SIP기반의 VoIP 시스템에서의 Caller Preference 구현

조현규° 고세령 장춘서
금오공과대학교 컴퓨터공학과
{blackjo° zzaru}@cespc1.kumoh.ac.kr, csjang@kumoh.ac.kr

Implementation of Caller Preference in SIP-based VoIP System

Hyun Gyu Jo° Se Lyung Ko Choon Seo Jang
Dept. of Computer Engineering, Kumoh National Institute of Technology

요 약

SIP(Session Initiation Protocol)는 사용자간의 멀티미디어 세션을 처리하기 위한 응용 계층의 시그널링 프로토콜로서 유연성 및 확장이 용이한 장점을 가지고 있다. Caller Preference는 이러한 SIP의 기본적인 프로토콜을 확장한 형태로서 송신자가 Preference를 명시하여 서버가 처리할 응답 기능을 선택하거나 수신자의 수신 능력(Callee Capabilities)에 따라 적절한 호처리를 진행할 수 있는 서비스이다. 본 논문에서는 SIP를 기반으로 하는 VoIP(Voice over IP) 시스템을 구현함에 있어 UA(User Agent)내에 Preference를 선택적으로 명시할 수 있는 기능을 포함시키고 또한 이의 요청에 대한 수용이 가능하고 호처리를 진행할 수 있는 네트워크 서버를 구현하였다.

1. 서 론

SIP(Session Initiation Protocol)는 하나 이상의 참가자들 사이에 세션을 생성, 수정, 종료하기 위한 응용 계층의 시그널링 프로토콜이다[1]. SIP는 호처리를 위해 텍스트 형태의 메시지를 사용하므로 쉽게 확장이 가능하고 이를 통해 다양한 응용 서비스로 적용이 가능한 장점을 가지고 있다.

SIP 프로토콜을 확장한 서비스 가운데 Caller Preference는 호설정을 위한 요청시 송신자가 선호하는 형태에 따라 서버가 응답을 처리하게 하고 또한 수신자의 수신 능력에 따라 호 설정을 가능하도록 하는 서비스이다. 이러한 Caller Preference를 이용하면 사용자는 휴대폰, PDA, 사무실 전화 또는 집 전화와 같은 모든 장치 번호들을 SIP URI와 같은 하나의 주소로 통합하고 이와 연결을 시도하는 다른 사용자는 각각의 장치 번호를 모두 기억할 필요없이 원하는 장치의 선택과 하나의 SIP URI를 이용하여 연결을 할 수 있다[2].

본 논문에서는 이러한 Preference의 기능을 포함하는 UA(User Agent)와 함께 이의 요청을 수용하고 처리할 수 있는 네트워크 서버를 구현하였다.

본 논문의 구성은 다음과 같다. 2장에서는 관련연구로서 SIP 기반에서의 Preference 서비스에 관하여 설명하고 3장에서는 구현된 시스템의 동작에 대한 전반적인 내용에 관하여 다루며 4장에서 결론을 맺는다.

2. Caller Preference

Caller Preference는 기본적인 SIP 프로토콜을 확장한 서비스로서 사용자가 수신자에게 호설정을 시도할 시에 자신이 선호하는 형태를 포함하여 요청 메시지를 보낼 수 있는 기능이다. 사용자는 요청 메시지내에 Preference에 관련된 새로운 헤더 필드를 선택적으로 포함할 수 있으며 이는 두개의 카테고리로 나뉘어진다.

첫째, Request Handling Preference이며 이는 Request-Disposition 헤더 필드를 통해 구현되어 URI의 선택, 프록시 또는 리다이렉트 처리 여부, forking의 여부, 재귀적 검색의 여부, 병렬 또는 순차적인 검색의 여부 등과 같은 서버에게 요구하는 특정 동작들을 기술한다. 두번째는 Feature Preference이고 이는 Accept-Contact 및 Reject-Contact 헤더 필드에 송신자의 통신 능력들을 표시하기 위해 사용되는 feature 파라미터들로 표현된다[2,3].

프록시(Proxy) 서버는 받은 요청 메시지에 포함된 Request Handling Preference에 따라 송신자가 지시한 형태로 서버의 역할을 수행한다. Feature Preference가 명시된 경우는 그 feature 파라미터와 레지스트라(Registrar)에 등록된 해당 Req-URI의 수신 능력을 나타내는 Contact feature 파라미터와의 매치 여부에 따라 수신자와의 연결을 진행한다. 매치 여부의 처리는 먼저 feature 파라미터들을 feature set predicate들로 변경하게 되는데 그림 1은 수신자의 Contact feature 파라미터를 변경한 예이다.

