컴퓨터 지원 협력 작업의 세션 초기 프로토콜에서 실행되는 오류 관리기

고 응 남
백석대학교 정보통신학부
e-mail : ssken@bu.ac.kr

An Error Manager running on SIP of Computer Supported Cooperated Work

Eung-Nam Ko
Division of Information & Communication, Baekseok University

요 약

본 연구는 멀티미디어 세션에 참여한 참여자간의 효율적인 의사소통과 상호협력 환경의 향상을 위하여 다양한 형태로 응용의 변화를 주는 세션 초기 프로토콜(SIP: Session Initiation Protocol)에서의 오류 관리기의 설계에 대한 연구이다. 제안하고자 하는 오류 관리기를 이용하면 멀티미디어 응용 개발 환경에서의 오류 발생 시에 객체를 동적으로 생성 및 제거함으로써 자신의 컴퓨터 시스템 상황에 맞는 세션 초기 프로토콜에서의 세션을 진행할 수 있고, 유동적인 네트워크 트래픽에서도 진행 중인 세션 초기 프로토콜에서의 세션을 유지시킬 수 있을 뿐만 아니라, 오류가 발생된 응용을 제외한 객체만의 조합으로 다양한 형태의 세션 초기 프로토콜에서의 세션을 만들 수 있다.

1. 서론

1990 년대에 들어 개인용 컴퓨터의 성능 향상으로 컴퓨터 네트워크로도 영상 장비를 이용한 화상 전송과 비교할 수 있을 정도의 영상 품질을 얻을 수 있게 되었다. 최근에는 하나의 일을 여러 사람이 동일한 작업 공간에서 면대면 효과를 유지하면서 협력하여 문제를 해결하는 컴퓨터 지원 협력 작업(CSCW: Computer Supported Cooperated Work)으로 발전하고 있다[1-4]. 동시에 여러 종류의 데이터를 전송해야 할 뿐만 아니라 미디어 데이터 사이의 관계성을 고려해야 하기 때문에 미디어 통신과 세션 관리 등이 필요하다[5]. 세션 초기 프로토콜(SIP: Session Initiation Protocol)을 확장하여 적용할 수 있는 응용 서비스 분야는 인터넷 전화 서비스뿐만 아니라 인스턴트 메시징 서비스, 컨퍼런스 서비스, 이동성 지원 서비스 등 인터넷 망을 통한 커뮤니케이션을 위한 응용 분야에 다양하게 적용할 수 있다[6].

최근 들어 이러한 컴퓨터 협력작업의 세션 초기 프로토콜 환경이 증가하고 있는데 반하여 이러한 시스템에서의 망관리, 특히 세션 종료 등 응용 오류에 대한 연구는 미흡한 실정이다[7,8].

제안하고자 하는 세션 초기 프로토콜에서의 오류 관리기(Error Manager)를 이용하면 멀티미디어 응용 개발 프레임워크에서 오류 발생시 자동적으로 객체를 동적으로 생성 및 제거함으로써 자신의 컴퓨터 시스템 상황에 맞는 세션을 진행할 수 있고, 유동적인 네트워크 트래픽에서도 진행 중인 세션을 유지시킬 수 있을 뿐만 아니라, 오류를 제외한 응용 객체만의 조합으로 다양한 형태의 세션을 만들 수 있다. 본 연구에서 제안하는 모델은 분산 멀티미디어 프레임워크의 세션 초기 프로토콜에서 사용하는 객체를 컴포넌트화하여 동작하게 하고, 멀티캐스트를 이용한 네트워크를 기초로 하였다. 본 논문의 구성은 2 장에서는 컴퓨터 지원 협력 작업, 3 장에서는 세션 초기 프로토콜에서의 오류 관리기, 4 장에서는 평가 및 결론을 기술한다.

2. 컴퓨터 지원 협력 작업

기존의 분산 멀티미디어 상호 협동 환경을 지원하기 위한 연구들이 주로 UNIX 시스템을 기반으로 하여 이루어졌다.

