무선 센서 네트워크를 위한 모바일 체크포인팅

이상호° 허준영 조유근
서울대학교 컴퓨터공학부
{shyi°, jyheo, cho}@ssrnet.snu.ac.kr

홍지만
광운대학교 컴퓨터공학부
gman@daisy.kw.ac.kr

Mobile Checkpointing for Wireless Sensor Networks

Sangho Yi° Junyoung Heo Yookun Cho
Dept. of Computer Science and Engineering, Seoul National University

Jiman Hong
Dept. of Computer Science and Engineering, Kwangwoon University

요 약

무선 센서 네트워크는 다양한 환경에서 자연의 정보를 수집하여 인간이 필요로 하는 형태로 정보를 제공하는 네트워크이다. 이러한 센서 네트워크는 수많은 무선 센서 노드들로 이루어지고, 각 센서 노드는 자율적으로 자연의 정보를 계측하고, 이웃 노드와 통신하며, 결함 발생시에 이를 허용하고 대처할 수 있어야 한다. 본 논문에서는 무선 센서 네트워크를 위한 모바일 체크포인팅 기법을 소개한다. 모바일 체크포인팅 기법을 통하여, 보다 안정적인 결함 허용 무선 센서 네트워크를 구현해 낼 수 있고, 실험 결과를 통하여 이 기법을 사용하였을 때에 전체 센서 네트워크가 보다 더 안정적으로 동작할 수 있음을 보인다.

1. 서 론

무선 센서 네트워크는 자연의 여러 가지 정보를 센서를 통하여 얻어내고, 이러한 정보를 수집하여 인간이 필요로 하는 형태로 정보를 가공하여 전달하여 주는 네트워크이다[1]. 이러한 무선 센서 네트워크는 수많은 무선 센서 노드들로 이루어진다. 각 센서 노드는 비용 적인 측면에서 매우 작은 크기로 구성되어야 전체 네트워크의 비용 효율성을 이루어낼 수 있게 된다. 이러한 센서 노드는 온도나 습도 및 광량 정보 등을 얻어낼 센서와 간단한 계산을 수행할 수 있는 CPU, 무선 통신을 위한 R/F 모듈, 부팅을 위한 작은 크기의 ROM 과 메인 배터리 등으로 구성 된다[2]. 이러한 센서 노드는 비용 효율적인 측면 때문에 안정한 저장장치를 갖는 것이 어려우며, 만약 안정한 저장장치를 갖게 된다면 센서 노드의 제작비용이 증가하므로 비용 효율성을 떨어뜨리게 된다[2].

각 센서 노드는 자율적으로 자연의 정보를 계측하고, 이웃 노드들과 무선으로 통신하며, 결함 발생시에 이를 허용하고 능동적으로 대처할 수 있어야 전체 센서 네트워크를 보다 더 안정적으로 동작시킬 수 있다. 센서 네트워크에 발생할 수 있는 결함은 매우 다양하다. 예를 들자면, 산불 감지를 위한 센서 네트워크에는 갑작스런 기상 악화, 혹은 지진 등의 자연 환경의 변화나, 센서 노드 자체의 결함으로 일시적인 리셋 현상이 발생할 수 있다[2]. 이러한 결함 발생을 자율적으로 대처하기 위한 결함 허용 방안으로는 여러 가지 있고, 이 중에서 가장 널리 쓰이는 방법으로 체크포인팅 기법이 있다.

체크포인팅은 상태 정보를 갖고 오랜 시간 동안 동작하는 프로세스를 위한 효과적인 결함 허용 기법으로, 프로세스의 상태 정보를 안정한 저장장치에 저장 후 복원하는 형태로 사용한다[3]. 현재까지 다양한 체크포인팅 기법이 연구 및 소개되었다[4-8]. 체크포인팅에서의 가장 큰 비중을 차지하는 디스크 쓰기 부하를 줄이기 위하여 압축 체크포인팅[4], 점진적 체크포인팅[5] 등이 연구되었고, 체크포인팅의 지연 시간을 최소화하기 위하여 포크 체크포인팅[6] 및 디스크 없는 체크포인팅 기법[7] 등이 연구되었다.

본 논문에서는 무선 센서 네트워크를 위한 모바일 체크포인팅 기법을 소개한다. 이 모바일 체크포인팅 기법은 기본적으로 디스크 없는 체크포인팅 기법의 연장선상에 있고, 추가적으로 무선 네트워크 상에서 체크포인팅을 수행하기 위한 알고리즘을 포함한다. 이 기법을 통하여, 보다 안정적인 결함 허용 무선 센서 네트워크를 구현해 낼 수 있고, 실험 결과를 통하여 이 기법을 사용하였을 때에 전체 센서 네트워크가 보다 더 안정적으로 동작할 수 있음을 보인다.

