

# 에너지 효율적인 홈 네트워크를 위한 트래픽 기반 전력 절감 모드 결정 알고리즘의 설계 및 성능 분석

공인엽<sup>†</sup>, 황원주<sup>\*\*</sup>

## 요 약

홈 게이트웨이는 유비쿼터스 홈 서비스를 지속적으로 제공하기 위해 24시간 가동되므로 연간 소모하는 전력의 규모가 크다. 네트워크의 유휴 시간에 최소 전력을 사용하는 전력 절감 모드를 적용하면 이러한 전력 소모를 줄일 수 있다. 본 논문의 전력 절감 모드 결정 알고리즘은 유휴 시간이 시작되면 유휴 시간의 누적된 정보를 토대로 현재의 유휴 시간의 길이를 예측하고, 그 결과에 따라 전력 절감 모드로 전환한다. 전력 절감 모드에서 간단한 제어용 프로토콜 데이터는 활성 모드로의 전환 없이 프록시로 처리한다. 그리고 사용자 트래픽이 존재할 경우에만 활성 모드로 전환한다. 이러한 알고리즘을 검증하기 위해서는 홈 네트워크에서 트래픽 데이터를 수집하여, 시뮬레이션을 수행하였다. 그 결과 기존 방식에 비해 최소 14%에서 최대 49%의 전력 절감 효과를 보였다.

## Design and Performance Analysis of a Traffic-based Power Saving Mode Decision Algorithm for Energy-efficient Home Networks

In Yeup Kong<sup>†</sup>, Won Joo Hwang<sup>\*\*</sup>

## ABSTRACT

Home gateway is always full-powered for ubiquitous home services, and consumes much energy yearly. Power-saving algorithm to conserve this energy must reduce the energy consumption and preserve always-on services. Our algorithm predicts current idle period using the history of the past idle period when the idle period starts, and then determines whether the power mode is changed to the saving mode or not. On the power saving mode, it processes the simple protocol data for network control using proxying with no wakeup. And it changes the power mode to active mode when user's traffic exists. As the results of the simulation using real traffic, our algorithm saves the energy consumption from 14 % to 49 % as compared with existing method.

**Key words:** Sleep/Wakeup, Power Control (전력 제어), Home Gateway (홈 게이트웨이)

## 1. 서 론

홈 게이트웨이는 유비쿼터스 홈 서비스를 지속적으로 제공하기 위해서 항상 활성 모드의 최대 전력을 소비한다. 점차 홈 네트워크가 확산되는 추세에서 홈

게이트웨이가 연간 소비하는 전력량도 기하급수적으로 증가한다. 이에 지식경제부에서는 '대기전력이 1W 이하인 홈 네트워크 게이트웨이, 에너지 소비량이 기존 대비 절반인 적외선 건조기, 신냉매를 활용한 고효율 가스 냉난방 기기, 유해 배기 가스 배출량

※ 교신저자(Corresponding Author) : 황원주, 주소 : 경남 김해시 어방동(621-749), 전화 : (055)320-3847, FAX : (055) 322-6275, E-mail : ichwang@inje.ac.kr

접수일 : 2008년 3월 31일, 완료일 : 2008년 8월 26일

<sup>†</sup> 정회원, 금오공과대학교 전자공학부 전임강사

(E-mail : iykong@kumoh.ac.kr)

<sup>\*\*</sup> 정회원, 인제대학교 정보통신공학과 조교수

※ 본 연구는 2007년 인제대학교 박사후연수과정 지원 사업에 의한 것임

을 대폭 줄인 산업용 보일러 등이 개발될 전망이다.'라고 발표하였다[1]. 이러한 추세는 국내뿐만 아니라 국제적으로도 이슈가 되고 있다[2].

에너지 절약을 통한 환경 보호를 위해서는 홈 네트워크 서비스가 활성화되지 않은 시간에는 전력을 최소화한 대기 전력 모드를 사용하는 방식을 적용해야 한다. 즉, 네트워크의 유휴 시간 동안에는 홈 게이트웨이의 불필요한 전원 소모를 제거하고, 서비스가 활성화되었을 때 홈 게이트웨이가 활성 모드의 전원을 공급받아 동작하는 방식을 적용할 필요가 있다. 이러한 방식에서는 전력 절감 모드로의 전환을 결정하는 기준이 전체 시스템의 전력 효율을 결정한다. 전력 절감 모드가 유지되는 시간이 짧게 될 경우, 모드 전환에 소요되는 전력과 지연 시간의 부하가 상대적으로 커지기 때문이다.

홈 네트워크에서 사용되는 전력 절감 방식은 각 정보 단말이 사용되지 않을 때 대기 전력으로 전환하는 것에만 주로 초점이 맞춰져 있다. 반면 홈 게이트웨이에서 전력 절감 모드를 결정하는 방식에 대해 연구한 사례로는 두 가지 접근 방법이 있다. 첫 번째 방법은 상용화된 것으로서, 미리 지정한 시간동안 사용되지 않을 때 전력 절감 모드로 전환하고, 사용자 이벤트가 있으면 무조건 활성 모드로 전환하는 간단한 방식이지만 네트워크 트래픽의 변화를 반영할 수 없다. 두 번째 방법은 논문 연구 단계로서, 과거에 기록된 유휴 시간을 기준으로 현재의 유휴 시간을 예상하여 전력 절감 효과가 있다고 판단되는 경우에만 전력 절감 모드로 전환한다[3,4]. 하지만 이 방식은 트래픽의 변화를 반영할 수 있으나 전력 절감 모드로의 전환에 소모되는 지연 오버헤드가 1초 이상인 경우 비효율적이다. 본 알고리즘의 적용 대상은 전력 절감 모드 전환에 필요한 지연 오버헤드가 2초이므로 두 번째 방식은 비효율적이다.

먼저 본 논문에서는 가변적인 네트워크 트래픽의 특성을 반영하기 위해 유휴 시간의 패턴을 고려한 두 번째 방식을 적용하였다. 더불어 전력 절감 효율을 보완하기 위해서 전력 절감 모드에서는 간단한 제어용 프로토콜이나 연결 확인용 응용 프로토콜 트래픽까지 최대한 프록시로 처리하도록 설계하였다. 이를 통해 전력 절감 모드의 유지 시간을 확보함으로써 전력 절감 효율을 높였다.

이에 본 연구에서는 유휴 시간의 특성을 고려하여

전력 절감 모드를 사용하되 프록시 처리를 최대화함으로써, 전력 절감 효과를 높인 알고리즘을 도출하였다. 그리고 이를 검증하기 위해 홈 네트워크로부터 트래픽을 수집하여 시뮬레이션으로 수행하고 성능을 평가하였다.

