

인터럽트 병합 최적화를 통한 네트워크 장치 에너지 절감 방법 연구*

이재열** · 한재일*** · 김영만****

A Study on Energy Savings in a Network Interface Card Based on Optimization of Interrupt Coalescing*

Jaeyoul Lee** · Jaeil Han*** · Young Man Kim****

■ Abstract ■

The concept of energy-efficient networking has begun to spread in the past few years, gaining increasing popularity. A common opinion among networking researchers is that the sole introduction of low consumption silicon technologies may not be enough to effectively curb energy requirements. Thus, for disruptively boosting the network energy efficiency, these hardware enhancements must be integrated with ad-hoc mechanisms that explicitly manage energy saving, by exploiting network-specific features. The IEEE 802.3az Energy Efficient Ethernet (EEE) standard is one of such efforts. EEE introduces a low power mode for the most common Ethernet physical layer standards and is expected to provide large energy savings. However, it has been shown that EEE may not achieve good energy efficiency because mode transition overheads can be significant, leading to almost full energy consumption even at low utilization levels. Coalescing techniques such as packet coalescing and interrupt coalescing were proposed to improve energy efficiency of EEE, but their implementations typically adopt a simple policy that employs a few fixed values for coalescing parameters, thus it is difficult to achieve optimal energy efficiency. The paper proposes adaptive interrupt coalescing (AIC) that adopts an optimal policy that could not only improve energy efficiency but support performance. AIC has been implemented at the sender side with the Intel 82579 network interface card (NIC) and e1000e Linux device driver. The experiments were performed at 100 M bps transfer rate and show that energy efficiency of AIC is improved in most cases despite performance consideration and in the best case can be improved up to 37% compared to that of conventional interrupt coalescing techniques.

Keyword : Energy Efficient Ethernet, Packet Coalescing, Interrupt Coalescing, Adaptive Interrupt Coalescing, Packet Delay, Performance

Submitted : July 23, 2015

1st Revision : August 19, 2015

Accepted : August 21, 2015

* 이 논문은 2015년도 정부(미래창조과학부)의 재원으로 정보통신기술진흥센터의 지원(No. R0101-15-237, 고집적 저전력 프로세서 기반 30% 이상 에너지 절감 범용 운영체제 및 가상화 핵심기술 개발)과 2015년도 국민대학교 교내연구비의 지원으로 수행되었음.

** 국민대학교 대학원 컴퓨터공학과 석사과정

*** 국민대학교 컴퓨터공학과 교수, 교신저자

**** 국민대학교 컴퓨터공학과 교수

1. 서 론

IT서비스에 요구되는 급속한 에너지 증가를 완화 하기 위해 다양한 에너지 절감 노력이 진행되고 있다. 2010년 전 세계 데이터 센터의 네트워크관련 소비 전력은 전체 데이터 센터 소비 전력의 약 5%에 이른다(Koomey, 2011). IEEE 802.3 Working Group은 네트워크의 에너지를 절감하기 위해서 2010년 9월에 IEEE 802.3az 에너지 효율적 이더넷(Energy Efficient Ethernet, EEE) 표준안을 제정하였다(EEE, 2010).

에너지 효율적 이더넷은 LPI(Low Power Idle) 모드를 사용함으로써 에너지를 절감한다. LPI 모드는 전송할 패킷이 없을 경우 링크 상태를 저전력으로 전환함으로써 에너지가 절감 될 수 있도록 한다. 에너지 효율적 이더넷은 데이터가 버스트 특성을 가질 때는 에너지 절감율이 크지만 패킷의 크기가 작고 주기적으로 전송되는 상황에서는 잦은 모드 전이로 인해 에너지 절감율이 감소하는 문제점이 있다(Reviriego et al., 2011; Christensen et al., 2010; Prasad et al., 2004). 이런 에너지 효율적 이더넷의 문제를 해결하기 위해 패킷 병합(packet coalescing), 인터럽트 병합(interrupt coalescing)과 같은 병합 기술이 제안되었다.

패킷 병합(Christensen et al., 2010)은 임계시간과 임계패킷 수라는 조건을 설정하고 조건을 만족하면 패킷을 전송하는 메커니즘이다. 그러나 임계시간과 임계패킷 수가 고정적이기 때문에 패킷 병합을 적절히 적용하기 위해서는 먼저 시스템의 사용패턴을 분석해야 하는 문제가 있다.

인터럽트 병합(Prasad et al., 2004)은 송신(또는 수신)되는 패킷마다 CPU에게 인터럽트로 처리 요청을 하는 대신, 정해진 짧은 시간 구간에 송신(또는 수신)되는 여러 패킷을 묶어 하나의 인터럽트로 처리함으로써 CPU 로드를 줄이는 것이 주 목적이거나 에너지 효율 개선에 사용할 수도 있다. 그러나 인터럽트 발생 주기가 미리 고정적으로 설정되어 있어 상황에 따라 최적의 주기로 인터럽트를 병합

하지 못하는 문제가 있다.

본 논문에서는 기존 병합의 문제로 인해 에너지 절감율이 떨어지는 것을 해결하기 위해 적응적 인터럽트 병합을 제안한다. 적응적 인터럽트 병합은 네트워크 로드와 요구되는 성능을 고려하여 최적의 인터럽트 발생 주기를 설정하도록 함으로써 에너지를 절감하도록 한다.

본 논문의 구성은 다음과 같다. 제 2장은 관련 연구를 기술하고, 제 3장은 인터럽트 병합 최적화와 적응적 인터럽트 병합에 대해 설명한다. 제 4장은 실험 환경 및 실험 결과에 대해 논하고, 제 5장은 결론을 기술한다.

2. 관련 연구

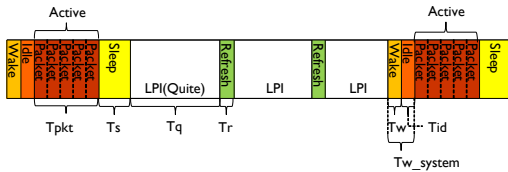
본 장에서는 메시지 병합 최적화를 통한 네트워크 장치 에너지 절감 방법을 연구하기위한 배경 지식인 기존 이더넷 네트워크 장치 에너지 사용, 에너지 효율적 이더넷, 패킷 병합, 인터럽트 병합을 설명한다.

2.1 기존 이더넷 네트워크 장치

기존 이더넷 네트워크 장치(Network Interface Card, NIC)는 에너지 사용을 고려하지 않아 송신 또는 수신할 데이터가 없는 상황에서도 최대 파워를 소모한다(Reviriego et al., 2011). 따라서 네트워크 장치를 사용하지 않음에도 불구하고 에너지를 낭비하게 되며, 기존 네트워크 장치는 항상 최대 파워를 소모하게 된다. 다음 절에서는 기존 네트워크 장치의 에너지 소모량 절감을 위해 제안된 기술을 설명한다.

