

협동적 멀티미디어 응용을 위한 프로토콜의 설계 (Design of Protocol for Collaborative Multimedia Applications)

황 인 준[†]
(Eenjun Hwang)

요약 멀티미디어 프리젠테이션 응용은 대개 여러 개의 협동 서버(collaborating server)로부터 데이터의 검색을 필요로 하며, 검색된 데이터의 프리젠테이션은 상호 의존적이다. 이것은 프리젠테이션을 위해서 다른 협동 서버에 저장되어 있는 데이터들이 주어진 시간적 계약 조건에 따라 연속적으로 검색되어 전달되어야 하기 때문이다. 이러한 응용에서 서버는 먼저 이용 가능한 네트워크 대역폭과 시스템 자원을 평가하여 초기 검색 스케줄을 생성하고 협동 서버는 시스템 부하에 따라 검색 스케줄에 적절한 변경을 제안할 수 있어야 한다. 전체 스케줄에서 이러한 부분적 변경은 연차적으로 다른 협동 서버의 검색 스케줄에 영향을 미친다. 따라서, 멀티미디어 프리젠테이션을 구성하는 객체에 대한 검색 스케줄을 확정하기 위해서는 이러한 일련의 협상이 협동 서버 사이에 진행되어야 한다. 본 논문에서는, 협동적 멀티미디어 프리젠테이션을 위한 자원의 협상과 승인을 다루기 위해 RLCP(Resource Lock Commit Protocol)라는 프로토콜을 제안하고 분산 비디오 프리젠테이션 응용에 적용해 본다.

키워드 : 분산 멀티미디어 프리젠테이션, 협동 서버, 검색 스케줄

Abstract Many multimedia presentation applications involve retrieval of objects from more than one collaborating server. Presentations of objects from different collaborating servers might be interdependent. This implies that objects should be retrieved from remote servers and delivered continuously according to given time constraints. Such applications need an estimate of the available network resources to each of the collaborating servers and local system resources in order to identify a schedule for retrieving the objects composing the presentation. A collaborating server can suggest modifications of the retrieval schedule depending on its load. These modifications can potentially affect the retrieval schedule for other collaborating applications. Hence, a sequence of negotiations has to be carried out with the collaborating servers in order to commit for a retrieval schedule of the objects composing the presentation. In this paper, we propose an application sub-layer protocol, RLCP(Resource Lock Commit Protocol), for handling the negotiation and commitment of the resources required for a collaborative multimedia presentation and apply it to distributed video presentation application.

Key words : Distributed Multimedia Presentation, Retrieval Schedule, Collaborating Server

1. 서 론

협동 멀티미디어 프리젠테이션 응용은 분산되어 있는 다수의 협동 서버로부터 객체의 검색을 필요로 한다. 예를 들면, 문서를 구성하고 있는 객체들이 여러 개의 서버[7]에 분산되어 있는 분산 멀티미디어 문서 프리젠테이션이나 비디오 블록들이 여러 개의 서버에 분산되어 있는 분산 비디오 프리젠테이션이 여기에 해당된다고

볼 수 있다. 이러한 응용에서 서버에서의 객체 검색은 상호 의존적이다. 다시 말해서, 각 객체의 프리젠테이션 시간은 다른 객체들의 프리젠테이션 시간에 따라 상대적으로 결정된다. 많은 경우 객체들이 여러 협동 서버에 중복되어 저장되어 있을 수가 있다. 이런 경우, 응용은 가용 네트워크 자원과 협동 서버의 부하를 고려하여 실제 객체를 검색할 특정 서버를 선택할 수 있다. 따라서, 객체에 대한 검색 스케줄(retrieval schedule)은 객체간의 상호 관계뿐만 아니라 필요한 네트워크 자원이나 시스템 자원 등을 고려하여 결정해야 한다.

본 논문에서는 프리젠테이션에 필요한 시스템이나 네트워크 자원의 절충(negotiations)을 다루기 위한 RLCP

• 본 연구는 아주대학교 정착 연구 지원에 의한 것임.

[†] 정 회 원 : 아주대학교 정보통신전분대학원 교수
ehwang@madang.ajou.ac.kr

논문접수 : 2001년 5월 11일
신사와료 : 2001년 11월 12일

라는 용용 서브계층 프로토콜을 제안한다. RLCP는 협동을 필요로 하는 용용이 효과적으로 수행될 수 있게 융통성 있는 프리젠테이션 스케줄링을 지원한다. 특히, RLCP는 검색 스케줄의 보장을 위해 네트워크 서비스 제공자(Network Service Provider)가 미리 자원 예약(advanced resource reservation) 기능을 제공한다고 가정한다.

RLCP의 특징은 협동 서버에서 필요로 하는 네트워크나 시스템 자원들에 대한 협상을 단계적(phased) 방식으로 수행한다는 것이다. 그 이유는 실제 사용 네트워크 자원을 알아야 객체가 검색 스케줄을 만들 수 있기 때문이다. 객체 검색을 요청 받은 서버는 실제 객체가 검색되어야 하는 시점이 정해져야 그 시점에서 사용 시스템 자원들을 결정할 수 있다. 그러므로, RLCP는 첫 단계에서 사용 네트워크 자원을 확인하고, 다음 단계에서 서로 다른 협동 서버에서 검색될 객체들을 확정하고 객체 검색에 필요한 시스템 자원을 협상하게 된다. 하나의 프리젠테이션을 위해서는 연관된 모든 협동 서버와 시스템과 네트워크 자원에 대한 이러한 두 단계 협상을 수행한다. 만약 협동 서버와의 협상에서 지연이 발생하면, RLCP는 다른 협동 서버와의 스케줄을 수정해야 할 필요가 발생하는 데 이를 위해, RLCP는 자원의 승인(commitment)과 수정(modification)이라는 세 번째 단계를 실행한다.

이 논문의 구성을 다음과 같다. 2장에서는 RLCP가 필요한 협동 멀티미디어 응용의 특징에 대해 살펴보고 필요한 시스템 자원들 간에 존재하는 의존성에 대해 알아본다. 3장에서는 RLCP에 대해 설명하고 그 특징과 제공하는 서비스에 대해 서술하고 4장에서 RLCP가 사용될 수 있는 응용의 예를 살펴본다. 5장에서 RLCP의 구현과 적용에 대해 언급하고 6장에서 RLCP와 다른 응용 서브 프로토콜들의 특징을 비교한다.

2. 문제의 정의

협동 멀티미디어 프리젠테이션 응용을 효과적으로 지원하기 위해서는 여러 가지 제약 조건들이 고려되어야 한다. 특히, 프리젠테이션을 구성하는 객체의 시간적 제약 조건과 프리젠테이션에 필요한 시스템 및 네트워크 자원의 가용성 등은 반드시 고려되어야 한다. 이 장에서는 협동 멀티미디어 프리젠테이션 응용의 예로써 분산 멀티미디어 문서 프리젠테이션과 분산 비디오 프리젠테이션 응용을 살펴보고 그러한 응용을 효과적으로 지원하기 위해 고려해야 될 사항들을 고찰해본다.

