

논문 2010-47CI-1-20

# 서비스 충돌인지 기반 적응형 서비스 구성 방식

(Adaptive Service Configuration based on Service Collision Awareness)

윤 원 식\*

(Wonsik Yoon)

## 요 약

본 논문에서는 서비스 충돌을 회피할 수 있는 효율적인 시스템 구조도를 설명하고, 이를 위한 수정된 서비스 세션 테이블을 제안하여 서비스 충돌회피 절차를 제안하였다. 제안된 방식은 서비스 충돌을 인지하여 이를 회피함으로써 불필요한 서비스 요청메시지를 감소시키며 또한 사용가능한 디바이스에게 직접 서비스를 요청함으로써 서비스 제공 전달시간을 줄일 수 있음을 확인할 수 있다. 따라서 제안된 방식은 퍼스널 서버로 하여금 효율적으로 배터리를 소비하게 하며, 신속한 서비스를 제공받을 수 있게 하는 장점이 있다.

## Abstract

We present the system architecture for service collision avoidance and propose the modified service session table and service collision avoidance scheme. The simulation results show that the proposed scheme reduces the number of unnecessary service request messages and service provisioning delivery time. Therefore Personal Server (PS) uses the battery efficiently and can use the service rapidly.

**Keywords :** 서비스 구성기법(Service Configuration Technique), 서비스 충돌인지(Service Collision Awareness), 역할기반 접근제어(Role-based Access Control), 퍼스널 서버(Personal Server)

## I. 서 론

최근 PDA, 스마트 폰, PMP 등이 대중화됨에 따라 사용자는 VOD/AOD와 같은 멀티미디어 서비스를 비롯하여 일정관리 서비스, 메시지 서비스, 지역뉴스 서비스, 이동 중 여행정보 서비스 등의 다양한 서비스를 이용할 수 있는 퍼스널 서버(personal server)를 휴대할 가능성이 증대하고 있다<sup>[1~2]</sup>. 하지만 휴대단말의 제한된 배터리 용량과 협소한 디스플레이, 낮은 처리능력으로 인해 사용자에게 높은 만족감을 주는 서비스를 제공하기에는 한계가 있을 수 있다. 이를 위해 퍼스널 서버는 인접한 디바이스를 인지하고 이를 이용함으로써 보다 양질의 서비스를 사용자에게 제공할 수 있어야 한다.

이때 반드시 허용된 사용자만이 공간내에 존재하는 인접 디바이스를 사용할 수 있게 하는 적절한 접근제어 방식이 필요하며, 이러한 접근제어 방식은 유비쿼터스 공간내에서 인증된 사용자가 공간내에 공유된 자원을 정해진 권한에 따라 부여해야 한다<sup>[3]</sup>. 또한 접근제어 방식은 사용자의 다양한 요청을 적절하게 반영해야 하며 동적으로 끊임없는 서비스를 제공해야 한다.

본 논문에서는 사용자가 퍼스널 서버를 통해 현재 공간내에 존재하는 공유 디바이스의 자원을 효율적으로 사용하기 위한 서비스 구성기법을 제안한다. 제안한 방식은 공간내의 다수의 사용자가 동일한 디바이스를 사용하고자 할 때 발생하는 어플리케이션 계층에서의 서비스 충돌을 인지하여, 퍼스널 서버가 불필요한 서비스 요청메시지를 발생시키는 것을 방지하게 한다. 이는 퍼스널 서버의 제한된 배터리를 효율적으로 사용하게 하며, 또한 사용자가 어플리케이션 서비스를 제공받는데 걸리는 시간을 줄일 수 있다. 이를 위해 유비쿼터스 공

\* 정회원, 아주대학교 전자공학부  
(Department of Electrical and Computer Eng.,  
Ajou University)  
접수일자: 2009년10월7일, 수정완료일: 2010년1월8일

간에서 사용자에게 보다 적응적으로 서비스를 제공할 수 있도록 서비스를 구성할 수 있는 개선된 서비스 세션 테이블을 제안한다.

서론에 이어 II장에서는 기존의 접근제어 및 서비스 구성기법을 기술하며, III장에서는 제안한 서비스 충돌인지 기반 서비스 구성 기법을 기술하며, 이어서 IV장에는 성능평가를 행하며, 마지막으로 결론을 맺는다.

