

분산형 다자간 다중연결 신호 프로토콜

準會員 姜 鍾 國*, 正會員 金 永 翰**, 金 炳 基**

Distributed Multiparty Multiconnection Signalling Protocol

Jong-Kook Kang* Associate Member, Young-Han Kim**, Byung-gi Kim* Regular Members

※본 논문은 1995년도 한국통신 교환기술연구소 지원과제의 연구결과임.

요 약

멀티미디어 통신에서 호에 참여하는 단말(endpoint)과 교환기는 설정되는 대역폭과 베어러 수용 능력에 따라 동일 호에 다양한 특성의 연결을 설정할 수 있어야 한다. 또한 다자간 통신에서 호 참여자가 증가하는 경우, 호 설정에 소요되는 설정지연이 크게 증대되어서는 안된다. 본 논문에서는 다양한 연결 형태의 다자간 멀티미디어 통신을 지원하기 위한 신호 프로토콜로서 DMMSP(distributed multiparty multiconnection signaling protocol)를 제안한다. DMMSP에서는 기존 프로토콜의 신호능력의 한계를 극복하고 다양한 연결 형태를 지원하기 위한 신호 능력을 제공하여 각 단말과 교환기는 베어러 수용 능력에 따라 독립된 연결을 설정한다. 또한 호 처리(call processing)절차를 분리하여 분산 수행함으로써 설정지연을 줄이고 설정 절차를 최소화한다. 기존의 신호 프로토콜과 DMMSP의 호 제어 절차를 비교 정리하고 각 제어 방법에 따른 신호 절차를 제시하였다. 신호 절차는 다자간 멀티미디어 회의 호를 예로 들어 기존 프로토콜과 DMMSP를 이용한 절차로 구분하였고, 각각의 절차에 대해 정량적으로 비교 분석하였다.

ABSTRACT

The endpoints and exchanges involved in a call can establish heterogeneous connection each other according to required bandwidth and bearer availability in multimedia communication. And as participating users are increased, call setup delay must not be exceedingly increased.

In this paper, we propose DMMSP(distributed multiparty multiconnection signaling protocol) which can support heterogeneous connections and multimedia communications. DMMSP gets over a limitation of existing B-ISDN

*승실대학교 전자계산학과

**승실대학교 정보통신공학과

論文番號:96317-1005

接受日字:1996年 10月 5日

protocol and provides signaling capabilities to support various kinds of connections, and each endpoint can setup individual connection with bearer availability. Moreover since DMMSIP separately performs call processing in distributed scheme, call setup delay can be minimized. We present call control procedures which can be applied to existing B-ISDN protocol and DMMSIP respectively. We take multiparty multimedia conference call as an example that is applied to the existing B-ISDN protocol and DMMSIP, and compare and quantitatively analyze each procedures.

I. 서 론

텍스트와 그래픽 데이터 및 동영상과 음성 데이터를 포함하는 멀티미디어 서비스는 광대역 네트워크의 대표적인 서비스라고 할 수 있다. 이러한 멀티미디어 서비스를 지원하는 호에 참여하는 단말과 교환기는 설정되는 대역폭과 베어러 수용 능력에 따라 동일 호에서 다양한 특성의 연결 설정을 요구할 수 있다. 또한 다자간 통신에서 호 참여자가 증가하는 경우 호 설정에 소요되는 설정지연은 중요한 제한 사항으로 간주된다. 그러나 기존의 B-ISDN(broadband-integrated service digital network) 신호 프로토콜은 순차적인 호 설정 절차를 사용함으로써 다자간 호 설정에 있어서 많은 참여자를 포함하는 경우 설정 지연이 참여자의 수에 비례하여 커질 수 있다. 또한 연결 형태를 단순히 정의⁽¹⁾함으로써 베어러 수용능력에 따른 차별화된 연결 형태를 지원하지 못한다. I.31x에서 정의한 연결 형태는 호에 참여하는 각 단말이 동일 대역폭의 트래픽 특성을 가진다. 그러나 멀티미디어 통신에서는 호에 참여하는 각 단말의 대역폭 수용능력이 서로 다를 수 있다. 따라서 신호 프로토콜은 동일 호에 이질적인 베어러 특성의 단말을 포함할 수 있는 신호 능력을 지원하여야 한다.

지금까지 다자간 호를 지원하기 위해 제안된 신호 프로토콜 모델로는 호와 베어러의 분리 제어 방식을 도입한 순차적인 호 처리 방법⁽²⁾과 DCPA(distributed call processing architecture)⁽³⁾, EXPANSE⁽⁴⁾, 그리고 RACE/MAGIC(research & development in advanced communication technologies in Europe/multi-service applications governing integrated control)에 기초한 프로토콜⁽⁵⁾ 등이 있었다.

호와 베어러의 분리 제어 방식을 도입한 순차적인 호 처리 방법은 기존의 B-ISDN 신호 프로토콜에서 정의한 호와 베어러의 동시설정에 따르는 문제점을

개선하였다. 즉 착신측 호 수락이 불가능 경우라도 호 설정시 대역폭을 점유하지 않으므로 망의 대역폭 이용률을 크게 할 수 있고 호가 활성화된 상태에서도 서비스 파라미터의 협상이 가능하다는 장점을 가질 수 있다. 그러나 호와 베어러의 설정 절차가 순차적인 방법을 따르므로 설정지연이 커지게 되는 문제점이 생기게 된다. 특히 여러 참여자를 포함하는 다자간 회의 호와 같은 경우 설정지연은 매우 심각하게 될 수 있다. 또한 단순한 연결 형태만을 지원하므로 동일 호에 다양한 베어러 특성을 가지는 단말을 수용하기 어려운 단점이 있다.

DCPA는 AT&T Bell 연구소에서 개발된 신호 프로토콜로서 호 처리는 클라이언트-서버 개념을 이용하여 호 제어, 연결 제어, 채널 제어로 분리되어 수행된다. 이와 같이 분리된 제어 기능은 호 서버, 연결 서버, 서비스 서버, 그리고 채널 서버 등에서 각각 분산 수행됨으로써 다양한 서비스를 용이하게 수용할 수 있는 장점을 가진다. 그러나 서비스 제어 및 연결 제어 등이 호 서버를 통해 이루어짐으로써 호 처리 절차가 집중되는 단점을 가진다.

EXPANSE는 Bellcore 연구소에서 개발된 신호 프로토콜로서 개체 지향적인 호 모델과 새로운 프로토콜 구분, 그리고 메시지 흐름 모델 등으로 구성된 서비스에 독립적인 통합시스템이다. 차별화된 계층구조를 가지는 기본 호 개체는 서비스 정의를 효율적이고 유연하게 하고 복잡한 서비스의 세부사항을 관리하는 부담을 줄여 신호 절차를 최소화한다. 즉 EXPANSE에서는 TRM(transport resource management)이라는 분리된 연결 제어 부를 포함하는데 이를 통해 연결 관리 기능을 수행하며 여러 개의 교환기를 제어한다. 따라서 다소 집중화된 구조를 가지며 단말은 보다 추상화된 계층과 인터페이스 됨으로써 연결을 직접적으로 액세스하지 못한다는 단점을 가지고 있다.

RACE/MAGIC에 기초한 신호 프로토콜은 이탈리아

아 CORITEL 연구소에서 개발된 프로토콜로서 지능망 개념과 구조를 B-ISDN에 통합하고 기존의 호와 베어러의 분리 제어를 자원 제어라는 새로운 제어기능을 추가 정의하여 분리시킴으로써 복잡한 멀티미디어 서비스를 쉽게 제공할 수 있다는 장점을 가진다. 이 구조에서는 복잡한 서비스 로직과 특정자원의 제어를 SCN(service control node)에 집중화하고 SREX(special resource exchange)라는 전용 교환기를 통해 물리적인 제어를 수행하도록 한다. 그러나 실제 네트워크에서 SREX는 고정적으로 정해 질 수 없으며 동적으로 지정되어야 한다. 즉 교환기와 경로 설정 문제에 따른 SREX의 위치는 가변적이므로 네트워크를 구성하는 모든 교환기는 SREX로서의 기능을 수행할 수 있어야 한다. 또한 집중화된 노드가 여러 기능을 통합하여 운영하므로 제어 절차가 집중된다는 단점을 가진다.

이상에서 살펴본 신호 프로토콜 모델들은 Q.2931⁽⁶⁾과 Q.2764⁽⁷⁾ 등 ITU-T(international telecommunication union-telecommunication) SG11(study group 11)에서 B-ISDN 신호 프로토콜로서 권고한 프로토콜(이하 기존 프로토콜이라고 부름)과 마찬가지로 동일 호에 베어러 수용능력이 서로 다른 단말을 포함하는 신호능력을 제공하지 못한다. 본 논문에서는 특정자원의 제어와 연결 특성의 중재기능을 수행하는 서버를 각각 정의하여 제어 절차를 분리 수행함으로써 제어 절차가 특정 서버로 집중되는 문제점을 해소하고, 동일 호에 다양한 베어러 수용능력을 가지는 단말을 포함할 수 있는 신호 프로토콜 DMMSPP(distributed multi-party multiconnection signaling protocol)를 제안하였다. 즉 시스템 구조와 프로토콜 기능 모델을 정의하고 각 서버 및 교환기간의 제어정보 교환을 위한 트랜잭션 절차와 각 단말과 교환기 또는 교환기와 교환기사이에서 요구되는 메시지를 정의하였다. 또한 다자간 멀티미디어 회의형 서비스를 예로 들어 DMMSPP 및 기존 B-ISDN 신호 프로토콜을 이용한 호 설정 절차를 비교 분석한다.

