

멀티미디어 네스티드 세션 관리에서의 오류제어

고응남*, 황대준**

*천안대학교 정보통신학부

**성균관대학교 전기 전자 및 컴퓨터공학부

e-mail : ssken@cheonan.ac.kr

djhwang@yurim.skku.ac.kr

An Error Control on a Multimedia Nested Session Management

Eung-Nam Ko*, Dae-Joon Hwang**

*Division of Information & Communication, Cheonan University

**School of Electrical, Electronics & Computer Engineering, Sungkyunkwan University

요 약

본 논문에서는 멀티미디어 응용 개발 프레임워크에서 멀티미디어 응용 개발 프레임워크에서 네스티드 세션 관리에서의 오류 복구 방식에 대하여 기술한다. 네스티드 세션이란 하나의 부모 세션 아래에 여러 개의 자식 세션이 존재하는 것을 말한다. 제안하고자 하는 오류 제어는 멀티미디어 응용 개발 프레임워크에서 오류 발생시 오류를 하나의 메시지로 취급한다. 네스티드 세션에서 오류가 발견되면 복구는 스택을 사용하고 도미노 효과를 제거하기 위하여 네스티드 세션 및 검사점을 이용한 복구 알고리즘을 제안한다.

1. 서론

컴퓨터가 등장한 이후 여러가지 인간 활동을 지원하기 위해 컴퓨터를 응용하고자 하는 많은 노력이 있어 왔다. 지금까지 컴퓨터가 지원해 주었던 인간 활동들은 거의 모두가 한사람이 처리하던 일을 자동화한 것이었다. 그런데, 실제로 사람들이 직장이나 학교 등 다양한 조직체 안에서 수행하는 일들을 보면 여러 사람들이 함께 협동하여 작업하는 경우가 많다. 결국 사람들은 협동 작업을 컴퓨터를 이용하여 지원하고자 하는 필요를 느끼게 되었고 1980년 중반 이후로 멀티미디어 및 컴퓨터 지원 협동 작업(CSCW: Computer Supported Cooperative Work), 또는 그룹웨어(groupware)로 이름 붙여진 이 분야의 연구개발이 활발히 이루어져 왔다[1,2]. 멀티미디어 데이터는 동시에 여러 종류의 데이터를 전송해야 할 뿐만 아니라 미디어 데이터 사이의 관계성을 고려해야 하기 때문에 미디어 통신

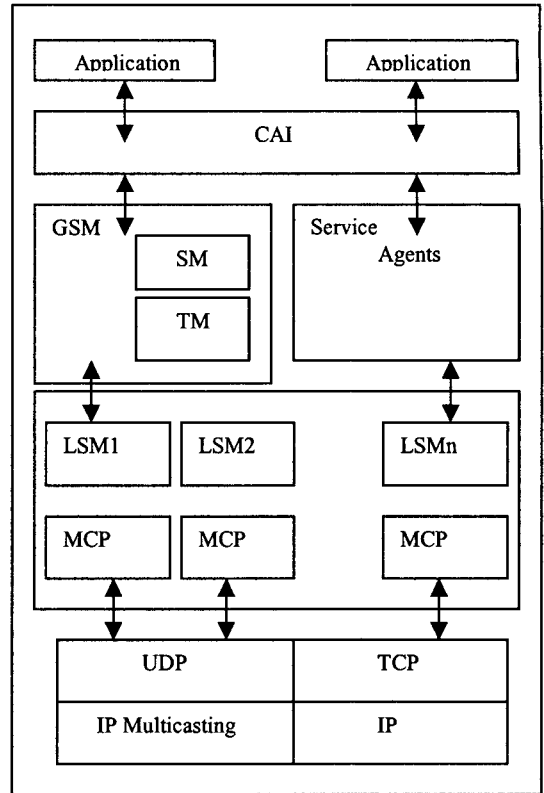
과 세션 관리 등이 필요하다[3]. 최근 들어 이러한 컴퓨터 협력작업 환경이 증가하고 있는데 반하여 이러한 시스템에서의 망 관리, 특히 세션 종료 등 응용 오류에 대한 연구는 미흡한 실정이다[4,5]. 특히 다중 세션이나 네스티드 세션에서의 오류에 대한 연구는 더욱 미흡한 실정이다.

