

실시간 수송계층 프로토콜의 설계

박 창 윤†

요 약

본 연구에서는 호스트에서 실시간 통신을 지원하는 실시간 수송계층 프로토콜 (RTTP)을 고안/설계하였다. 현존하는 실시간 통신 응용은 물론 향후의 새로운 응용에도 적용될 수 있는 일반적인 사용자 요구 모델을 개발하고, 이를 기반으로 하여 연결을 위주로 하는 실시간 수송계층의 서비스 프리미티브를 정의하였다. 다양한 네트워크에 적용될 수 있도록 실제 네트워크에 독립적인 가상 네트워크 계층을 정의하고, 이를 기준으로 프로토콜을 구현하는 체계를 제안하였으며, 또, 수송계층 자체의 처리시간을 예측 가능하게 하기 위해 라이브러리 함수 형태로 구현하는 방법을 채택하였다. RTTP의 주기능으로서 사용자의 실시간 요구를 네트워크의 품질인자로 변환하는 품질인자 변환 규칙을 개발하였다. 또, 실시간 데이터의 특징을 이용하여, 요구되는 신뢰성에 따라 조절하는 선택적인 오류 제어 방법을 개발하였다.

Design of A Real-Time Transport Protocol (RTTP)

Changyun Park†

ABSTRACT

This paper describes *Real-Time Transport Protocol (RTTP)* that provides for a user connection-oriented services with performance guarantee. Assuming that real-time network services will be provided, RTTP solves problems at end-hosts. RTTP has defined a new set of primitives for real-time communication services based on the analysis of user requirements on various applications. Introducing the notion of *Abstract Network Layer* independent of real networks, RTTP's implementation scheme can be applied to various networks. The study also introduces an implementation strategy, *protocol as a library*, to provide predictability of protocol processing time, which should be known *a priori* for performance guarantee. Internally, RTTP has a set of QoS translation rules that change a user's performance requirements to the equivalent network QoS parameters. RTTP also introduces a new opportunistic error control method that can trade-off between overhead and error correction according to a user's requirement on reliability.

1. 서 론

기존의 통신 프로토콜들은 데이터의 신뢰성있는 전송만을 보장할 뿐 전송 시간에 대한 보장은 없다.

따라서, 음성 및 영상 등과 같이 어떤 시간 범위 안에서만 유효한 의미를 갖는 실시간 데이터를 전송하는 응용에서는 이들 프로토콜을 그대로 사용하는 것은 부적당하며, 전송시간에 대한 보장이 제공되는 새로운 통신 프로토콜이 요구된다. 이와 같이, 데이터나 시스템의 특성상 데이터 전송이 어떤 시간 범위에서 이루어져야 한다는 시간제약성을 갖는 통신을 통칭하여 실시간 통신(Real-Time Communication)이라 부

※ 이 논문은 1994년도 한국학술진흥재단의 공모과제 연구비에 의하여 연구되었음.

† 정 회 원: 중앙대학교 컴퓨터공학과

논문접수: 1996년 5월 25일, 심사완료: 1997년 1월 21일

른다[14, 15].

실시간 통신 프로토콜은 항공교통 제어와 같은 분산 실시간 시스템은 물론, 최근 각광 받고 있는 멀티미디어 시스템 등의 구축을 위해서 반드시 필요한 기반 기술이다[25, 26]. 최근 실시간 통신을 요구하는 응용들이 확산되어 감에 따라 실시간 통신 프로토콜도 활발히 연구되기 시작하였다. 그러나, 실시간 컴퓨팅 분야의 경우와 마찬가지로, 실시간 통신 분야도 고속 통신의 특수한 경우로 단순히 처리하려는 경향이 많으며, 전송시간의 보장이라는 문제를 근본적이고 체계적으로 해결하려는 연구는 아직 초기 단계에 머물러 있다고 할 수 있다[3, 22]. 본 연구에서는 수송계층 프로토콜을 기준점으로 하여, 호스트 측에서의 실시간 통신의 제반 문제점들을 파악하고 그 해결책들을 제시하는 것을 목적으로 한다. 구체적으로, 여러 실시간 응용에 이용될 수 있는 일반적인 실시간 수송계층 프로토콜인 RTTP를 정의하고, 여러 네트워크 환경에 적용될 수 있는 체계적인 구현 방법으로 설계하는 것을 연구 목적으로 한다.

기존의 통신 모델에서 수송계층 프로토콜은 신뢰성이 없는 네트워크 서비스로부터 사용자에게 신뢰성 있는 전송 서비스를 제공하는 매우 중요한 기능을 담당하며, 그 구성도 매우 복잡하였다. 그러나, 실시간 통신의 경우에는 이와 같은 수송계층의 개념에 여러 변화가 생기게 된다. 첫째, 성능 보장이 되지 않는 서비스를 이용해서 성능 보장이 필요한 실시간 서비스를 제공할 수 없으므로, 실시간 수송계층은 실시간 전송 서비스를 제공하는 네트워크 계층 위에서만 구성이 가능하다는 점이다. 둘째, 실시간 통신에서는 정해진 시간 안에 도착하지 못하는 데이터는 오류로 간주되므로, 재전송을 통해 신뢰성을 향상시키는 것은 한계가 있게 된다. 이와 같은 실시간 특징 때문에 실시간 수송계층은 기존의 수송계층과는 다른 정의를 가져야 한다.

본 연구에서는 실시간 수송계층 프로토콜은 네트워크 계층이 제공하는 실시간 전송 서비스를 네트워크의 세부 사항들을 제거하여 사용자가 쓰기 쉬운 서비스로 제공하는 기능을 하며, 신뢰성 제공은 네트워크 서비스의 신뢰성과 전송시간의 한도 내에서 제한적으로 이루어진다는 관점을 견지한다. ATM 및 "Integrated Service"[3]의 실시간 서비스들과 RTTP의 차

이는, 이들이 호스트와 호스트 사이의 성능이 보장되는 전송 서비스를 다루는 반면, RTTP는 이들 서비스를 이용하여 제공되는 사용자 간의 실시간 서비스를 다룬다는 점이다. 다시말해, 계층적으로 RTTP는 ATM과 같은 실시간 네트워크계층과 응용계층 사이의 중간계층에 해당된다.

프로토콜 개발에 있어 다루어야 할 문제들은 그 프로토콜이 제공할 서비스들, 하위 프로토콜로부터 제공되는 서비스들, 그리고 주어진 기능을 구현하는 내부 기법 등이다. 이에 따라 본 연구에서도 다음과 같은 세 가지의 내용들이 연구되었다. 첫째, 실시간 수송계층의 기능과 프리미티브를 새롭게 정의하였다. 실시간 수송계층의 기능은 제공되는 서비스 프리미티브(primitive)를 통해 정의되는데, 프리미티브들은 응용계층의 다양한 실시간 통신 요구들을 충족시킬 수 있는 일반적인 형태를 갖추어야 한다. 이를 위해서 본 연구에서는 기존에 나와 있는 실시간 수송계층 프로토콜들을 참조하는 것은 물론, 새로이 창출되고 있는 통신 응용 서비스들도 조사하여 실시간 통신 사용자 요구 모델을 새롭게 고안하였다. 다음, 이 모델에 입각하여 일반적인 형태의 실시간 수송계층 서비스 프리미티브들을 정의하였다. 이때, 프리미티브들이 시스템과 네트워크 부하 상황에 따라 수행 가능하지 않을 수도 있으므로, 이 경우에 대비한 정확한 의미를 정의하였다.

둘째, 여러 네트워크에 공통적으로 적용될 수 있는 체계적인 구현 방법을 개발하였다. 현재, 네트워크계층에서 제공되는 실시간 통신 기능은 네트워크의 종류에 따라 큰 차이를 보이고 있다. 본 연구에서는 네트워크계층 서비스에 독립적인 구현 방법의 개발을 위해, 실시간 통신에 필수적이며 실제 네트워크에 독립적인 형태의 실시간 가상 네트워크 서비스를 정의하고, 이를 기준으로 실시간 수송계층을 두 개의 부분 계층으로 나누는 가상 네트워크계층(abstract network layer) 모델을 채택하였다. 하위 계층인 가상 네트워크계층은 실제 네트워크가 제공하는 서비스를 이용하여 가상 네트워크 서비스를 구현하며, 상위 계층인 RTTP 서비스 계층은 가상 네트워크계층이 제공하는 서비스를 이용하여 실시간 수송계층 서비스 프리미티브를 구현한다. 또, 실시간 통신에 필수적인 예측 가능성을 보장하기 위해 RTTP를 라이브러리 함수로

구현하는 방법을 제시하고 이에 수반되는 문제들을 해결하였다.

세째, 실시간 전송의 구현에 필요한 내부 기법을 개발하고, 이를 적용하여 RTTP를 설계하였다. 일반적인 실시간 수송계층과 마찬가지로 RTTP의 주기성은 사용자의 실시간 요구를 네트워크의 품질인자로 변환하는 것이다. 본 연구에서는 가상 네트워크 계층을 도입하였으므로 RTTP의 사용자 요구를 가상 네트워크의 품질 인자로, 또, 이를 ATM 네트워크의 품질 인자로 변환하는 규칙을 개발하였다. 한편, 실시간 데이터의 시간적 및 신뢰성에 관한 특징을 이용하기 위해, 본 연구에서는 요구되는 신뢰성에 따라 오류 제어의 부하 및 정확도를 조절할 수 있는 낙관적인 오류 검출 방법을 개발하였다.

