

무선 시스템 환경에서 대화형 응용을 위한 전력제어기법

민정희^o 차호정
연세대학교 컴퓨터과학과
{jhmin^o, hjcha}@cs.yonsei.ac.kr

Power Management Mechanism for Interactive Applications in Wireless Network Systems

Jung-hi Min^o, Hojung Cha
Dept. of Computer Science, Yonsei University

요 약

본 논문은 모바일 무선 시스템의 가용시간을 늘리고자 최근 사용량이 증가하고 있는 웹 응용으로 대표되는 대화형 응용을 사용할 때 시스템의 에너지를 효율적으로 줄일 수 있는 통합 전력 제어 기법을 제시한다. 기존의 방법들은 CPU와 WNIC의 소모 에너지 절감을 위하여 상호간에 영향이 없다는 가정하에 CPU와 WNIC에 대하여 각각의 정책을 수립하였다. 하지만 제시되는 매커니즘은 대화형 응용을 처리할 때 WNIC에서 얻을 수 있는 정보들을 CPU의 전압과 주파수 조절에 사용함으로써 시스템 레벨의 에너지 소모를 효율적으로 줄일 수 있다. 실험결과는 제시되는 매커니즘에 의해 기존의 CPU와 WNIC의 모드를 별개로 제어한 방법에 비해 평균 46%, 최대 62%의 소모 에너지 절감 효과를 보였다.

1. 서 론

최근 모바일 무선 시스템의 기능이 다양화 되고 기존의 통신 기능 외에 멀티미디어 응용, 웹 응용 등의 다양한 응용을 수행함에 따라 제한된 배터리 기반의 시스템임에도 불구하고 전력소모량이 증가하고 있다. 특히 모바일 시스템에서 웹 응용의 사용은 두드러진 증가세를 보이고 있다. 웹 응용으로 대표되는 대화형 응용은 사용자의 요청에 따라 데이터가 전송되고 처리되므로 사용자의 액티비티에 영향을 많이 받는다. 따라서 시스템 측면에서 전력 제어를 할 때 BE (Best Effort) 응용이나 CBR (Constant Bit Ratio) 응용에 비해 전력 제어 정책을 수립하기가 쉽지 않다 [1, 2, 7]. 본 논문은 대화형 응용에 대해 모바일 시스템의 요소 중 CPU와 WNIC (Wireless Network Interface Card)에 초점을 맞추어 응용의 특징을 활용하여 시스템의 전력 소모를 줄일 수 있는 효율적인 방법을 제시한다.

기존의 연구들은 대화형 응용에 대해 전력 제어를 할 때 CPU와 WNIC 상호간에 영향이 없다는 가정하에 CPU를 위한 DVFS (Dynamic Voltage and Frequency Scaling) 기법과 WNIC을 위한 DPM (Dynamic Power Management) 기법을 CPU와 WNIC을 위해 각각 제시하였다 [3 - 6, 11]. 또한 시스템을 위한 전력 제어 기법을 구성할 때도 CPU는 되도록 낮은 전압과 주파수로 동작시

키는 것이 유리하다는 가정하에 CPU와 WNIC에 대한 각각의 기법들을 단순히 결합하는 수준에 머물렀다 [7, 9, 12].

하지만 본 논문에서는 모바일 무선 시스템에서 대화형 응용을 사용할 때 CPU와 WNIC의 소모전력을 분석하고 상호간에 영향을 끼치는 요인을 적용하여 시스템 레벨에서 전력 소모를 줄일 수 있는 기법을 제시한다.

논문의 구성은 다음과 같다. 2장에서는 논문에서 제안하는 모바일 시스템 기반 대화형 응용을 위한 통합 전력 제어 기법에 대해 기술한다. 3장에서는 실험 방법 및 결과에 대해 기술하고 4장에서는 결론을 맺는다.

