

셀룰라 네트워크 환경에서의 효율적인 기지국 결합 복구 기법*

변계섭^o 김재훈
아 주 대 학 교 정보통신전문대학원 정보통신공학과
{ksbyun,jaikim}@madang.ajou.ac.kr

Efficient Failure Recovery Schemes for Mobile Support Station in Cellular Networks*

Kyue-Sup Byun^o Jai-Hoon Kim
Graduate School of Information and Communication, Ajou University

요 약

이동 호스트는 무선 통신망의 낮은 대역폭과 호스트들의 이동성, 부족한 저장장치와 배터리 수명 등으로 인하여 결합 발생 가능성이 높다. 이동 호스트의 결합에 효율적으로 대처하기 위한 결합 허용 기법에 관한 연구가 많이 진행되어 왔다. 셀룰라 네트워크에서는 이동 호스트이외에도 이동 호스트를 연결시키는 기지국은 보다 높은 수준의 가용도를 요구하므로 기지국의 결합에 대한 연구도 필수적이다.

본 논문에서는 이동 호스트 결합 복구를 위한 체크포인트링 기법을 기반으로 기지국 결합 복구를 위한 체크포인트링 이중화 기법을 제안하고 성능을 분석하였다. 또한 이동 호스트의 결합 복구를 위해 체크포인트가 존재하는 기지국의 복구를 기다리는 방법과 다른 기지국의 체크포인트를 이용하는 방법의 성능을 비교 분석하였다.

1. 서론

이동 호스트는 여러 자원의 제약성(즉, 낮은 대역폭과 호스트들의 이동성, 작은 저장장치, 충분하지 못한 배터리 수명)으로 인하여 결합의 발생 가능성이 높다. 이동 호스트 자체의 결합 이외에 네트워크 연결 단절, 무선 링크의 결합 등 기존 유선 네트워크에서의 결합 원인들과는 다른 여러 결합을 포함하고 있다[2]. 그러므로 이동 호스트의 결합 복구에 대한 많은 연구가 진행되어왔다. 이동 호스트의 이동성으로 인해 기존 유선에서의 체크포인트링 기법[8,9]은 이동 컴퓨팅 환경에 적합하도록 변경하여 제안되었다[5,6,10]. 이들 이동 호스트의 제약으로 인해 기지국에 자신들의 정보를 저장[1,5,6]하기도 하고 이동 호스트 자신의 로컬 디스크에 저장할 수도 있다[10]. 그러나, 이들 이동 호스트의 결합 복구에 관한 기법들은 관련된 기지국에 결합이 발생했을 때, 기지국이 복구 될 때까지 기다려야 하는 단점을 지니고 있다.

본 논문에서는 이동 호스트에 결합이 발생했을 때 이를 복구하고 기지국 결합 발생 시에도 이를 해결할 수 있는 복구기법을 제안한다. 이동 호스트의 결합을 해결하기 위한 Pradhan[1]의 Trickle기법을 기반으로 하여 기지국의 결합 복구 기법을 제안하였는데 기본 개념은 체크포인트를 복수개의 기지국에 다중화하는 방법으로 기지국의 가용도를 향상시켰다. 또한, 이동 호스트의 결합시 체크포인트가 존재하는 기지국에 동시에 결합 발생시 기지국 복구를 기다리는 방법과 체크포인트가 존재하는 다른 기지국의 정보를 이용하는 방법의 복구 비용을 비교 분석하였다.

2. 관련연구

이동 호스트의 결합에 대비하여 일정한 간격으로 이동 호스트의 프로세스 정보를 기지국에 저장하는 기법이 제안되었다[1,5]. 기지국에 프로세스 정보를 저장하는 이유는 이동 호스트가 가진 로컬 디스크에 프로세스의 정보를 저장하는 것보다 안전하다고 판단되기 때문이다. 이는 유선 환경에서부터 많은 연구가 진행되어 왔다[6,7]. 또한 이동 호스트는 이동성을 갖기 때문에 주기적인 저장 외에 이동한 이동 호스트가 요청한 동작들에 대해 로그를 기록할 필요가 있다. 이런 로그이나 체크포인트링은 셀룰라 네트워크 기반의 이동 컴퓨팅 환경에서 기지국에 저장되어진다. 이와 같이 이동 컴퓨팅 환경에서 이동 호스트의 결합 복구를 위해서 프로세스 정보를 안전한 저장소에 저장하기 위한 많은 연구가