본 논문에서 제안하는 것은 멀티미디어 응용 개발 프레임워크에서 네스티드 세션 관리에서의 오류 복구 방식에 대하여 기술한다. 네스티드 세션이란 하나의 부모 세션 아래에 여러 개의 자식 세션이 존재하는 것을 말한다. 오류가 발견되면 복구는 스택을 사용하고 도미노 효과를 제거하기 위하여 검사점을 이용한 복구 알고리즘을 제안한다. 본 논문의 구성은 2 장에서는 관련 연구, 3 장에서는 제안하는 멀티미디어 네스티드 세션관리에서의 오류 제어, 4 장에서는 평가 및 결론을 기술한다.

2. 관련 연구

소프트웨어 고장은 검사점(checkpoint) 설정, 재구성(reconfiguration) 및 복구 과정으로 처리되며, 재 수행량을 줄이며 로그 정보 분할에서도 일체된 상태를 유지시켜야 한다[6,7]. 검사점은 다른 노드 설정과 무관하게 설정하므로 도미노 효과를 유발한다. 특히, 롤백(rollback) 정보 전송시 무한정 그 실행을 반복하는 순환적 재시작 현상(livelock effect)이 발생할 수 있다[8].

기존의 분산 시스템에서 사용되는 검사점 설정 방법은 동기화된 방법과 비동기화된 검사점 설정 방법으로 구분된다. 기존의 동기적 또는 비동기적 검사점 설정 기법에는 정기적(periodic) 검사점 알고리즘이 기본적으로 사용된다. 이 방법은 각 프로세스가 일정한 시간 간격마다 체크포인트하는 방식이다. 기존의 동기적(synchronous) 검사점 설정 기법들은 각 사이트에서 검사점을 설정하는 시점에 다른 사이트의 검사점들과 완전 동기화를 이루도록 검사점을 설정한다. 따라서 고장이 발생한 경우에 빠른 시간에 일관성이 보장된 검사점을 찾아서 복구를 쉽게 할 수 있다. 그러나 이러한 방법은 검사점을 설정하는 시간이 많이 걸리고, 분산 트랜잭션들의 수행에 간섭 현상을 초래하게 되는 단점이 있다. 비동기적(asynchronous) 검사점 기법들은 검사점을 각 사이트에서 독립적으로 설정함으로써 쉽게 검사점을 설정할 수 있지만 고장이 발생하는 경우 일관성이 보장된 검사점을 찾아서 복구하는 작업을 어렵게 한다. 이러한 기법들은 최악의 경우 일관성이 보장된 검사점을 찾기 위해 시스템 초기 시점까지 복구 작업을 하는 도미노 효과를 발생하게 된다[9,10,11].



(그림 1) 멀티미디어 협력 작업 프레임워크

3. 멀티미디어 네스티드 세션 관리에서의 오류 제어

본 논문에서는 멀티미디어 응용 개발 프레임워크에서 네스티드 세션 관리를 위한 오류 복구 방식에 대하여 제안한다. 네스티드 세션이란 하나의 부모 세션 아래에 여러 개의 자식 세션이 존재하는 것을 말한다.

오류가 발견되면 복구는 스택을 사용하고 도미노 효과를 제거하기 위하여 검사점을 이용한 복구 알고리즘을 제안한다.