본 논문의 구성은 다음과 같다. 2장에서는 기존의 네트워크 트래픽 기반 전력 절감 방법에 대해 설명하고, 3장에서는 전력 절감 모드 결정 알고리즘의 설계에 대해 제안한다. 그리고 4장에서 제안 알고리즘의 구현 및 시뮬레이션 테스트, 그리고 성능 측정에 대해 상세히 설명하고, 마지막으로 5장에서는 결론과 향후 과제로 마무리한다.

## 2. 네트워크 트래픽 기반 전력 절감 방법

홈 네트워크의 트래픽은 이용 패턴이 불규칙적이다. 따라서 네트워크에 연결되는 정보 가전뿐만 아니라 홈 게이트웨이가 불규칙적인 네트워크 트래픽의 변화에 따라 전력 모드를 적절히 전환해야 한다. 단, 전력 절감 방식을 사용하더라도 홈 네트워크로의 연결성과 서비스를 지속적으로 유지해야 한다[5]. 무선 네트워크에 비해서 상대적으로 유선 네트워크 환경의 홈 네트워크를 고려한 전력 절감 방식에 대한 연구는 많지 않은 편이다. 이와 관련된 연구로는 네트워크 트래픽을 기반으로 유휴 시간을 예측하는 방식[3,4], 서비스 기반의 홈 서버 전력 관리 방식[6], 홈 네트워크를 구성하는 장치와 24시간 이용 패턴을 고려한 전력 절감 방식[7], 전력 절감과 서비스 품질 보장을 위한 미들웨어 구조[8] 등이 제안되었다.

이 중에서 본 논문에서는 과거의 유휴 시간을 추적하여 기록하고, 다음 유휴 시간을 예측하는 방식을 기반으로 한 전력 절감 방식을 고려하였다.

### 2.1 Exponentially Weighted Moving Average (EWMA) 또는 Exponential Smoothing 방식

이는 운영 체제에서 CPU를 스케줄링하는 방식에 적용된 것으로서, 운영 체제가 프로세스 스케줄링을 적절하게 하기 위해서 다음 CPU 버스트의 길이를 예측해야 한다[3]. 이를 위해 이전까지의 예측 값을 누적한 평균과 직전 결과 값을 적절한 비율로 반영하여 다음 CPU 버스트의 길이를 예측한다.

이를 재귀식으로 표현하면 식 (1)과 같다.

식 (1)에서  $I_{n+1}$ 은 새로운 예측 값이고,  $I_n$ 은 이전까지의 예측 평균값이며,  $i_n$ 은 가장 최근의 CPU 버스트 값이다.

$$\begin{aligned}
 I_{n+1} &= a \cdot i_n + (1-a) \cdot I_n \\
 &= a \cdot i_n + a \cdot (1-a) \cdot I_{n-1} + a \cdot (1-a)^2 \cdot I_{n-2} \\
 &\quad + \dots + a \cdot (1-a)^n \cdot i_0 + a \cdot (1-a)^{n+1} \cdot I_0
 \end{aligned}
 \tag{1}$$

$a$ 는 어느 쪽에 비중을 얼마나 둘 것인가에 관한 가중치로서,  $a$ 가 0이면  $I_{n+1} = I_n$  즉, 최근의 누적 평균은 영향을 주지 않는다는 의미이다. 반대로  $a$ 가 1이면  $I_{n+1} = i_n$  즉, 이전 누적 값은 무시하고 가장 최근의 CPU 버스트 값만 고려하겠다는 의미이다.

2.2 ExpQ 방식

ExpQ 방식은 위에서 설명한 EWMA 방식에 분위 값 추산 방식(Quantile Estimation)을 추가한 방식으로서 먼저, 패킷간 유휴 시간을 그림 1과 같이 정의하였다[4].

패킷과 패킷 사이에 유휴 시간동안 전력 절감 모드로 전환하고자 하는 경우, 하드웨어적으로 전력 절감 모드로 전환되는데 시간이 소요되며, 이를 E라고 표시하였다. 그리고 패킷 수신을 위해 다시 전력을 높일 때에도 시간이 소요되며, 이를 W라고 표시하였다. 따라서, 패킷 간 유휴 시간에서 E와 W를 제외한 나머지 시간인 S동안 저전력 모드를 유지할 수 있다. 또한 E와 W의 합이 I 보다 작아야 저전력 모드를 통해 전력을 절감할 수 있다.

다음으로 이 방식이 동작하기 위해 4가지의 모드를 정의하였는데, 동작 모드, 동작 대기 모드, 저전력 대기 모드, 저전력 모드가 있다. 동작 모드는 장치의 모든 요소에 전력이 완전히 공급된 상태를 말하며, 실제 패킷을 처리 중인 모드이다. 동작 대기 모드는 동작 모드에서처럼 전력 공급을 모두 받고 있는 모드

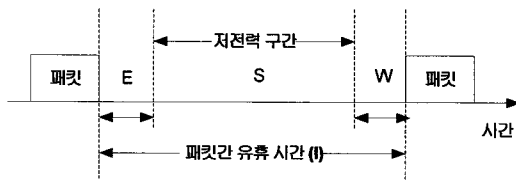


그림 1. 패킷간 유휴 시간의 정의

이지만, 타이머 T1의 설정에 따라 카운트다운을 하면서 패킷의 수신을 모니터링하는 점이 다르다. 저전력 모드는 저전력 상태를 유지하면서 패킷의 수신을 모니터링하는 모드이다. 저전력 대기 모드는 저전력 모드와 동일하게 저전력 모드이지만 타이머 T2의 설정 값에 따라 카운트다운하면서, 패킷의 수신을 모니터링하는 모드이다.

이 정의를 바탕으로 ExpQ 방식의 동작 원리는 그림 2와 같다.

동작 모드가 시작되는 시점에 다음 유휴 시간을 예측하는데, 이는 식 (2)와 같이 표현된다.

$$P_{n+1} = P_i \cdot (1-a) + I_i \cdot a
 \tag{2}$$

식 (2)에서  $P_{i+1}$ 는 새로운 예측치고,  $P_i$ 는 이전의 누적 예측치이며,  $I_i$ 는 직전의 유휴 시간이다.  $a$ 는 0에서 1까지의 상수이다. 임계치 T는 E와 W의 합으로 정의했으며 P가 T보다 작으면 타이머 T1을 예측된 시간인 Q로 설정한다. Q 값을 계산하는 것은 Jain과 Chlamtac의 알고리즘[9]을 사용하였다.