### II. 기존의 접근제어 및 서비스 구성 기법

#### 1. 역할기반 접근제어

공간내의 사용자 및 서비스, 인접 디바이스 간의 서비스 구성을 위해서는 적절한 접근제어 방식이 요구된다. 이러한 접근제어 방식 중 역할기반 접근제어 (RBAC : Role-based Access Control) 방식이 널리 사용된다. RBAC 방식은 사용자 개개인에게 권한을 부여하기 보다는 사용자의 역할에 따라 인접 디바이스의 사용권한 및 사용 가능한 자원을 할당하는 방식이다. 사용자와 역할간, 역할과 권한간의 관계를 설정을 위한 적절한 정책이 필요하며, 이러한 정책에 따라 사용자는 어플리케이션에 따른 다양한 역할에 매핑될 수 있으며 이러한 역할들은 어플리케이션 서비스에 따라 계층화된 역할 구조를 가지게 된다. 본 논문에서는 그림 1에서 보듯이 system role과 application role의 두 가지 역할을 정의하였다. System role은 사용자의 계정을 생성할 때 정의되며, 전체 시스템 내에서 사용자의 업무나 신분 에 따라 시스템 내의 디바이스들에 대한 접근권한을 부여하는데 사용된다. System role은 administrator를 포함하고 있으며, administrator는 접근제어를 위한 역할 및 자원사용 권한, 서비스 접근모드 등을 결정하는 정책을 관리한다. Application role은 퍼스널 서버를 통해 실행될 수 있는 어플리케이션 서비스에 따라 결정되며, 특정 어플리케이션 속성에 특성화된 접근제어 정책을 따라 system role과 적절한 매핑관계를 가진다.

그림 1은 학교 내부의 계층화된 역할구조의 예를 나타낸 것이다. 학교에서는 교수와 학생, 관리자로 system role이 구분되며, 강의 어플리케이션에 의해 강의자와 수강생으로 application role이 매핑된다. 이러한 계층적인 역할 모델은 application role에 따라 system role을 재사용할 수 있어 어플리케이션 구현을 쉽게 하며, 시스템 확장성을 용이하게 하는 장점이 있다.

여러 사용자가 함께 존재하는 유비쿼터스 공간내에

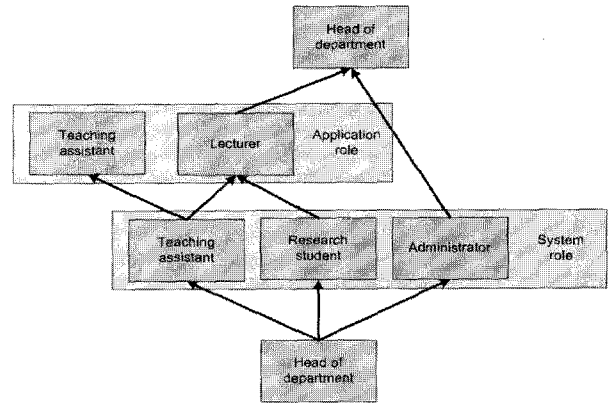


그림 1. 계층화된 역할 모델  
Fig. 1. Hierarchical role model.

서 각 사용자는 인접 디바이스를 적절하게 공유하면서 사용해야 한다. 또한 사용자에게 할당된 권한은 공간내에 존재하는 사용자의 수, 사용자의 권한 및 업무에 따라 제한될 수 밖에 없다. 이에 그룹 혹은 개인별 요구에 따라 서로 협력하며 공유된 디바이스를 효율적으로 사용할 수 있게 하는 서비스 접근모드가 필요하다. 참고 문헌 [4]에서는 individual mode, group mode, supervised usage mode와 같은 3가지의 접근모드를 제안하고 있다. Individual mode는 공간내에 서비스를 받고자 원하는 사용자가 한 명일 경우이다. 이 경우 사용자는 자신에게 허락된 권한내에서 인접 디바이스를 자유롭게 사용할 수 있다. Group mode는 공간내에 서비스를 요구하는 사용자가 다수일 경우이다. 이 경우 사용자는 자신에게 알맞은 system role을 부여받게 되며 각 role에 따라 허용된 권한내에서 공유된 디바이스를 사용하며, 이때 각 사용자는 동일한 권한을 부여받게 되며, 사용자가 증가함에 따라 개별 사용자가 공유할 수 있는 자원은 감소할 수 있다. 또한 supervised usage mode는 group mode에서 사용자에게 동일한 권한을 부여하는 것과는 달리 다수의 사용자가 각각의 우선 순위를 가지고 우선순위에 따라 공유된 자원의 사용권한을 높일 수 있다. 이 모드를 위해서는 system role에서 administrator의 권한을 가진 사용자가 있어야 하며, administrator의 요청이나 정해진 정책에 따라 모드를 변경할 수 있다.