서론에 이어 II장에서는 DMMSPP의 구조를 정의하고 III장에서는 DMMSPP의 프로토콜 기능 모델을 정리하였다. 이어서 IV장에서는 DMMSPP에서 사용되는 메시지 및 프리미티브에 대해서 살펴보고 V장에서는 다자간 멀티미디어 회의형 서비스를 예로 들어 기

존 프로토콜과 DMMSPP를 적용한 경우에 있어서의 호 절차를 정리하고 각각의 절차를 정량적으로 비교 분석하였으며 끝으로 VI장에서 결론을 맺었다.

II. DMMSPP의 구조

DMMSPP는 클라이언트-서버(client-server) 개념의 제어 메커니즘에 기초하여 호 처리(call processing)를 호 제어(call control), 연결 제어(connection control), 다양한 연결의 중재(connection coordinator), 특정자원의 제어(special resource control) 등의 세부 기능으로 구분한다. 이들 세분화된 각 제어 기능은 특정 서버에 의해 분산되어 수행된다. 표 1에서는 분리된 호 제어 기능과 이에 대응하는 서버를 정리하였다.

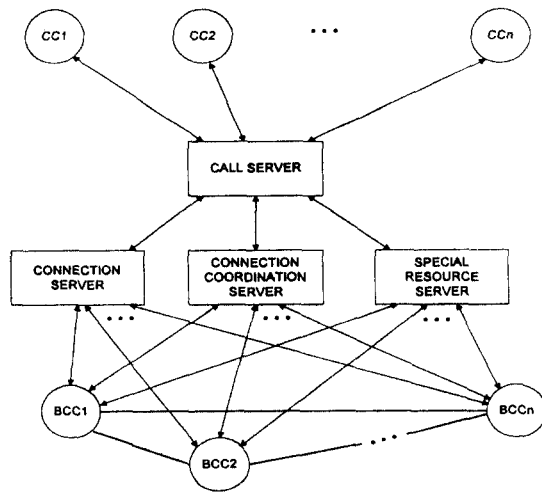
표 1. 서버별로 할당된 호 처리 절차

Table 1. Call processing procedures assigned to each server

호 처리 기능	서버
호 제어 (call control)	Call server
연결 제어 (connection control)	Connection server
다양한 연결의 중재 (connection coordination)	Connection coordination server
특정자원의 제어 (special resource control)	Special resource server

DMMSPP의 구조는 그림 1에서 나타난 바와 같이 각각의 제어기능을 수행하는 서버들과 호 제어(call control)와 베어러 제어(bearer connection control) 등의 기능분리를 이루는 교환기 등으로 구성된다.

DMMSPP에서는 호를 단말과 호 서버간의 신호 연계(association)로 정의한다. 호 설정자는 호 서버에게 다자간 멀티미디어 서비스를 위한 호 설정을 요구한다. 호 설정 요구를 수신한 호 서버는 개설되는 호에 global call identifier를 부여하고 수신된 호 설정 정보를 이용하여 호에 참여하는 모든 단말에게 호 설정을 동시에 수행하게 된다. 이때 호 서버는 각 단말과 트래픽 협상 및 가용성 검사 절차를 수행하고 호 설정 자료의 역방향 트래픽 정보를 수신한다. 각 단말과의 호 설정이 완료되면 global call identifier에 따라 동일 호에 연계시키는 절차를 수행한다. 따라서 호 서버는



CC : Call Control
 BCC : Bearer Connection Control

그림 1. DMMS의 구조
 Fig. 1 The architecture of DMMS

독립된 호를 결합하는 접합 지점(junction point)의 역할을 수행한다. 또한 호 서버는 개설되는 호의 서비스 특성을 분석하여 special resource server로 전달한다. 이 정보는 망내의 특정 자원(special resource)을 확보한 교환기를 지정하고 필요한 제어 절차를 요구하는 특정 자원 서버(special resource server)의 제어 정보로 이용된다.

각 단말과 호 설정을 완료한 후 호 서버는 호 설정자와 호에 참여하는 각 단말로부터 수신된 양방향 트래픽 정보를 이용하여 대역폭 매핑 테이블을 생성한다. 생성된 대역폭 매핑 테이블은 연결 중재 서버(connection coordination server)로 전달되어 다양한 특성의 연결을 관리할 수 있는 제어 정보로 사용된다. 또한 호 설정 정보에 포함된 각 단말의 주소 정보는 연결 서버(connection server)로 전달되어 각 교환기간 배어러 설정을 위한 경로 배정 정보로 이용된다. 이상에서 살펴본바와 같이 호 서버는 특정 서버에 제어 정보를 제공하고, 각 서버에서 수행되는 제어 기능을 요구하고 감독하는 서버 중재자(server coordinator)의 역할을 수행한다.

연결 서버는 호 서버로부터 수신된 주소 정보와 배

어려 정보를 이용하여 경로를 결정한 후 발/착신 교환기 및 중간 교환기에 경로 정보를 제공하여 교환기간 연결을 설정한다. 또한 경로 정보는 연결 중재 서버로 전달되어 다양한 특성의 연결을 중재하기 위해 사용된다.

연결 중재 서버는 호 서버로부터 수신된 대역폭 매핑 테이블 정보와 연결 서버로부터 수신된 경로 정보를 이용하여 호에 포함되는 모든 연결의 배치(configuration)정보를 생성한다. 생성된 배치정보에 기초하여 연결 중재 서버는 연결의 분산 및 통합 또는 선택기능을 수행하기 위한 교환기를 결정한다.

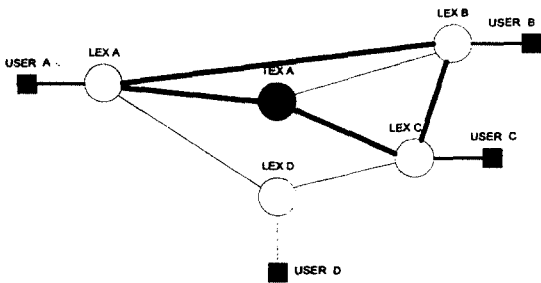
특정 자원 서버는 설정되는 호에 필요한 특정자원을 결정하기 위해 호 서버로부터 서비스 정보와 연결 중재 서버로부터 연결 배치 정보를 수신한다. 수신된 제어 정보를 이용하여 특정 자원 서버는 특정자원이 요구되는 교환기를 지정한다. 설정된 연결 형태에 따라 교환기에서 연결 중재 기능과 특정자원이 모두 요구되는 경우도 있고 각각 분리되어 요구될 수도 있다.

그 외 CC(call control)와 BCC(bearer connection control)는 기존 프로토콜의 기능 모델에서 정의된 개념을 사용한다. 그림 1에서 CC와 BCC는 분리되어 나타나 있다. 그러나 이러한 분리는 각 서버들의 기능 및 인터페이스를 명확히 하기 위한 것으로 실제로는 동일한 색인번호를 가지는 CC와 BCC는 한 교환기에 포함되는 응용 서비스 요소(application service element)이다.

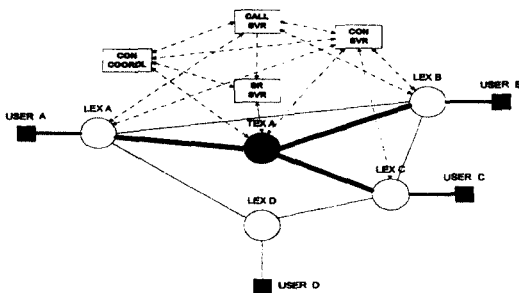
그림 2에는 DMMS 및 기존 프로토콜을 다자간 연결 설정에 적용한 경우의 경로 배정 결과를 비교하였다. 그림 2(a)에서는 기존 프로토콜을 이용하는 경우의 호/연결 설정 결과를 나타내는데 기존 프로토콜에서는 각 단말과의 점 대 다중점 호/연결의 중첩 설정을 통해 호 설정을 이루므로써 각 개별 호의 루트 파티로부터 리프 파티로의 최단 경로의 연결 설정을 이룬다. 즉 User B와 C를 위해 LEX(local exchange) B와 LEX C를 연결하는 경로가 생성된다. 또한 user A와 user B는 LEX A와 LEX B를 연결하는 최단 경로를 형성한다. 결국 각 사용자는 최단 경로를 중심으로 하는 독립적인 경로 설정을 이루게 된다.