타이머 T1이 타임아웃되기 전에 아무 패킷도 도착하지 않았다면 저전력 상태로 바꾸고 패킷이 도착할 때까지 저전력 모드를 유지한다. 만약 P가 T보다 크면 타이머 T1은 Q 값으로 설정하고, 타이머 T2는 예측치 P로 설정한 후, 저전력 대기 모드로 전환한다. T2가 타임아웃되기 전까지 패킷이 도착하지 않으면, 다시 전력을 높여서 동작 대기 모드로 전환하고, 타이머 T1만큼 기다린다. 동작 대기 모드에서 타이머 T1이 타임아웃되면 비로소 저전력 모드로 전환하고, 저전력 모드에서 패킷 수신 이벤트가 발생하면, 동작

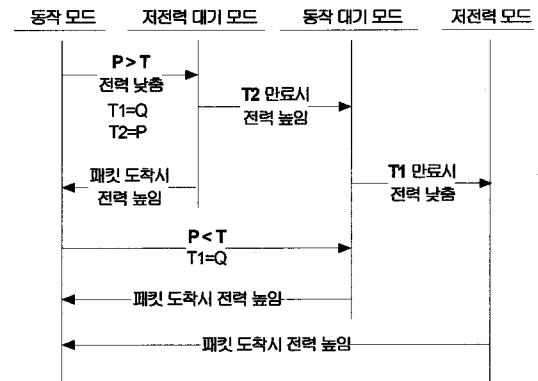


그림 2. 트래픽 기반 유휴 시간 예측 방법

모드로 전환한다.

이렇듯 ExpQ 방식은 EWMA 방식이 3개의 모드를 기반으로 동작하는 것과는 달리 4개의 모드를 기반으로 동작한다. 트래픽 트레이스를 기반으로 한 시뮬레이션에서 이 방식은 EWMA 방식에 비해 좋은 성능 평가를 보여주었으나 복잡도는 증가한다. 본 논문에서는 알고리즘이 복잡하지 않으면서 좋은 성능을 보이는 것이 목표이므로 EWMA 방식을 채택하였다.

### 3. 전력 절감 모드 결정 알고리즘의 설계

본 알고리즘은 기본적으로 직전까지의 유휴 시간을 고려하여 현재의 유휴 시간을 예상하고, 이에 따라 전력 절감 모드로의 전환 여부를 결정한다. 이 알고리즘의 주안점은 전력 절감 모드에 있을 때 최대한 프록시로 처리함으로써 전력 절감 모드의 유지 시간을 늘리는 것이다. 즉, 전력 절감 모드를 유지하면서 네트워크 상태 조회 등의 관리 패킷이나 연결 확인용 응용 프로그램 패킷 등은 직접 처리한다.

#### 3.1 프록시 처리용 패킷 분류

장치가 유휴 상태에 있을 때라도 네트워크상에는 항상 패킷이 송수신된다. 이러한 패킷을 3가지로 분류할 수 있다. 첫 번째는 응답을 필요로 하지 않는 패킷인데, 이는 동작을 취해줄 필요가 없거나 응답을 해줄 필요가 없거나 운영 체제의 프로토콜 스택에서 무시되는 패킷들이다. 여기에 속하는 것은 브로드캐스트되는 패킷이나 라우팅 프로토콜 패킷과 같이 상태를 알려주는 형태의 패킷들이 있으며, 포트 스캔과 같이 해킹용 패킷들도 이에 속한다. 두 번째는 응답 패킷이 단순하게 정해져 있는 패킷들인데, 취해야 하는 동작이나 응답이 정적이기 때문에 패킷에 따라 전력 절감 모드를 유지한 상태에서 프록시로 처리해 줄 수 있다. ARP나 ping, DHCP, 각종 조회용 패킷들이 이에 해당한다. 더불어 본 논문에서는 메신저의 연결 확인 패킷 또한 프록시로 처리하였다. 세 번째는 활성 모드에서 처리해야 하는 사용자 데이터로서, 첫 번째와 두 번째에 해당하지 않는 나머지 사용자 데이터 패킷이다. 즉, 일반적으로 사용하는 TCP, UDP와 응용 프로토콜 패킷이 이에 속한다.

#### 3.2 유휴 시간 예측 알고리즘

사용자 패킷이 없는 시간이 시작되었다고 해서 바로 전력 절감 모드로 바꾸게 되면, 곧 활성화해야 하는 이벤트 발생시 오히려 전력 효율이 떨어질 수 있다. 그러므로 사용자 패킷이 없는 시간이 시작된 후 어느 정도 유휴 시간이라고 판단할 수 있는 시간만큼 기다리는 것이 더 효율적이다. 하지만 유휴 시간은 가변적인 인터넷 트래픽의 특성으로 인하여, 평균적인 값으로 표현하기가 어렵다. 이에 유휴 시간의 패턴을 누적해오면서 이를 기반으로 다음 유휴 시간을 예측해보는 방식으로 접근하였다.

그림 3에서 R은 실행 모드를, I는 유휴 모드를 S는 전력 절감 모드인 시간을 나타낸다. 여기서 두 가지 오버헤드가 발생되는데, R 모드에서 S 모드로 전환시 지연시간이 발생되고, S 모드에서 R 모드로의 전환시 지연시간이 발생된다. 이를 각각 E와 W로 표기하였다. 그림 3의 (a)는 I가 E보다 크거나 같은 경우 발생하는 지연을 보여주고 있고, 그림 3의 (b)는 I가 E보다 작거나 같은 경우 발생하는 지연을 보여주고 있다.

R 모드에서의 전력 소모량을  $P_R$ , S 모드에서의 전력 소모량을  $P_S$ , R 모드에서 S 모드로 또한 S 모드에서 R 모드로 다시 전환되기 위해 필요한 평균 전력 소모량을  $P_{EW}$ 라고 정의하면, 일반적으로,  $P_R \geq P_{EW} \geq P_S$  이다. 이를 통해 시스템의 에너지 이득 (EG: Energy Gain)를 정의하면 식 (3)과 같다.

에너지가 절약되기 위해서는 식 (3)의 EG가 양수가 되어야 하는데, 이는 그림 3의 (a)에 해당해야 한다. 이를 식으로 표현하면 식 (4)에서 식 (8)과 같다.