본 논문의 구성은 다음과 같다. 2절에서는 관련 연구를 보이고, 3절에서는 모바일 체크포인팅의 자세한 설명과 함께 알고리즘을 보이고, 4절에서는 제안한 기법의 성능 측정 결과를 보이고 이를 분석한다. 마지막으로 5절에서 본 논문의 결론을 내린다.

2. 관련 연구

현재까지, 효율성을 증진시키기 위하여 다양한 체크포인팅 기법이 연구되었다[4-8]. 이 절에서는 잘 알려진

체크포인팅 기법들을 소개한다.

Beck 등은 컴파일러의 도움을 받는 메모리 배제 체크포인팅 방법을 제안하였다[8]. 컴파일러의 자료흐름 분석을 통하여, 읽기-전용 메모리와 쓰기-전용 메모리를 구분함으로써, 저장하지 않아도 될 메모리를 배제시켜 체크포인팅 오버헤드를 줄이는 방법이다. Plank 등은 디스크 없는 체크포인팅 기법을 제안하였다[7]. 이 방법은 다중 프로세서를 갖는 환경에서, 안정한 저장장치로 디스크를 사용하지 않고 다른 프로세서의 메모리 공간을 사용하는 것이다. 이 방법을 사용하면 체크포인트 생성의 가장 큰 비중을 차지하는 디스크 쓰기 부하를 없애고 이를 메모리 복사로 대체하여 보다 나은 성능을 보일 수 있게 된다.

3. 무선 센서 네트워크를 위한 모바일 체크포인팅

본 절에서는 무선 센서 네트워크에서 자율적으로 결함 발생을 대처하기 위한 방안인 모바일 체크포인팅 기법을 제안한다.

무선 센서 네트워크는 상당히 많은 수의 센서 노드들을 필요로 하기 때문에, 센서 노드의 비용 효율성이 가장 중요하게 된다. 이러한 특성 때문에 센서 노드는 안정한 저장장치를 갖기 힘들게 되므로, 일반적인 체크포인팅 기법은 사용하기에 어렵게 된다. 따라서 본 논문에서는, 디스크 없는 체크포인팅 기법에 바탕을 둔 모바일 체크포인팅 기법을 제안한다. 아래의 그림 1은 무선 센서 네트워크의 한 예를 보여준다.

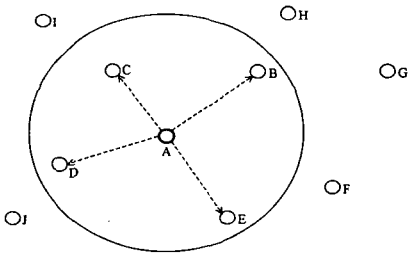

그림 1. 무선 센서 네트워크 예제

그림 1에서, A, B, C, D, E 는 각 센서 노드를 의미하고, A를 중심으로 한 원은 A의 무선 통신 반경을 의미한다. 이 그림에서 A의 이웃 노드는 B, C, D, E 가 되며, 각 노드는 자율적으로 잠자거나 깨어서 동작한다. 이 예제에서, A가 자연적 재해나 센서 노드 내부의 결함으로 인하여 일시적인 전력 공급 장애를 일으켰다고 가정하자. 이 때에 A는 리셋 될 것이며, 휘발성 메모리의 특성 때문에, A는 센서 노드로써의 기능을 상실하게 된다. 이러한 현상이 포아송 분포의 확률로 시간이 지남에 따라 빈번히 일어난다면, 전체 센서 네트워크를 이루는 대부분의 센서 노드들이 자신의 기능을 상실하여 센서 네트워크는 사용자가 원하는 대로 동작할 수 없을 것이다.

이렇게 결함을 내재한 센서 네트워크를 정상적으로 동작시키기 위하여 우리는 모바일 체크포인팅 기법을 사용할 수 있다. 이 기법은 체크포인팅 하는 센서 노드의 프로세스 이미지를 이웃한 노드들로 전달하고, 이웃 노드들이 필수적으로 깨어있어야 하는 시간을 정의하여, 결함 발생시에 이웃 노드로부터 체크포인트 이미지를 전달받아서 안정적인 상태로 복원할 수 있도록 한다.

