

유비쿼터스 멀티미디어 응용을 위한 그리드 시스템의 서비스 플랫폼

정희원 박은정*, 종신회원 신현식**

Service Platform of Grid Systems for Ubiquitous Multimedia Applications

Eun-jeong Park* *Regular Member*, Heon-shik Shin** *Lifelong Member*

요 약

무선 통신의 발전으로 인하여 PDA, 휴대전화 등의 이동기기를 위한 멀티미디어 서비스의 개발 또한 빠르게 발전하고 있다. 이러한 모바일 멀티미디어 서비스의 개발에는 모바일 컴퓨팅의 요구조건을 효율적으로 제어할 수 있는 플랫폼이 필요하다. 본 논문에서는 무선 통신의 잦은 통신단절과 모바일 디바이스의 시스템 자원부족에 대응하기 위하여 모바일 에이전트 기반의 그리드 시스템을 사용하는 적응형 서비스 플랫폼을 제시하고, 서비스 플랫폼의 활용으로 모바일 원격 교육 또는 화상 회의에 사용되는 RtoA (Ready-to-Attend) 프레임워크를 구성하였다. RtoA는 사용자들에게 원활한 유비쿼터스 서비스를 제공하기 위하여 그리드 시스템을 이용한 핸드오프, 스피커, 리스너 서비스를 제공한다. ns-2를 이용한 성능평가의 결과 RtoA는 모바일 디바이스의 에너지 소모량을 줄이고, 응답시간을 향상시키는데 효율적임이 증명되었다.

Key Words : ubiquitous computing, grid computing, multimedia services, ubiquitous applications

ABSTRACT

Advances in wireless network are enabling the development of ubiquitous multimedia services. These multimedia services need efficient platforms to comply with the requirements of mobile computing. We introduce an adaptive service platform based on mobile agent and grid systems while specifying the challenges of ubiquitous multimedia services and focusing on frequent disconnections and scarce resources. We applied our platform to framework RtoA (Ready-to-Attend) which supports mobile users to access compute-intensive multimedia service, specifically, mobile education and video conferencing. RtoA includes hand-off, speaker and listener service which enable people to attend a conference or a class with satisfying quality of multimedia service. ns-2 based simulation verifies that our scheme is an efficient way to reduce energy consumption of mobile devices and to improve the response time of mobile applications.

I. 서 론

무선 통신의 발전은 원격 교육, 모바일 화상 회의, 양방향 네비게이션, 원격 진료 등의 유비쿼터스

멀티미디어 서비스를 가능하게 하였다. 이러한 서비스는 대부분 무선통신의 잦은 끊김과 모바일 디바이스의 한정된 시스템 자원으로 인하여 서비스 품질의 저하를 가져오게 된다. 특히 모바일 디바이스

* 서울대학교 전기,컴퓨터공학부 (ejpark@cslab.snu.ac.kr),
논문번호 : KICS2005-11-464, 접수일자 : 2005년 11월 15일

** 서울대학교 전기,컴퓨터공학부 (shinhs@cslab.snu.ac.kr)

의 짧은 배터리 지속시간은 원격 교육이나 모바일 화상회의 응용에서 서비스의 기능여부를 결정짓는 중요한 요소가 된다. 이러한 현상에 대처하기 위하여 모바일 디바이스와 사용자의 컨텍스트(context)에 적응하는 다양한 접근방법이 제시되어 왔고, 그 중 서비스 지속시간을 결정하는 에너지 소모량의 최소화를 위한 방법들은 하드웨어 및 소프트웨어 방면에서 활발히 연구되었다.

본 논문에서는 모바일 디바이스에서 유비쿼터스 멀티미디어 서비스를 지원하는 서비스 플랫폼 RtoA (Ready-to-Attend)를 소개한다. RtoA는 이동 기기의 에너지 소모량을 최소화하여 멀티미디어 서비스의 지속시간을 확장하고, 동적 QoS(quality of service) 조정을 제공한다. 이를 위하여 이동 기기의 소프트웨어 컴포넌트 중 에너지 요구량이 많은 부분을 그리드 시스템에서 수행하는 소프트웨어 이전(software migration) 기법을 이용하였다. 또한 통합적 재구성 알고리즘을 통하여 소프트웨어 이전과 서비스의 QoS 조정시기와 정도를 상호 보완적으로 결정하였다. 이러한 통합적 재구성 기법을 기반으로 유비쿼터스 멀티미디어 서비스의 구현에도 제시하였다. RtoA가 대상으로 하는 모바일 멀티미디어 환경은 그림 1과 같다.

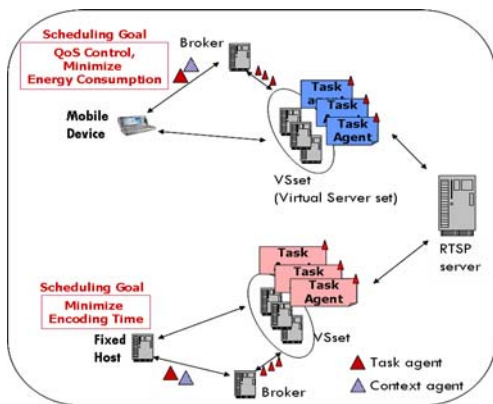


그림 1. 유비쿼터스 멀티미디어 서비스 환경

본 논문의 구성은 다음과 같다. II장에서는 모바일 컴퓨팅과 그리드 시스템의 관련 연구를 소개하고, III장에서는 RtoA 서비스 플랫폼의 구조와 소프트웨어 재구성 기법을 소개한다. IV장에서는 플랫폼을 기반으로 구현되는 멀티미디어 서비스의 예를 제시했고, V장에서 ns-2 기반의 RtoA 시뮬레이터를 통한 성능평가 결과를 보인 후, VI장에서 결론으로 마친다.