2.2 에너지 효율적 이더넷

에너지 효율적 이더넷(EEE, 2010)은 이더넷에서 에너지를 절감하기 위한 표준 프로토콜이다. 에너지 효율적 이더넷에는 Active 모드와 LPI 모드, 두 가지 모드가 존재한다. Active 모드에서는 네트워크



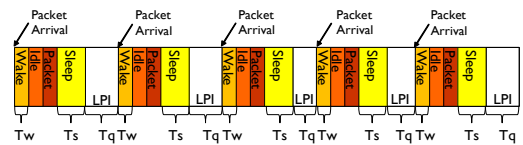
<Figure 1> Mode Transitions in EEE

장치의 파워가 공급되고 패킷이 전송된다. 그러나 더 이상 전송할 데이터가 존재하지 않으면 네트워크 장치는 저전력 상태인 LPI 모드로 변경된다. Quite 모드라고도 불리는 LPI 모드에서는 네트워크 장치에 전원이 적게 공급되며 패킷 송/수신을 중단한다. 이런 특성 때문에 LPI 모드에서 에너지를 절감할 수 있다. 패킷이 네트워크 카드에 도착하게 되면 네트워크 카드는 몇 마이크로 초안에 Active 모드로 변경되고 패킷을 전송한다. <Figure 1>에서 Ts는 네트워크 장치가 Active 모드에서 LPI 모드로 변경될 때 필요한 전환시간이고, Tw는 네트워크 장치가 LPI 모드에서 Active 모드로 변경될 때 필요한 전환시간이다. Tw_system은 시스템이 패킷을 송신할 수 있도록 준비하는 시간이다. Tid는 전력을 많이 소모하지만 패킷을 전송하지 않는 시간으로써 시스템이 패킷을 송신할 수 있도록 준비하는 시간(Tw_system)에서 네트워크 장치가 Wake하는 시간(Tw)을 제외한 시간이다. Tpkt는 패킷 전송 시간이다. Tq는 네트워크 장치가 에너지를 가장 적게 소비하는 LPI 모드 시간이다. 마지막으로 링크의 상태를 유지하기 위해 Tq 시간이 지나면 주기적으로 Tr 시간동안 Refresh 신호를 보낸다.

<Table 1>은 IEEE 802.3az 표준(IEEE 802, 2010)에서 정하고 있는 Ts, Tw, Tq, Tr, Tw_system의 시간을 보여준다. <Table 1>의 시간을 자세히 살펴보면 100M bps에서는 Tq 시간이 20,000μs 인 반면, 10G bps에서는 39.68μs로 상당히 줄어든 것을 확인할 수 있다. 또한, 1G bps와 10G bps에서는 Tw와 Tw_system이 같은 시간을 가지고 있어 Idle 모드 시간이 0μs 인 반면, 100M bps에서는 9.5μs의 시간을 갖는 것을 확인할 수 있다.

<Table 1> Time defined in IEEE 802.3az Standard (μs)

Type	Ts	Tw	Tq	Tr	Tw_system
100Mbps	200	20.5	20,000	200	30
1Gbps	182	16.5	20,000	198	16.5
10Gbps	2.88	4.48	39.68	1.28	4.48



<Figure 2> Close-To-Worst Case for EEE Efficiency

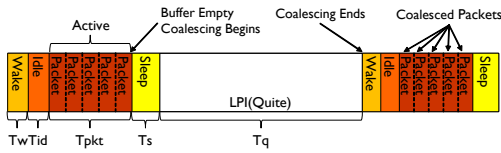
패킷 전송시간인 Tpkt의 경우 패킷 크기가 1500 바이트일 경우 100M bps에서 120μs, 1G bps에서 12μs, 10G bps에서 1.2μs로 나타낼 수 있다(Revi-riego et al., 2011). Tr의 경우 100M bps에서 200μs로 다소의 시간이 소요되나 매우 간헐적으로 발생한다. 본 논문에서는 AIC의 초기 평가를 위해 실험시 100M bps만을 고려하며 병합 임계시간을 20,000 μs로 설정한다. 이 경우 Tr은 병합 임계시간에 비해 상대적으로 매우 작으며 간헐적으로 발생하여 실험에 대한 영향이 미미하므로 고려하지 않는다.

2.2.1 에너지 효율적 이더넷의 문제점

위에서 설명한대로 에너지 효율적 이더넷은 전송할 패킷이 없을 경우 LPI 모드로 전환됨으로써 네트워크 장치에서 소비되는 에너지를 절감할 수 있다. 그러나 에너지 효율적 이더넷에서 데이터 크기가 작은 패킷이 주기적으로 발생하는 경우 잦은 Wake와 Sleep으로 인한 모드 전이 오버헤드로 인해 에너지 절감율이 떨어진다. <Figure 2>는 데이터 크기가 작은 패킷이 주기적으로 발생하여 에너지 효율적 이더넷에서 에너지 절감율이 최저인 상황을 보여준다.

2.3 패킷 병합

패킷 병합(Christensen et al., 2010)은 에너지 효

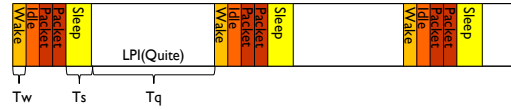


<Figure 3> EEE with Packet Coalescing

올적 이더넷의 문제점을 극복하기 위해 제안되었으며, 패킷을 병합하기 위해 즉 여러 개의 패킷을 모아 한번에 보내기 위한 임계 시간과 임계 패킷 수를 설정한다. 두 가지 조건 중 임계 시간 또는 임계 패킷 수를 만족하게 되면 패킷을 전송 할 수 있는 상태로 전환한다. 첫 번째 패킷이 도착한 순간부터 타이머를 동작시키며, 임계시간 동안 병합된 패킷 수가 임계 패킷 수에 미치지 못하면 타이머가 만료되어 상태가 전환된다. 반대로 임계시간 전 즉, 타이머가 만료하기 전에 병합된 패킷 수가 임계 패킷 수에 도달하게 되면 임계 패킷 수 조건에 의해 상태가 전환된다. <Figure 3>은 패킷 병합을 적용한 사례를 보여주고 있으며, <Figure 2>와 비교해 볼 때 Wake와 Sleep 모드 전이가 줄어들고 LPI 모드 시간이 증가함을 알 수 있다.