2.1 분산 멀티미디어 문서 프리젠테이션

그림 1은 멀티미디어 문서를 구성하는 객체가 CS로 표시된 네 개의 협동 서버에 분산되어 있는 경우로써 문서를 구성하는 객체와 그들 간의 프리젠테이션 정보를 보여준다. 구성 객체 중에서 O2는 O2'나 O2''를 변환하여 생성할 수 있으나 이때 추가적인 CPU 처리 시간이 필요하다. 문서의 프리젠테이션과 관련하여 다음과 같은 제약조건을 가정해 보자.

- 문서 프리젠테이션은 명시된 구간(interval) 어디에 서나 시작할 수 있다. 예를 들어, O1은 $[t_{01} - \delta t, t_{01} + \delta t]$ 의 어느 시점에 표시될 수 있다.
- 객체 O1과 O2는 동시에 보여져야 한다. 마찬가지로, 객체 O3과 O4는 동시에 시작되어야 한다.
- 객체 O3과 O4는 객체 O1과 O2의 프리젠테이션이 끝나고 바로 시작되어야 한다.

그림 1에서 서버 S가 프리젠테이션을 수행하기 위해서는 필요한 객체를 협동 서버로부터 전송 받아야 한다. 이를 위해서는 관련된 협동 서버가 동의하는 객체를 검색할 정확한 시간을 담고 있는 검색 스케줄이 필요하다. 만약 어떠한 협동 서버가 검색 스케줄에 동의하지 못할 경우에는 스케줄을 수정하여야 한다. 예를 들어, CS2가 객체 O2의 전송을 Δt 만큼 늦추어야 하는 경우를 생각해 보자. 만약 서버 S가 검색된 객체를 Δt 만큼 베티킹할 수 없다면 O1의 검색도 같은 시간 동안 지연시키거나 아니면 O2'나 O2''를 변환시켜 O2를 얻어야 한다. 다시 말해서, CS1이나 CS4가 O2의 형태로 객체를 변환하는 기능을 지원한다면, S는 이를에게 필요한 객체의 전송을 요구할 수 있다. 객체 O3와 O4의 검색은 O1과 O2의 프리젠테이션에 종속되어 있으므로 앞의 객체들이 지연되면 이 객체들의 검색 또한 지연되어야 한다. 결과적으로, 이것은 멀티미디어 문서를 구성하고 있는 구성 객체들의 검색 스케줄은 상호 의존적임을 보여준다.

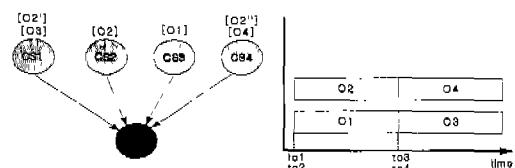


그림 1 분산 멀티미디어 문서의 프리젠테이션

2.2 분산 비디오 프리젠테이션

분산 비디오 프리젠테이션은 하나의 프리젠테이션을 구성하는 전체 비디오 데이터가 하나의 서버에 저장되

어 있지 않고 여러 개의 세그먼트로 나뉘어 네트워크상의 여러 VoD 서버에 중복되어 저장되어 있는 경우이다. 만약 요청된 비디오를 구성하는 세그먼트가 그 서버에 존재하지 않을 경우 서버는 다른 VoD 서버에 그 세그먼트를 얻기 위한 요청을 하게 된다.

그림 2에서 고객 C1이 서버 S에 접속하여 비디오 프리젠테이션을 요청하고 있으며 프리젠테이션에 필요한 25개의 비디오 블록들은 S를 포함한 여러 개의 협동 서버에 중복되어 분산되어 있다. 서버 S는 b1에서 b5까지의 블록을 제외한 나머지 b6에서 b25까지의 블록을 다른 협동 서버에서 전송 받아야 하며, 이를 위해 협동 서버가 동의하는 블록에 대한 검색 스케줄을 만들어야 한다. 검색 스케줄에 대한 협동 서버의 동의를 구하는 것은 프리젠테이션이 일단 시작되면 중간에 끊어짐이 없이 계속 진행될 수 있게 필요한 자원에 대한 관리를 요청하는 것이다.

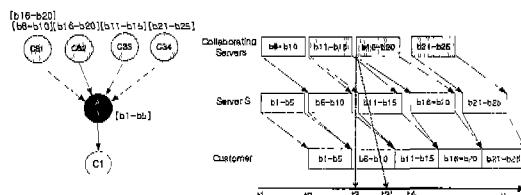


그림 2 분산 비디오 프리젠테이션

협동 서버에서 필요한 세그먼트를 전송 받기 위해서서버 S는 그 세그먼트가 필요한 시간을 명시해야 한다. 만약 협동 서버가 요구된 시간에 전송을 승인할 수 없는 경우, 전송 시간을 변경하든가 아니면 같은 블록을 가지는 다른 협동 서버를 시도하게 된다. 예를 들어, 그림 2에서 CS2가 세그먼트 b16-b20의 전송을 요청된 시간 t3'에는 승인할 수 없으나 t3'에는 승인할 수 있는 경우, 서버 S는 다음과 같은 두 가지 선택이 있다.

- 서버 CS1에 세그먼트 b16-b20의 전송을 요청한다. 만약 t3'에 요청이 승인되면 서버 S는 현재의 검색 스케줄을 계속해 나갈 수 있다.
 - 그렇지 않다면, S는 CS1뿐만 아니라 CS2에서 세그먼트 b16-b20의 전송을 위해 필요한 최소한의 지연을 알아내야 한다. t3'이 세그먼트가 전송 가능한 가장 이른 시간이라고 하면 결국 $\delta t (=t3' - t3)$ 만큼의 프리젠테이션 지연이 발생하게 된다. 이 경우에, S는 이전 세그먼트에 대해 스케줄을 다시 수행해야 한다.
- 분산 비디오 프리젠테이션의 경우에도 검색된 객체들 사이에 존재하는 상호 종속성을 확인할 수 있다.

2.3 시스템 자원들 간의 의존성

앞에서 열거한 응용 시나리오는 검색 스케줄과 네트워크와 시스템 자원들 사이에 다음과 같은 종속 관계가 존재함을 보여 주고 있다.

