경우(Case c"으로 표시)에 대해서 앞의 탐색 알고리즘을 적용하면 다음의 폴링 순서를 유도할 수 있다.

Case c': (STA 1, STA 3, STA 2, STA 10, STA 4, STA 7, STA 5, STA 6, STA 14, STA 16, STA 15, STA 9, STA 19, STA 18, STA 13, STA 12, STA 20, STA 11, STA 8, STA 17)

Case c'': (STA 1, STA 3, STA 2, STA 9, STA 15, STA 4, STA 7, STA 5, STA 18, STA 19, STA 10, STA 13, STA 12, STA 20, STA 11, STA 8, STA 14, STA 16, STA 6, STA 17)

Case c'의 최적해를 구하기 위해 소요되는 시간은 약2분이었

으며 Case c"의 최적해를 구하기 위해 소요되는 시간은 약7분이었다. 하지만 Case d에서 STA 23이 고립되었을 경우(Case d'으로 표시)에 대해서는 너무 많은 시간이 소요됨으로써 이에 대한 최적해를 구할 수 없었다.

앞의 Case c'과 Case c"에 대해서 유도된 폴링 순서는 AP가 N개의 STA를 폴링하기 위한 순서를 나타내는데 실제 AP는 서로 순차적으로 연결된 STA들을 묶어서 하나의 폴링 프레임을 사용하여 폴링하게 된다. 이를 위해서는 서로 순차적으로 연결된 STA의 그룹을 유도하여야 할 것이다. 이러한 STA의 그룹을 유도하는 방식은 매우 간단하다. 먼저 유도된 폴링 순서의 처음과 끝을 연결하여 링 형태로 만든 다음 인접한 STA 간에 연결되지 않은 지점을 찾는다. 그리고 링에서 서로 연결되지 않은 지점을 기점으로 하여 STA의 그룹을 구성하면 각 그룹의 STA의 순서는 서로 순차적으로 연결된 폴링 순서가 된다. 예를 들어 Case c'에 대해 유도된 앞의 폴링 순서에서 STA 16과 STA 15가 서로 연결되어 있지 않으며 STA 15와 STA 9가 서로 연결되어 있지 않다. 따라서 이러한 지점을 기점으로 하여 그룹을 구성하면 다음과 같이 두 개의 서로 순차적으로 연결된 폴링 순서가 유도된다.

(STA 9, STA 19, STA 18, STA 13, STA 12, STA 20, STA 11, STA 8, STA 17, STA 1, STA 3, STA 2, STA 10, STA 4, STA 7, STA 5, STA 6, STA 14, STA 16),
(STA 15)

그리고 Case c" 대해서도 앞의 방식을 적용하면 마찬가지로 서로 순차적으로 연결된 3개의 폴링 순서를 다음과 같이 유도할 수 있다.

(STA 4, STA 7, STA 5, STA 18, STA 19, STA 10, STA 13, STA 12, STA 20, STA 11, STA 8, STA 14, STA 16, STA 6, STA 17, STA 1, STA 3, STA 2),
(STA 9),
(STA 15)

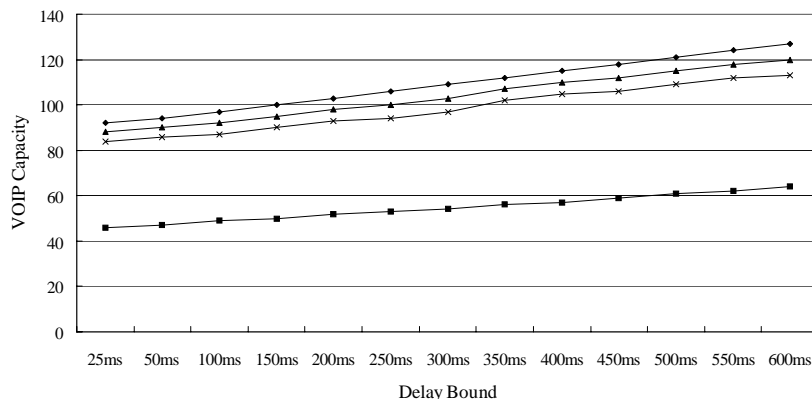


Figure 4. Simulation result(PCF, Case c, c' and c'').

