

WiFi의 간섭을 평가하기 위한 IEEE 802.15.4 채널분석기의 구현

Implementation of IEEE 802.15.4 Channel Analyzer for Evaluating WiFi Interference

송 명 렬* · 진 현 준†

(Myong-Lyol Song · Hyun-Joon Jin)

Abstract - In this paper, an implementation of concurrent backoff delay process on a single chip with IEEE 802.15.4 hardware and 8051 processor core that can be used for analyzing the interference on IEEE 802.15.4 channels due to WiFi traffics is studied. The backoff delay process of IEEE 802.15.4 CSMA-CA algorithm is explained. The characteristics of random number generator, timer, and CCA register included in the single chip are described with their control procedure in order to implement the process. A concurrent backoff delay process to evaluate multiple IEEE 802.15.4 channels is proposed, and a method to service the associated tasks at sequentially ordered backoff delay events occurring on the channels is explained. For the implementation of the concurrent backoff delay process on a single chip IEEE 802.15.4 hardware, the elements for the single channel backoff delay process and their control procedure are used to be extended to multiple channels with little modification. The medium access delay on each channel, which is available after execution of the concurrent backoff delay process, is displayed on the LCD of an IEEE 802.15.4 channel analyzer. The experimental results show that we can easily identify the interference on IEEE 802.15.4 channels caused by WiFi traffics in comparison with the way displaying measured channel powers.

Key Words : IEEE 802.15.4, WiFi, Interference, CSMA-CA, Concurrent backoff

1. 서 론

온도·습도와 같은 자연환경에 대한 감시, 축산업에서 동물의 감시, 지능형 건물을 위한 각종 센서 및 제어기, 가전 제품, 리모콘, 병원에서 환자의 상태를 감시하기 위한 간단한 센서 등등 소량의 데이터를 전송할 목적으로 다양한 형태의 센서네트워크의 운용이 증가하고 있다. 또한 이러한 센서네트워크에서 발생하는 데이터를 전송하는 기술로 중심 주파수가 2.4 GHz인 ISM (industrial, scientific, and medical) 대역에서 동작하는 IEEE 802.15.4 프로토콜이 널리 사용되고 있고 때때로 WiFi가 운용되고 있는 공간에 설치되기도 한다. 그런데 IEEE 802.15.4 네트워크의 노드들이 사용하는 채널의 주파수 스펙트럼이 WiFi 네트워크 채널의 주파수 스펙트럼과 서로 중첩되고 WiFi 네트워크의 트래픽 부하가 큰 경우 IEEE 802.15.4 노드들에게 간섭이 발생하여 IEEE 802.15.4 프로토콜을 이용하는 상위계층 응용에서 요구되는 조건을 만족시키기 어려운 문제가 발생한다[1-4]. 이러한 문제에 대한 사전적인 대처방법으로, WiFi 트래픽에 의해 간섭이 발생하는 IEEE 802.15.4 채널들을 피해서 IEEE 802.15.4 노드들을 동작시킬 수 있도록 IEEE 802.15.4 채널

들에 대한 분석이 필요하다.

WiFi에 의해 간섭이 발생하는 IEEE 802.15.4 채널을 탐색하는 방법으로는, WiFi 트래픽에 대한 주파수 스펙트럼이 다수의 인접한 IEEE 802.15.4 채널들과 중첩되기 때문에, WiFi 트래픽이 발생하는 동안 WiFi 스펙트럼과 중첩된 IEEE 채널들에서 동시에 채널전력이 증가한다[5]. 이러한 성질을 이용하여 다수의 IEEE 802.15.4 장치를 사용하고 각 장치의 주파수를 서로 인접한 IEEE 802.15.4 채널에 설정한 후, 채널전력의 측정 및 평가를 위해 각 IEEE 802.15.4 장치에서 RSSI (received signal strength indicator) 레지스터값을 샘플하여 채널전력의 변화가 여러 장치에서 동시에 동일한 패턴으로 나타나는지를 확인하거나 빈 채널이라고 판단되는 수준보다 커진 상태의 지속시간이 WiFi의 특정 프레임의 지속시간과 일치하는지를 확인하는 방법으로 IEEE 802.15.4 채널에 나타나는 WiFi의 간섭을 판단하는 방법이 연구되었다[6-7]. 그러나 이러한 방법들은 다수의 IEEE 802.15.4 장치가 필요한 문제가 있다. 한편, WiFi 네트워크의 액세스포인트 (access point)에서 주기적으로 비콘 (beacon) 이 발생하므로 IEEE 802.15.4 장치에서 IEEE 802.15.4 채널에 대한 RSSI 샘플 데이터를 CMF (common multiple folding) 알고리즘으로 신호처리하여 주기성을 발견할 때 그 채널에 WiFi의 간섭이 있다고 판단하는 방법은 신호처리를 위해 저장해야 하는 데이터의 양과 신호처리를 위한 프로세서의 부하가 요즈음 널리 사용되는 IEEE 802.15.4 프레임의 송수신을 지원하는 Texas Instrument사의 CC2530과 같은 단일칩 (single chip)으로 구현하기 어려운 문제가 있다[8].

따라서 이 논문에서는, 이와 같은 기존 연구의 문제점을 극복하면서 IEEE 802.15.4 프레임 송수신기능 하드웨어와 8051 코어 (core)를 포함하는 단일칩에 구현될 수 있는,

† Corresponding Author : Dept. of Information and Communication Eng., Hoseo Univ., Korea

E-mail : hjjin@hoseo.edu

* Dept. of Information and Communication Eng., Hoseo Univ., Korea

접수일자 : 2014년 4월 2일

수정일자 : 2014년 4월 17일

최종완료 : 2014년 5월 8일

IEEE 802.15.4 채널들에 나타나는 WiFi의 간섭을 분석하기 위한 장치의 구현에 대해 연구하였다. 이를 위해 2장에서 단일 채널 IEEE 802.15.4 CSMA-CA (carrier sense multiple access - collision avoidance) 알고리즘을 설명하고 이를 구현하는데 필요한 단일칩이 제공하는 요소들의 특성과 제어절차에 대해서 기술한다. 3장에서 다수의 IEEE 802.15.4 채널에 대한 병렬적인 백오프지연기법을 제시하고, 하나의 IEEE 802.15.4 하드웨어로 다중 채널에 대한 백오프지연과정이 실행될 수 있도록, 백오프지연 잔여시간을 이용하여 발생하는 모든 백오프지연 사건들을 순차적으로 정렬하여 처리하는 제어방법에 대해서 설명한다. 4장에서 병렬적인 백오프지연과정의 부수적인 결과로 얻어지는 다중 채널에 대한 매체접근지연을 LCD에 출력하여 보여주는 기능이 있는 IEEE 802.15.4의 채널에 대한 WiFi의 간섭 분석 장치의 구조를 설명하고, 시험 네트워크를 이용하여 측정된 결과를 채널전력을 나타내는 방식과 비교하고 5장에서 결론을 맺는다.