즉, 에너지 절약 효과를 얻기 위해서는 유휴 시간의 특성이 식 (4)과 식 (8)를 만족해야 한다. 여기서 중요한 것은, 전력 절감 모드로의 전환을 결정할 임계치인  $S_{th}$ 를 정의하는 것이다. 임계치  $S_{th}$ 는 식 (8)에서 보는 바와 같이 본 알고리즘이 적용될 홈 게이트웨이의 전력 소모량 특성인  $P_R, P_S, P_{EW}$ 와 전력 모드

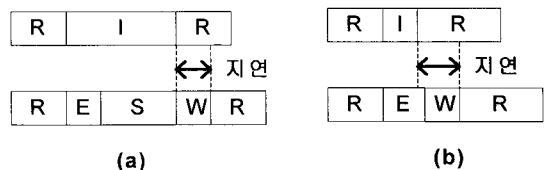


그림 3. 전력 절감 모드 시나리오

$$EG = I \cdot P_R - (E + W) \cdot P_{EW} - P_S \cdot S \quad (3)$$

$$I \geq E \quad (4)$$

$$EG = I \cdot P_R - (E + W) \cdot P_{EW} - P_S \cdot S \quad (5)$$

$$= I \cdot P_R - (E + W) \cdot P_{EW} - P_S \cdot (I - E) \quad (5)$$

$$= I \cdot (P_R - P_S) - E \cdot (P_{EW} - P_S) - W \cdot P_{EW} > 0 \quad (6)$$

$$\Rightarrow I > \frac{E \cdot (P_{EW} - P_S) + W \cdot P_{EW}}{(P_R - P_S)} \quad (7)$$

$$\Rightarrow S_{th} = \frac{E \cdot (P_{EW} - P_S) + W \cdot P_{EW}}{(P_R - P_S)} \quad (8)$$

전환에 소요되는 시간인 E, W를 반영하여 결정된다. 또한 본 논문에서는 현재 유휴 시간을 예측하기 위해 식 (1)을 사용하였으며, a 값으로 0.5를 사용하여 이전의 누적 예측치와 직전 유휴 시간을 50%씩 반영하도록 하였다.

### 3.3 유휴 시간 예측 값의 보정

유휴 시간 예측 알고리즘을 구동하다 보면 다음과 같은 경우에, 예측된 유휴 시간과 실제 유휴 시간의 차이가 커지는 현상이 발생한다. (그림 4의 (a) 부분)

그림 4의 (a)에서 보는 바와 같이, 유휴 시간이 I1, I2, I3, I4의 순서로 나타나는데, I3은 다른 유휴 시간에 비해서 매우 긴 경우를 보여주고 있다. 제안 알고리즘을 적용하면 I3과 I4에 대해 계산된 예측치는 각각 I3'과 I4'가 된다. 이러한 경우, I3'는 실제 I3 시간에 비해 짧은 시간으로 예측되는 문제 (underestimation)가 발생하고, I4'는 실제 I4 시간에 비해 긴 시간으로 예측되는 문제 (overestimation)가 발생한다. 이를 예측 오류라 할 수 있다. 이렇듯 중간에 긴 유휴 시간이 갑자기 나타나면 그 순간과 직후에 계산되는 예측치의 정확도를 떨어뜨릴 수 있다.

이는 그림 4의 (b)와 같이 보정될 수 있다. 예측된

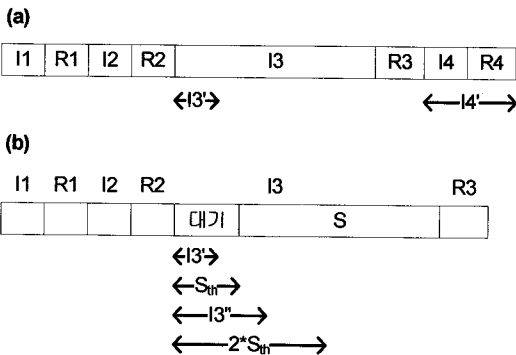


그림 4. 예측 오류의 보정

I3' 값이 임계치(S<sub>th</sub>)의 1배보다 작으면 예측치를 다시 계산한다. 다시 계산한 예측치 I3'' 값이 임계치의 2배보다 작으면 예측치를 또 다시 계산한다. 이렇게 반복될 수 있는데, 이러한 보정 방법을 식으로 표현하면 식 (9)와 같다.

$$\begin{aligned} & \text{if}(a \cdot i_n + (1-a) \cdot I_n > c \cdot I_n) \\ & I_{n+1} = c \cdot I_n \end{aligned} \quad (9)$$

상수 c값은 네트워크의 트래픽 특성에 따라 달라질 수 있으나 본 논문에서의 시뮬레이션에서는 c값을 2로 설정하는 것이 적당하였다. 즉, 예측치가 이전 예측치의 2배가 넘으면 2배의 값을 설정하였다. 즉, 예측치가 갑자기 커져도 이전 임계치의 2배를 넘지 않게 안정되도록 유지하였다.

### 3.4 제안 알고리즘의 상세 설계

설명한 바와 같이 설계한 알고리즘의 동작 흐름은 그림 5와 같다.

그림 5에서 보는 바와 같이 3가지의 모드 (보정 모드, 실행 모드, 저전력 모드)를 기반으로 한다. 실행 모드에서 사용자 패킷 입력이 더 이상 발생되지 않으면 알고리즘에 따라 유휴 시간에 대한 예측치를 계산한다. 예측치가 임계치보다 작으면 카운터를 초기화하고 실행 모드를 유지한다. 실행 모드의 유지 상태에서 카운터가 임계치에 도달하면 보정 모드로 전환된다. 반면 예측치가 임계치보다 크면 전력 절감 모드인 저전력 모드로 전환한다.

보정 모드에서는 유휴 시간에 대한 예측치를 다시 계산하는데, 이 값이 임계치보다 작으면 다시 실행 모드를 유지하면서 대기하는 상태인 실행 모드로 전환되고, 임계치보다 크면 저전력 모드로 전환된다. 저전력 모드에서는 사용자 데이터 패킷 수신 등의

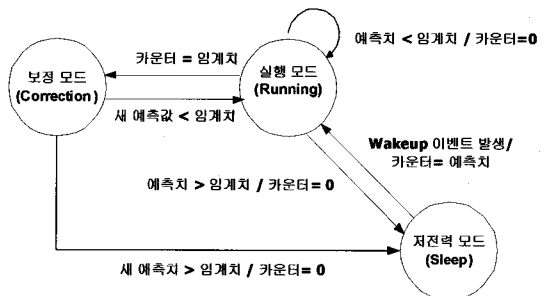


그림 5. 제안 알고리즘

활성화 이벤트가 발생시 즉시 실행 모드로 전환하며, 카운터 값은 유휴 시간 예측치로 설정한다.