기존 프로토콜에서는 동일 호에 포함된 각 연결이 개별적으로 네트워크 자원을 점유하므로 자원 낭비 등의 문제점이 생긴다. 반면에 DMMS를 이용하는

경우 연결 서버 및 연결 증대 서버를 이용한 공통 경로를 설정함으로써 네트워크 자원을 효율적으로 사용한다. 즉 그림 2(b)에서 user A, B, C는 TEX(transit exchange) A를 경유하는 공통의 경로를 이용하고 있다. 이와 같은 공통 경로를 통한 연결 설정은 각 교환기의 종단간 연결에 대한 홉(hop) 수가 증가하는 면이 있을 수 있으나 호 참여자가 늘어날수록 자원의 효율적 사용을 극대화할 수 있는 이점이 있다.



(a) 기존 프로토콜을 사용하는 경우의 연결 설정



(b) DMMS를 이용한 경우의 연결 설정

LEX : Local Exchange CON COORDI : Connection Coordination Server
 TEX : Transit Exchange CON SVR : Connection Server
 CALL SVR : Call Server SR SVR : Special Resource Server

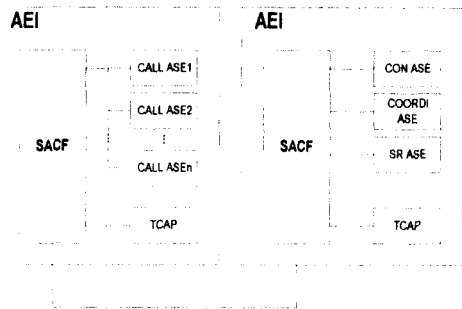
그림 2. 다자간 연결 설정 결과

Fig. 2 The connection establishment between multiparties

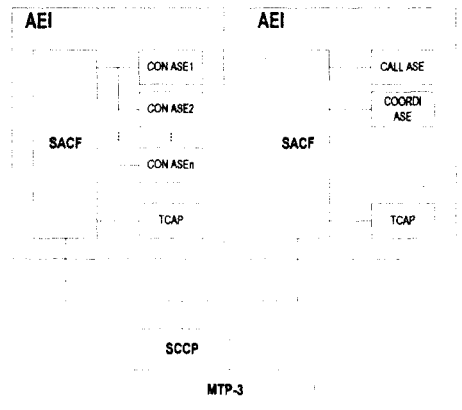
III. 프로토콜 구조

DMMS의 신호 절차는 그림 3의 프로토콜 기능 모델에 따라 구성된다. 기능적으로 분리된 각 서버에서 수행되는 제어기능은 TCAP⁽⁸⁾(transaction capability application part)의 응용 서비스로 정의하고 SCCP⁽⁹⁾

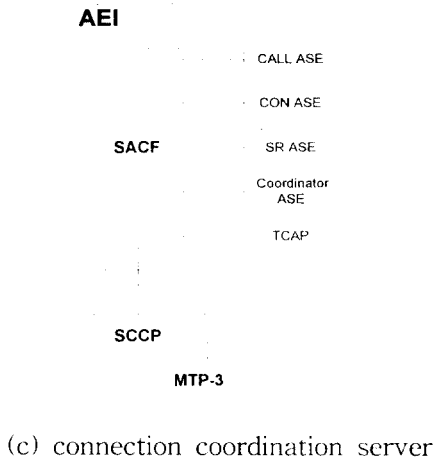
(signaling connection control part)의 비연결 서비스를 이용한다. 각 교환기에서 사용하는 배리어 제어는 연결형 서비스인 MTP-3⁽¹⁰⁾(message transfer part layer3)를 이용하여 배리어 설정을 위한 메시지를 교환한다. 이와 같은 프로토콜의 구조는 망내의 제어 정보를 비연결형 서비스를 사용하여 빠르게 교환할 수 있다.



(a) call server



(b) connection server



SACF : Signaling Association Control Function
 SCCP : Signaling Connection Control Part
 MTP-3 : Message Transfer Part Layer 3
 CC ASE : Call Control ASE
 TCAP : Transaction Capability Application Part
 ASE : Application Service Element
 AEI : Instance of Application Entity
 BCC ASE : Bearer Connection Control ASE

그림 3. 프로토콜 구조
 Fig. 3 Protocol functional model

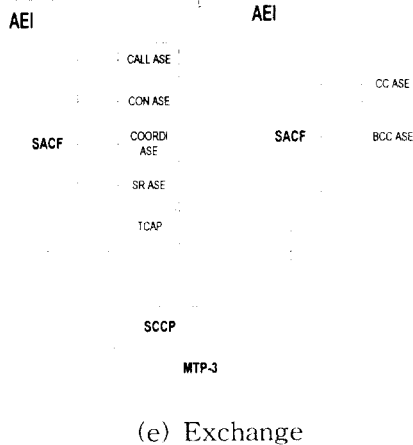
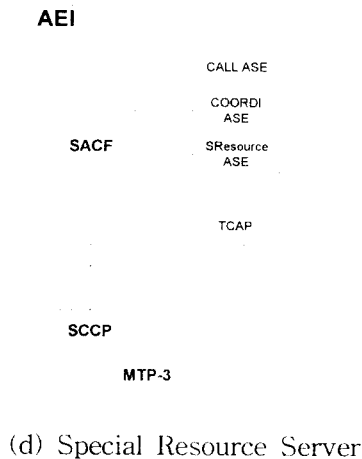


그림 3(a)에서는 호 서버의 프로토콜 구조를 나타내었다. 교환기의 호 제어 응용 서비스 요소(CC ASE)와의 호 설정을 위해 호에 참여하는 단말의 수만큼의 호 응용서비스요소(CALL ASE)가 존재한다. 또한 각 서버와의 신호 연계를 위해 각 서버에 대한 응용 서비스 요소가 존재한다. 각 교환기와 서버와의 제어정보 교환은 TCAP 신호 절차와 SCCP의 비연결 서비스를 이용하여 네트워크에 접속된다.

그림 3(b)에는 연결 서버의 구조를 나타내었다. 호 서버와 유사하게 각 교환기와 신호연계를 위해 n개의 연결 응용 서비스 요소(CON ASE)를 생성한다. 또한 제어정보의 교환을 위해 상호 관련되는 호 서버와 연결 중재 서버에 대하여 각각의 응용서비스 요소를 생성한다.

그림 3(c), (d)는 각각 연결 중재 서버와 특정 자원 서버의 구조를 나타내었다. 이들의 프로토콜 구조 역시 호 서버와 연결 서버의 구조와 유사하다. 각각의 서버가 관련되는 교환기와 서버와의 신호 연계를 위해 특정의 응용서비스 요소가 생성된다.

그림 3(e)에서는 각 교환기의 프로토콜 구조모델을 나타내었다. 교환기의 구조는 망내 제어 정보의 교환을 위해서 SCCP의 비연결형 서비스를 이용하는 TCAP 응용 서비스 요소들과 연결형 서비스인 MTP-3를 통하는 CC, BCC 응용 서비스 요소로 구성된다. 교환기는 각 서버로부터 제어정보를 수신하여 물리적인 자원의 확보 등 구체적인 제어 절차를 이행하는데 이러한 경우 각 서버와의 신호연계를 설정하기 위해, 구축된 TCAP 응용서비스 요소를 이용한다. 그러나 기존의 프로토콜과 마찬가지로 교환기간 베어러 설정이나 사용자와의 신호 절차를 위한 CC ASE와 BCC ASE는 연결형 서비스인 MTP-3를 이용한다.

그림 3에서의 여러 서버와 교환기의 프로토콜 구조는 여러 개의 ASE를 포함하는 복수개의 AEI가 생성되는 형태를 가진다. 이때 다수개의 AEI와 ASE의 식별 절차는 각각의 연결 서비스를 제공하는 SCCP와 MTP-3에 따라 다르다. SCCP의 비연결형 서비스를 이용하는 경우 각 AEI는 SCCP 프로토콜의 SSN(sub system number)에 의해 식별되고 MTP-3를 직접 이용하는 경우 SID(signaling identifier)에 의해 식별된다. 또한 AEI내에 여러 응용서비스 요소는 각 트랜잭션을 수행하는데 할당되는 invoke id와 correlation id로 식별될 수 있다.

IV. 메시지 및 프리미티브

DMMSP는 기본적으로 기존 프로토콜(CS-1, 2.1)에서 발전된 형태의 구조를 가진다. 따라서 각 사용자 단말과 교환기는 호 제어 절차를 위해 사용되는 기존 프로토콜의 메시지 및 프리미티브를 수용한다. 각 사용자 단말과 네트워크 교환기의 사용자 네트워크 인터페이스(user network interface)는 권고안 Q.2931에서 정의한 메시지를 이용한다. 그러나 종단간(end-to-end) 신호 연계의 호 개념 대신에 사용자는 호 서버와의 신호 연계를 통해 호 설정을 이룬다. 따라서 권고안 Q. 2931에서 정의한 메시지를 사용하는 반면에 이들 메시지를 구성하는 정보 요소(information element)는 변경되어야 한다. 한편 망내의 교환기간 인터페이스(network node interface)의 경우는 각 서버와의 제어 신호 교환과 교환기간 배어러 설정 절차 등 크게 두 가지 형태의 신호 절차로 이루어진다. 각 서버와의 제어신호 교환을 위한 신호 절차는 TCAP 프로토콜의 트랜잭션 절차와 SCCP 프로토콜의 비연결형 서비스를 이용한다. 따라서 TCAP의 응용서비스 요소로 정의된 각 서버의 응용서비스 요소는 peer-to-peer 트랜잭션을 위한 프리미티브와 이에 따르는 파라미터를 이용한다. 그러나 각 교환기간의 배어러 설정 절차는 권고안 Q.2764에서 정의한 신호 절차를 따르는데 인접한 교환기와의 국부적인 연결 설정 절차를 수행한다. 다음은 UNI(user network interface)와 NNI(network node interface)에서 기존 프로토콜을 사용할 때 변경되는 정보 요소와 새로 정의되는 프리미티브를 세부적으로 정의한다.