2.3.1 패킷 병합의 문제점

패킷 병합은 에너지 효율적 이더넷보다 에너지 절감율을 개선하였지만 여전히 문제점은 남아있다. 패킷 병합에서의 문제는 패킷을 병합하기 위한 임계시간과 임계 패킷 수가 고정되어 있는 것이다. 임계시간과 임계 패킷 수를 너무 적게 설정하게 되면 패킷 병합률이 떨어지게 되어 에너지를 절감율도 떨어지게 된다. 반대로 임계시간과 임계 패킷 수를 너무 크게 설정하게 되면 패킷 지연시간이 크게 발생하여 네트워크의 성능에 악영향을 미치게 된다. <Figure 4>는 <Figure 2>의 상황에서 임계 패킷 수가 2인 경우, 즉 패킷을 2개씩 병합했을 때의 상황을 보여준다. 패킷 병합 수를 더 크게 하면 LPI 모드 시간을 더 늘릴 수 있음에도 불구하고 고정된 임계패킷 수 때문에 2개 이상을 병합하지 못한다.



<Figure 4> Packet Coalescing(Packet Count Limit = 2)

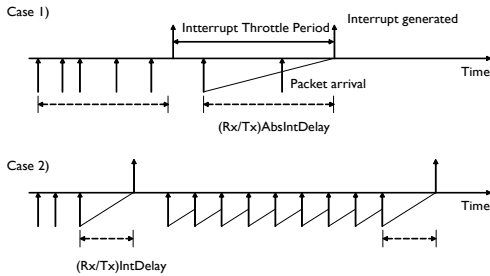
2.4 인터럽트 병합

인터럽트 병합(Prasad et al., 2004; MSDN, 2015; Makineni, 2006)은 송/수신된 패킷을 처리하기 위하여 일정 시간 동안 발생한 여러 인터럽트를 모아서 하나의 인터럽트로 처리함으로써 CPU의 오버헤드를 줄이는 것이 주목적이나 에너지 절감에도 사용된다. 오늘날 대부분의 기가비트 이더넷 네트워크 장치들은 인터럽트 병합을 지원한다.

사용자는 네트워크 장치 드라이버와 하드웨어 세팅을 보거나 설정할 수 있는 도구(ETHTOOL, 2015) 또는 네트워크 장치 드라이버 옵션을 통해서 인터럽트 병합에 요구되는 주요 매개변수를 설정할 수 있다. 예를 들어, 인텔 e1000e 장치 드라이버(E1000, 2014)는 다음과 같은 매개변수가 있다.

- InterruptThrottlePeriod(ITP) : 인터럽트 발생 후 다음 인터럽트를 발생시킬 시간, 즉 인터럽트 발생 주기를 설정
- (Rx/Tx)AbsIntDelay : 첫 번째 패킷 도착 후 다음 인터럽트를 발생시킬 시간
- (Rx/Tx)IntDelay : 마지막 패킷을 받은 후 다음 인터럽트를 발생시킬 시간

<Figure 5>의 상황 1은 (Rx/Tx)AbsIntDelay 타이머가 이전 인터럽트 이후 첫 번째 패킷을 받았을 때 다음 인터럽트를 발생시킬 시간으로 리셋되고 타이머가 만료될 때까지 일정 시간동안 패킷을 모으는 경우이며, 상황 2는(Rx/Tx)IntDelay 타이머가 만료되기 전에 패킷이 도착하면 도착 할 때마다 (Rx/Tx)IntDelay 타이머가 리셋 되어 마지막 패킷을 받은 후 다음 인터럽트를 발생시킬 시간까지 패킷을 모으는 경우를 보이고 있다. 3가지 매개



(Figure 5) Interrupt Coalescing Parameters

변수 중에서 하나라도 사전 설정된 조건을 만족하면 인터럽트가 발생한다. 본 논문에서 제안하는 적응적 인터럽트 병합에서는 인터럽트 발생 조건으로 ITP만을 사용한다.

e1000e 드라이버는 인터럽트 병합을 위한 정책이 단순하며 4가지 모드를 제공한다(GBE, 2015). 첫 번째로 Off 모드는 인터럽트 병합을 하지 않는다. 두 번째로 Dynamic 모드는 응답속도가 빠를 경우 (Low Latency)에만 ITP 값을 $50\mu s$ 로 설정한다. 세 번째로 Dynamic conservative 모드는 네트워크 부하가 갑자기 증가하거나 링크 속도가 1G bps 미만이면 ITP 값을 $250\mu s$ 로 설정하고, 응답속도가 빠르면 ITP 값을 $50\mu s$ 로 설정한다. 마지막으로 Simplified balancing 모드는 네트워크 부하에 따라 ITP 값을 $125\mu s$ 에서 $500\mu s$ 까지 설정한다. 인텔 e1000e 장치 드라이버는 기본 모드로 Dynamic conservative 모드를 사용한다.

2.4.1 인터럽트 병합의 문제점

IEEE 802.3az 표준을 따르는 100M bps 네트워크 인터페이스 카드는 기본 ITP 값이 $250\mu s$ 로 고정되어 있다. 따라서 패킷 병합과 같이 인터럽트 발생 주기 시간이 고정되어 효율적인 인터럽트 병합을 어렵게 한다. 예를 들어 100M bps에서 $250\mu s$ 주기로 패킷을 1,500바이트씩 전송하는 응용 프로그램이 있다면, T_{w_system} 이 $30\mu s$, T_{pkt} 가 $120\mu s$, T_s 가 $200\mu s$ 이므로 하나의 패킷을 보내는데 최소 $350\mu s$ 이 소요되어 하나의 ITP로는 패킷을 병합하지 못한다.

3. 인터럽트 병합 최적화

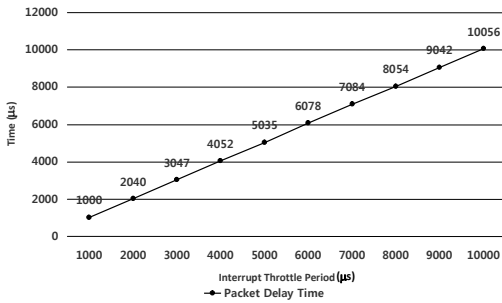
IEEE 802.3az 표준은 LPI 모드로 전이하고 다시 빠져 나오는 시점에 대한 정책은 기술하지 않고 있다(IEEE, 2010; Mostowfi and Christensen, 2011). 따라서 LPI 모드 전이에 대한 정책은 정해진 것이 없다. 기존의 인터럽트 병합 구현 방법은 고정된 몇 개의 병합 임계시간을 기반으로 한 단순한 정책을 적용하고 있어 최적의 에너지 효율성을 얻기 어렵다. 또한 에너지 효율 개선은 성능 저하를 야기하는 문제가 있어(Christensen et al., 2010) 인터럽트 병합 정책은 에너지 효율성과 더불어 성능 요구사항도 만족할 수 있어야 한다. 본 장에서는 인터럽트 병합 정책의 최적화를 통해 에너지 효율성과 성능 요구사항을 모두 고려한 적응적 인터럽트 병합(Adaptive Interrupt Coalescing, AIC)과 구현에 대하여 기술한다.