진행되었다[1,5,8,9]. Pradhan[1]은 이동 호스트에서 발생하는 메시지 로그에 대해서 메시지가 발생할 때마다 동기적으로 메시지 로그를 기록하는 Logging기법과 메시지를 받을 때마다 체크포인트링을 수행하는 No Logging기법으로 나누어 체크포인트링을 수행하였으며, 이 두 기법을 핸드오프마다 체크포인트와 메시지 로그를 관리하는 기법에 따라 세 가지로 구분하고 비용을 분석하였다. 그러나 이동 호스트의 결합에 대하여 복구가 진행되는 동안이나 결합이 발생하지 않은 상황에서도 체크포인트가 존재하는 기지국에 결합이 발생하면 이동 호스트는 즉각적인 서비스를 요하는 업무(중권정보, 전자메일 서비스, 티켓예약 서비스 등)를 더 이상 수행할 수 없게 된다. 이런 기지국 결합 복구 기법에 대하여 Alagar[3]에 의하여 연구 되었다. Alagar는 기지국 결합에 대하여 또 다른 기지국에 이동 호스트에 대한 프로세스 정보를 중복하여 이를 해결하였다.

본 논문에서는 관련연구[1]의 Trickle기법을 기반으로 하여 기지국의 이중화 기법을 제안하였다. Trickle기법을 선택한 이유는 Lazy기법은 기지국을 이중화 하였을 때, Trickle기법에 비하여 핸드오프 시에 많은 네트워크 비용이 소요되기 때문이며 관련연구[3]와 비교하여 [3]에서는 이웃한 기지국에서 중복할 기지국을 선택하는 반면 제안한 이중화 기법은 이동 호스트가 거쳐간 이동 경로를 이용 거쳐간 셀 내의 기지국에 정보를 중복한다는 점에서 다르며, 추가적인 비용이 소요되지 않으므로 효과적이다.

3. Modified Trickle 결합복구기법

이동 컴퓨팅 환경에서의 복구 기법은 Pradhan[1]에 의해 제안되었다. 그러나 Pradhan의 논문에서는 기지국에 결합이 발생하였을 때, 발생하는 비용에 대해 고려하지 않았다. 그림 1에서와 같이 Pradhan[1]이 제안한 Trickle기법은 현재 이동 호스트(MH)가 BS3에 존재할 때, 메시지 로그와 프로세스 상태(체크포인트)를 지역의 기지국(BS3)에 저장한다. 만일 마지막 로그와 체크포인트를 BS1지역에서 BS1지역에서 저장하고 BS2를 거쳐 BS3지역까지 이동했다면, 로그와 체크포인트를 한 홉(hop) 떨어진 기지국(BS2)으로 전송한 후 저장하여 결합 복구시 최대한 홉 떨어진 기지국에서 필요한 로그와 체크포인트를 얻을 수 있도록 한다. 이동 호스트가 이동할 때는 또 다른 셀로 이동하는 핸드오프 중에 컨트롤 메시지를 이전에 속해있던 셀의 기지국(BS2)에 보냄으로써

* 본 연구는 한국과학재단 목적기초연구(2001-1-30300-106-2) 지원으로 수행되었음.

마지막에 속해있던 기지국(BS2)을 알 수 있다. 새로운 셀의 기지국(BS3)에서 체크포인트가 발생하면 이전의 셀의 기지국(BS2)에 저장되어있던 로그와 체크포인트 정보를 삭제한다. 그러나 이동 호스트가 결합이 발생하여 복구도중에 현재 속해 있던 기지국(BS3)에서 체크포인트가 수행되지 않았다면 이전에 속해 있던 셀의 기지국(BS2)에서 속해 있던 셀의 기지국(BS2)에서 로그와 체크포인트에 대한 정보를 얻을 수 있다. 이동 호스트에 결합이 발생하였을 때, 이들 정보들을 이용하여 결합을 복구할 수 있다. 그러나 이러한 Trickle기법은 이동 호스트에 대한 결합만을 고려하였으므로 기지국이 결합을 발생하였을 때는 복구할 수 없는 단점을 가지고 있다. 이런 상황을 극복하기 위해 본 논문에서는 Trickle기법을 기반으로 체크포인트를 이중화하는 기법을 제안한다. Modified Trickle기법은 이동 호스트와 기지국 결합 모두를 복구하기 위한 방안을 제시한다.

