

# 이중 셀룰러 네트워크에서의 Redundant Lazy 기법의 성능 개선

권원석,<sup>o</sup> 김성수

아주대학교 정보통신전문대학원

{wonseok, sskim}@madang.ajou.ac.kr

## Performance Improvement of Redundant Lazy Scheme in Two-Tier Cellular Network

Wonseok Kwon, Sungsoo Kim

Professional Graduate School of Information and Communication Technology, Ajou University

### 요 약

현재 이동 컴퓨팅 연구 중에서 이동 호스트의 결합을 효율적으로 대처하는 결합 허용 기법에 관한 연구가 미미한 실정이다. 이동 호스트의 결합 복구 기법인 Lazy 기법은 비용 면에서는 효율적이지만 이동 호스트의 체크포인트를 소유하고 있는 기지국의 결합 발생시 이동 호스트가 결합으로부터 복구될 수 없다는 단점을 가지고 있다. Redundant Lazy 기법은 연결 리스트를 중복화 함으로서 Lazy 기법의 문제점을 해결한 기법이다. 그러나 Redundant Lazy 기법도 체크포인트를 가지고 있는 기지국의 결합이 발생하면 이동 호스트가 복구될 수 없는 경우가 발생한다. 본 논문에서는 이중 셀룰러 네트워크에 Redundant Lazy 기법을 적용하여 이러한 문제를 해결할 수 있는 성능 개선 방안을 논의한다.

### 1. 서론

이동 호스트는 이동 호스트 자체의 결합, 네트워크 연결 단절(disconnection), 무선 링크의 결합 등의 기존 유선 네트워크에서는 찾아볼 수 없는 새로운 결합 원인들을 포함하고 있다[1].

메시지 로깅(message logging)과 체크포인팅(checkpointing)은 호스트가 결합이 발생했을 때 효율적으로 복구할 수 있는 결합 허용 기법 중 하나이다. 기존의 유선 네트워크 상에서 고정된 호스트들에 대한 메시지 로깅과 체크포인팅에 관한 연구는 활발히 진행되고 있으나 이동 컴퓨팅 환경처럼 호스트에 이동성이 추가 될 경우에는 기존의 메시지 로깅이나 체크포인팅 기법을 그대로 적용하기에는 어려움이 따른다. 그 첫 번째 이유로는 이동 호스트는 로그나 체크포인트를 저장하는 안정적인 저장 장치를 소유하기 어렵다는 것이다. 이러한 문제점을 해결하기 위해 로그나 체크포인트를 기지국에 저장하는 다수의 방법들이 제안되었다. 두 번째로 이동 호스트가 하나의 셀에 고정되어 위치하고 있다면 기존의 기법들을 그대로 수정 없이 사용할 수 있으나 이동 호스트는 그 자체의 이동성 때문에 여러 개의 셀들을 옮겨 다니는 핸드오프(handoff)를 수행하므로 기지국에 저장되어 있는 로그나 체크포인트를 관리하기가 어렵게 된다.

이동 호스트의 결합 복구 기법인 Lazy 기법[1]은 연결 리스트에 포함된 기지국의 결합이 발생했을 경우 호스트는 기지국으로부터 디스크에 기록된 프로세스의 상태 정보를 가져올 수 없게 되어 복구를 수행할 수 없는 단점을 가지고 있다. Redundant Lazy 기법[2]은 연결 리스트를 중복화 함으로서 Lazy 기법의 문제점을 해결한 기법이다. 그러나 Redundant Lazy 기법도 체크포인트를 가지고 있는 기지국 자체의 결합이 발생하면 이동 호스트가 복구될 수 없는 경우가 발생한다.

본 논문은 이중 셀룰러 네트워크에 Redundant Lazy 기법을 적용하여 이러한 문제를 해결할 수 있는 성능 개선 방안을 논의한다.