(1) UNI 메시지

DMMSP의 UNI에서는 권고안 Q.2931에서 정의한 메시지를 사용한다. 그러나 다자간 멀티미디어 통신을 위해 메시지에 포함되는 정보 요소는 변경이 요구된다. 추가되는 정보 요소를 정의하면 다음과 같다.

- 텔레서비스 타입(teleservice type)
- 전역 호 식별자(global call identifier)
- 호 참여자의 수(number of participation endpoints)
- 호 참여자 주소(participating endpoint address)
- ATM Traffic Descriptor

(2) NNI 메시지 및 프리미티브

NNI에서 제어정보를 교환하는 절차는 클라이언트-서버 개념을 이용한 트랜잭션으로 이루어진다. 따라서 각 서버에서 정의된 응용서비스 요소는 TCAP의 트랜잭션을 통한 개별적인 프리미티브를 사용한다. 표 2는 각 서버와 서버간 또는 서버와 교환기간의 제어 정보 교환을 위해 정의되는 트랜잭션의 오퍼레이션을 기술하였다. 즉 CALL_INVOKE 오퍼레이션은 호 설정 개시를 위해 발신측 교환기에서 호 서버로 또는 호 서버에서 착신측 교환기로 전달되는 오퍼레이션으로 호 설정 정보가 포함된다. 호 설정 정보를 수신한 각 교환기 및 호 서버는 CALL_RETURN_RESULT 오퍼레이션을 통해 단말 특성 등 수용 가능한 호 설정 정보를 포함하여 응답하게 된다. CON_INVOKE 오퍼레이션은 연결 서버로부터 교환기로 경로 정보를 전달하기 위해 사용되는 오퍼레이션으로서 각 교환기는 인접교환기로의 연결 설정이 완료되면 CON_RETURN_RESULT 오퍼레이션을 통해 연결 서버로 그 결과를 전달한다. COORDI_INVOKE 오퍼레이션은 연결 중재 서버로부터 연결 중재 역할을 수행할 교환기로 전달되는 오퍼레이션으로서 이 오퍼레이션을 수신한 교환기는 중재 기능의 수용 여부를 판단한 후 COORDI_RETURN_RESULT 오퍼레이션을 통해 응답한다. SR_INVOKE 오퍼레이션은 특정 자원 서버로부터 특정 자원을 운영할 교환기로 전달되는 오퍼레이션으로서 이 오퍼레이션을 수신한 교환기는 특정 자원의 운영에 대한 가능 여부를 판단한 후 SR_RETURN_RESULT 오퍼레이션을 통해 응답한다. CALL_CON_INVOKE 오퍼레이션은 호 서버로부터 연결 서버로 각 단말의 주소 정보 등 전역적인 경로 결정을 위한 제어 정보를 전달

하기 위한 오퍼레이션으로서 이 오퍼레이션을 수신한 연결 서버는 각 교환기간 연결 설정이 완료된 후 호 서버로 CALL_CON_RETURN_RESULT 오퍼레이션을 통해 경로 설정 결과를 통보한다. CALL_COORDI_INVOKE 오퍼레이션은 호 서버로부터 연결 중재 서버로 각 단말의 베어러 수용능력 등 연결 중재 기능을 위한 제어 정보를 전달하기 위한 오퍼레이션으로서 이 오퍼레이션을 수신한 연결 중재 서버는 연결 중재 기능을 수행할 교환기를 지정하고 지정된 교환기로부터의 수용 응답을 수신한 후 호 서버로 CALL_COORDI_RETURN_RESULT 오퍼레이션을 통해 수행 결과를 통보한다. CALL_SR_INVOKE 오퍼레이션은 호 서버로부터 특정 자원 서버로 각 참여자의 트래픽 특성 등 특정 자원의 정의 및 제어를 위한 정보를 전달하기 위해 사용되는 오퍼레이션으로서 이 오퍼레이션을 수신한 특정 자원 서버는 요구되는 특정 자원을 정의하고 적용될 교환기를 지정한다. 그리고 지정된 교환기로부터의 수용 응답을 수신

한 후 호 서버로 CALL_SR_RETURN_RESULT 오퍼레이션을 통해 수행 결과를 통보한다. CALL_ESTABLISHED_INVOKE 오퍼레이션은 호 서버로부터 착신측 교환기로 호와 연결의 설정이 완료되었음을 통보하기 위해 사용된다. 그러나 호를 개시한 발신측 교환기로는 CALL_RETURN_RESULT 오퍼레이션을 통해 호 설정 절차의 종료를 통보한다.

네트워크를 구성하는 각 교환기는 호 서버와 연결 서버로부터 호 정보와 베어러 정보를 수신한다. 또한 선택적으로 연결 중재 기능이나 특정 자원을 사용하는 교환기는 연결 중재 서버와 특정 자원 서버로부터 각각의 제어정보를 수신하여 해당 기능을 수행한다. 이들 서버들과의 제어정보는 TCAP 응용서비스로 정의되어 SCCP의 비연결형 서비스를 이용하는데 적용되는 프리미티브는 각 서버에서 정의된 형태와 동일하다. 그러나 MTP-3 연결형 서비스를 이용하는 CC_ASE와 BCC_ASE는 권고안 Q.2764에서 정의한 프리미티브를 사용한다.

표 2. 서버 및 교환기간 트랜잭션 오퍼레이션
Table 2. The transaction operations between servers and exchanges

오퍼레이션	서버 및 교환기(EX)
CALL_INVOKE ----- CALL_RETURN RESULT	call server - EX(CC ASE)
CON_INVOKE ----- CON_RETURN RESULT	connection server - EX(BCC ASE)
COORDI_INVOKE ----- COORDI_RETURN RESULT	connection coordi. server - EX (connection coordinator ASE)
SR_INVOKE ----- SR_RETURN RESULT	SR server - EX(SR ASE)
CALL_CON_INVOKE ----- CALL_CON_RETURN RESULT	call server - connection server
CALL_COORDI_INVOKE ----- CALL_COORDI_RETURN RESULT	call server - connection coordi. server
CALL_SR_INVOKE ----- CALL_SR_RETURN RESULT	call server - SR server
CALL_ESTABLISHED_INVOKE	call server - EX(CC ASE)

이상에서 정의한 DMMSP의 프로토콜 검증을 호 설정 절차 예를 통하여 간접적으로 수행하였으며 보다 정확한 검증을 위해 각 교환기와 서버의 상태를 정의하고 각 상태의 상태 변이를 상태 천이도(state transition diagram)로 나타내었다. 한편 실제적인 적용을 통한 검증은 시험 시스템 구현을 위한 구현 알고리즘을 통해 수행할 수 있다.⁽¹¹⁾

(3) 교환기 측의 상태 정의 및 상태 천이도

교환기 측 시스템의 상태는 다음의 15개로 정의되며 그림 4에서는 이를 이용하여 호 제어 절차에 따르는 발/착신측 및 중간 교환기 측의 상태 천이도를 나타내었다.

- ① Null(ES0): 호가 존재하지 않는 상태.
- ② Call Initiated(ES1): 발신측 교환기가 호 개시 참여자로부터 호 개시 메시지를 수신했거나 착신측 교환기가 호 서버로부터 호 개시 오퍼레이션을 수신했을 때의 상태.
- ③ Call Proceeding(ES2): 발신측 교환기가 호 서버로 호 개시 오퍼레이션을 보내거나 착신측 교환기가 호 참여자로 호 개시 메시지를 보냈을 때의 상태.
- ④ Call Present(ES3): 발신측 교환기가 호 서버로부터 정상적인 호 개시 응답을 수신하거나 착신측 교환기가 호 서버로 호 참여 응답을 보냈을 때의

상태.

