Journal of Korea Multimedia Society Vol. 8, No. 7, July 2005(pp. 953-960)

모바일 호스트의 이동성에 따른 안정적인 복구 기법

최가현†, 황병연††

요 약

모바일 컴퓨팅 환경은 네트워크가 손상되기 쉬우며 모바일 호스트(Mobile Host: MH)를 지원하는 모바일 서포트 스테이션(Mobile Support Station: MSS)이 안정적이지 못하다. 따라서 오류가 발생했을 경우, 저장된 정보를 복구하기 위해서는 많은 양의 네트워크 비용이 요구된다. 안정적인 복구를 위해서는 이미 보내진 메시지를 어떤 방식으로 저장할 것인가 하는 것과 어떤 핸드오프 기법을 사용해서 복구해 낼 것인가 하는 점을 고려해야 한다. 본 논문에서는 모바일 호스트의 이동성에 따라 복구 기법을 달리하는 ST(Switching Trickle) 기법을 제안한다. 제안된 기법의 성능평가를 위해 이동성에 따라 점차 증가하는 누적비용을 시뮬레이션 한다. 실험 결과를 통해 기존에 구현된 기법들과 비교하여 제안된 기법에서 전체적인 비용이 감소된 것을 확인할 수 있다. 뿐만 아니라 제안된 기법은 MH의 이동성이 크지 않을 경우, 체크포인트(checkpoint)와 메시지 로그 정보를 안정적으로 저장할 수 있다.

Stable Recovery Strategy According to Mobility of Mobile Host

Ga-Hyun Choi†, Byung-Yeon Hwang††

ABSTRACT

The Network can be easily damaged in the mobile computing environment. Mobile Support Station (MSS) which supports the Mobile Host(MH) could also be unstable too. Therefore, when a critical error occurs the repairing cost can be high. For stable repair, the hand-off strategy should be considered to save the continuing message and checkpoint information, and to reduce the cost of network repair. In this paper, we propose the Switching Trickle(ST) strategy which reduces the repairing cost for the MH's mobility. ST strategy changes the way which stores the information in connection with the number of write events(r) that occur from the hands-off strategy. Accumulated cost by increased mobility has been simulated too. From the results of simulation, the proposed strategy has shown more cost effective than the conventional strategy. The proposed strategy also has shown very supportive in saving checkpoint and message log information stably.

Key words: Recovery Strategy(복구 기법), Mobility(이동성), Mobile Host(이동 호스트)

1. 서 론

무선 서비스의 불안정한 요소 중 하나는 오류가 발생할 가능성이 많다는 것인데, 대부분의 시스템은 이 점을 보안하기 위해 체크포인트를 사용해서 복구하는 기법을 사용하고 있다. 체크포인트를 사용하는

※ 교신저자(Corresponding Author) : 황병연, 주소 : 경기도 부천시 원미구 역곡2동 산43-1(420-743), 전화 : 02)2164-4363, FAX : 02)2164-4777, E-mail : byhwang@catholic.ac.kr
접수일 : 2003년 1월 29일, 완료일 : 2003년 4월 14일
† 가톨릭대학교 컴퓨터공학과 졸업(석사)
(E-mail : namu25@dreamwiz.com)
†† 종신회원, 가톨릭대학교 컴퓨터정보공학부 교수
※ 본 연구는 2005년도 가톨릭대학교 교비연구비의 지원으로 이루어졌음.

방법은 일반적인 복구 방법이지만 무선 네트워크 상에서도 적용될 수 있는 방법이다. 모바일 호스트는 모바일 서포트 스테이션에 연결되어 있으면서 자신들의 변경 사항을 MSS에 보고하게 된다. 이런 MH는 셀(cell)이라고 하는 영역 안을 돌면서 메시지를 주고받는다[1,2]. 셀 영역을 벗어나게 되면 이동 위치 정보를 MSS에 보고한다. 이런 과정을 통해 MH와 MSS는 데이터의 무결성과 위치 정보의 투명성을 유지해 나간다[3,4].

무선 네트워크 상에서 끊김 연산은 자주 발생하기 때문에 대처 방안뿐만 아니라 이미 끊긴 상태를 어떻게 복구하느냐 하는 문제도 중요하다. 빠른 복구를 위해서는 앞에서 언급한 체크포인트 정보뿐만 아니라 메시지 로그 정보를 안전하게 기록하는 것도 중요하다. 복구 기법은 메시지 정보를 저장할 때 체크포인트 정보만을 저장할 것인지 아니면 그 외의 메시지 로그 정보를 저장할 것인지 선택하여 사용할 수 있으며, 핸드오프가 일어날 경우 어떤 정보를 저장하느냐에 따라 시스템의 성능이 달라질 수 있다.