### 2. 관련 연구

Pradhan[1]이 제안한 두 가지의 체크포인팅 프로토콜 중 첫 번째 프로토콜은 프로세스가 메시지를 받을 때마다 새로운 체크포인트를 생성하는 방법이고 두 번째는 주기적으로 프로세스를 체크포인팅하는 것과 동시에 메시지를 받을 때마다 그 메시지를 로깅하는 방법이다. Pradhan은 이 두 가지의 체크포인팅 프로토콜을 세 가지의 서로 다른 핸드오프 기법과 조합하여 총 여섯 가지의 기법을 제안하고 각 제안 기법들의 비용 분석을 수행하였다.

Neves[3]는 이동 컴퓨팅을 위한 적응적(adaptive) 체크포인팅 기법을 제안하였다. 이 기법에서는 체크포인팅을 하드(hard) 체크포인팅과 소프트(soft) 체크포인팅으로 나누어 하드 체크포인트는 기지국으로 전송하고 소프트 체크포인트는 이동 호스트의 로컬 디스크에 저장한다.

Yao[4]는 이동 IP(mobile IP) 환경에서의 메시지 로깅에 관한 연구를 수행하였다. 이 논문에서는 이동 호스트가 셀 사이를 이동하면서 체크포인팅을 수행하면 해당 셀의 MSS(Mobile Support Station)에 체크포인트를 저장하고 HA(Home Agent)에게 체크포인팅 수행 사실을 알리는 기법을 제안하였다. 또한 이 논문에서는 여러 셀에 저장되어 있는 체크포인트에 대한 쓰레기 값(garbage)을 처리하는 연구도 수행하였다.

### 3. 결합 복구 기법

#### 3.1 이중 셀룰러 네트워크

이동 호스트의 체크포인트를 가지고 있는 기지국의 결합을 허용하기 위해서 본 논문에서는 셀룰러 네트워크를 이중 셀룰러 네트워크로 가정한다. 이중 셀룰러 네트워크는 마이크로 셀(micro cell)과 매크로 셀(macro cell)로 구성되며 다수의 마이크로 셀이 하나의 매크로 셀에 포함된다. 마이크로 셀은 이동성이 낮은 이동 호스트를 서비스하는 셀이며 매크로 셀은 상대적으로 이동성이 큰 이동 호스트를 서비스한다. 마이크로 셀로의 네트워크 연결 확립을 위한 연결 요청이나 핸드오프 요구가 overflow 됐을 경우 매크로 셀은 이 연결 요청을 대체할 수 있다. 본 논문에서의 이동 호스트는 마이크로 셀 단위로 이동하는 호스트

This work is supported in part by the Ministry of Information & Communication of Korea("Support Project of University Foundation Research<00>" supervised by IITA)

로 가정한다.

**3.2 프로세스 상태 저장 기법**

본 논문에서는 이동 호스트의 프로세스 상태 저장 기법을 두 가지로 나누어 이중 셀룰러 네트워크에서의 Redundant Lazy 기법의 비용 분석을 수행하였다. 첫 번째 프로세스 저장 기법은 No Logging 기법으로 이동 호스트는 일정한 주기에 따라서 체크포인팅 작업을 수행하지 않고 프로세스의 상태가 변하는 쓰기 동작이 일어날 때마다 체크포인팅을 수행하는 기법이다. 프로세스의 상태를 변화시키는 쓰기 동작의 유형은 사용자의 입력과 다른 이동 호스트로부터 들어오는 메시지로 구분할 수 있다. Logging 기법은 이동 호스트가 주기적으로 프로세스를 체크포인팅하고 쓰기 동작이 발생할 때마다 해당 동작을 로깅하는 기법이다.

**3.3 Redundant Lazy 기법의 성능 개선**

Redundant Lazy 결합 복구 기법은 Lazy 기법의 단점을 보완한 결합 복구 기법이다. Lazy 기법은 이동 호스트가 쓰기 동작이나 체크포인팅 주기 사이에 핸드오프를 수행하면 방문하였던 기지국의 연결 리스트를 생성하여 자신의 로그나 체크포인트가 저장된 기지국을 확인하는 기법이다. 이 기법은 연결 리스트로 관리되던 기지국 중 하나의 기지국에 결합이 발생하면 연결 리스트의 가장 끝에 위치한 기지국에 저장되어 있던 자신의 체크포인트를 전송 받을 수 없는 단점을 가지고 있다.