사용자에게 어플리케이션 서비스를 제공하기 위해서는 앞에서 언급한 사용자의 역할 할당 및 접근모드를 기반으로 공간내에 존재하는 사용자 및 디바이스의 접근권한, 접근모드 등을 나타내는 서비스 세션 정보가 필요하다. 각 세션은 공간내에서 사용자의 역할정보를

포함하고 있으며, 현재 제공되는 어플리케이션 서비스에서 정의한 역할에 따른 디바이스의 접근제어 리스트 정보를 가지고 있어야 한다. 실제 접근제어는 서비스 세션정보에 따라 결정되며 테이블을 통해 관리된다.

### 2. 서비스 구성기법

서비스 구성기법은 사용자가 원하는 서비스를 적절히 제공할 수 있는 디바이스를 선택하여 서비스 세션을 구성하는 것이다[5]. 또한 서비스 세션을 위해 선택되는 디바이스는 사용자의 위치나 업무, 서비스가 요청되는 시기에 따라 수시로 변하게 된다. 이러한 서비스 구성 기법은 사용자에게 요청하는 서비스에 대해서 사용자가 만족할 수 있는 품질을 제공할 수 있어야 하며, 끊임없는 서비스 제공을 위해 사용가능한 디바이스를 미리 예약하여 사용자가 서비스 요청시 서비스 제공시간을 줄일 필요도 있다. 그리고 서비스 예약기법은 사용자의 priority나 스케줄 및 이동성 정보 등의 상황 정보를 기반으로 예측형 서비스 예약 구성기법이 가능해야 한다. 이러한 동적인 서비스 구성기법을 통해 사용자는 어디에서나 언제든지 원하는 서비스를 제공받을 있게 함으로써 서비스 가용성을 높일 수 있게 한다.

지능적인 서비스 구성을 위해서 공간내의 디바이스들은 사용자가 원하는 서비스를 적절히 제공할 수 있는지 여부를 판별하기 위한 서비스 컴포넌트를 가져야 한다<sup>[6]</sup>. 즉 하나의 어플리케이션 서비스는 다양한 서비스 컴포넌트들의 조합을 통해 제공될 수 있다. 예를 들어 동영상 보기 위해서는 동영상을 보여줄 수 있는 디스플레이 부분과 음성을 재생할 수 있는 오디오와 같은 서비스 컴포넌트들이 제공되어야 사용자에게 적절한 어플리케이션 서비스를 제공할 수 있는 것이다. 이러한 서비스 컴포넌트는 디바이스가 가지는 다양한 서비스 컴포지션(service composition)을 정의하는 방법에 의존적이며, 참고문헌 [7~9]에서와 같이 다양한 프로젝트들

표 1. 어플리케이션 서비스에 따른 서비스 구성 테이블

Table 1. Service configuration table for application service.

| Application service | Service composition | Configuration access list    |
|---------------------|---------------------|------------------------------|
| Anywhere media      | display, speaker    | DTV, PC, .....               |
| Interactive lecture | display             | project, DTV, monitor, ..... |

을 통해 활발히 연구가 진행되고 있다. 이러한 서비스 컴포넌트는 재사용이 가능하며, 서로 다른 컴포넌트간의 협업을 통해 보다 나은 서비스를 제공할 수도 있다. 이러한 서비스 컴포넌트를 활용하여 퍼스널 서버는 어플리케이션 서비스를 적절하게 제공할 수 있는 후보 디바이스를 찾아 서비스 구성 테이블을 생성하여 관리해야 한다. 그리고 각 서비스 구성리스트는 사용자에게 보다 나은 서비스 질을 제공할 수 있는 우선순위에 따라 구성된다. 표 1은 어플리케이션 서비스에 따른 서비스 구성 테이블 예를 보여준다.

