

멀티미디어 멀티캐스트를 위한 효율적인 피드백 정보 수집

궁 상 환[†] · 강 민 규[†] · 구 연 설^{††}

요 약

본 연구의 목적은 멀티캐스트 통신에서 송신자가 멀티미디어 데이터를 한 그룹의 구성원들에게 전송할 때, 그룹내의 수신자들이 전송된 데이터를 잘 수신하고 있는가에 대한 상황을 정기적으로 확인하여 이질적인 단말 및 통신망 환경에서 효율적인 멀티미디어 서비스를 제공하고자 하는 데 있다. 특히, 연구에서는 송신자의 전송 데이터를 수신자가 어떻게 수신하고 있는가에 대한 상태를 파악하는 과정에서, 수신자들의 전송 정보를 절감 시킴으로써 송신자 측에서 상태 정보의 폭주로 인한 성능 저하를 개선하는 데 초점을 두고 있다.

본 연구의 기본 아이디어는 그룹의 행위를 어떻게 보는가의 관점에서 출발하는 데, 제안된 방식에서는 그룹의 구성원 중 가장 열악한 상황에 있는 수신자 즉, 대표 수신자가 그룹의 상황을 대표하는 것으로 가정한다. 다시 말해, 이 대표 수신자의 성능이 저하되면 전체 그룹의 성능이 저하된 것으로 간주하고, 반대로 대표수신자의 성능이 향상되면 전체 그룹의 성능이 향상되는 것으로 간주하게 된다.

제안 알고리즘에서는 대표 수신자가 다른 수신자 보다 먼저 상태 정보를 송신자에게 통보 하도록 하며, 다른 수신자들은 이 대표 수신자의 상태 정보를 수신할 때 자신의 상태와 비교하여, 자신이 더욱 열악한 경우에만 추가적인 응답을 수행하도록 함으로써 불필요한 응답을 제거하고자 하는 것이다.

An Efficient Feedback Collection for Multimedia Multicast

Sang Hwan Kung[†] · Min Gyu Kang[†] · Yeon Seol Koo^{††}

ABSTRACT

The purpose of this study is to enhance the quality of multimedia service under the heterogeneous end-terminals and network environment by monitoring the data receiving status of the group members regularly when the sender multicasts real-time multimedia data to a group. Especially, it focuses to reduce the total number of status information responded to the sender from the receivers. Accordingly, it upgrades the sender's performance by suppressing the number of status information packets generated by the receivers.

The key idea of this study starts from how we profile the activity of all the members in a group. We assume that the receiver status in the worst status, so called primary receiver, represents the status of the whole group. This means that the whole group is assumed as being degraded in performance if the primary receiver is degraded, and that the whole group is assumed as being upgraded if the primary one is upgraded.

In this algorithm, the primary receiver announces its status information to the whole group prior to other receivers, and every receiver listening to the primary and other receivers' status compares its own status with them. Accordingly, any receiver may give up the status notification in case its status is not worse than others, resulting in the reduction of unnecessary responses to the sender.

[†] 정 회 원 : 한국전자통신연구원

^{††} 정 회 원 : 충북대학교 컴퓨터과학과

논문접수: 1997년 10월 18일, 심사완료: 1998년 1월 26일

1. 서 론

그룹웨어 응용은 임의의 송신자가 보내는 데이터를 그룹내의 수신자들이 적절히 수신할 수 있을 정도로 보내는 것이 중요하다. 예를 들어 하나의 응용에서 비디오 데이터를 그룹 구성원에게 송신하게 될 때, 어떤 수신자는 데이터를 잘 수신할 수 있는 반면 어떤 수신자는 성능상의 이유 등으로 전송된 데이터를 정확히 수신하지 못하는 경우가 있게 된다. 비단 이것은 수신 단말의 성능 문제 뿐만 아니라 송수신자간의 통신망 대역폭이나 계속해서 변동되는 트래픽 상황에도 원인이 있게 된다.

송신측에서는 이러한 이질적이고 동적인 상황에서 보다 효율적으로 데이터를 수신자에게 전달하기 위해서는 수신자들의 수신 상태를 정기적으로 감시하고 확인된 수신자들의 상태 즉, Quality of Service(QoS)에 따라 카메라로부터 입력되는 프레임율을 조절한다든가 또는 전송률 제어나 오류 제어 등 다양한 조치가 요구된다.

본 연구의 목적은 송신자가 멀티미디어 데이터를 전송할 때, 그룹내의 수신자들이 전송된 데이터를 적절히 잘 받고 있는가를 수시로 파악하여 데이터의 입력율이나 전송률 등을 조절함으로써 이질적인 시스템이나 통신망 환경에서 최적화된 서비스를 달성하도록 하는 데 있다. 특히 본 연구의 초점은 송신자의 송신 데이터를 수신자가 어떻게 수신하고 있는가에 대한 상태를 파악하는 데 있으며, 여러 수신자들로부터 응답되는 상태 통보 데이터를 절감하는 기법에 관심을 두고 있다. 이러한 응답 데이터의 최소화에 대한 필요성은 멀티캐스트 통신에서 그룹의 크기가 커지면 커질수록 더욱 심각한데, 최근의 많은 연구는 다양한 상태 데이터를 포함하는 수신자의 확인 정보가 송신측에 너무 많이 입력되는 것을 해결하고자 하고 있다[2][8][14].

그룹통신 환경에서 송신자가 그룹 내 수신자들의 상태를 통보하는 대표적 방법으로는 두 가지를 들 수 있는데, 송신자가 임의의 시간 경과 후나 또는 일련의 데이터 블록을 전송한 후에 수신자에게 상태 통보를 요구하는 제어 패킷을 송출하여 정보를 수집하는 방법 [14]과 수신자가 주기적으로 자발적인 상태 통보를 해오는 방법 [2]이 있다. 한편 수신자의 상태 통보 방법