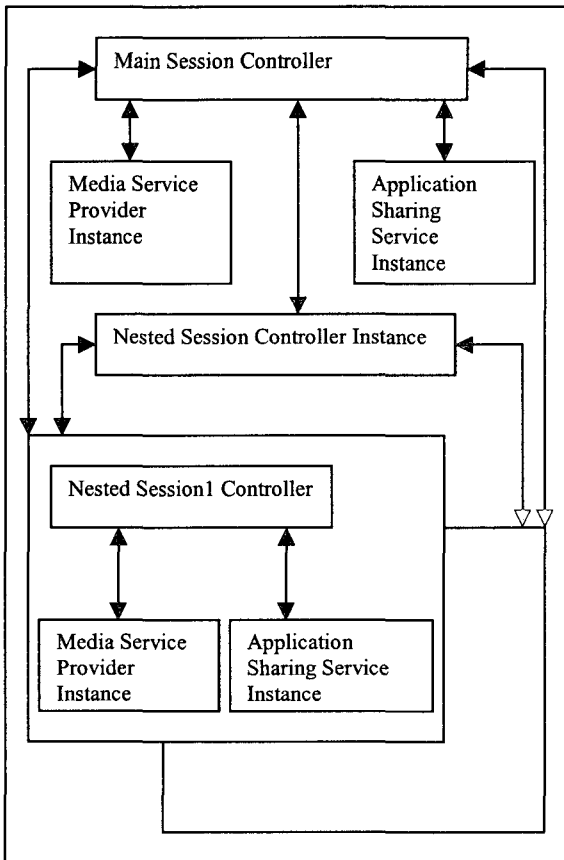
3.1 멀티미디어 협력 작업 프레임워크

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 연구의 제안 모델에서는 (그림 1)처럼 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 사용한다.

CAI 는 Common Application Interface, GSM 은 Global Session Manager, LSM 은 Local Session Manager, MCP 는 Multichannel Port, UDP 는 User Datagram protocol, IP 는 Internet Protocol 의 약어이다. 여기서 LSM 과 MCP 를 포함하는 계층은 DooRaeMTP 라 불리며 동시에 다수의 독립적인 세션을 개설할 수 있는 다중 세션을 지원하고 또한 하나의 세션은 미디어 데이터 별로 각각 한 개씩의 채널을 갖는 다채널 방식을 지원한다. 특히 오류 발생시 이것을 제어할 수 있는 채널도 할당한다. DooRaeMTP 는 다채널 방식의 그룹 통신을 지원하는 세션관리 프로토콜이다. 다채널 통신을 위해 응용 프로그램을 공통 응용 인터페이스(CAI: Common Application Interface)를 통하여 전체세션 관리기(GSM)에게 채널의 할당을 요청한다. 전체세션 관리기는 요청한 채널을 할당하고, 할당 받은 채널을 가지고 지역 세션 관리기를 생성한다. 지역세션 관리기(LSM)는 다른 참여자에게 공통 응용 인터페이스를 통하여 세션의 참여를 요청하고 할당 받은 채널 번호를 알려준다. 그리고 세션이 시작되면서부터 이 채널을 가지고 통신을 하게 된다. 이때 할당 받은 채널은 응용 프로그램이 자신이 사용할 미디어의 종류를 명시하여 요청한다. 다채널 포트(MCP)는 미디어별로 할당 받은 포트 번호를 가지고 참여자(참여자 관리기의 MCP)와 채널을 설정한다[12].

3.2 네스티드 세션

네스티드 세션이란 하나의 부모 세션 아래에 여러 개의 자식 세션이 존재하는 것을 말한다. 가장 실세계에 가까운 자연스런 세션의 형태가 네스티드 세션(Nested Session)이라고 할 수 있다. 네스티드 세션을 모델링하기 위해서는 한 세션에서의 다중 인스턴스의 허용과 네스티드 세션 간의 분리가 우선적으로 요구된다. 기본적으로 각 네스티드 세션은 하나의 미디어 서비스 인스턴스를 가진다. 즉, 각각의 네스티드 세션이 형성될 때마다 미디어 인스턴스 관리자를 생성하게 된다. 네스티드 세션 생성 후 자원 인스턴스와 세션 관리자와의 관계는 (그림 2)와 같다.



(그림 2) 네스티드 세션에서의 자원 인스턴스와 세션 관리자

네스티드 세션 관리자가 생성하는 미디어 서비스 인스턴스는 세션의 생성자에 의해 지정되지 않는 한 기본적으로 하나의 미디어 인스턴스 관리자만을 허용한다. 네스티드 세션간의 분리는 부모 세션 관리자에 의해 할당된 네스티드 세션 ID 로만 구분된다. 네스티드 세션의 데이터 채널은 부모와 동일한 채널을 사용하고 각 패킷의 헤더에 해당 네스티드 세션의 ID 를 넣