복구 기법은 기본적으로 두 단계로 이루어지는데 상태 정보를 저장하는 기법[5]과 핸드오프가 일어날 때 어떤 방법으로 정보를 저장할 것인가에 따라 나눠지는 기법[6]이 있다. 기존의 기법들[7-9]은 상태 저장 기법과 핸드오프 기법에 따라서만 복구 기법을 구현하였다. 본 논문에서는 MH의 이동성에 따라 각각 다른 복구 기법을 사용하여 전체적인 누적 비용을 줄이는데 그 목적을 두고 ST(Switching Trickle)기법을 제안한다. ST기법은 핸드오프마다 쓰게 되는 write 이벤트의 비율을 고려한 기법이다. write 이벤트가 자주 발생하지 않는 경우 로깅 기법을 적용해서 안정성을 보장 받고, 핸드오프보다 더 많은 write 이벤트가 발생하는 경우 로깅 기법을 적용하지 않는 방법을 사용해서 핸드오프 하는데 드는 비용을 줄인다. ST기법은 복구비용도 효과적으로 감소시킬 수 있을 뿐만 아니라, 누적되는 네트워크 비용을 감소시킬 수 있다.

본 논문의 구성은 다음과 같다. 2장에서는 복구 기법을 구현하기 위해 그 동안 연구되어 온 상태 저장 기법과 핸드오프 기법을 살펴보고, 3장에서는 실험을 위해 사용한 용어들을 살펴본 후, 본 논문에서 제안한 ST기법의 동작 원리를 알아본다. 4장에서는 시뮬레이션의 비교 대상이 되는 기법들을 간략히 살펴본 뒤 파라미터를 다르게 하여 시뮬레이션 결과를 비교분석 해본다. 끝으로 5장에서 결론을 맺는다.

2. 관련 연구

2.1 상태 저장(state saving) 기법

MH들 간에 주고받는 체크포인트와 로그 정보는 MSS를 통해서 전달된다. 우선 해당 셀의 MSS에 체크포인트와 로그 정보가 전해진 후 해당하는 MH에 전달된다. 이 때 write 이벤트가 발생해야만 실제적인 메시지가 저장된다. 그 외 이미 받은 정보를 수신측에 전달하기 위한 ACK(acknowledgement) 메시지와 제어(control) 메시지는 write 이벤트로 취급하지 않는다[7].

N(No Logging)기법은 MH에서 write 이벤트를 생성할 경우에만 MSS에 메시지의 내용을 기록하도록 하는 방식이다. write 이벤트가 발생하지 않는 경우 MH로부터 MSS에 기록되는 내용이 없다.

L(Logging)기법은 체크포인트가 주기적으로 일어난다. write 이벤트는 주기적으로 일어나는 체크포인트들 사이에서 일어난다. write 이벤트를 처리하기 위해서는 ACK 메시지가 있어야 한다. ACK 메시지를 받지 않아도 MH는 write 이벤트를 처리하지만, 그 write 이벤트에 대한 결과를 즉시 처리하지 않는다. 결과 값은 MSS로부터 ACK 메시지가 확실하게 도착한 것을 확인한 후 처리한다. 오류가 발생한 후 MH가 다시 작업을 시작할 때 MH는 MSS에게 메시지를 보내 가장 최근에 저장된 체크포인트뿐만 아니라 write 이벤트 메시지를 받아서 작업 중이던 write 이벤트를 재생시킨다. 이런 과정을 통해 오류가 발생하기 전 상태로 돌아가 작업을 처리하는 것이다. 체크포인트와 로그 정보가 MSS에 중복 저장되는 것을 막기 위한 작업도 동시에 진행한다.

2.2 핸드오프(hand-off) 기법

핸드오프 처리를 설명하기 위해 간단한 상황을 예로 들어 설명하겠다. 그림 1은 일반적인 핸드오프 연산의 이해를 돕기 위한 그림이다. MH2는 MSS2에 체크포인트와 로그 정보를 저장한다. 실행하는 중간에 MH2는 MSS3로 이동하고 다시 MSS4로 이동한다. MH2가 MSS4에 도착했을 때 MH2에서 오류가 발생한다고 가정하자. MH2는 MSS2를 기억하고 있어야 한다. 왜냐하면, MSS2에 체크포인트와 로그에

관한 정보가 저장되어있기 때문이다. 다음 세 가지 방법은 핸드오프가 일어날 경우 어떻게 체크포인트와 로그에 관한 정보를 저장할 것인지를 제시한다. Pessimistic기법(P기법)은 MH가 핸드오프 하는 동안 체크포인트 정보는 새로운 다른 셀의 MSS로 전송된다. 로깅 기법을 사용하면 체크포인트뿐만 아니라 로그 정보까지 새로운 셀의 MSS로 전송된다. 체크포인트와 로그 정보를 받으면 새로운 셀의 MSS는 이전 셀의 MSS에 ACK 메시지를 보낸다. 체크포인트와 로그 정보의 ACK 메시지를 받은 이전 MSS는 체크포인트와 로그 정보를 제거한다. 따라서 더이상 은 체크포인트와 로그에 관한 정보를 갖고 있지 않다. 이 기법의 단점은 핸드오프 하는 동안 전송되는 데이터의 양이 꽤 많아진다는 것이다. 불안정한 네트워크 상황에서 정보를 전송하게 된다면 핸드오프 시간이 길어질 뿐만 아니라 전송 비용이 상당히 높아질 가능성이 있다.