- ⑤ Connect Initiated(ES4):연결 서버로부터 인접 교환기에 대한 경로 정보를 수신했을 때의 상태.
- ⑥ Outgoing Connect Proceeding(ES5):인접 교환기로의 순방향 연결(connection)을 위한 메시지를 보내고 응답을 기다릴 때의 상태.
- ⑦ Connect Established (ES6):인접 교환기로부터 순방향 연결에 대한 정상적인 응답을 수신했을 때의 상태.
- ⑧ Incoming Connect Proceeding(ES7):인접 교환기로부터 역방향 연결(backward connection)의 설정 요구를 수신했을 때의 상태.
- ⑨ Connect Accepted(ES8):인접 교환기로부터 역방향 연결(backward connection)의 설정 요구에 대한 정상적인 응답을 보냈을 때의 상태.
- ⑩ Connect Established Complete(ES9):인접한 교환기와의 양방향 연결 설정을 정상적으로 수행한 후 연결서버로 정상적인 응답을 보냈을 때의 상태.
- ⑪ Incoming Coordi Proceeding(ES10):연결중재서버로부터 연결 중재 기능의 요구 오퍼레이션을

수신했을 때의 상태.

- ⑫ Coordi Accepted(ES11):연결중재서버로 연결 중재 기능의 요구 오퍼레이션에 대한 정상 응답(연결 중재 기능의 수용)을 보냈을 때의 상태.
- ⑬ Incoming SR Proceeding(ES12): 특정자원서버로부터 특정 자원의 사용 요구 오퍼레이션을 수신했을 때의 상태.
- ⑭ SR Accepted(ES13):특정자원서버로부터 연결 중재 기능의 요구 오퍼레이션에 대한 정상 응답(특정 자원의 사용)을 보냈을 때의 상태.
- ⑮ Active(ES14): 발/착신측 교환기가 각각 호와 연결 및 연결 중재 기능, 특정 자원의 사용 등 모든 설정이 종료되었다는 응답을 호 서버로부터 수신하고 이를 호 개시자 및 호 참여자로 정상 응답하였을 때의 상태.

(4)서버 측의 상태 정의 및 상태 천이도

서버 측 시스템의 상태는 다음의 17개로 정의되며 그림 5에서는 호 제어 절차에 따르는 각 서버 측의 상태 천이도를 나타내었다.

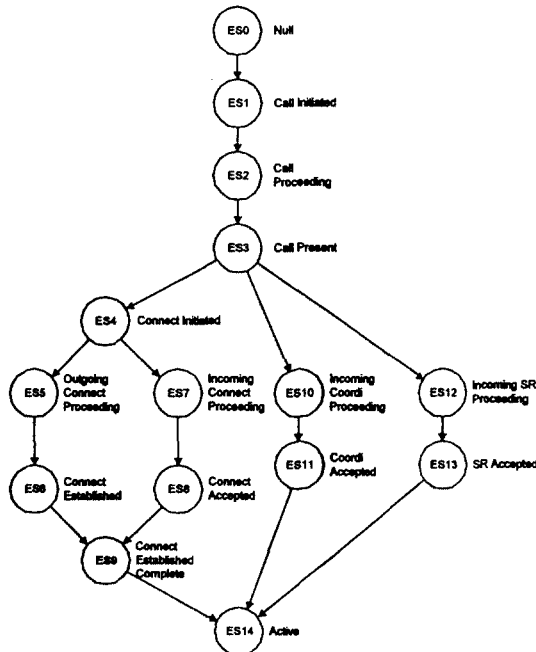


그림 4. 교환기 측의 상태 천이도
Fig. 4 State transition diagram in exchange

- ① Null(SS0): 호가 존재하지 않는 상태.
- ② Call Received(SS1): 발신측 교환기로부터 호 설정 요구 오퍼레이션을 수신했을 때 호 서버에 존재하는 상태.
- ③ Call Proceeding(SS2): 호 참여자 측의 모든 착신측 교환기로 호 설정 요구 오퍼레이션을 보냈을 때 호 서버에 존재하는 상태.
- ④ Call Accepted(SS3): 호 참여자 측의 착신측 교환기로부터 호 설정 요구에 대한 정상 응답을 수신했을 때 호 서버에 존재하는 상태.
- ⑤ Call Present(SS4): 발신측 교환기로 호 설정 요구에 대한 정상 응답을 보냈을 때 호 서버에 존재하는 상태.
- ⑥ Sub_Op Initiated(SS5): 연결서버와 연결중재서버 그리고 특정자원서버로 제어 정보를 전송할 때 호 서버에 존재하는 상태.
- ⑦ Wait Sub_Op Complete(SS6): 연결서버와 연결중재서버 그리고 특정자원서버로부터 제어 기능의 종료를 기다릴 때 호 서버에 존재하는 상태.
- ⑧ Connect Requested(SS7): 호 서버로부터 경로 정보를 수신했을 때 연결서버에 존재하는 상태.
- ⑨ Connect Proceeding(SS8): 호에 참여하는 각 교

환기로 인접 교환기에 대한 경로 정보를 보냈을 때 연결서버에 존재하는 상태.

- ⑩ Connect Accepted(SS9): 호에 참여하는 각 교환기로부터 인접 교환기로의 연결 설정이 이루어졌다는 응답을 수신할 때 연결서버에 존재하는 상태.
- ⑪ Coordi Requested(SS10): 호 서버로부터 대역폭 배핑 테이블 등 연결 중재 기능을 위한 제어 정보를 수신했을 때 연결중재서버에 존재하는 상태.
- ⑫ Coordi Proceeding(SS11): 연결 중재 기능을 수행할 교환기로 연결 중재 기능요구 오퍼레이션을 보냈을 때 연결중재서버에 존재하는 상태.
- ⑬ Coordi Accepted(SS12): 연결 중재 기능을 수행할 교환기로부터 연결 중재 기능의 수용 응답을 수신했을 때 연결중재서버에 존재하는 상태.
- ⑭ SR Requested(SS13): 호 서버로부터 특정 자원의 제어를 위한 제어 정보를 수신했을 때 특정자원 서버에 존재하는 상태.
- ⑮ SR Proceeding(SS14): 특정자원을 사용할 교환기로 특정자원의 제어 요구 오퍼레이션을 보냈을 때 특정자원서버에 존재하는 상태.
- ⑯ SR Accepted(SS15): 특정 자원을 사용한 교환기로부터 특정 자원의 기능 수행 요구에 대한 지상

적인 응답을 수신할 때 특정자원서버에 존재하는 상태.

- ⑰ Active(SS16): 각 호 참여자 측의 교환기로 호와 연결 및 연결 중재 기능과 특정 자원의 사용 등 제어 기능이 정상적으로 이루어졌음을 응답할 때 호 서버에 존재하는 상태. 또한 각 서버에서의 개별적인 제어 기능이 정상적으로 수행되어 호 서버로 정상응답을 보냈을 때 각 서버에 존재하는 상태.

V. 호 설정 절차 예

다자간 회의 호를 위한 호/연결 설정에서 기존 프로토콜은 호 설정자가 루트 파티가 되는 점 대 다중점 호/연결을 설정한 후, 각 호 참여자가 루트파티가 되는 점 대 다중점 호/연결 설정을 중첩 설정함으로써 다자간 회의 호를 설정할 수 있다. 반면에 DMMSF에서는 호 참여자로부터 호 서버로의 호 설정을 이룬 후, 호 서버로부터 각 참여자로의 개별 호를 설정한다. 그리고 호 서버로부터 제어 정보를 수신한 연결서버와 연결 중재 서버를 통한 공통 경로를 형성하여 호 설정을 이룬다. 이러한 DMMSF와 기존 프로토콜에서의 호/연결 제어 절차를 그림 6에 나타내었다.

또한 그림 7에서는 다자간 호 설정 절차의 예로서 DMMSF와 기존 프로토콜을 사용하여 user A, B, C, D가 참여하는 멀티미디어 회의형 서비스를 위한 호 설정 절차를 나타내었다. 그림에서 user A는 호 개설자가 되는데 호 개설자 대신 호에 참여하지 않는 제 3자일 경우에도 유사한 설정 절차를 이룰 수 있다.

기존 프로토콜을 사용하여 호 설정을 이루는 경우 다중점 대 다중점의 연결 형태를 지원하는 신호 프로토콜이 정의되지 않았으므로 점 대 다중점의 연결 형태를 지원하는 신호 프로토콜⁽¹²⁾⁽¹³⁾을 중첩 사용함으로써 다중점 대 다중점 호를 설정한다. 그림 7(a)에서는 호 개설자로부터 호 참여자로의 호 설정 절차에서만 사용되는 메시지를 나타내었는데 이는 절차를 단순하고 이해하기 쉽게 나타내기 위해서이다. 그림 7(a)의 절차를 정리하면 다음과 같다.