#### 4. 구현, 시뮬레이션 및 성능 측정

##### 4.1 시뮬레이션 프로그램의 구성

알고리즘을 홈 게이트웨이에 적용하기 전에 먼저 시뮬레이션을 통해 알고리즘의 설계와 성능을 평가하였다. 본 알고리즘이 적용되는 홈 게이트웨이는 전력 절감 모드 전환에 필요한 지연 오버헤드가 1초 이상이므로 상용 ACPI 방식과 성능을 비교한다. 시뮬레이션 프로그램의 전체 흐름은 그림 6과 같다.

먼저, 사용자 인터페이스 화면으로부터 여러 트래픽 입력 중에 하나를 선택한다. 그리고 시작 버튼을 누르면 시뮬레이션 프로그램은 해당 트래픽 입력 파일을 한 줄씩 읽어 들인다. 트래픽 입력 파일의 형식은 다음과 같이 각 줄이 하나의 패킷을 의미하고, 일련번호, 타임스탬프, 송신지 주소, 수신지 주소, 프로토콜 종류로 구성된다.

예: 1 3.698128 203.241.249.109 211.233.29.200 TCP

시뮬레이션 프로그램은 이러한 입력 파일을 한 줄씩 읽으면서 타임스탬프 값을 보고 패킷 이벤트를 생성한다. 패킷 수신에 대해서 두 가지 알고리즘이 동시에 동작하는데, (a) 알고리즘은 기존에 개인용 컴퓨터에서 사용하는 ACPI (Advanced Configuration and

Power Interface) 표준 [10]을 기반으로 동작한다. ACPI는 소프트웨어와 하드웨어가 운영 체제, 마더보드, 주변장치(시디롬, 하드디스크 등)의 전력 사용에 대해 어떻게 대처하는지에 대해 정의한 것으로서, 전력 관리와 플러그 앤 플레이를 운영체제 안에 내장하는 것으로 오직 필요한 만큼만 주변장치에 전력이 공급되도록 한다. 가장 많이 쓰이는 주요 기능은 전력 관련 기능이라고 할 수 있으며, 네트워크를 통한 원격 전원 관리 기능도 포함되어 있다. 윈도우에서는 ACPI를 지원하고 있으며, 설정된 시간 동안 키보드 입력 등의 사용자 이벤트가 발생하지 않는 시간이 지속되면 시스템 대기 모드로 전환한다. 시스템 대기 모드로의 전환 기준은 보통 20분으로 설정되어 있으므로, 20분간 활동이 없으면 전력 절감 모드로 전환하도록 구현하였다.

(b) 알고리즘은 본 논문에서 적용한 알고리즘으로서, 다음 절에서부터 상세히 설명한다. 시뮬레이션이 진행되다가 사용자가 중지 버튼을 누르거나 파일 입력이 끝나면 전력 소모량을 화면에 제시하고, (a) 알고리즘에 비해 (b) 알고리즘이 얼마나 전력 절감 효과가 있는지를 보여준다. 그리고 시뮬레이션 프로그램이 종료되면 모든 로그를 텍스트 파일로 저장한다.

패킷 트래픽은 네트워크에 수집한 여러 시나리오를 적용하였으며, 그 중에서 본 논문에서는 다음과 같은 시나리오를 예로 제시하였다.

##### 4.2 공통 변수 및 상수 정의

제안 알고리즘의 구현과 시뮬레이션 프로그램의 구현을 위해 공통적으로 사용되는 변수와 상수는 다음과 같다.

표 1에서 THRESHOLD\_FOR\_PREDICTION은 식 (8)에 의해 계산된다.

##### 4.3 시뮬레이션 프로그램의 구현

구현한 알고리즘을 검증하기 위해서 먼저, 패킷 수신부에 그림 7과 같은 루틴을 추가하였다.

패킷 수신부에서는 먼저 패킷 수신 이벤트를 항상 체크한다. 시뮬레이션에서는 트래픽 입력 파일에 기록된 타임스탬프를 보고 패킷 수신 이벤트를 발생시키게 되지만, 차후 홈 게이트웨이에 통합할 때에는 리눅스 커널의 패킷 수신부에 추가 구현된다. 만약

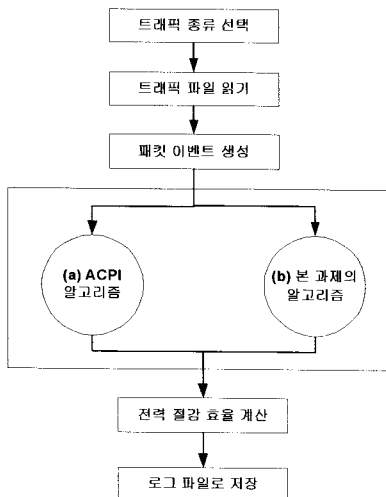


그림 6. 시뮬레이션 프로그램의 동작 흐름

표 1. 시뮬레이션 상수 및 변수 설정

변수명	설정	설명
TIME_TO_WAKEUP	1(초)	Wakeup시 소요시간
TIME_TO_SLEEP	1(초)	Sleep으로 전환시 소요 시간
POWER_WHEN_RUNNING	19.5(Wh)	동작 중일 때의 시간당 소비전력
POWER_WHEN_SLEEPING	7.5(Wh)	Sleep 중일 때의 시간당 소비전력
POWER_WHEN_TO_SLEEP	13.5(Wh)	Sleep으로 시간당 소비전력
POWER_WHEN_TO_WAKEUP	21(Wh)	Wakeup시 시간당 소비전력
THRESHOLD_FOR_PREDICTION	2.25(초)	계산된 임계치 시간
TIMER_FOR_SLEEP	3(초)	설정된 임계치 시간