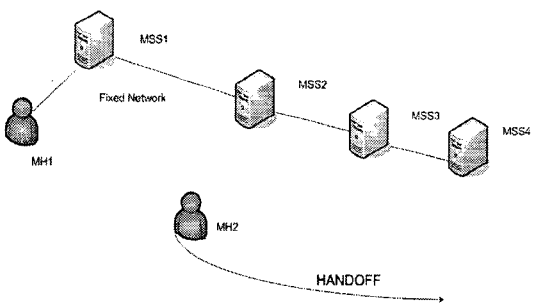

그림 1. 핸드오프 연산

Lazy기법(L기법)은 핸드오프 하는 동안 체크포인트나 로그에 어떤 변화도 일어나지 않는다. 대신 MH는 방문한 셀의 MSS를 연결 리스트로 만들어낸다. N기법을 사용한다면 write 이벤트 다음에 현재 셀의 MSS에 체크포인트 정보가 저장된다. L기법을 사용한다면 로그 정보뿐만 아니라 가장 최근에 발생한 체크포인트정보까지 저장된다. 핸드오프가 일어나는 동안 새로운 셀의 MSS는 이전 셀의 레코드를 저장한다. 따라서 셀에서 셀로 MH가 이동할 때 연결리스트 형태로 관리된다. L기법의 복구 방법은 P기법과 다르게 좀 복잡하다. 만약 현재 MH가 상주하고 있는 셀의 MSS에 상태 정보가 없다면 연결 리스트로 연결된 MSS에서 체크포인트와 로그 정보를 얻어와야 하기 때문에 복구 작업의 부담이 커지며, 요구되는 네트워크 비용도 커진다.

Trickle기법(T기법)은 체크포인트와 로그 정보들이 항상 MSS 근처에 있다는 것을 가정한다. 따라서 MH가 상주한 MSS나 이전 MSS에 체크포인트와 로그 정보가 저장된다. MSS끼리는 서로 평균 한 hop 정도의 거리만큼 떨어져 있다고 가정한다. 핸드오프 하는 동안 MSS는 체크포인트와 로그 정보를 전송받기 위해서 제어 메시지를 MH에 보낸다. 현재 MSS는 MH의 이전 위치를 알려주기 위해서 새로운 셀의 MSS에 제어 메시지를 보낸다. 이렇게 제어 메시지를 주고받는 방식을 통해 새로운 셀의 MSS는 MH의 이전 셀에 관한 정보를 유지할 수 있다. 체크포인트와 로그 정보가 중복 저장되는 것을 막기 위해 현재 MSS는 이전 MSS에 공지를 보낸다. 공지 내용은 이전 체크포인트와 로그를 지우라는 내용이다. 복구하는 동안 현재 MSS가 체크포인트와 로그 정보를 갖고 있지 않다면 그 이전의 MSS로부터 정보를 얻는다. 그런 다음에 MSS는 정보를 요구한 MH에 체크포인트와 로그 정보를 전송한다. MH는 체크포인트와 로그 정보를 받아서 재생시켜 오류가 발생하기 전 상태에 도달한다.

3. ST 기법

복구비용을 계산하기 위해 본 논문에서 사용되는 용어들은 다음과 같다.

- λ : MH의 오류율(failure rate of the MH)
- μ : MH의 핸드오프 비율(handoff rate of the MH)
- r : 통신-이동성 비율(communication-mobility rate) 핸드오프 당 발생하는 write 이벤트 수
- ρ : write 이벤트의 진행 정도 (1이면 write 이벤트가 완벽하게 진행된 것을 의미)
- Tc : 체크포인트 간격 (N 기법일 때 $Tc = 1/\beta$)
- k : 체크포인트 당 발생하는 write 이벤트 수
  L 기법에서 $k = \beta Tc$, N 기법에서 $k = 1$
- α : 무선 네트워크의 인자(wireless network factor)

유선 상에서 한 hop 거리에 메시지를 전달하는데 드는 비용에 대한 무선 상에서 한 hop 거리에 메시지를 전달하는데 드는 비용의 비율

Nc(t) : t 단위 시간에 발생하는 체크포인트 수

Nl(t) : t 단위 시간에 로그 된 메시지 수
Cc : 유선 상에서 한 hop 거리에 있는 체크포인트 정보 전달의 평균 비용
Cl : 유선 상에서 한 hop 거리에 있는 메시지를 전송하는데 드는 평균 비용
Cm : 유선 상에서 한 hop 거리에 있는 제어 메시지를 전송하는데 드는 평균 비용
Ch : 핸드오프 연산을 위한 평균 비용
Cr : 복구 연산을 위한 평균 비용
Ct : 전체 평균 비용