- 멀티미디어 프리젠테이션을 구성하고 있는 객체에 대한 검색 스케줄을 생성하려면 협동 서버에 이르는 네트워크의 가용 대역폭을 알아야 한다.
- 서로 다른 협동 서버에서의 객체 검색은 상호 연관성이 있으며, 검색 스케줄은 참여하는 모든 협동 서버에 의한 동의가 필요하다. 만약 협동 서버가 구성된 스케줄에 동의할 수 없는 경우에는 검색 스케줄은 수정되어야 한다.
- 멀티미디어 프리젠테이션을 구성하고 있는 객체들이 협동 서버에 의한 별도의 처리 과정을 거쳐야 할 경우도 있다. 예를 들면, 다른 비디오 포맷으로의 변환이 필요한 경우이다. 이런 경우, 검색 스케줄이 확정되어야 협동 서버가 객체를 처리하는 데 필요한 자원들의 가용성 여부를 결정할 수 있다.
- 프리젠테이션을 구성하는 객체가 중복되어 있는 경우에 객체가 검색될 서버를 선택할 수 있다. 따라서, 어느 한 협동 서버가 객체 검색 요청을 거부하는 경우, 응용은 같은 객체를 가지고 있는 다른 서버를 시도할 수 있다.

이러한 응용은 여러 가지 제약 조건들을 가진다. 사용자 C_i 가 요청한 비디오 프리젠테이션 $pres(C_i)$ 는 비디오 세그먼트들로 구성이 되며 비디오 세그먼트는 다시 (v, b_i, b_j) 로 나타낼 수 있다. 여기서, v 는 비디오 식별자이며 b_i 와 b_j 는 세그먼트의 시작과 끝 블록을 나타낸다. 어느 한 시점에서 초당 S1개의 데이터 블록을 읽을 수 있는 협동 서버에 n 개의 사용자 요청 $pres(C_1), \dots, pres(C_n)$ 이 주어져 있다고 가정하자. 시간 t 에 사용자 C_i 를 위해 읽혀져야 하는 데이터 블록을 $Read(C_i, t)$ 라고 하면, 사용자 요청에 대한 제약 조건들을 정의할 수 있다. 제약 조건에는 네트워크나 디스크 대역폭과 같은 자원에 관한 제약 조건과 프리젠테이션의 연속성과 같은 응용에 내포된 제약 조건 등이 있다. 제약 조건들을 정의하기 위해 사용자 C_i 가 요청한 프리젠테이션에 대해 다음을 정의할 수 있다.

$$\begin{aligned} st(C_i) &= \min\{ t \mid Read(C_i, t) \neq 0 \} \\ et(C_i) &= \max\{ t \mid Read(C_i, t) \neq 0 \} \\ pt(C_i, v, b) &= \min\{ \exists t \mid ((v, r, s) \\ &\quad \in Read(C_i, t)) \quad r \leq b \leq s \} \end{aligned}$$

여기서, $st(C_i)$ 는 C_i 의 요청에 대해 처음으로 결과가 나타나기 시작하는 시간을 나타내며, C_i 의 요청에 대한

대기 시간을 나타낸다. $et(C_i)$ 는 C_i 의 요청에 대한 마지막 결과가 보이는 시간을 말한다. $pt(C_i, v, b)$ 는 클라이언트 C_i 가 신청한 비디오 v 의 블록 b 가 저장 장치에서 읽혀지는 시간을 의미한다.

- **프리젠테이션의 연속성** : $st(C) \leq t \leq et(C)$ 인 모든 시간 동안에 사용자 C_i 에 대해 일단 프리젠테이션이 시작되면 끝날 때까지 중간에 끊어짐이 없어야 한다.
- **대역폭 제약 조건**: 대역폭 제약 조건은 저장 서버가 단위 시간에 최대 S 개의 데이터 블록을 읽을 수 있다는 것을 나타낸다. 따라서, 대역폭 제약 조건은 임의의 시점 t 에서 모든 프리젠테이션에 필요한 대역폭의 합이 저장 서버의 최대 대역폭 보다는 작아야 함을 규정한다.

$$\sum_{client C} Read(C, t) \leq S$$

- **소비율 제약 조건**: 사용자의 환경에 따른 서로 다른 소비율을 고려할 필요가 있다. 만약 사용자 C_i 가 단위 시간에 w_i 개의 블록을 소비할 수 있는 환경일 때, 소비율 제약 조건은 다음과 같이 표현된다.

$$(\forall t) st(C_i) \leq t \leq et(C_i) \rightarrow card((m, b) | (m, h, r) \in Read(C_i, t) \& r \leq h \leq s) = w_i$$

- **대기 제약 조건(응답 시간 제약 조건)**: 사용자 C_i 에 대한 대기 제약 조건은 다음과 같이 표현될 수 있다.

$$st(C_i) \leq t_s$$

이와 같은 대기 시간은 사용자가 직접 명시할 수도 있고 실제 응용에서 사용자에게 제시하는 조건의 하나가 될 수도 있다.

이 외에도 응용이 검색 스케줄을 결정하기 위해서는 프리젠테이션 시작 시간이나 기간(duration)과 같은 객체 프리젠테이션 명세들과 대역폭, 지연(delay), delay jitter와 같은 가용한 네트워크 QoS 등이 고려되어야 한다.

3. Resource Lock Commit Protocol

본 장에서는 앞에서 살펴본 분산 멀티미디어 문서나 분산 비디오 프리젠테이션과 같은 응용에서 검색 스케줄 생성에 사용될 수 있는 3단계 응용 프로토콜인 RLCP에 대해 설명한다. RLCP는 기본적으로 수신자 시작(receiver-initiated) 프로토콜이며 스케줄의 보장을 위해 네트워크 서비스 제공자가 제공하는 사전 자원 예약(advanced resource reservation) 기능을 가정한다. RLCP는 네트워크 자원 협상 단계, 시스템 자원 협상 단계, 그리고 승인 및 수정 단계를 거쳐 필요한 검색 스케줄을 확정한다.

여기서 자원에 대한 lock은 요청된 자원에 대한 임시 예약(reservation)을 의미하고 승인은 요청된 시간에 자원을 할당하는 데 동의함을 의미한다. 만약 요청한 응용이 미리 명시된 기간 내에 승인을 얻지 못하면 lock된 자원은 다른 요청을 위해 사용 자원으로 간주된다. 그리고 요청된 자원에 대한 수정은 자원의 양이나 필요한 시간에 대한 변경을 의미한다.

3.1 RLCP의 단계들

멀티미디어 프리젠테이션에 필요한 서버와 네트워크 자원에 대한 협상은 세 단계로 진행되며 그 구체적인 내용은 다음과 같다.

(a) **네트워크 자원 협상 단계**: RLCP는 협동 서버와의 통신을 위해 네트워크 서비스 제공자로 하여금 사용 네트워크 자원을 파악하도록 요청한다. RLCP는 [18]에서 언급된 것과 같은 heuristic 알고리즘을 이용하여 필요한 네트워크 자원을 추정할 수 있다. 네트워크 자원은 또한 협동 서버와의 협상 동안 잠시 끊어들 수 있다. 이 단계에서 연결 요청이 수용되는 경우에 이루어지는 메시지 교환은 그림 3에 나타나고 있다.