II. 모바일 컴퓨팅과 그리드 시스템

소프트웨어의 재구성 방법 중 소프트웨어 이전 기법은 모바일 컴퓨팅 분야에서 꾸준히 사용되어 왔다. KMOT^[1]과 Possession^[2] 시스템은 모바일 디바이스와 프락시 서버간의 소프트웨어 배치와 동적 재구성 방식을 제시하였고, ProxyVM^[3]에서는 클라이언트 자원 집중적(resource-intensive) 요소를 베이스 스테이션이나 프락시 서버로 오프로딩(offloading) 하는 기법을 선보였다. PARM^[4]은 에너지 소모량을 최소화하기 위하여 소프트웨어 컴포넌트들을 선택하여 프락시 서버로 이전시키는 방법을 소개했다. 이와 같이 대부분의 연구들은 소프트웨어 컴포넌트의 이전기법 자체를 중요시하거나, 에너지 소모량을 최소화하는 기법에만 집중했다. 그러나, 이러한 시스템은 여러 모바일 디바이스를 지원해야하는 프락시 서버로 부하집중을 이루게 되고, 프락시 서버의 성능에 따라 모바일 서비스의 성능이 결정된다. 또한 확장성이 떨어지므로, 많은 수의 모바일 디바이스를 지원하지 못한다. 이에 본 논문에서는 단일 프락시 서버를 대체하여 그리드 시스템을 이용한 소프트웨어 이전기법과 유비쿼터스 교육 및 협업 응용을 위한 서비스 구현을 소개한다. 본 논문의 그리드 시스템은 JXTA, P2P 등 다양한 구현방법을 사용하여 확장 가능한 시스템으로 가정한다.

본 논문의 그리드 시스템을 이용한 소프트웨어 재구성은 모바일 디바이스에서 필요한 에너지 소모량의 최소화 뿐 아니라, 그리드 시스템의 부하 분산을 고려하여 소프트웨어의 이전을 수행하였다. 또한 그리드 시스템과 모바일 디바이스의 가용자원에 대한 컨텍스트에 따라 멀티미디어 서비스의 품질을 동적으로 조정하는 부분을 통합하여 상호 보완적인 결정을 내리게 하였다.

전통적 그리드 시스템에서 모바일 디바이스를 포함한 그리드 시스템으로 확장될 때의 가장 큰 문제점은 무선 네트워크의 잦은 통신 단절이다. 최근의 연구에서 이러한 문제를 해결하기 위한 방법들을 제시해왔는데, 대부분의 기법들은 큐잉(queueing)과 자동응답(autonomous action)을 결합한 형태로 나타난다. 큐잉 기법은 통신단절시, 프로세스와 데이터를 그리드 시스템의 큐잉 서버(queueing server)에 일시적으로 저장하는 방법으로 SOAP 라우터, 모바일 큐잉 서버, keep-alive 서버 등의 형태에서 볼 수 있다^[5-7]. 통신단절로 인하여 일시적으로 중단된 프로세스는 재연결이 가능한 상태가 감지되면, 자동

응답(autonomous action)을 통하여 수행을 계속해야 한다. 자동 응답은 여러 형태의 구현이 가능한데, 최근의 연구들에서는 모바일 에이전트(mobile agent)를 이용한 자동응답이 활발히 연구되어 왔다. 모바일 에이전트는 소프트웨어의 현재 상태를 유지하면서, 한 호스트에서 다른 호스트로 이동 가능한 소프트웨어 컴포넌트로, 부하분산, 서비스 맞춤화, 결합 허용 등의 다양한 목적으로 모바일 컴퓨팅에서 이용되어 왔다⁸⁾.

OSGi 등의 일반적인 그리드 서비스의 성능을 개선하기 위한 방법으로 모바일 에이전트와 그리드 서비스의 통합도 주장되고 있다⁹⁾. 전통적인 그리드 시스템은 끊임없는 통신 상태를 가정한 서비스를 제공하므로, 유연성이 부족하다는 단점을 갖고 있는데, 모바일 에이전트를 이용한 그리드 서비스는 시스템의 유연성을 보완하는데 좋은 구현방법이 되고 있다.

본 논문의 RtoA 서비스 플랫폼은 모바일 디바이스의 가용자원과 소프트웨어의 프로파일을 참조하여, 에너지 소모량이 많은 태스크 에이전트를 그리드 시스템으로 스케줄링 하는 방법을 이용한다. 이를 위하여 미들웨어는 모바일 디바이스의 자원상태, 통신상태, 사용자의 이동 등을 모니터링하여 컨텍스트를 구성한다(그림 2).

```

<Service>
<Node-Context>
  <Network-Context>
    <Wireless-Bandwidth> 5Mbps </Wireless-Bandwidth>
    <Disconnection-Rate> 0.1 </Disconnection-Rate>
  </Network-Context>
  <Device-Context>
    <CPU> 1166 MTOPS </CPU>
    <Available-Memory> 100MB </ Available-Memory>
    <Residual-Battery> 30% </Residual-Battery>
  </Device-Context>
  ...
</Node-Context>
</Service>
    
```

그림 2. 모바일 디바이스의 컨텍스트

III. RtoA 서비스 플랫폼

유비쿼터스 멀티미디어 서비스를 위한 그리드 시스템의 서비스 플랫폼은 그림 3과 같다. RtoA 플랫폼은 모바일 디바이스와 그리드 시스템의 마스터인 브로커 서버에 각각 위치하고, 미들웨어 계층은 기본 미들웨어와 서비스 미들웨어 계층으로 나뉜다. 모바일 디바이스와 브로커 양측의 서비스 미들웨어는 서로 협력하여 모바일 디바이스의 소프트웨어 컴포넌트를 재구성하게 된다. 앞서 설명한 바와 같이, RtoA의 소프트웨어 컴포넌트는 모바일 에이전

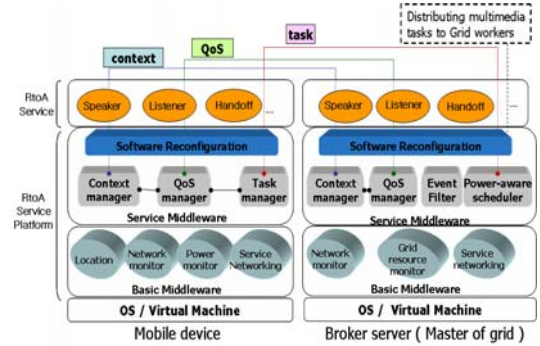


그림 3. RtoA 서비스 플랫폼 구조

트로 구현되어 있고, 작업의 크기에 따라 다수의 에이전트로 분할 가능한 작업들이다.