2.1 기존 컴퓨터 지원 협력 작업

MERMAID 는 분산형 응용 공유 구조를 선택하면서, 공유 이벤트의 분배를 이벤트 발송 부분에서 처리함으로써 다양한 응용의 지원을 고려하고 있다. MMConf 는 분산 형 응용 공유 구조를 선택하였으며, X-윈도우즈를 기반으로 설계되어 있다. Critique 은 복제형 응용 공유 구조를 선택하였으며, 여기에서 발생하는 일치화 문제를 해결하는데 중점을 두었다. QuiX 는 중앙 집중형과 복제형 구조를 선택하였으며, 특히 매킨토시와 X 윈도우 시스템 등의 이기종으로 구성된 환경에서의 응용 공유 방법을 제안했다. EMX 은 X 에 기반을 둔 이기종 컴퓨터 환경에서 응용을 공유할 수 있으며, 모든 사용자들이 공유되는 응용을 완전히 제어할 수 있도록 하는데 중점을 두었다. SCOOT 은 기존의 응용 프로그램을 최소한의 수정으로 공동작업에 적합한 응용으로 확장하는 방법에 대해 논의한다. Argo 은 프록시 서버를 통해서 기존의 X 응용 프로그램을 공유하는데, 특정 응용들만 공유 가능하다. 또한, 여기에서는 윈도우 시스템 기반과 툴 킷 시스템 기반의 복제를 제안하였다. CECED 은 중앙 집중형 구조와 복제형 구조의 혼합 구조를 지원하며, 화면 공유 개념을 확장하였다. BERKOM 은 어떤 상황 하에서라도 새로운 참여자가 공유 환경에 참여할 수 있는 동적 공유 기능과 암시적 발언권 전달 정책을 사용하였다. XpleXer 은 X 윈도우 시스템에서 응용 공유를 지원하는데, 선택적 윈도우 공유, 동적 공유 등을 지원한다[9-17].

2.2 기존 컴퓨터 지원 협력 작업의 문제점

기존 컴퓨터 지원 협력 작업은 오류 발생시 자동적으로 객체를 동적으로 생성 및 제거함으로써 자신의 컴퓨터 시스템 상황에 맞는 세션을 진행할 수 없고, 유동적인 네트워크 트래픽에서도 진행 중인 세션을 유지시킬 수 없을 뿐만 아니라, 오류를 제외한 응용 객체만의 조합으로 다양한 형태의 세션을 만들 수도 없다. 응용사용자나 응용 개발자의 관점에서 보면 자원의 사용과 분배에 관계없이 사용자 응용 인터페이스를 통해 필요 자원을 요구하면 각 자원의 서비스 제공자는 가능한 서비스를 응용에게 제공한다. 분산 멀티미디어 공동 작업의 세션 초기 프로토콜 환경을 구축하고 응용 개발을 위해 필요한 자원의 할당, 제어 등을 제공함으로 응용 개발자들은 응용 고유의 기능만을 추가할 수 있도록 해야 한다.

3. 세션 초기 프로토콜에서의 오류 관리기

응용에서 오류가 발생했을 때 동작하는 세션 초기 프로토콜 환경에서의 오류 관리기에 대한 설계 사항 중의 하나인 진행 중인 세션에서의 객체 생성 및 종료에 대하여 기술한다.