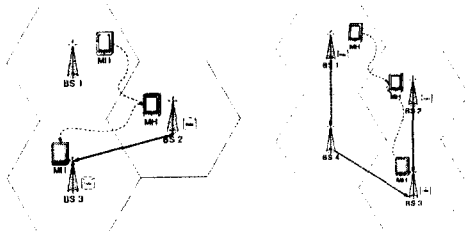


그림 1. Trickle 기법 그림 2. Modified Trickle 기법

그림 2는 Modified Trickle기법을 나타내며 이전 셀의 기지국 역시 결합이 발생하였을 때, 이동 호스트의 정보를 요청하여도 이에 응답하지 못하는 상황에서 그림 2에서와 같이 이동 호스트가 현재의 셀(BS3)로 이동하기 이전 셀의 기지국(BS2)뿐만 아니라 그 이전 셀의 기지국(BS1)에도 이동 호스트의 로그나 체크포인트 정보를 삭제하지 않고 저장하였다가 현재의 셀 바로 이전 셀의 기지국(BS2)에 결합이 발생하였을 때, 다른 경로(BS4)를 이용하여 복구에 필요한 정보를 얻음으로써 이동 호스트의 결합을 극복할 수 있다. 현재 이동 호스트가 지역 기지국에서의 결합 발생시 이동 호스트는 다른 기지국과도 통신이 단절되므로 지역 기지국이 복구될 때까지 기다려야 한다.

4. 비용 분석

본 장에서는 제안하는 Modified Trickle기법의 핸드오프 시에 발생하는 통신 비용, 이동 호스트와 기지국의 결합허용을 위한 추가비용, 기지국이나 이동 호스트에 결합이 발생하였을 때 복구에 소요되는 평균비용을 고려하였다. 또한 Logging과 No Logging 기법에 따라 비용을 비교 분석하였다. 표 1은 성능 분석에 사용된 파라미터의 기호와 설명을 나타낸다.

4.1 시스템 모델링(System Modeling)

한 핸드오프 간격을 그림 3과 같은 상태전이도로 표시할 수 있다. 여기에서 핸드오프 간격은 두 핸드오프 사이의 시간 간격으로 정의한다. 상태 0은 핸드오프의 초기 상태를 말하며, 상태 0에서 1로 전이하는 과정은 핸드오프 시까지 이동 호스트가 쓰기 동작 중에 결합이 발생하지 않았을 때를 의미한다. 상태 0에서 상태 2로의 전이는 이동 호스트가 핸드오프 이전에 결합을 발생시킨 경우를 의미한다.

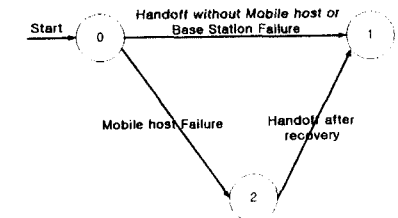


그림 3. Modified Trickle기법의 Markov Chain 모델

본 논문에서는 이동 호스트의 결합 발생률과 기지국의 결합 발생률을 함께 고려하였다. 그러므로 상태 0에서 상태 2로 전이할 확률을 P_{02} 라 하면 이 확률은 다음과 같이 정의된다.

$$P_{02} = \frac{\lambda_m}{\lambda_m + \mu}$$

<표 1 파라미터>

| | |
|---------------------|--|
| λ_m | 이동 호스트의 결합 발생률 |
| λ_b | 기지국의 결합 발생률 |
| μ_b | 기지국의 결합 복구율 |
| γ | C_i / C_c , 인접한 기지국끼리 체크포인트 전송할 때 애플리케이션 메시지 전송비용 |
| α | 무선 네트워크 요소(wireless network factor) |
| $C_{cell/mx}^{mh}$ | x 홉 떨어진 이동 호스트가 기지국에 체크포인트/ 애플리케이션 메시지/ 콘트롤 메시지를 전송하는 평균 비용 |
| $C_{cell/mx}^{bs}$ | x 홉 떨어진 기지국들간에 체크포인트/ 애플리케이션 메시지/ 콘트롤 메시지를 전송하는 평균 비용 |
| $C_{cell/mx}^{mh'}$ | x 홉 떨어진 이동 호스트가 기지국에 결합없이 체크포인트/ 애플리케이션 메시지/ 콘트롤 메시지를 전송하는 평균 비용 |
| $C_{cell/mx}^{bs'}$ | x 홉 떨어진 기지국들간에 결합없이 체크포인트/ 애플리케이션 메시지/ 콘트롤 메시지를 전송하는 평균 비용 |
| N_h' | 현재 속한 기지국으로부터 떨어진 홉(hop) 수 |
| C_i | 이동 호스트가 핸드오프를 수행하는데 드는 총 비용 |
| S | 기지국 간의 홉 수 |
| T_c | 체크포인트 주기, no logging 기법을 위해서 T_c 는 $1/\beta$ 이다. |
| μ | 이동 호스트의 핸드오프율 |
| k | 체크포인트 당 쓰기 동작의 수 |
| r | 통신 이동율, 핸드오프 당 쓰기 동작의 기대 값 $1/\mu$ 과 같다. |
| ϵ | C_m / C_c , 기지국간의 체크포인트를 전송하는 비용에 대한 콘트롤 메시지를 전송하는 비용의 비율 |