2. 단일 채널 IEEE 802.15.4 CSMA-CA 알고리즘의 구현

2.1 단일 채널 IEEE 802.15.4 CSMA-CA

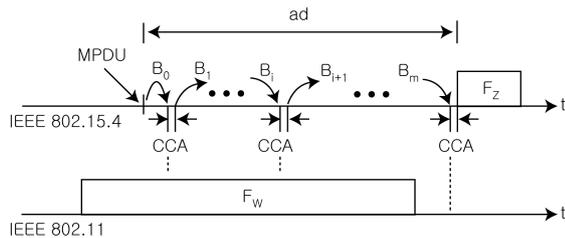


그림 1 IEEE 802.15.4 CSMA-CA 알고리즘
Fig. 1 IEEE 802.15.4 CSMA-CA algorithm

그림 1에 IEEE 802.11 네트워크와 공간과 주파수가 중첩된 환경에서 IEEE 802.15.4 네트워크의 노드에서 프레임 전송을 위한 CSMA-CA 알고리즘의 동작을 나타냈다. 여기서 F_z 와 F_w 는 각각 IEEE 802.15.4와 IEEE 802.11 프레임을 의미한다. IEEE 802.15.4의 CSMA-CA는 매체접근제어 계층 프로토콜에서 MPDU (MAC protocol data unit)가 생성되면 난수발생기로 백오프지연타이머 (backoff delay timer)를 초기화시키고 이 타이머가 감소되어 0이 될 때까지 지연한 후 CCA (clear channel assessment) 기능으로 무선채널이 비어있다고 판단되면 MPDU를 포함하는 프레임을 전송하고, 채널이 사용중인 경우에는 이전보다 더 넓은 범위에서 선택되는 난수만큼 다시 지연하는 과정을 무선채널이 사용가능하거나 규정된 최대차수까지 반복하는 방식이다[1]. 그런데 단일 최대차수의 백오프지연이 종료되는 시점에도 채널이 사용중이면 백오프지연실패로 판단한다. 백오프지연 타이머가 감소되는 동안 무선채널을 연속적으로 확인하지는 않고 백오프지연을 반복할 때마다 식 (1)과 같이 지수함수 형태로 증가하는 범위에서 선택된 난수 B_i 로 백오프지연타이머가 갱신된다.

$$B_i = RV(2^{BE_i} - 1) \quad (1)$$

여기서 i 는 백오프지연차수인데 범위가 $0 \leq i \leq N$ 인 정수를 갖고 N 은 최대백오프지연차수를 의미한다. 그리고 함수 $RV(x)$ 는 범위 $0 \leq i \leq x$ 에서 균일분포 특성을 갖는 정수형 난수로 결정된다. 식 (1)에서 지수항 BE_i 는 $i > 0$ 에 대해 식 (2)와 같이 결정된다.

$$BE_i = \min(BE_{i-1} + 1, macMaxBE) \quad (2)$$

여기서 $\min(a,b)$ 는 a, b 의 최소값이며 IEEE 802.15.4 표준의 MAC PIB (PAN information base)에 규정된 바와 같이 $BE_0 = macMinBE$ 이고 $macMinBE$ 와 $macMaxBE$ 는 각각 BE_i 의 최소값과 최대값을 의미한다.

한편, 그림 1에서는 차수 m 의 백오프지연을 거친 후에 무선채널이 비어있는 상황을 맞게 되는데, MPDU가 발생된 시각부터 이때까지 걸리는 시간 ad 를 매체접근지연 (medium access delay)으로 정의하고 이 논문에서는 백오프지연과정을 통해 측정된 매체접근지연을 WiFi 트래픽의 간섭을 분석하기 위한 수단으로 사용한다.

2.2 CSMA-CA 알고리즘 구현을 위한 IEEE 802.15.4 프레임 송수신기능 단일칩의 요소

앞 절에서 설명한 IEEE 802.15.4 CSMA-CA 알고리즘을 구현하기 위해서는 백오프지연의 크기를 결정하기 위한 난수발생기, 백오프지연을 위한 타이머, 채널의 사용여부를 판단하기 위한 CCA기능, 그리고 이들을 제어하기 위한 프로그램 코드가 실행될 프로세서가 필요하다. 한편, IEEE 802.15.4 CSMA-CA 알고리즘, 운영체제, 응용프로그램을 실행하는 무선센서네트워크 노드들의 구현에 2.4 GHz의 주파수 대역에서 동작하는 Texas Instrument사의 CC2430, CC2530, CC2420, CC2520 칩들이 널리 사용되어왔다. 이들 중 CC2420과 CC2520은 IEEE 802.15.4 프레임 송수신기능을 갖추고 있고, CC2430과 CC2530은 IEEE 802.15.4 프레임 송수신기능 외에도 추가로 CSMA-CA 알고리즘의 일부분과 응용프로그램의 실행에 사용될 수 있는 8051 코어를 포함하는 단일칩 형태이다. 센서네트워크에 관한 초기의 연구에서는 CC2430과 CC2420이 사용되었지만, CC2430은 반도체 제조사에서 이제 지원하지 않기 때문에, 여기에서는 CC2530을 사용하여 CSMA-CA 알고리즘을 구현할 때 필요한 각 요소들의 특성과 이들을 제어하기 위한 절차에 대해서 설명한다.

2.2.1 난수발생기

그림 2에 다항식으로 $X^{16} + X^{15} + X^2 + 1$ 과 같이 표현되는 구조의 LFSR (linear feedback shift register) 형태의 하드웨어를 이용하여 백오프지연과정에 사용되는 난수를 얻는 과정을 나타냈다. 그림 2에서 LFSR는 16비트 크기의 레지스터인데 각각 RNDL과 RNDH라는 이름을 통해서 하위 8비트와 상위 8비트 위치에 데이터를 쓰거나 읽을 수 있다. 그림 2에서 LFSR 하드웨어 난수발생기를 초기화 시키는데 사용되는 값도 불규칙하게 선택되도록 RF (라디오 주파수)