3.1 시스템 모델

모바일 디바이스의 에너지 소모량을 최소화하기 위하여, 각 모바일 에이전트는 1)그리드로 이전하여 수행하는데 필요한 모바일 디바이스에서의 에너지 소모량과 2)이전되지 않고 모바일 디바이스에서 수행될 때의 에너지 소모량을 각각 예측해야 한다. 본 논문의 그리드 시스템은 이질적 노드들로 구성되어 있으므로, 각 에이전트의 배치에 따라 응답시간과 에너지 소모량 면에서 차이를 가지게 된다. 에너지 소모량은 연산과 통신에 소모되는 에너지 값으로 예측되는데, 이는 미들웨어에서 모니터링한 표 1의 컨텍스트를 기준으로 계산된다.

표 1. 소프트웨어 재구성을 위한 컨텍스트

모바일 에이전트 a_i 의 컨텍스트	
QoS	Q_i
연산을 위한 에너지 소모율	P_i^{comp}
스터브(stub)를 위한 에너지 소모율	P_i^{stub}
연산량	$Size_i^{comp}$
데이터 통신량(전송/수신)	$Size_{st}^{tx}/Size_{st}^{rx}$
스터브 통신량(전송/수신)	$Size_{st}^{st}/Size_{st}^{rx}$
모바일 디바이스 k의 컨텍스트	
잔류 에너지	R_k
가용메모리	AM_k
가용CPU	AC_k
네트워크대역폭(전송/수신)	$DR_{k,up}/DR_{k,down}$
통신 에너지 소모율(전송/수신)	$P_{k,up}/P_{k,down}$
통신단절율	λ_k
위치	L_k
그리드 노드 S_j 의 컨텍스트	
클라이언트수	NC_j
네트워크 대역폭	DR_j
가용 메모리	AM_j
가용 CPU	AC_j

모바일 디바이스 k 는 모바일 에이전트의 집합 $A = \{a_1, a_2, \dots, a_n\}$ 를 갖고 있고, 소프트웨어 이전에 사용되는 그리드 시스템은 호스트의 집합 $S = \{S_1, S_2, \dots, S_m\}$ 로 이루어져 있을 때, 모바일 디바이스 k 의 총 에너지 소모량은 각 에이전트 a_i 의 수행위치 $p(i) \in \{k \cap S\}$ 에 따른 에너지 소모량 $energy_cost_{k,i,p(i)}$ 의 총합으로 다음과 같다.

$$E_k = \sum_{i=1}^n energy_cost_{k,i,p(i)} \quad (1)$$

각 $energy_cost_{k,i,p(i)}$ 는 연산과 통신에 드는 에너지 소모량의 합으로 측정된다. 수행시간을 T_i 라고 할 때, a_i 가 모바일 디바이스에서 수행된다면, 모바일 디바이스 k 는 이 수행을 위하여 다음과 같이 에너지를 소모한다.

$$\left[\begin{array}{l} P_i^{comp} \times T_i \quad (\text{연산용}), \\ \frac{Size_i^{rx}}{DR_{k,up}} \times P_{k,up} + \frac{Size_i^{rx}}{DR_{k,down}} \times P_{k,down} \quad (\text{통신용}). \end{array} \right.$$

반면 a_i 가 그리드 시스템의 $S_{p(i)}$ 호스트로 이전된다면, 모바일 디바이스에서는 a_i 와의 결과 입출력 및 관리를 담당하는 스텔브 에이전트에 의해 다음과 같이 에너지 소모량이 결정된다.

$$\left[\begin{array}{l} P_i^{stub} \times T_i \quad (\text{연산용}), \\ \frac{Size_st_i^{rx}}{DR_{k,up}} \times P_{k,up} + \frac{Size_st_i^{rx}}{DR_{k,down}} \times P_{k,down} \quad (\text{통신용}). \end{array} \right.$$

이제 예측된 총 에너지 소모량 E_k 가 최소화도록 모든 a_i 의 수행위치 $p(i)$ 를 결정해야 한다. 이때 추가로 고려해야할 제약조건은 그리드 시스템의 특정 서버로 과도한 부하가 집중되는 것을 막는 부하분산으로, 이를 위한 수용제어(admission control)를 모델링해야 한다. 이질적 성능을 갖는 호스트 S_j 들로 구성된 그리드 시스템 $S = \{S_1, S_2, \dots, S_m\}$ 에서 호스트 S_j 는 모바일 디바이스 k 의 소프트웨어 이전을 위하여 $Limit_{k,j}$ 의 할당량을 허용하도록 하고, 이 할당량은 $Limit_{k,j} = w_1 \times AM_j / NC_j + w_2 \times AC_j / NC_j$ 로 가용 메모리와 CPU 자원에 대한 가중치 w_1 과 w_2 를 이용하여 최적화하도록 하였다. 모바일 디바이스 k 의 에이전트 a_i 가 노드 S_j 로

이전될 때 ($p(i) = j$)의 부하값을 $workload_{k,i,p(i)}$ 로 할 때, 모바일 디바이스 k 의 에이전트 이전을 위하여 필요한 그리드 노드 S_j 의 자원은 부하함수(workload function) $W_{k,j}$ 로 계산되고, 다음의 수식이 각 그리드 노드의 수용제어를 위한 제약조건으로 고려되어야 한다.

$$\left[\begin{array}{l} W_{k,j} = \sum_{i,p(i)=j} workload_{k,i,p(i)} \\ W_{k,j} \leq Limit_{k,j} \end{array} \right. \quad (2)$$

그리드 노드 S_j 뿐 아니라, 모바일 디바이스 k 에 대한 수용제어도 필요하다^[4]. 이것은 모바일 디바이스의 잔류 에너지량이 줄어들수록 많은 에이전트가 이전되도록 하는 방법으로 모바일 디바이스 k 는 $Limit_{k,on-device} = w_0 \times R_k$ 의 범위 내에서 에이전트를 수행시켜야 한다. 즉, k 의 잔류에너지가 줄어들수록 k 에서의 실행이 허용되는 에이전트의 수가 감소해야 하므로 다음의 수식이 필요하다.

$$\left[\begin{array}{l} W_{k,on-device} = \sum_{i,p(i)=on-device} workload_{k,i,p(i)} \\ W_{k,on-device} \leq Limit_{k,on-device} \end{array} \right. \quad (3)$$

본 논문의 소프트웨어 재구성성을 위한 에이전트의 배치기법은 위에서 설명한 1)에너지 소모량의 최소화, 2)그리드 노드의 수용제어, 3)모바일 디바이스의 수용제어 조건을 포함한 것으로 시스템 모델의 플로우 네트워크는 그림 4와 같다.