Redundant Lazy 기법은 연결 리스트를 중복 생성하여 하나의 연결 리스트에 속한 기지국의 결합이 발생하여도 나머지 또 다른 연결 리스트를 통해 이동 호스트의 결합이 발생했을 경우에도 로그와 체크포인트를 전송 받아 복구할 수 있다. 그러나 Redundant Lazy 기법도 연결 리스트의 마지막 기지국인 체크포인트가 저장되어 있는 기지국의 결합 발생 시 이동 호스트가 복구될 수 없는 단점이 있다.

이중 셀룰러 네트워크에서 마이크로 셀의 이동 호스트는 매크로 셀의 기지국에 접속을 할 수 있기 때문에 본 논문에서 이러한 문제를 이중 셀룰러 네트워크를 사용하여 마이크로 셀의 기지국에 이동 호스트의 체크포인트를 저장하고 매크로 셀의 기지국에도 똑같은 체크포인트를 저장하여 체크포인트를 중복시킴으로서 해결한다.

**4. 성능 평가**

이 장에서는 3장에서 기술된 프로세스 상태 저장 기법과 이중 셀룰러 네트워크에서의 Redundant Lazy 기법을 조합하여 비용 분석을 수행한다. 본 논문에서의 비용 분석이란 이동 호스트가 마이크로 셀에서 핸드오프와 핸드오프 사이에 메시지 로깅과 체크포인팅을 하는데 소요되는 비용과 이동 호스트의 결합 발생 시 소요되는 복구 비용의 합을 계산하는 과정을 의미한다. 그리고 비용이란 이동 호스트의 네트워크 사용량을 의미한다. 또한 이동 호스트가 매크로 셀의 경계를 넘어가는 핸드오프에 대한 고려는 하지 않는다고 가정한다.

Redundant Lazy 기법의 비용 분석을 하기 위해 아래와 같이 용어와 기호를 정의한다.

- $\alpha$  : 마이크로 셀에서의 무선 네트워크 사용 비용
- $\beta$  : 매크로 셀에서의 무선 네트워크 사용 비용
- $\lambda_m$  : 이동 호스트의 결합 발생률
- $\lambda_b$  : 마이크로 셀 기지국의 결합 발생률
- $\mu$  : 이동 호스트의 핸드오프율
- $r$  : 핸드오프 당 쓰기 동작의 기대 값,  $r = 1/\mu$
- $k$  : 체크포인트 당 쓰기 동작의 수
- $N_c(t)$  : 시간  $t$  에서 체크포인트의 수
- $N_l(t)$  : 시간  $t$  에서 메시지 로그의 수
- $C_c^{MC}$  : 마이크로 셀에서 기지국에 체크포인트를 전송하는 비용
- $C_c^{mC}$  : 매크로 셀에서 기지국에 체크포인트를 전송하는 비용
- $C_l^{MC}$  : 마이크로 셀에서 기지국에 로그를 전송하는 비용

- $C_l^{mC}$  : 매크로 셀에서 기지국에 로그를 전송하는 비용
- $C_m^{MC}$  : 마이크로 셀에서 기지국에 컨트롤 메시지를 전송하는 비용
- $C_m^{mC}$  : 매크로 셀에서 기지국에 컨트롤 메시지를 전송하는 비용
- $C_h$  : 핸드오프 동작의 평균 비용
- $C_R$  : 이동 호스트의 복구 비용
- $C_i$  : 이동 호스트가 핸드오프를 수행하는데 드는 총 비용

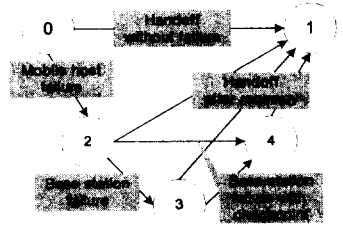


그림 1 상태 전이도

**4.1 모델링**

두 개의 연속되는 핸드오프 사이의 시간 간격을 핸드오프 간격이라고 정의하고 핸드오프 간격은 그림 1과 같은 상태 전이도로 표현할 수 있다. 본 논문에서는 이동 호스트의 마이크로 셀의 핸드오프만 고려하고 매크로 셀의 경계를 넘나들면서 발생하는 매크로 셀의 기지국의 체크포인트 관리에 고려하지 않는다.