어 전송함으로써 데이터의 목적지를 구분하도록 한다. 네스티드 세션에의 참여는 세션 생성자의 권한으로 한정하고 있다. 세션 생성자는 네스티드 세션 제어기 인터페이스를 통해 네스티드 세션의 생성, 종료, 정보, 수정, 참여자 변경 등의 모든 작업을 수행한다. 세션 생성자가 네스티드 세션의 생성을 위해 네스티드 세션 ID 와 네스티드 세션 이름, 참여할 참여자, 네스티드 세션의 발언권 방식과 사용할 수 있는 자원을 한정할 수 있다. 네스티드 세션의 참여자들은 다른 네스티드 세션으로 이동하고 싶을 경우 세션 의장에게 요청한 후 세션 생성자의 승제로 이동할 수 있다. 네스티드 세션에 속한 참여자들은 네스티드 세션에서의 데이터와 세션 정보에만 접근할 수 있고 다른 네스티드 세션에 대한 접근은 일체 금지된다. 네스티드 세션에 속하지 않은 참여자들은 기존의 세션과 동일한 형태로 세션에 참여하게 되고 다른 네스티드 세션에 대한 정보나 접근을 할 수 없다.

3.3 네스티드 세션에서의 오류 제어

오류 복구는 오류 중에서 소프트웨어 오류인 경우에만 복구할 수 있다. 오류 복구는 먼저 검사점 설정(check point)을 한 후에 오류 감지 발생하면 그 검사점 까지 되돌아가서 재 수행하는 방식을 제안한다. 이때 프로세스들 간에는 메시지(message) 교환이 일어나는 경우도 발생한다. 메시지 교환 시스템에서는 도미노 효과(domino effect)가 발생할 수도 있다. 프로세서들 간의 정보 교환과 복구점이 조화되지 않으면 프로세스 사이에 계속적인 롤백(rollback) 전달의 사태가 일어나는 것을 도미노 효과라 한다. 본 논문에서는 멀티미디어 네스티드 세션에서의 도미노 효과를 제거할 수 있는 방법을 제안한다.

메시지는 헤더 정보와 데이터로 구분된다. 헤더 정보는 (그림 3)과 같다. 여기에서 n 은 네스티드 세션 ID 및 구분, w 는 검사점을 표시, x 는 통신의 종류(송신, 수신) 표시, y 는 프로세스들간의 관계 즉, 프로세스들간의 관계를 표시하는 플래그, z 는 주고 받은 프로세스들의 식별 번호 등을 표시하는 플래그, a 는 프로세스 간 주고 받은 누적 번호 기록, b 는 메시지의 종류(값 1-정상 데이터, 0-신호 데이터), 을 의미한다.

n	w	x	y	z	a	b
---	---	---	---	---	---	---

(그림 3) 메시지의 헤더 정보

스택은 검사점과 복구 두개의 스택을 갖는다. 프로세스간에 메시지를 받을 때 받는 즉시 검사점을 설정한 후에 오류제어 관리자에서 관리하고 있는 해당 프로세스의 검사점 스택에 저장한다. 즉, 모든 프로세스들이 메시지를 받으면서 그 프로세스에서 받은 내용을 검사점 스택에 저장한다. 이 고장 프로세스를 발견하면 그 프로세서가 실행하고 있을 때 그 프로세스에서 롤백(rollback)하여 재시작해야 한다. 그 네스티드

세션 및 검사점을 알아내기 위하여 스택에 있는 항목을 하나씩 읽어서 그 네스티드 세션 및 검사점이 나타날 때까지 읽는다. 이때 오류 프로세스는 네스티드 세션의 검사점을 발견할 때까지 검사점 스택에 있는 내용을 읽으면서 복구 스택에 저장한다. 즉, 검사점 스택의 일부 항목과 복구 스택의 항목은 역순이 된다. 그 다음에 복구 스택의 항목을 하나씩 읽어내면서 롤백 클래스가 어느 것인지를 결정한다. 검사점의 유형에 따른 복구 방법에 대한 개요는 롤백 클래스에서도 그 유형에 따라서 복구 알고리즘이 다르게 실행된다. 고장난 프로세스는 복구 신호를 준다. 다른 플랫폼에 있는 프로세스 간에는 메시지 교환이 발생하므로 도미노효과(Domino Effect)를 고려한다. 이를 제어 관리자에 저장되어 있는 log 파일의 내용을 읽어서 last in-first out 순서로 읽으면서 정보를 통하여 네스티드 세션 및 검사점을 알아낸다. 네스티드 세션 및 검사점을 알아내면 그 네스티드 세션 및 검사점에서부터 복구한다. 복구된 프로그램들은 네스티드 세션의 정상적인 참여자가 되고 메인 세션도 정상 세션으로 복원이 된다.