2. 대화형 응용을 위한 통합 전력 제어 기법

본 논문에서는 WNIC에서 수집되는 정보를 이용하여 CPU의 전압과 주파수를 조절하는 방법을 제시한다. WNIC에서 얻을 수 있는 정보는 대역폭 (bandwidth) 사용률, listen interval, 수신 소요시간 (elapsed time), 요청 간격 (request interval), 수신된 패킷 크기 (packet size), 버스트레벨 (burst level) 등이 있다 [1, 8, 10]. 위의 정보를 통하여 네트워크 트래픽을 분석하여 대화형 응용임을 알 수 있다. 대표적인 대화형 응용으로는 웹 응용과 텔레비전 응용이 있다.

제시되는 매커니즘은 2단계로 구성된다. 제 1단계는 모니터링을 통해 네트워크 트래픽을 대화형 응용으로 분류하는 단계이다. 제 2단계는 WNIC에서 수집된 정보를 토

본 연구는 서울시 산학연 협력사업 (과제번호: 10692)과 과학기술부 국가지정연구실사업 (과제번호: 2006-01546)으로 수행하였음

대로 효율적으로 WNIC과 CPU의 상호 영향을 고려하여 전력 모드를 설정하는 단계이다. 다음 식들은 제시되는 매커니즘의 제 1단계를 위하여 네트워크 트래픽을 분석하여 대화형 응용으로 분류하는 데 사용된다. 식 (1)은 사용된 대역폭과 최대 대역폭을 가지고 대역폭 사용률을 계산하는 방법이다. 일정하게 높은 대역폭 사용률을 보이는 ftp응용으로 대표되는 BE 응용과 일정하게 중간 이상의 대역폭 사용률을 보이는 streaming 응용으로 대표되는 CBR 응용과 비교하여 대화형 응용 중 웹 응용의 경우 불규칙하면서 표준편차가 크고 평균값은 중간 정도의 대역폭 사용률을 보이며 텔넷 응용의 경우 불규칙하면서 낮은 대역폭 사용률을 보이게 된다 [1, 4].

$$BandwidthUsageRatio = \frac{UsedBandwidth * 100}{MaximumBandwidth} \quad (1)$$

수신된 패킷 크기는 텔넷 응용보다는 웹 응용이, 웹 응용보다는 스트리밍 응용이, 스트리밍 응용보다는 ftp응용이 더 큰 경향을 나타낸다. 수신된 패킷의 크기는 모아서 전송되는 정도인 버스트레벨과 관계가 있다. 버스트레벨은 한 순간에 얼마나 많은 양의 데이터 통신이 이루어지는가를 나타낸다. 동일한 데이터를 전송받을 때 수신된 패킷의 크기가 작을수록 버스트레벨이 높아진다. 수신된 패킷의 크기가 클수록 데이터 전송 빈도와 버스트레벨이 낮아진다. 웹 응용의 경우 버스트레벨이 높은 특징을 보이고 텔넷 응용의 경우 낮은 버스트레벨을 보인다. 이는 데이터의 전송량과도 관계가 있다. 식 (2)는 버스트레벨을 계산하는 방법을 보여준다.

$$BurstLevel = \frac{ReceivedPacketSize}{MaximumPacketSize} \quad (2)$$

위와 같이 대역폭 사용률, 수신된 패킷 크기, 버스트레벨을 통하여 네트워크 트래픽이 대화형응용으로 분류된다면 제 2단계인 CPU와 WNIC의 효율적인 전력제어를 위해 WNIC으로부터 listen interval, 수신 소요시간, 요청간격의 정보를 추출한다. Listen interval을 조절할 때는 QoS가 보장되는 가를 고려해야 하는데 수신된 데이터에 따라서 QoS의 저하율이 다르다. [1]에 의하면 웹 응용에 대해 listen interval 500ms까지 QoS가 보장되며 표 1과 같이 본 실험을 통하여서도 확인하였다. 따라서 listen interval 조절은 최대 500ms까지로 한다. 식 (3)은 BLI (*Best listen interval*)을 구하는 방법이다. BLI를 결정하는 데는 요청간격 (Request Interval), 평균