다음으로 계산된 score 값을 이용하여 Caller Preference, Qa를 결정한다. Qa 값은 각 Accept-Contact predicate에 대해 얻은 score 값의 산술 평균치로 구해지며 이 Qa 는 프록시 서버가 수신자와의 연결을 시도할 때 순서를 정하는데 사용된다. 프록시 서버는 우선 Contact 헤더의 q 값에 의해 순서를 정렬하고 이 중 같은 q 값을 가지는 Contact 주소들 내에서는 Qa에 의해 정렬된다. 식 1에 Qa의 계산식을 보였다.

만약 요청 메시지에 Preference 헤더 필드를 포함하고

있지 않으면 프록시 서버는 묵시적으로 처리한다.

Feature 파라메터를 포함하는 Contact 헤더

```
Contact: <sip:johg@sip2.kumoh.ac.kr >:audio:video:mobility="fixed";
        +message="TRUE":other-param=66372;
    methods="INVITE,OPTIONS,BYE,CANCEL,ACK":schemes="sip,http";
        uri-user="< johg>":uri-domain="sip2.kumoh.ac.kr"
```

Feature set predicate로 변환한 형태

```
(& (audio=TRUE)
   (video=TRUE)
   (mobility=fixed)
   (message=TRUE)
   (| (methods=INVITE) (methods=OPTIONS) (methods=BYE)
      (methods=CANCEL) (methods=ACK))
   (| (schemes=sip) (schemes=http))
   (uri-user=" johg")
   (uri-domain=" sip2.kumoh.ac.kr"))
```

그림 1. Contact predicate로의 변경

$$Qa = \begin{cases} 0, & \text{if } M=0 \\ \dfrac{\sum_{i=1}^{M} Si}{M} & \text{if } M>0 \end{cases}$$

식 1. Qa의 계산식

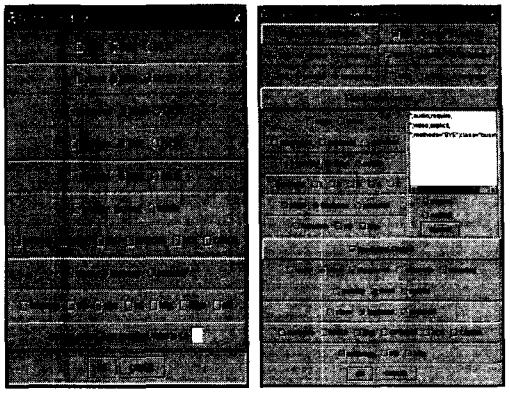
여기서 Si는 각 Accept-Contact predicate에 대해 얻은 score이다. M은 해당 Contact predicate와 매치되는 Accept-Contact predicate의 수를 의미한다.

3. 시스템구현

시스템의 구성은 SIP 기반의 프록시 서버에 Preference 정보를 수용하고 이의 처리를 가능하게 하여 linux OS상에서 운영하였다. UA는 자바를 사용하여 GUI 형태로 작성하였으며 PC 윈도우즈 상에서 구현하였다.

구현된 시스템의 동작은 다음과 같다. 먼저 UA가 수신자로서의 수신 능력들을 포함하여 레지스트라에 등록하는 경우에는 그림 2 (a)와 같이 독립된 사용자 인터페이스 화면을 띄우고 feature predicate를 생성하여 이를 내부적으로 변환한 feature 파라미터를 REGISTER 요청 메시지의 Contact 헤더 필드에 추가한다. 요청을 받은 레지스트라는 이를 저장 및 관리하며 만약 feature 파라미터에 q 값이 생략된 경우에는 내부적으로 "1.0"으로 세팅한다.

UA의 INVITE 요청시 Preference 또한 선택적으로 사용할 수 있도록 하였으며 그림 2 (b)와 같이 독립적인 사용자 인터페이스를 통하여 Request-Disposition, Accept-Contact, Reject-Contact 헤더 필드의 사용 여부 및 원하는 파라미터를 선택하고 feature predicate를 생성한다. 이는 다시 내부적으로 feature 파라미터로 변환 후 요청 메시지의 헤더 필드로 포함시켜 송신을 한다.