3.1 적응적 인터럽트 병합

적응적 인터럽트 병합은 성능 요구사항을 고려하면서 최고의 에너지 효율성을 높일 수 있도록 인터럽트 병합 정책을 최적화하기 위해 인터럽트마다 인터럽트 주기 즉 ITP를 조정할 수 있도록 한다. 따라서 각 인터럽트는 자신만의 ITP를 가지게 되며 서로 다른 에너지 소모와 패킷 지연시간을 보이게 된다. 적응적 인터럽트 병합은 가장 최근의 정확한 상황 정보를 기반으로 최적의 결과를 얻을 수 있도록, 현재 인터럽트에서 처리되는 패킷 수와 패킷 지연시간을 바탕으로 다음 인터럽트의 ITP가 결정되도록 한다.

3.2 병합 알고리즘 및 구현

적응적 인터럽트 병합은 에너지 효율성과 더불어 성능 요구사항도 고려한다. 패킷을 송/수신할 때의 성능은 패킷 송/수신 지연시간으로 측정할 수 있다. 본 논문은 적응적 인터럽트 병합의 초기 연구로써 효율성을 검증하기 위한 초기 실험을 패킷 크기는



<Figure 6> ITP versus Packet Delay

1,500 바이트로 고정하고 인텔 82579 NIC와 e1000e 드라이버를 사용하여 100M bps 대역폭(bandwidth)에서 송신측을 대상으로 수행하였다. 실험 결과에 의하면 <Figure 6>에서 보는 것처럼 적응적 인터럽트 병합에서의 패킷 지연시간은 인터럽트 주기인 ITP에 거의 선형으로 비례한다. 따라서 D 가 패킷지연 시간일 때 아래와 같은 식을 도출 할 수 있다.

$$D \cong ITP$$

NIC에서 시작시간 s 부터 종료시간 e 까지 소모된 에너지 사용량은 다음 식으로 계산 한다.

$$E_{[s,e]} = \left(\frac{I-L}{2} + 2 \right) \sum_{i=1}^n (T_{w_i} + T_{s_i}) + A \sum_{i=1}^n T_{pkt_i} + I \sum_{i=1}^n T_{id_i} + L \sum_{i=1}^n T_{q_i}$$

- $E_{[s,e]}$: 시작시간 s 부터 종료시간 e 까지 소모된 총 에너지 사용량
- A : Active 모드의 에너지 소모량
- I : Idle 모드의 에너지 소모량
- L : LPI 모드의 에너지 소모량
- T_{w_i} : $[s, e]$ 구간에서 i 번째 Wake 시간
- T_{s_i} : $[s, e]$ 구간에서 i 번째 Sleep 시간
- T_{pkt_i} : $[s, e]$ 구간의 i 번째 인터럽트에서 패킷 처리시간

- T_{id_i} : $[s, e]$ 구간에서 i 번째 Idle 시간
- T_{q_i} : $[s, e]$ 구간에서 i 번째 LPI 시간
- n : $[s, e]$ 구간에서의 인터럽트 발생 횟수

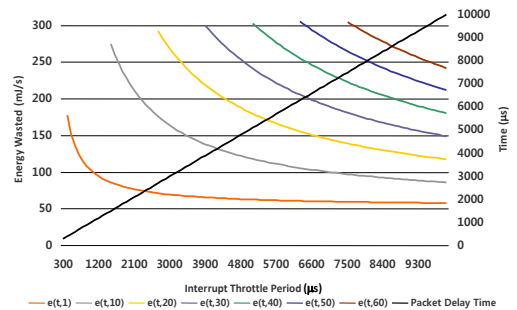
본 논문에서의 에너지 사용량 계산은 인텔 82579 사양(Intel 82579, 2011)에 주어진 <Table 2>를 참조하였다.

1초 동안 측정된 결과를 토대로 얻은 총 에너지 사용량과 패킷 지연시간을 하나의 그래프로 나타내면 <Figure 7>과 같다. ITP 값을 t , 한 번의 인터럽트에 처리되는 패킷 개수를 p 이라고 할 때, 왼쪽의 y 축은 t 와 p 에 따른 네트워크 장치 에너지 소비량 $e(t,p)$ 를 나타내고, 오른쪽의 y 축은 t 에 따른 패킷 지연시간 $d(t)$ 를 나타낸 것이다. $e(t,p)$ 인 경우 ITP 값이 줄어들수록 즉, 1초 동안에 인터럽트 발생횟수가 증가 할수록 패킷 전송량은 많아지기 때문에 에너지 소비량은 증가함을 보이고 있다.

위에서 설명한대로 성능을 고려한 최적의 에너지 절감방법을 찾기 위해서는 패킷 지연시간과 에너지의 두 요소를 모두 고려하여야 한다. 본 연구는 두

<Table 2> NIC Power Consumption

Mode	Power
Active	315mW
Idle	207mW
LPI	53mW



<Figure 7> Experimental Results on Energy Waste and Packet Delay

요소를 모두 고려한 총 비용 계산 모델로 다음 식을 사용한다.

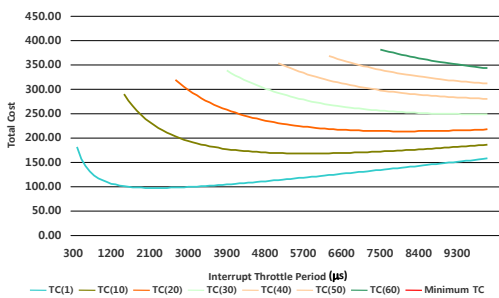
$$TC = \alpha D + E_{[s,e]}$$

- TC : 에너지와 패킷 지연시간의 합
- α : 패킷 지연시간의 가중치
- D : 패킷 지연시간

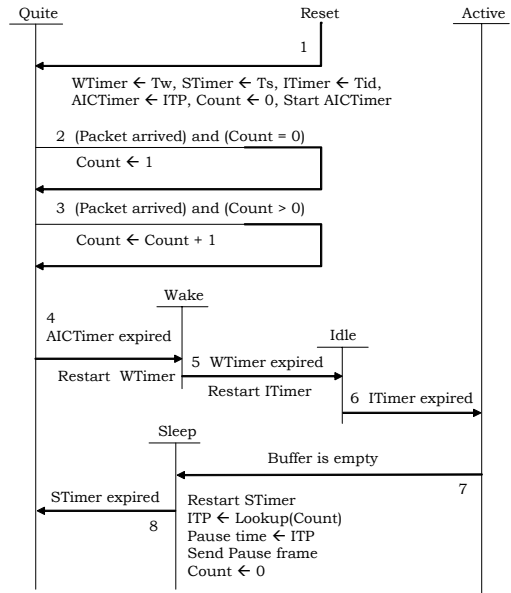
α 의 값은 0.01부터 0.1까지 0.01 간격으로 10개 로 설정하였다.