(b) **시스템 자원 협상 단계**: 두 번째 단계에서는 RLCP가 가용 네트워크 자원을 기반으로 하여 검색 스케줄을 생성한다. 이를 위해서는 필요한 객체가 저장되어 있는 협동 서버와 객체가 전송되어야 할 시간 등이 정해져야 한다. 멀티미디어 프리젠테이션 응용을 위한 검색 스케줄 생성 알고리즘은 여러 논문에서 다루고 있다[2, 7, 8, 12]. 이러한 알고리즘들은 이용 가능한 네트워크 자원을 기반으로 검색 스케줄을 계산하기 위해 RLCP에 적절하게 활용될 수 있다. 그림 4는 시스템 자원 협상을 위해 이루어지는 절차를 보여준다.

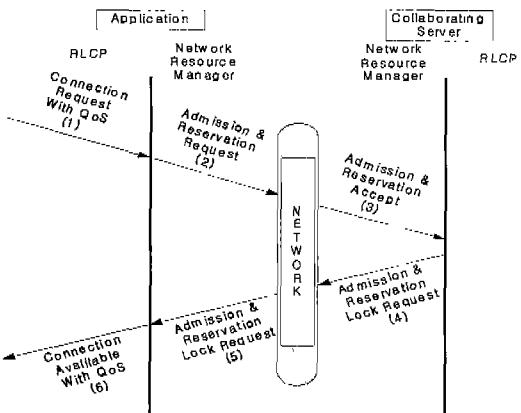


그림 3 RLCP의 첫 단계: 네트워크 자원 협상

(c) 승인 및 수정 단계: 만약 모든 협동 서버가 객체에 필요한 자원을 지원하는 데 동의하면, RLCP는 둘 중 두 자원들을 사용하도록 승인한다. 협동 서버 중 일부가 요청을 거절한다면 변경을 제안하는 경우, RLCP는 협동 서버의 일부에 대해 요청된 자원을 수정해야 할 경우도 있다. 이러한 수정은 검색 스케줄링 알고리즘[7]을 이용하여 결정할 수 있으며 그 수정은 다음과 같은 것이 된다.

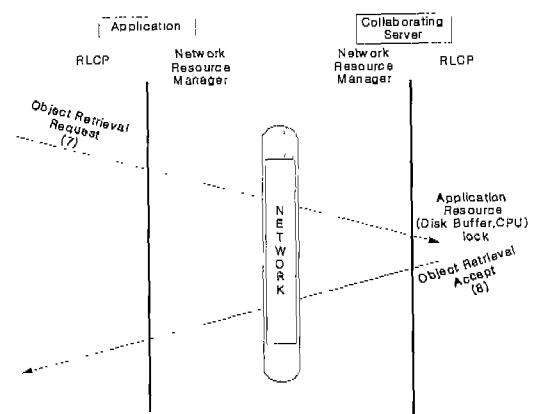


그림 4 RLCP의 둘째 단계: 시스템 자원 협상

- 만약 객체가 중복되어 있다면, 객체를 가진 다른 협동 서버에 요청을 보낸다.
- 협동 서버의 부하가 덜 걸리는 시점으로 객체 검색을 연기한다.

수정을 다루기 위해서, RLCP는 그림 5에 나타난 세 번째 단계를 수행한다. 이 단계에서 프리젠테이션 응용은 협동 서버의 반응을 검사한다. 만약 모든 서버가 요청된 스케줄을 승인한다면, RLCP는 모든 협동 서버에 자원 승인 요청을 보낸다(events 9~12). 만약 어떤 서버가 요청된 스케줄을 승인할 수 없다면, 프리젠테이션 응용은 modify object retrieval request 메시지를 보내서 이제까지의 검색 스케줄에 필요한 수정이 네트워크 서비스 제공자와 다른 협동 서버에까지 반영되게 한다.

검색하고자 하는 객체가 여러 협동 서버에 중복되어 있는 경우 그러한 객체에 대한 검색은 다음과 같이 다를 수가 있다.

- 모든 협동 서버 사이의 가용 네트워크 자원을 평가하여 프리젠테이션 스케줄에 적당한 서버를 선택한다. 이 방식은 어떤 객체에 대한 최상의 서버를 선택할 수 있다는 장점이 있으나 네트워크 트래픽의 증가를 유발할 수 있다.

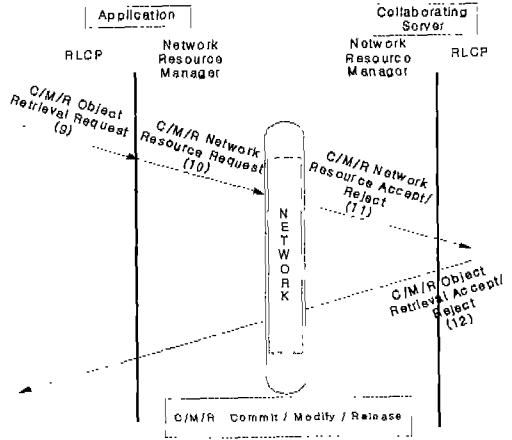


그림 5 RLCP의 세 번째 단계: 승인 및 수정

- Heuristics을 사용하여 여러 서버 중에서 하나를 선택하고 가용 네트워크 자원에 대해 평가를 한다. 이 방식은 네트워크 트래픽을 최소화시키지만 선택된 서버가 최적이 아닐 수도 있다.

3.2 RLCP의 FSM(Finite State Machine)

그림 6은 멀티미디어 프리젠테이션을 시작한 RLCP와 연관된 Finite State Machine을 보여준다. 멀티미디어 프리젠테이션이 수행될 때 RLCP는 필요한 네트워크 자원에 대한 초기치를 설정하고 네트워크 처리량을 식별하는 상태로 전이한다. 요청한 협동 서버에 가용 QoS를 구한 후 검색 스케줄을 결정하고 객체 검색 요청을 하게 된다. 만약 모든 협동 서버가 검색 요청에 동의한다면, 멀티미디어 프리젠테이션이 시작된다(Retrieve and Present Objects 상태로 표시). 그렇지 않다면, RLCP는 객체 검색 스케줄을 수정하는 상태로 전이한다.

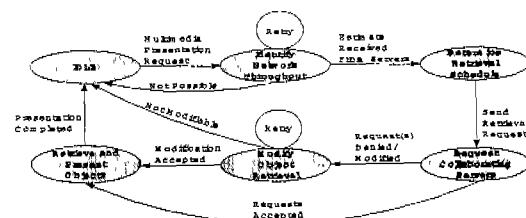


그림 6 RLCP Initiator Finite State Machine

그림 7에서는 협동 서버의 RLCP와 연관된 FSM을 보여 주고 있다. 만약 객체 검색 요청이 승인할 수 있다면, 협동 RLCP는 Resource Locked 상태로 전이한다.