본 논문의 구성은 다음과 같다. 2장에서는 다양한 응용에서의 실시간 통신에 관한 요구를 분석하고, 이에 따라 RTTP의 프르미티브들을 정의한다. 본 연구에서 개발한 RTTP의 기능 및 내부 기법들은 3장에서 다루며, 4장에서는 가상 네트워크 모델을 기반으로 한 구조 및 구현 기법을 설명한다. RTTP의 구체적인 설계는 5장에서 시술한다. 6장에서는 본 연구를 관련된 다른 연구들과 비교하여 특징을 설명하며, 7장에서 결론 및 향후 연구를 설명한다.

2. 사용자 요구 분석 및 RTTP 서비스 정의

실시간 통신 수송계층을 설계할 때, 가장 먼저 고려하여야 할 사항은 실시간 수송계층이 어떤 수준의 기능을 제공하여야 할 것이냐는 점이다. 실시간 통신의 정의에서 본 바와 같이 실시간 수송계층의 대표적 기능은 데이터의 적시전송이다. 그러나, 적시 전송의 정의에 따라 다양한 수준의 실시간 전송 서비스가 가능하다. 따라서, 어떠한 기능을 제공할 것이냐에 대한 결정은 결국 사용자 어떠한 수준의 적시전송 기능을 요구하느냐에 우선 좌우된다고 볼 수 있다.

2.1 실시간 통신 요구에 대한 분석

실시간 통신 서비스에 대한 요구는 공장 제어 등과 같은 분산 실시간 시스템을 구성하는 기술로서 오래 전부터 연구되어 왔다. 또, 음성이라는 데이터가 갖는 실시간성 때문에, 전화와 같은 음성통신을 개발하는

분야에서도 사용자가 요구하는 서비스 수준에 대한 연구 결과들을 내놓고 있다. 그러나, 이와 같은 연구들은 해당 응용 분야 내에서의 사용자 요구들만을 중점적으로 분석하였기 때문에 일반성이 결여되어 있다는 문제점이 있다. 또, 시스템이 구현될 네트워크 환경에 대한 가정을 담고 있기 때문에 중요한 요구 사항들이 생략되는 경우도 있다.

실시간 통신에 대한 사용자의 요구를 비교적 체계적으로 연구한 것은 Ferrari이다[8]. Partridge 등도 관련된 연구를 하였지만 사용자의 요구 자체보다는 사용자의 요구에 따라 파생되는 전송데이터의 네트워크에 대한 요구사항에 초점을 맞추고 있다[4, 19]. Ferrari의 연구에서 사용자의 실시간 요구를 표현하는 요구 인자들은 다음과 같은 항목으로 구성된다.

- 전송시간(delay) 및 전송시간 변이(delay jitter)
- 전송처리율(throughput)
- 신뢰성(reliability)

이와 같은 요구인자들과 표현 방법들은 대부분의 현존하는 실시간 통신 응용들의 요구사항들을 기술할 수 있다고 인정되고 있는 추세이다. 그러나, 이들은 여전히 사용자 측면보다는 네트워크의 품질인자를 고려한 입장에서 정의되고 있으며, 그에 따라 표현의 용이성, 즉, 사용자가 이들 항목으로 요구를 표현하는 것이 얼마나 쉬운가에 대한 고려가 부족하다고 할 수 있다. 예를 들어, 요구인자와 함께 기술되는 교통량명세(traffic descriptor)의 "token bucket model" [19] 등은 사용자에게는 부담이 될 수 있다. 또, 이들은 멀티미디어 응용을 위주로 한 사용자 요구의 분석이라고 할 수 있다. 따라서, 본 연구에서는 사용자 입장에서 편의성을 고려하며, 멀티미디어 응용뿐만 아니라 기존의 실시간 시스템의 요구들도 고려하여 요구인자들을 정의한다.

그림1과 그림2는 대표적인 실시간 응용인 분산 제어 시스템과 화상회의와 같은 playback응용에서 나올 수 있는 실시간 통신의 예를 코드 요약으로 각각 나타낸 것이다. 그림의 두 실시간 통신은 데이터가 어떤 정해진 시간 안에 상대방에게 전달되어야 한다는 점에서는 같다. 그러나, 실시간 통신이 언제 요구되는가와 같이 몇 가지 점에서는 차이를 보인다. 따라서,

다양한 실시간 통신 응용을 포함할 수 있는 일반적인 실시간 수송서비스를 정의하기 위해서는 사용자의 실시간 통신 요구에 대한 체계적인 분류가 먼저 필요하다.

```
report_status_change()
{
    s = read_sensor();

    if (s>LIMIT)
        transmit(control, s);
}
```

(그림 1) 분산제어 시스템에서 실시간 통신의 예
(Fig. 1) An example of real-time communications in distributed control systems

```
transmit_samples()
{
    pbuff[i++] = read_sample();

    if (i>=BUFSIZE) {
        transmit(player, pbuff);
        i = 0;
    }
}
```

(그림 2) Playback 응용에서 실시간 통신의 예
(Fig. 2) An example of real-time communications in playback applications

본 연구에서는 실시간 통신의 종류를 다음과 같은 네 가지 요소에 입각해 분류한다.

1. 데이터 종류

- 스트림 데이터 (stream data)
전송되는 데이터가 어떤 큰 데이터의 일부분이 되면서 전후로 전송되는 데이터와 함께 의미를 갖는 경우.
- 메시지 데이터 (message data)
전송되는 데이터가 독립적인 의미를 갖는 경우.

2. 통신 요청 주기

- 주기적 (periodic)
통신 요청이 주기적인 경우.
- 비주기적 (aperiodic)
통신 요청이 비주기적인 경우. (비주기적인 경우라

할지라도, 연속되는 두 요청의 최소 간격은 정의되어야 한다.)

3. 데이터 크기

- 고정 크기 (constant size)
데이터의 크기 및 비율이 요청될 때마다 일정한 경우.
- 가변 크기 (variable size)
요청될 때마다 변하는 경우.

4. 신뢰성

- 오류 불허용 (error-free)
전송 중 데이터에 오류가 허용되지 않는 경우.
- 오류 허용 (error tolerant)
어떤 정해진 한도 내에서 전송 중 데이터에 오류가 생기거나 데이터가 상실이 허용되는 경우.

이와 같은 네 가지의 요소로 실시간 통신을 분류하면 총 16가지 종류가 나오게 된다. 16가지 모두에 대해 실시간 통신 응용이 현재 존재하는 것은 아니므로 위의 분류는 너무 추상적이라고 생각할 수 있다. 그러나, 새로운 통신 응용이 계속 생겨나는 추세를 감안하면, 16가지의 종류의 사용자 요구가 가능하다고 할 수 있다. 대표적인 실시간 통신 응용을 예로 들어 그 분류를 살펴보면 다음과 같다.

- 음성 데이터 전송
스트림 데이터 /주기적/고정 크기/ 오류 허용
- MPEG 데이터 전송
스트림 데이터 /주기적/가변 크기/ 오류 허용
- 분산 실시간 제어 (그림1의 경우)
메시지 데이터 /비주기적/고정 크기/ 오류 불허용

위와 같은 분류에 입각하여 보면, 사용자가 실시간 통신을 요구한다는 것은 공통적으로 요구되는 전송 시간에 대한 요구와 함께 위의 네 가지 요소를 명시한다는 것으로 볼 수 있다. 즉, 사용자의 실시간 요구는 전송시간과 위의 네 가지, 총 다섯 개값의 리스트로 정의된다. 사용자 요구의 값을 표현하는데 있어서 신뢰성을 제외한 모든 값은 결정적(deterministic) 표현(8)으로 정의되며, 신뢰성은 통계적 표현을 사용하도록 한다. 각 요소에 대한 통계적 표현은(예, 전송 시간에 대한 통계적 요구) 결정적 요구에 부합되지 않는 경우를 통신 실패로 분류하고 신뢰성으로 이를

표현하도록 한다.

본 논문에서 사용할 표기법, 단위 및 상수 값들을 정의하면 다음과 같다.(전송시간에 대한 요구는 위에서 언급한 Ferrari의 전송시간 및 전송시간 변이를 사용하도록 한다.)

1. 데이터 종류 ($C=(STRM, MSG)$)
2. 통신 요청 주기 (P)
 - 주기적: $P=(PRD, p$ (period) (sec))
 - 비주기적: $P=(APRD, i$ (min. inter-request time) (sec))
3. 데이터 크기 (S)
 - 고정 크기: $S=l$ (data length) (bit)
 - 가변 크기: $S=l_{max}$ (max. data length) (bit)
 - 평균 크기: $S=(l_{avg}, n, u)$ (average length, burst count, unit time)
4. 신뢰성 (R)
 - 오류 불허용: $R=100$ (%)
 - 오류 허용: $R=(r$ (min. acceptable rate) (%), u (reliability interval unit) (sec))
5. 전송시간 ($T=(D, J)$)
 - D : 최대 전송시간
 - J : 최대 전송시간변이

여기서 사용자가 요구하지 않는 요소는 X 로 표시하며, 각 요소의 의미에 따라 실제 값은 결정되도록 한다.(예, 최대 전송시간 변이 경우, ∞)

예를 들어, [8]에서 제시된 1 Mbit 프레임을 초당 30 프레임을 전송하여야 하는 주문형 비디오 시스템의 실시간 요구는 다음과 같이 표현된다.