상태 전이 (0, 1)의 비용 C_{01} 은 상태 0에서 상태 1로 전이되기 전까지의 예상되는 비용으로 다음과 같이 계산할 수 있다[1]. $C_{01} = (\alpha C_c) \times N_h(T) + (\alpha C_i) \times N_h(T) + C_b$ (1)

이동 호스트가 핸드오프 간격 사이에서 발생시키는 총 비용 C_i 는 핸드오프를 수행하는데 소요되는 총 비용 C_{01} 과 결합 발생시 복구 비용 C_r 합한 것으로 다음과 같이 계산된다. $C_i = C_{01} + P_{02}C_r$ (2)

여기서 C_r 은 이동 호스트의 결합시 복구 비용으로 이동 호스트에서 현재 기지국까지의 무선 네트워크와 기지국들 사이의 유선 네트워크를 모두 사용한다. 이동 호스트가 이동하는 상황에서 결합이 발생할 경우, 이동 호스트의 정보가 저장된 기지국에서 정보를 가져오게 되는데 이 때 소요되는 비용은 아래와 같다(로그를 기록하는데 소요되는 비용과 콘트롤 메시지에 대한 비용 역시 같은 방법으로 계산할 수 있다. 지면상 체크포인팅 비용만을 나타내었다)

유선 네트워크 환경에서 소요되는 비용 : x 홉 떨어진 기지국사이의 통신 비용, ($x > 0$)

$$C_{cr}^{bs} = xC_c + \frac{(x+1)\lambda_b}{(x+1)\lambda_b + \mu_b} \cdot \frac{1}{\mu_b}, C_{cr}^{mh} = xC_c$$

무선 네트워크 환경에서 소요되는 비용 : x 홉 떨어진 기지국과 이동 호스트의 통신 비용, ($x \geq 0$)

$$C_{cr}^{mh} = (\alpha + x)C_c + \frac{(x+1)\lambda_b}{(x+1)\lambda_b + \mu_b} \cdot \frac{1}{\mu_b}, C_{cr}^{bs'} = (\alpha + x)C_c$$

4.2 No Logging/Logging Modified Trickle 기법

Modified Trickle 기법은 기지국의 결합까지 대처하기 때문에 기지국의 결합 발생률도 함께 고려하여야 한다. 이동 호스트가 새로운 기지국의 셀 내로 이동했을 때, 콘트롤 메시지는 이동하기 이전의 기지국에게 이동 호스트의 상태를 요청하기 위한 것이다. 핸드오프 동작 시에 소요되는 비용은 자신이 속한

로컬 기지국으로부터 이동 호스트의 상태 정보를 가져오는 비용과 바로 앞과 전 기지국의 정보를 가져오는 데 소요되는 비용의 합으로 계산할 수 있다. No Logging 기법을 사용하기 때문에 $N_c(T)=r, N_c(T)=0, C_h=C_{cl}^{mh}+2C_{cl}^{bs}$ 이 되고 이것을 식 (1)에 대입하게 되면 비용 C_{01} 과 C_r 은 다음과 같이 계산된다. 여기에서 v 는 핸드오프 동안 발생하는 평균 로그크기를 나타낸다.

$$C_{01} = rC_{cl}^{mh} + C_{cl}^{bs} + N_h'(C_{cl}^{bs} + C_{cl}^{bs}) + N_h' C_{m2}^{bs}$$

$$C_r = a \cdot N_h'(C_{cl}^{mh} + C_{m1}^{mh}) + (1 - N_h')(C_{cl}^{mh} + C_{m0}^{mh})$$

$$b. N_h' \left[\left(\frac{\mu_h}{2\lambda_h + \mu_h} \right) (C_{cl}^{mh} + C_{m1}^{mh}) + \left(\frac{\lambda_h}{2\lambda_h + \mu_h} \right) (C_{cl}^{mh} + C_{m1}^{mh}) + (C_{cl}^{mh} + C_{m0}^{mh}) \right] + (1 - N_h') (C_{cl}^{mh} + C_{m0}^{mh}) + N_h' \left(\frac{\lambda_h}{\lambda_h + \mu_h} \right) T_c$$