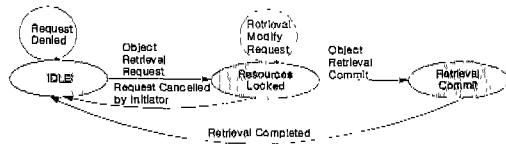


그림 7 RLCP Collaborator Finite State Machine

객체 검색 스케줄에 대한 수정 요청에 대해서는 거부나 승인 메시지를 initiator에게 보내고 같은 상태에 머무른다. Initiator가 검색 스케줄을 승인했을 때, 협동 서버는 검색 승인 상태로 전이한다. 객체 검색이 끝난 후, 협동 RLCP는 Idle 상태로 전이한다.

RLCP는 구조상으로 그림 8에 보이는 것처럼 응용 협동 멀티미디어 응용과 네트워크 서비스 제공자 사이에 위치한다.

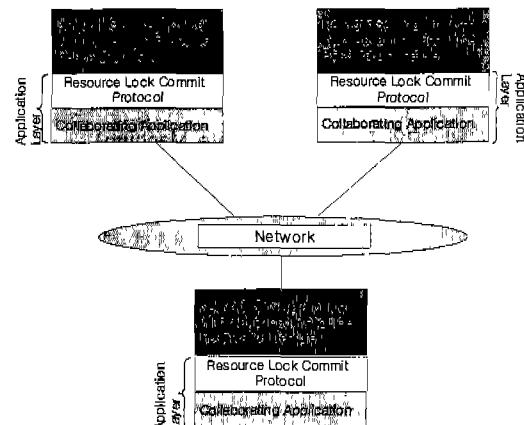


그림 8 RLCP의 계층적 위치

3.3 RLCP 서비스

RLCP는 멀티미디어 프리젠테이션을 구성하는 객체들과 시간적 제약 관계가 주어지면 필요한 자원의 예약을 위해 NSP나 디스크 매니저, 버퍼 매니저와 상호 작용을 한다.

표 1은 RLCP가 멀티미디어 프리젠테이션 응용에 제공하는 서비스를 요청과 응답 항목으로 보여준다. 여기에서 응용은 R_Initiate Presentation 요청을 통해 멀티미디어 프리젠테이션을 시작한다. RLCP는 객체가 검색될 시간과 서버를 인식하기 위해 [2, 7, 8, 12]에서 제안하고 있는 검색 스케줄 생성 알고리즘을 사용한다. 만약 프리젠테이션이 지연되거나 시작될 수 없다면, RLCP는 R_DelayStart 또는 R_RejectPresentation 메

세지로 응답을 한다. 응용은 프리젠테이션을 초기화할 때 자연의 거부나 수용을 응용에 전달하기 위해 R_PresentationDelay 요청을 사용한다.

표 1 멀티미디어 프리젠테이션 응용에 제공되는 RLCP 서비스들

	Service	Parameters
Requests	R_Initiate Presentation Presentation Configuration	R_PresentationDelay Accept/Reject
Responses	R_Reject Presentation Reason	R_DelayStart Time

표 2에서는 협동 멀티미디어 프리젠테이션을 위해 RLCP가 협동 RLCP에게 제공하는 서비스를 보여준다. RLCP의 구동을 위해서는 R_SendObject나 R_Modify ObjectSendTime과 같은 서비스를 통해 협동 RLCP에게 객체 검색 요청을 보낸다. 요청에 대해 협동 RLCP는 소요되는 디스크 대역폭이나 버퍼 공간을 확보하기 위해 디스크 매니저나 버퍼 매니저와 접촉한 다음 R_SendObjectResponse나 R_ModifySendTimeResponse와 같은 메시지로 응답한다. 마지막으로 네트워크 서비스 제공자와 RLCP 사이에는 다음과 같은 항목으로 상호 작용을 한다.

- 명시된 시간과 기간 동안 요청된 QoS를 충족시키는 네트워크 연결에 대한 요청
- 네트워크 연결 시간과 기간에 대한 수정
- 네트워크 서비스 제공자로부터 요청한 자원들을 대한 사용 확약

표 2 협동 RLCP와의 상호 작용을 위한 서비스들

	Service	Parameters
Requests	R_SendObject ObjectId, TimeRequired	R_ModifyObjectSendTime ObjectId, TimeRequired
Responses	R_SendObjectResponse Accept, Reject, Delay	R_ModifySendTimeResponse Accept, Reject, Delay

4. RLCP 사용 예

앞에서 설명한 분산 비디오 프리젠테이션의 경우를 생각해보자. 서버 S는 사용자 C1에 대해 시간 t_0 에 시작하는 프리젠테이션을 위한 검색 스케줄을 생성하려고 하며, 협동 VoD 서버들에게서 비디오 블록 b6-b25를 요청하려 한다. 그림 9는 첫 단계에서 서버 S가 NSP에게 가용 네트워크 자원에 대한 예상치를 요청하는 과정

이다. 이 예상치를 기반으로 S는 아래에 나타난 것과 같은 최초의 검색 스케줄을 생성한다고 가정해 보자.

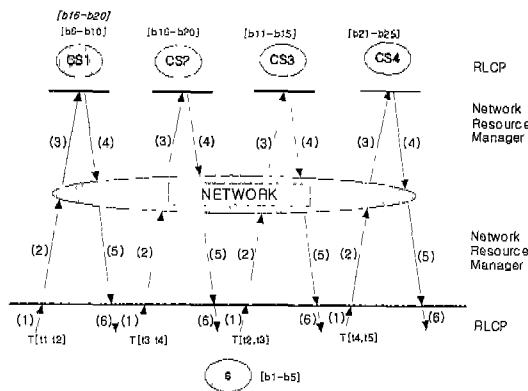


그림 9 협동 서버에 네트워크 자원 요청 과정

- 블록 b1-b5는 로컬에서 검색 가능하다. 영화 프리젠테이션은 시간 t_0 에서 시작되며, 상영 시간을 바탕으로 다른 블록들에 대한 검색 스케줄을 다음과 같이 결정한다.
- 블록 b6-b10은 시간 t_1 에 CS1에 의해 전송된다.
- 블록 b11-b15은 시간 t_2 에 CS3에 의해 전송된다.
- 블록 b16-b20은 시간 t_3 에 CS2에 의해 전송된다.
- 블록 b21-b25은 시간 t_4 에 CS4에 의해 전송된다.

그림 10에서는 서버 S가 위의 초기 스케줄을 기반으로 하여 협동 서버에 객체 검색 요청을 하는 과정을 보이고 있다. 만약 모든 협동 서버가 객체 검색을 위해 필요한 자원들을 할당할 수 있다면 S에게 객체 검색을 승인 메시지를 보낸다. 비디오 프리젠테이션을 위해 임시로 락어둔(locked) 네트워크 자원들도 할당한다.