### 3. 기존의 접근제어 및 서비스 구성기법의 문제점

그림 2에서 보는 바와 같이 접근모드가 group mode 일 때 여러 퍼스널 서버들이 자신에게 주어진 권한내에서 공간내의 인접 디바이스를 이용하고자 할 때 임의의 사용자에게 의해 디바이스가 사용되고 있어 주변의 다른 사용자가 공유된 디바이스로 서비스를 받지 못하는 서비스 충돌 현상이 발생할 수 있다. 예를 들어 PS 1은 무선 인터페이스로 UWB를 사용하며, PS2는 802.11기반의 WLAN을 사용한다. 또한 공간내에는 각각의 무선 인터페이스에 대한 AP가 설치되어 있다고 가정하자. 이때 PS 1이 Digital TV를 통해 VOD 서비스를 제공받고 있다. 이 경우 PS 2가 동일한 서비스를 Digital TV에 요청하면 무선링크 계층에서 인지할 수 없는 어플리케이션 계층에서의 서비스 충돌이 발생한다. 이러한 서비스 충돌로 인해 퍼스널 서버는 불필요한 메시지를 발생시켜 비효율적으로 배터리를 소모한다.

또한 서비스 컴포넌트 기반으로 어플리케이션 서비스를 적절히 제공할 수 있는 후보 디바이스를 통해 서비스를 제공받을 때 그림 3에서 보듯이 우선순위가 높은 후보 디바이스에게 먼저 서비스 요청 메시지를 보내

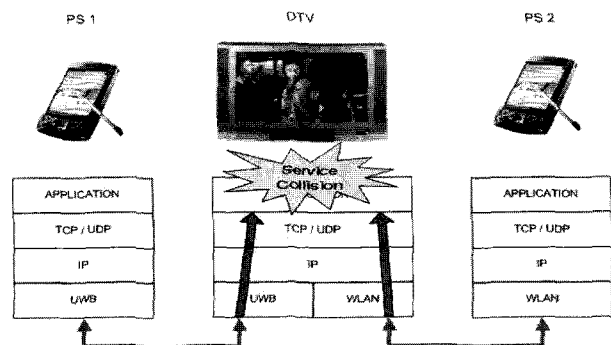


그림 2. Group mode에서의 서비스 충돌  
Fig. 2. Service collision in group mode.

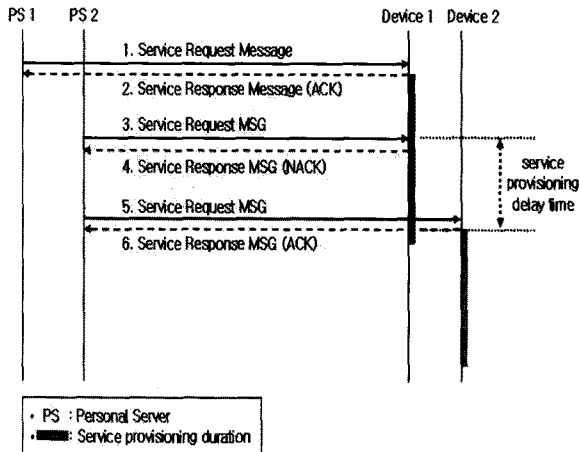


그림 3. Group mode에서의 서비스 제공 지연  
Fig. 3. Service provisioning delay in group mode.

게 된다. 이때 서비스 요청메시지를 받은 디바이스가 사용 중일 때 사용자는 차선책인 디바이스에게 서비스 요청 메시지를 보내게 된다. 이러한 불필요한 서비스 요청 메시지를 주고받는 데 걸리는 시간은 사용자에게 서비스를 제공하는데 지연을 발생시킨다.

### III. 제안하는 서비스 충돌인지 기반 서비스 구성 기법

본 논문에서 제안한 서비스 접근제어 방식 및 서비스 구성기법의 기본 아이디어는 앞장에서 언급한 어플리케이션 계층에서의 서비스 충돌을 회피하기 위해 서비스 세션 테이블을 개선함으로써 서비스 충돌을 인지하는 것이다. 이를 통해 서비스 충돌을 일으키는 불필요한 메시지를 줄임으로써 배터리 낭비를 줄일 뿐만 아니라, 서비스 충돌회피 기법을 통해 서비스 구성 테이블에 있는 후보 디바이스에게 직접 서비스 요청메시지를 보냄으로써 서비스 제공 시간을 줄일 수 있다.