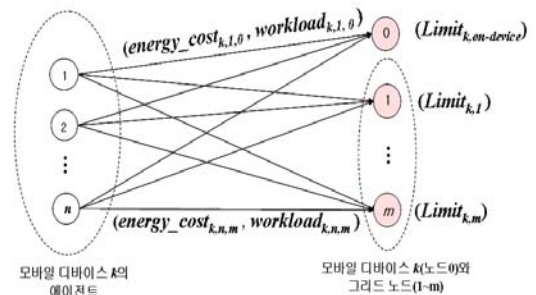


그림 4. 소프트웨어 재구성성의 플로우 네트워크

3.2 소프트웨어 재구성성

앞서 설명한 세 조건을 만족시키는 에이전트의 스케줄링 알고리즘은 NP-hard 문제에 속한다^[10]. 따라서 근사 함수(approximation function)를 사용하여 소프트웨어의 초기배치를 수행하고(Phase 1)^[11, 12],

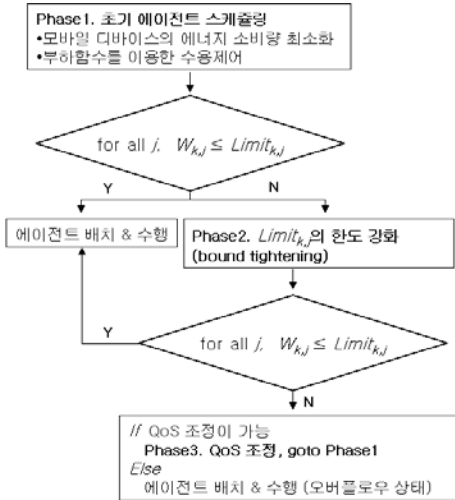


그림 5. RtoA의 소프트웨어 재구성 다이어그램

필요한 경우 선택적으로 $Limit_{k,j}$ 에 대한 제약을 강화하도록 하였다(Phase 2)^[13]. 소프트웨어 재구성 알고리즘의 수행 순서는 그림 5와 같다.

Phase1과 Phase2의 결과, $Limit_{k,j}$ 또는 $Limit_{k,on-device}$ 에 대한 제약조건이 만족되지 못하면, 모바일 디바이스와 그리드 시스템을 이용하여 현재의 품질로는 소프트웨어 컴포넌트를 모두 수행시킬 수 없음을 의미한다. 따라서, Phase1과 Phase2의 수행 이후, 모바일 디바이스와 그리드 시스템의 수용제어결과에 따라 QoS 제어를 수행하도록 한다(Phase3). QoS 제어는 [14]의 예에서처럼 각 QoS 단계별 비디오 파라미터의 변화와 에너지 소모율의 변화를 측정하여 프로파일로 사용가능하다. 즉, 에이전트 a_i 의 P_i^{comp} , P_i^{stub} , $Size_i^{comp}$, $Size_i^{comm}$ 이 줄어들어 스케줄링이 가능하게 된다. 전체 재구성 알고리즘의 의사코드는 그림 6과 같다.

```

1 Software_Reconfiguration() {
2   Initial_Assignment()
3   if ( $R_k > T_k$ ) then
4     if ExcessWL*Set then
5       Bound_Tightening()
6       if ExcessWL*Set then
7         QoS_Control()
8       end
9     end
10  end
11  else Finalize_Migration()
12 }
13 Initial_Assignment() {
14  An initial assignment  $p(i)$  is obtained from an
15  approximation algorithm for the generalized assignment
16  problem [11][12].
17 }
18 Bound_Tightening() {

```

```

13 Let ExcessWL*Set :=  $j : W_{k,j} > Limit_{k,j}$ 
14 Agent := 1
15 while Agent > 0 do
16   Agent := 0
17    $cost_{min} := \infty$ 
18   foreach Agent  $i$  such that  $p(i) = j$ 
19     and  $j \in ExcessWL*Set$  do
20     foreach host  $t$  not in ExcessWL*Set do
21       if  $Limit_{k,t} - W_{k,t} > workload_{k,t}$  then
22          $cost := \frac{energy\_cost_{k,i,t} - energy\_cost_{k,i,j}}{\min\{workload_{k,i,j}, W_{k,j} - Limit_{k,j}\}}$ 
23         if  $cost < cost_{min}$  then
24            $cost_{min} := cost$ 
25           Agent :=  $i$ 
26           old :=  $j$ 
27           new :=  $t$ 
28         end
29       end
30     end
31   end
32   if Agent > 0 then
33      $p(Agent) := new$ 
34      $W_{k,old} := W_{k,old} - workload_{k,Agent,old}$ 
35      $W_{k,new} := W_{k,new} + workload_{k,Agent,new}$ 
36     if  $W_{k,old} \leq Limit_{k,old}$  then
37       ExcessWL*Set := ExcessWL*set - {old}
38     end
39   end
40 end
41 QoS_Control() {
42   while (ExcessWL*Set) do
43     VictimAgent := 1, Gain_by_Degrademax := 0
44     MaxExcessWL* :=  $j : \max_{j \in S} (Limit_{k,j} - W_{k,j})$ 
45     foreach Agent  $i$  such that  $p(i) = MaxExcessWL*$  do
46       Gain_by_Degrade(i) := ( $workload_{k,i,p(i)}(Q) - workload_{k,i,p(i)}(Q+1)$ ) +  $a(energy\_cost_{k,i,p(i)}(Q) - energy\_cost_{k,i,p(i)}(Q+1))$ 
47       if Gain_by_Degrade_max < Gain_by_Degrade(i) then
48         VictimAgent :=  $i$ 
49         Gain_by_Degrade_max := Gain_by_Degrade(i)
50       end
51     end
52      $Q_{VictimAgent} := Q_{VictimAgent} + 1$ 
53     update  $W_{k,MaxExcessWL*}$ 
54   end
55 }
56 Finalize_Migration() { Finalize the migration process by
57   preparing for the end of the mobile device's battery life }