Markov chain 표현의 3-단계를 사용해서 두 핸드오프 사이에 일어날 수 있는 상황을 설명하면 다음과 같다[11]. 상태 0은 핸드오프를 시작하는 초기 상태이다. 두 핸드오프가 일어나는 간격 동안 MH는 메시지를 받거나 write 이벤트를 발생할 수 있다. 핸드오프가 일어나는 동안 오류 없이 완벽하게 진행되면 상태 0에서 상태 1로 전이된다. 만약에 오류가 발생하면 상태 0에서 상태 2로 전이된다. 상태 2에서 다시 상태 1로 가면 체인을 모두 돌게 된다. Nc(T)와 Nl(T)를 T 시간 간격에 수정된 체크포인트와 메시지라고 했을 때, 오류가 발생하지 않는다는 것을 보장할 경우(상태 0에서 상태 1로 이동) 핸드오프 비용(C_{01})을 정의하면 다음과 같다.

$$C_{01} = (\alpha Cc) * Nc(T) + (\alpha Cl) * Nl(T) + Ch$$

복구비용(Cr, 상태 0에서 상태 2로 이동)은 오류가 발생하기 전까지의 MH의 상태를 저장하기 위해 요구되는 비용이다. 전체 비용(Ct)은 오류 발생 유무와 관계없이 핸드오프 간격 동안 발생되는 기대 비용이다. 전체 비용은 다음 식과 같이 표현할 수 있다. 여기서 P_{02}는 오류가 발생할 확률을 나타내며, 이 비용은 상태 저장 기법과 핸드오프 기법에 따라 다르게 요구된다.

$$Ct = C_{01} + P_{02}Cr$$

r(통신 이동성 비율, 핸드오프 당 발생하는 write 이벤트 수)에 따라 상태 저장 기법을 달리하여 전체 비용을 줄이고자 제안한 방법이 ST기법이다. r이 1(핸드오프가 1회 발생할 때 write 이벤트가 1회 발생할 경우)보다 작을 경우 핸드오프 하는 동안 제어 메시지는 체크포인트와 로그 정보를 전송하기 위해 이전 MSS에 한 번 보내지며, MH의 이전 위치를 알려주기 위해 또 한 번 제어 메시지를 보낸다. 따라서 핸드오프 비용은 두 개의 제어 메시지를 보내는 비용과 한 hop 떨어진 MSS에 실질적인 체크포인트와 로그 정보를 전송하는 비용이 요구된다. 이 비용은 기본적인 핸드오프 연산만을 계산한 비용이다. 핸드오프 비용(Ch)을 다음 식과 같이 나타낼 수 있다. 여기서 v는 메시지 로그의 평균 크기를 의미한다.

$$Ch = vCl + Cc + 2Cm$$

핸드오프 비용뿐만 아니라 체크포인트와 로그 정보가 MSS에 중복 저장되는 것을 막기 위해 기존에 있던 정보를 제거하는 비용 또한 C_{01} 비용에 포함된다. 현재 MSS에서 중복된 체크포인트와 로그 정보가 있는 MSS까지의 평균 거리를 Nh hop이라고 가정할 때, 제어 메시지를 Nh만큼 떨어진 MSS로 보내는데 요구되는 비용은 NhCm이다. 따라서 오류가 발생하지 않을 경우 핸드오프 비용(C_{01})은 다음과 같이 표현할 수 있다.

$$C_{01} = (r\alpha Cc)/k + p\alpha Cl + p\alpha Cm + vCl + Cc + 2Cm + NhCm$$

오류가 발생했을 경우 이전 정보를 복구하기 위해선 MH와 이전 MSS에 제어 메시지를 보내 정보를 요구한다. 이 때 Cm(α + Nh)의 비용이 요구되며, 제어 메시지를 받은 MH와 MSS가 체크포인트와 로그 정보를 보내는데 요구되는 비용이 (α + Nh)(vCl + Cc)이다. 이 비용은 다음 식으로 표현할 수 있다. 그림 2는 r이 1보다 작을 경우 복구 비용(Cr)을 나타낸다.