<Figure 3>의 Case c, d와 앞에서 설명한 Case c', c"에 대해서 기존의 PCF에 의한 VoIP 서비스 용량과 본 논문에서 새롭게 제안한 폴링 방식에 의한 VoIP 서비스 용량을 시뮬레이션을 통해 분석하여, 그 분석 결과를 <Figure 4>와 <Figure 5>에 나타내었다. <Figure 4>에서 다이아몬드로 표시된 선이 Case c에서 본 논문에서 제안한 방식에 의한 VoIP 서비스 용량이고 삼각형으로 표시된 선이 Case c'에서 본 논문에서 제안한 방식에 의한 VoIP 서비스 용량이고 x로 표시된 선이 Case c"에서 본 논문에서 제안한 방식에 의한 VoIP 서비스 용량이며 다른 선이 기존의 PCF 프로토콜에 의한 VoIP 서비스 용량이다. 그리고 <Figure 5>에서 다이아몬드로 표시된 선이 Case d에서 본 논문에서 제안한 방식에 의한 VoIP 서비스 용량이고 다른 선이 기존의 PCF 프로토콜에 의한 VoIP 서비스 용량이다. <Figure 4>와 5의 VoIP 서비스 용량은 STA와 AP 간의 전송 에러 확률이 0.1%인 경우 99% 이상의 VoIP 데이터 프레임이 어떤 주어진 전송 지연 시간(delay bound)을 만족시키면서 동시에 서비스 가능한 최대의 VoIP 상향 트래픽 스트림 개수로 정의된다. <Figure 4>와 <Figure 5>의 시뮬레이션 분석 결과의 95% 신뢰구간의 간격은 2보다 작다(<Figure 4>와 <Figure 5>의 PCF에 의한 VOIP 서비스 용량을 구하기 위한 이론적인 분석 방법은 Choi(2004)에서 연구되었다).

시뮬레이션 분석을 위해서 CP(Contention Period)와 CFP는 각각 10ms라 하였고, CFP 중 5ms를 폴링 프레임의 전송과 상향 링크를 통한 데이터 전송에 할당하였다. Comanicu and Mandayam(2000)으로부터 VoIP 트래픽 스트림의 버스트 구간의 길이와 버스트 도착 간격을 각각 1.5초와 1초로 두고자 한다. 그리고 각 VoIP MPDU 내의 payload의 길이는 88 bits이다 (Zahedi and Pahlavan, 2000). 그리고 IMBE(Improved Multi-band Excitation) 음성 코딩 방식을 사용할 경우 하나의 버스트 구간에서 생성되는 MPDU 개수는 $4.8 \text{ Kbps} * 1.5 \text{ 초} / 88 \text{ bits} = 82$ 개이다. 여기서 4.8 Kbps는 IMBE 음성 코딩 방식을 사용했을 경우의 음성 코딩률이다(Zahedi and Pahlavan, 2000).