α 가 어떤 특정 가중치 값으로 주어졌을 때 현재 인터럽트에서 처리되는 패킷 수, 즉 패킷 전송량마다 TC 값이 계산되며, 하나의 특정 패킷 전송량에 대한 TC 값 중 최소 값을 가지는 ITP가 다음 인터럽트의 최적 ITP가 된다. <Figure 8>은 가중치가 0.01로 주어졌을 때 1초 동안 실험한 결과로 얻은 TC 그래프이다. <Figure 8>에서 $TC(i)$ 는 하나의 인터럽트에서 전송된 패킷 수가 i 일 때의 TC 를 나타내며, TC 최소값은 각 $TC(i)$ 에서의 최소 값을 이은 선을 나타낸다.

본 연구에서는 적응적 인터럽트 병합 구현시 다음 인터럽트의 최적 ITP를 신속히 찾기 위해 패킷 지연시간의 각 가중치마다 Lookup 테이블을 생성한다. 하나의 인터럽트에 전송 가능한 실제 최대 패킷 수는 60개이므로 Lookup 테이블은 60개의 항목을 가지고 있으며, 각 항목은 패킷 수와 최적 ITP로 구성된다.



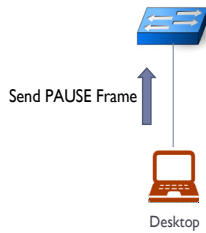
<Figure 8> Experimental Results on Total Cost where $\alpha = 0.01$



<Figure 9> FSM of Adaptive Interrupt Coalescing

<Figure 9>는 적응적 인터럽트 병합 알고리즘을 보여주는 FSM(Finite State Machine)이다. 처음 시작하면 모든 타이머와 패킷 카운터가 각각의 설정 값으로 초기화되고, 시스템이 LPI 모드로 전이하면서 AIC 타이머가 시작된다. 그 후 첫 번째 패킷이 도착하면 패킷 카운터가 1로 초기화된다. AIC 타이머가 만료되면 NIC가 활성화되어 모든 패킷을 전송하고, 다음 인터럽트의 ITP를 Lookup 테이블에서 찾은 최적 ITP로 설정하며, PAUSE 프레임은 스위치로 보내고, 패킷 카운터를 0으로 설정한다. 모든 패킷을 다 보내서 버퍼가 비게 되면 LPI 모드로 전이하면서 S 타이머를 시작한다.

적응적 인터럽트 병합을 구현하려면 컴퓨터의 NIC와 스위치(또는 라우터 등)의 동기화가 필수적이다<Figure 10>. 컴퓨터의 TCP/IP 라이브러리에서 패킷(데이터)을 병합하여 LPI 모드 시간을 증가시켜도 스위치로부터 어떤 패킷에 대한 수신(ACK)메시지가 도착하게 되면 NIC는 다시 Active 모드로 변경된다. 따라서 NIC가 일정시간 LPI 모드 상태로 들어가게 되면 LPI 모드를 유지해야 하는 시간 정보를 스위치에게 전달하여 LPI 모드 시



〈Figure 10〉 Synchronization between a Desktop and a Switch

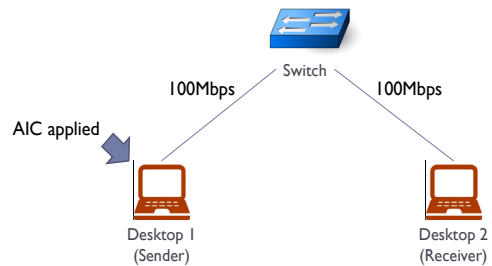
간동안 스위치가 패킷을 전송하지 못하도록 하여야 한다. PAUSE Frame(IEEE Supp., 1993)은 이더넷 2계층 프로토콜로써 TCP/IP 흐름제어에서 사용되며, PAUSE Frame을 수신한 송신측은 일정 시간동안 패킷 전송을 하지 않는다. 따라서 PAUSE Frame을 장치 드라이버에서 RAW 패킷으로 생성하여 스위치로 전송하면 LPI 모드 시간동안 스위치가 패킷을 전송하지 않아 NIC에서 LPI 모드 유지 시간을 보장 받을 수 있다.

4. 실험 결과

본 연구는 적응적 인터럽트 병합의 효율성에 대한 초기 연구로써 장치 드라이버에서의 프로토콜 구현 난이도 및 복잡성을 고려하여 일단 송신측에 대한 연구로 국한하였다. 다음은 적응적 인터럽트 병합의 실험 환경 및 실험 결과에 대한 설명이다.

4.1 실험 환경

송신측 적응적 인터럽트 병합 실험을 위한 실험 환경은 〈Figure 11〉에서 보는 것처럼 패킷 전송 컴퓨터와 수신 컴퓨터 그리고 두 컴퓨터를 연결하는 스위치로 구성된다. 전송 컴퓨터에서 사용된 네트워크 장치는 인텔 82579LM NIC이며, 장치 드라이버는 e1000e Linux 장치 드라이버이다. 네트워크 전송률은 100M bps로 설정하여 실험을 수행하였다. 장치 드라이버에서는 다른 네트워크 병합 기술을 제한하기 위해 Scatter Gather, TCP Seg-



〈Figure 11〉 Network Configuration for Experiments

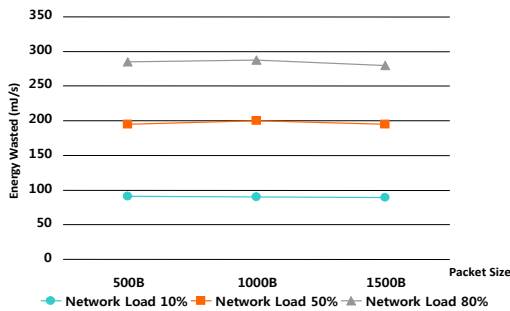
mentation offload, General Segmentation offload에 대한 옵션 설정을 해제하였다. Scatter Gather 옵션은 MTU 크기(1,500B) 내에서 데이터를 병합하고 TCP Segmentation offload, General Segmentation offload 옵션은 패킷크기를 MTU크기 이상인 65,535바이트 까지 증가시키기 때문이다.

실험은 패킷 크기와 네트워크 부하에 따른 영향을 주기적 패킷 전송과 임의적 패킷 전송으로 나누어 수행하였으며, 실험 결과가 방대하여 다음 7가지 경우만을 논한다.