(a) (b)

그림 2. feature 파라미터 생성을 위한 사용자 인터페이스

그림 3과 그림 4는 이를 통하여 생성된 REGISTER 및 INVITE 요청 메시지의 실제 예이다.

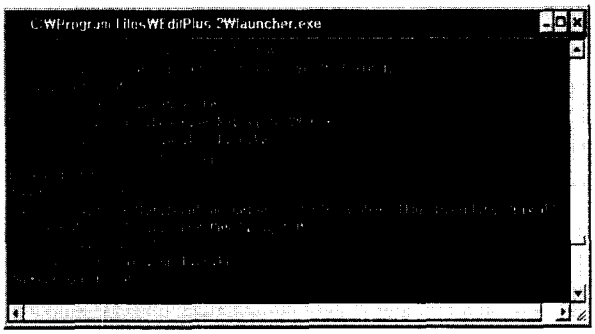


그림 3. REGISTER 요청 메시지

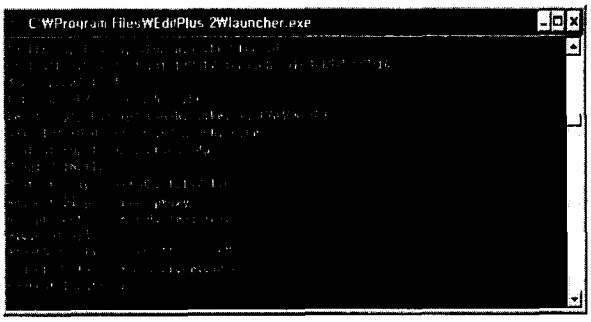


그림 4. INVITE 요청메시지

UA로부터 INVITE 요청 메시지를 받은 프록시 서버는 메시지를 파싱하여 Preference에 관한 헤더 필드를 검사하여 Request-Disposition 헤더 필드를 포함하고 있으면 송신자가 지시한 형태로 서버의 역활을 처리한다. 만약 메시지내에 Accept-Contact 헤더 필드와 Reject-Contact 헤더 필드가 존재시는 그림 5와 같은 흐름에 따라 처리하고 결과에 의해 수신자와의 연결을 시도한다.

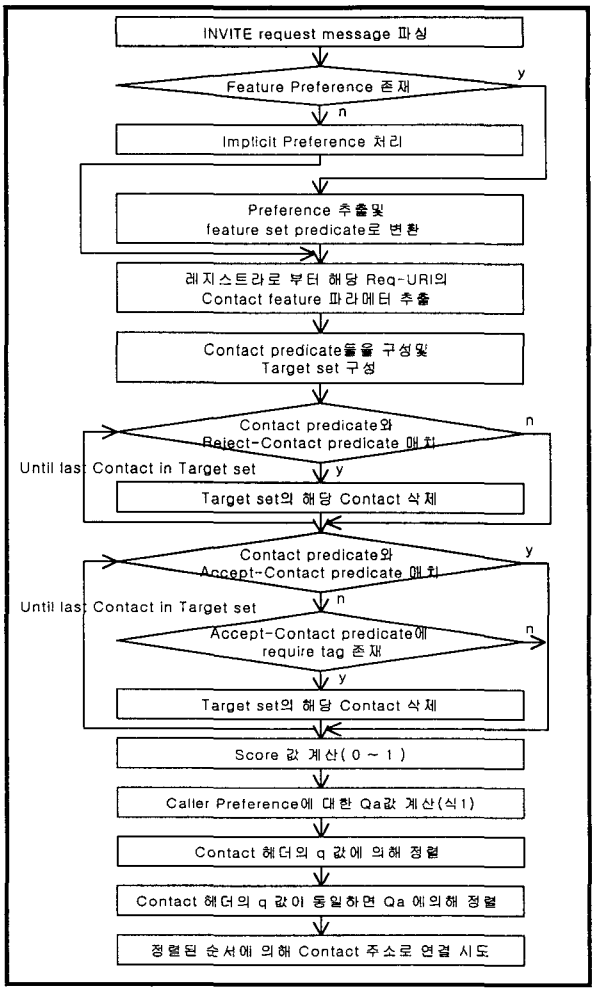


그림 5. 프록시 서버의 Preference 처리

그림 5에서 score 값의 계산은 각 Accept-Contact predicate내의 항목과 Target set에 남아 있는 각 Contact predicate내의 매치되는 항목의 수를 이용하여 먼저 계산하고 그림 6과 같이 require 및 explicit 태그의 여부에 따라 최종적으로 처리된다.