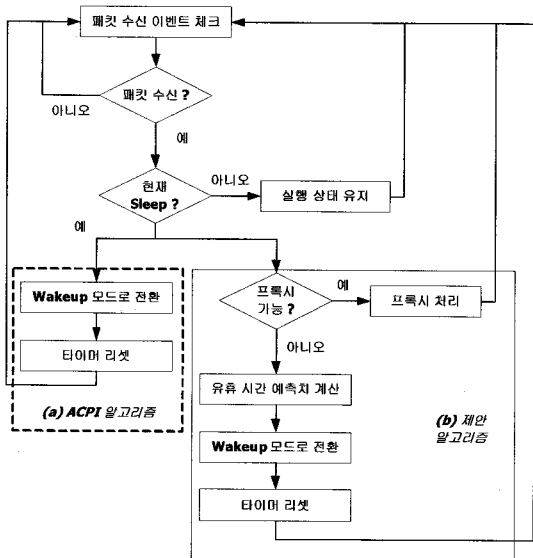


그림 7. 시뮬레이션 - 패킷 수신부 처리

패킷이 수신되었으면, 현재 상태가 전력 절감 모드인지 체크하는데 전력 절감 모드가 아니면 현재의 모드를 유지하고 전력 절감 모드이면 두 가지 알고리즘이 각각 구동된다. ACPI 알고리즘의 경우, 바로 활성화(Wakeup) 모드로 전환하고, 타이머를 초기화한다. 이에 반해 제안 알고리즘은 프록시 처리를 할 수 있는 패킷인 경우 전력 절감 모드인 채로 프록시 처리를 한다. 프록시 처리 대상이 아닌 일반 사용자 패킷에 대해서는 유휴 시간 예측치를 계산하고, 활성 모드로 전환한 후, 타이머를 초기화한다.

다음으로, 성능 비교를 위해서 본 연구에서는 그림 8과 같이 ACPI 알고리즘을 구현하였다.

ACPI에서 실행 모드는 Working 모드라고 하며, G0와 S0 상태가 이에 해당한다. 이 상태에서 사용자 입력 이벤트가 없는 시간을 카운트하여 대기 모드

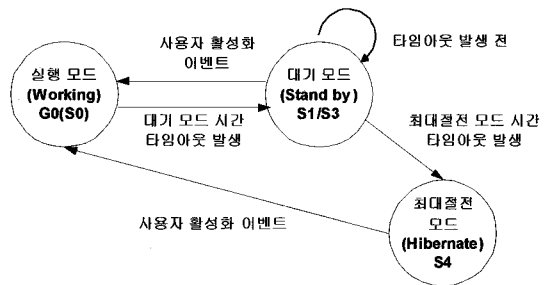


그림 8. 시뮬레이션 - ACPI 알고리즘

시간 설정 값을 기준으로 타임 아웃되면 대기 모드로 바뀐다. 대기 모드는 Standby 모드라고 하며, S1 상태나 S3 상태가 이에 해당하는데, 기본 값으로는 20분간 사용자 입력 이벤트나 웨이크업 패킷이 발생되지 않으면 대기 모드로 바뀐다. 만약 사용자가 최대절전 모드를 사용하도록 설정하였고, 그 시간 동안 아무 이벤트가 발생하지 않았다면 최대절전 모드로 바뀐다. 이는 하이버네이트(hibernate)라고 하며, S4 상태가 이에 해당한다. 보통 30분으로 설정되어 있다. 대기 모드나 최대절전 모드에서 사용자 활성화 이벤트(키보드 입력, 웨이크업 패킷의 도착 등)가 발생하면, 실행 모드로 전환한다.

#### 4.4 시뮬레이션 실행 결과 및 성능 분석

시뮬레이션 프로그램은 Visual C++을 사용하여 개발되었으며, 화면은 그림 9와 같이 구성된다.

그림 9에서 보는 바와 같이 '[INPUT] Traffic Trace' 부분은 트래픽 입력 파일을 보여주는 것으로서 맨 앞에 있는 [ ] 안에는 시간이 표시되고, ( ) 안에는 패킷 일련 번호가 표시되며, 그 다음으로는 타임스탬프와 프로토콜 종류가 표시된다. 패킷 캡처 프로그램인 Wireshark [11]를 사용하여 수집된 트래픽을

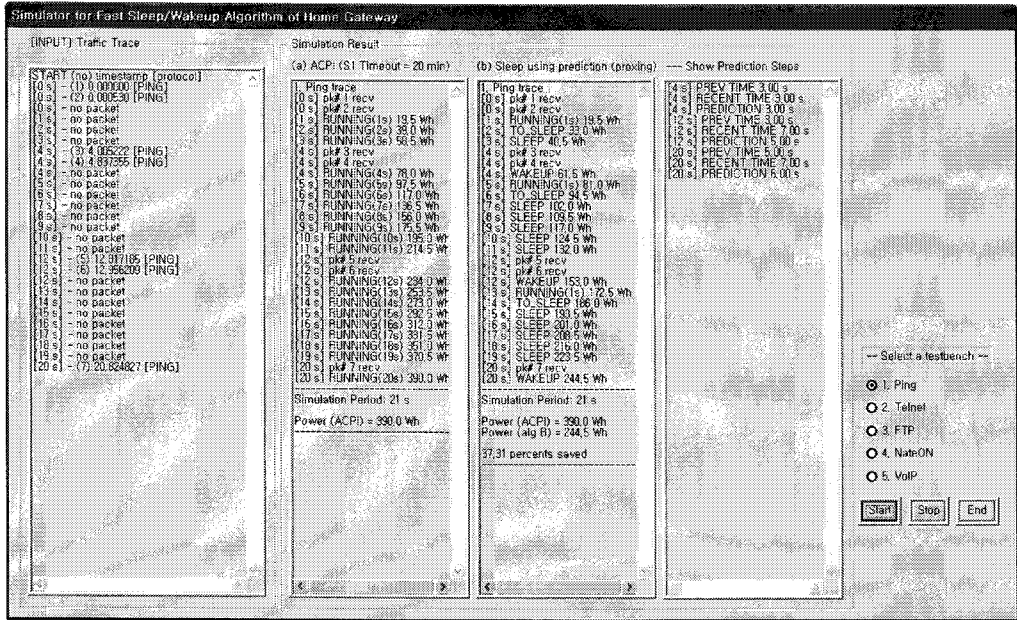


그림 9. 시뮬레이션 프로그램의 화면 구성

텍스트 파일로 전환하여 사용하였다.

오른쪽 3개의 박스 부분은 시뮬레이션 결과 화면으로서, (a) 부분은 ACPI 알고리즘을 적용한 경우의 처리 로그이고, (b) 부분은 제안 알고리즘을 적용한 경우의 처리 로그이다. 그리고 가장 오른쪽 영역에서는 제안 알고리즘에서 계산되는 예측치 값을 보여주는데, 직전까지의 누적 예측치 (PREV TIME), 직전의 유휴 시간 (RECENT TIME), 이를 통해 계산된 예측치 (PREDICTION)를 보여준다.