$$Cr = (Nh + \alpha)(vCl + Cc + Cm)$$

r이 1(핸드오프가 1회 발생할 때 write 이벤트가 1회 발생할 경우)보다 클 경우 복구 기법을 스위칭하게

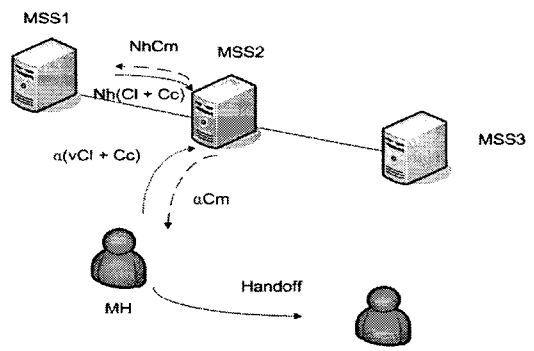

그림 2. 복구를 위해 요구되는 비용 (Cr, r<1)

된다. 현재 MSS에서 새로운 MSS로 체크포인트와 MH의 위치 정보를 전송할 때 두 개의 제어 메시지를 보내는 비용 Cm을 전송하며, 그 때 실질적인 체크포인트 전송 비용 Cc도 전송된다. 이 비용은 다음 식으로 표현할 수 있다.

$$Ch' = Cc + 2Cm$$

r이 1보다 작으면서, 오류가 발생하지 않는다고 보장할 경우 요구되는 비용은 r이 1보다 클 때와 유사하게 이전에 저장되었던 체크포인트 정보를 제거하는 비용 NhCm이 요구된다. 이전 MSS는 기존에 저장되었던 체크포인트 정보를 보내는 비용 rαCc가 필요하며, 핸드오프 연산을 위해 2Cm + Cc의 비용이 추가로 요구된다. 따라서 오류가 발생하지 않을 경우 핸드오프 비용(C_{01}')은 다음 식과 같다. 그림 3은 r이 1보다 클 경우 복구 비용(Cr′)을 나타낸다.

$$C_{01}' = (r\alpha + 1)Cc + 2Cm + NhCm$$

복구를 위해서는 이전 MSS에 저장된 체크포인트 정보를 얻는 비용 NhCc이 필요하며, 현재 MH가 갖고 있는 체크포인트 정보 αCc가 필요하다. 뿐만 아니라, MH로 정보를 요청하기 위한 제어 메시지 비용 αCm과 중복 저장된 정보를 제거하기 위한 비용 NhCm도 추가적으로 필요하다. 따라서 이 비용은 다음 식으로 표현할 수 있다.

$$Cr' = (Nh + \alpha)(Cc + Cm)$$

다음 두 개의 알고리즘은 제안된 ST기법에 대해 각각 오류가 발생하지 않을 경우의 핸드오프 비용(C_{01}, C_{01}')과 복구비용을 적용한 전체 비용(Ct, Ct′)을 나타낸다.

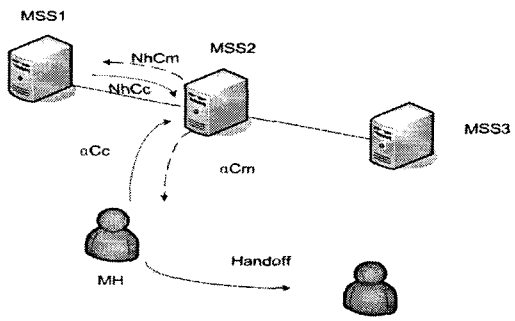

그림 3. 복구를 위해 요구되는 비용 (Cr′, r >= 1)

```
% ST hand-off cost without failure
% α wireless network rate, λ failure rates
α = 10 ; λ  = 0.01 ;
μ = 0.1; Tc = 0.1; ρ =0.5 ; k = 0.1 ; β = 1; Nh = 0.01;
v = (k-1) / 2 ;
Cm = 0.01; Cl = 10; Cc = 100;
r=0.01:0.1:2.0;
if r < 1.0
 C01 = (r * α * Cc)/k + (p * r * α * Cl) + (p * r * α * Cm)
    +(v * Cl) + Cc + (Nh * Cm) + 2* Cm ;
else  C01 ′ = (r * α * Cc) + Cc + (2 * Cm) + (Nh * Cm);
end

% ST  Total cost with failure
% Ct = C01 + P02 * Cr
% α wireless network rate,  λ failure rates
α = 10 ; λ  = 0.01 ;
μ = 0.1; Tc = 0.1; ρ =0.5 ; β = 1; Nh = 0.01;
v = (k  1) / 2 ; k = β * Tc ; P02 = λ/(λ +μ);
Cm = 0.01; Cl = 10; Cc = 100;
r = 0.01:0.1:2.0;
if r < 1.0
C01= (r * α * Cc)/k + (p * r * α * Cl) + (p * r * α * Cm)
   + (v * Cl) + Cc + (Nh * Cm) + 2* Cm ;
  Cr = (Nh + α)* ((v * Cl) +Cc + Cm);
  Ct = C01 + (P02 * Cr) ;
else C01′= (r * α * Cc) + Cc + (2 * Cm) + (Nh * Cm);
  Cr′  = (Nh + α) * (Cc + Cm);
  Ct′  = C01′ + (P02 * Cr′) ;
End
```

4. 성능 평가

4.1 시뮬레이션 환경

네트워크 비용을 측정하기 위해 사용한 상대 비용은 표 1과 같다. 여기서 상대 비용은 write 이벤트 사이의 간격이 1일 경우의 비용을 의미한다.