```

그림 6. 소프트웨어 재구성 알고리즘

IV. RtoA 서비스

RtoA 서비스 플랫폼을 이용하여 구현하는 유비쿼터스 멀티미디어 서비스로 원격교육 또는 화상회의를 위한 세가지 서비스를 설계하였다. 세가지 서비스는 멀티미디어 서비스를 위한 speaker, listener 서비스, 모바일 디바이스의 이동성을 지원하는 핸드 오프 서비스로, 유비쿼터스 응용을 위한 이러한 서비스는 MCU(multipoint control unit)를 구비한 시스템 환경에서 주로 구현되어왔으나, 본 논문에서는 MCU 없는 일반적인 그리드 시스템을 이용한 서비스 방식을 소개한다.

4.1 Speaker 서비스

그림 7은 speaker가 모바일 디바이스에서 세션을

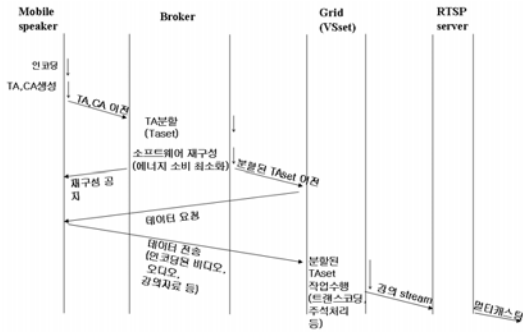


그림 7. 모바일 speaker 서비스

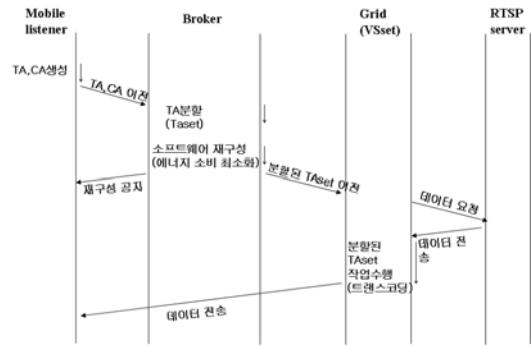


그림 9. 모바일 listener 서비스

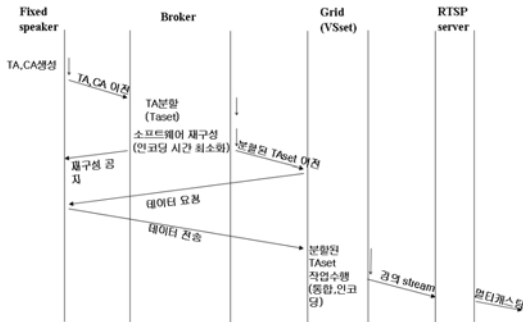


그림 8. 고정 speaker 서비스

시작한 경우의 서비스 수행단계를 보여준다. 이 경우, 멀티미디어 데이터는 전송비용 대 연산비용을 고려한 재구성 알고리즘의 스케줄링 결과로 인하여, 모바일 디바이스에서 인코딩 되도록 태스크 에이전트(TA)들이 배치되고, 인코딩 파라미터도 현재의 컨텍스트를 반영한 컨텍스트 에이전트(CA)에 따라 조정될 것이다. 그러나, 강의 텍스트 등의 다양한 자료와 비디오의 통합, 주석추가(annotation), 통역서비스, text to speech, speech to text 등의 부가 작업은 모바일 디바이스의 연산능력과 배터리 소모를 고려할 때 그리드 시스템에서 수행되는 것이 타당하고, 이는 RtoA의 스케줄링 결과로 나타나게 된다. 즉, 분할된 태스크 에이전트의 집합(TAset)이 그리드 시스템을 대상으로 스케줄링되어 수행된다.

반면, 고정 호스트에서 speaker의 세션이 시작될 경우에는 그림 8과 같이 에너지 소모량 대신 응답 시간을 최소화하는 일반적인 그리드 스케줄링 알고리즘을 수행하도록 한다. 고속의 네트워크를 가정하면, 재구성 알고리즘의 스케줄링 결과로 인코딩 태스크의 많은 부분이 그리드 시스템에서 이루어질 수 있다. 실제로 네트워크의 그리드 시스템을 이용한 비디오 인코딩 기술은 좋은 성능을 나타내고 있다¹⁵⁾.

4.2 Listener 서비스

모바일 디바이스를 위한 멀티미디어 서비스는 주로 두가지 형태로 나타나게 된다. 하나는 접속-지향적(connection-oriented) 서비스와 스트림 버퍼링을 사용하는 것이고, 다른 하나는 로컬 프라시 서버에서 스트림을 트랜스코딩하여 모바일 디바이스의 QoS 수준과 자원에 최적화되도록 하는 방법이다. 그림 9의 listener 서비스는 그리드 시스템(VSset)의 유휴 자원을 이용하여, 모바일 사용자의 컨텍스트에 맞춤화된 멀티미디어 서비스를 제공하는 것이 목적이다. 결과적으로 VSset은 모바일 listener 입장에서 볼 때, 어플리케이션과 통신하고 QoS를 지원하는 멀티미디어 서버를 에뮬레이트하게 된다. 잔류 에너지가 적은 모바일 listener를 위하여 VSset에서는 적정 수준의 QoS로 트랜스코딩을 수행하게 되고, 그 결과 적은양의 네트워크 트래픽과 연산을 하게 되어, 모바일 디바이스의 에너지 소모량을 줄일 수 있다. 또는 다대다 방식의 화상회의를 위한 통합인코딩을 VSset에서 수행하여 MCU(multipoint control unit)가 없는 환경에서도 다대다 화상회의 서비스를 구현할 수 있다.