4. 평가 및 결론

본 연구에서는 멀티미디어 응용 개발 프레임워크에서 네스티드 세션 관리에서의 오류 복구 방식에 대하여 제안하였다. 네스티드 세션이란 하나의 부모 세션 아래에 여러 개의 자식 세션이 존재하는 것을 말한다. 본 연구의 특징은 기존 방법에 없는 멀티미디어 네스티드 세션에서의 오류 발생시 도미노 효과를 제거할 수 있는 방법을 스택을 사용하여 제안하였다. 스택은 검사점과 복구 두개의 스택을 이용하였다. 모든 프로세스들이 메시지를 받으면서 그 프로세스에서 받은 내용을 검사점 스택에 저장한다. 그 네스티드 세션 및 검사점을 알아내기 위하여 스택에 있는 항목을 하나씩 읽어서 그 네스티드 세션 및 검사점이 나타날 때까지 읽고 오류 프로세스는 네스티드 세션의 검사점을 발견할 때까지 검사점 스택에 있는 내용을 읽으면서 복구 스택에 저장한다. 네스티드 세션 및 검사점을 알아내면 그 네스티드 세션 및 검사점에서부터 복구한다. 복구된 프로그램들은 네스티드 세션의 정상적인 참여자가 되고 메인 세션도 정상 세션으로 복원이 된다.

앞으로 연구가 필요한 분야는 네스티드 세션의 수 만큼 네트워크 트래픽이 증가하므로 이것을 줄일 수 있는 방안에 대한 것 등이다. 또한 스택의 크기 및 메시지 헤더 크기를 줄이는 방법도 중요한 연구 분야의 하나이다.

참고문헌

- [1] 김문석, 성미영, "동기적 웹브라우저 공유를 지원하는 협동 작업 시스템", 한국정보처리학회논문지 B 제 8-B권 제 3호 pp.283-288, 2001년 6월.
- [2] 이재호, "협력작업을 위한 에이전트 기반 소프트웨어", 한국정보처리학회 논문지 제 16 권 제 7 호, pp.24-30, 1998.
- [3] W.R. Stevens, TCP/IP Illustrated, Volume I, II, Addison Wesley Pub., 1993.
- [4] 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.
- [5] 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.
- [6] S.S. Yau and R.C. Cheung, "Design of Self-Checking Software", in Proc. 1975 Int.Conf.Reliable Software, pp.450-457, April 1975.
- [7] J.J Horing, H.C. Lauer, P.M. Mellar-Smith and B. Randell, "A Program Structure for Error Detection and Recovery", Lecture Notes in Computer Science 16, Berlin: Springer-Verlag, pp.171-187, 1974.
- [8] R. Koo, & S. Toueg, "Checkpointing and Rollback recovery for Distributed Systems", IEEE Trans. Software Eng., Vol.SE-13, No.1, pp.23-31, 1987.
- [9] 박윤용, 권성익, 조수현, "분산 트랜잭션 처리시스템 서 2-단계 확인 프로토콜을 근거로 하는 검사점 설정 및 오류 복구 알고리즘", 한국정보처리학회 논문지 제 3권 제 2호, pp.327- 338, 1996.3.
- [10] Hong Va Leong, Divyakant Agrawal, "Using Message semantics to Reduce Rollback in Optimistic message Logging Recovery Schemes", 1063-6927/94, 1994 IEEE, pp.227- 234, 1994.
- [11] Dhiraj K. Pradhan, Nitin H. Vaidya, "Roll-Forward and Rollback Recovery: Performance Reliability Trade-off", 0363-8928/94, 1994 IEEE, pp.186-195, 1994.
- [12] 박길철, 황대준, "네트워크 환경에서 멀티미디어 객체 동기화 모델 설계", 한국정보처리학회, 제 1 권 2 호, 1994년 8월, pp, 568-571.