LL(Logging-Lazy)기법은 MSS로 MH가 로깅할 때마다 정보를 저장하며, 핸드오프가 일어날 때마다 새로운 MSS에 이전 MSS의 정보를 레코드로 담는 연결리스트 형식을 취한다. LL기법을 사용하기 위해 요구되는 네트워크 비용은 다음과 같다. MSS

표 1. 시뮬레이션 파라미터

파라미터	상대 비용
핸드오프 비율(μ)	0.1
체크포인트 간격(Tc)	0.1
write 이벤트의 진행 정도(ρ)	0.5
write 이벤트 사이의 간격(β)	1
체크포인트와 로그 정보가 중복 저장된 MSS와 현재 MSS와의 평균 hop 수(Nh)	0.01
오류가 발생할 확률(P_{02})	λ/(λ+μ)
제어 메시지를 보내는 비용(Cm)	0.01
로그 정보를 보내는 비용(Cl)	10
체크포인트 정보를 보내는 비용(Cc)	100

가 MH의 위치에 관한 정보를 새로운 MSS에게 전송하기 위해 Cm 만큼의 비용이 필요하며, 평균 Nh hop 만큼 떨어져 있는 MSS의 정보를 제거하기 위해 NhCm가 필요하다. ρ는 write 이벤트가 진행된 일부분을 의미하며, ρr은 다음 핸드오프가 일어나기 전까지 처리되는 write 이벤트의 정도를 나타낸다. 무선 네트워크를 지날 때마다 MSS는 ACK 메시지 비용으로 αCm을 지불한다. 체크포인트와 로그 정보 또한 무선 네트워크 상에서 전송되기 때문에 (rαCc)/k + ραCl 의 비용을 필요로 한다. 따라서 LL기법의 오류가 발생하지 않을 경우 핸드오프 비용(C_{01}_LL)은 다음 식과 같다.

$$C_{01}\_LL = (r\alpha Cc)/k + \rho r\alpha Cl + \rho r\alpha Cm + NhCm + Cm$$

LL기법의 복구비용(Cr_LL)은 다음과 같다. 오류가 발생하기 전의 체크포인트와 로그정보를 찾기 위해선 Nh hop만큼 떨어진 곳의 MSS를 찾아 체크포인트와 로그 정보를 전송받아야 한다. 이렇게 전송받는데 요구되는 비용은 Nh(vCl + Cc)이며, 전송하기 위해 메시지를 요청하는데 요구되는 비용은 NhCm 이다. 현재 MSS가 MH로부터 체크포인트와 로그 정보를 요청하고 전송받는데 요구되는 비용은 α(vCl + Cc + Cm)이다. 따라서 Cr_LL을 정리하면 다음 식과 같다.

$$Cr\_LL = (Nh + \alpha)(vCl + Cc + Cm)$$

NP기법(No Logging-Pessimistic)은 write 이벤트가 발생할 때마다 체크포인트를 생성한다. 생성된 체크포인트는 무선 네트워크를 거쳐 MSS에 전달된다. NP기법을 사용하는데 요구되는 비용은 다음과 같다. 체크포인트 비용을 무선 네트워크 상에서 전달하기 위해 αCc의 비용이 필요하다. 핸드오프가 일어날 때 새로운 MSS로 전달하는 체크포인트 비용은 Cc이며, 위치에 관한 정보를 전달하기 위한 비용 Cm 또한 필요하다. 따라서 오류가 발생하지 않을 경우 핸드오프 비용(C_{01}_NP)는 다음 식과 같다.

$$C_{01}\_NP = (r\alpha+1)Cc + Cm$$

NP기법에서 체크포인트는 write 이벤트가 발생할 때마다 생성된다. 따라서 MH는 복구비용으로 체크포인트 정보를 전달한다. 체크포인트 정보를 요청하는 비용 Cm과 Cc 비용을 합한 값이 NP기법의 복구비용이다.

$$Cr\_NP = \alpha(Cc + Cm)$$

4.2 시뮬레이션 결과

매트랩 version 6.5에서 각 기법에 따라 C_{01}(오류가 발생하지 않을 경우 핸드오프 비용)과 Ct(전체 비용)의 누적 비용을 측정하였다. 그림 4는 r에 따라 변화하는 C_{01}을 각 기법에 따라 나타낸 것이다. NP기법은 write 이벤트마다 체크포인트 정보만을 보내기 때문에 상대적으로 낮은 값을 갖는다. 또한 무선 네트워크 상에서 전달되는 정보는 제어 메시지와 체크포인트 정보뿐이기 때문에 α(무선 네트워크의 인자, wireless network factor)에 큰 영향을 받지 않는다. 반면 LL기법은 핸드오프가 일어날 때에만 로그 정보를 전송하기 때문에 세 기법 중 가장 큰 값을 갖는다. ST기법은 r이 1보다 작을 경우 LL기법과 유사하게 체크포인트 정보뿐만 아니라 로그 정보를 전송한다. 따라서 점차적으로 증가하다가 r이 1인 시점에서 갑자기 낮아지는 모습을 확인할 수 있다.