#### 1. 제안하는 서비스 세션 테이블

제안하는 서비스 세션 테이블은 접근모드가 group mode일 경우 발생하는 서비스 충돌현상을 회피하기 위해서는 기존의 서비스 세션 테이블에 새로운 필드를 추가하여 개선할 필요가 있다. 표 2는 서비스 충돌회피를 위한 개선된 서비스 세션 테이블을 나타낸 것이다. 제안한 개선된 서비스 세션 테이블은 현재 사용자가 사용하는 디바이스 정보를 관리하는 'used device' 필드와 사용자가 선택한 디바이스를 사용하는 시간 정보인

표 2. 서비스 충돌회피를 위한 개선된 서비스 세션 테이블

Table 2. Improved service session table for service collision avoidance.

| Access Matrix |     |    |     |     |      |
|---------------|-----|----|-----|-----|------|
| Role          | DTV | PC | PDA | Mon | Proj |
| Father        | ○   | ○  | ○   | ○   | ○    |
| Mother        | ○   | ○  | ○   | ○   | ○    |
| Son           | ○   | ○  | ×   | ○   | ×    |
| Guest         | ×   | ×  | ×   | ○   | ×    |
| Adult         | ○   | ○  | ○   | ○   | ○    |
| Child         | ×   | ○  | ×   | ○   | ×    |
| Super         | ○   | ○  | ○   | ○   | ○    |

| Current Role Translation |          |          |             |        |
|--------------------------|----------|----------|-------------|--------|
| user                     | sys_role | app_role | used device | SPV    |
| u1                       | father   | adult    | DTV         | 10 min |
| u2                       | mother   | adult    | PC          | 20 min |
| u3                       | son      | child    | -           | -      |
| u4                       | guest    | adult    | -           | -      |

| Service Configuration |                   |                            |
|-----------------------|-------------------|----------------------------|
| Application service   | Service component | Configuration access list  |
| Anywhere media        | display, speaker  | DTV, PC, ...               |
| Interactive lecture   | display           | project, DTV, monitor, ... |

'SPV(Service Provisioning Value)' 필드를 추가하였다. 또한 특정 어플리케이션 서비스에 따라 사용자들이 공간내의 인접 디바이스를 함께 공유하는 group mode와 구별하기 위해 개별적으로 사용자들이 디바이스를 사용하는 competition mode를 구분하여 지원한다. 이 모드에서 퍼스널 서버는 다음 절에서 설명할 서비스 충돌회피 기법을 사용하게 된다.

#### 2. 제안하는 서비스 충돌회피 기반 접근제어 방식

앞 절에서 서술한 수정된 서비스 세션 테이블의 정보를 활용하여 퍼스널 서버는 아래와 같은 절차에 따라 제안하는 서비스 충돌 회피기법을 수행한다.

첫 번째로 사용자는 시스템 내에 자신의 계정을 생성하면서 system role을 생성하고 역할에 따른 시스템 내에서의 디바이스에 대한 접근권한에 대한 정보를 access matrix 테이블에 저장한다. 이를 표 2 (a)에 나타내었다.

두 번째로 사용자가 공간내의 인접 디바이스를 이용하여 어플리케이션 서비스를 제공받고자 할 경우, 사용자는 자신의 profile 정보를 기반으로 어플리케이션 서비스에 정의된 application role에 매핑이 되며, current role translation 테이블에 자신의 영역을 할당받게 된다. 이를 표 2 (b)에 나타내었다.

세 번째로 현재 공간내의 상황, 즉 사용자의 수와 실행되는 어플리케이션 서비스, 사용자/어플리케이션의 예약된 스케줄 정보에 따라 관리자나 홈서버는 서비스 접근모드를 결정하며, 이를 표 2 (c)에 나타내었다.

네번째로 사용자가 어플리케이션 서비스를 받고자 할 경우, 퍼스널 서버는 current role translation 테이블의 used device 필드와 service configuration table의 configuration access list를 참조하여 사용가능한 디바이스를 선택하고 서비스 요청 메시지를 보낸다. 선택된 디바이스로부터 ACK 메시지를 받으면 서비스 세션 테이블의 current role translation 테이블에서 used device 필드와 SPV(service provisioning value) 필드를 업데이트 하며, 이를 표 2 (b),(d)에 나타내었다.