3.1 세션 초기 프로토콜에서의 네트워크 환경

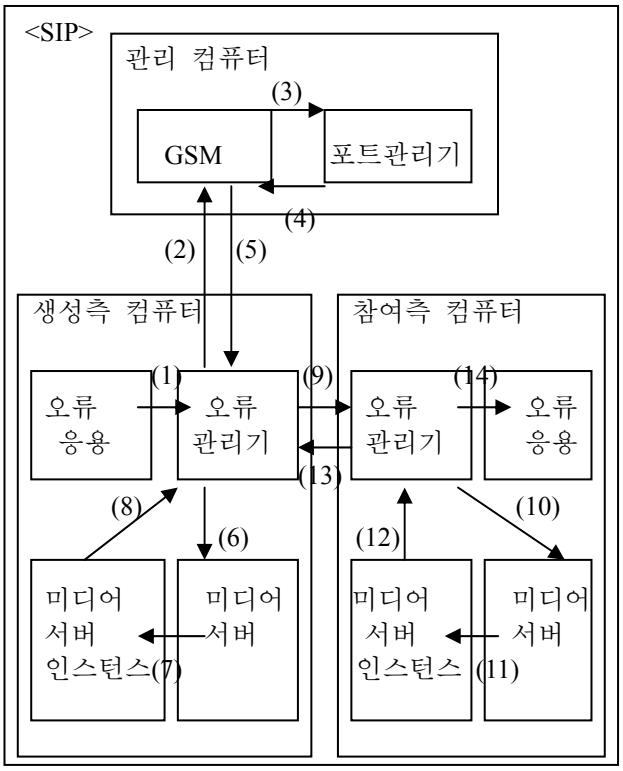
그룹통신을 지원하기 위한 방법은 TCP/IP 나 UDP/IP 를 이용하고 전송계층의 프로그램 지원으로 그룹 통신을 지원하는 방법과, 멀티캐스트를 이용하는 방법이 있을 수 있다. 본 제안 모델에서는 IP 계층에서 호스트에 제공하는 멀티캐스트를 이용하였다. UDP/IP 브로드캐스팅도 다수의 호스트에 동시 전송이 가능하지만 호스트를 지정할 수 없어 그룹 전송을 하지 못하고 그룹의 가입과 탈퇴가 자유롭지 못하기 때문에 IP 멀티캐스트를 사용하였다. SIP 는 User agents, Registras, Proxy servers, Redirect servers 네 가지 논리적 구성 요소로 구분할 수 있다. User agent 는 연결을 요청하는 시스템과 연결 요청을 받는 시스템을 말한다. Registras 는 부여된 네트워크 도메인 내의 사용자들을 관리하는 시스템이다. Proxy servers 는 응용 계층의 라우터로서 요청 메시지를 받아 상대방 시스템이나 다른 Proxy 서버에 전달하고 요청 메시지에 대한 응답 메시지를 보내주는 시스템이다. Redirect server 는 요청 메시지를 받아 해당 메시지에 적합한 User agent 다른 위치나 해당 User agent 를 찾을 수 있는 서버를 알려준다. 또한 SIP 연결 방식은 SIP 클라이언트가 SIP 서버와 서로 통신하는 서버-클라이언트 형태를 하고 있다[18].

3.2 세션 초기 프로토콜 환경에서의 오류 관리기

세션 초기 프로토콜 환경에서 세션 내에서의 오류 관리기를 위한 환경에서 전체 세션 관리자(GSM), 포트 관리기, 생성 측 컴퓨터 및 참여측 컴퓨터를 갖는다. 전체 세션관리자는 세션의 전반적인 관리를 하고, 포트 관리기는 포트의 안정적 분배를 한다. 즉, 생성자 측 컴퓨터와 참여자 측 컴퓨터는 동일한 형태의 응용 환경을 갖는다.

(그림 1)에서 오류가 발생한 응용은 변경하고자 하는 객체를 선택한 후 그 요구를 오류 관리기에 전송하면 오류 관리기는 GSM 에게 객체간의 보장된 전송을 위하여 필요한 포트번호를 할당 받는다. 이 할당받은 포트번호를 각각의 객체에게 전송시켜 줌으로써 생성자 컴퓨터 영역에서의 객체 추가의 기능을 완료한다. 이러한 결과를 참여자 컴퓨터의 오류 관리기에 전송함으로써 동일한 방법으로 미디어 서버 및 미디어 서버 인스턴스에게 동일한 포트번호를 전송하게 된다. 따라서 각각의 미디어 서버 인스턴스 간의 데이터 전송이 독자적으로 이루어지게 된다. 객체 삭제를 원할 때도 동일한 순서에 의하여 오류 관리기는 동작하게 된다. 오류 관리기는 사용을 원하는 객체에 할당

할 포트번호를 전체 세션 관리자에게 요청한다. 전체 세션 관리자로부터 제공된 포트번호를 오류 관리기를 통하여 각 객체에게 전송한다. 오류 관리기는 이 값을 세션에 참여 중인 모든 참여자의 오류 관리기에 전송한다. 세션 관리기로부터 얻은 정보를 이용하여 각 객체간의 통신을 맺는다. 세션 초기 프로토콜 환경에서의 오류 관리기는 객체간의 통신을 종료시킨 후 전체 세션 관리자에게 포트번호를 반환한다. 전체 세션 관리자는 다른 오류 관리기에게 재 제공할 수 있다.