NP기법은 메시지에 대한 중요도가 낮은 메시지를 전송할 경우 높은 효율을 나타낼 것이며, LL기법은 중요도가 높은 메시지를 안정적으로 보내고자 할 경우 최적의 성능을 보일 수 있는 기법이다. 그에 반해 ST기법은 메시지의 중요도를 정확히 파악하지 못하거나 안정성을 확신하지 못하는 상황에서도 사용할 수 있는 기법이다.

그림 5, 6, 7은 r이 커짐에 따라 달라지는 LL기법과 ST기법(0~2×10^5), NP기법(0~8×10^4)의 누적 비용을 측정한 것이다. r이 1 이전의 값을 가질 때는 비슷

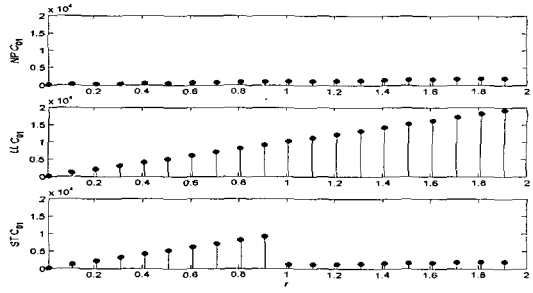

그림 4. 오류가 없을 경우 핸드오프 비용(C_{01})

한 비율로 증가하지만, 1 이상의 값을 가질 때는 ST 기법이 두드러지게 안정적인 값을 갖는 것을 확인할 수 있다. 두 기법 모두 로깅 시 정보를 기록하는 기법을 사용하여 체크포인트와 로그 정보를 안전하게 저장할 수 있다. 하지만 ST기법은 선별적으로 LL기법을 사용하여 네트워크 비용을 줄일 수 있다는 또 다른 장점을 갖고 있다. 한편, ST기법이 상대적으로 NP기법보다 상대적으로 누적비용이 큰 값을 갖는다. 하지만 NP기법은 로깅 시 정보를 저장하지 않기 때문에 안정성면에서 ST기법보다 열악하다고 볼 수 있다.

ST기법은 정보를 저장할 때 체크포인트 정보뿐만 아니라 로그 정보를 저장하기 때문에 오류가 발생한 후 복구했을 경우, 송신측에서 보낸 정보와 수신측에서 받은 정보의 일치하는 정도가 체크포인트 정보만을 저장하는 LL기법보다 높게 나타난다. 수신측 정보의 정확성은 복구 기법의 안정성을 측정하는 중요한 요소이다. 따라서 로그 정보를 저장하는 기법이 안전하다는 것을 알 수 있다. 위의 누적 비용 실험을 통해 ST기법이 비용 면에서 LL기법보다 경제적이며, 안정성면에서 NP기법보다 우수하다는 것을 알 수 있다.

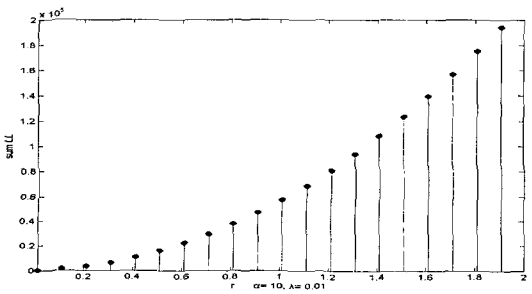

그림 5. LL 기법의 누적 비용(0~2×10^5)

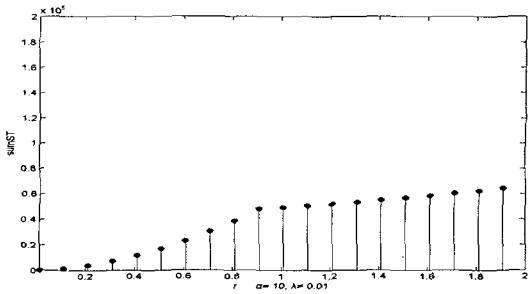

그림 6. ST 기법의 누적 비용(0~2×10^5)