- $C=STRM$
- $P=(PRD, 1/30$ sec)
- $S=1$ M bit
- $R=(90\%, 5$ sec)
- $T=(D: X$ sec, $J: 5$ msec)

2.2 RTTP 서비스 프리미티브 정의

RTTP의 프리미티브들과 성능인자들은 앞의 실시간 요구 분석을 기반으로 하여 정의된다. RTTP의 실시간서비스는 연결위주(connection-oriented) 서비스

형태를 갖는다. 실시간 통신에서 연결없는(connectionless) 서비스가 필요한가에 대한 문제는 많은 논란의 여지가 있다. 본 연구에서는 실시간 통신의 요체가 전송 성능의 보장이며 통신 자원의 예약 없이 전송 성능의 보장은 불가능하다는 관점에 따라, 모든 실시간 통신은 계약을 체결하는 것에 해당되는 연결설정이 선행되어야 한다고 가정한다. 따라서, 그림1과 같이 예외사건이 발생한 경우에만 통신하는, 의미적으로 연결없는 통신인 경우에도 최대 통신 빈도수를 감안한 연결은 시스템 초기화 시에 이미 설정되어야 한다.

한편, RTTP에서의 연결은 단방향 연결을 의미한다. 실시간 통신 응용에서는 통신하는 양쪽이 서로 다른 사용자 요구를 원하는 경우가 많으므로, 효율적으로 이를 처리하기 위해 송신 측이 수신 측으로 단방향 연결을 설정하도록 한다. 연결설정을 하는 프리미티브는 다음과 같다. (기본적인 개념들은 이미 언급되었으므로, 각 항목에 대한 자세한 설명은 생략한다.)

- 프리미티브::
 - open_request(type, destination-id, QoS, response)
- type::
 - Active Open: 송신측에서의 open
 - Passive Open: 수신측에서의 open
- QoS 성능인자::
 - 데이터 종류 (C)
 - 통신 요청 주기 (P)
 - 데이터 크기 (S)
 - 신뢰성 (R)
 - 전송시간 (T)
 - *: 최대 전송시간
 - *: 최대 전송시간변이
- response::
 - accept + connectionID + QoS
 - reject + reason

송신측의 연결 요구는 (즉, open_request(Active Open,...)) 송신측과 수신측 사이에 주어진 품질인자를 만족하는 연결 설정을 시도하고 그 결과를 돌려 받는다. 수신측의 연결 요구는 (즉, open_request(Passive Open,...)) 송신측이 연결을 요구하면 받아 들인다는

의미를 갖으며, 연결이 설정되면 해당 QoS를 포함한 정보를 돌려 받는다.

연결 설정이 이루어지고 난 뒤, 실제 데이터를 전송할 때 사용하는 프리미티브는 다음과 같다.

- 프리미티브::
 - send_data(destination-id, message, length, userTO)
- response::
 - success
 - timeout
 - fail +reason

기본적으로 RTTP의 데이터 전송 프리미티브는 프로그램 언어의 출력문 경우와 마찬가지로 비정지성(nonblocking)을 갖는다. 즉, 데이터 전송을 요구한 사용자는 전송의 완료까지 기다리지 않고 RTTP 수송계층에서의 작업이 완료되는 대로 다음 작업을 계속 수행할 수 있다. 이와 같은 결정의 배경에는 RTTP가 연결 위주의 통신이며 성능이 보장된 연결을 통해 전송이 이루어지기 때문이다. 많은 경우, RTTP는 네트워크계층에게 전송을 요청한 후 즉시 성공 응답을 보낸다. 그러나, 사용자 요구의 한도를 넘어 오류가 발생하였거나 사용자의 요구가 100% 신뢰성을 요구한 경우, 응답을 보류하고 오류 복구를 수행한다. 사용자가 정의한 시간 제한(userTO) 안에 오류 복구가 성공하면 성공 응답을, 시간 제한이 먼저 발생하면 시간 제한 응답을, 시간적으로 복구가 불가능하다고 판단되면 실패 응답을 보낸다(불가능 판단에 관한 자세한 사항은 오류 제어에서 설명). 시간 제한 또는 전송 실패를 응답 받은 사용자는 이를 예외사건 처리 등 자체적으로 처리하도록 한다.

데이터 수신 프리미티브의 정의는 다음과 같다.

- 프리미티브::
 - receive_data(destination-id, mes_buf, length, userTO)
- response::
 - success
 - fail
 - error type

데이터 수신 프리미티브의 정의에서 success/fail/error

type 중 하나를 응답하도록 한 이유는, 실시간 전송에서는 전송 중에 발생한 오류를 재전송과 같은 방법을 이용하여 수송계층이 자체적으로 해결하는 것이 필요 없을 때도 있으며, 경우에 따라 불가능할 수도 있기 때문이다. 수신인 경우에도 사용자는 시간 제한(userTO)을 두어, 비정상적인 수신인 경우 사용자가 시간적인 대처를 할 수 있도록 한다. 실패가 응답된 경우는, 수신된 데이터에서 오류가 발생하였고, 그것이 사용자가 요구한 신뢰성의 한도를 넘어서는 오류라는 의미이다. 따라서, 사용자는 대체 수송계층 연결을 설정하는 등 예외사건 처리를 수행하여야 한다. 오류 종류(error type)는 오류가 발생하였지만 신뢰성의 범위 내이라는 의미이며, 사용자가 오류 처리를 할 수 있도록 오류 종류를 응답한다. 수신 중에 시간 제한이 먼저 발생하는 경우는 오류의 일종으로 응답된다.

실시간서비스를 정의하는데 있어 한가지 유의할 사항은 실시간서비스란 사용자와 네트워크 간의 계약이며, 실시간서비스의 구현, 즉, 성능보장이란 이 계약을 수행하는 것이란 점이다. 사용자가 만일 계약을 파기하여 요구한 전송량보다 많은 데이터를 보낸다면, 네트워크가 전송시간 등의 성능을 보장하지 않아도 된다. 이와 같은 사항은 서비스 프리미티브에는 명기되지 않더라도 묵시적으로 정의된다.

3. RTTP의 기능

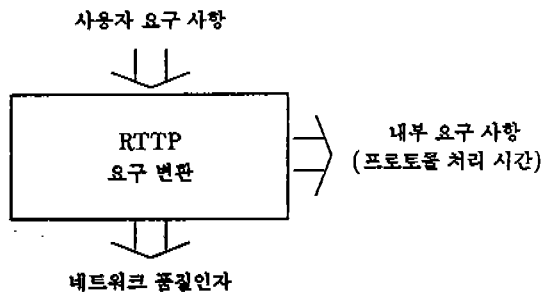
기존의 통신 프로토콜에서 수송계층은 가장 핵심적인 계층으로서, 전송 중의 제반 문제들을 모두 해결하여 상위계층에게 신뢰성있는 통신 기능을 제공하는 역할을 담당하였다. 이러한 역할을 수행하기 위해 수송계층은 연결 설정 및 해제, 사용자 지정(addressing), 멀티플렉싱, 오류 제어, 흐름 제어, 분해/재조합 등을 수행하였다. 여기서는 위의 기능들이 RTTP에서 어떻게 수정 또는 확장되어 구현되는 지에 대해서 설명하도록 한다.

3.1 연결 설정 및 해제

RTTP는 “수송계층 연결은 네트워크계층 연결의 사용자를 위한 포장”이라는 관점을 갖는다. 즉, 전송 성능이 보장된 하나의 네트워크계층 연결에 단순히

수송계층의 처리를 추가함으로써 하나의 수송계층 연결을 구성하도록 한다. 이와 같은 결정의 근거는 실시간 통신은 기본적으로 연결 위주 통신이며, 멀티플렉싱을 지원하며 성능 보장을 이루는 것은 매우 어렵기 때문이다.

따라서, RTTP에서 사용자의 연결 설정 및 해제 요구는 기본적으로 네트워크 계층으로의 연결 설정 및 해제 요구를 통해 간단히 이루어질 수 있다. 이 과정에서 사용자의 연결 설정 요구에 수반된 사용자 요구를 네트워크계층 연결 설정 시에 요구할 품질인자로 변환하는 작업이 필요하게 된다. 이와 같은 “품질인자 변환” 작업을 그림으로 나타내면 그림3과 같다.



(그림 3) 품질인자 변환
(Fig. 3) QoS Translation

“품질인자 변환”에 관한 기본적인 사항은 “사용자 요구 변환”이라는 이름으로 Ferrari에 이미 연구되었다[8]. 그러나, 그 결과를 실제 품질인자 변환에 응용할 때는 다음과 같은 두 가지 사항에 유의하여야 한다. 첫째, 사용자 요구의 요소와 네트워크의 품질인자가 다른 경우, 정확한 해석을 통해 동등하게 변환하여야 한다는 점이다. 이 작업의 난이도는 두 요소들 간의 유사성에 의해 결정되겠지만, ATM과 기존의 실시간 요구 사이에는 원활한 변환이 가능한 것으로 알려져 있다[13].

둘째, 수송계층 내부에서 주어진 메시지의 처리에 사용되는 시간의 범위를 예측하는 것은 간단한 일이 아니라는 점이다. 특히, 수송계층의 구현 방법에 따라서는 정확한 범위를 예측하는 것이 현실적으로 불가능할 수도 있다. 예를 들어, 수송계층을 별도의 프로세스로 구성한다면, 수행시간을 예측하기 위해서는 스케줄링의 분석이 필요하다. 따라서, 메시지 당 수송

시간의 처리시간이 예측 가능하도록 수송계층을 구성하여야 한다. 본 연구에서는 수송계층을 라이브러리 함수로 구현하도록 하여, 사용자 프로세스가 계속 수행하도록 하였다. 따라서, 수송계층에서의 처리 시간은 실시간 시스템 분야에서 연구되어진 프로그램 수행시간 분석 기법[17]을 이용하여 예측할 수 있다.