만약 CS2가 시간 t_3 에 블록 b16-b20를 승인할 수 없고 대신에 CS2는 t_3 에 블록을 전송할 수 있다고 가정해 보자. 그러면 S는 충복된 블록을 가지고 있는 CS1이 t_3 에 블록을 제공할 수 있는지를 알아보려 할 수 있다. 만약 CS1이 t_3 에 전송을 승인한다면, S는 프리젠테이션을 위한 검색 스케줄을 확정할 수 있으나, 그렇지 않다면 S는 $t=(t_3, t_4)$ 만큼 프리젠테이션을 지연시켜야 한다. 이것은 이전 블록에 대한 전송 승인도 t 만큼 지연시켜야 함을 뜻한다. 이런 경우에, 그림 11에 나타나 있듯이, S는 RLCP의 세 번째 단계를 수행하여 발생한 지연을 반영한 검색 변경을 요청하고 네트워크 자원에 대해서도 필요한 변경을 계산하여 요청한다. 그림에서는 자원 변경 요청에 대한 승인 과정을 보여주고 있다.

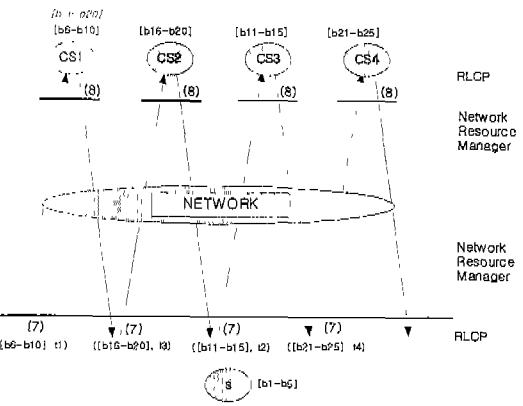


그림 10 객체 검색 요청 과정

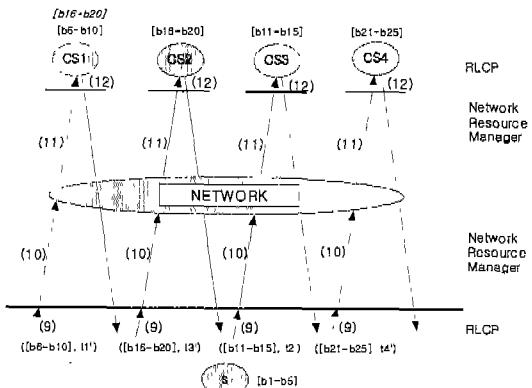


그림 11 요청된 자원에 대한 변경 과정

5. 구현 및 실험

본 논문에서는 RLCP의 적용성을 알아보기 위해 RLCP를 분산 비디오 프리젠테이션 응용에 적용하여 보았다. 표 3은 실험에 사용된 비디오 데이터 및 사용자 요청의 특징을 보여주고 있다. 요청은 각 협동 서버가 평균 1분

표 3 분산 비디오 프리젠테이션 실험 데이터

실험 변수	변수값
비디오의 개수	300
사용자 요청의 개수	600
협동 서버의 개수	5
비디오 세그먼트의 크기	5-25 블록(평균 12 블록)
블록의 크기	6조 분량의 압축 데이터
사용자 요청 평균 도착 시간	5-25초(평균 12초)
디스크 너퍼 크기	30MB
디스크 대역폭	1.9MB

마다 하나씩 도착하는 것으로 가정하였다. 그리고 사용자의 비디오 요청 패턴은 실제 비디오 대여 가게에서 수집된 값[15]을 이용하였다.

또한 비디오 데이터가 중복되어 저장되는 정도가 검색 스케줄의 생성에 미치는 영향을 알아보기 위하여 실험에서 중복 비율(Replication Ratio)을 아래와 같이 정의하여 사용하였다.

$$\text{Replication Ratio} = \frac{\sum_{\text{server } s} \text{Blocks} \in \text{Server}(s)}{\sum_{\text{video } v} \text{Blocks} \in \text{Video}(v)}$$

중복 비율이 1이면 모든 비디오 데이터가 중복이 되어 있지 않은 경우이다. 중복 비율이 커질수록 더 많은 비디오가 중복되어 검색 스케줄 생성에 유연성을 제공하나 저장 공간의 낭비를 초래할 수 있다. 따라서 대개 매우 빈번히 요청되는 비디오나 초기 자연을 줄이기 위해 비디오의 첫 부분만을 중복시킨다. 실험에서는 중복 비율을 1, 1.3, 1.5로 달리 하면서 미치는 영향을 측정하였다.

특히 분산 비디오 응용에서 비디오 세그먼트의 전송과 처리를 위해 협동 서버간에 세 가지 형태의 협동 방식을 고려하였으며 그들의 장단점을 비교하였다. 협동 방식은 다음과 같다.

- 블록 기반 프리젠테이션(Block-Oriented Presentation: BOP) 블록 단위로 협동 서버간에 검색 스케줄을 생성하고 처리한다.
- 세그먼트 기반 프리젠테이션(Segment-Oriented Presentation: SOP) 일련의 연속된 블록인 세그먼트 단위로 협동 서버간에 검색 스케줄을 생성하고 처리한다.
- 혼합형 프리젠테이션(Hybrid Presentation) 위의 두 가지 방식을 결합한 형태로 실제 검색 스케줄은 세그먼트 단위로 하지만 전송된 데이터의 처리는 블록 기반 방식을 취하는 혼합된 형태이다.

또한 베파의 효과적인 사용을 위해 동적인 베파 할당 방식을 사용하였다. 다른 베파 할당 방식을 사용한 결과는 [15]에 나타나 있다. 실험을 통해서 얻어진 결과를 살펴보면, 우선 그림 12은 협동 서버 전체를 통틀어 대기 시간 이내에 수행된 사용자 요청의 개수를 보여준다. 그림에서 블록 기반 방식과 혼합형 프리젠테이션 방식이 세그먼트 기반 방식보다 훨씬 더 많은 사용자 요청을 만족시켰음을 볼 수 있다. 이것은 스케줄링의 단위가 적을수록 정교하고 시스템 자원을 효과적으로 쓸 수 있는 검색 스케줄을 만들 수 있기 때문이다. 또한 중복 비율이 증가할수록 더 많은 사용자 요청을 만족시킬 수가 있는 데 이것은 중복이 많이 될수록 검색 스케줄을 생성할 때 고려할 수 있는 협동 서버의 수가 증가하고 따

라서 어느 특정 협동 서버의 부하에도 불구하고 검색 스케줄을 완성할 수 있는 가능성이 커지기 때문이다.