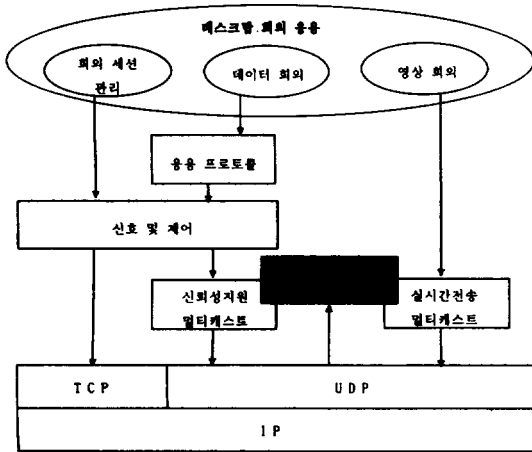
도 자신의 통보 데이터가 오직 송신자에게만 도달하게 할 수도 있으며, 또한 송신자를 포함한 그룹내의 모든 수신자에게 동시에 상태 통보를 주는 방법이 있다. 사실 송신자의 상태 통보 요구는 두 방법 중 어느 방법도 가능하나, 본 연구에서는 단지 알고리즘의 단순성을 고려하여 송신자가 일련의 패킷 블록을 전송한 후에 명시적인 제어 패킷을 보내는 방법을 채택한다. 한편 수신자의 상태 통보 방식은 송신자와 다른 수신자 모두가 청취할 수 있도록 하는 방법을 전제로 하는 데, 이것은 Real Time Control Protocol(RTCP)에서 QoS 정보를 수집하는 방법과 동일하므로 본 제안 방식을 이용할 경우 RTCP의 정보 수집 방식을 개선시킬 수 있다고 하겠다.

2. 관련 연구

2.1 그룹통신 시스템 구조

(그림 1)은 그룹웨어 응용을 위한 그룹통신 시스템의 구조를 간략히 보여 준다. 그룹웨어 응용은 다양한 공동작업의 도구를 제공하지만, 그룹통신을 위한 하부구조는 여러 가지 응용에서 공통적으로 활용이 가능하므로 여기서는 영상회의를 중심으로 한 간단한 모델을 중심으로 전체 구조와 중요한 구성 요소들을 살펴 보고자 한다. 이 시스템 구조는 이질적인 통신망을 전제로 하므로 인터넷 프로토콜인 TCP/IP 계층 위의 미들웨어와 데스크탑 기반의 영상회의 응용을 고려한다. 특히 그룹을 구성하는 다수의 멤버간 효율적인 통신을 위해 UDP/IP 프로토콜을 이용한다.

최근의 TCP/IP 프로토콜이 멀티캐스트를 지원한다고는 하여도 신뢰성이 보장이 안되기 때문에 신뢰성이 보장된 멀티캐스트가 많이 연구되고 있다. 신뢰성이 보장된 멀티캐스트 프로토콜은 텍스트와 같은 데이터 정보나 제어 정보를 그룹을 구성하는 멤버들에게 전달할 때 매우 유용하다. 예를 들어 회의에 참가 중인 멤버가 회의에서 탈퇴하고자 할 때 제어 정보를 모든 구성원에게 멀티캐스트하게 되면 모든 구성원은 쉽게 각자의 참가자 정보를 변경할 수 있다. 그럼에도 불구하고 오디오 및 비디오 데이터는 부분적인 데이터 손실이 발생하여도 정보의 전달에 큰 영향이 없으므로 UDP는 멀티미디어 정보 전송의 수단으로 많이 활용된다. 한편 회의 소집 요청을 명시적



(그림 1) 그룹통신 시스템 구조
(Fig. 1) Group communication system architecture

으로 사용하는 경우 유니캐스트 전송프로토콜인 TCP는 매우 중요한 역할을 한다. 즉, 회의 소집을 위해 시스템 각각을 호출할 때는 멀티캐스트 주소가 아닌 각 시스템의 IP 주소를 활용하여 호출하고 호출된 시스템들에게 멀티캐스트 주소를 알려 주어 그룹 관리 채널로 활용할 수가 있는 것이다. 피드백 제어를 기반으로 한 멀티캐스트 모듈은 결국 신뢰성이 보장된 멀티캐스트 프로토콜로 그룹 활동의 제어를 위해 매우 효율적으로 이용될 수가 있는 것이다. 신호 및 제어 모듈은 데스크탑 회의 응용의 세션관리를 지원한다. 세션관리는 회의에 호출하는 방식이나 하부 통신망의 종류에 따라 유니캐스트 및 멀티캐스트 연결을 적절히 구성하여 통합된 세션으로 관리하게 된다.

그림의 오른쪽 부분은 멀티미디어 전송 부분을 설명한다. 우선 Realtime Transport Protocol(RTP) 프로토콜은 오디오와 비디오 등의 실시간 데이터 전송을 지원하며 다자간 회의 등을 지원하기 위해 UDP 및 IP와 같은 프로토콜들을 활용한다. 특히 멀티미디어의 전송에 목적이 있으므로 데이터 손실을 복구하기 위한 오류제어나 흐름제어를 제공하기 보다는 데이터의 실시간성을 보장하는 데 치중하며, 연결 설정 등의 기능도 제공하지 않는다. RTP의 중요한 기능으로는, 우선 오디오 및 비디오 코덱으로부터 전달된 스트림 데이터를 적당한 크기의 RTP 패킷으로 구성하고 순서화를 위한 패킷 번호의 부여, 미디어 자체나

미디어 간의 동기화를 보장하기 위한 time stamping을 제공하는 것 등을 살펴 볼 수 있다.[3] RTCP는 RTP와 매우 밀접한 관련을 갖는다. RTCP는 세션에 참가 중인 모든 참가자들에게 데이터 패킷과 동일한 전달 메커니즘을 사용하여, 주기적으로 제어 패킷을 전송하여 수신측 또는 송신측의 QoS 정보를 전달하거나 또는 제어를 위한 정보를 검출할 수 있는 수단을 제공한다. RTCP의 QoS에 대한 피드백 방식은 QoS 관리자의 의사결정을 유도하여 비디오 등의 입력량을 조절하게 하는 등 일반적인 전송계층 프로토콜의 흐름 및 혼잡제어와 같은 기능을 가능하게 하는 것이다[1][4]. 이상에서 살펴본 시스템 구조는 ITU-T에서 권고하는 비디오 회의 표준인 H.323과 기본 골격이 일치함을 알 수 있는데, 여기서 신호 및 제어 부분은 H.225와 H.245와 매핑되며, RTP/RTCP는 H.225.0과 일치한다[5][6][7].

본 논문의 연구는 소개된 시스템 구조에서 수신자들의 QoS 정보를 수집하는 RTCP와 이러한 수집된 정보를 분석하고 제어 행위를 결정하는 QoS 관리 모듈에 초점을 맞추고 있다.