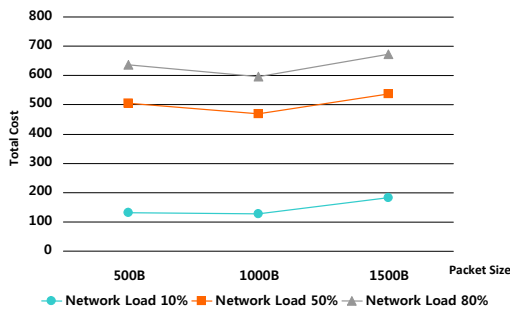
- No EEE+No IC : 에너지 절감을 하지 않는 기존 기술
- EEE+No IC : 인터럽트 병합 기술이 적용되지 않은 에너지 효율적 이더넷
- EEE+IC : 인터럽트 병합 기술이 적용된 에너지 효율적 이더넷
- EEE+AIC2, EEE+AIC5, EEE+AIC8 : 적응적 인터럽트 병합 기술이 적용된 에너지 효율적 이더넷. AIC2, AIC5, AIC8은 총 비용 식에서 패킷 지연시간의 가중치를 각각 0.02, 0.05, 0.08을 사용한 경우임.
- Ideal : 에너지 소모가 네트워크 부하에 정비례하는 이상적인 경우

4.2 패킷 크기에 따른 실험 결과

〈Figure 12〉와 〈Figure 13〉은 테스트 프로그램에서 주기적으로 전송할 패킷 크기를 각각 500,



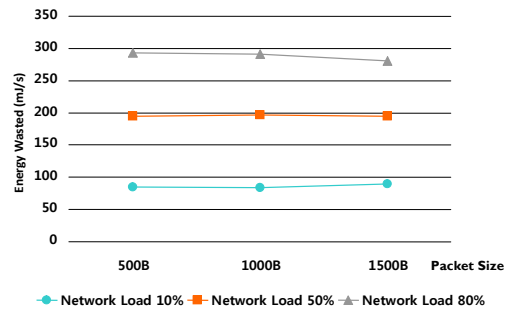
<Figure 12> EEE+AIC2 Power Consumption for Periodic Packet Transmission



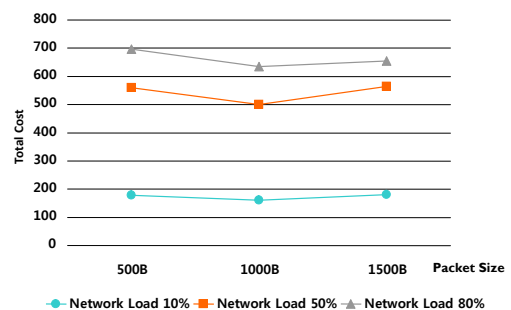
<Figure 13> EEE+AIC2 Total Cost for Periodic Packet Transmission

1,000, 1,500바이트로 설정하고 패킷을 전송했을 때 측정시간 1초 동안 EEE+AIC2에서의 에너지 소비량과 총 비용(Total Cost)을 보여준다. <Figure 12>에서 보이는바와 같이 한 번에 전송하는 패킷 크기가 달라졌음에도 불구하고 네트워크 부하에 따른 에너지 소비량은 비슷한 결과를 나타내고 있다. 그러나 에너지 소비량과 패킷 지연시간을 합한 <Figure 13>의 TotalCost를 보면 패킷 크기 1,000 바이트에서 가장 낮은 값을 기록하고 있다. 이 결과는 주기적으로 패킷을 전송할 때 전송하는 패킷 크기가 1,000바이트 정도 일 때 가장 성능이 좋다는 것을 의미한다.

<Figure 14>와 <Figure 15>는 테스트 프로그램에서 전송주기를 임의적으로 설정하고 전송할 패킷 크기를 각각 500, 1,000, 1,500바이트로 패킷을 전송했을 때 측정시간 1초 동안 EEE+AIC2에서의 에너지 소비량과 총 비용을 보여준다. 주기적으로



<Figure 14> EEE+AIC2 Power Consumption for Random Packet Transmission

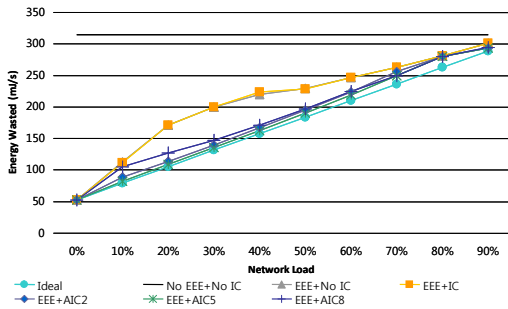


<Figure 15> EEE+AIC2 Total Cost for Random Packet Transmission

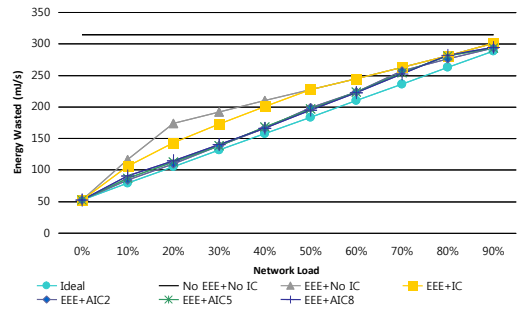
패킷을 전송하는 <Figure 12>와 같이 에너지 소비량은 전송 패킷 크기에 상관없이 네트워크 부하에 따라 증가하는 것을 볼 수 있다. <Figure 13>의 TotalCost와 <Figure 15>의 TotalCost 그래프는 유사하다. 즉, 주기에 상관없이 패킷 크기가 1,000 바이트일 때 가장 좋은 성능을 나타내는 것을 알 수 있다.