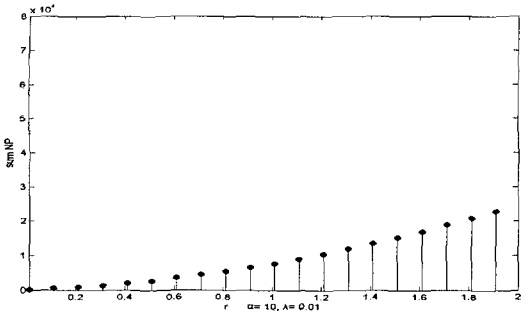

그림 7. NP 기법의 누적 비용(0~8×10^4)

5. 결 론

본 논문에서는 이동 컴퓨팅 환경에서 이동 호스트들의 네트워크 비용과 안정성에 관한 문제를 살펴보았다. 특히 네트워크 비용을 모바일 호스트들의 움직임에 따라 스위칭하여 줄여보고자 하였다. 복구 기법은 상태 저장 기법과 핸드오프 기법을 어떻게 적절히 접목시키느냐에 따라 전체적인 비용을 절약할 수 있다.

본 논문에서 제안한 기법은 핸드오프 당 발생하는 write 이벤트 수(r)에 따라 상태 저장 기법을 달리하였다. 따라서 복구가 필요한 상황을 r의 값으로 나눠 로깅이 필요한 상황과 그렇지 못한 상황을 분류하여 네트워크 비용을 산출하였다. 또한 핸드오프 기법 중에서 제어 메시지를 사용하여 이동 호스트와 이동 지원국간에 의사소통이 가능한 기법을 사용하였다. 시뮬레이션을 통해 각 기법의 성능을 비교하였으며 오류율에 변화를 주어 무선 네트워크 상에서 정보 전달 비용이 시스템을 구현하는데 큰 변수로 작용한다는 것을 확인했다. 제안한 기법은 로깅 방법을 사용하기 때문에 대체적으로 안정적이며 제어 메시지를 통해 이동 호스트와 이동 지원국의 의사소통이

가능하기 때문에 다른 기법에 비해 적은 네트워크 비용으로 복구가 가능함을 보였다. 향후, 네트워크 비용뿐만 아니라 메시지들의 정확성이나 중요도를 고려한 복구 기법을 연구하고자 한다.

참 고 문 헌

[1] D. Barbara, "Mobile Computing and Database A Survey," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 11, No. 1, pp. 108-117, 1999.

[2] G. H. Forman and J. Zahorjan, "The Challenge of Mobile Computing," *IEEE Computers*, Vol. 27, No. 6, pp. 34-47, 1994.

[3] D. K. Pradhan, P. Krishan, and N. H. Vaidya, "Recoverable Mobile Environment Design and Trade off Analysis," *Proc. of the IEEE Fault Tolerant Computing*, pp. 16-25, 1996.

[4] E. N. Elnozahy and W. Wzaenepoel, "Manetho: Transparent Rollback-Recovery with Low Overhead, Limited Rollback and Fast Output Commit," *IEEE Transactions on Computers*, Vol. 41, No. 5, pp. 526-531, 1992.

[5] A. Borg, J Baumbach, and S. Glazer, "A Message System Supporting Fault Tolerance," *Proc. of the Symposium on Operating Systems Principles*, pp. 90-99, 1983.

[6] L. Alvisi and K. Marzullo, "Message Logging: Pessimistic, Optimistic, Causal and Optimal," *IEEE Transactions on Software Engineering*, Vol. 24, No. 2, pp. 149-159, 1994.

[7] R. Prakash and M. Singhai, "Low Cost Checkpointing and Failure Recovery in Mobile Computing Systems," *IEEE Transactions on Parallel and Distributed Computing Systems*, Vol. 7, No. 10, pp. 1035-1048, 1996.

[8] T. Park, N. Woo, and H. Y. Yeom, "An Efficient Recovery Scheme for Fault-Tolerant Mobile Computing Systems," *Future Generation Computer Systems*, Vol. 19, No. 1, pp. 37-53, 2002.

[9] T. Park, N. Woo, and H. Y. Yeom, "An Efficient Optimistic Message Logging Scheme for Recoverable Mobile Computing Systems," *IEEE Transactions on Mobile Computing*, Vol. 1, No. 4, pp. 265-277, 2002.

[10] K. S. Trivedi, *Probability and Statistics with Reliability Queueing and Computer Science Applications*, Prentice Hall, 1982.

최 가 현

2002년 가톨릭대학교 컴퓨터공학과(공학사)
2004년 가톨릭대학교 컴퓨터공학과(공학석사)
2004년~현재 ㈜엑사텔레콤 연구원

관심분야 : Mobile 데이터베이스, 웹 데이터베이스, XML

황 병 연

1986년 서울대학교 컴퓨터공학과(공학사)
1989년 한국과학기술원 전산학과(공학석사)
1994년 한국과학기술원 전산학과(공학박사)
1994년~현재 가톨릭대학교 컴퓨터정보공학부 교수
1999년~2000년 University of Minnesota Visiting Scholar

관심분야 : GIS, XML, 데이터 마이닝, 정보검색