결론적으로, RTTP에서 연결 설정 및 해제에의 기능은 2.2에서 정의한 사용자 요구를 네트워크계층의 (실제적으로는 뒤에서 설명할 가상계층의) 품질인자로 변환하고, 이를 수반한 네트워크계층의 연결 설정 및 해제를 요청하는 일을 하는 것이다. 변환 과정 중에 메시지 당 수송계층의 처리 시간이 반드시 필요하므로, RTTP는 수행시간이 예측 가능하도록 구현되어야 한다.

3.2 멀티플렉싱

RTTP는 멀티플렉싱을 지원하지 않는다. 수송계층에서의 멀티플렉싱은 적용되는 환경에 따라 다양한 의미를 갖는다. 본 연구에서는 두 가지의 의미의 멀티플렉싱을 고려하였다. 첫째, TCP와 같이 여러 상위계층의 연결을 하나의 네트워크계층 연결로 합하는 의미에서의 멀티플렉싱이다. 여기서의 멀티플렉싱은 여러 사용자 간의 통신 연결을 성능을 보장하면서 하나의 네트워크계층으로 묶는 것을 의미한다. 그러나, 기존의 통신에서와는 달리 단순한 비동기식 시분할 방식을 통한 멀티플렉싱으로는 성능 보장을 유지할 수 없으며, 멀티플렉싱을 한다 하더라도 전체적인 효율은 오히려 떨어질 가능성이 많다. RTTP의 현재 입장은 네트워크계층에서 이미 제공하는 성능이 보장된 연결을 부가의 처리를 하여 멀티플렉싱하는 것은 의미가 없다는 것이다.

멀티플렉싱의 두 번째 의미는 하나의 수송계층 모듈이 여러 응용들을 동시에 지원하는 경우를 말한다. RTTP는 이와 같은 의미의 멀티플렉싱도 지원하지 않는다. 뒤의 프로토콜 구조에서 다시 언급하겠지만, RTTP 모듈은 RTP[20]에서 처럼 라이브러리 함수 형태로 존재하며, 각각의 응용이 이를 포함하도록 하였다. 이와 같은 결정의 이유는 RTTP가 멀티플렉싱을 지원하면, RTP 내에서의 큐에서의 지연 등 메시지 처리 시간을 예측하는 것이 매우 어려워지기 때문이다.

3.3 사용자 지정

RTTP가 멀티플렉싱을 하지 않으므로 RTTP에서 목적지에 해당하는 사용자를 지정하는 것은 단순한 문제가 된다. 그러나, 각 사용자가 RTTP 모듈을 포함하여 동작하므로, 어느 한 순간 여러 사용자가 RTTP를 사용한다면 여러 개의 RTTP 모듈이 존재하게 된다. 결국, 네트워크계층 또는 하위 수송계층이 목적지에 해당하는 RTTP를 지정하는 어드레싱, 즉, 사용자를 지정하는 문제가 생기게 된다.

하위계층이 사용자를 지정하는 기능이 있는 경우(예, UDP의 포트번호), 위 문제는 이를 이용하여 간단히 해결된다. 만일, 하위계층이 프로토콜만을 지정할 뿐 사용자를 지정하는 기능이 없는 네트워크 계층인 경우는(예, IP), RTTP와 네트워크계층 사이에 간단한 교환 계층을 둬으로써 해결할 수 있다. 이 교환계층이 하는 일은 RTTP 연결을 설정할 때 네트워크계층의 연결 명칭과 사용자 명칭을 연계하여 테이블에 보관하고, 메시지가 수신됐을 때에는 테이블을 조회하여 사용자에게 메시지를 전달하는 것이다.

3.4 오류 제어

앞의 프리미티브 정의에서 언급한 바와 같이, RTTP는 성능 보장을 약속하는 네트워크계층을 기반으로 동작하므로, 전송오류는 매우 드물다고 생각할 수 있다. 또, 음성 및 영상 데이터의 경우, 100%의 신뢰성을 요구하지 않으므로 TCP에서의 같은 철저한 오류 제어는 불필요할 수 있다. 한편, 여러 차례 언급한 바와 같이, 실시간 데이터의 특성 때문에 재전송을 통한 오류의 정정은 불가능할 수도 있다. 이와 같은 이유 때문에 멀티미디어 통신에 관한 최근의 연구에서는, 오류수정코드를 사용하거나 응용프로그램에서 오류를 처리하도록 하여, 수송계층에서 데이터에 대한 오류 제어를 하지 않는 경우도 있다(예, RTP[20]).

그러나, 본 연구에서는 데이터에 대한 오류 검사와 복구의 임무를 응용계층에서 수행하는 것은 통신 기능의 분담에 있어 옳지 않다는 입장이다. 또, 본 연구가 목표로 하는 실시간 수송계층 프로토콜은 멀티미디어 응용은 물론 신뢰성을 요구하는 다른 종류의 응용프로그램들도 지원하여야 하므로 오류 제어를 반드시 수행하여야 한다. RTTP의 오류 제어 원칙은 다음과 같이 요약된다.

1. 오류 제어는 사용자가 요구한 신뢰성의 한계에 접근하였을 때만 수행한다.
2. 오류 제어는 시간적으로 정정이 가능하다고 판단될 때만 수행한다.
3. 수신 측은 오류가 신뢰성의 한계에 접근하였을 때만 정상 도착 보고(ACK)를 보낸다.
4. 수신 측은 오류 발생시에는 오류 보고(NAK)를 보낸다.
5. 송신 측은 예상한 시간 안에 오류 보고가 없으면 오류가 발생하지 않은 것으로 간주한다.

첫번째 원칙은 음성 및 비디오 통신 응용의 예에서와 같이, 신뢰성의 한계 내에서 발생하는 오류는 사용자가 처리하도록 하자는 것이다. 두 번째 원칙을 제외한 나머지 원칙들은 오류 발생률이 매우 작은 고속 네트워크에서 정상적으로 도착할 때마다 응답을 보내는 것은 매우 비효율적이므로, 오류 발생 경우이거나 오류제어 수행 중에만 응답을 보내 제어 부하를 최소화하자는 취지이다. 두 번째 원칙은 실시간 데이터의 특성을 고려하여, 오류 정정이 불가능한 경우는 오류 제어를 수행하지 않고 이를 즉시 사용자에게 알리는 목적에서 세워진 것이다. 이렇게 함으로써, 불필요한 처리를 피할 수 있으며, 사용자가 발생한 예외 사건에 대처할 수 있는 시간적 여유를 갖을 수 있도록 한다.

발생한 오류가 신뢰성의 한계에 접근하였는 지를 판단하기 위해서는 현재의 신뢰성 상황에 대한 정보가 필요하다. (사용자의 요구가 100% 신뢰성을 요구한 경우는 항상 신뢰성 한계에 접근하였다고 간주한다.) 이를 위해, RTTP에서는 수신이 이루어질 때마다 오류의 발생 여부에 따라 신뢰성을 조정하도록 하였다. 또, 송신을 주로 하는 사용자(예, VoD 서버)의 경우를 위해, 송신을 할 때에도 신뢰성 구간마다 수신된 오류 제어 메시지를 검사하여 신뢰성 상황을 조정하도록 하였다. 신뢰성 구간과 신뢰성 한계치는 연결을 설정할 때 주어지는 사용자의 신뢰성 요구(R)를 기준으로 결정된다. 예를 들어 앞의 예처럼 사용자가 $R=(95\%, 5sec)$ 을 요구하면, 낙관적으로 신뢰성 구간은 5초, 신뢰성 한계치는 5%의 오류율로 정하거나, 또는, 비관적으로 2.5초와 4%로 각각 정할 수 있다. 한편, 한계치의 설정을 사용자가 임의로 정하도록 수

신 프리미티브를 확장할 수도 있다. 예를 들어, 수신 측에서 오류가 두 번 더 발생하면 신뢰성이 깨지는 상황이 되면, RTTP의 신뢰성 한계치를 설정하여 오류 제어를 하도록 할 수도 있다.

RTTP에서의 오류 수정의 가능 여부에 대한 판단 기준은 재전송할 메시지의 예상 도착시간이 사용자의 최대 허용 지연시간을 넘으면 불가능으로 판단한다는 것이다. 이는 개념적으로는 매우 간단하지만 실제로 구현하는데는 별도의 시간 정보 관리가 필요하다. 송신 측에서 재전송 여부에 대한 판단 기준을 수식으로 표현하면 다음과 같다.

average delay: 평균 지연 시간

max. delay: 최대 허용 지연 시간

send request time: 송신 요청 시간 (사용자가 전송을 요청한 시간)

current time: 현재 시간 (부정 응답을 받은 시간)

elapsed time: 송신 요청 후 소비한 시간 (current time - send request time)

laxity: 최대 허용 지연시간까지 남은 시간 (max. delay - elapsed time)

$$\left\{ \begin{array}{l} \text{laxity} \geq \text{average delay} \Rightarrow \text{retransmit} \\ \text{otherwise} \quad \quad \quad \Rightarrow \text{infeasible to recover} \end{array} \right.$$

평균 지연 시간이란 데이터가 송신측 RTTP에서 수신측 RTTP로 전송되는데 걸리는 평균적 시간을 말한다. 본 연구에서는 오류 검사 등 RTTP 내부에서 소요되는 시간은 지연 시간에 포함시키지만, RTTP와 사용자 사이의 데이터 이동 시간은 포함하지 않도록 정의한다. (사용자와 RTTP가 같은 메모리 공간을 사용하므로 포인터를 사용하면 데이터 이동 시간은 무시할 수 있을 정도로 작을 것으로 판단한다.) 평균 지연 시간은 데이터와 함께 보내지는 시간 정보와 측정 시간을 이용해 수신측에서 계산되며, 이는 RTTP 내의 시간 정보 교환 프로토콜에 의해 송신측에게 전달된다.