(a) 부분과 (b) 부분에서는 매초마다 알고리즘에 따라 전력 모드 (RUNNING, TO\_SLEEP, SLEEP 등)를 표시하고, 이에 대한 전력 소모량을 누적한다. 중지(Stop) 버튼을 누르거나 입력 파일의 끝에 도달하면 결과화면을 출력해주는데, 총 시뮬레이션 시간과, 전력 소모량을 보여준다. 특히 (b) 알고리즘에 대해서는 (a) 알고리즘에 비해 얼마나 전력 효율이 높아졌는지 퍼센트 비율로 보여준다. 위의 경우에는 ACPI 알고리즘의 경우, 7초간 117 Wh를 소모하였고, 제안 알고리즘의 경우 94.5 Wh를 소모하여, ACPI 알고리즘에 비해 19.23 % 전력 효율이 높아졌음을 알 수 있다.

이러한 시뮬레이션 프로그램을 사용하여 홈 네트워크에서 주로 사용되는 응용 프로그램인 방문자 화

인, 텔넷, FTP, 그리고 메신저, 인터넷 전화 트래픽에 대해 다섯 가지 시나리오를 테스트해보았다.

방문자 확인 트래픽을 테스트했을 때, ACPI 알고리즘의 경우, 275초간 3543.0 Wh를 소모하였고, 제안 알고리즘의 경우 2419.5 Wh를 소모하여, ACPI 알고리즘에 비해 54.72 % 전력 효율이 높아졌다. 텔넷 트래픽을 테스트했을 때, ACPI 알고리즘의 경우, 40초간 877.5 Wh를 소모하였고, 제안 알고리즘의 경우 526.4 Wh를 소모하여, ACPI 알고리즘에 비해 40.00 % 전력 효율이 높아졌다. FTP 트래픽을 테스트했을 때, ACPI 알고리즘의 경우, 29초간 546.0 Wh를 소모하였고, 제안 알고리즘의 경우 453.0 Wh를 소모하여, ACPI 알고리즘에 비해 17.03 % 전력 효율이 높아졌다. 메신저 트래픽을 테스트했을 때, ACPI 알고리즘의 경우, 16초간 292.5 Wh를 소모하였고, 제안 알고리즘의 경우 250.5 Wh를 소모하여, ACPI 알고리즘에 비해 14.36 % 전력 효율이 높아졌다. 그리고 인터넷 전화 서비스를 고려하기 위해 실시간 전송(RTP: Real Time Transport) 프로토콜을 사용하는 H.223 트래픽을 사용하여 테스트하였다. 그 결과 ACPI 알고리즘의 경우, 32초간 624.0 Wh를 소모하였고, 제안 알고리즘의 경우 315.0 Wh를 소모하여, ACPI 알고리즘에 비해 49.52 % 전력 효율이 높아졌다.



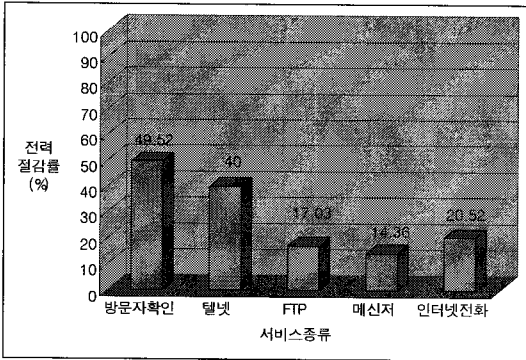


그림 10. 성능 평가 - 전력 절감률

따라서 ACPI의 전력 소모량을 기준으로 본 알고리즘의 전력 소모량을 비율로 나타내면 그림 10과 같다.

그림 10에서 보는 바와 같이 사용자 트래픽이 상대적으로 적었던 방문자 확인 서비스와 텔넷, 인터넷 전화 서비스의 경우, 최소 20%에서 최대 49%까지 전력 소모를 절감할 수 있음을 알 수 있다. 트래픽의 세부 패턴은 홈 네트워크 사용자에게 따라 매우 가변적이므로 이러한 결과는 세부 패턴에 따라 전력 절감 효과도 달라질 수 있다. 이는 본 알고리즘이 네트워크 트래픽의 변화를 반영하고 있음을 내포하고 있는 것이다. FTP나 메신저와 같이 사용자 트래픽의 비율이 높은 응용 프로그램에서도 비교적 좋은 절감 효과를 얻을 수 있음을 알 수 있다.

다음으로 본 알고리즘이 전송 속도에 미치는 영향은 그림 11과 같다.

본 알고리즘을 적용한 후 전송 속도에 어떠한 영향을 미치는지 분석하기 위해 본 논문에서는 전력

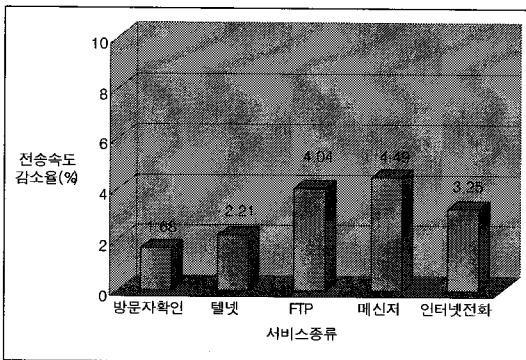


그림 11. 전송 속도 감소율

절감 알고리즘을 적용하기 전에 비해서 적용한 후에 전송 속도가 감소한 비율을 퍼센트 단위로 계산하였다. 여기서 전송 속도는 초당 전송되는 비트수로 정의하며 Mbps의 단위이다. 알고리즘 적용 후 에너지 절약 효율을 위해 지연이 발생하게 되는데, 시뮬레이션용 데이터는 고정되어 있으므로 전송속도가 감소한다. 그림 11은 이 비율이 어느 정도인지를 평가하는 항목이다.

그림 10의 전력 절감 효과와 관련하여 그림 11의 전송 속도 감소율을 분석하면 다음과 같다. 방문자 확인 서비스나 텔넷 서비스와 같이 유휴 시간이 자주 나타나고 길게 유지되는 서비스의 경우, 저전력 모드가 유지되는 시간이 길어진다.

따라서 Sleep/Wakeup의 반복이 적고, 이로 인한 지연이 적게 발생하므로 전송 속도도 크게 떨어지지 않는다. 그러나 FTP 서비스, 메신저 서비스, 인터넷 전화 서비스와 같이 패킷 송수신이 꾸준히 이루어지는 경우, Sleep/Wakeup이 자주 발생되고 Sleep 시간이 길게 유지되지 못하므로 전력 절감 효율이 떨어진다. 또한 잦은 Sleep/Wakeup으로 인해 지연 시간이 증가하므로 전송 속도 감소율은 높아진다. 이러한 결과는 트래픽의 특성과 전력 절감 알고리즘의 동작 원리 및 임계치 설정에 따라 달라질 수 있다.