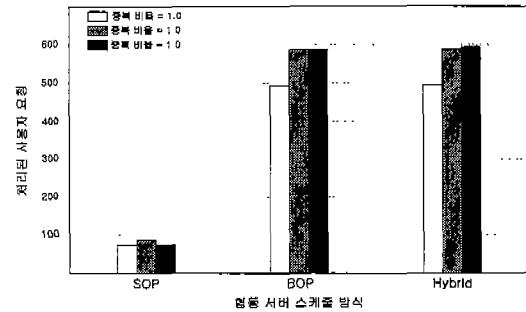


그림 12 처리된 사용자 요청의 수

다음으로는 사용자의 요청에서 프리젠테이션이 시작될 때까지의 시간을 나타내는 응답 시간을 측정한 결과를 그림 13에 보인다. 블록 기반이나 혼합형 모두 세그먼트 기반 협동 방식보다는 더 짧은 응답 시간을 보였다. 또한 데이터의 중복을 허용할수록 사용자 응답 시간이 줄어드는 것을 볼 수 있다. 이것은 앞에서 언급한 것처럼 중복 비율이 증가할수록 검색 스케줄을 생성할 수 있는 경우의 수가 늘어남에 따라 프리젠테이션을 지원시켜야 하는 상황이 줄어들기 때문이다.

그림 14는 프리젠테이션 수행에서 사용자 요청에 대해 사용된 디스크/네트워크 대역폭을 나타내 준다. 특히, 세그먼트 기반 방식에서는 수용된 사용자 요청이 상대적으로 적기 때문에 만약 비디오 세그먼트가 로컬 디스크에 존재한다면 거의 아무런 경쟁 없이 검색될 수 있다. 블록 기반이나 혼합형 프리젠테이션에서는 수용된 사용자 요청이 증가하여 시스템 자원에 대한 경쟁이 치열해지고 결과적으로 협동 서버들 간에 요청이 증가하여 결과적으로 디스크 및 네트워크 대역폭의 양이 증가하게 된다.

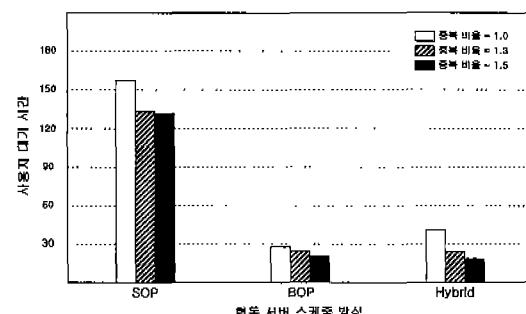


그림 13 협동 방식에 따른 사용자 응답 시간

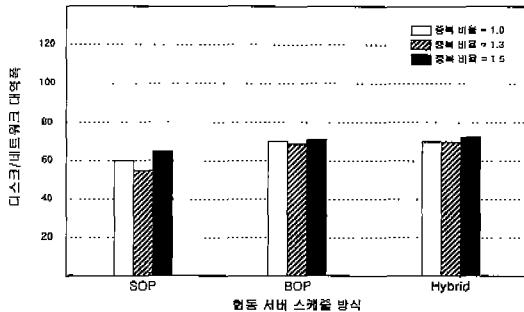


그림 14 협동 방식에 따른 디스크/네트워크 대역폭

이제까지의 실험으로 볼 때 블록 기반이나 혼합형 프리젠테이션의 경우 서로 비슷한 성능을 보였으며 특히 세그먼트 기반의 프리젠테이션에 비해서는 월등히 우수함을 알 수 있다. 그렇지만 블록 기반의 경우 블록 단위로 검색 스케줄이 만들어지므로 실제 전체 검색 스케줄을 생성하는 데 걸리는 시간이 혼합형 프리젠테이션에 비해 훨씬 오래 걸린다. 그림 15는 각각의 협동 방식에 따른 검색 스케줄 생성에 소요되는 시간을 보여준다.

결과적으로 블록 기반과 혼합형 협동 방식은 엇비슷한 성능을 보인 반면 검색 스케줄 생성에 소요된 시간이 혼합형의 경우 훨씬 짧았으며 이는 실제 협동 방식에서 세그먼트 단위의 스케줄링과 블록 단위의 데이터 처리 및 자원 관리가 수반되어야 함을 보여준다.

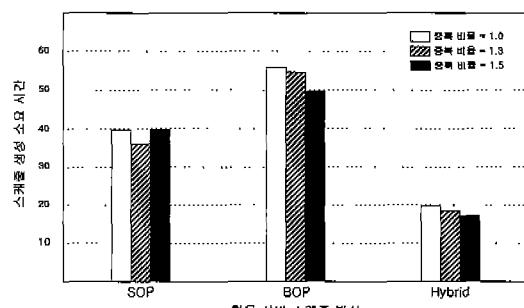


그림 15 협동 방식에 따른 스케줄링 시간

6. 기존 프로토콜과의 비교

멀티미디어 응용에 필요한 여러 자원에 대한 절충은 주로 transport 프로토콜 레벨에서 다루어졌다[1]. 응용 레벨에서의 요구 사항은 자원 절충을 더 힘들게 만든다. 분산 멀티미디어 응용에서 객체의 검색과 자원 절충을 다루기 위해 제안된 대표적인 프로토콜과 기술은 다음

과 같다.

- Application and Network Synchronization Protocol (ASP와 NSP)[2]
- Multimedia Application Protocol(MMAP)[10]
- Negotiation and Resource Reservation Protocol (NRP)[12]

Application and Network Synchronization Protocol

Protocol: [2]에서는 ASP(Application Synchronization Protocol)와 같은 응용 서비스 레벨에서의 프로토콜과 NSP(Network Synchronization Protocol)와 같은 네트워크 서비스 레벨에서의 프로토콜을 결합하여 네트워크 서비스의 하나로 객체들 간의 동기화 지원을 제안하고 있다. ASP는 네트워크상에 산재한 여러 소스에 저장된 객체들로 구성되는 복합 멀티미디어 객체의 검색에 관련한 서비스를 제공하기 위해 제안되었다. 이 프로토콜은 여러 형태의 응용에 필요한 기능과 일반적인 네트워크 인터페이스를 제공하는 NSP와 연합되어 있다. 이러한 프로토콜들은 멀티미디어 프리젠테이션 응용을 위한 자원 절충을 다루기보다는 객체 프리젠테이션의 동기화에 초점을 맞추고 있다.

Multimedia Application Protocol(MMAP): MMAP에서 제안하는 것은 QoS 요구사항을 설정하고 그것을 NSP와 절충하며 사용자와 NSP 사이에 일어나는 동적 반응을 다루는 것들이 일반적인 프로토콜 서비스 인터페이스로 제공된다[10]. MMAP에 의해 제공되는 서비스는 이러한 점들을 강조하므로 orchestrated 멀티미디어 응용에도 적용될 수 있다. 그러나 MMAP는 객체 프리젠테이션 사이에 존재하는 종속 관계를 다루지 못하며 이 논문에서 다루었던 멀티미디어 프리젠테이션의 협동적 특성을 제대로 지원하지 못한다.