그림 1에서 이동 호스트가 결합이 발생하지 않고 핸드오프를 수행하는 것이 상태 0에서 1로 가는 것이고 만약 결합이 발생하였을 경우 상태 2로 전이되어 체크포인트가 저장되어 있는 위치를 첫 번째 연결 리스트를 통해 확인하여 결합을 체크포인트로부터 복구한 후 상태 1로 전이된다. 만약 첫 번째 연결 리스트에 속해 있는 기지국의 결합이 발생했을 경우 상태 2에서 상태 3으로 전이되어 두 번째 연결 리스트를 통해 체크포인트가 저장되어 있는 기지국을 확인하여 복구하게 된다. 또한 체크포인트가 저장되어 있는 기지국의 결합이 발생하여 연결 리스트를 통해 복구 될 수 없을 경우 상태 2나 상태 3에서 상태 4로 전이되어 매크로 셀의 기지국에 저장되어 있는 체크포인트를 가지고 복구를 수행한 후 상태 1로 전이된다. 각 상태로의 전이 확률  $P_{02}, P_{23}, P_{24}, P_{34}$ 는 다음과 같이 계산된다.

$$P_{02} = \frac{\lambda_m}{\lambda_m + \mu}, \quad P_{23} = \frac{N_h \lambda_b}{N_h \lambda_b + \mu}, \quad P_{24} = \frac{\lambda_b}{\lambda_b + \mu}, \quad P_{34} = \frac{\lambda_b}{\lambda_b + \mu}$$

여기서  $N_h$ 는 마지막 체크포인트가 일어나고 이동 호스트의 결합이 발생하기 전까지 발생한 평균 핸드오프의 수행 회수로 이동 호스트가 결합이 발생하기 전까지 이동한 셀의 개수를 의미하고 다음과 같이 계산된다[1].

$$N_h = \mu T_{exp}$$

여기서  $T_{exp}$ 는 체크포인트 간격에서 체크포인트가 일어난 시점부터 다음 체크포인트가 일어나기 전에 이동 호스트의 결합이 발생할 때까지 예상되는 존속시간으로 다음과 같이 계산된다[1].

$$T_{exp} = \int_0^r \frac{t \lambda e^{-\lambda t}}{1 - e^{-\lambda t}} dt = \frac{1}{\lambda} \frac{T_c e^{-\lambda r}}{1 - e^{-\lambda r}}$$

상태 전이 (0,1)의 비용  $C_{01}^{mC}$ 은 마이크로 셀의 핸드오프 간격 사이에서 핸드오프가 수행되기 전에 발생한 동작의 예상되는 총 비용으로 다음과 같이 계산한다.

$$C_{01}^{mC} = (\alpha C_c^{mC}) \times N_c(T) + (\alpha C_l^{mC}) \times N_l(T) + C_h$$

비용  $C_{01}^{MC}$ 은 이동 호스트가 로깅이나 체크포인팅을 수행할 때마다 매크로 셀의 기지국에 백업하는 비용으로 다음과 같이 계산한다.

$$C_{01}^{MC} = (\beta C_c^{MC}) \times N_c(T) + (\beta C_l^{MC}) \times N_l(T)$$

결합에 관계없이 이동 호스트가 마이크로 셀에서의 핸드오프 간격에서 발생 할 수 있는 총 비용은 다음과 같다.

$$C_r = C_{01}^{mc} + C_{01}^{MC} + P_{02} C_R \quad (1)$$

이동 호스트의 결합 발생 시 소요되는 복구 비용  $C_R$ 은 다음과 같이 계산한다.

$$C_R = (1 - P_{23}) \{ (1 - P_{24}) C_r^F + P_{24} C_r^{MC} \} + P_{23} \{ (1 - P_{34}) C_r^S + P_{34} C_r^{MC} \}$$

여기서 이동 호스트가 첫 번째 연결 리스트를 통해 체크포인트가 저장되어 있는 기지국을 확인하여 전송 받은 후 복구되는 비용은  $C_r^F$ 이고 첫 번째 연결 리스트에 속해 있는 기지국의 결합 발생으로 인하여 두 번째 연결 리스트를 통해 복구하는 비용이  $C_r^S$ 이다. 만약 체크포인트를 가지고 있는 기지국의 결합이 발생하였을 경우 이동 호스트가 위치하고 있는 매크로 셀의 기지국으로부터 체크포인트를 전송 받는 비용이  $C_r^{MC}$ 가 된다.