2.2 피드백 정보의 활용

그룹을 구성하는 수신자들의 QoS에 대한 피드백 정보는 송신자에게 다양한 제어를 가능하게 한다[8][9][10]. 대표적인 예로는 카메라와 같은 입력 장치로부터 인입되는 정보량을 입력처리 과정 중의 다양한 파라미터를 조정하여 조절할 수 있는데, 이러한 파라미터들은 주로 비디오의 압축과정에서 요구되는 파라미터들로 refresh rate이나 양자화 임계값(quantization threshold), 동작 감지 임계값(motion detection threshold) 등이 있다. Refresh rate은 카메라로부터 입력되는 프레임의 속도를 나타내는 데, 이 refresh rate의 감소를 통해 전체적인 비디오 프레임의 수를 줄일 수가 있는 것이다. 결국 압축을 처리하는 코더(coder)의 평균 출력율을 줄이게 됨으로써, 수신자에게 전달되는 비디오의 프레임의 양을 줄이는 효과는 갖지만, 반면 비디오의 품질은 떨어진다. 양자화 임계값은 비디오 화면을 코딩하는 데 사용되는 밀집도(granularity)를 결정해 주는 데, 이 임계값을 증가시킨다는 것은 픽셀 수를 감소한다는 것과 동일한 효과를 갖게 되고, 코더의 출력율은 감소하게 된다. 최근의 비디오 압축기

술은 화면과 화면 사이에 변동된 정보만을 보냄으로써 압축효과를 높이는 데, 동작 감지 임계값은 이전의 비디오 프레임의 블록들과 충분히 다른 임의의 프레임의 블록들을 결정하는 기준으로 이용된다. 이 임계값을 증가시키는 것은 비디오 움직임에 대한 코더의 감지능력을 완화시켜 이미지의 질을 저하시킨다. 이러한 입력 정보량의 조절은 결국 통신망을 통해 수신자들에게 전달되는 정보 전송량의 조절로 연결된다.

한편 QoS의 감시는 송신자간에 발생할 수 있는 오류를 보정하는 데도 이용될 수 있다. 실시간 멀티미디어의 경우는 데이터의 손실 뿐만 아니라 지연이나 지터와 같은 특성이 매우 중요하므로, 데이터의 손실이 발생한 경우 재전송에 의해 복구하고자 하는 경우 지연을 유발시켜 멀티미디어의 실시간 특성을 훼손하게 된다. 멀티미디어 서비스를 위해 손실 패킷의 복구 방법 중 하나로 연구되는 것은 Forward Error Control(FEC) 기법인데, 이 기법은 미리 전송되는 패킷을 중복하여 보내도록 하고 만약 수신측에서 데이터 손실을 발견하게 되면 중복으로 보내진 동일한 데이터 패킷을 이용하는 방안이다.

QoS 감시 정보를 이용하는 또 다른 예는 임의의 멀티미디어 응용이 다양한 서비스 옵션을 가지고 있을 때 가능하다. 예를 들어 비디오를 활용하는 응용에서 JPEG이나 MPEG과 같이 복수의 압축 알고리즘을 제공할 때 상황에 따라 알고리즘의 선택이 가능해지며, 변화되는 성능에 따라 화면 색상을 칼라나 흑백 중 선택적인 서비스도 가능해진다.

2.3 피드백 정보수집 방식

그러면 이러한 용도로 활용될 피드백 정보를 어떤 방법으로 효율적으로 수집할 것인가. 물론 이 문제의 해결도 여러 가지 관점에서 접근해 볼 수는 있으나, 본 논문에서는 수신자의 정보가 너무 폭주하여 송신자의 부하가 높아지고 성능이 저하되는 것을 방지하는 측면에서 접근해 보고자 한다. 최근의 많은 연구들은 다양한 QoS의 수집에 의한 제어나 혹은 멀티캐스트 전송 프로토콜의 흐름 및 오류제어 관점에서 피드백 정보를 다루고 있다. QoS를 수집하는 것과 멀티캐스트 프로토콜에서 수신자의 패킷 수신을 확인하기 위해 응답을 받는 것은 차이가 있겠으나 여러 수신자에서 하나의 송신자에게로 향하는 정보의 양

을 최소화하는 기법은 충분한 유사성을 내포하고 있다. 본 절에서는 이러한 내용들을 관련 연구들을 중심으로 살펴 보고자 한다.

앞에서 설명한 RTCP 프로토콜은 각각의 수신자가 송신자의 QoS 요청에 대해 모든 수신자가 멀티캐스트 방식으로 응답을 하므로 매우 많은 정보의 양이 발생하지만 모든 수신자는 다른 수신자들의 응답 정보를 받으므로 피드백에 의한 전체적인 대역폭의 사용 여부를 판단할 수 있다. 또한 통신망 제공자 역시 이러한 정보를 쉽게 청취할 수 있어 비디오 회의가 차지하는 통신망의 사용을 파악할 수도 있다는 장점이 있다. 물론 RTCP에서는 전체 수신자들의 응답을 허락하지만 응답을 위한 전체 대역폭에 제한을 두어 응답의 폭주를 절감하는 효과를 보고자 하나, 이러한 절감은 단순한 통신망의 트래픽을 줄이는 목적이므로 이 절감으로 정보 수집의 간격이 길어지고, 결국 얻어야 하는 정보가 감소된다는 단점을 면할 수 없다.

멀티캐스트 프로토콜에서 수신자의 정보가 폭주되는 것을 해소하는 대표적인 방법은 수신자들이 일제히 응답 정보를 송신하지 않고 라운드 로빈 형태로 한 수신자씩 번갈아 가면서 대표 수신자가 되어 응답을 주는 것이다[2][12]. 이 방법은 QoS 수집을 위해서는 합당치 않은 데 이것은 모든 그룹의 구성원이 전부 응답하여야 비로서 모든 정보를 얻을 수 있기 때문이다. 한편 Xpress Transport Protocol(XTP)의 Slotting/Damping 알고리즘은 복수의 수신자로부터 송신자에게 피드백되는 QoS 정보를 감소시키는 데도 이용할 수 있다. 이 방식은 각각의 수신자별로 무작위 시간 동안의 지연 후에 응답을 하도록 하고 한 수신자가 응답한 내용이 다른 수신자의 내용 보다 더 열악한 상태에 있는 경우, 더 양호한 상태의 수신자는 응답을 포기함으로써 전체적인 응답을 줄이는 것이다[14]. 그러나 이 방식의 문제는 각각의 수신자가 기다려야 하는 무작위 시간간격을 설정하기가 용이하지 않고 이 시간이 작을 경우 기다리는 수신자들이 응답하는 수신자를 청취할 수 없다는 문제가 있다.