송신과 수신 양쪽의 전송시간이 같다고 가정한다면, 위의 오류제어는 평균 전송시간이 최대 허용 지연 시간의 1/3 미만일 때 효과적일 수 있다. 또, 한번의 오류제어를 하지 않음으로써 얻는 성능 향상이 작다 하더라도, 예외사건을 미리 알아내어 처리할 수

있는 시간을 얻을 수 있다. 결국, 두 번째 원칙의 효율성은 이와 같은 효과와 시간 정보를 유지하기 위해 필요한 부하의 비로 결정되며, 최대 허용 지연 시간과 평균 지연 시간과의 차이가 주요 요소가 된다. 최대 허용 지연 시간과 평균 지연 시간과의 관계는 실제 네트워크의 운용 등과 같은 많은 연구가 필요한데, 현재 발표된 결과를 보면[6], RTTP가 효율적일 가능성이 높은 것으로 판단된다.

RTTP의 오류 제어 방법은 매우 간단하며, 오류 발생이 신뢰성의 한계보다 작게 유지 되는 경우 상황 정보의 교환 이외는 부하가 없다. 그러나, 한계 상황에 이르러서 시작되는 오류 제어가 실제로 오류를 복구시키는가에 대해서는 의문이 있을 수 있다. 첫 번째 문제는 송신 측에서 한계 상황을 깨닫는 시점이 실제로는 이미 신뢰성이 깨어진 상태이거나 한계 상황을 벗어난 경우일 수 있다는 점이다. 두 번째 문제는 사용자에게 보고되는 응답이 실제 사실과 일치하지 않을 수 있다는 점이다. 이들은 낙관적인 오류 제어 방법들이 갖는 공통적인 부정확성의 문제들로서, 낙관적 오류 제어의 범주에 속하는 RTTP 오류 제어도 마찬가지가 되는 것이다.

그러나, RTTP는 한계치의 값과 신뢰성 상황 검사 주기를 조절함으로써, 오류 제어의 부정확성을 조절할 수 있다. 한계치의 값을 낮게 설정하면 필요 없는 오류 제어를 수행할 수도 있지만 오류를 복구할 가능성도 높아지게 된다. 마찬가지로 신뢰성 상황 검사 주기를 짧게 하면 오류 제어를 수행하는 시점이 상대적으로 정확해지며 올바른 오류 복구가 이루어질 가능성이 높아지게 된다. 사용자가 연결 설정시에 제공하는 신뢰성 요구 사항은 이 값들을 결정하는 기준으로만 사용하며, 실제의 값은 네트워크 계층 연결의 신뢰성, 성능 감시를 통해 측정된 결과 등을 고려하여 RTTP가 결정한다. 예를 들어, 평균적인 오류율을 측정하고, 이것이 주어진 사용자의 신뢰성 요구에 접근하면 한계치를 낮추고 검사 주기를 짧게 하며, 여유가 있으면 한계치를 높이고 검사 주기를 늘린다. 한편, 신뢰성의 한계에 이르러서 수행될 수 있는 불필요한 오류 제어와 그에 따른 대역폭의 낭비는 신뢰성이 다시 복구되는데 걸리는 시간 동안에만 부과되는 한시적인 비용이 된다.

정리를 하면, RTTP와 오류 제어가 갖는 장점은 한

제치를 조정하는 함으로써 오류 제어의 부하와 오류 복구의 성공을 조절할 수 있다는 점이다. 이에 따라, 100%의 신뢰성을 요구하는 응용에서부터, 평균적인 신뢰성을 요구하는 응용까지 모두 적용이 가능하게 된다. RTTP 오류 제어 방법의 실제적인 적절성은 많은 실험을 통하여 검사되어야 할 것이다.

3.5 기타의 기능들

흐름 제어 RTTP에서는 연결 설정 시에 정의되는 사용자 요구 사항에 따라 송신/수신/네트워크에서 모두 성능을 보장하므로, 별도의 흐름제어는 수행하지 않는다. 데이터의 원활한 흐름에 문제가 생기는 유일한 경우는 송신자가 연결 설정 시에 약속한 사항을 지키지 않고 과도한 데이터를 전송하는 경우이다. 이에 대한 RTTP의 입장은 계약 위반의 책임은 계약을 위반한 측에서 져야 한다는 것이다. 즉, 과도한 데이터에 의해 발생하는 성능 지원의 실패는 송신자, 즉, 사용자 자신이 해결하여야 한다는 것이다. 성능 보장을 지원하는데 필수적인 교통량 형성(traffic shaping)과 폴리싱(policing) 기능은 품질인자의 보장을 지원하는 네트워크에서는 반드시 지원될 것이므로, RTTP에서 이들을 중복하여 지원할 필요는 없다고 판단된다.

분해/재조합 수송계층이 어느 특정한 네트워크계층만을 지원하는 것이 아니라면, 분해/재조합의 기능은 필수적이다. RTTP의 분해/재조합 기능은 기존의 수송계층의 분해/재조합 방법을 그대로 따르도록 한다. 여기서 한가지 유의할 사항은 메시지를 분해할 때 생기는 사용자 요구의 변화도 변화된다는 점이다. 예를 들어, 최대 허용 지연시간이 D 인 메시지를 분해하는 경우, 분해된 메시지외 마지막 조각은 앞의 조각들이 모두 전송된 뒤에 전송되므로 최대 허용 지연시간이 줄어들어야 한다. RTTP는 이와 관련된 Ferrari의 "사용자 요구의 변환 방법[8]"을 이용하여 이 문제를 해결하도록 한다.

시간 정보 첨부 수신 측에서 수신된 데이터가 사용자의 전송시간 요구에 부합하는 지를 판단하기 위해서는 데이터가 송신 요청된 시간 등에 관한 정보가 필요하다. 이와 같은 정보는 송신 측에서 데이터를 송신할 때 같이 보내주어야 한다. 이를 일반적으로 말하면, 실시간 수송계층 프로토콜에서는 시간 정보가 일반 제어 정보의 일부분으로서 반드시 필요하다

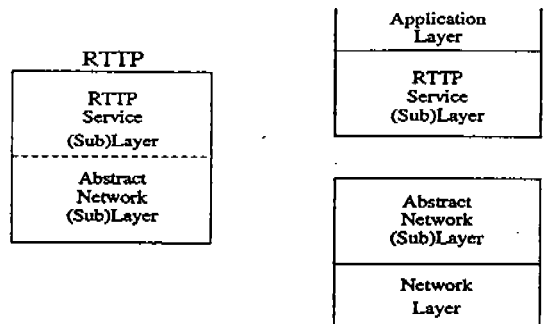
고 할 수 있다.

실시간 통신에 필요한 시간 정보는 여러 가지가 있을 수 있으며, RTP[20]는 이러한 예들을 잘 보여주고 있다. RTTP에서 데이터를 송신할 때마다 필요한 시간정보는 "송신 요청 시간 (send request time)"이며, 이는 RTTP 헤더의 일부분으로 전송된다. 그 외의 시간 정보들은 별도의 제어 메시지를 통해 교환되도록 한다. 한편, 송신 측에서 보낸 시간 정보를 수신 측이 올바르게 해석하기 위해서는 양쪽의 시각이 동기화되어 있어야 한다. 본 연구에서는 통신을 수행하는 양쪽이 NTP[18] 등을 이용하여 동기화되어 있다고 가정한다. 향후의 연구에서 시간 정보 교환을 통해 필요한 동기화를 자체적으로 이루도록 RTTP를 확장할 계획이다.

4. RTTP 구조

4.1 가상 네트워크계층 모델

RTTP는 논리적으로는 그림4의 원련에 보여진 바와 같이, 네트워크에 독립적으로 기능들을 수행하여 RTTP 프리미티브를 구현하는 "RTTP 서비스 계층"과, 실제 네트워크가 제공하는 서비스들을 이용하여 RTTP 서비스 계층이 가정하는 서비스를 구현하는 "가상 네트워크계층"의 두 부계층으로 구성된다. 본 연구에서는 RTTP 서비스 계층의 개발과 가상 네트워크계층의 서비스 정의에 중점을 둔다. 가상 네트워크계층을 구현하는 것은 ATM 네트워크 위에서의 구현 예를 통해 가상 네트워크계층 접근 방식이 타당하다는 점만을 보이기로 한다.