<Figure 6>과 같이 각 MPDU에는 UDP(User Datagram Protocol) 계층, IP(Internet Protocol) 계층 그리고 MAC 계층의 헤더가 포함되어 있다. 그러므로 버스트의 크기는 $4.8 \text{ Kbps} * 1.5 \text{ seconds} + 82 * (\text{UDP, IP 그리고 MAC 헤더 길이}) = 7,200 \text{ bits} + 82 * (16 \text{ bits} + 224 \text{ bits} + 224 \text{ bits}) = 45,248 \text{ bits}$ 이다. 여기서 UDP 계층, IP 계층 그리고 MAC 계층의 헤더 길이는 IEEE Std 802.11(1999)와 Zahedi and Pahlavan(2000)으로부터 구해졌다 (편의상 MAC 계층의 CRC 필드는 MAC 헤더에 포함시켰다). VoIP 트래픽 스트림당 실제 최대 전송률은 $45,248 \text{ bits} / 1.5 \text{ seconds} = 30.2 \text{ Kbps}$ 이며 평균 전송률은 $30.2 \text{ Kbps} * 1.5 \text{ seconds} / (1.5 \text{ seconds} + 1 \text{ second}) = 18.1 \text{ Kbps}$ 이다. IEEE 802.11a 무선랜 시스템에서 SIFS = $16 \mu\text{s}$, PIFS = $25 \mu\text{s}$ 이며 물리 계층의 최대 데이터 전송률은 54Mbps이고 물리 계층의 헤더 전송시간은 $24 \mu\text{s}$ 이다(IEEE Std 802.11a(1999)).

1.5 seconds + 82 * (UDP, IP 그리고 MAC 헤더 길이) = 7,200 bits + 82 * (16 bits + 224 bits + 224 bits) = 45,248 bits이다. 여기서 UDP 계층, IP 계층 그리고 MAC 계층의 헤더 길이는 IEEE Std 802.11(1999)와 Zahedi and Pahlavan(2000)으로부터 구해졌다 (편의상 MAC 계층의 CRC 필드는 MAC 헤더에 포함시켰다). VoIP 트래픽 스트림당 실제 최대 전송률은 $45,248 \text{ bits} / 1.5 \text{ seconds} = 30.2 \text{ Kbps}$ 이며 평균 전송률은 $30.2 \text{ Kbps} * 1.5 \text{ seconds} / (1.5 \text{ seconds} + 1 \text{ second}) = 18.1 \text{ Kbps}$ 이다. IEEE 802.11a 무선랜 시스템에서 SIFS = $16 \mu\text{s}$, PIFS = $25 \mu\text{s}$ 이며 물리 계층의 최대 데이터 전송률은 54Mbps이고 물리 계층의 헤더 전송시간은 $24 \mu\text{s}$ 이다(IEEE Std 802.11a(1999)).

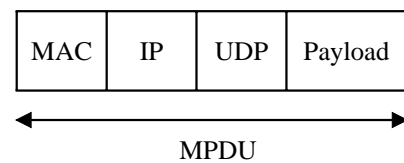


Figure 6. Format of VoIP MPDU.

STA와 AP 간의 전송 에러 확률이 0.1%인 경우 3장에서 설명한 STA 간의 연결 정보 수집 방법을 사용하여 Case c와 d의 경우 AP는 평균적으로 각각 2.8ms와 4.2ms만에 STA 간의 연결 정보를 모두 수집할 수 있었다(2.8ms와 4.2ms는 폴링 주기인 2.7ms와 4.0ms보다 약간 큰 값이다. 전송 에러로 인하여 연결 정보 수집을 위하여 폴링 주기보다 약간 더 많은 시간이 필요하다). <Figure 4>와 <Figure 5>의 분석 결과로부터 제안한 최적의 폴링 방식은 폴링 프레임의 전송 횟수를 줄이는 것으로 현재의 PCF 성능을 크게 향상시킨다는 것을 알 수 있다.

5. 결론과 추후 연구

본 논문에서는 PCF 프로토콜의 성능을 향상시키기 위하여 폴링 프레임의 전송 횟수를 최소화하는 최적의 폴링 방식을 제안하였다. 그리고 기존의 PCF 프로토콜에 의한 VoIP 서비스 용