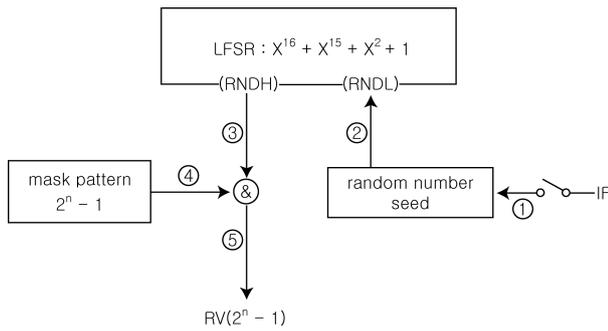


그림 2 백오프지연을 위한 난수의 생성
 Fig. 2 Random number generation for backoff delay

수신기를 켜고 RF 신호 수신기로부터 형성되는 경로상의 중간주파수 (IF : intermediate frequency)단에 나타나는 값을 ①과 같이 여러차례 샘플하여 준비한 후 이 값을 RNDL을 통해 ②와 같이 기록하는 방식으로 LFSR 하드웨어 난수발생기를 초기화한다. 한번 초기화된 난수발생기는 외부로부터 새로운 데이터의 유입이 없이도 LFSR에 클럭이 인가될 때마다 새로운 상태를 갖게 되는데 이 상태값은 균일분포를 갖는 난수의 특성을 보인다[9]. LFSR의 상태값인 난수는 ③과 같이 RNDH를 통해서 읽혀지고 ④에서 참조되는 비트마스크패턴 (bit mask pattern) $2^n - 1$ 과 비트단위 AND 연산을 거쳐서 ⑤와 같이 차수 i 의 백오프지연의 크기 $B_i = RV(2^n - 1)$ 이 결정되고, 여기서 $n = BE_i$ 이다.

2.2.2 타이머

백오프지연의 단위는 길이가 320 μs 인 단위슬롯이므로 백오프지연타이머는 320 μs 의 정수배에 해당되는 시간이 경과한 후에 종료되고 타이머 인터럽트 요청신호를 발생시킬 수 있어야 한다. 타이머 인터럽트 요청신호가 발생되면 인터럽트 서비스루틴에서 CCA기능을 이용하여 채널을 평가한 후 필요시 다시 백오프지연타이머를 새로운 값으로 갱신시키는 일을 수행한다.

이 논문에서는 CC2530의 동작을 위한 시스템클럭 (system clock)의 발생을 위해 32 MHz 크리스탈 (crystal)이 사용되었다. CC2530을 사용하는 “Z-stack”에서는 CSMA-CA 알고리즘이나 IEEE 802.15.4 프레임의 송수신과 관련된 정교한 타이밍을 위해 16비트 타이머 T2와 타이머 T2에서 오버플로우 (overflow)가 발생할 때마다 증가되는 24비트 오버플로우 카운터 및 비교기를 조합하여 사용하고, 응용프로그램에서는 가급적 다른 타이머들의 사용을 권장하고 있다[10]. 타이머 T1은 16비트, T3와 T4는 8비트 규모이고, 각 타이머는 32 MHz의 시스템클럭을 프로그램이 가능한 프리스케일러 (prescaler)를 통해 분주하여 사용할 수 있다. 타이머 T2와 연계된 오버플로우 카운터를 사용하는 방법이 타이머 제어 코드의 실행에 소요되는 프로세서의 부하를 가장 줄일 수 있는 방법이지만, 이 논문에서는 “Z-stack”과 같은 기존에 구현된 IEEE 802.15.4 프로토콜 코드와 자원사용의 충돌을 방지하기 위해 타이머 T1을 사용하여 백오프지연타이머 기능을 구현하였다. 타이머 T1에 1 μs 마다 클럭이 공급될 수 있도록 T1의 프리스케일러를 설정하고 난수발생기에서 생

성된 값에 단위슬롯의 크기 320을 곱한 값으로 초기화 시켜서 타이머를 동작시킨다. 예를 들어 생성된 난수가 7인 경우, $320 \mu s/slot * 7 slot = 2,240 \mu s$ 이므로 2,240을 타이머에 초기화 시키면 2,240 μs 후에 타이머가 종료될 때 미리 설정된 타이머 인터럽트 서비스루틴에서 채널에 대한 평가와 다시 백오프지연에 들어갈지를 결정한다.

이러한 방식으로 16비트 타이머를 이용할 경우 16비트 타이머의 최대값은 $2^{16} - 1$ 이므로 지원될 수 있는 백오프지연 최대크기는 $(2^{16} - 1) / 320 = 204.8$ 슬롯이다. 그런데 $2^7 - 1 < 204.8 < 2^8 - 1$ 이므로 어떤 차수에서 백오프지연의 최대 크기가 $\{2^3 - 1, 2^4 - 1, 2^5 - 1, 2^6 - 1, 2^7 - 1\}$ 인 경우에는 16비트 타이머만으로도 백오프지연동작이 구현가능하지만, $2^8 - 1$ 이상인 경우에는 204.8보다 큰 값이 발생할 수 있으므로 백오프지연차수가 더 큰 경우에는 추가적인 소프트웨어 처리가 요구되고 프로세서 코어의 부하가 증가하게 된다. 이 논문에서는 IEEE 802.15.4 표준에서 CSMA-CA 알고리즘의 백오프지연크기와 최대백오프지연차수를 결정하는 파라미터들 가운데서 macMinBE, macMaxBE, macMaxCSMABackoffs의 값을 각각 권고치인 3, 5, 4로 사용하므로 최대차수의 백오프지연에서 선택되는 난수의 최대치는 $2^7 - 1$ 이 되어 16비트 타이머만으로도 백오프지연 동작을 구현할 수 있다.

2.2.3 CCA 레지스터

무선채널에서 신호가 나타나면 채널전력이 증가되고 신호가 없는 경우에는 백색잡음 수준이 되므로, 채널에 나타나는 신호의 세기를 나타내는 RSSI 값은 채널에 어떤 신호가 나타나고 있는 동안 측정된 값은 신호가 없을 때에 비해서 크다. 그리고 채널에 신호가 나타나고 있는지를 구분하기 위한 기준치를 CCA_THR 레지스터에 미리 설정한 후 채널에 대해 측정된 RSSI 값이 기준치보다 클 경우 CCA 레지스터 비트가 셋 (set)되고 낮은 경우 리셋 (reset)되는 방식으로 동작하기 때문에, CSMA-CA 알고리즘에서 CCA기능은 백오프지연타이머가 종료될 때마다 CCA 레지스터의 해당 비트를 읽고 그 값에 따라서 채널의 사용여부를 판단한다. 한편, CCA_THR 레지스터에 설정된 기준치가 높으면 전력이 강한 신호가 채널에 나타나는 경우에 채널을 사용중으로 평가하지만, 설정된 기준치가 낮은 경우에 세기가 작은 신호에도 채널이 사용중으로 인식하므로 응용에 따라 적정값을 설정해야 한다.