(그림 1) SIP 에서의 동적 오류 관리기 흐름도

3.3 세션 초기 프로토콜 환경에서의 오류 관리기의 기능

멀티미디어 프레임워크의 세션 초기 프로토콜 환경에서 사용하는 객체로는 음성 교환을 위한 오디오 서버, 면대면 화상 정보 전달을 위한 비디오 서버, 공동작업을 위한 노트인 화이트보드, 윈도우 화면을 공유하는 응용공유 등을 들을 수 있다. 이러한 객체들이 지역 세션 관리기를 통하여 원활하게 동작하며 상호 연동적으로 동작하고 있지만 고정된 형태의 세션을 계속 지속하여야 한다는 단점을 가지고 있다. 이러한 문제를 해결하기 위하여 오류가 발생한 시점에서 원하는 형태의 객체를 사용토록 하기 위하여 세션 초기 프로토콜 환경에서의 오류 관리기의 적용이 필요하게 된다. 세션 생성자가 처음 세션을 형성할 때 예상된 사용 객체에 대하여 초기에 설정을 한다. 이 때 필요로 하는 요소는 다음 <표 1>과 같다. 이처럼 사용하고자 하는 객체 자원에 대하여 초기 환경 설정 시 True 또는 False 값을 설정할 수 있다. 세션 진행 중 요구되는 사용 객체의 변화에 대하여서는 <표 2>에서와 같이 생성 및 종료에 해당하는 객체에 대한 사용 여부를 지정하게 된다.

<표 1> 세션 생성 초기화

```
m_xSM.SetEnvironment1
( _strSessionName, _strCreatorName, _siSessionMode,
_siFloorMode, _strAppName, (long*)dwParticipantIP,
_bAudio, _bVideo, _bAppSharing, _bWhiteBoard,
          _bElectBoard);
```

<표 2> 객체 변화 요청

```
m_xSM.CreateObject(
  _bAudio, _bVideo, _bAppSharing,
  _bWhiteBoard, _bElectBoard);

  m_xSM.DestroyObject(
  _bAudio, _bVideo, _bAppSharing,
  _bWhiteBoard, bElectBoard);
```

객체 생성 시 True 값을 가지는 해당 오류 객체에 대하여 오류 관리기에게 요청을 하게 되면 오류 관리기는 전체 세션 관리기를 통하여 기 등록된 세션 정보에 객체 정보를 갱신하고 포트 관리기를 통하여 여분의 포트를 얻어오게 된다. 변경된 객체정보에 대하여 오류 관리기는 참여자 측의 모든 오류 관리기에게 동일한 정보를 전달하여주어 세션의 동기를 맞추게 되며 이에 따라 새로 생성된 객체는 해당 포트를 사용하여 직접 데이터 전송을 하게 된다. 객체 종료 시에도 이와 유사한 경로를 통하여 객체 종료 명령을 전달하게 되는데 기존에 사용하던 포트는 포트 관리기에게 반납하게 되어 포트의 재사용이 가능하도록 하였다.

4. 평가 및 결론

본 연구는 세션 초기 프로토콜 환경에서 멀티미디어 세션에 참여한 참여자간의 효율적인 의사소통과 상호협력 환경의 향상을 위하여 다양한 형태로 응용의 변화를 주는 세션 초기 프로토콜 환경에서의 오류 관리기의 설계에 대한 연구이다. 기존의 멀티미디어 응용 개발 프레임워크에 세션 초기 프로토콜 환경에서의 오류 관리기의 추가로 다음과 기능의 향상을 얻을 수 있다. 첫째, 세션 초기 프로토콜 환경에서 오류 관리기의 다양한 객체 사용 및 해제 기능을 통하여 참여자 컴퓨터 시스템의 사양에 적절한 세션을 유지시킬 수 있어 제한된 자원만으로도 세션에 참여할 수 있게 되어 보다 사용 가용영역을 확장할 수 있다. 둘째, 사용하고자 하는 객체만을 선택하여 특성화된 세션 초기 프로토콜 환경에서 세션을 유지시킬 수 있을 뿐만 아니라, 세션 진행 중 전과 다른 객체를 선택함

으로써 다양한 세션 형태를 만들어 갈 수 있게 되어 세션의 지속성 및 다양성을 이룰 수 있다. 셋째, 한정된 포트자원의 재활용을 통하여 효율적인 포트번호를 사용을 할 수 있어 다른 네트워크 프로그램과 발생할지도 모를 오류수신을 방지할 수 있다.