표 1. listen interval 변화에 따른 QoS

Listen interval \ 웹 페이지 타입	100ms	400ms	800ms
텍스트위주 웹 페이지	100%	100%	89%
보통 웹 페이지	100%	100%	80%
이미지위주 웹 페이지	100%	100%	76%

수신 패킷 크기, 대역폭, 수신 소요시간 (Elapsed Time), 버퍼에 저장된 사용 가능한 패킷의 수 등이 사용된다. 요청간격을 RI , 수신 소요시간을 ET , 버퍼에 저장된 사용 가능한 패킷을 ABP 라 하자. c 는 비례를 나타내는 상수이다.

$$BLI = c * \frac{\#ofABP * PacketSize * RI * ET}{AvailableDataRate} \quad (3)$$

수신 소요시간은 request를 보낼 때 time stamp for sending을 찍고, 패킷이 들어왔을 때 time stamp for receiving을 찍어서 그 차이를 구한다. 수신 소요시간은 커널에서 제공하는 정보이다. 식 (4)는 사용자 인터랙티비티와 관계있는 요청간격을 구하는 방법이다. time stamp for sending을 TSS 라 하자. TSS_{pre} 는 바로 직전의 time stamp for sending을 의미한다.

$$RI = TSS - TSS_{pre} \quad (4)$$

그림 1은 WNIC 관련하여 추출한 정보를 사용하여 CPU와 WNIC 전력모드를 설정하는 매커니즘을 보여준다. 식 (3)에 의해 구해진 Best Listen Interval동안은 네트워크 트래픽이 없으므로 CPU를 lowVF (low Voltage and Frequency)로 돌리고 WNIC은 PSM (Power Saving Mode) 모드로 설정한다.

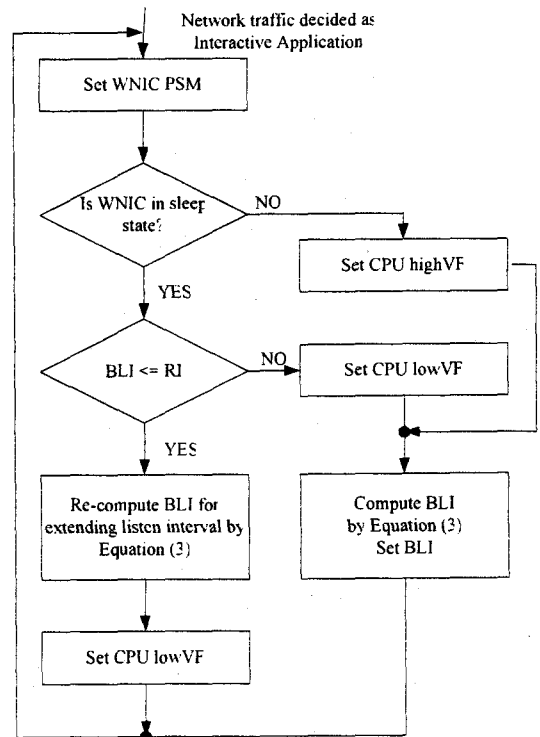


그림 1. 대화형응용을 위한 제시되는 CPU와 WNIC 전력 모드 제어 매커니즘

단, CPU를 lowVF로 구동하기 위하여는 책정된 listen interval동안 CPU를 lowVF로 구동시키는데 소모되는 전력 과 전압 전환에 드는 오버헤드인 전력의 합이 책정된 listen interval동안 highVF (high Voltage and Frequency)로 구동시키는 것보다 적어야 한다는 조건을 만족하여야 한다.