(그림 4) RTTP의 논리적 구조 및 구현 구조 (Fig. 4) Logical and Implementation Architecture of RTTP

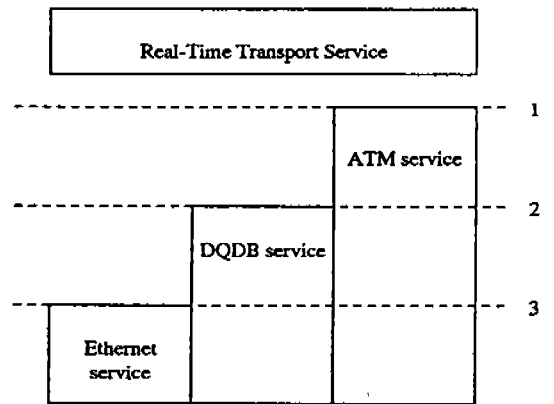
그림 4의 오른쪽은 RTTP의 실제 구현 구조를 나타낸 것이다. 이와 같은 구현 구조의 중요한 의미는 RTTP가 프로시듀어 형태로 이루어져야 구성되어야 한다는 점이다. 이는 실시간 통신에서 필수적인 프로토콜 처리 시간의 예측을 가능하게 하기 위해 의도적으로 결정한 것이다. 그러나, 프로토콜을 프로시듀어 형태로 구현하면 프로토콜 처리에 여러 제약을 갖는다. 특히, 모든 처리가 사용자 계층의 명령에 의해 시작됨으로 제어 정보의 처리와 같은 RTTP 자체의 처리는 어렵게 된다. 프로시듀어 형태로 역시 구현되는 RTP[20]에서는 제어용 프로토콜인 RTCP가 따로 있어 이와 같은 문제를 해결하고 있다. RTTP는 이와 같은 제어 프로토콜 역시, 프로토콜 처리 시간의 예측을 스케줄링까지 포함하는 문제로 복잡하게 만들 우려가 있으므로 이를 적용하지 않는다. 대신, 데이터의 송수신 사이에 제어 정보를 처리하는 방법을 채택하였다.

4.2 가상 네트워크계층 정의

실시간 통신에 대한 요구가 증대됨에 따라 네트워크에서도 교통량제어(traffic control) 등을 통한 성능 보장을 통해 실시간서비스를 제공하는 네트워크들이 많이 연구 및 개발되었다. 연구의 방향은 크게 들로 나눌 수 있는데, 첫째, 기존의 네트워크에 제어기법을 추가하여 실시간서비스를 구현하는 방법과 둘째, 새로운 접근방식의 네트워크를 구축하는 방법이다. 첫 번째 방법의 예로는 대표적인 비실시간 네트워크인 ethernet에서 전역시간(global time)을 이용하여 기본적인 실시간 통신을 제공하는 방법이 연구 등이 있다 [24]. 두 번째 방법의 예로는 Timed token protocol[10], multiple token ring[16] 등이 연구되었으며, FDDI, DQDB, ATM 등은 널리 보급되고 있다.

새로이 개발된 네트워크에서 실시간서비스를 제공하는 방법은, 각각의 사용자가 전송할 데이터가 있어서 네트워크에 요청하면 어떤 유한한 시간 안에 네트워크를 사용할 수 있도록 하는 것을 근간으로 한다. 또, 이를 위해서는 각각의 사용자는 제한된 범위에서만 네트워크를 사용하도록 제어되어야 한다. 이와 같은 한정된 네트워크 접근시간(bounded network access time)을 구현하기 위해, FDDI에서는 토큰보유시간(token holding time)을 두어 접근시간을 예측할 수 있

도록 하거나, DQDB와 ATM에서는 사용자가 전송률을 네트워크에 미리 예약(reservation)하도록 하고 네트워크는 이를 접수하고 보장하는 방식을 채택하고 있다. 가상 네트워크계층은 우선 실시간 수송계층이 실시간서비스를 구현하는데 충분한 기능을 제공하여야 한다. 또, 다양한 네트워크들을 감안하여 실시간서비스를 제공하기 위한 작업들의 공통분모를 찾아서 정의되어야 한다. 가상 네트워크계층의 기능 수준을 결정하는 문제를 그림으로 설명하면 그림 5와 같다.



(그림 5) 가상 네트워크계층 서비스의 수준
(Fig. 5) Level of Abstract Network Layer Service

그림 5와 같이 네트워크계층이 제공하는 실시간서비스들의 수준이 다른 경우, 이론적으로 알맞은 선택은 (2)가 될 것이다. 그러나, 미래에 가장 보편화될 ATM에게는 (2) 수준의 가상 네트워크계층은 중복처리의 부하를 가져오며, 경우에 따라 실시간 기능의 부분적 감소를 가져올 수도 있다. 따라서, 본 연구의 가상 네트워크계층은, 현실적인 선택으로서, (3)과 같이 ATM의 실시간서비스 수준의 기능을 갖도록 정의한다.

이와 같은 결정의 한가지 문제점은 IP와의 관계이다. IP는 ATM과 마찬가지로 향후에도 계속 사용될 필수적인 프로토콜이라는 점은 널리 인정되고 있다. 또, "IP over ATM"이라는 이름 아래 ATM 위에 IP를 구현하여 현존하는 컴퓨터네트워크와 ATM망과의 통신을 제공하려는 연구가 매우 활발히 진행되고 있다[5]. 계층구조 면에서 볼 때, 가상 네트워크계층은 IP의 상위계층이므로 IP의 실시간 기능을 이용하여

가상 네트워크계층을 구현하여야 한다. 그러나, 현재의 IP는 실시간 기능이 거의 없다고 볼 수 있으며, 진행 중인 IP 확장에 관한 연구에서도 본 연구에서 요구하는 수준의 실시간 기능 추가는 매우 미약하다. 결국, 실시간서비스를 위해 별도의 프로토콜 스택을 사용하지 않는다면, 중복처리의 문제가 발생하게 되며 가상 네트워크계층의 수준도 재고려 되어야 할 것이다. 최근 활발히 연구되고 있는 "Integrated Services"[3]가 이와 관련된 전체적인 해결책을 제시할 것으로 기대된다.

RTTP에서 정의하는 가상 네트워크계층의 서비스는 성능을 보장하는 연결 위주의 통신이다. 서비스를 구체적으로 정의하는 품질인자들은 다음과 같다. 정의는 ATM 네트워크의 서비스 수준을 고려하였으며 [1], 향후의 연구와의 호환성을 위해 Integrated Services 그룹의 "Network Element Service Specification"[23]을 참조하였다.

- 최대 전송비율 (peak rate): r_p
- 평균 전송량 (선택 항목)
 - 평균 전송비율 (average rate): r_a
 - 최대 연속크기 (max. burst size): b
- 전송시간
 - 최대 전송시간: d
 - 최대 전송시간변이: j
- 허용 오류율 (max. acceptable error rate): e

5. RTTP의 설계

5.1 RTTP PDU 구조

RTTP의 PDU 구조는 그림 6과 같다. 각 필드 별로 그 의미를 설명하면 아래와 같다.

- Version 향후의 개정과 호환성을 위한 RTTP의 버전
- Type PDU의 종류를 나타낸다.
- Sequence Number 패킷의 상실, 재전송, 재배열 등 오류제어에 사용된다.
- Length 현 PDU의 길이를 나타낸다.
- Timestamp 송신 측이 사용자로부터 데이터를 받은 시각을 나타낸다. 이를 통해 수신 측은 전체 지연 시간을 계산하여 시간 오류의 여부를 판정한다.(송

신 측과 수신 측은 별도의 시간 동기화 방법 또는 시간 정보의 교환을 통해 동기화가 이루어져 있다고 가정한다.)

- Checksum 수신된 PDU의 오류 검출에 사용된다.
- Data/Control PDU 종류에 따라 데이터 또는 제어 정보를 담게 되며 가변 길이이다.

	0	8	16	31
Version	Type	Sequence Number		
Length				
Timestamp				
Checksum				
Data/Control				

(그림 6) RTTP의 PDU 구조
(Fig. 6) Structure of RTTP PDU

5.2 RTTP 프로토콜 처리

RTTP의 구현이 (즉, RTTP 서비스계층의 구현) 프로시저어 형태로 이루어지므로, 프로토콜의 처리는 사용자에게 제공되는 RTTP 서비스를 구현하는 프로시저어에서 이루어진다. 본 장에서는 RTTP의 특징적 기능인 품질인자 변환, 오류 제어 등을 연결 설정 및 데이터 송수신 프리미티브의 구현을 통해 설명하기로 한다.

(1) 연결 설정 서비스

RTTP의 연결 설정 과정에서 가장 중요한 작업은 네트워크계층의 연결을 설정하기 전에 사용자의 요구 사항들을 가상계층의 품질인자들로 변환하는 품질인자 변환이다. 이를 수시적으로 표현하면 다음과 같다.

$$\begin{aligned}
 & \text{QoS Translation}(T): UR \rightarrow QoS_{AN} \\
 & \text{where } UR(\text{Use Requirements}) = (C, P, S, R, T) \\
 & \quad QoS_{AN}(\text{QoS of Abstract Network Layer}) \\
 & \quad = (r_p, [(r_a, b)], (d, j), e)
 \end{aligned}$$

RTTP의 품질인자 교환 규칙은 그림7과 같다. 대부

본의 교환 규칙은 매우 간단한데 그 이유는 가상 네트워크계층의 수준을 높게 잡았기 때문이다. 메시지의 분할 경우는 앞에서 언급한 바와 같이 Ferrari의 연구[8]를 적용하면 된다.

$$r_p = \begin{cases} p^s \times l & \text{if } S = l \text{ (고정크기 경우)} \\ p \times l_{max} & \text{if } S = [, l_{max}] \text{ (가변크기 경우)} \end{cases}$$

$$r_a = p \times l_{avg} \text{ if } S = (, l_{avg}) \text{ (평균크기가 주어진 경우)}$$

$$b = (l_{max} \times n) \text{ if } S = (, l_{avg} , (n, n))$$

$$d = D - C_{RTTP}$$

where C_{RTTP} is the processing time of RTTP
(i.e., the sum of the execution times of send_data and receive_data)

$$j = J$$

$$e = 1 - R$$

*이루기에 불성공인 경우는 최소 통신량 전체, 이를 (우기) 대신에 복수하여 면한다.