2.3 단일 채널 CSMA-CA 기능의 구현을 위한 제어절차

이 논문에서는 위에서 설명한 난수발생기, 타이머, CCA 레지스터를 이용한 채널평가기능을 활용하여 IEEE 802.15.4 CSMA-CA 알고리즘의 백오프지연과정이 그림 3에 나타난 절차와 같이 구현되었다. 그림 3의 (a)에 백오프지연과정의 주프로그램에서 실행되는 함수들의 처리절차와 (b)에 백오프지연타이머가 종료되었을 때 발생하는 타이머 인터럽트 서비스루틴에서 처리절차를 나타냈고, 간략한 표현을 위해 변수의 형 (type) 선언, 타이머를 제어하기 위한 레지스터의 설정과정, 타이머 인터럽트 서비스루틴의 설정과정 등은 생략하였다. 여기에서 변수 i 는 백오프지연차수, N 은 백오프지

```
// maxRand[] = {7, 15, 31, 63, 127};
isFinished = FALSE;
i = 0;
V = RV(maxRand[i]);
ad = V;
loadTimerStart (V);
while (isFinished == FALSE) ;
```

(a)

```
if (CCA())
    isFinished = TRUE;
else {
    if (++i > N) isFinished = TRUE;
    else {
        V = RV(maxRand[i]);
        ad += V;
        loadTimerStart (V);
    }
}
```

(b)

그림 3 IEEE 802.15.4 CSMA-CA 제어절차
Fig. 3 IEEE 802.15.4 CSMA-CA control procedure

연최대차수, V 는 백오프지연크기, ad 는 매체접근지연, $isFinished$ 는 백오프지연과정의 종료여부를 의미한다. 배열 $maxRand[] = \{7, 15, 32, 63, 127\}$ 은 그림 2의 번호 ④에서 적용되는 것처럼 난수발생기에서 생성되는 각 백오프지연차수의 난수범위를 지정하는 역할이다. 그리고 함수 $RV(maxRand[i])$ 는 매개변수 $maxRand[i]$ 에 의해 지정된 범위에서 균일분포를 갖는 난수를 구하고, $loadTimerStart(V)$ 은 매개변수 V 에 기록된 값으로 백오프지연타이머 레지스터를 초기화시킨 후 타이머를 동작시키는 기능, $CCA()$ 는 채널의 사용중 여부를 판단하는 기능을 수행한다. 그림 3(a)에서는 백오프지연과정을 시작하면서 변수들을 초기화하고 차수 0의 백오프지연과정에 들어가서 타이머를 동작시킨 후 백오프지연과정의 종료가 확인될 때까지 “while ();”문에서 대기한다. 대기하는 동안 백오프지연에 사용된 타이머가 종료될 때마다 발생하는 타이머 인터럽트 서비스루틴 그림 3(b)에서는 먼저 $CCA()$ 기능으로 채널을 평가하고 채널이 비어있는 경우 명령문 “ $isFinished = TRUE;$ ”를 실행한다. 만일 채널이 사용중이고 아직 최대차수의 백오프지연을 실행하지 않은 경우에는 다음 차수의 백오프지연을 위한 난수를 발생시키고 매체접근지연을 갱신하고 타이머를 다시 동작시킨다. 그림 3(a)에서 명령문 “ $while (isFinished == FALSE);$ ”에서 대기중이던 주프로그램은 그림 3(b)에서 변수 $isFinished$ 의 값이 $TRUE$ 로 바뀌면 다음 단계로 진행되는데, 이때까지 누적되어온 변수 ad 의 값은 매체접근지연을 의미하게 되어 차후에 채널의 간섭이나 부하를 판단하는 근거로 참조하여 활용된다.

3. 다중 채널에 대한 병렬적인 백오프지연과정 구현

3.1 다중 채널에 대한 병렬적인 백오프지연과정

2장에서 설명한 IEEE 802.15.4 CSMA-CA 알고리즘의 실행시 각 차수마다 백오프지연타이머에 기록되는 백오프지연시간을 누적해나가면 백오프지연과정이 끝날 때 매체접근지

연을 얻게 되므로 이 방법을 다수의 IEEE 802.15.4 장치에서 각각 설정된 채널에 대해 적용하면 동일한 시간대에 각 채널들을 평가할 수 있는 매체접근지연을 얻을 수 있는데, 이 장에서는 하나의 IEEE 802.15.4 장치로 다수의 IEEE 802.15.4 채널들에 대해 동시에 병렬적으로 백오프지연하는 방법과 구현에 대해서 설명한다.

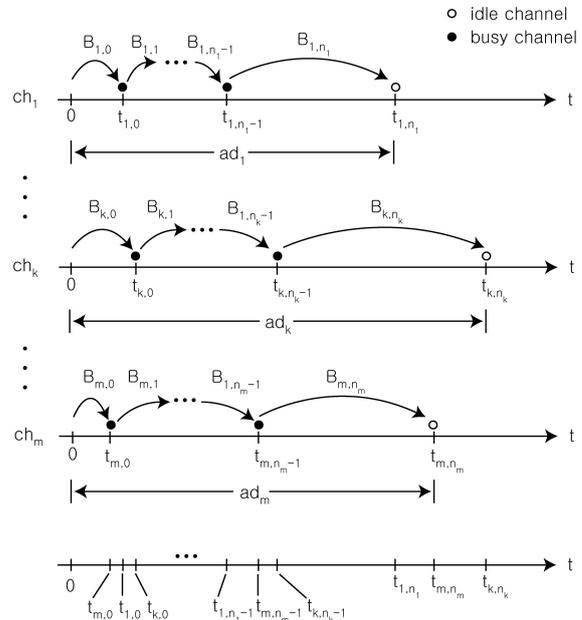


그림 4 다중 채널에 대한 병렬적인 백오프지연과정
Fig. 4 Concurrent backoff delay process on multiples channels

그림 4에 채널 ch_1 에서 ch_m 까지 m 개의 IEEE 802.15.4 채널들에서 시각 0에서 시작되는 CSMA-CA 알고리즘의 백오프지연과정을 나타냈다. 채널 ch_k 에서 n_k 를 백오프지연 종료시의 백오프지연차수라 할 때, $\{B_{k,0}, B_{k,1}, \dots, B_{k,n_k-1}, B_{k,n_k}\}$ 는 각 차수의 백오프지연 크기, $\{t_{k,0}, \dots, t_{k,n_k-1}, t_{k,n_k}\}$ 는 각 차수의 백오프지연 종료시각, ad_k 는 매체접근지연을 의미한다. 여기서 k 는 $1 \leq k \leq m$ 인 정수이다. 백오프지연과정이 진행되는 동안 각 채널에서는 각 차수의 백오프지연이 종료되는 시각마다 CCA 기능으로 채널이 비어있는지를 판단한다. 그림 4에 표현된 m 개의 채널에서 발생하는 백오프지연 종료시각은 채널별로 $\{t_{1,0}, \dots, t_{1,n_1-1}, t_{1,n_1}\}, \dots, \{t_{k,0}, \dots, t_{k,n_k-1}, t_{k,n_k}\}, \dots, \{t_{m,0}, \dots, t_{m,n_m-1}, t_{m,n_m}\}$ 과 같이 표현된다.