#### 4.2 No Logging Redundant Lazy 기법

No Logging Redundant Lazy 기법에서  $N_c(t)=r$ ,  $N_h(t)=0$ 이고 체크포인트링이 일어난 후 이전 체크포인트링을 삭제하는 비용은  $2N_h C_m^{mc}$ 이고  $C_h$ 는  $2 C_m^{mc}$ 이기 때문에  $C_{01}^{mc}$ 과  $C_{01}^{MC}$ 은 다음과 같이 계산된다.

$$C_{01}^{mc} = r\alpha C_c^{mc} + 2N_h C_m^{mc} + 2C_m^{mc}, \quad C_{01}^{MC} = r\beta C_c^{MC}$$

이동 호스트의 결합 복구는 체크포인트를 가지고 있는 기지국에 메시지를 보내 요청한 후 체크포인트를 전송 받음으로 이루어지므로 복구 비용  $C_h$ 에서  $C_r^F$ ,  $C_r^S$ ,  $C_r^{MC}$ 은 다음과 같이 계산한다.

$$C_r^F = (N_h + \alpha)(C_c^{mc} + C_m^{mc}), \quad C_r^S = C_r^F + \delta$$

$$C_r^{MC} = \beta(C_c^{MC} + C_m^{MC}) + N_h C_m^{mc}$$

여기서  $C_r^S$ 의  $\delta$ 는 첫 번째 연결 리스트의 결합 발생 여부를 판단할 때 드는 비용으로 연결 리스트의 기지국의 결합이 일양 분포(uniform distribution)를 가지고 발생한다면  $\delta = N_h C_m^{mc}/2$ 이 된다.  $C_r^{MC}$ 의  $N_h C_m^{mc}$ 은 연결 리스트를 통해 기지국의 결합이 발생했다는 사실을 알기 전까지 연결 리스트의 기지국들에게 메시지를 보내는데 사용한 비용이다. 그러므로 No Logging Redundant Lazy 기법의 총 비용은 식 (1)에 위에서 구한 식을 대입함으로써 구할 수 있다.

#### 4.3 Logging Redundant Lazy 기법

Logging Redundant Lazy 기법은  $N_c(t)=r/k$ ,  $N_h(t)=\rho r$ 이기 때문에  $C_{01}^{mc}$ 과  $C_{01}^{MC}$ 을 다음과 같이 구할 수 있다.

$$C_{01}^{mc} = (r/k)\alpha C_c^{mc} + praC_r^{mc} + praC_m^{mc} + 2N_h C_m^{mc} + 2C_m^{mc}$$

$$C_{01}^{MC} = r\beta C_c^{MC} + pr\beta C_l^{MC} + pr\beta C_m^{MC}$$

여기서  $praC_r^{mc}$ ( $praC_m^{MC}$ )은 메시지 로깅에 대한 비용이고  $praC_m^{mc}$ ( $praC_m^{MC}$ )은 로깅에 대한 기지국의 응답(acknowledgement) 비용이다.

복구 비용  $C_h$ 은 No Logging Redundant Lazy 기법에 로그에 대한 고려만 하면 다음과 같이 구할 수 있다.

$$C_r^F = (N_h + \alpha)(C_c^{mc} + \nu C_l^{mc} + C_m^{mc}), \quad C_r^S = C_r^F + \delta$$

$$C_r^{MC} = \beta(C_c^{MC} + \nu C_l^{MC} + C_m^{MC}) + N_h C_m^{mc}$$

여기서  $\nu$ 는 기지국에 저장되는 로그의 평균 크기로 핸드오프가 포아송(Poisson) 분포를 따른다고 가정하면  $\nu = (k-1)/2$ 가 된다[1]. 그러므로 Logging Redundant Lazy 기법의 총 비용은 식 (1)에 위에서 구한 식을 대입함으로써 구할 수 있다.