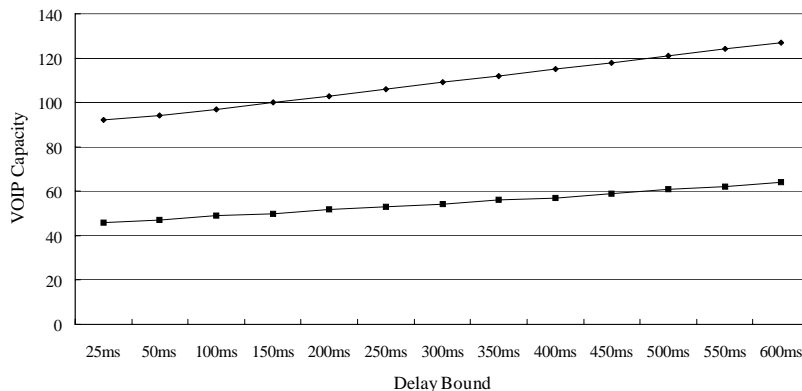


Figure 5. Simulation result(PCF and Case d).

량과 최적의 폴링 방식에 의한 VoIP 서비스 용량을 시뮬레이션을 통하여 비교 분석함으로써 제안한 방식의 성능 향상 효과를 입증하였다. 특히 BSS 내의 STA이 모두 서로 순환적으로 연결되는 경우 STA를 연결하는 최적 폴링 순서는 빠른 시간 내에 찾을 수 있음을 실험 결과를 통해 확인할 수 있었다. 하지만 STA 개수가 많은 BSS 내에서 STA이 서로 순환적으로 연결되지 않는 경우 최적 폴링 순서를 찾기 위해서는 많은 시간이 소요되는데 이러한 문제점을 개선하기 위하여 좀더 효율적인 탐색 방식에 대한 연구가 앞으로 계속적으로 필요하다. 특히 무선랜에 연결된 STA의 개수가 클 경우 여러 개의 TSP로 나누어서 문제를 해결하는 multiple TSP에 대한 해법을 이용할 필요가 있다.

참고문헌

- IEEE Std 802.11 (1999), *Wireless LAN medium access control (MAC) and physical layer specifications*.
- IEEE Std 802.11a (1999), *Wireless LAN medium access control(MAC) and physical layer specifications: high speed physical layer in the 5 GHz band*.
- Blaser, M. (2004), An 8/13 approximation algorithm for the asymmetric maximum TSP, *Journal of Algorithm*, **50**, 23-48.
- Choi, W.-Y. (2004), A Centralized MAC-Level Admission Control Algorithm for Traffic Stream Services in IEEE 802.11e Wireless LAN, *AEU-International Journal of Electronics and Communications*, **58**, 305-309.
- Comaniciu, C. and Mandayam, N. (2000), Delta modulation based prediction for access control in integrated voice/data CDMA systems, *IEEE Journal on Selected Areas in Communications*, **18**, 112-122.
- Fischer, M. (2001), QoS baseline proposal for the IEEE 802.11e, IEEE Doc. IEEE 802.11-00/360.
- Jeong, D.W. and Lee, C.Y. (2003), Downlink-first scheduling of real-time voice traffic in IEEE 802.11 wireless LANs, *Journal of Korean Institute of Industrial Engineers*, **29**, 150-156.
- Lewenstein, M. and Sviridenko, M. (2003), Approximating asymmetric maximum TSP, *Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms*, 646-654.
- Lo, S.C., Lee, G. and Chen, W.T. (2003), An Efficient Multipolling Mechanism for IEEE 802.11 Wireless LANs, *IEEE Transactions on Computers*, **52**, 764-778.
- Nilsson, N.J. (1980), *Artificial Intelligence*, Springer-Verlag.
- Wolsey, L.A. (1998), *Integer programming*, John Wiley & Sons.
- Zahedi, A. and Pahlavan, K. (2000), Capacity of a wireless LAN with voice and data services. *IEEE Transactions on Communications*, **48**, 1160-1170.
- Zhao, L. and Fan, C. (2002), M-PCF: modified IEEE 802.11 PCF protocol implementing QoS, *Electronics Letters*, **38**, 1611-1613.