- (1) User A는 LEX A로 호 설정메시지 SETUP을 보낸다. 이때 LEX A는 첫 번째 리프 파티를 user B로 정하고 LEX B로 연결설정 메시지 IAM를

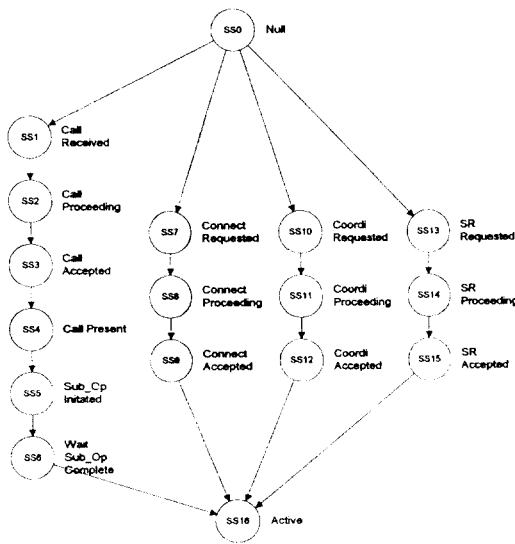
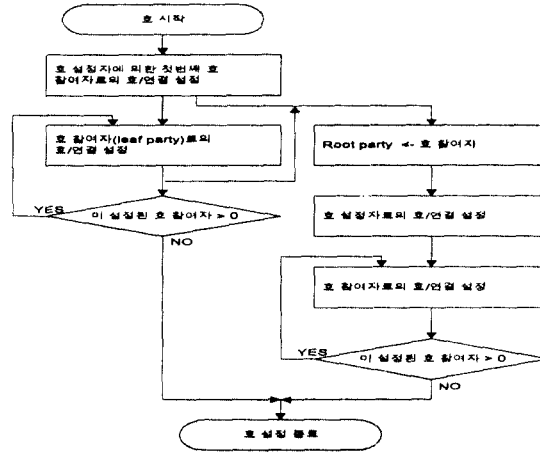
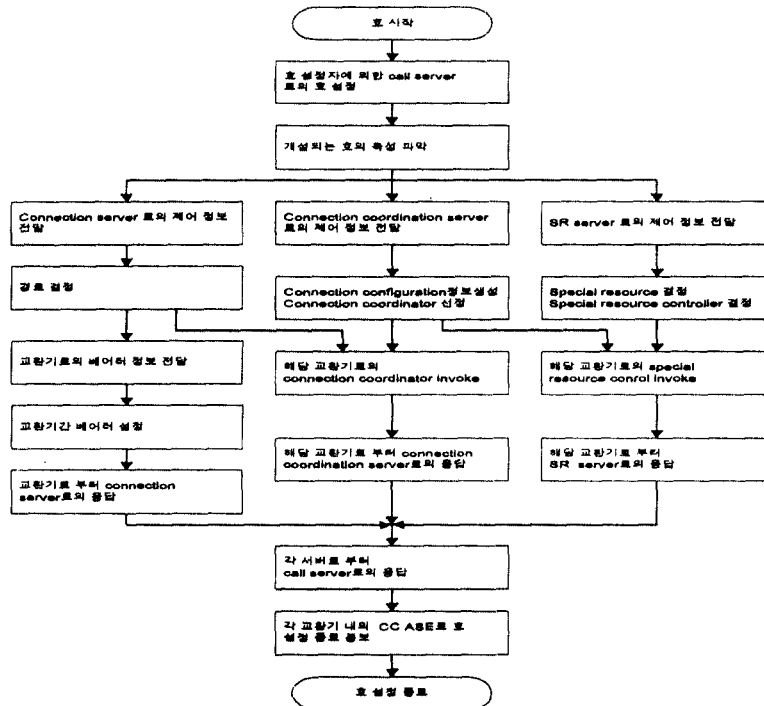


그림 5. 서버 측의 상태 전이도
Fig. 5 State transition diagram in server



(a) 기존 프로토콜에서의 호 제어 메카니즘



(b) DMSP에서의 호 제어 메카니즘

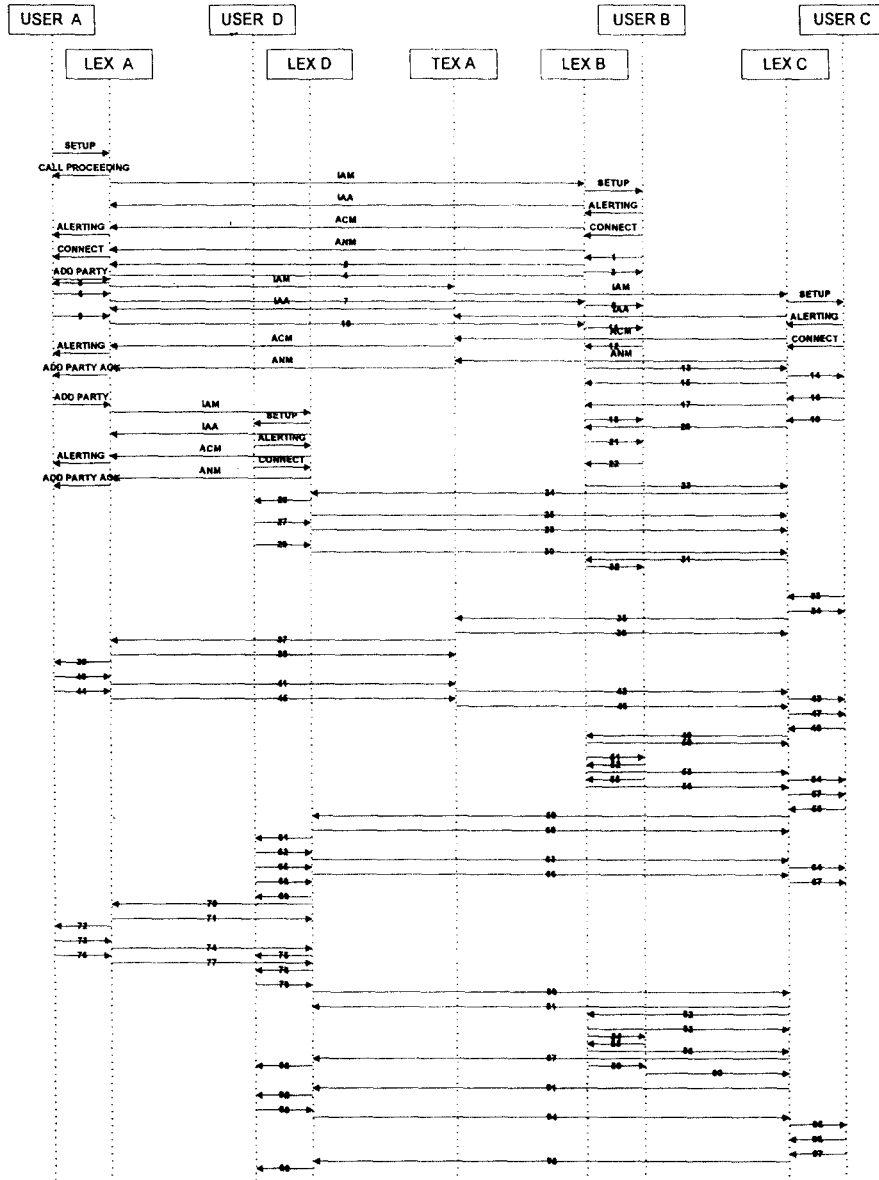
그림 6. 다자간 호/연결 설정에서의 호 처리 절차
Fig. 6 Call processing procedure in multiparty call/connection setup

보낸다. 아울러 user A에게는 CALL PROCEEDING 메시지를 보냄으로써 호 설정절차가 진행 중임을 알린다.

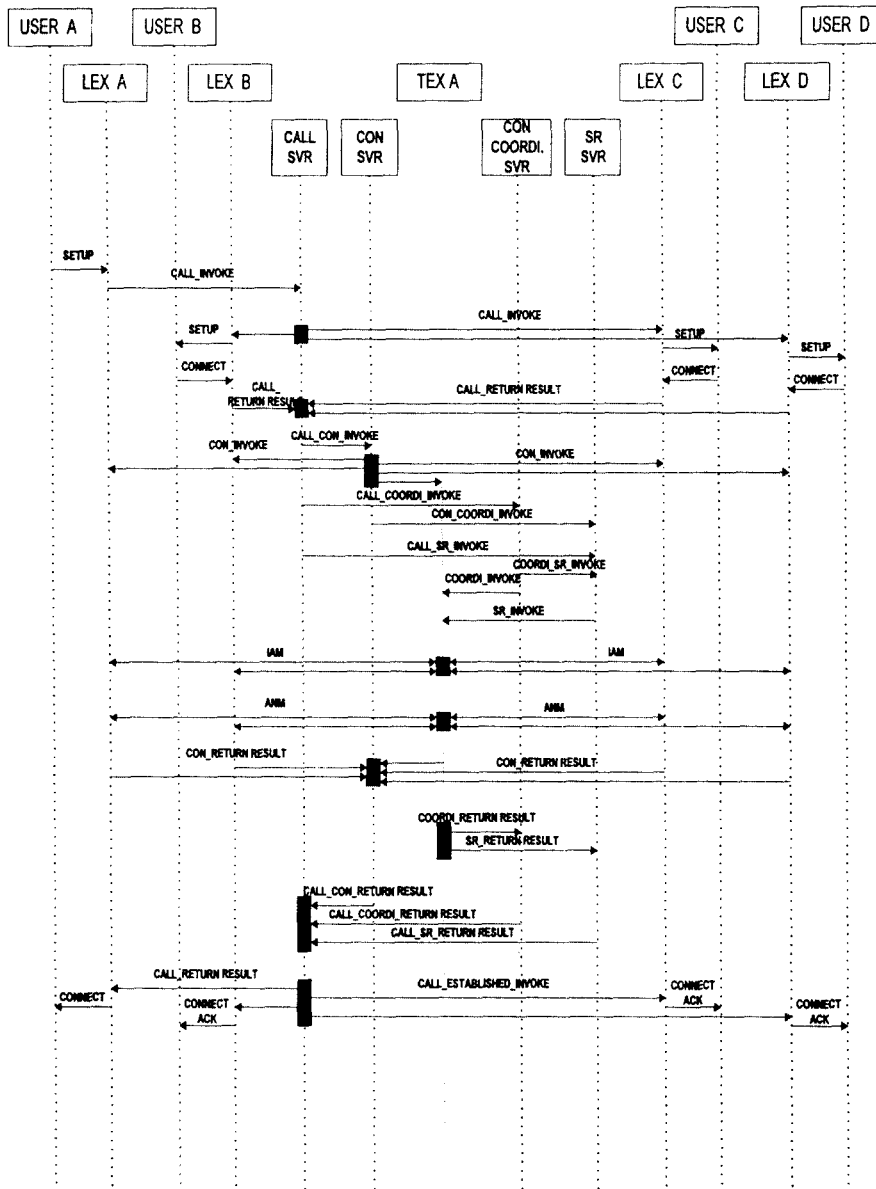
- ② LEX A로부터 호 설정 메시지를 수신한 LEX B 는 user B로 SETUP메시지를 보내고 이에 대한