#### 4.4 비용 분석 결과

두 기법의 총 비용을 비교하기 위해 먼저 각 기법의 총 비용을

$C_c^{MC}$ 에 대해 정규화(normalization)를 한다. 비용 분석을 위해 각 파라미터를 다음과 같이 가정하였다.

$$C_r^F = 0.1, \quad C_m^{mc} = 10^{-4}, \quad C_c^{MC} = 10, \quad C_l^{MC} = 1, \quad C_m^{MC} = 10^{-3}$$

$$\rho = 0.5, \quad \alpha = 1, \quad \lambda_m = 10^{-5}, \quad \lambda_b = 10^{-7}$$

그림 2와 3은  $r$ 을 0.01에서 100까지 변화시키고 매크로 셀의 무선 네트워크 사용 비용인  $\beta$ 를 각각 마이크로 셀의 무선 네트워크 사용 비용인  $\alpha$ 와 비교해 1배와 5배로 놓았을 때 발생하는 두 기법의 총 비용을 보여주고 있다. 그림에서 볼 수 있듯이 Logging 기법과 No Logging 기법의 총 비용 차이는  $r$ 의 값이 작을수록 차이가 없고 값이 커질수록 Logging 기법의 총 비용이 더 작아지는 것을 볼 수 있다.

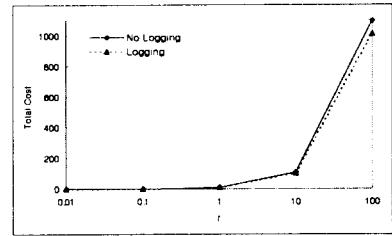


그림 2 두 기법의 총비용 ( $\beta=1$ )

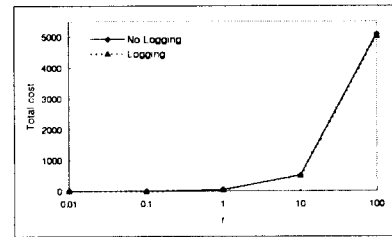


그림 3 두 기법의 총비용 ( $\beta=5$ )

#### 5. 결론

본 논문에서는 결합 허용 기법의 하나인 메시지 로깅과 체크포인트링 기법을 이용하는 Redundant Lazy 기법을 이중 셀룰러 네트워크에 적용하는 문제에 대해 논의하였다. 구체적으로 Lazy 기법의 단점을 해결한 Redundant Lazy 기법을 이중 셀룰러 네트워크에 적용하고 수정함으로써 Redundant Lazy 기법의 또 다른 문제를 해결하였다.

추가 연구되어야 할 사항으로 Redundant Lazy 기법을 이중 셀룰러 네트워크에 적용하였을 때와 적용하지 않았을 때의 신뢰도를 분석하여 이중 셀룰러 네트워크를 이용하면 신뢰도가 크게 향상된다는 것을 증명하는 것이 필요하겠다. 또한 이동 호스트가 매크로 셀의 경계를 넘나들 때마다 매크로 셀 기지국에 저장되어 있는 자신의 체크포인트 관리 문제를 비용 분석에 포함시켜야 할 것이다.

#### 참고문헌

- [1] D.K. Pradhan, P. Krishna, and N.H. Vaidya, "Recovery in Mobile Environments: Design and Trade-off Analysis," Proceedings of the 26th International Symposium on Fault Tolerant Computing, pp. 16-25, June 1996.
- [2] W. Kwon and S. Kim, "Failure Recovery of Mobile Host in Mobile Environments," Proceedings of International Conference on Parallel and Distributed Processing Techniques and Applications, pp. 2072-2077, June 2000.
- [3] N. Neves and W.K. Fuchs, "Adaptive Recovery for Mobile Environments," Communication of the ACM, Vol. 40, No. 1, pp. 68-74, January 1997.
- [4] B. Yao, K. Ssu, and W.K. Fuchs, "Message Logging in Mobile Computing," Proceedings of the 29th International Symposium on Fault Tolerant Computing, pp. 294-301, June 1999.