그러므로 No Logging Modified Trickle 기법의 총 비용은 식(2)에서 구할 수 있다. 즉, $C_{NLMT} = C_{01} + P_{02}C_r$ 이 된다. 또한 Logging 기법에서는 로그에 대한 정보를 기록하는데 추가 비용이 소요된다. 여기에서 핸드오프 동작이 포아송 분포를 따른다고 가정하면 $v' = (k-1)/2$ 이 되며, 결합 발생시 로그 크기의 기대값을 나타낸다. No Logging 기법에서의 비용에 로그를 전송받는 비용만을 추가로 고려하면 된다. Logging Modified Trickle 기법의 총 비용은 $C_{LMT} = C_{01} + P_{02}C_r$ 이 된다.

5 성능 평가

제한한 Modified Trickle 기법에서 logging 기법에 대한 비용을 분석하기 위해 최적의 체크포인트 간격이 요구된다. Logging 기법에서 최적의 k 는 전체 비용을 최소화할 수 있는 값을 의미한다. Logging 기법에서 전체 비용을 최소화할 수 있는 최적의 k 값을 구하기 위해 아래와 같은 방법을 정의한다.

$$\frac{\partial C_{LMT}}{\partial k} = 0, \frac{\partial^2 C_{LMT}}{\partial^2 k} > 0, \frac{\partial C_{NLMT}}{\partial k} = 0, \frac{\partial^2 C_{NLMT}}{\partial^2 k} > 0$$

위의 방법을 이용하여 최적의 k 값을 구할 수 있다. 비용 분석을 위해 C_c 에 대해서 정규화했다. γ 는 C_c/C_c 와 같다. 또한 ϵ 는 C_c/C_c 와 같다. 그러므로 $C_c = \gamma C_c, C_m = \epsilon C_c, C_c = 1$ 이다. 실제적으로 $C_c \ll C_c$ 라고 가정한다. 결국 위의 식으로부터 $C_c = \gamma, C_m = \epsilon$ 를 얻을 수 있다. 본 논문의 비용 분석에서는 컨트롤 메시지 $C_c = \epsilon = 10^{-4}$, 로깅 비용을 $\gamma = C_c = 10^{-1}$, 쓰기 동작에 대한 사용자 입력의 비율을 $\rho = 0.5$, 무선 네트워크 요소 $\alpha = 10$ 이라고 파라미터를 가정하였다. No logging 기법인 경우 $T_c = 1/\beta$ 로 정의 하였다. 그림 3에서 $\lambda_m = 10^{-2}, \lambda_b = 10^{-4}, r = 10$ 로 정의하였으며, 기지국 결합 발생시 기지국의 복구율에 따른 복구비용을 나타낸다. 기지국의 복구율이 증가할수록 기지국의 복구를 기다리는 비용이 적게 소요됨을 알 수 있다. 즉, 이동 호스트의 정보를 중복하여 저장한 기지국에서 정보를 가져오기 위해 새로운 경로를 설정하는 방법보다 기지국의 복구를 기다렸다가 정보를 가져오는 비용이 적게 소요된다. 그림 4는 $\lambda_m = 10^{-2}, \lambda_b = 10^{-4}$ 에서의 Logging과 No Logging 기법에서의 r 에 따른 전체 비용을 나타내었다. Logging 기법의 경우 $\mu_b = 0.1$ 일 때, $r = 0.1$ 을 기준으로 r 값이 감소할수록 새로운 경로를 선택하는 방법이, 증가할수록 복구를 기다리는 방법이 유리함을 알 수 있다. 기지국에서 복구를 하는 동안 이동 호스트는 기다리는 기법과 새로운 경로를 설정하여서 다른 기지국에 저장되어 있는 이동 호스트들에 대한 체크포인트나 로그에 대한 정보를 가져오게 하는 것이다. 그러나 기지국의 일시적인 결합(transient failure)에 대해서는 이동 호스트가 이를 위해서 기다리는 시간이 상대적으로 적기 때문에 새로운 경로를 설정하는 것보다 기지국의 복구를 기다리는 것이 더 효율적이지만, 기지국의 복구가 지연되었을 때는 새로운 경로를 통한 기법이 이동 호스트의 결합 복구에 더 효율적이다. 표 2는 각 기법별 복구 비용의 비교를 나타낸다. Logging과 No Logging 기법에 비교해보면, 공통적으로 $\mu_b = 10^{-1}$ 과 $\mu_b = 10^{-2}$ 사이에서 기지국 복구율이 증가할수록 복구를 기다리는 것이 유리하며 반면 감소할수록 새로운 경로를 설정하는 방법이 유리함을 알 수 있다.