응답을 기다린다.

- ③ SETUP 메시지를 수신한 user B는 가용성 및 호 완성 점검 절차 등 호 설정절차를 수행한 후 LEX B로 ALERTING 메시지를 보낸다. ALERTING 메시지를 LEX B로 보낸 user B는 사용자



(a) 기존 프로토콜을 이용한 호 설정 절차에



(b) DMMSP를 이용한 호 설정 절차예

그림 7. 다자간 호 설정 절차 예

Fig. 7 The examples of multiparty call setup procedure

에 의한 호 수락 결정이 이루어지면 CONNECT 메시지를 보내어 예약된 연결을 확인(confirm)하는 절차를 수행한다.

- ④ ALERTING 메시지를 수신한 LEX B는 LEX A를 향하여 ACM 메시지를 보낸다. 그리고 CONNECT 메시지를 수신한 LEX B는 user B로 CONNECT ACK 메시지를 보내고 LEX A를 향하여 ANM를 보낸다.
- ⑤ 정상적인 연결 설정으로 ANM 메시지를 수신한 교환기 LEX A는 user B와의 설정 완료 메시지인 CONNECT 메시지를 user A로 보낸다.
- ⑥ User B로의 설정 절차를 마친 user A는 user C와 user D로 순차적인 파티추가 절차를 수행한다. 이때 user A는 LEX A로 user C를 추가하기 위한 ADD PARTY 메시지를 보낸다.
- ⑦ ADD PARTY 메시지를 수신한 LEX A는 LEX B와 user B로의 설정절차와 동일한 파티추가 절차를 수행한다. User C의 파티추가 절차가 정상적으로 수행되면 LEX A는 user A로 ADD PARTY ACK 메시지를 보냄으로써 user C의 파티추가 절차를 종료한다.
- ⑧ ADD PARTY ACK 메시지를 수신한 user A는 user C의 파티추가 절차와 동일한 방법으로 user D의 파티 추가 절차를 수행한다.

User A가 루트파티가 되는 점 대 다중점 호/연결 설정 절차가 각 파티로 종료될 때 각 파티는 user A가 첫 번째 리프 파티가 되고 자신이 루트 파티가 되는 점 대 다중점 호/연결 절차를 수행한다. 그림에서 1번에서 32번까지 색인된 신호 절차는 user B에 의한 설정 절차이고 33번에서 66번까지의 절차와 67번에서 99번까지의 절차는 각각 user C와 D에 의해 진행되는 점 대 다중점 연결 설정 절차이다. 각 회의 참여자가 루트 파티가 되는 연결 설정은 user A로부터 각 참여자로 연결 설정이 종료되는 시점에서 시작된다. 또한 그림에서는 복잡성을 피하기 위해 user B에 의한 연결 설정만이 user B로의 설정 종료 시에 시작되고 user C와 D에 의한 절차는 분리하여 나타내었다.

그림 7(b)에서는 DMMS를 사용하여 호 설정을 이루는 호 설정 절차를 나타내고 있다. 각 서버간 또는 서버와 교환기간의 제어정보 교환은 트랜잭션의 오퍼레이션을 통해 이루어지는 반면에 교환기간의 배

어려 설정에서는 기존의 프로토콜에서의 메시지가 사용되었다. DMMS를 사용한 호 설정 절차를 정리하면 다음과 같다.

- ① User A는 LEX A로 UNI 신호 절차를 통해 호 설정 메시지를 보낸다. 이때 user A는 SETUP 메시지에 회의에 참여할 모든 참여자의 주소와 회의형 서비스의 트래픽 특성을 포함한다.
- ② 호 설정 메시지를 수신한 LEX A는 호 서버로 TCAP의 트랜잭션을 통해 CALL_INVOKE 오퍼레이션을 요구한다.
- ③ CALL_INVOKE를 수신한 호 서버는 호 설정 정보에 포함된 모든 호 참여자에게 CALL_INVOKE 오퍼레이션을 동시에 수행한다. 이때 개설되는 호에 진역 호 식별자를 부여하고 모든 참여자에게 서비스 특성과 트래픽 정보를 포함하는 호 설정 정보를 전달한다.
- ④ 호 서버로부터 CALL_INVOKE 오퍼레이션을 수신한 각 LEX는 UNI 신호 절차를 통해 호 참여자에게 호 설정을 요구한다.
- ⑤ 호 설정 요구를 수신한 각 호 참여자는 자신의 트래픽 수용능력과 가용성(availability) 및 호환성(compatability)점점 절차를 수행한 후 역방향 트래픽정보를 포함하여 호 수락 메시지 CONNECT를 LEX에게 보낸다.
- ⑥ 호 수락 메시지를 수신한 각 교환기는 호 설정 결과 정보를 포함한 CALL_RETURN_RESULT 오퍼레이션을 호 서버로 보낸다. 호에 참여하는 모든 교환기로 CALL_INVOKE 오퍼레이션을 동시에 수행하는 것과 함께 이에 대한 CALL_RETURN_RESULT 오퍼레이션도 각 단말과 교환기의 신호 처리 능력에 따라 동시에 이루어질 수 있다. 그림에서 화살표(메시지나 트랜잭션의 흐름)가 모아지거나 분산되는 검점 네모는 해당 신호 절차가 동시에 수행되는 것을 의미한다.
- ⑦ 모든 호 참여자로부터 호 설정 정보를 수신한 호 서버는 호 설정을 종료하고 베어러 설정과 특정 자원의 설정 절차를 수행한다. 먼저 호 서버는 연결 서버로 모든 호 참여자의 주소 정보를 제공하고 수신된 호 설정 결과에 따른 트래픽 특성을 제공한다. 그리고 정방향과 역방향의 대역폭 특성에 대한 대역폭 매핑 테이블을 생성한다. 대역

쪽 매핑 테이블이 생성되면 연결 중재 서버로 CALL_COORDI_INVOKE 오퍼레이션을 통해 전달된다. 또한 호 서버는 호 개설자로부터 수신된 서비스 특성 정보에 의해 특정자원을 설정하기 위한 서비스 정보를 추출한다. 생성된 서비스 정보는 CALL_SR_INVOKE 오퍼레이션을 통해 특정 자원 서버로 전달된다.

⑧ 호 서버로부터 각 참여자의 주소 정보를 수신한 연결 서버는 전역적인 경로 배정을 수행하여 TEX A를 통한 공통 경로를 생성한다. 경로배정이 끝나면 연결 서버는 각 교환기로의 트랜잭션을 통해 CON_INVOKE 오퍼레이션을 요청한다. 이 오퍼레이션에는 인접 교환기와의 경로 정보와 베어러 설정 정보를 포함한다. 또한 연결 서버는 CON_COORDI_INVOKE 오퍼레이션을 통해 연결 중재 서버로 경로 배정 정보를 전달한다. 호 서버와 연결 서버로부터 대역폭 매핑 테이블과 경로배정 정보를 수신한 연결 중재 서버는 TEX A를 연결 중재 역할을 수행할 교환기로 결정하고 TEX A로 COORDI_INVOKE 오퍼레이션을 보낸다. 여기서 예로든 회의형 서비스에서는 연결 중재기능으로 집성기/분배기(aggregator/distributor)가 결정된다.

⑨ 호 서버와 연결 중재 서버로부터 서비스 특성 정보와 연결 배치 정보를 수신한 특정 자원 서버는 TEX A를 특정 자원이 사용될 교환기로 선정하고 SR_INVOKE 오퍼레이션을 통해 특정자원의 역할을 포함한다. 이 예에서는 TEX A가 연결 중재기능과 특정 자원을 할당하는 등, 두 기능을 모두 수행하는 교환기로 지정되고 특정 자원으로는 음성에서는 브리지와 영상에서는 선택기/분배기(selector/distributor)가 명시된다.