확률기반 방식에서는 각각의 수신자는 송신자에게 임의의 확률을 가지고 응답한다. 즉, 송신자로부터 일단 수신자들에게 정보 요구가 보내지면 송신자는 임의의 시간 동안 수신자로부터 응답을 기다린 후 응답이 없으면 재차 요구를 보내게 되는 데, 이 임의의 시

간 동안 송신자가 그룹으로부터 가장 열악한 상황을 통보 받을 수 있으리라고 예상하기는 힘들다. 이 방식은 수신자의 중요성에 따라 확률을 배분할 수 있다는 장점이 있기는 하나, 문제는 그룹의 크기가 알려지지 않을 경우 확률을 결정하는 방식이 용이하지 않다[11].

무작위 지연 응답 방식에서는 각각의 수신자는 응답을 송신자에게 보내기 전에 임의의 무작위 시간동안 지연한 후 전송을 한다. 이 방식에서 만약 무작위 시간의 범위가 작은 값이 선택될 경우 송신측에서는 커다란 무작위 지연의 효과를 거두지 못한다. 그러나 이 방식은 이 지연 값이 그룹의 크기에 따라 적당히 선택될 경우 송신자가 모든 수신자의 응답을 처리할 수 있다는 장점이 있다[13].

개선된 확률기반 방식에서는 수신자의 그룹크기를 파악하고, 확장성을 위한 확률적 이유에 근거한다. 송신자와 모든 수신자는 데이터 블록을 전송하는 각 라운드의 시작 시점에 작은 크기의 무작위 키 값들을 계산하고, 송신자가 수신자들로부터 응답을 요청하고자 할 때, 생성된 키와 significant digit의 수를 표시하는 번호를 송출한다. 만약 일정 시간 내에 수신자로부터 응답이 없으면 significant digit 수의 값은 하나가 줄어 들어 해당 키 값에 적용되는 수신자가 있을 확률은 더욱 증가하게 된다. 이 과정은 폭주를 표시하는 응답의 수가 요구된 만큼 도달하거나 혹은 더 이상 significant digit가 존재하지 않을 때까지 반복된다[8].

한편, 범위 점진 증가형 탐색 방식에서 ttl(time-to-live) 즉, 패킷이 전달되어야 하는 범위를 점차 증가시켜서 멀티캐스트 트리를 따라 항해하는 것으로, 이 방식의 문제는 가장 열악한 상황의 수신자들이 송신자의 근처에 있지 않을 수 있으며, 또한 ttl밴드내의 수신자의 분포가 일정하지 않다는 데 있다[11].

3. 효율적인 QoS 정보 수집 방식

3.1 방식의 개요

효율적인 피드백 정보 수집 방식의 기본 아이디어는 여러 구성원들로 구성된 그룹의 행위를 어떻게 보는가의 관점에서 출발한다. 한 그룹에 속해 있는 수신자들은 수신자가 위치한 통신망 상의 환경이나 단말의 성능, 또한 현재 단말에서 동작하는 소프트웨어

들의 부하 등 다양한 이유로 어떤 시점에 서로 다른 성능을 보이게 된다. 데이터를 동시에 전해 주는 송신자의 입장에서는 제일 양호한 수신자의 성능에 맞출 것인가 혹은 제일 열악한 성능의 수신자에게 맞출 것인가 혹은 평균 수준에 맞출 것인가를 결정하여야 한다. 최고의 성능에 맞추게 된다면 다른 모든 수신자는 제대로 서비스를 받을 수 없게 될 것이고 이것은 평균 수준을 택하는 경우도 마찬가지가 된다. 멀티미디어 서비스를 위한 전형적인 아이디어는 OoS를 사전에 협상하고 이 QoS의 임계값 범위에 있도록 강요하는 것이다. 그러나 현재와 가까운 미래에는 통신망이나 단말의 QoS 보장은 불가능하며 인터넷 프로토콜을 이용하는 경우 더욱 그렇다. 이러한 이유로 현실적인 아이디어로 받아들여 지는 것이 가장 열악한 수신자의 성능에 맞추어 송신자가 데이터를 전송하도록 하는 방법이다.

본 알고리즘의 핵심은 그룹의 행위를 가장 열악한 수신자의 상태로 파악하여 거기에 맞추어 데이터를 보내도록 하는 것이다. 즉, 송신자의 상태요구에 대한 응답의 책임은 가장 열악한 한 수신자 만이 갖게 되며 다른 수신자는 이 수신자의 응답 상태를 자신의 상태와 비교해 봄으로써 자신의 상태가 더 양호한 경우나 같은 경우 중복 응답을 피함으로써 전체적인 응답 패킷 수를 감소시키는 것이다. 단, 수준이하의 수신자로 인해 모든 가능한 수신자의 피해를 막기 위해 최저 임계값을 적용하도록 하되, 그 이상의 범위에서는 QoS 감시결과를 통한 다양한 제어가 가능하도록 하는 방법이다.

제안 방식의 절차를 소개하기에 앞서 우선 라운드와 대표 수신자라는 두 용어를 정의하여 이해를 돕고자 한다.

- 라운드(round): 송신자가 일련의 데이터 패킷을 보낸 후에는 수신자들로 수신상태를 요구하게 되는데, 이때 송신자가 수신상태를 요구하기 이전까지의 전송되는 일련의 데이터 블록의 단위를 라운드라고 한다.
- 대표 수신자: 그룹내의 수신자들 중 성능이 가장 열악하다고 판단된 수신자로 각각의 라운드별로 결정되는데, 송신자의 상태 요구 제어 패킷을 받으면 반드시 응답할 의무를 갖는 수신자를 말한다.