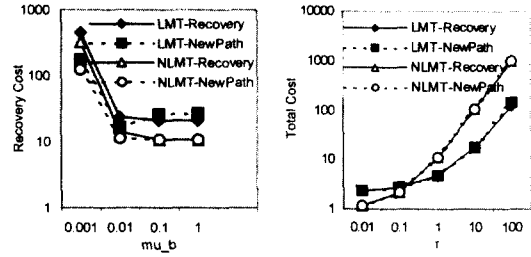


그림 3. μ_b 에 따른 복구비용

그림 4. 전체 비용($\mu_b = 0.1$)

<표 2. Modified Trickle에서 각 기법별 복구 비용 비교>

| 이동성, 기지국 복구율 | Logging 기법 | No Logging 기법 |
|----------------|------------|---------------|
| 이동성이 작으면 | $R < N$ | $R = N$ |
| 기지국 복구율이 큰 경우 | | |
| 이동성이 크면 | $R > N$ | $R > N$ |
| 기지국 복구율이 작은 경우 | | |

*기지국 복구를 기다림(R), 새로운 경로를 설정(N)

하는 것이 유리함을 알 수 있다. 위의 분석결과의 공통적인 사항은 기지국 결합에 대한 복구를 기다리기 때문에 평균 $1/\mu_b$ 의 시간이 지연되며, 이동성이 많은 이동 호스트는 핸드오프 발생 빈도가 많으며 기지국 결합 발생시 이를 복구하는 복구율이 크면 할수록 Logging 기법의 경우 기지국 복구를 기다리는 것이 유리하며, 반대의 경우 본 논문에서 제안한 새로운 경로를 설정하여 결합을 복구하는 Modified Trickle 기법이 유리함을 알 수 있다.

6. 결론 및 향후 과제

이동 호스트의 결합에 대한 복구 기법은 이동 호스트 자체의 제한적인 자원으로 인해 많은 이동 컴퓨팅 환경에서 제약을 받는다. 본 논문에서는 이동 호스트에 결합이 발생했을 때, 이를 복구하고 또한 기지국 결합 발생 시에도 해결할 수 있는 복구기법을 제안하였다. 제한한 기지국 결합 복구 기법은 Pradhan[1]에서 제안된 이동 호스트의 결합 복구 기법 중 Trickle 기법을 변형한 Modified Trickle 기법이다. 본 논문은 기지국 결합에 대하여 고려하였다.

참고 문헌

- [1] D.K. Pradhan, P. Krishna, and N.H. Vaidya, "Recovery in Mobile Environments: Design and Tradeoff Analysis," Proc. of the 26th International Symposium on Fault Tolerant Computing, pp. 16-25, Jun. 1996.
- [2] P. Krishna, N. H Vaidya, and D. K. Pradhan, "Recovery in Distributed Mobile Environments", IEEE Workshop on Advances in Parallel and Distributed System, Oct. 1993.
- [3] S. Alagar and R. Rajagopalan and S. Venkatesan, "Tolerating Mobile Support Station Failures," TR UTDCS-16-93, University of Texas, Dallas, 1993.
- [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, Jun. 1999.
- [5] K.M. Chandy and L. Lamport, "Distributed Snapshots: Determining Global States of Distributed Systems," ACM Trans. on Computer Systems, Vol. 3, No. 1, pp. 63-75, Feb. 1985.
- [6] E. Elnozahy, D. Johnson, and Y.M. Wang, "A Survey of Rollback-Recovery Protocols in Message-Passing Systems," Technical Report CMU-CS-96-181, School of Computer Science, Carnegie Mellon University, Oct. 1996.
- [7] N. Neves and W.K. Fuchs, "Adaptive Recovery for Mobile Environments," Communication of the ACM, Vol. 40, No. 1, pp. 68-74, Jan. 1997.
- [8] R. Prakash and M. Singhal, "Low-Cost Checkpointing and Failure Recovery in Mobile Computing Systems," IEEE Transaction on Parallel and Distributed Systems, pp. 1035-1048, Oct. 1996.