네트워크 트래픽이 있는 경우는 WNIC이 active한 상태를 의미하며, AP로부터 데이터 패킷을 받아 처리해야 하므로 CPU를 highVF로 설정한다. listen interval외에 요청 간격도 CPU와 WNIC 모드 설정에 영향을 끼치는 요소이다. 사용자의 요청간격은 개인에 따라 상황에 따라 다르므로 예측하기 쉽지 않다. 하지만 본 논문에서 제시하는 매커니즘은 CPU와 WNIC의 전력모드 설정에 초점을 두었으므로 BLI보다 사용자의 요청간격이 크거나 같은지만 검사하면 된다. 요청간격이 BLI보다 크거나 같다는 것은 WNIC이 PSM 모드로 sleep상태가 많음을 의미한다. 따라서 BLI를 사용자의 요청간격을 반영하여 BLI를 다시 계산해서 더 긴 시간동안 WNIC을 sleep상태로 운영한다. 또한 부가적으로 버스트레벨이 CPU의 모드 설정에 참고 자료로 사용될 수 있다. 버스트레벨이 높으면 전송되는 데이터 전송률이 높음을 의미하므로 CPU는 highVF, 버스트레벨이 낮으면 CPU는 lowVF가 유리하다.

3. 실험

본 논문에서 제안한 대화형 응용을 위한 통합 전력 제어 기법을 평가하기 위하여 Bulverde PXA270 Mainstone II를 사용하여 CPU와 WNIC에서 소모되는 전력을 각각 측정하였다. WNIC으로는 CISCO AIRONET350을 사용하였으며, 측정 도구로는 Agilent 34970A를 사용하였다. 실험 대상 응용으로는 웹넷 응용과 웹 응용을 사용하였는데, 웹 응용을 위해서는 텍스트 위주 웹 페이지, 보통 웹 페이지, 이미지 위주 웹 페이지를 대상으로 실험하였다. CPU는 104MHz (1.11V), 208MHz (1.24V), 312MHz (1.37V), 416MHz (1.52V)를 사용하였다.

그림 2는 웹넷 응용에 대한 listen interval의 변화에 따른 CPU와 WNIC의 에너지 소모의 합을 비교를 보여준다. 웹넷의 경우 CPU의 전압 및 주파수보다 listen interval의 영향이 더 큼을 알 수 있다. 또한 제시되는 BLI이 에너지 소모를 줄이는데 효율적임을 보여준다. 웹넷 응용의 경우 BLI로 listen interval을 조절할 경우 최대 28%까지 소모 에너지를 줄일 수 있었다.

그림 3는 웹 응용에 대한 제시된 매커니즘에 따른 CPU와 WNIC의 전력 소모량을 보여준다. 그림 3에서 볼 수 있듯이 QoS를 만족시키는 listen interval 조절에 대해 기존의 CPU는 lowVF로, WNIC은 PSM 모드로 설정하는 기법에 비해 오히려 CPU를 highVF로 운영하는 것이 CPU와 WNIC의 전력의 합에는 더 유리함을 볼 수 있다. 이는 network를 통해 들어오는 데이터를 빨리 처리하는 것이 WNIC이 active되어 있는 시간을 줄여주어 WNIC의 소모 전력 면에는 유리하기 때문이다. 제시되는 매커니즘에 의