하나의 IEEE 802.15.4 장치로 그림 4에 표현된 m 개의 채널에 대해 병렬적으로 백오프지연과정을 수행하기 위해서는 나열된 m 개 채널의 백오프지연 종료시점마다 해당 채널로 주파수를 변경하고 CCA 기능으로 채널평가를 수행한다. 이것이 가능하기 위해서는 m 개 채널에서 발생하는 백오프지연 종료시점들이 발생하는 순서대로 정렬되어야 하고, 정렬된 각 종료시점마다 신속하게 채널을 변경해서 CCA 기능을 수행하고 다음 차례의 종료시점까지 타이머를 사용하여 지연할 수 있어야 한다. 그런데, 백오프지연과정은 채널의 미래 상태를 사전에 알 수 없고 백오프지연크기도 난수적으로

결정되기 때문에, 백오프지연과정이 종료될 때까지 발생될 수 있는 모든 차수의 백오프지연 종료시점들을 미리 알 수는 없다. 따라서 어떤 채널에서 한 차수의 백오프지연이 종료된 시점에 그 채널의 상태를 CCA기능으로 평가한 후 그 채널의 다음 차수 백오프지연 종료시점과 다른 채널들에서 진행중인 차수의 백오프지연 종료시점들을 오름차순으로 정리한다. 그리고 정리된 백오프지연 종료시점들 중 가장 빨리 도래하는 채널을 결정하고 시간이 경과하여 그 시점이 되면 IEEE 802.15.4 장치의 채널을 해당 채널의 주파수로 변경하고 CCA기능으로 그 채널을 평가하는 과정을 반복적으로 수행함으로써 다수의 채널에 대해 병렬적으로 백오프지연과정을 수행할 수 있게 된다.

3.2 백오프지연 잔여시간

3.1절에서 설명한 다수의 IEEE 802.15.4 채널에 대한 병렬적인 백오프지연과정을 하나의 IEEE 802.15.4 장치에 구현하기 위한 방법을 그림 5의 예를 이용하여 설명한다. 그림 5에 채널 ch_1 에서 ch_m 까지 m 개의 IEEE 802.15.4 채널들에서 백오프지연과정이 진행되고 있고 IEEE 802.15.4 장치 프로세서 코어의 타이머에서는 병렬적으로 진행되는 채널에서 발생하는 모든 백오프지연의 종료시점마다 타이머 인터럽트 요청신호가 발생하도록 타이머의 제어레지스터를 설정한다. 그림 5에서 표현된 임의의 백오프지연 종료시각, 예를 들어 채널 ch_m 에서 크기가 $B_{m,r-1}$ 인 차수 $r-1$ 의 백오프지연이 종료되는 시각 $t_{m,r-1}$ 에 타이머 인터럽트 요청신호가 발생하면, 타이머 인터럽트 서비스루틴에서는 CCA기능으로 채널 ch_m 이 비어있는지 평가하는데 그림 5에서는 사용중으로 평가되므로 난수를 발생시켜 크기가 $B_{m,r}$ 인 백오프지연의 종료시각 $t_{m,r}$ 이 새롭게 결정된다. 그리고 시각 $t_{m,r-1}$ 에서 채널 ch_1, ch_k 에 각각 크기가 $B_{1,p}, B_{k,q}$ 이고 차수 p, q 인 백오프지연이 진행중이므로 종료시각 $t_{1,p}, t_{k,q}$ 가 이미 결정되어 있는 상황이다. 시각 $t_{m,r-1}$ 에서 채널 ch_m 에 대해 CCA기능을 수행한 후에, 이미 결정되어 있는 백오프지연 종료시각들 $t_{1,p}, t_{k,q}, t_{m,r}$ 중 가장 먼저 도래하는 시각 $t_{k,q}$

에 다시 타이머 인터럽트 요청신호가 발생되도록 타이머에 새롭게 기록되어야 할 시간 V_n 를 식 (3)과 같은 방식으로 결정한다.

$$V_n = \min\{rb_1, rb_k, rb_m\} \tag{3}$$

여기서 $\min\{ \}$ 은 괄호 안의 원소들 중 최소를 의미하고, rb_1, rb_k, rb_m 는 시각 $t_{m,r-1}$ 에 이미 결정되어 있는 백오프지연의 종료시각 $t_{1,p}, t_{k,q}, t_{m,r}$ 까지 잔여시간을 의미한다. 각 채널에서 잔여시간은 새로운 차수의 백오프지연이 결정될 때마다 난수발생기에 의해 생성된 백오프지연의 크기로 초기화되고, 다음 차례의 채널평가를 위한 최소잔여시간 V_n 이 결정된 후에 유지되고 있던 잔여시간에서 V_n 을 뺀다. 그림 5의 예에서는 시각 $t_{m,r-1}$ 에 채널 ch_m 에서 차수 r 의 지연이 새롭게 결정되므로 채널 ch_m 의 잔여시간은 $rb_m = B_{m,r}$ 로 초기화된 후, 타이머에 기록할 최소잔여시간이 식 (3)과 같이 V_n 으로 결정되면, 타이머에 의해 V_n 만큼의 시간이 경과한 후의 시각 $t_{k,q}$ 에 다시 최소잔여시간을 구하는데 사용할 채널들의 잔여시간을 $rb_1 = rb_1 - V_n, rb_k = rb_k - V_n = V_n - V_n = 0, rb_m = rb_m - V_n$ 으로 갱신한다.

어떤 채널에 대한 백오프지연과정의 종료조건을 CCA기능에 의해 채널이 비어있다고 평가되는 경우 또는 최대백오프지연차수에 이른 경우라 할 때, 위에서 설명한 방법을 타이머가 종료될 때마다 발생하는 인터럽트 서비스루틴에서 실행하고 타이머에 기록되는 값 $\{V_0, \dots, V_{n-1}, V_n, \dots\}$ 을 누적해나가면서 각 채널마다 종료조건이 발생했을 때 그동안 누적된 시간을 별도의 메모리에 저장하면, 병렬적인 백오프지연과정이 진행되던 채널들 중 마지막으로 종료조건에 도달하는 순간 별도의 메모리에 저장된 값들은 각 채널에 대한 매체접근지연을 의미하고 WiFi 간섭을 평가하기 위한 IEEE 802.15.4 채널 분석에 활용할 수 있게 된다.

3.3 병렬적인 백오프지연과정의 구현을 위한 제어절차

이 절에서는 앞에서 설명한 백오프지연 잔여시간을 활용하여 하나의 IEEE 802.15.4 장치로 다수 채널에 대한 병렬적인 백오프지연과정을 구현하기 위한 주요함수들과 제어절차에 대해서 설명한다. 그림 6에 이를 위한 주요함수들과 제어절차들을 나타냈는데, 2장에서 설명했던 것과 동일한 난수발생기, 타이머, CCA기능이 필요하므로 이들의 사용방식에 대해서는 변경하지 않고 제어절차의 구조를 거의 그대로 유지하고 다수의 채널로 확장할 때 필요한 부분이 추가되는 형태로 표현되었다. 단일 채널 백오프지연과정과는 달리 다중 채널에 대해 병렬적인 백오프지연과정을 수행하기 때문에 백오프지연과정이 전체 채널에서 모두 종료될 때까지 각 채널에 대한 백오프지연과정의 진행상태를 나타내는, 그림 7에 표현된 구조체 `channel_status`의 구성요소들을 포함하는, 변수들을 유지해야 한다.