4.3 핸드오프 서비스

RtoA는 서비스의 투명성과 지속성을 위하여 핸드오프 서비스를 제공한다(그림 10). 모바일 listener가 Broker1이 지원할 수 없는 Broker2의 지역으로 이동할 경우, Broker2는 모바일 디바이스에게 새로운 컨텍스트 에이전트(CA)를 요청한다. 생성된 컨텍스트 에이전트는 Broker2로 이전하고, Broker1에 위치한 태스크 에이전트의 정보도 Broker2로 넘겨진다. Broker2에서는 트랜스코딩 작업을 새로운 가상서버 집합인 VSset2에 분산시키기 위하여 새로운 재구성 알고리즘을 수행하고, 결과를 모바일 listener에게 공지한다. Broker2는 VSset1에서 트랜스코딩

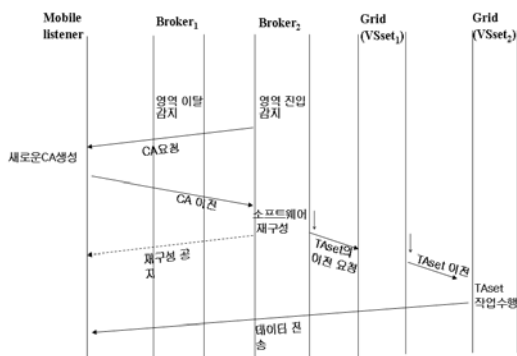


그림 10. 핸드오프 서비스

작업을 수행하고 있는 태스크 에이전트 TAsset에게 VSset2로 이전할 것을 명령하고, 태스크 에이전트들은 VSset2에서 작업을 계속 수행하게 된다.

4.4 RTSP 서버와 멀티캐스팅

위에서 설명된 서비스들의 마지막 단계에 위치한 RTSP 서버는 RTSP(real-time streaming protocol)를 구현한 서버로, 정지, 되감기, 빨리감기 등의 비디오 동작을 지원하는 표준 RTSP 서버이다. RTSP는 전송 계층으로 RTP(real-time protocol)를 사용하고, 송신측에 현재 전송의 품질을 피드백하기 위하여 RTCP(real-time control protocol)를 이용한다. 앞서 설명된 RtoA 서비스에서 RTSP 서버는 동기화된 사용자들을 위하여 그리드 시스템에서 생성된 스트림을 입력으로 사용하고 있다. 비동기화된 사용자들을 위하여 저장 서버를 추가할 경우, RTSP 서버는 저장 서버의 스트림을 읽어, 서비스를 완성하게 된다.

RTSP의 비디오 스트리밍 기능 외에도 동시에 여러명의 수신자를 허용하기 위하여 멀티캐스팅 기능이 필요하다. 멀티캐스팅이 가능한 네트워크 환경일 경우, IGMP(internet group management protocol)를 이용하여 다자간 회의 및 교육의 유비쿼터스 응용이 가능하나, 이러한 환경이 아니더라도 응용 레벨의 멀티캐스트 오버레이 네트워크(MON)를 통하여 서비스가 가능하다.

이상의 유비쿼터스 멀티미디어 서비스를 종합하면, RtoA 서비스 플랫폼을 사용하여 모바일 사용자와 고정 사용자 모두를 지원하는 서비스의 구성이 가능하다. 모바일 speaker의 경우, 모바일 디바이스의 에너지 소비를 최소화하도록 소프트웨어를 재구성하여 그리드 시스템으로 이전시키고, 고정 speaker는 서비스 응답시간을 최소화하도록 재구성하여

그리드 시스템을 이용한다. speaker 서비스의 결과로 생성된 멀티미디어 스트림은 비동기적 사용자들을 위하여 저장 서버에 저장되기도 하고, 동기적 사용자들을 위하여 RTSP 서버를 통하여 즉시 listener 측의 그리드 시스템으로 도달하게 된다. listener 측의 그리드 시스템은 모바일 디바이스의 컨텍스트에 적합한 QoS 수준이나, 다대다 화상회의를 위한 트랜스코딩을 수행하여 모바일 디바이스의 에너지 소비를 감소시킬 수 있다.

V. 성능평가

5.1 실험환경

본 논문에서 제시된 소프트웨어 재구성 기법의 성능을 측정하기 위하여 ns-2 기반의 시뮬레이터를 작성하였다(그림 11). RtoA 시뮬레이터는 IBM의 Aglet^[16]을 모바일 에이전트로 모델링하여 소프트웨어 이전을 수행하였고, Evalvid^[17]의 mpeg 모델을 ns-2에 추가하였다. CPU는 인텔의 성능자료를 사용하였고^[18], 그 외의 시스템 사양과 배터리 모델은 [19]의 실제 시스템에서 측정된 파라미터를 이용하였다.

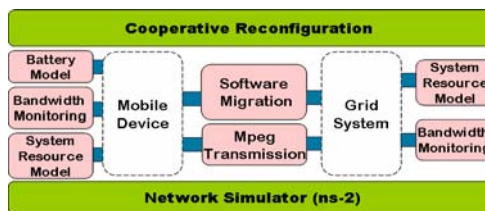
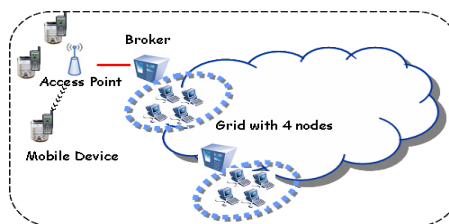


그림 11. ns-2 기반의 RtoA 시뮬레이터



System model			
Type	System	CPU (MTOPS)	Mem.
Mobile Device	iPAQ 3650	Intel StrongArm 206MHz (1166)	32MB
Node1	Itanium2	Intel Itanium2 1.3GHz (8450)	4GB
Node2	Itanium2	Intel Itanium2 1.3GHz (8450)	4GB
Node3	Pentium4	Intel Pentium4 2.4GHz (6400)	1.5GB
Node4	Pentium4	Intel Pentium4 2.4GHz (6400)	1.5GB

Battery model of a mobile device (avg. power rate for application a_i and mobile device k)

P_i^{comp}	P_i^{stub}	P_{idle}	$P_{k,up}$	$P_{k,down}$
2.15	1.8	1.5	2.05	1.95

(J/s)

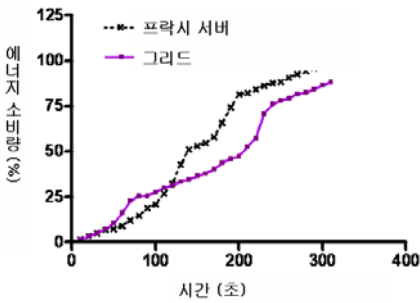
그림 12. 실험환경

실험을 위한 시스템 구성은 그림 12와 같다. 무선 네트워크의 데이터 전송률은 1000Kb/s이고, 모바일 디바이스는 시뮬레이션 공간 내에서 랜덤한 방향으로 이동한다. 본 실험에서 그리드 노드는 소프트웨어 이전을 위하여 30%의 자원을 사용하도록 설정하였다.