송신자 알고리즘

```

    primary_receiver_no = 1; /* Initially, primary receiver is the receiver that has receiver no.1 */
AGAIN:
    round_no = round_no + 1;
    for(n=0; n < round_size; n++) { /* round_size is the number of packets in a round */
        data_type = USER_DATA;
        send-multicast(data_type, round_no, user_data);
    }
    data_type = REQ_QoS;
    send_multicast(data_type, round_no, primary_receiver_no);
    for all the data in the incoming buffer {
        read_multicast(QoS_data, receiver_no);
        if(receiver_no == primary_receiver_no)
            primary_QoS = QoS_data;
        else
            others_QoS = QoS_data;
        if(primary_QoS is worse than others_QoS) {
            primary_receiver_no = receiver_no; /* change primary receiver */
            modify data input or transmission_rate;
        }
    }
    goto AGAIN;

```

(그림 2) 송신자 알고리즘
(Fig. 2) Sender-side algorithm

3.2 피드백 정보 수집 방식

제안하고자 하는 피드백 정보 수집 기법의 처리 절차를 (그림 2)와 (그림 3)을 통해 송, 수신자별로 각각 설명하면 다음과 같다.

- ① 초기화: 수신자 번호 1번의 멤버를 주 수신자로 선정한다.
- ② 하나의 라운드내의 데이터(USER_DATA)를 연속해서 송신한 것으로 계산한다.
- ③ 데이터 전송 후 대표 수신자에게 수신상태를 요구하는 제어 패킷(REQ-QoS)을 송출한다.
- ④ 수신자로부터 QoS 상태 통보를 위해 전송되는 데이터(QoS_data)을 버퍼에서 읽고 패킷 수를 계산

한다.(수신자로부터 입력되는 QoS 데이터 패킷은 계속적으로 동작하는 스레드에 의해 입력 즉시 버퍼에 저장된다.)

- 해당 라운드에서 가장 열악한 상태의 수신자 번호와 수신 상태정보를 저장한다.
- 가장 열악하다고 판단된 수신자가 현재의 대표 수신자 인가를 비교한다.
- 만약 같은 수신자인 경우 이제까지의 대표 수신자가 계속해서 대표 수신자로 남지만 다른 경우 대표 수신자를 변경한다.
- ⑤ 초기화 이후의 과정을 반복한다.
- ⑥ 송신자로부터 전송되는 데이터를 읽고 유형에 따라 처리하는 데, 입력 데이터의 유형에는 제어 패

수신자 알고리즘

AGAIN:

```

read_multicast(data_type, round_no, receiver_no, other_QoS, user_data);
switch(data_type) {
    case REQ_QoS:
        if(my_receiver_no == primary_receiver_no)
            send_multicast(round_no, my_QoS, my_receiver_no);
        break;
    case QoS_ACK:
        if(other_QoS is worse than my_QoS)
            send_multicast(round_no, my_QoS, my_receiver_no);
        break;
    case USER_DATA:
        compute_QoS(round_no, my_QoS);
        break;
}
goto AGAIN;

```

(그림 3) 수신자 알고리즘
(Fig. 3) Receiver-side algorithm

킷(REQ_QoS, QoS_ACK)과 사용자 데이터 패킷(USER_DATA)들이 있다.

⑦ 입력 데이터가 REQ_QoS 패킷인 경우

- 자신이 대표 수신자인 경우는 라운드별로 집계된 QoS 데이터를 해당 라운드의 버퍼에서 읽어 송신자에게 제어 패킷(QoS_DATA)으로 통보한다. 이 패킷은 송신자 외에 모든 다른 수신자에게도 통보된다.
- 만약 수신자 자신이 대표 수신자가 아닌 경우에는 별도의 처리를 하지 않고 이 패킷에 대해 응답하게 될 대표 수신자의 제어 정보를 기다린다.

⑧ 입력 데이터가 QoS_ACK 패킷인 경우

- 만약 이 패킷이 대표 수신자의 정보일 때는 나의 상태와 비교하여, 만약 내 자신이 주 수신자의 상태보다 더 열악하다고 판단되면 해당 수신자는 나의 상태 정보를 제어 패킷(my_QoS)에 실어 송신자에게 보낸다.

- 대표 수신자가 아닌 다른 수신자의 정보인 경우는 이 패킷에 포함된 상태 정보와 나의 상태 정보를 비교하여 나의 상태가 더 열악한 경우는 대표 수신자나 또 다른 수신자의 상태 정보 패킷을 계속해서 기다리지만 만약 나의 상태가 입력된 수신자의 상태 보다 양호한 경우는 이 라운드에 대한 나의 상태 통보를 포기하는 flag를 설정한다.

⑨ 입력 데이터가 USER_DATA인 경우

- 자신의 QoS 데이터를 집계한다.
- ⑩ 상기 과정을 반복한다.

최초 대표 수신자의 선정이 잘못 될 경우 많은 수신자로 부터의 응답이 예상될 수도 있으나, 두번째 라운드 부터는 적용이 가능하게 된다. 이것은 초기에 여러 수신자 중 열악한 성능의 수신자를 미리 확인할 수 없으므로 대표 수신자는 임의의 수신자가 무작위

로 선정된다. 이때, 만약 이 수신자가 좋은 성능을 가진 단말일 경우 다른 많은 수신자 단말들이 이 대표 수신자 보다 더 많은 패킷 손실 등을 초래하여 열악한 QoS 상태 값을 가질 수 있으므로 송신자에게 피드백되는 QoS 정보는 매우 많아진다. 그러나 첫번째 라운드에서 최악의 QoS를 갖는 대표 수신자가 결정되고 난 이후에는 이 대표 수신자 보다 더 열악한 수신자의 수는 처음의 경우에 비해 감소되는 것을 예상할 수가 있다.

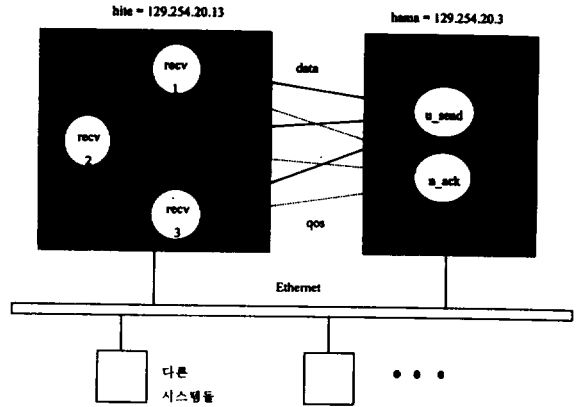
이 알고리즘에서 고려해야 할 또 다른 사항은 그룹의 크기가 커지면 특정한 수신자의 QoS 값은 일정한 수준의 멀티미디어 서비스를 제공하기에 무리가 따를 수 있고, 다른 수신자들에게도 영향을 줄 수 있다. 따라서 이러한 경우는 최소 QoS 임계값을 제시하여 수용 범위 밖의 수신자로부터의 통보를 억제시키는 방법을 활용하는 것이 필요하다.