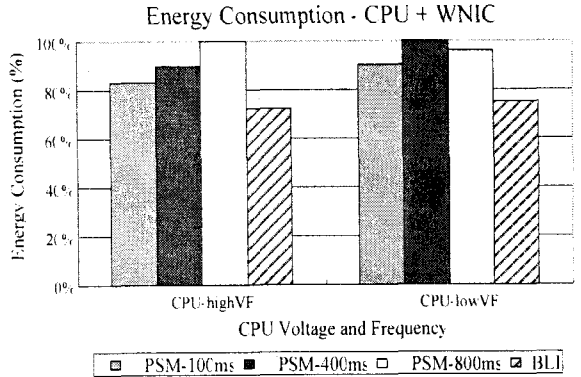


그림 2. 웹넷 응용에 대한 listen interval에 따른 에너지소모 비교

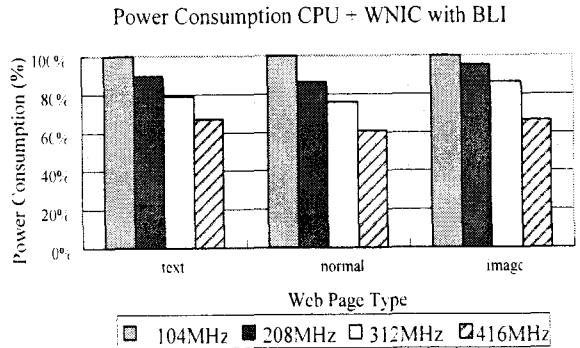


그림 3. 웹 응용에 대한 제시된 매커니즘에 따른 CPU와 WNIC의 전력소모의 합

해 BLI를 적용하였을 경우 CPU를 lowVF로 운영하는 경우에 비해 평균 35%, 최대 39%의 소모 전력 절감 효과를 보였다. 그림 4은 웹 응용에 대해 제시된 매커니즘에 따른 시간 소모 결과이다. 텍스트 위주 웹 페이지, 보통 웹 페이지, 이미지 위주 웹 페이지에 대해 공통적으로 CPU를 lowVF로 운영하는 것보다 highVF로 운영할 때 시간소모가 적음을 알 수 있다. 이는 소모 전력과 소모 시간이 적용되는 에너지 소모의 결과에 영향을 미친다.

그림 5는 웹 응용에 대한 제시되는 매커니즘에 따른 CPU와 WNIC의 에너지 소모의 합을 보여주고 있다. 실험 결과 CPU를 lowVF로 운영하는 경우에 비해 highVF로 운영하는 것이 오히려 평균 57%, 최대 62%까지 소모 에너지의 절감 효과를 보였다. CPU를 lowVF로 운영할 경우 WNIC을 통해 들어오는 데이터를 highVF에 비해 느리게 처리하게 되어 오히려 WNIC을 active한 상태로 오래 있게 만든다. 이는 WNIC에서 소모하는 전력과 시간을 증가시키는 결과를 초래한다. 따라서 CPU만 고려할 때는 lowVF가 소모 전력을 줄일 수 있지만 WNIC과의 소모 전력의 합을 고려하면 오히려 lowVF가 비 효율적이다.