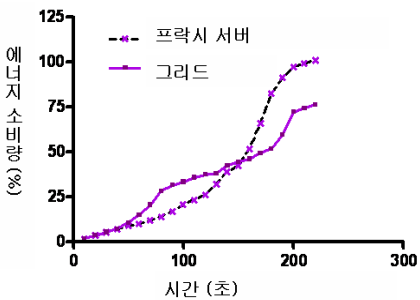
5.2 실험결과

모바일 디바이스는 RtoA 소프트웨어 재구성 알고리즘을 통해 현재의 잔류 에너지량에 근거하여 태스크 에이전트를 모바일 디바이스에서 수행시키거나, 그리드 시스템으로 이전하였다. 그리드 시스템을 이용한 재구성과 단일 프락시 서버를 이용한 재구성의 성능을 비교하기 위하여, 단일 프락시 서버는 그림 12의 시스템 모델 중 Node1의 2배의 시스템 성능을 갖도록 구성하였다. 그리드 시스템도 Node1과 Node2의 성능이 Node3와 Node4 보다 뛰어난 이질적인 노드들로 구성하였다.

성능적으로는 모바일 디바이스의 에너지 소비량(그림 13), 모바일 디바이스의 배터리 지속시간(그림 14), 평균 응답시간(그림 15), QoS 제어와의 통합(그림 16)을 측정하였다.



(1) 분할된 에이전트의 수 : 40



(2) 분할된 에이전트의 수 : 50

그림 13. 모바일 디바이스의 에너지 소비량

그림 13은 각각 작업량(데이터와 연산량)이 4:5인 작업에 대한 소프트웨어 재구성을 실험한 결과이다. 그림 13(1)은 40개로 분할된 태스크 에이전트가 4개의 그리드 노드를 대상으로 재구성될 때와 단일 프락시 서버를 대상으로 재구성될 때의 에너지 소비량을 나타내고, 그림 13(2)는 50개의 분할된 태스크 에이전트에 대한 에너지 소비량을 나타낸다.

그래프에서 보는바와 같이 서비스 초기에는 단일 프락시 서버를 이용한 재구성 기법의 에너지 소모량이 그리드 시스템을 이용한 재구성기법에 비해 더 적다. 이는 프락시 서버의 시스템 성능이 그리드를 구성하는 개별 노드보다 높고, 프락시 서버의 재구성 기법이 모바일 디바이스의 에너지 소비를 최소화하는 방향으로만 에이전트의 이전을 수행했기 때문이다. 반면, 그리드의 재구성 기법은 에너지 소모량의 최소화뿐만 아니라, 그리드 시스템에서의 부하분산도 고려해야 하므로, Node1과 Node2에서의 자원경쟁을 예측하고, 미리 에이전트들을 Node3와 Node4로 분산시켰기 때문에 초기의 에너지 소모량이 프락시 서버에 비해 늘어났다. 그러나, 서비스 시간이 늘어남에 따라, 프락시 서버는 자원경쟁이 심화되어, 에너지 소모량이 급격히 늘어나게 되므로, 그리드 시스템에 의한 에이전트의 재구성기법이 더 긴 배터리 지속시간을 갖게 되고(그림 14), 응답시간에서도 좋은 성능을 보였다(그림 15).

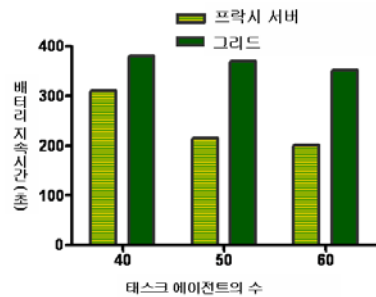


그림 14. 모바일 디바이스의 배터리 지속시간

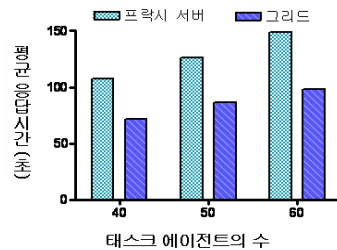


그림 15. 에이전트의 평균 응답시간

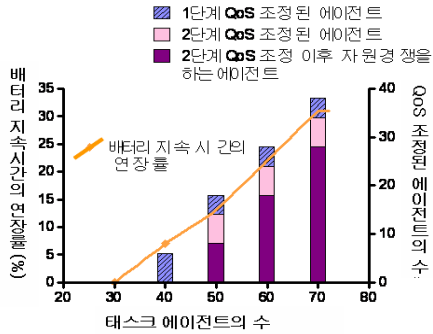


그림 16. 소프트웨어 이전과 QoS 제어의 통합

QoS 조정과 소프트웨어 이전의 통합에 대한 실험결과는 그림 16과 같다. 본 실험에서는 QoS 수준 Q_i 를 {FL1, FL2, FL3}의 세단계로 나누어 각 에이전트는 최대 두 번의 QoS 조정이 가능하도록 실험하였다. 그림 16에서 좌측의 Y축은 모바일 디바이스의 배터리 지속시간의 연장률을 나타내고, 우측의 Y축은 QoS 조정된 태스크 에이전트의 수를 나타낸다. 30개 이하의 분할된 태스크 에이전트가 있는 상태에서는 QoS 조정 없이 모든 태스크가 그리드 시스템과 모바일 디바이스 안에서 수용가능하였다. 그러나, 40개 이상의 분할된 태스크 에이전트가 있는 상태에서는 그리드 시스템의 수용제어 알고리즘에 의해서 QoS 조정된 에이전트가 나타났고, 50개 이상의 에이전트에서는 두 번의 QoS 조정에도 불구하고 그리드 시스템에서 허용한 자원보다 많은 자원을 요구하게 되었다. 70개의 태스크 에이전트가 수행될 때는 약 54%의 태스크 에이전트가 QoS 조정되었고, 배터리 지속시간에서 약 32%의 연장률을 보였다.