앞으로의 연구과제는 세션 초기 프로토콜 환경에서 오류 및 객체 관리기와 해당 객체와의 관계를 정형화된 형태의 모듈인 SDK 이나 DLL 형태로 제작하여 쉽게 오류 관리기의 기능을 소화할 수 있는 객체를 지원하여 여러 형태의 오류 발생에도 세션 초기 프로토콜 환경에서 오류 관리기가 수용할 수 있도록 하여야 한다. 그리고 세션 초기 프로토콜 환경에서 표준 프로토콜에 적용에 대한 보완이 필요하다고 본다.

참고문헌

[1] R. Steinmetz and K.N. Nahrstedt, “Multimedia, Communications & Applications”, Prentice Hall, 1995.

[2] F.R.Edwards and M.F.Schultz, “A priority Media Access Control Protocol for Video Communication Support on CSMA/CD LANs”, ACM Press and Springger International, Vol.2, No. 1-6, 1994, pp.243-255. .

[3]J.D. Palmer and N.A.Field, “Computer Supported Cooperative Work”, IEEE Computer, May 1994, pp.15-17.

[4]J.Grudin, “Computer Supported Cooperative Work: History and Focus”, IEEE Computer, May 1994, pp.19-26.

[5]W.R. Stevens, TCP/IP Illustrated, Volume I, II, Addison Wesley Pub., 1993.

[6]안영두, 이순흠, “SIP 기반 유무선 통합 컨퍼런스 시스템 개발”, 한국정보기술학회 논문지 제 5 권 제 3 호, 2007 년 9 월, pp.9-16.

[7]Victor P. Nelson and Bill D. Carroll, “Fault-Tolerant Computing”, IEEE Computer Society Order Number 677, Library of Congress Number 86-46205, IEEE Catalog Number EH0254-3, ISBN 0-8186-0677-0.

[8]Eung-Nam Ko, Chul Hwang, Dae-Joon Hwang, “Implementation of an Error Detection-Recovery Software for Interactive Multimedia Environment by using Hook Technique: EDRSHT”, In proceedings of IEEE/IEE ICT’99, Cheju, Korea, June 15-18, 1999, pp.340-344.

[9]T. Ohmori and K. Watabe, Distributed Cooperative Control for Application Sharing Based on Multiparty and Multimedia Desktop Conferencing Systems:MERMAID, 4th IEEE ComSoc International Workshop on Multimedia Communications, April 1-4, 1992.

[10]Torrence Crowley and Raymond Tomlinson, MMConf: An Infrastructure for Building Shared Multimedia Applications, CSCW ’90 Proceedings, October 1990.

[11]J. Chris Lauwers and Allyn L. Romanow, Replicated Architectures for Shared Window Systems: A Critique, Proceedings of the Conference on Office Information Systems, March 1990.

[12]Vincent Phuah and Steve Gutfreund, Developing Distributed Multimedia Applications, 4th, IEEE ComSoc International Workshop on Multimedia Communications, April 1-4, 1992.

[13]Earl Craighill and Kathryn Gruenefeldt, SCOOT: An Object-Oriented Toolkit for Multimedia Collaboration, Proceedings ACM Multimedia ’94, October 15-20 1994.

[14]Hania Gajewska and David D. Redell, Argo: A System for Distributed Collaboration, Proceedings ACM Multimedia ’94, October 15-20 1994.

[15]Earl Craighill and Keith Skinner, CECED: A System For Informal Multimedia Collaboration, Proceedings ACM Multimedia ’93, August 1-6 1993.

[16]Michael Altenhofen and Thomas Steinig, The BERKOM Multimedia Collaboration Service, Proceedings ACM Multimedia ’93, August 1-6 1993.

[17]Wladimir Mineko, The Application Sharing Technology, The X Advisor, June 1995.

[18]김두현, 마평수, 김채규, “”인터넷 정보가전용 멀티마디어 서비스 미들웨어 기술“, 한국정보처리학회지, 제 8 권 제 5 호, 2001 년 9 월, pp.53-59.