마지막으로 사용자의 우선순위 기반의 서비스 재구성 기법을 제공하기 위해서 퍼스널 서버는 인접의 퍼스널 서버들과 공유된 디바이스의 사용권한에 대한 우선순위 확보를 위한 협상절차를 거친다. 이러한 협상은 사용자의 우선순위와 어플리케이션 서비스의 우선순위에 따라 결정된다.

예를 들어 보다 높은 우선순위를 가진 사용자가 낮은 우선순위를 가진 사용자에 의해 사용되고 있는 디바이스의 사용권한을 얻고자 한다면 그림 4에서 보듯이 우선순위 협상절차에 따라 서비스 세션을 변화시킬 수 있다. 또는 사용자가 요청하는 어플리케이션 서비스가 긴급한 업무일 경우 현재 디바이스를 사용하고 있는 사용자는 일시적으로 디바이스 사용권한을 양보하고 자신은 네 번째 절차에 따라 다른 후보 디바이스를 통해 서비스를 제공받는다.

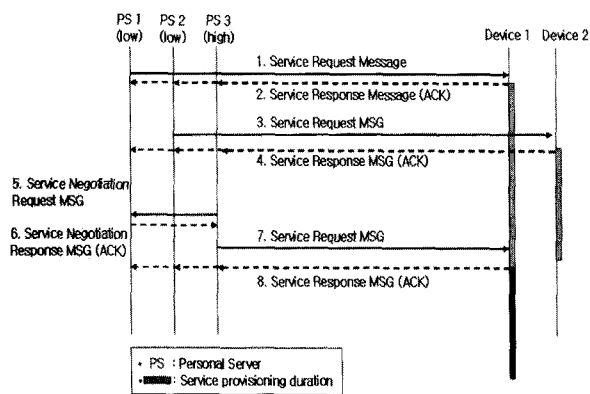


그림 4. 우선순위 기반 서비스 협상절차 메시지 흐름도  
Fig. 4. Service negotiation message flow based on the priority of user and application service.

IV. 성능 평가

성능 평가를 위해서 다음과 같은 시뮬레이션 환경을 가정하였다. 각 사용자는 Poisson random access를 통

해 1분 단위로 서비스 요청 메시지를 발생시킨다. 서비스 요청메시지에 대한 ACK 응답을 받은 퍼스널 서버는 동일한 서비스 제공시간을 세팅하며, 서비스 요청메시지를 발생시켰을 때 인접한 디바이스가 다른 퍼스널 서버에 의해 사용되고 있으면 NACK 메시지를 받은 것으로 간주한다. 또한 서비스 요청메시지에 NACK 메시지를 받았을 경우 퍼스널 서버는 정해진 우선순위에 따라 정해진 사용가능한 디바이스에 서비스 요청 메시지를 전달한다.

1. 충돌된 서비스 요청메시지

그림 2에서 보듯이 퍼스널 서버가 서비스 충돌회피를 하지 않았을 경우 불필요한 서비스 요청 메시지가 발생한다. 본 논문에서 제안한 서비스 충돌회피 기법을 사용하여 줄일 수 있는 서비스 요청 메시지 수  $N_{saved}$ 는 식 (1)과 같이 표현될 수 있다.

$$N_{saved} = N_{REQ\_wo} - N_{REQ\_w} \tag{1}$$

여기서  $N_{REQ\_wo}$ ,  $N_{REQ\_w}$ 는 각각 정해진 시뮬레이션 기간동안에 서비스 충돌회피 기법을 사용하지 않을 경우와 제안한 서비스 충돌회피 기법을 사용하였을 경우 공간내의 사용자에 의해 발생한 서비스 요청메시지 수의 총합이다. 그림 5는 공간내의 사용자 수에 따른  $N_{saved}$ 를 나타낸 것이다. 여기서 d는 공간내에 사용가능한 디바이스의 수이고,  $\lambda$ 는 한 명의 사용자가 서비스 요청메시지를 발생시키는 빈도를 의미한다. 그림 5에서 보는 바와 같이 제안한 서비스 충돌회피 기법을 사용하