위의 그림 1에서 A가 모바일 체크포인팅을 수행할 경우에 노드 A와 인접 노드들 간의 데이터 전달 과정을 나타내면 다음의 그림 2와 같다.

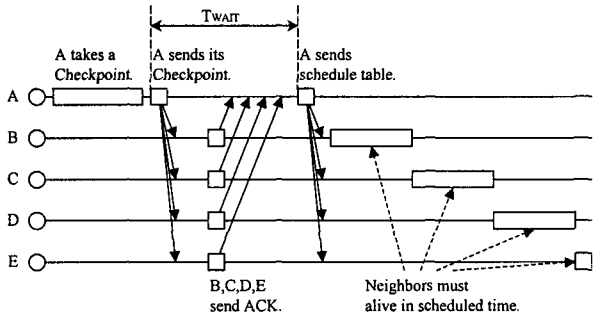

그림 2. 모바일 체크포인팅의 데이터 전달 과정

위의 그림 2에 보인 바와 같이, 먼저 A가 인접 노드들에게 자신의 체크포인트 이미지를 전송한다. 무선 채널의 특성상, 단 한번의 전송으로 통신 반경 안에서 현재 깨어있는 모든 센서 노드들에게 전송이 가능하므로 A의 체크포인트 이미지 전송은 단 한번 일어나게 된다[2]. 그 후에는 각 노드로부터 일정 시간동안 응답(ACK) 메시지를 기다린다. 마지막으로, 응답 메시지를 보내온 인접 노드들 중에서 적어도 하나 이상이 반드시 깨어있도록 하기 위하여, A가 정의한 스케줄 정보를 전송한다. 위의 그림 2에서는 B, C, D, E가 인접 노드들이 되고, 이 노드들이 라운드-로빈 형태로 스케줄 된다. 따라서 적어도 하나 이상의 인접 노드는 깨어있게 된다. 그림 3은 일련의 과정을 블록 다이어그램으로 나타내었다.

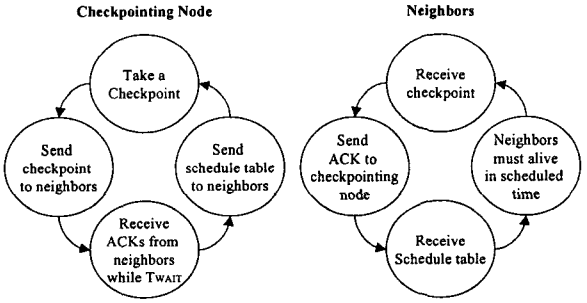

그림 3. 모바일 체크포인팅 블록 다이어그램

4. 성능 평가

본 절에서는 무선 센서 네트워크를 위한 모바일 체크포인팅 기법의 성능 측정 결과를 보인다. 무선 센서 네트워크를 시뮬레이션 하기 위하여 우리는 자체 제작한 센서 시뮬레이터를 사용하였고, 시뮬레이션과 관련된 파라미터들은 다음과 같다.

표 2. 시뮬레이션 파라미터

센서 필드	100 m * 100 m
센서 노드 수	100
센서 통신 반경	50 m
센서 초기 에너지	0.3 J
체크포인팅 패킷 크기	1024 바이트
센싱데이터 패킷 크기	64 바이트
응답메시지 패킷 크기	16 바이트
스케줄정보 패킷 크기	64 바이트
센싱 주기	1 초
체크포인팅 주기	100 초

위의 파라미터를 바탕으로 무선 센서 네트워크에서 모바일 체크포인팅 기법을 사용한 것과 사용하지 않은 것에 대하여 다음의 두 가지를 측정하였다. 먼저 정상적으로 작동하는 센서 노드들의 수의 변화를 시간에 따라 측정하였고, 그 다음으로는 전체 센서 네트워크의 에너지 잔량을 측정하였다. 이를 통하여 체크포인트를 사용함으로써 얻어지는 전체 시스템의 안정성과, 체크포인팅 비용의 관계를 비교할 수 있게 된다.

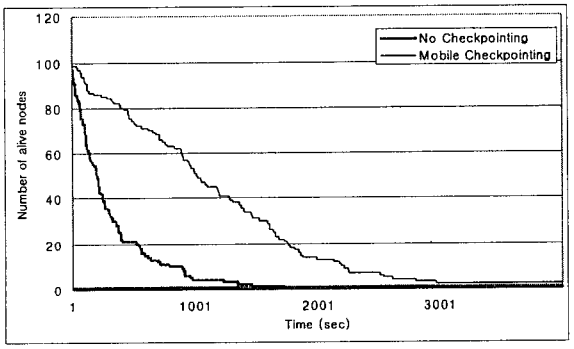

그림 4. 정상적으로 작동하는 노드 수의 변화

그림 4의 결과는 정상적으로 작동하는 노드 수의 변화를 나타낸 것이다. 이 결과를 통하여, 모바일 체크포인팅 기법을 사용한 경우에 그렇지 아니한 경우 보다 전체 네트워크가 보다 안정적으로 동작함을 볼 수 있다.