(그림 7) RTTP의 품질 변환 규칙
(Fig. 7) QoS Translation Rules of RTTP

(2) 데이터 전송 서비스

RTTP의 데이터 전송에 있어서 가장 특징적인 기능은 오류 제어이다. 메시지의 분할 및 재조합 등의 기능은 다른 수송계층 등과 유사하므로 본 연구에서는 생략하기로 한다.

오류 제어를 포함한 데이터의 송신은 사용자가 RTTP 프리미티브인 send_data를 수행함으로써 이루어진다. 그림 8은 send_data의 코드 요약을 나타낸 것이다. 그림에서 보이는 바와 같이, RTTP는 메시지의 송신을 마친 후, 주기적으로 제어 메시지를 검사하여 오류 상황을 판단한다. 신뢰성 유지에 문제가 없으면, 실제 전송 상황에 관계없이 즉시 사용자에게로 성공 응답을 보내어 사용자가 다음 작업을 할 수 있도록 하며, 오류가 미리 정해 놓은 신뢰성 한계치에 이르면 오류 제어를 시작한다. 오류 제어의 방법은 앞의 RTTP 기능에서 이미 설명하였으므로 생략하기로 한다.

데이터의 수신은 사용자가 RTTP 프리미티브인 receive_data를 수행함으로써 이루어진다. 그림 9는 receive_data의 코드 요약을 나타낸 것이다. RTTP는 메시지를 네트워크계층으로부터 넘겨 받은 후, 전송 시간을 포함한 오류의 유무를 검사한다. 오류 제어는 송신 측에 대응하여 동작하며, 실제 오류가 발생하였거나 신뢰성 한계치에 이르렀을 때만 수행한다. 또,

오류가 발생하였더라도 그 종류와 신뢰성 상황에 따라 사용자에게 수신 실패를 알려거나, 오류 종류와 함께 데이터를 넘겨주어 사용자가 이를 처리하도록 한다.

```

send_data(destID, message, userTO)
{
    make a rttp_msg by prepending a header to the message ;
    send_to_network(destID, rttp_msg) ;

    if (reliability interval expired) {
        while (more message) {
            if it is a response
                read and update reliability-threshold ;
            else skip; /* leave the message and try the next */
        }
    }
    if ( reliability-threshold is not reached )
        return success ;
    else { /* must do error control */
        timeout = min(2 x avgdelay, maxdelay, userTO) ;
        wait for the response for timeout ;
        if (a response arrived) {
            if (response is ACK) {
                update reliability-threshold ;
                return success ;
            }
            else {
                while ( neither maxdelay nor userTO is not elapsed ) {
                    retransmit ;
                    timeout = min(2 x avgdelay, maxdelay, userTO) ;
                    wait for the response for timeout ;
                    if (ACK response arrives) {
                        update reliability-threshold ;
                        return success ;
                    }
                }
            }
            if (timeout = userTO) return timeout ;
            else return fail ;
        }
    }
    else
        if (timeout = userTO) return timeout ;
        else return fail ;
}
    
```

(그림 8) RTTP의 send_ata 처리
(Fig. 8) Algorithm of send_data in RTTP

```

receive_data(connectionID, msgbuf, userTO)
{
    read_from_network(destID, rttp_msgbuf) ;
    check if the rttp message is late or if it has an error;
    if ( no error ) {
        if ( reliability-threshold reached ) /* sender is waiting for ACK */
            send a ACK response ;
        update reliability-threshold ;
        move data from rttp_msgbuf to msgbuf ;
        return success ;
    }
    else {
        update reliability-threshold ;
        send a NAK response ;
        if ( reliability-threshold reached ) {
            laxity = maxdelay - (current time - timestamp) ;
            if ( laxity < minimum round-trip time) return fail ;
            /* wait for retransmission from the sender up to the user timeout */
            timeout = min(laxity, userTO) ;
            while ( timeout is not expired ) {
                
```

```

peeking incoming rtpp messages
if (a retransmitted message) {
    move data from rtpp.msghbf to msghbf;
    return success;
}
}
return fail;
}
else { /* it is an acceptable error */
    if ( bit error ) move data from rtpp.msghbf to msghbf;
    return error_type;
}
} /* end of error case */
}
    
```

(그림 9) RTTP의 receive_data 처리
(Fig. 9) Algorithm of receive_data

5.3 ATM 네트워크에서 가상 네트워크계층의 구현

가상 네트워크계층을 구현하는 일은 실제 네트워크가 제공하는 서비스를 이용하여 가상 네트워크계층 서비스를 구현하는 것과 같다. 본 절에서는 실시간 기능을 갖춘 대표적인 네트워크인 ATM을 이용하여 가상 네트워크계층을 구현하는 작업에 대해 간략히 설명하도록 한다.

ATM은 가상채널 (Virtual Channel)을 통해 연결 위주의 통신을 제공하며, 제공하는 품질인자도 가상 네트워크계층의 품질인자와 유사하다. 이와 같이 ATM 네트워크의 서비스가 가상 네트워크계층의 서비스와 매우 유사하므로 ATM 위에서 가상 네트워크를 구현하는 일은 품질인자의 변환이 그 주요 업무가 된다.

ATM 네트워크의 사용자 인터페이스로서 현재 가장 많이 사용되는 것은 기존의 TCP/IP를 통해서 사용하는 것이다. 그러나, 이 경우 앞에서도 설명한 바와 같이 ATM의 QoS 기능은 상실되고 만다. ATM 장비 제작자에 의해 제안된 사용자 인터페이스에도 ATM의 QoS 기능을 모두 지원하는 것은 드물다.(예를 들어, FORE System의 인터페이스는 최대 전송률만 지원된다[9].) 따라서, 본 연구에서는 구체적인 서비스 구현 방법 대신에, 가상 네트워크계층과 ATM 사이의 품질인자의 변환 규칙을 제시하는 것으로 가상 네트워크계층의 구현을 설명하도록 한다. ATM의 품질인자는 ATM Forum의 UNI정의[1]를 따랐다. 가상 네트워크계층의 품질인자 교환 규칙은 그림 10과 같다.

6. 관련 연구와의 비교

실시간 통신에 관해 지금까지 연구된 내용들을 살펴보면, 대부분은 네트워크 내부에서 성능 보장을 하는 실시간 네트워크에 관한 것들이다[10, 16, 24]. 본 연구에서는, 이들 연구들이 단순히 가정하였던, 사용자를 포함한 호스트 쪽에서의 문제들을 연구하였다.

최근의 몇몇 연구에서는 통신의 양끝 간에서 주어진 마감시간(deadline)까지 메시지 전달을 보장하는 실시간 수송계층 서비스에 관해 다루고 있다.[14]에서는 "real-time channel"이라는 실시간 특성을 갖는 연결 서비스를 연구하였으며, Tenet 그룹에서도 유사한 서비스를 실제로 구현하여 실험하였다[2]. 또, 기존의 수송계층에 실시간 서비스를 추가하려는 노력도 진행 중이다[12]. 그러나, 위의 연구들은 네트워크를 통해 성능 보장을 어떻게 하느냐에 초점을 맞추었으며, 수송계층은 단순히 제안된 네트워크 계층 실시간 서비스를 사용자에게 제공하는 역할만을 한다. 따라서, 실시간 수송계층은 네트워크 계층에 의존하는 형태

교통량 설명자(Traffic Descriptor)의 연자들은 다음과 같이 변환된다.

$$PCR = \frac{53}{48} \times (1 + OV_{AAL}) \times r_p$$

$$CDV = j$$

$$SCR = \frac{53}{48} \times (1 + OV_{AAL}) \times r_s$$

$$MSB = \lceil \frac{48 \times (1 + OV_{AAL})}{48} \rceil$$

여기서 PCR, CDV, SCR, MSB는 연결 설정 과정에서 사용되는 ATM의 Traffic Descriptor의 인자들로서, Peak Cell Rate, Cell Delay Variation, Sustainable Cell Rate 및 Maximum Burst Size in number of cells를 각각 나타낸다.

QoS 인자들인 CER (Cell Error Ratio), CLR (Cell Loss Ratio), CTD (Cell Transfer Delay) 등은 다음 조건이 만족하는 범위에서 선택한다.

$$\sum(\text{error ratio}) \leq \epsilon - \delta$$

where δ is a room for late transmission

$$CTD \leq d - (M - 1) \times \frac{1}{PCR}$$

where M is the maximum number of cells for an Abstract Network Layer Packet

$$\text{(for a 64K packet, } M = \lceil \frac{64000(1 + OV_{AAL})}{48} \rceil \text{)}$$

* 해당 AAL 처리 비용 즉, $OV_{AAL} = \frac{AAL \text{ overhead}}{AAL \text{ data unit}}$

(그림 10) ATM 품질인자로의 변환 규칙

(Fig. 10) Translation Rules to ATM QoS Parameters

를 지니며, 결국 응용계층도 영향을 받는 결과를 가져왔다. 본 연구는 네트워크에 독립적인 실시간 수송계층을 만드는데 초점을 맞추고 있으며, 예측 가능성을 위한 구현 구조를 제시하고 있다.