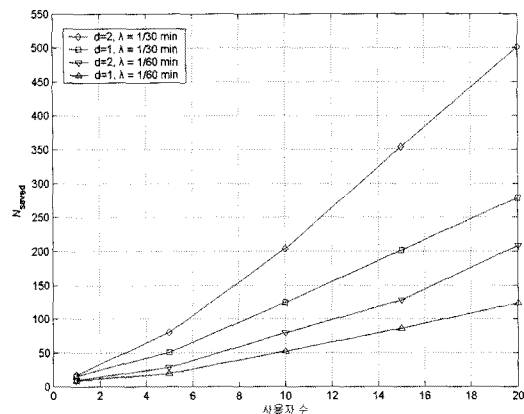


그림 5. 서비스 충돌회피를 통해 줄일 수 있는 서비스 요청 메시지 수  
Fig. 5. Reduced service request message by service collision avoidance.

였을 때 줄일 수 있는 서비스 요청메시지는 공간내의 사용자가 증가할수록 많음을 볼 수 있으며, 또한 사용자당 서비스 요청빈도에 따라 급격히 증가함을 볼 수 있다. 제안한 서비스 충돌회피 기반의 접근제어 방식은 이와 같은 불필요한 서비스 메시지를 줄임으로써 퍼스널 서버가 불필요한 서비스 요청메시지를 무선링크에 전송하지 않음으로써 배터리를 보다 효율적으로 사용할 수 있고 무선 채널 용량을 효율적으로 사용할 수 있다.

### 2. 서비스 제공 전달시간

그림 3에서 보듯이 서비스 충돌을 회피하지 않았을 경우 service configuration table에 정의된 access list 순서에 따라 퍼스널 서버가 인접 디바이스에게 서비스를 요청하게 되어 서비스가 제공되는 시간에 지연이 발생하게 된다. 그림 6은 공간내의 사용자 수에 따른 요청한 어플리케이션 서비스가 인접 디바이스를 통해 서비스가 제공되는 전달시간(tdeliver)을 보여주고 있다. tdeliver은 서비스 요청메시지 및 서비스 응답메시지가 전달되는 시간과 요청메시지에 대한 처리시간의 총합을 의미하며, 서비스 회피기법을 사용했을 경우와 사용하지 않을 경우 서비스 제공 전달시간은 각각 tdeliver\_w, tdeliver\_wo 으로 표현한다.

이때 tdeliver\_w는 사용가능한 후보 디바이스에게 직접 메시지를 전달하므로 식 (2)와 같으며, tdeliver\_wo는 식 (3)과 같이 서비스 충돌이 발생한 횟수 Ncollision에 비례하여 서비스 제공전달 시간이 결정된다.

$$tdeliver_w = tREQ + tRES + PTMSG \tag{2}$$

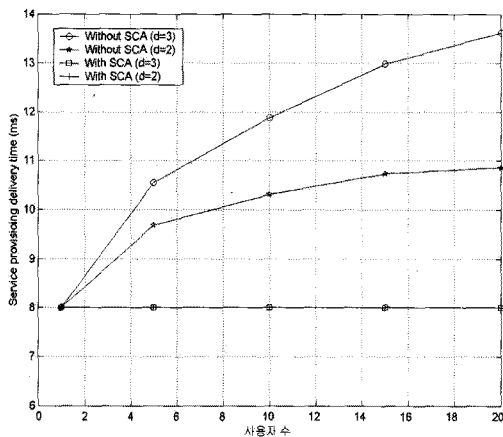


그림 6. 서비스 제공 평균 전달시간  
Fig. 6. Service provisioning average delivery time.

$$tdeliver_wo = (tREQ + tRES + PTMSG) \cdot (Ncollision + 1) \tag{3}$$

여기서 tREQ, tRES는 각각 퍼스널 서버와 인접 디바이스간 무선링크의 메시지 지연시간을 의미한다. PTMSG는 인접 디바이스가 수신한 서비스 요청메시지 처리하는 시간을 의미한다.

그림 6에서 보듯이 서비스 충돌회피 기법을 사용했을 경우에는 서비스 제공 전달 시간이 다른 사용자에 의해 사용되지 않는 인접 디바이스에게 직접 서비스를 요청하기 때문에 공간 내에 사용가능한 디바이스 수에 영향을 받지 않고 일정한 값을 갖는다. 하지만 서비스 충돌회피 기법을 사용하지 않았을 경우 공간내의 사용자가 증가함에 따라 어플리케이션 서비스가 제공되는 전달시간이 지연됨을 볼 수 있다. 또한 사용가능한 디바이스의 수가 늘어남에 따라 서비스 제공 전달시간이 증가하는 것을 알 수 있다.