4. 시험 및 평가

4.1 시험 방법

시험의 목적은 송신자의 상태정보 수집 요구에 대해 대표 수신자와의 상대적 비교에 의한 응답 패킷의 절감이 없을 경우 수신자들이 응답해야 하는 최대 응답 패킷수와 본 논문에서 제안하는 대표 수신자 개념 기반의 피드백 수집방식에 의한 응답 패킷수를 비교하여 응답 패킷수의 감소효과를 분석하는 것이다. 본 제안 방식을 적용하지 않은 경우의 응답 패킷수는 산술적인 계산에 의해 쉽게 얻을 수 있으므로 시험에서는 본 방식이 적용되었을 때의 패킷수만을 측정하여 비교를 통해 알고리즘의 효율을 분석하여 보았다. (그림 4)는 시험시스템을 구성하는 통신망과 단말, 시험 프로그램을 간단히 보여 준다. 많은 수와 다양한 종류의 단말을 이용하여 시험하는 것이 현실적인 분석 데이터를 얻는 데 효과적이겠지만, 시험환경 구축의 어려움으로 Ethernet에 연결된 두 대의 SUN 시스템(Solaris 2.5)인 hama와 hite 시스템을 이용하여 시험하였다. 이 중 hama 시스템에서는 데이터를 송신하는 모듈과 수신자로 부터 보내 오는 상태 정보를 수신하는 전용 모듈로 구성된다. 한편 hite 시스템에는 수신자의 수 만큼 데이터 수신용 프로세스를 생성하여 각자이 별도의 수신자 단말 역할을 담당하도록

하였다.



(그림 4) 시험시스템 구성
(Fig. 4) Test system configuration

시험은 사용자 패킷 200개가 전송되는 동안 수신자들로부터 전송되는 상태 응답 즉, ACK 패킷의 수를 관찰하였다. <표 1>은 시험 유형과 시험에 이용된 파라미터 값들을 보여 주고 있는데, 시험 유형에는 크게 수신자의 수를 변화에 따른 측정과 오류 발생율의 변화에 따른 측정, 한 라운드를 구성하는 패킷수의 변동에 따른 측정, 그리고 서로 다른 오류율(복합 오류율)을 가졌을 때의 측정 등을 단계적으로 실시하였다.

<표 1> 시험 유형별 파라미터
(Table 1) Parameters for test case

시험 유형	변화 값	고정 값
수신자 수 변화에 따른 ACK 수 측정	2, 3, 4, 5, 6	- 오류율: 10% - 라운드내 패킷 수: 5개
오류율 변화에 따른 ACK 수 측정	5%, 10%, 15%, 20%, 25%	- 수신자수: 5명 - 라운드내 패킷 수: 5개
한 라운드내 패킷 수 변화에 따른 ACK 수 측정	3, 5, 7, 9	- 수신자수: 5명 - 오류율: 10%
수신자별 오류율이 다른 경우의 ACK 수 측정	수신자 #1: 5% 수신자 #2: 10% 수신자 #3: 15% 수신자 #4: 20% 수신자 #5: 25%	- 수신자수: 5명 - 라운드내 패킷 수: 5개

모든 파라미터의 값들을 복합적으로 변화시키게 되면 매우 많은 종류의 시험이 필요하기 때문에, 본 시험에서는 관측의 대상이 파라미터를 중심으로 변화시키는 파라미터와 해당 시험 유형에서 고정 값을 갖는 파라미터의 두 가지로 구분하였으며, 시험 유형에 따른 파라미터의 값은 표에서와 같다.

한편 본 시험에서의 관측은 수신자의 ACK를 얼마나 줄일 수 있는가에 초점을 두고 있으므로 실제로 송수신자 간에 사용자 데이터의 전송은 수행하지 않고, 수신자별로 랜덤 번호에 기초한 오류 데이터를 인위적으로 생성하여 실제 데이터의 전송에서 발생한 것으로 이용하였다.

4.2 시험의 평가

시험의 결과는 시험의 유형별로 <표 2>, <표 3>, <표 4>, <표 5>에 나타나 있고, 이러한 측정 값들을 토대로 계산해 본 알고리즘의 효율이 <표 6>에, 그리고 <표 6>을 그래프로 표현한 것이 (그림 5)과 (그림 6), (그림 7)에 나타나 있다.

<표 2>, <표 3>, <표 4>, <표 5>에서는 본 알고리즘이 적용되지 않았을 때 송신자가 수신자로부터 받게 되는 최대 ACK 패킷 수를 계산하였다. <표 2>에서 수신자가 2명인 경우의 예를 보면 우선 총 시험 패킷 수는 200개이고 한 라운드가 5개의 데이터 패킷으로 구성되므로 모든 수신자가 응답해야 한다고 할 때 각 수

<표 3> 오류율 변화에 따른 ACK수

<Table 3> Acknowledgement count based on error rate

오류율	최대	1	2	3	4	5	6	7	8	9	10	평균
5%	5×40 =200	49 (8)	50 (8)	61 (13)	61 (17)	58 (10)	60 (14)	61 (12)	55 (13)	61 (13)	53 (10)	56.9 (11.8)
10%	5×40 =200	94 (25)	96 (25)	83 (20)	99 (28)	60 (10)	91 (27)	96 (26)	86 (23)	86 (24)	90 (27)	88.1 (23.5)
15%	5×40 =200	115 (28)	99 (25)	86 (19)	90 (25)	110 (30)	98 (26)	105 (27)	110 (27)	88 (20)	108 (26)	(100.9) (25.3)
20%	5×40 =200	90 (26)	114 (41)	99 (25)	118 (36)	100 (28)	100 (31)	81 (23)	100 (32)	97 (30)	111 (37)	(101.0) (30.9)
25%	5×40 =200	119 (37)	125 (40)	143 (39)	114 (41)	128 (37)	121 (37)	134 (42)	119 (34)	121 (36)	139 (45)	(126.3) (38.8)