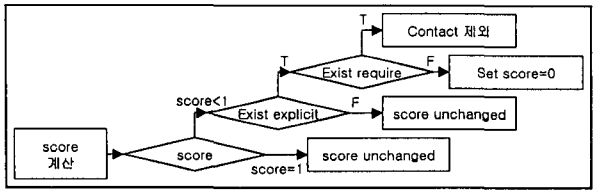


그림 6. score 적용

그림 7은 사용자 "johg"로 부터 그림 4와 같은 Preference를 포함한 요청 메시지를 받은 프록시 서버가 이를 처리하고 최종적으로 연결을 시도하는 화면이다. 화면의 내용은 서버가 처리하는 과정에서 최종적으로 Target set에 남은 3개의 Contact 가운데 Contact 헤더의 q 값에 의해 정렬하고 동일한 "0.2"의 q 값을 가지는 두개는 계산된 Qa에 의해 정렬하는 부분이다. 여기서 사용자 "kmh"의 Contact는 그림 8과 같은 5가지 형태의 수신 능력을 포함하여 레지스트라에 저장되어 있다.

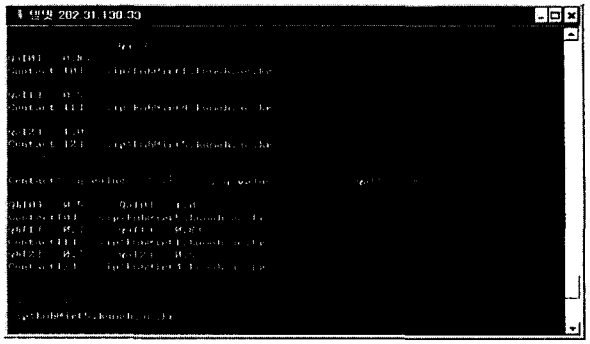


그림 7. 서버의 최종 Preference 처리 화면

	"kmh"의 feature parameter를 포함한 Contact 내용
1	sip:kmh@irt1.kumoh.ac.kr;audio="TRUE";video="TRUE" ;methods="INVITE,BYE";q=0.2 ;uri-user="<kmh>" ;uri-domain="irt1.kumoh.ac.kr"
2	sip:kmh@irt2.kumoh.ac.kr;audio="FALSE" ;msgserver="TRUE" ;methods="INVITE";q=0.2 ; uri-user="<kmh>" ;uri-domain="irt2.kumoh.ac.kr"
3	sip:kmh@irt3.kumoh.ac.kr;audio="TRUE";video="TRUE" ;msgserver="TRUE" ;methods="INVITE";q=0.3 ;uri-user="<kmh>" ;uri-domain="irt3.kumoh.ac.kr"
4	sip:kmh@irt4.kumoh.ac.kr;audio="TRUE" ;methods="INVITE,OPTIONS";q=0.2 ;uri-user="<kmh>" ;uri-domain="irt4.kumoh.ac.kr"
5	sip:kmh@irt5.kumoh.ac.kr;q=0.5

그림 8. 사용자 "kmh"의 Contact 정보

4. 결론

본 논문에서는 SIP를 기반으로 하는 VoIP 시스템을 구현함에 있어 Caller Preference 기능을 추가하였다. 이를 위해 UA에는 요청 메시지내에 Request-Disposition, Accept-Contact, Reject-Contact 헤더 필드를 선택적으로 삽입할 수 있도록 하였다. 이때, Feature 파라미터들은 구현된 GUI 사용자 인터페이스를 이용하여 쉽게 생성할 수 있도록 하였다. 프록시 서버는 Preference를 포함한 UA의 요청 메시지를 수용할 수 있도록 기능을 추가하여 구현하였다. 또한 사용자는 자신의 Contact 주소를 레지스트라에 등록시 Contact 헤더 필드에 수신 능력을 나타내는 feature 파라미터를 추가할 수 있도록 구현하였다.

5. 참고문헌

[1] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, E. Schooler "Session Initiation Protocol", RFC 3261, June 2002.

[2] J. Rosenberg, H. Schulzrinne, P. Kyzivat, "Caller Preferences and Callee Capabilities for the Session", draft-ietf-sip-callerprefs-08, September, 2003.

[3] J. Rosenberg, H. Schulzrinne, P. Kyzivat, "Caller Preferences for the Session Initiation Protocol", draft-ietf-sip-callerprefs-09, December 29, 2003.