Negotiation and Resource Reservation Protocol

(NRP): NRP는 전체 분산 응용을 포함하는 end-to-end 관점에서 응용 레벨 QoS를 서술하기 위해 제안된 것이다[12]. NRP는 사용자가 명시한 QoS 값의 범위와 통신 링크나 end system에서 이용 가능한 자원을 기반으로 절충을 수행한다. NRP는 미디어에 종속적인 특성들(예를 들어, 비디오 프레임 크기가 $480 * 360$)로 표시된 응용 QoS를 네트워크 QoS로 매핑하는 것을 도와준다. 이러한 매핑은 다음과 같이 세 단계로 진행된다: (i) 응용 QoS를 네트워크 QoS로 매핑, (ii) 각각의 통신 링크에 대한 QoS 값의 고정, (iii) 최종 QoS 값에 따라 자원 예약의 완화. 또한 NRP는 비디오 프레임 크기와 같은 장치의 용량에 따라 가용의 네트워크 자원을 조절하는 것을 돋는다. 본 논문에서는 여러 개의 협동 서버에서

객체를 검색하기 위한 스케줄을 생성하기 위해 RLCP를 제안하고 있다. 스케줄의 결충은 객체의 프리젠테이션 사이에 존재하는 의존 관계를 고려한다. 따라서 RLCP의 목적은 NRP의 그것과는 다르다.

7. 결 론

네트워크 레벨이나 응용 레벨에서 멀티미디어 프리젠테이션에 의해 요청되는 자원들을 결충하고 예약하는 것은 아주 복잡한 태스크이며 응용에 매우 종속적이다. 본 논문에서는 여러 개의 협동 서버에 분산되어 있는 객체들로 구성된 프리젠테이션을 수행하는 멀티미디어 응용에 대해 고찰하였다. 이러한 응용의 핵심 요구 사항은 요구되는 시간에 협동 서버에서 객체를 전송 받기 위한 검색 스케줄의 생성이다. 이러한 검색 스케줄은 네트워크와 응용 레벨에서의 자원 가용성이 의존한다. 따라서 검색 스케줄을 결정하는 것은 결국 네트워크나 응용 레벨에서 필요한 자원들을 결충하고 예약하는 과정을 반드시 수반한다.

본 논문에서는 분산 멀티미디어 프리젠테이션 응용에 필요한 검색 스케줄의 생성을 다루기 위한 응용 서브계층 프로토콜인 RLCP를 제안하였다. RLCP는 네트워크와 응용 레벨에서 자원을 예약하게 해주며 단계적으로 우선 가용 네트워크 자원을 측정하고 그것을 바탕으로 협동 서버에 대한 초기 검색 스케줄을 생성한다. 이 검색 스케줄은 두 번째 단계에서 승인을 위해 해당 협동 서버에 전달된다. 이 단계에서 협동 서버는 필요한 시스템 자원을 확인하고 끌어 둔다. 만약 그 스케줄이 관련된 모든 협동 서버에 의해 승인되면 RLCP는 그 스케줄을 확정하게 된다. 만약 그렇지 않으면, 검색 스케줄이 확정될 때까지 일련의 자원 수정 단계가 진행된다. RLCP는 Tenet suit protocol[4]나 ST2+[9]와 같은 미리 자원 예약 기능을 지원하는 네트워크 프로토콜에 의해 제공되는 서비스를 이용할 수 있다.

참 고 문 헌

- [1] D. Ferrari, *Client Requirements For Real-Time Communication Services*, IEEE Communication Magazine, Vol. 28, No. 11, pp. 65-72, Nov. 1990.
- [2] T.D.C. Little and A. Ghafoor, *Multimedia Synchronization Protocols for Broadband Integrated services*, IEEE J. on Selected Areas of Communications, Vol. 9, No. 9, pp 1368-1382, Dec. 1991.
- [3] L. Zhang, S. Deering, D. Estrin, S. Shenker, and D. Zappala, *RSVP: A New Resource ReSerVation Protocol*, IEEE Network, pp. 8-18, Sep. 1993.
- [4] D. Ferrari, et al, *Network Support for Multimedia : A Discussion of Tenet Approach*, Computer Networks and ISDN Systems, Special issue on Multimedia Networking, 1994.
- [5] K. Nahrstedt and Ralf Steinmetz, *Resource Management in Networked Multimedia Systems*, IEEE Computer, Vol. 28, No. 4, April 1995.
- [6] K. S. Candan, B. Prabhakaran, and V. S. Subrahmanian, *CHIMP : A Framework for Supporting Multimedia Document Authoring and Presentation*, Proceedings of ACM Multimedia Conference, Boston, Nov. 1996.
- [7] S.V. Raghavan, B. Prabhakaran and S.K. Tripathi, *Quality of Service Considerations For Distributed, Orchestrated Multimedia Presentation*, Proceedings of High Performance Networking, pp. 217-238, France, July 1994.
- [8] K. Nahrstedt and J. Smith, *The QoS Broker*, IEEE Multimedia, Vol. 2, No. 1, pp. 53-67, 1995.
- [9] L. Delgrossi and L. Berger, *Internet Stream Protocol Version ST2+*, RFC 1819, Aug. 1995.
- [10] S.V. Raghavan, B. Prabhakaran, and S.K. Tripathi, *Handling QoS Negotiations in Distributed Orchestrated Presentation*, Journal of High Speed Networking, Vol. 5, No. 3, pp. 277-292, 1996.
- [11] E. Hwang, B. Prabhakaran, and V.S. Subrahmanian, *Distributed Video Presentation*, Proceedings of International Conference on Data Engineering, Florida, 1998.
- [12] G. Dermier, W. Fiederer, I. Burth, and K. Rothermel, *A Negotiation and Resource Reservation Protocol (NRP) for Configurable Multimedia Applications*, Proceedings of IEEE Multimedia Conference, pp. 113-116, Japan, 1996.
- [13] W. Zhao and S.K. Tripathi, *A Resource Reservation Scheme for Synchronized Distributed Multimedia Sessions*, Second International Workshop on Multimedia Information System, New York, 1996.
- [14] E. Hwang and B. Prabhakaran, *Protocol for Collaborative Multimedia Presentation*, Proceedings of International Conference on Multimedia & EXPO, New York, 2000.
- [15] E. Hwang, P. Brabhakaran, and V.S. Subrahmanian, *Presentation Planning for Distributed Video Systems*, accepted for publication on Trans. on Data and Knowledge Engineering.
- [16] Video Store Magazine, Dec. 1992.



황인준

1988년 서울대학교 컴퓨터공학과(학사).
1990년 서울대학교 컴퓨터공학과(석사).
1993년 Univ. of Maryland at College Park 전산학과(박사). 1993년 6월 ~
1999년 8월 Hughes Research Lab. 연구교수. 1998년 8월 ~ 1999년 8월 Bowie State Univ.,

Assistant Professor. 1999년 9월 ~ 현재 아주대학교 정보통신전문대학원 조교수. 관심분야는 데이터베이스, 멀티미디어 시스템, 정보 통합, 전자 상거래, XML 응용