<표 4> 라운드 내 패킷수 변화에 따른 ACK수

<Table 4> Acknowledgement count based on number of packets in a round

수신자수	최대	1	2	3	4	5	6	7	8	9	10	평균
3	67×5 =335	144 (43)	137 (43)	150 (51)	151 (55)	151 (48)	149 (46)	145 (48)	147 (48)	140 (41)	144 (50)	145.8 (47.3)
5	40×5 =200	97 (28)	96 (24)	93 (24)	101 (29)	106 (32)	99 (29)	93 (22)	85 (21)	79 (22)	98 (28)	94.7 (25.9)
7	29×5 =145	51 (10)	46 (12)	48 (8)	44 (13)	43 (7)	49 (10)	44 (8)	40 (7)	47 (10)	40 (5)	45.2 (9)
9	23×5 =115	36 (11)	38 (10)	35 (12)	35 (8)	38 (13)	38 (11)	39 (11)	30 (6)	36 (10)	42 (11)	33.2 (10.3)

<표 2> 수신자 변화에 따른 ACK수

<Table 2> Acknowledgement count based on receiver size

수신자수	최대	1	2	3	4	5	6	7	8	9	10	평균
2	2×40 =80	50 (10)	49 (9)	45 (5)	47 (7)	46 (6)	54 (14)	52 (12)	56 (16)	52 (12)	54 (14)	50.5 (10.5)
3	3×40 =120	69 (22)	59 (16)	56 (14)	67 (120)	66 (16)	59 (16)	59 (16)	79 (26)	60 (14)	67 (17)	64.1 (17.7)
4	4×40 =160	82 (24)	76 (21)	83 (23)	83 (26)	74 (20)	80 (23)	73 (19)	81 (24)	82 (25)	81 (20)	79.5 (22.5)
5	5×40 =200	77 (19)	98 (29)	87 (25)	87 (22)	84 (25)	84 (24)	89 (23)	99 (27)	68 (13)	84 (20)	85.7 (22.7)
6	6×40 =240	126 (32)	109 (26)	83 (22)	97 (27)	118 (30)	94 (27)	76 (17)	103 (26)	93 (20)	97 (24)	99.6 (25.1)

()는 대표 수신자의 변동횟수

<표 5> 수신자별로 오류율이 다른 경우의 ACK 수

<Table 5> Acknowledgement count for different error rate of each receiver

오류율	최대	1	2	3	4	5	6	7	8	9	10	평균
5%~25%	40×5 =200	61 (11)	82 (27)	100 (38)	75 (24)	89 (30)	86 (29)	69 (18)	55 (8)	63 (13)	51 (8)	73.1 (20.6)

신자는 200/5 즉, 40개의 ACK 패킷을 보내야 하기 때문에 수신자가 2명인 경우 총80개의 ACK가 예상된다. 각각의 변동 파라미터에 대해서는 10회씩 시험을 통하여 실제 발생한 ACK 패킷 수를 기록하였고 아울러 시험 중에 발생한 대표 수신자의 변경 횟수를

참고로 기록하였다. 이 자료를 토대로 평균 값을 산출하였으며, 시험 중 발생한 최대 값과 최소 값을 기록하여 알고리즘이 보여 준 가장 효과적인 경우와 가장 비효과적인 결과도 함께 관찰할 수 있도록 하였다.

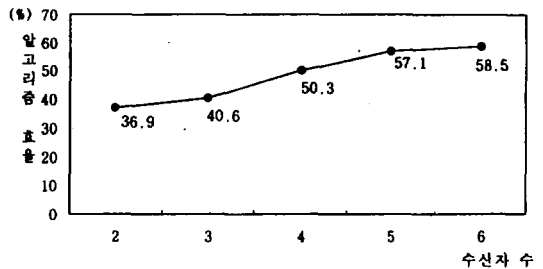
〈표 6〉은 이 시험을 요약한 것으로 대표 수신자의 개념이 없이 모든 수신자가 다 응답할 경우의 최대 ACK값에 대하여 실제 알고리즘이 적용되었을 때 측정된 ACK의 백분율을 ACK율로 표시하고 있다. 또한 알고리즘의 효율성은 이 ACK율을 1에서 뺀 값으로 해석하였다. 〈표 6〉의 (1)과 (그림 5)에서 본 수신자 수에 따른 알고리즘의 효율을 보면 수신자 수가 증가할수록 더욱 효과적임을 알 수 있다. 〈표 6〉의 (2)와 (그림 6)에서는 오류율이 적을 때 알고리즘의 효과가 더 큼을 알 수 있고, 〈표 6〉의 (3)과 (그림 7)에서는 한 라운드내의 패킷 수가 클수록 더 효과적임을 보여 준다. 〈표 6〉의 (4)와 (그림 5)의 점선 부분은 매우 의미 있는 관측을 보여 주는 데, 이것은 수신자가 다양한 오류율을 가진 경우 가장 적은 오류율의 수신자가 보여 주는 결과와 유사한 결과를 준다는 점이다. 즉, 이 경우는 각 수신자의 단말의 성능이 다른 경우를 대변하고 있는 것이다.

〈표 6〉 알고리즘 효율 분석
 (Table 6) Algorithm efficiency analysis

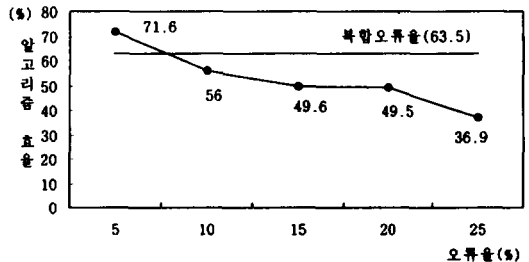
시험유형	파라미터 값	ACK 율(%)	알고리즘 효율(%)
(1) 수신자 변화	2	63.1	36.9
	3	53.4	46.6
	3	49.7	50.3
	5	42.9	57.1
	6	41.5	58.5
(2) 오류율 변화	5%	28.4	71.6
	10%	44.0	56.0
	15%	50.4	49.6
	20%	50.5	49.5
	25%	63.1	36.9
(3) 라운드내 패킷수 변화	3	43.5	56.5
	5	47.3	52.7
	7	31.1	68.9
	9	28.8	71.2
(4) 복합 오류율		36.5	63.5

관측된 결과가 방식의 효율성을 주고 있다고는 하지만 이것은 상대적인 효율성으로, 모든 수신자가 다 응답하는 경우와 비교할 때 매우 효과적임을 보여 준다. 그러나 그룹의 구성원이 대규모 집단이 될 때 송신자가 처리해야 하는 ACK의 수는 역시 많게 된다. 이러한 경우 관련 연구에서 보여 준 개선된 확률기반 방식을 이용하게 되면 우선 전체 그룹 중 임의의 집단을 선정하고 그 집단에 대해 본 알고리즘을 적용하여 배가의 효과를 기대할 수 있다.