구조체 `channel_status`의 구성요소들 중 i 는 백오프지연 차수, rb 는 차수 i 의 백오프지연종료시점까지 잔여시간, st 는 백오프지연과정 종료여부 표시, ad 는 매체접근지연을 의미한다. 또한 2장에서 설명한 단일 채널 백오프지연과정의 구

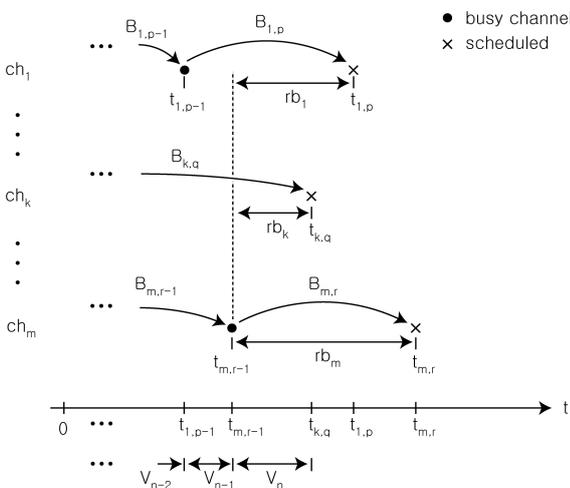


그림 5 잔여시간을 이용한 타이머값의 결정
Fig. 5 Decision of timer values using remaining times

```

isFinished = FALSE;
init_rb_st (); // ch[k].i, ch[k].rb, ch[k].st
min_rb (); // set Vn and k
gTime = Vn; // global time
update_rb ();
loadTimerStart (Vn);
while (isFinished == FALSE) ;
    
```

(a)

```

setChannel (k);
if (CCA())
    ch[k].st = FINISHED;
else if (ch[k].++i > N) // MAX_BACKOFF?
    ch[k].st = FINISHED;
else
    ch[k].rb = RV(maxRand[ch[k].i]);

if (ch[k].st == FINISHED)
    ch[k].ad = gTime; // ch[k]'s MAC delay

if (allChannelDone ())
    isFinished = TRUE;
else {
    min_rb ();
    gTime += Vn;
    update_rb ();
    loadTimerStart (Vn);
}
    
```

(b)

그림 6 병렬적인 백오프지연과정 제어절차
Fig. 6 Control procedure for concurrent backoff delay process

```

struct channel_status {
    unsigned char i;
    unsigned char rb;
    unsigned char st;
    unsigned char ad;
};
    
```

그림 7 구조체 channel_structure
Fig. 7 Structure channel_structure

현과 비교할 때 함수 `init_rb_st ()`, `min_rb ()`, `update_rb ()`, `setChannel ()`, `allChannelDone ()`이 추가되었다. 여기서 `init_rb_st ()`는 동시에 백오프과정이 진행되는 모든 채널에 대해 구조체 `channel_status`의 구성요소들을 초기화 시키고, `min_rb ()`는 백오프지연과정이 아직 종료되지 않은 채널들에서 진행중인 차수의 종료시점까지 잔여시간이 최소인 채널의 채널참조번호 `k`와 잔여시간 `Vn`를 결정한다. 그리고 `update_rb ()`는 `min_rb ()`에서 결정된 `Vn`값을 매개변수로 하는 `loadTimerStart (Vn)`를 실행하기 전에 백오프과정이 진행중이 모든 채널의 잔여시간을 `Vn`씩 감소시키는 기능을, `setChannel (k)`은 `k`가 가리키는 채널 `ch[k]`로 채널을 변경하는 기능을, `allChannelDone ()`은 모든 채널에서 백오프지연과정이 종료되었는지를 확인하는 기능을 수행한다. 그림 6(a)에서는 병렬적인 백오프지연과정을 실행할 때 필요한 초기화과정과 최초로 백오프지연될 채널의 참조번호 `k`와 지연값 `Vn`를 결정하여 타이머에 기록하고 동작시킨 후 모든 채널에서 병렬적인 백오프지연과정이 종료될 때까지 대기하

는 절차를 나타냈는데, “while ();”문의 조건이 참이 되면 대기과정이 끝나고 병렬적인 백오프지연과정이 종료된다. 이후에 백오프지연과정이 수행된 채널들에 대한 매체접근지연은 채널 `ch[k]`에 대해 변수 `ch[k].ad`를 통해 참조할 수 있다. 그림 6(b)에 타이머 인터럽트 서비스루틴에서 수행되는 일들을 나타냈다. 인터럽트가 발생하면 이와 연관된 채널참조번호 `k`를 매개변수로 하는 함수 `setChannel (k)`를 이용하여 IEEE 802.15.4 채널을 `ch[k]`로 변경한 후 함수 `CCA ()`를 이용하여 채널을 평가한다. 함수 `CCA ()`의 결과에 따라 채널이 비어있으면 그 채널에 대한 탐색을 종료하고, 사용중이면 최대차수의 백오프지연에 도달하지 않은 경우에 새로운 난수로 채널 `ch[k]`의 잔여시간을 갱신한다. 그리고 함수 `allChannelDone ()`을 사용하여 모든 채널에 대해 백오프지연과정이 종료되었다고 판정되기 전까지는 계속해서 함수 `min_rb ()`를 이용하여 잔여시간이 최소인 채널을 찾은 후 타이머를 다시 동작시킴으로써 예약된 다음 차례의 인터럽트 서비스를 설정하는 과정을 반복한다. 한편, 인터럽트 서비스루틴에서 채널 `ch[k]`가 비어있다고 평가되거나 최대차수의 백오프지연에 이른 경우에는 채널 `ch[k]`에 대한 매체접근지연이 변수 `ch[k].ad`에 기록되는데 이 값은 그림 6(a)의 “while ();”문 이후에 이용할 수 있다.