⑩ 한편 각 교환기는 서버로부터의 제어정보를 수신한 후 인접 교환기와 베어러 설정 절차를 수행한다. 이때 사용되는 메시지는 권고안 Q.2762와 Q.2763에서 정의된 메시지(IAM/ANM)가 사용된다. 이 예에서는 각 교환기가 TEX A로 구간 연결을 설정하고 구간 설정된 연결은 전역 호 식별자와 BCC ASE와의 신호 연계에 대한 SID를 사용하여 상호 연계된다. 각 교환기에 의해 인접 교환기간 연결 설정이 완료되면 연결 서버로 연

결 설정 결과를 CON_RETURN_RESULT 오퍼레이션을 통해 전달하는데 이 예에서는 정상적인 설정 절차를 가정함으로써 부정적인 재 설정 절차는 고려하지 않는다. 각 교환기로부터 정상 응답 프리미티브를 수신한 각 서버는 호 서버로 각각의 수행 결과를 보내고 이를 수신한 호 서버는 호 설정을 위한 모든 절차가 성공적으로 종료되었음을 각 단말기에게 보낸다. 이때 사용되는 TCAP 동작은 CALL_ESTABLISHED_INVOKE인데 이는 호 설정 종료를 알리기 위해 정의된 특정 오퍼레이션이다.

한편 설정된 회의 호에 새로운 참여자가 추가 또는 삭제되는 경우 기존 프로토콜에서는 호 설정자로의 통보를 통한 점 대 다중점 호/연결 절차의 추가 수행으로 이루어질 수 있다. 그러나 DMMS에서 호 참여자의 추가/삭제시 호 서버로의 CALL_INVOKE 오퍼레이션을 전달함으로써 호 서버에 의해 참여자 추가/삭제 절차를 수행할 수 있다.

그림 7에서 보이는 바와 같이 기존 프로토콜과 DMMS를 사용한 각각의 신호 절차는 여러 측면에서 비교될 수 있다. 각각의 프로토콜에서 사용된 메시지와 설정지연의 측면에서 정량적으로 비교해 보면 표 3에서 나타낸 바와 같다.

표 3. 기존 프로토콜과 DMMS의 비교

Table 3. The Comparison DMMS with the existing protocol

항 목	기존 프로토콜	DMMS
총 메시지 수	134	60
STEP 수	67	21
설정 절차의 병렬성	50 %	75 %

먼저 교환 메시지 수에 있어서 기존 프로토콜을 사용하는 경우 순차적인 점 대 다중점 호의 중첩사용을 통해 호 설정을 이루므로 점 대 다중점 호 설정에 필요한 메시지 수에 회의 참여자 수를 곱한 값만큼의 메시지가 요구된다. 그러나 DMMS는 서버간의 제어 메시지의 교환을 통해 동시 호 설정을 이룰 수 있는 제어절차를 사용하므로 기존 프로토콜을 사용하

는 경우의 필요 메시지에 비해 약 45% 정도의 메시지만이 사용됨을 볼 수 있다.

그림 7의 신호 절차에서 각 메시지가 교환기나 사용자 단말에서 처리되는 시간이 균일하다고 가정하여 단일 메시지 처리 구간과 중복 처리 구간을 각각 단위 스텝(step)으로 정의하고 멀티미디어 회의 호를 구성하기 위한 신호 절차에 필요한 스텝 수를 비교하였을 때 기존 프로토콜을 이용하여 호를 설정하는 경우 호 설정에 소요되는 스텝 수를 계산하면 67 스텝이 된다. 그러나 DMMSP는 서버에 의해 제어 메시지가 동시 전달되어 개별적으로 처리가 이루어지므로 기존 프로토콜을 사용하는 경우보다 적은 수의 스텝이 소요된다. 따라서 이와 같은 회의형 서비스를 지원하는 호에서 DMMSP는 기존 프로토콜에 비하여 짧은 시간에 호 설정을 이룰 수 있다.

또한 멀티미디어 회의 호를 설정하기 위한 제어 절차의 병렬성을 비교해 보면 기존 프로토콜에서는 전체 호 설정을 위한 제어 절차의 50%가 동시 수행을 통해 호 설정을 이룰 수 있는 반면에 DMMSP는 75%가 병렬로 호 설정 절차를 수행할 수 있다.

V. 결 론

본 논문에서는 다양한 연결 형태의 다자간 멀티미디어 통신을 지원하기 위한 신호 프로토콜로서 DM-MSP를 제안하였다. DMMSP는 클라이언트-서버 개념을 이용하여 호 처리를 분산 수행한다. 즉 호 서버, 연결 서버, 연결 중재 서버, 특정 자원 서버 등 분산된 각 기능에 전용의 서버를 구성하여 독립적인 호 설정을 이룬다.

DMMSP에서는 분리된 호 제어 기능을 수행하는 서버와 각 서버간의 제어정보 교환을 위한 제어 네트워크를 구성해야 하는 문제점이 있다. 그러나 호 서버와 연결 서버에 의해 호와 연결 설정을 각각 병렬로 수행함으로써 다자간 호 설정에서 문제가 될 수 있는 설정지연을 최소화할 수 있다. 또한 연결 서버에 의해 경로배정을 전역적으로 수행하여 공통경로를 생성함으로써 네트워크 자원의 활용을 높이고 네트워크내에서 수용할 수 있는 호를 증가시키는 효과가 생기므로 네트워크의 성능 향상에도 기여할 수 있다. 그 밖에 연결 중재 서버와 특정 자원 서버를 정의

함으로써 양방향의 이질적인 특성을 가지는 다양한 연결 형태를 지원할 수 있으며 멀티미디어 서비스나 다양한 코드 형태를 가지는 서비스를 지원하는 등 서비스를 보다 유연하게 수용할 수 있는 장점이 있다.

이상의 DMMSP의 장점을 다자간 멀티미디어 회의 호를 예로 하여 기존 신호 절차와 비교하였다. 이와 같이 제안된 DMMSP 신호 구조는 B-ISDN 신호 기능 진화의 최종 목표구조에 적용될 수 있을 것이다.

참 고 문 헌

1. ITU-T Recommendation I.31x, B-ISDN network requirements.
2. 박승준, 송주석, "호와 베어러 분리제어를 고려한 B-ISDN 다자간/다중연결 호 구성절차 모델링", 한국통신학회논문지, Vol. 20, No. 1, pp. 149-164, Jan., 1995.
3. Thomas F. La Porta, "An Experimental Signaling Architecture and Modular Signaling Protocols", Proceeding of ISS'95, pp. 244-248.
4. Steven Minzer, "A Signalling Protocol for Complex Multimedia Services", IEEE JSAC., Vol. 9, No. 9, Dec 1001, pp. 1383-1394.
5. Luca Cipriani, Luigi Ronchetti, Stefano Salsano, "An Intelligent Broadband Network Architecture for Multimedia Multiparty Service Control", Proceeding of Globecom '95, pp. 587-591.
6. ITU-T Recommendation Q.2931, B-ISDN DSS2 UNI Layer3 Specification for Basic Call/Connection Control.
7. ITU-T Recommendation Q.2764, B-ISDN SS7 B-ISUP for Basic Call Procedure
8. ITU-T Draft Recommendation Q.771-774 Transaction Capabilities Application Part.
9. ITU-T Recommendation Q.711-714 Signalling Connection Control Part(SCCP).
10. ITU-T Recommendation Q.701-705 Message Transfer Part(MTP).
11. 강종국, "광대역중합정보통신망에서의 분산형 다자간 다중 연결 신호 방식 연구", 숭실대학교 석사 학위 논문, Feb., 1997.

- 12. ITU-T Draft Recommendation Q.2971, B-ISDN DSS2 UNI Layer3 Specification for Point-to-Multipoint Call/Connection Control.
- 13. ITU-T Draft Recommendation Q.2722.1, B-ISDN SS7 B-ISUP Point-to-Multipoint Call/Connection Control.



강 종 국(Jong-Kook Kang) 준회원
 1995년 2월: 숭실대학교 전자계산학과 졸업(공학사)
 1997년 2월: 숭실대학교 대학원 전자계산학과 졸업(공학석사)
 1996년 12월~현재: (주)한화/정보통신 연구소 무선가입자망 연구그룹

※주관심분야: 무선가입자망(WLL), Wideband CDMA CAI protocol, ATM(B-ISDN) protocol 등

김 영 한(Young-Han Kim)

정회원

한국통신학회지 제22권 제5호 참조



김 병 기(Byung-Gi Kim) 정회원
 1977년: 서울대학교 전자공학과 졸업(공학사)
 1979년: 한국과학기술원 전산학과 졸업(이학석사)
 1997년: 한국과학기술원 전산학과 졸업(이학박사)
 1979년~1982년: 경북대학교 전자공학과 전임강사

1982년~현재: 숭실대학교 정보과학대학 컴퓨터학부 교수

※주관심분야: ATM, 이동통신, 컴퓨터구조, 병렬처리