### V. 결 론

본 논문에서는 RBAC(Role-based Access Control) 방식을 이용한 서비스 충돌인지 기반 서비스 접근제어 방식을 제안하였다. 유비쿼터스 공간내에서 퍼스널 서버에 있는 어플리케이션 서비스를 인접한 디바이스를 사용하여 제공받고자 할 때, 효율적인 접근제어 방식이 필요하다. 사용자가 많아짐에 따라 사용자 간의 서비스 충돌로 인해 발생하는 불필요한 서비스 요청메시지 및 서비스 제공 전달시간 지연을 감소시키는 것은 매우 중요하다. 제안된 서비스 충돌회피 기반 서비스 접근제어 방식은 현재 공간 내에 사용가능한 디바이스의 사용유무를 서비스 세션 테이블을 활용하여 인지함으로써 어플리케이션 계층에서 발생할 수 있는 서비스 충돌을 해결하였다. 또한 무선 환경의 링크 계층과의 상호 정보 교환 기법을 활용한다면 보다 나은 링크 계층에서 성능 향상을 기대할 수 있을 것이다. 성능분석 결과를 통해 제안한 방식을 적용시켰을 때 충돌된 서비스 요청메시지 및 서비스 제공 전달시간이 감소됨을 알 수 있다. 따라서 제안된 방식을 통해 퍼스널 서버는 배터리를 효율적으로 사용할 수 있으며, 사용자는 신속한 서비스를 제공받을 수 있는 장점이 있으며 퍼스널 서버가 점차적으로 보편화되는 유비쿼터스 공간에서 매우 효율적인 방식으로 활용될 수 있다.

참 고 문 헌

- [1] M. M. Lankhorst, H. van Kranenburg, A. Salden and A.J.H. Peddemors, "Enable Technology for Personalizing Mobile Services," Proc. System Sciences, 2002. HICSS, pp.1464-1471, Jan. 2002.
- [2] R. Want et al., "The Personal Server: Changing the Way We Think about Ubiquitous Computing," Proc. 4th Int'l Conf. Ubiquitous Computing, IEEE CS Press, pp.194-209, 2002.
- [3] M. Sloman and E. Lupu, "Security and Management Policy Specification," IEEE Network, Vol.16, pp.10-19, Mar. 2002.
- [4] G. Sampemane, P. Naldurg, and R. H. Campbell, "Access Control for Active Space," Proc. ACSAC'02, pp.343-352, Dec. 2002.
- [5] Dongyan Xu, Duangdao W. and Nahrstedt. K., "Multimedia Service Configuration and Reservation in Heterogeneous Environments," Proc. Distributed Computing Systems, pp.512-519, April 2000.
- [6] M. Roman, B. Ziebart and R. H. Campbell, "Dynamic Application Composition : Customizing the Behavior on an Active Space," Proc. PerCom 2003, pp.169-176, Mar. 2003.
- [7] M. Roman, R. H. Campbell and K. Nahrstedt, "Gaia: A Middleware Infrastructure to Enable Active Spaces," IEEE Pervasive, Vol1(4), pp.74-82, 2002.
- [8] B. Brumitt, B. Meyers, J. Krumm, A. Kern and S. Shafer, "EasyLiving: Technologies for Intelligent Environments," Proc. of Handheld and Ubiquitous Computing, pp.12-29, 2000.
- [9] J. P. Sousa and D. Garlan, "Aura: an Architectural Framework for User Mobility in Ubiquitous Computing Environments," Proc. of IEEE/IFIP Conf. on Software Architecture, pp.29-43, 2002.

저 자 소 개



윤 원 식(정회원)

1984년 서울대학교 제어계측 공학과 학사 졸업.

1986년 한국과학기술원 전기 및 전자공학과 석사 졸업.

1991년 한국과학기술원 전기 및 전자공학과 박사 졸업.

1991년 금성전기 선임연구원

1994년 금성정밀 책임연구원

1995년 University of Victoria 방문교수

2001년 콘텔라 CTO

2007년 University of Florida 방문교수

1994년~현재 아주대학교 전자공학부 교수

<주관심분야 : 이동통신, 무선네트워크>