최근에 IETF를 중심으로 미래의 통신서비스로서 Integrated Service를 정의하는 연구가 진행하고 있으며, 이 중, 실시간 서비스로서는 "guaranteed service", "predictive service" 등이 정의되고 있다. 그러나, 여기서 정의되는 서비스들은 사용자가 직접 이용하는 수송계층 수준의 서비스보다는 네트워크계층의 서비스에 속한다. 따라서, 이들은 본 연구의 가상 네트워크계층 정의에 가깝다고 할 수 있다.

한편, 멀티미디어 회의 시스템을 주응용으로 하는 실시간 수송계층 프로토콜의 표준안을 만들려는 노력도 진행 중이다(RTP)[20]. 내부 기법에 있어서, RTP는 본 연구와 유사점을 갖고 있지만, 오디오/비디오 응용을 지원하는 것에 그 목적을 두고 있다. 따라서, 본 연구와는 달리 연결 위주의 서비스가 아니며, 오류 제어와 성능 보장보다는 멀티캐스팅 등에 비중을 두고 있다. 또, 사용자 요구에 대한 지원도 한정적이다.

결론적으로 실시간 통신은 최근 활발히 연구되기 시작하였지만 대부분의 연구는 네트워크 계층에 초점을 맞추고 있으며, 수송계층에 관해 아직 체계적인 결과는 나와있지 않은 상태이다. 그러나, 사용자가 이용할 수 있는 실시간 통신을 위해서는 실시간 수송계층 자체에 대한 연구도 반드시 필요하다. 본 연구는 이 문제를 중점적으로 다루고 있다고 할 수 있다.

7. 결 론

본 연구에서는 우선 사용자 요구에 대한 분석을 통해, 여러 응용에서 사용될 수 있는 일반적인 형태의 실시간 수송계층 서비스를 정의하였다. 다음, 이의 구현 방법으로서 가상 네트워크계층의 개념을 도입하였으며, 프로토콜 처리 시간의 예측 가능성을 높이기 위해 라이브러리 함수로 구현하는 방법을 채택하였다. 마지막으로, 실시간 수송계층 구현에 내부적으로 필요한 품질인자 교환 방법을 개발하였으며, 효율적이고 융통성 있는 오류 제어 방법을 개발하였다.

본 논문에서 설계한 RTTP는 그 기능으로서 "네트워크 계층 연결의 사용자를 위한 포장"이라는 관점을

갖고 있다. 따라서, 설계 단계에서 별도로 정확성을 검증하여야 할 기능은 없다고 판단된다. 본 논문에서 제안된 오류제어 방법은 기본적으로 재전송에 기초하고 있으므로 정성적인 검증은 큰 의미가 없다고 판단된다. 논문의 중요 내용은 사용자 프리미티브의 정의, 구현 방법 및 구체적인 설계가 되는데, 이들에 대한 타당성 검증은 프로토콜의 실제 구현과 응용 프로그램 작성을 통한 실험을 통해서 이루어져야 한다고 생각된다.

본 연구의 특징은 실시간 통신에 관한 호스트 쪽에서의 제반 문제들을 일반적이면서도 실제 구현이 가능한 방법으로 해결하였다는 점이다. 따라서, 향후 실시간 시스템 및 멀티미디어 시스템에 관한 기반 연구로서 활용될 수 있으리라 기대한다. 또, 이와 같은 경험을 바탕으로 현재 진행되고 있는 수송계층의 프로토콜 표준안 작성에도 참여할 수 있다고 기대된다.

향후에 계속되어야 할 연구는 첫째, 개발된 방법들의 실효성을 점검하는 일이다. 즉, 설계된 RTTP의 시제품을 구현하고 실험하는 연구이다. 실시간 기능은 다른 통신 기능과는 달리 간단한 시뮬레이션을 통해 검사하기가 어렵다. 따라서, 실질적인 검사를 위해서는 기존의 시뮬레이션 환경을 (예, x-kernel[11]) 확장하거나, 실시간 네트워크를 구축하는 등의 장기간에 걸친 연구가 필요할 것이다. 둘째, RTTP의 기능들을 개선하는 연구이다. RTTP의 오류 제어 및 멀티플렉싱에서 설명한 바와 같이, 다른 처리 기법에 관해 좀 더 깊이 있게 연구하고 이를 현재의 기법과 비교하여 RTTP를 효율적으로 개선하는 연구가 계속되어야 할 것이다. 마지막으로, RTTP의 기능을 확장하는 일이다. 예를 들어, 많은 실시간 응용이 요구하는 멀티캐스팅 기능을 RTTP에 포함시키는 연구 등이 필요하다.

참 고 문 헌

- [1] ATM Forum, *ATM: User-Network Interface Specification*, Prentice Hall, 1993.
- [2] A. Banerjee and et al., "The Tenet Real-Time Protocol Suite: Design, Implementation and Experience," TR-94-059, Univ. of California at Berkeley, Nov. 1994.
- [3] R. Braden and et al., "Integrated Services in the

Internet Architecture: an Overview," Internet Draft, Integrated Services WG, 1994.

[4] R. Braden and et al., "Resource Reservation Protocol (RSVP) Version 1," Internet Draft, Integrated Services WG, July 1995.

[5] R. Cole , "IP over ATM: A Framework Document," Internet Draft, 1994.

[6] B. Depmsey, J. Liebeherr and A. Weaver, "On Retransmission-Based Error Control for Continuous Media Traffic in Packet-Switching Networks," Tech. Report, Computer Science Department, Univ. of Virginia, 1994.

[7] D. Ferrari and Verma, "A Scheme for Real-Time Channel Establishment in Wide-Area Networks," *IEEE JSAC*, 8, 3, PP 368-379, 1990.

[8] D. Ferrari, "Client Requirements for Real-Time Communication Services," RFC 1993, Nov. 1990.

[9] FORE Systems, *ForeRunner SBA-200 ATM Sbus Adapter User's Manual*, FORE Systems, Inc., 1995.

[10] R.M. Grow, "A Timed Token Protocol for Local Area Networks," *Proc. Electro/82, Token Access Protocols*, May 1982.

[11] N. Hutchinson and L. Peterson, "The x-kernel: An architecture for implementing network protocols," *IEEE TRSE*, 17, 1, pp. 1-13, Jan. 1991.

[12] ISO/IEC JTC1/SC6/WG4 N811-Telecommunications and Information Exchange Between Systems, "Proposed Draft Text for a High-Speed Transport Protocol(HSTP) Specification," Dec. 1992.

[13] J. Jung and D. Seret, "Translation of QoS Parameters into ATM Performance Parameters in B-ISDN," *Proc. of INFOCOM 93*, pp. 748-755, 1993.

[14] D. Kandler, K. Shin and D. Ferrari, "Real-time Communication in Multi-hop Networks," *Proceedings of 11th International Conf. on Distributed Computing Systems*, May 1991.

[15] J. Kurose and et al., "Multiple-access protocols and time-constrained communication," *ACM Computing Surveys*, 16, 1, pp. 43-70, March 1985.

[16] Y. H. Lee and Li-Tao Shen, "Real-Time Communications in Multiple Token Ring Network," *Proc. of Real-Time System Symposium (RTSS)*, pp. 146-155, 1990.

[17] S. Lim and et al., "An Accurate Worst Case Timing Analysis for RISC Processors," *IEEE Transactions on Software Engineering*, 21, 7, pp. 593-604, 1994.

[18] D. Mills, "Network Time Protocol (NTP)," RFC 1305, Internet Engineering Task Force, April 1992.

[19] C. Partridge, *Gigabit Networking* Addison-Wesley, 1994.

[20] H. Schulzrinne and et al., "RTP: A Transport Protocol for Real-Time Applications," Internet Draft, Audio-Video Transport WG, 1995.

[21] L. Sha and et al., "Scheduling Real-Time Communication on Dual-Link Networks," *Proc. of RTSS*, pp. 188-197, 1992.

[22] S. Shenker and C. Partridge, "Specification of Guaranteed Quality of Service," Internet Draft, Integrated Services WG, 1995.

[23] S. Shenker and J. Wroclawski, "Network Element Service Specification Template," Internet Draft, Integrated Services WG, 1995.

[24] W. Zhao and K. Ramamritham, "Virtual Time CSMA Protocols for hard real-time communications," *IEEE TRSE*, 13, 8, pp. 938-952, Aug. 1987.

[25] 강현국, "멀티미디어를 지원하는 상위계층 프로토콜," '93 하계 컴퓨터 통신 Workshop 논문집, pp. 136-148, 1993.

[26] 박승철, 최양희, "멀티미디어 네트워크의 프로토콜 설계," '93 하계 컴퓨터 통신 Workshop 논문집, pp. 149-166, 1993.



박 창 운

- 1984년 서울대학교 공과대학 전자계산기공학과 졸업(학사)
- 1986년 서울대학교 대학원 전자계산기공학과(석사)
- 1992년 Univ. of Washington, Dept. of Computer Science(공학박사)

1992년~1993년 Univ. of Michigan, Dept. of EECS, Post-Doc

1993년~현재 중앙대학교 공과대학 컴퓨터공학과 조교수

관심분야: 컴퓨터네트워크, 실시간 시스템