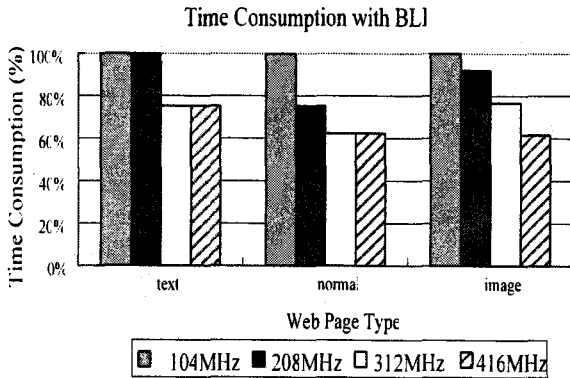


그림 4. 웹 응용에 대한 제시된 매커니즘에 따른 시간소모

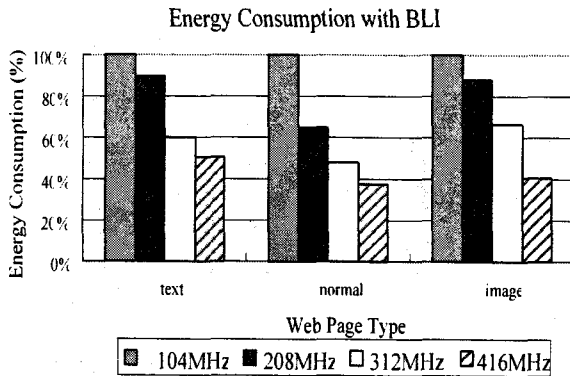


그림 5. 웹 응용에 대한 제시된 매커니즘에 따른 CPU와 WNIC 의 에너지소모의 합

4. 결론

본 논문은 모바일 시스템의 에너지소비를 효율적으로 줄이기 위하여 최근 사용량이 급증하고 있는 웹 응용을 핵심으로 대화형 응용을 처리할 때 효율적인 통합전력제어 기법을 제시한다. 우리는 본 논문을 통하여 기존의 연구가 고려하지 못했던 CPU와 WNIC 상호간의 영향을 분석하고 WNIC에서 수집되는 정보를 이용하여 CPU의 전압과 주파수를 조절하였다. 제시되는 매커니즘은 기존의 단순히 CPU와 WNIC 각각의 전력제어기법을 결합한 방법과 달리 대역폭 사용률, listen interval, 수신소요시간, 사용자 요청간격, 수신된 패킷 크기, 버스트레벨의 정보를 가지고 네트워크 트래픽을 대화형 응용으로 분류할 수 있었으며 CPU의 전압과 주파수를 조절하는 기준으로 사용하였다. 또한 WNIC의 sleep 상태를 늘릴 수 있도록 listen interval을 동적으로 제어하였다. 이를 통해 대화형 응용을 사용할 때 모바일 시스템의 구성 요소 중 CPU와 WNIC에서 소모하는 에너지를 평균 46%, 최대 62%까지 효율적으로 줄일 수 있었다.

참고문헌

[1] Weissel, A., Faerber, M., Bellosa, F., "Application Characterization for Wireless Network Power Management," Proceedings of the International Conference on Architecture of Computing Systems (ARCS'04), 2004.

[2] Chandra, S., Vahdat, A., "Application-specific Network Management for Energy-aware Streaming of Popular Multimedia Formats," Proceedings of USENIX Annual Technical Conference, 2002.

[3] AbouGhazaleh, N., Mayo, R. N., Ranganathan, P., "Idle time Power management for personal Wireless Devices," Technical Reports, HP laboratories, 2003.

[4] Jones, C. E., Sivalingam, K. M., Agrawal, P., Chen, J. C., "A survey of energy efficient network protocols for wireless networks," Wireless Networks, Vol.7, No.4, 343-358, 2001.

[5] Martin, T.L., "Balancing Batteries, Power and Performance: System Issues in CPU Speed-Setting for Mobile Computing.," Ph.D. thesis, Department of Electrical and Computer Engineering, Carnegie Mellon University, 1999.

[6] Flautner, K., Mudge, T., "Vertigo: automatic performance-setting for linux," Proceedings of the Fifth USENIX Symposium on Operating Systems Design and Implementation, 105-116, 2002.

[7] Zhong, L., Jha, N. K., "Dynamic Power Optimization of Interactive Systems," VLSI Design, 1041-1047, 2004.

[8] Hohlt, B., Doherty, L., and Brewer, E., "Flexible power scheduling for sensor networks," Proceedings of the Third International Symposium on Information Processing in Sensor Networks, Berkeley, CA, Apr. 2004.

[9] Sorber, J., Banerjee, N., and Corner, M. D., "Turducken: Hierarchical Power Management for Mobile Devices," Proceedings of The Third International Conference on Mobile Systems, Applications, and Services (MobiSys '05), Seattle, WA, June 2005.

[10] Qadeer, W., Rosing, T. S., Ankcorn, J., Krishnan, V., Micheli, G., "Heterogeneous wireless network management," Mobile and Media Systems Laboratory, HP Laboratories Palo Alto, HPL-2003-252, December 3, 2003.

[11] Flautner, K., Reinhardt, S., Mudge, T., "Automatic performance-setting for dynamic voltage scaling," Proceedings of the International Conference on Mobile Computing and Networking (MOBICOM-7), Rome, 260-271, 2001.

[12] Rakhmatov, D. S., Vrudhula, "Energy management for battery-powered embedded systems," ACM Transactions on Embedded Computing System 2 (3), 277-324, 2003.