V. 결론

본 논문에서는 유비쿼터스 멀티미디어 서비스를 지원하는 그리드 시스템 기반의 소프트웨어 재구성 기법을 제시하였다. 소프트웨어 재구성은 모바일 디바이스에서 멀티미디어 서비스의 지속시간을 확장하기 위하여, 에너지 소모량을 최소화하고, 그리드 시스템의 부하분산을 수행하는 소프트웨어 이전 기법을 이용했다. 또한 이전 기법과 협력적으로 동작하는 QoS 제어를 통하여 시스템 자원을 효율적으로 관리하였고, 무선 네트워크의 잦은 통신단절에 효과적으로 대응하고자, 모바일 에이전트 기법을 사용하였다. 이러한 기법들을 서비스 플랫폼 형태로 제공

하였고, 플랫폼 활용의 한 예로써, 모바일 디바이스를 이용한 원격강의용 멀티미디어 서비스를 구성하였다.

참고 문헌

- [1] T. Kunz, "Mobile Code Middleware for Mobile Multimedia Information Access," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol.6, pp. 68-70, October 2002.
- [2] M. Mochizuki and H. Tokuda, "Possession System: Middleware for Adaptive Multiuser Applications in a Mobile Environment," *Proceedings of the IEEE International Conference on Distributed Computing Systems*, pp. 198-205, May 1999.
- [3] V. Venkatachalam, L. Wang, A. Gal, C. Probst, M. Franz, "ProxyVM: A Network-based Compilation Infrastructure for Resource-constrained Devices," *Technical Report No. 03-13*, Department of Computer Science, University of California, Irvine, March 2003.
- [4] S. Mohapatra and N. Venkatasubramanian, "PARM: Power Aware Reconfigurable Middleware," *Proceedings of the International Conference on Distributed Computer Systems*, pp. 312-320, May 2003.
- [5] A. Hampshire, "Extending the Open Grid Services Architecture to Intermittently Available Wireless Networks," *Proceedings of UK eScience All Hands*, 2004.
- [6] S. Kurkovsky, Bhagyavati and A. Ray, "Modeling a Grid-based Problem-solving Environment for Mobile Devices," *Journal of Digital Information Management*, vol.2. no.2, 2004.
- [7] S. Park, Y. Ko and J. Kim, "Disconnected Operation Service in Mobile Grid Computing," *Proceedings of the International Conference on Service Oriented Computing*, pp. 499-513, 2003.
- [8] A. Fuggetta, G. P. Picco and G. Vigna,, "Understanding Code Mobility," *IEEE Tran-*

sactions on Software Engineering, vol.24. no.5, pp. 342-361, 1998.

[9] W. Zhang, J. Zhang, D. Ma, B. Wang and Y. Chen, "Key Technique Research on Grid Mobile Service Devices," *Proceedings of the International Conference on Information Technology for Application*, pp. 142-150, 2004.

[10] R. K. Ahuja, T. L. Magnanti and J. B. Orlin, "Network Flows," *Prentice Hall*, 1993

[11] K. Jansenand and L. Porkolab, "Improved Approximation Schemes for Scheduling Unrelated Parallel Machines," *Proceedings of the ACM Symposium on Theory of Computing*, pp. 408-417, 1999.

[12] D. B. Shmoys and E. Tardos, "An Approximation Algorithm for the Generalized Assignment Problem," *Mathematical Programming*, vol. 62, no. 3, pp. 461-474, 1993.

[13] E. Park and H. Shin "Software-migration Policy for Power-aware Mobile Computing under Network Resource Constraint," *Proceedings of the International Workshop on Mobility Aware Technology and Applications*, pp. 25-29, 2004.

[14] S. Mohapatra, R. Cornea, N. Dutt, A. Nicolau and N. Venkatasubramanian, "Integrated Power Management for Video Streaming to Mobile Handheld Devices," *Proceedings of the ACM International Conference on Multimedia*, pp. 582-591, 2003.

[15] K. van der Raadt, Y. Yang, H. Casanova, "Practical Divisible Load Scheduling on Grid Platforms with APST-DV," *Proceedings of the International Parallel and Distributed Processing Symposium*, pp. 29b-29b, April 2005.

[16] IBM Tokyo Research Lab. "Aglet: Mobile Java Agents," <http://www.trl.ibm.com/aglets>

[17] C. Ke, "How to Evaluate MPEG Video Transmission using the NS2 Simulator," http://hpds.ee.ncku.edu.tw/~smallko/ns2/Evalvid_in_NS2.htm

[18] Intel Corporation, "Intel Boxed Processors," <http://support.intel.com/support/processors/sb/CS-017346.htm>

[19] Z. Li, C. Wang and R. Xu, "Task Allocation for Distributed Multimedia Processing on Wirelessly Networked Handheld Devices," *Proceedings of the International Parallel and Distributed Processing Symposium*, pp. 79-81, 2002.

박 은 정 (Eun-jeong Park)

정회원



1996년 2월 이화여자대학교 컴퓨터학과 졸업
 2000년 2월 서울대학교 전기, 컴퓨터공학부 석사
 2000년 3월~현재 서울대학교 전기, 컴퓨터공학부 박사과정
 <관심분야> 모바일 컴퓨팅, 분산

컴퓨팅, 미들웨어

신 현 식 (Heon-shik Shin)

종신회원



1973년 2월 서울대학교 응용물리학과 졸업
 1980년 2월 미국 텍사스대학교 의공학과 석사
 1985년 2월 미국 텍사스대학교 전기, 컴퓨터공학과 박사
 1986년~현재 서울대학교 전기,

컴퓨터공학부 교수

<관심분야> 실시간 계산, 분산 시스템, 모바일 컴퓨팅, 입출력 처리