#### 4.5 생활 패턴에 따른 홈 네트워크의 전력 절감 효과

홈 네트워크의 전력 소비는 구성원의 생활 패턴에 따라 영향을 받는다. 논문 [7]에서 제시한 바와 같이 다음과 같은 생활 패턴이 예가 될 수 있다.

그림 12에서 패턴 A는 가족 구성원들이 낮 시간동안 외출하는 경우이고, 패턴 B와 C는 낮 시간동안 가정 내에서 홈 네트워크를 사용하는 구성원이 있는 경우이다. 본 알고리즘의 시뮬레이션은 사용자가 실내에 있는 'Stay' 시간대에 해당하는 것으로서, 시뮬레이션 결과에서 보는 바와 같이 불필요한 에너지

Pattern	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23		
A	Sleeping											Going-out		Sleeping												
	Sleeping											Going-out		Sleeping												
B	Sleeping											Going-out		Going-out		Sleeping										
	Sleeping											Going-out		Going-out		Sleeping										
C	Sleeping											Going-out		Going-out		Sleeping										
	Sleeping											Going-out		Going-out		Sleeping										

그림 12. 홈 네트워크에서의 생활 패턴의 예

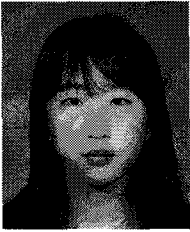
소모를 줄일 수 있다. 한편 'Sleeping' 시간과 'Going-out' 시간 동안에는 방문자 알림 서비스 등의 간헐적인 통신이 필요하거나 홈 네트워크가 거의 사용되지 않는 시간대이므로 'Stay' 시간대에 비해 전력 절감 효과가 더욱 커진다. 따라서 24시간을 기준으로 한 경우, 본 알고리즘의 전력 절감 효과가 보다 크다고 볼 수 있다.

## 5. 결 론

유비쿼터스 홈 서비스를 실현하기 위해 홈 게이트웨이는 24시간 가동되어야 하므로 소모 전력을 절감하는 기술을 필요로 하고 있다. 이는 홈 네트워크에 입출력되는 트래픽을 감시하여 사용자의 서비스 트래픽이 있는 경우 시스템을 활성 모드로 구동하고, 사용자의 서비스 트래픽이 없는 경우 시스템을 전력 절감 모드로 구동하여 소비전력을 줄이는 개념을 토대로 한다. 본 논문에서는 가변적인 네트워크 트래픽을 반영하기 위해 지난 유휴 시간의 기록에 따라 전력 절감 모드를 결정하는 알고리즘을 설계하였다. 시뮬레이션 결과 기존의 ACPI 방식에 비해 최소 14%에서 최대 49%의 전력 절감 효과를 보였다. 따라서 본 알고리즘은 네트워크 트래픽의 변화를 반영하면서도 전력 절감 효율이 좋은 알고리즘이다. 향후 연구로는 전력 절감 모드의 사용시 발생하는 지연 시간을 줄여서 서비스 품질을 보장하는 연구로 본 알고리즘을 확장하는 것이다.

## 참 고 문 헌

- [1] 지식경제부, "홈네트워크 대기전력 저감기술 등 고효율에너지기기 개발 본격 추진," 보도자료, 2006.
- [2] U. S. Department of Energy, "Tips on Saving Energy & Money at Home," <http://www1.eere.energy.gov/consumer/tips/>, 2006.
- [3] C. H.Hwang and A. C. H. Wu, "A Predictive System Shutdown Method for Energy Saving of Event-Driven Computation," *ACM Transactions on Design Automation of Electronic Systems*, Vol.5, No.2, p. 226-241, 2000.
- [4] P. C. Gunaratne and K. J. Christensen, "Predictive power management method for network devices," *IEE Electronics Letters*, Vol.31, No.13, 2005.
- [5] K. J. Christensena, C. Gunaratnea, B. Nordmanb, and A. D. Georgec, "The next frontier for communications networks: power management," *Elsevier Computer Communications*, Vol.27, Issue 18, pp. 1758-1770, 2004.
- [6] J. S Han, I. T. Han, and K. R. Park, "Service-Oriented Power Management for an Integrated Multi-Function Home Server," *IEEE Transactions on Consumer Electronics*, Vol.53, No.1, 2007.
- [7] H. S. Mok, S. Y. Son, J. H. Hong, and S. H. Kim, "An Approach for Energy-Aware Management in Ubiquitous Home Network Environment," *Proceedings of SEUS 2007*, LNCS 4761, pp. 293-300, 2007.
- [8] R. Cornea, N. Dutt, R. Gupta, S. Mohapatra, A. Nicolau, C. Pereira, S. Shukla, N. Venkatasubramaniam, "ServiceFORGE: A Software Architecture for Power and Quality Aware Services," *Proceedings of Workshop at the International FME Symposium*. 2003.
- [9] R. Jain and I. Chlamtac, "The P2 algorithm for dynamic calculation of quantiles and histograms without storing observations," *Communications of the ACM*, pp. 1076-1085, 1985.
- [10] HP, Intel, Microsoft, Phoenix, Ohshiba, "Advanced Configuration and Power Interface Specification Rev. 3.0b," <http://www.acpi.info/DOWNLOADS/ACPIspec30b.pdf>, 2006.
- [11] G. Combs, "Wireshark: network protocol analyzer 0.99.8 Released," <http://www.wireshark.org/>, 2008.



공 인 엽

2000년 2월 부산대학교 컴퓨터공  
학과 졸업  
2002년 2월 부산대학교 컴퓨터공  
학과 석사  
2007년 2월 부산대학교 컴퓨터공  
학과 박사  
2007년 4월~2008년 3월 인제대  
학교 PostDoc

2008년 9월~현재 금오공과대학교 전자공학부 전임강사  
관심분야 : RFID/USN, 홈네트워크, IPv6



황 원 주

1998년 2월 부산대학교 컴퓨터공  
학과 졸업  
2000년 2월 부산대학교 컴퓨터공  
학과 석사  
2002년 9월 일본 오사카대학 정  
보시스템공학과 박사  
2002년 10월~현재 인제대학교  
정보통신공학과 조교수

관심분야 : 홈네트워크, 무선센서네트워크, 통신 최적화  
이론