4.3 네트워크 부하에 따른 실험 결과

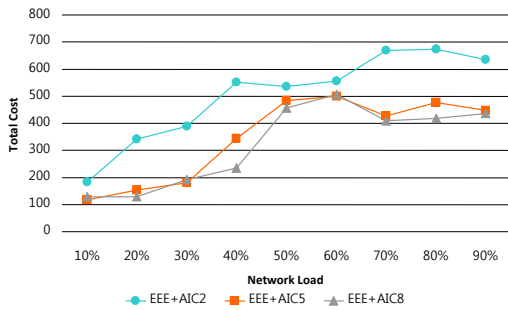
<Figure 16>과 <Figure 17>은 주기적으로 패킷을 전송하는 상황에서 네트워크 부하별 에너지 사용량과 총 비용을 보이고 있다. 한 번에 전송하는 패킷 크기는 1,500바이트다. No EEE+No IC에 해당하는 기존 네트워크 장치는 항상 전체 파워를 소모하기 때문에 네트워크 부하에 상관없이 315 mJ/s의 에너지를 사용하고 있다. Ideal 경우는 부



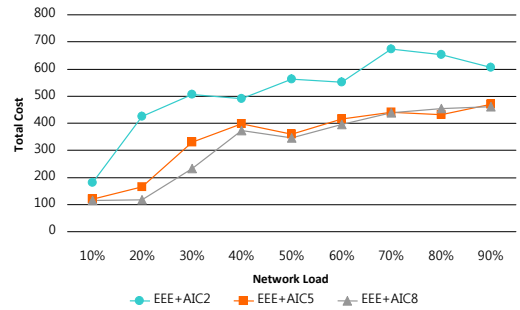
<Figure 16> Power Consumption for Periodic Packet Transmission



<Figure 18> Power Consumption for Random Packet Transmission



<Figure 17> Total Cost for Periodic Packet Transmission



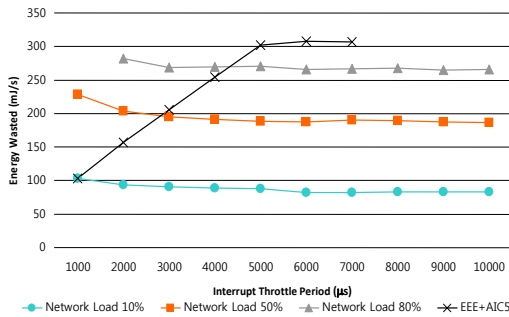
<Figure 19> Total Cost for Random Packet Transmission

하가 0%일 때 53mJ/s부터 시작해 에너지가 선형적으로 증가하는 이상적인 그래프다. <Figure 16>은 No EEE+No IC와 Ideal을 제외한 나머지 5개의 그래프가 관심 대상이다. 에너지 효율적 이더넷만 사용하는 EEE+No IC와 에너지 효율적 이더넷과 인터럽트 병합을 사용하는 EEE+IC의 에너지 소비량이 비슷하게 나온 이유는 위에서 설명했듯이 기본 인터럽트 병합에서 ITP값이 너무 작아 인터럽트를 병합할 시간이 충분치 못했기 때문이다.

0% 이후에는 EEE+AIC5가 가장 에너지를 적게 소비하고 있고, AIC 중에서는 EEE+AIC8가 에너지를 가장 많이 소비하고 있다. 하지만 차이는 미미한 수준이다. 네트워크 부하 40%에서 EEE+AIC5가 EEE+IC보다 최대 37% 에너지를 절감하는 것으로 나타났다. 그러나 에너지 소비율과 패킷 지연시간을 합한 <Figure 17>의 TotalCost 결과를

보면 EEE+AIC8이 가장 적은 값을 기록하고 있다. 이 결과는 EEE+AIC8이 AIC 중에서는 에너지를 가장 많이 소비했지만, 패킷 지연시간이 적어 패킷 지연시간을 고려하는 상황이라면 EEE+AIC5보다 EEE+AIC8이 더 적합하다는 것을 의미한다. 반대로 EEE+AIC2의 TotalCost가 EEE+AIC5나 EEE+AIC8의 TotalCost 보다 상당히 큰 것을 볼 수 있다. 이 결과는 EEE+AIC2의 경우 에너지는 상당히 절감 했지만, 패킷 지연시간이 커져 EEE+AIC5나 EEE+AIC8보다 성능이 떨어지는 것을 의미한다.

<Figure 18>과 <Figure 19>는 임의적 주기로 패킷을 전송하는 상황에서 네트워크 부하별 에너지 사용량과 TotalCost를 보이고 있다. 한 번에 전송하는 패킷 크기는 1,500바이트다. <Figure 18>에서 <Figure 16>과는 다르게 EEE+IC가 EEE+No IC보다 에너지를 적게 소비하는 것을 볼 수 있다. 주기적으로 패킷을 전송 할 경우 인터럽트 병합



<Figure 20> Power Consumption over ITP and Network Load

시간보다 패킷 전송 주기가 크기 때문에 인터럽트 병합의 기회를 놓치게 되지만, 패킷 전송 주기가 임의의 시간을 갖게 될 경우에는 주기를 놓치지 않고 인터럽트 병합의 기회를 갖게 되어 에너지를 절감할 수 있다. 임의적 주기로 패킷을 전송하는 경우에는 AIC 대부분이 높은 에너지 절감율을 보이고 있다. TotalCost는 <Figure 17>과 마찬가지로 EEE+AIC2가 가장 높은 값을 나타내고 있다. EEE+AIC5와 EEE+AIC8은 비슷한 TotalCost를 나타내고 있지만 EEE+AIC8이 대부분의 구간에서 최소 값을 나타내고 있다.

위 결과를 종합하면 인터럽트 병합 주기를 패킷 전송량(네트워크 부하)에 따라 결정하는 것이 에너지 절감에 효율적임을 알 수 있다(<Figure 20>). 인터럽트 병합 주기를 고정적으로 결정하는 경우 네트워크 부하 증감에 따라 에너지를 절감 할 수 있는 인터럽트 병합 주기가 있음에도 불구하고 고정된 인터럽트 병합 주기 때문에 에너지를 절감 할 수 없다. 예를 들어, 인터럽트 병합주기를 1,000µs로 고정하는 경우 네트워크 부하 50%에서 228.64 mJ/s의 에너지를 소비한다. 그러나 인터럽트 병합 주기가 3,000µs 이상에서는 195.39mJ/s의 에너지를 소비한다. 약 17%의 에너지 절감기회를 놓치고 있다. 반면에 EEE+AIC5의 경우 네트워크 부하에 따라 인터럽트 병합주기를 결정하기 때문에 네트워크 부하 50%에서 EEE+AIC5는 평균 ITP가 2,700 µs이다. 이때 EEE+AIC5는 187.92mJ/s의 에너지

를 소비한다. 인터럽트 병합주기 1,000µs 보다 21%의 에너지를 절감하고 있다. 따라서 인터럽트 병합 주기를 고정하는 것보다는 인터럽트 병합주기를 네트워크 부하에 따라 인터럽트 병합주기를 결정하는 것이 에너지 절감에 도움이 된다.