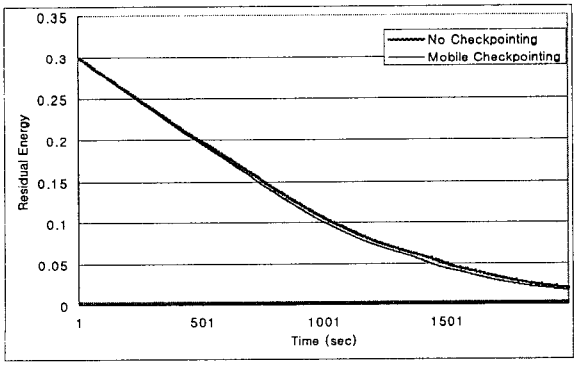

그림 5. 에너지 잔량의 변화

그림 5는 내재적 결함을 제거한 상태에서의 체크포인팅 수행비용을 시간에 따라 나타낸 것이다. 이 결과를 통하여, 체크포인팅을 사용하였을 때와 사용하지 않았을 때의 에너지 사용정도가 그리 크지 않음을 알 수 있었다. 따라서 약간의 체크포인팅 비용을 감내하고 모바일 체크포인팅 기법을 사용한다면, 내재적 결함을 가진 무선 센서 네트워크의 경우, 보다 더 안정적으로 동작하게 할 수 있게 된다.

5. 결론

무선 센서 네트워크는 다양한 환경에서 자연의 정보를 수집하여 인간이 필요로 하는 형태로 정보를 제공하는 네트워크이다. 이러한 센서 네트워크는 수많은 무선 센서 노드들로 이루어지고, 각 센서 노드는 자율적으로 자연의 정보를 계측하고, 이웃 노드와 통신하며, 결함 발생시에 이를 허용하고 대처할 수 있어야 한다. 본 논문에서는 무선 센서 네트워크를 위한 모바일 체크포인팅 기법을 소개하였다. 모바일 체크포인팅 기법을 통하여, 보다 안정적인 결함 허용 무선 센서 네트워크를 구현해 낼 수 있고, 우리의 실험 결과를 통하여 이 기법을 사용하였을 때에 전체 센서 네트워크가 보다 더 안정적으로 동작할 수 있음을 보였다. 만약 센서 네트워크가 더욱 더 많은 수의 센서 노드로 구성된다면 본 논문에서 제안한 모바일 체크포인팅 기법을 사용하였을 때, 센서 네트워크는 더욱 더 안정적으로 동작할 수 있을 것이다.

[참고문헌]

[1] Heinzelman, W.R., Chandrakasan, A., Balakrishnan, H. Energy-efficient communication protocol for wireless micro sensor networks, Hawaii International Conference on System Sciences, 2000.

[2] Anna Hac, Wireless Sensor Network Designs, John Wiley and Sons, 2003.

[3] E. N. Elnozahy, D. B. Johnson and W. Zwaenepoel, The Performance of consistent checkpointing, 11th symposium on reliable distributed systems, pp. 39-47, October 1992.

[4] Plank, J., Xu, J., Netzer, R., Compressed differences: An algorithm for fast incremental checkpointing, Technical Report UT-CS-95-302, 1995.

[5] Pank, J., Beck, M., Kingsley, G., Li, K., Libckpt: transparent checkpointing under unix, Usenix Winter Technical Conference. pp. 213-223, 1995.

[6] J. Hong, S. Kim, Y. Cho, Cost analysis of optimistic recovery model for forked checkpointing, IEICE Transactions on Information and Systems E86-D pp. 1534-1541, 2003.

[7] Plank, J., Li, K., Puening, M., Diskless checkpointing, IEEE Transactions on Parallel and Distributed Systems pp. 303-308, 1998.

[8] Plank, J., Beck, M., Kingsley, G., Compiler-assisted memory exclusion for fast checkpointing, IEEE Technical Committee on Operating Systems and Application Environments, Special Issue on Fault-Tolerance, pp. 62-67, 1995.