4. WiFi 간섭을 평가하기 위한 IEEE 802.15.4 채널분석기의 구현 및 시험

그림 8에 3장에서 설명한 병렬적인 백오프지연 방법으로 구현 다수의 IEEE 802.15.4 채널에 대한 매체접근지연을 실시간으로 액정표시장치 (LCD)에 출력함으로써 WiFi 트래픽으로 인해 IEEE 802.15.4 채널들에 나타나는 전송지연을 평가하고 WiFi 트래픽의 간섭여부를 판단하는데 사용될 수 있는 IEEE 802.15.4 채널분석기의 구조를 나타냈다. 이 장치는 IEEE 802.15.4 프레임의 송수신기능 하드웨어와 프로토콜 및 응용프로그램을 구현할 수 있는 8051 프로세서 코어를 포함하는 단일칩 CC2530가 부착되어 있는 CC2530EM 보드를 입력 스위치, 출력 LCD, 외부 호스트 (host) 컴퓨터와

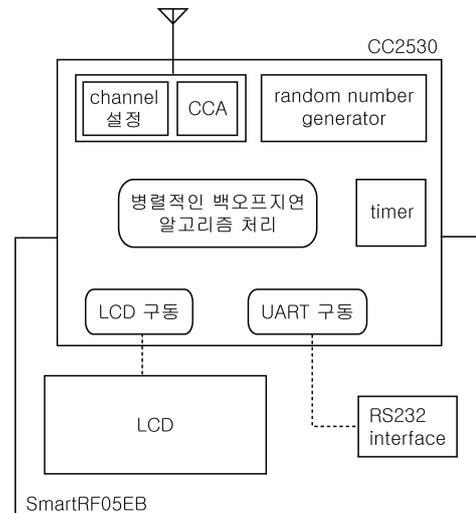


그림 8 IEEE 802.15.4 채널분석기의 구조
Fig. 8 The structure of IEEE 802.15.4 channel analyzer

통신을 위한 RS232 인터페이스를 제공하는 SmartRF05EB 보드에 연결시킨 형태로 구현되었다[11]. CC2530 내부에서는 병렬적인 백오프지연 알고리즘의 구현을 위해 채널설정을 위한 레지스터, CCA 평가를 위한 레지스터, 백오프지연의 크기를 결정하기 위한 난수발생기, 백오프지연 크기만큼 시간지연을 위한 타이머와 같은 요소들이 사용된다. 병렬적인 백오프지연 알고리즘의 처리결과 부수적으로 얻는 채널 $ch[k]$ 에 대한 매체접근지연이 그림 6에서 변수 $ch[k].ad$ 에 저장되므로, LCD 구동부에서는 이 값을 LCD의 출력범위에 맞추어 막대그래프로 나타낸다. 그리고 이와 동시에 LCD에 출력된 매체접근지연은 사후처리 및 분석을 위해 UART (universal asynchronous receiver and transmitter) 구동부를 통해 외부 호스트 컴퓨터로 전송된다.

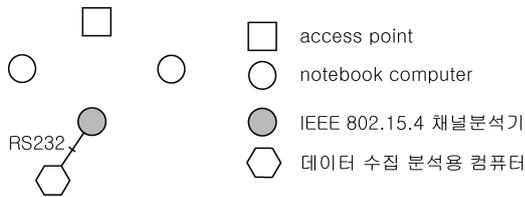


그림 9 시험 시스템
Fig. 9 Test system

그림 9에 이 논문에서 구현한 WiFi 트래픽의 간섭을 분석하기 위한 IEEE 802.15.4 채널분석기에 대한 시험 시스템을 나타냈다. 이 시스템은 노트북컴퓨터와 액세스포인트로 구성된 WiFi 네트워크에서 WiFi 트래픽이 발생되고 그림 8과 같은 구조의 IEEE 802.15.4 채널분석기와 사후처리와 분석을 위해 RS232 포트를 통해 IEEE 802.15.4 채널들에 대한 매체접근지연을 수집하는 호스트 컴퓨터로 구성되어 있다. 실험을 위해 WiFi 채널 8을 사용하도록 액세스포인트를 설정하고 노트북컴퓨터에서는 유무선 네트워크에 대한 벤치마킹 도구인 “Nuts About Nets, LLC”의 “netstress” 프로그램을 사용하여 MTU (maximum transfer unit) 크기를 1,500 byte, 초당 발생되는 패킷수를 1,016로 설정한 상태에서 WiFi 트래픽을 발생시켰다.

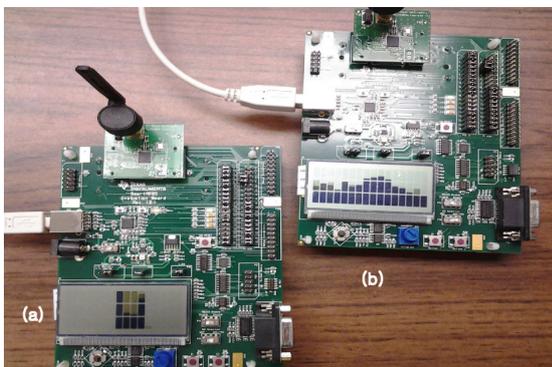


그림 10 시험중인 IEEE 802.15.4 채널분석기
Fig. 10 IEEE 802.15.4 channel analyzer under test

그림 10(a)에 이와 같은 실험환경에서 동작중인 IEEE 802.15.4 채널분석기의 사진을 나타냈는데, LCD에는 좌측에

서 우측방향으로 IEEE 802.15.4 채널 11에서부터 26까지 매체접근지연이 막대그래프로 표시된다. LCD에서 가운데서로 인접한 네 개의 기둥은 IEEE 802.15.4 채널 18, 19, 20, 21에서 매체접근지연이 크게 나타나는 것을 의미하는데, 이것은 이들 채널들과 주파수 스펙트럼이 중첩되는 WiFi 채널 8에서 부하가 큰 시험용 WiFi 트래픽이 발생하고 있기 때문이다. 한편, 그림 10(b)에는 동일한 실험환경에서 RSSI를 이용하여 IEEE 802.15.4의 채널전력을 측정하여 LCD에 출력하도록 구현된 것을 나타냈는데, IEEE 802.15.4의 채널 18, 19, 20, 21에서 비교적 높은 채널전력이 나타나고 있고 이들 채널에서부터 멀어질수록 채널전력이 작게 나타나는 것으로 관찰된다. 그림 10(a)와 (b)의 결과를 볼 때, 병렬적인 백오프지연방식으로 구현 매체접근지연을 활용한 그림 10(a)의 IEEE 802.15.4 채널분석기는 트래픽 부하가 큰 WiFi의 간섭이 발생하는 채널들과 그렇지 않은 채널들 사이에 매체접근지연의 차이가 크고 그림 10(b)와 같이 채널전력을 관찰하는 방식에 비해서 간섭이 발생하는 채널을 상대적으로 쉽게 구분할 수 있는 특성을 보인다.