5. 결 론

정보기술의 발전은 다양한 IT 서비스(Sim et al., 2015; Augustsson et al., 2015; Kok et al., 2015)를 가능하게 하였으나 에너지 사용이 급속이 증가하였다. 따라서 네트워크에서의 에너지 절감을 위한 노력으로 2010년 9월에 IEEE 802.3az 에너지 효율적 이더넷(Energy Efficient Ethernet, EEE) 표준안이 제정되었다. 그러나 에너지 효율적 이더넷은 데이터가 버스트 특성을 가질 때는 에너지 절감율이 크지만 모드 전이(mode transition) 오버헤드가 커져 패킷의 크기가 작고 주기적으로 전송되는 상황에서는 잦은 모드 전이로 인해 에너지 절감율이 감소하는 문제점이 있다. 이와 같은 문제를 해결하기 위해 패킷 병합(packet coalescing)이나 인터럽트 병합(interrupt coalescing)과 같은 병합 기술이 제안되었으나 임계시간, 임계패킷 수, 인터럽트 발생 주기와 같은 병합 변수가 미리 고정적으로 설정되어 있어 상황에 따라 최적으로 병합을 수행하지 못하여 에너지 절감율이 떨어지는 문제가 있다.

본 논문에서 제안한 적응적 인터럽트 병합은 에너지 절감에 최적의 인터럽트 발생 주기를 찾을 수 있도록 하여 기존 병합의 에너지 절감율을 개선하였다. 또한 에너지 절감과 더불어 요구되는 성능까지 고려하여 최적의 인터럽트 주기를 찾을 수 있도록 함으로써 보다 진일보한 에너지 절감 방안을 제시 하였다. 적응적 인터럽트 병합의 효율성을 입증하기 위해 인텔 82579 네트워크 인터페이스 카드와 e1000e Linux 장치 드라이버를 사용하여 송신측 적응적 인터럽트 병합을 구현하고 100M bps 전송율에서 실험하였다. 실험 결과는 적응적 인터럽트 병합이 기존 병합보다 대부분의 경우 에너지

절감율이 우수하고 최대 37%까지 에너지 절감율이 개선됨을 보였다. 그러나 본 연구는 적응적 인터럽트 병합에 대한 초기 연구에 불과하며, 기가비트 전송률로 송신 및 수신 양방향을 포함한 실험환경에서 적응적 인터럽트 병합의 효용성을 검증하는 등 앞으로 많은 연구가 필요하다.

References

- Augustsson, N., J. Holmstrom, and A. Nilsson, "From Technological Transitions to Service Transitions : A Study of Attenuation Effects in IT Service Provisioning", *Korean Society of IT Services*, Vol.14, No.2, 2015, 337-354.
- Christensen, K., P. Reviriego, B. Nordman, M. Bennett, M. Mostowfi, and J.A. Maestro, "IEEE 802.3 az : the Road to Energy Efficient Ethernet", *IEEE Communications Magazine*, Vol.48, No.11, 2010, 50-56.
- E1000, "Intel : Intel Gigabit Ethernet Driver", 2014, Available at <http://sourceforge.net/projects/e1000>(Accessed September 1. 2015).
- EEE, "Energy Efficient Ethernet IEEE 802.3az", 2010, Available at <http://standards.ieee.org/getieee802/download/802.3az-2010>(Accessed September 2. 2015).
- ETHTOOL, Available at <https://www.kernel.org/pub/software/network/ethtool/>(Accessed September 1. 2015).
- GBE, Available at <http://www.intel.co.kr/content/www/kr/ko/ethernet-controllers/gbe-controllers-interrupt-moderation-appl-note.html> (Accessed September 2. 2015).
- IEEE 802, Available at <http://www.ieee802.org/3/az/index.html>, 2010(Accessed September 1. 2015).
- IEEE Supp., IEEE Standards for Local and Metropolitan Area Networks : Supplements to Carrier Sense Multiple Access With Collision Detection(CSMA/CD) Access Method and Physical Layer Specifications, 1993.
- Intel 82579, "Intel@ 82579 Gigabit Ethernet PHY", 2011, Available at <http://www.intel.com/content/dam/doc/datasheet/82579-gbe-phy-datasheet-vol-2-1>(Accessed September 1. 2015).
- Kok, A., R. Esten, and R.W. Helms, "Knowledge Sharing in the New World of Work : Effects of the New Way of Working", *Korean Society of IT Services*, Vol.14, No.2, 2015, 315-336.
- Koomey, J., "Growth in Data Center Electricity use 2005 to 2010", *A report by Analytical Press, completed at the request of The New York Times*, 2011.
- Makineni, S., R. Iyer, P. Sarangam, D. Newell, L. Zhao, R. Illikkal, and J. Moses, *Receive Side Coalescing for Accelerating TCP/IP Processing*, High Performance Computing-HiPC 2006, Springer Berlin Heidelberg, 2006, 289-300.
- Mostowfi, M. and K. Christensen, "Saving Energy in LAN Switches : New Methods of Packet Coalescing for Energy Efficient Ethernet", *Green Computing Conference and Workshops (IGCC)*, 2011, 1-8.
- MSDN, Available at <https://msdn.microsoft.com/en-us/library/windows/hardware/ff556017%28v=vs.85%29.aspx>, (Accessed September 1. 2015).
- Prasad, R., M. Jain, and C. Dovrolis, "Effects of Interrupt Coalescence on Network Measurements", *Passive and active network measurement, Lecture Notes in Computer Science*, Vol.3015, 2004, 247-256.
- Sim, S.B., C.S. Yoo, and S.M. Hong, "Case Study and Performance Analysis of IT Convert-

- gence Business Models for Defense Sector”, *Korean Society of IT Services*, Vol.14, No.2, 2015, 275-288.
- (심승배, 유천수, 홍수민, “국방 IT융합 비즈니스 모델 사례와 성과 분석”, 한국IT서비스학회 지, 제14권, 제2호, 2015, 275-288.)
- Reviriego, P., K. Christensen, J. Rabanillo, and J. A. Maestro, “An Initial Evaluation of Energy Efficient Ethernet”, *IEEE Communications Letters*, Vol.15, No.5, 2011, 578-580.

◆ About the Authors ◆



Jaeyoul Lee (jylee195@gmail.com)

Jaeyoul Lee received a BSc and an MSc in Computer Science from Kookmin University. His research interests include information security, cloud computing, and operating systems.



Jaeil Han (jhan@kookmin.ac.kr)

Jaeil Han received a BSc in Mathematics from Yonsei University, and an MSc and a PhD from Syracuse University. He is currently a professor at School of Computer Science, Kookmin University. His current research focuses on energy-aware computing and IoT middleware. His research interests include service science, service computing, green computing, distributed systems, big data, middleware, object-oriented systems, security, and open software.



Young Man Kim (ymkim@kookmin.ac.kr)

Young Man Kim received a BSc in Mechanical Engineering from Seoul National University, and an MSc and a PhD from Ohio State University. He is currently a professor at School of Computer Science, Kookmin University. His current research focuses on computer network and middleware. His research interests include computer and network security, SaaS platform, and ubiquitous sensor network.