5. 결 론

이 논문에서는 WiFi 트래픽으로 인해 IEEE 802.15.4 노드들에게 발생하는 전송지연을 분석하고 간섭여부를 판단하는데 사용될 수 있는 IEEE 802.15.4 채널분석기를 하나의 고주파 하드웨어와 8051 코어를 갖는 단일칩에 구현한 연구에 대해서 기술하였다. WiFi 트래픽이 IEEE 802.15.4 노드의 프레임 송수신에 영향을 주는 간섭을 판단하기 위해 IEEE 802.15.4 채널에 대해 RSSI값을 이용하여 채널전력을 측정하여 신호처리하거나 증가된 채널전력이 WiFi의 특정한 프레임의 지속시간과 일치하는 현상이 일어나는지를 확인하는 방식의 기존연구에 대해서 분석하였다. 이러한 기존의 연구는 WiFi 채널의 주파수 스펙트럼이 서로 인접한 다수의 IEEE 802.15.4 채널의 스펙트럼과 중첩되므로 다수의 IEEE 802.15.4 하드웨어를 사용하여 인접한 다수의 채널에서 동일한 채널특성이나 시간특성이 각각의 하드웨어에서 동시에 발생하는지를 확인함으로써 WiFi 트래픽의 발생 여부를 판단하는 방식인데, 다수의 IEEE 802.15.4 하드웨어가 필요하고 채널전력을 샘플하여 신호처리를 하는 프로세서의 부담이 증가하는 문제점이 있다.

이 논문에서는 이러한 문제점을 개선하여, 채널전력을 샘플하여 신호처리하는 기존의 방식과 달리 채널에 다른 트래픽이 증가할 때 매체접근지연이 증가하는 특성이 있으므로, 채널에 대한 매체접근지연을 간섭에 대한 평가요소로 활용하고 하나의 IEEE 802.15.4 하드웨어로 다수의 채널에 대해 병렬적으로 백오프지연과정을 수행하여 각 채널에 대한 매체접근지연을 동시에 구하는 방법에 대해서 설명하였다. IEEE 802.15.4 하드웨어를 포함하는 단일칩 CC2530에 단일 채널 백오프지연과정의 구현에 필요한 기능들 즉, CCA기능, 난수발생기, 타이머와 이를 제어하는 프로그램의 절차에 대해서 설명하였다. 그리고 이를 확장하여 다중 채널에 대한 병렬적인 백오프지연과정의 구현을 위해, 추가로 필요한 기능들과 제어프로그램의 절차와 부수적으로 연계 되는 각 채널에 대한 매체접근지연을 사용하는 방법에 대해서 기술하

였다. 단일칩 CC2530과 LCD 출력장치, RS232 인터페이스를 제공하는 SmartRF05EB 보드에 구현한 WiFi의 간섭을 분석하기 위한 IEEE 802.15.4 채널분석기의 구조에 대해서 기술하고, 이를 시험하기 위한 WiFi 네트워크와 IEEE 802.15.4 채널분석기로 구성된 시스템과 시험조건에 대해서 기술하였다. LCD 출력장치를 통해 관찰된 실험결과는 이 논문에서 구현된 IEEE 802.15.4 채널분석기는 채널전력을 측정하여 보여주는 방식에 비해 WiFi의 간섭이 발생하는 IEEE 802.15.4 채널들을 쉽게 구분할 수 있음을 보여준다.

앞으로 이 논문에서 연구된 결과들을 활용하거나 연관된 다음과 같은 분야에 대해서 연구가 이루어져야 한다. 병렬적인 백오프지연방법을 이용하면 다중 채널에서 발생하는 매체접근지연을 구할 수 있는데, 나중에 WiFi의 간섭이 발생하는 IEEE 802.15.4 채널을 탐색하는데 사용하기 위해서는 이 값들을 여러차례 구해서 별도의 저장공간에 보관해야 하므로 메모리가 추가로 필요하다. 그리고 이를 바탕으로 WiFi의 간섭을 피하기 위한 IEEE 802.15.4 채널을 자동적으로 탐색하기 위해서 IEEE 802.15.4 노드의 관점에서 WiFi 트래픽이 나타나고 있다는 것을 추정하기 위한 프로세서 코어의 부하가 증가한다. 따라서 WiFi 간섭이 발생하는 IEEE 802.15.4 채널들을 실시간으로 탐색할 수 있는 적절한 메모리의 요구조건, 연산량을 줄일 수 있는 구현방안, 8051 프로세서 코어 외에 다른 다양한 성능의 프로세서 코어로 구현하는 방안에 대해서 연구가 필요하다.

References

[1] LAN MAN Standards Committee of the IEEE Computer Society, Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low Rate Wireless Personal Area Networks (WPANs), IEEE Std 802.15.4-2006, Sep. 8, 2006.

[2] LAN MAN Standards Committee of the IEEE Computer Society, Part 11 : Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, ANSI/IEEE Std 802.11-2012 (Revision of IEEE Std 802.11-2007), Mar. 2012.

[3] ZigBee Alliance, "ZigBee and Wireless Frequency Coexistence", ZigBee White Paper, Jun. 2007.

[4] P. Yi, A. Iwayemi, and C. Zhou, "Developing ZigBee Deployment Guideline Under WiFi Interference for Smart Grid Applications", IEEE Trans. on Smart Grid, vol. 2, no. 1, pp. 110-120, Mar. 2011.

[5] G. Thonet, P. Allard-Jacquín, and P. Colle, "ZigBee-WiFi Coexistence", Schneider Electric White Paper and Test Report, pp. 1-38, Apr. 2008.

[6] R. Xu, G. Shi, J. Luo, Z. Zhao, and Y. Shu, "MuZi : Multi-channel ZigBee Networks for Avoiding WiFi Interference", Proc. of 4th International Conference on Cyber, Physical and Social Computing (CPSCom), pp. 323-329, Oct. 2011.

[7] J. Ansari, T. Ang, and P. Mähönen, "WiSpot : Fast

and Reliable Detection of Wi-Fi Networks using IEEE 802.15.4 Radios", Proc. of MobiWac'11, pp. 35-44, Nov. 2011.

[8] R. Zhou , Y. Xiong , G. Xing , and L. Sun, "ZiFi : Wireless LAN Discovery via ZigBee Interference Signatures", Proc. of MobiCom'10, pp.49-60, Sep. 2010.

[9] Texas Instrument, CC253x System-on-Chip Solution for 2.4-GHz IEEE 802.15.4 and ZigBee Applications : User's Guide, 2012.

[10] HAL Drivers Application Programming Interface, Document Number : SWRA193 Version 1.4.

[11] Texas Instrument, Z-Stack User's Guide for SmartRF05EB and CC2530, Document Number : SWRU189 Ver. 1.6.

저 자 소 개



송 명 렬 (宋 明 烈)

1985년 연세대학교 전자공학과 (BS), 1988년 연세대학교 대학원 전자공학과 (MS), 1996년 연세대학교 대학원 전자공학과 (Ph.D), 1988년-1991년 육군사관학교 전자과 전임강사, 1996년-현재 호서대학교 정보통신공학과 교수



진 현 준 (陳 鉉 竣)

1984년 고려대학교 전자공학과 (BS), 1986년 고려대학교 대학원 전자공학과 (MS), 1997년 (미) Lehigh Univ. 전산학과 (Ph.D), 1986년-1991년 삼성전자 시스템개발실, 1998년-현재 호서대학교 정보통신공학과 교수