또한 관련 연구의 분석에서 살펴 본 다른 기법들과 비교해 볼 때, 다른 기법들은 전체 그룹 중에서 소규모의 그룹을 무작위로 샘플링하는 데 치중하고 있으나, 제안 방식에서는 수신자들의 상태 값을 기초로 하여 응답 수신자들을 선택하여 단말 및 통신망의 현실적인 변화 상황이 반영된다는 점이 궁극적인 차이가 된다고 보겠다.



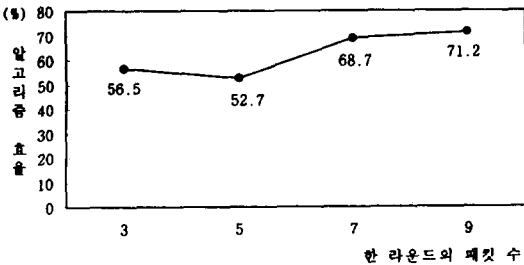
(그림 5) 수신자 변화에 따른 알고리즘 효율 분석
 (Fig. 5) Algorithm efficiency analysis based on receiver size



(그림 6) 오류율 변화에 따른 알고리즘 효율 분석
 (Fig. 6) Algorithm efficiency analysis based on error rate

그러나 본 방식은 대표 수신자가 먼저 응답을 함으로써 실제 다른 수신자들은 RTT(Round Trip Time)/

2의 지연 후에 응답이 가능하다는 단점이 있다. 이러한 지연 측면에서 타 연구와의 비교는 무작위로 샘플링을 하지 않고 각 수신자별로 각각의 지연 시간을 가진 후에 다른 수신자와의 비교 후에 응답을 줄이는 XTP의 알고리즘과 비교가 타당하다고 하겠다. 앞서 지적한 바와 같이 XTP의 경우 만약 각 수신자들에게 할당되는 지연시간이 작을 경우 알고리즘은 효율이 감소되며, 모든 수신자가 다른 수신자들의 상태를 모두 청취하는 목적과 송신자에게 피드백되는 정보를 최소화하는 목적을 동시에 만족시키고자 할 경우 최대 $RTT * \text{수신자의 수} / 2$ 의 지연이 발생한다. 이러한 분석을 통해 제안 방식은 다른 수신자들의 상태 청취와 송신자의 폭주의 감소·정보전달 지연의 감소 등의 복합적인 목적을 균형있게 처리하는 알고리즘이라고 볼 수 있다.



(그림 7) 라운드내 패킷수 변화에 따른 알고리즘 효율 분석
(Fig. 7) Algorithm efficiency analysis based on number of packets in a round

또한 대표 수신자의 상태 정보가 올 때 까지 다른 수신자들은 기다려야 하기 때문에 지연이 불가피하다고는 하나, 여기서는 감시된 QoS 값을 통해 취해지는 제어 행위가 입력율이나 전송률을 조정하는 일 등이 되므로 현실적으로는 감내될 수 있다고 보겠다.

한편 본 방식은 감시 대상이 되는 파라미터가 패킷 손실 뿐 아니라 지연이나, 지터 등 복수의 요소가 될 때, 대표 수신자를 결정하는 알고리즘이 추가되어야 한다. 우리는 이 문제에 대해 대표 수신자 대신 가장 열악한 파라미터 값을 기준으로 동일한 알고리즘을 적용할 수는 있지만 이 경우 파라미터 별로 복수의 대표 수신자가 있는 것과 같은 결과가 되고, 알고리즘의 처리는 매우 복잡해 지게 된다.

5. 결 론

이제 까지 멀티미디어 데이터의 멀티캐스트를 필요로 하는 그룹웨어의 환경과 효율적인 멀티미디어 데이터의 전송을 위한 피드백 제어의 필요성과 방식, 그리고 그룹내의 수신자들로부터 데이터 수신 상태에 대한 정보를 효율적으로 수집하는 방식과 이 방식의 효율성을 입증하고자 하는 실험 결과에 대한 분석을 살펴 보았다.

대표 수신자의 상태 통보가 모든 다른 수신자에게 통보되고 이 정보를 토대로 각 수신자가 자신의 상태 정보의 통보가 불필요한 경우 제어 패킷의 전송을 생략함으로써 송신자의 부하를 줄여 보고자 하는 본 논문의 알고리즘은 실험을 통해 효과적임을 알 수 있었다. 한편 본 알고리즘이 갖고 있는 문제에 대한 분석과 보완 방안도 함께 살펴 보았다.

장차 이 방식은 H.323을 중심으로 구현 중에 있는 비디오 회의 시스템에 적용 될 것이며, RTCP에서 모든 수신자가 응답하는 방식에 대한 보완으로도 검토 및 활용이 될 예정이다.

참 고 문 헌

- [1] 궁상환, 윤석환, 황승구, "멀티미디어 서비스를 위한 QoS 지원 자원관리 모델 연구," 한국정보처리학회 논문지, Vol. 24, No. 4, 1997.
- [2] 장홍성, 남형민, 이창건, 최양희, 정선종, 김종상, "라운드로빈 ACK와 선택적 재전송을 이용한 신뢰성있는 멀티캐스트 프로토콜," 한국정보과학회 학술발표논문집, Vol. 20, No. 2, 1993.
- [3] H. Schulzrinne, "RTP Profile for Audio and Video Conference with Minimal Control," RFC 1890, May, 1996.
- [4] H. Schulzrinne, S. Casner, R. Frederic, and V. Jacobson, "RTP: A Transport Protocol for Real-time Applications," RFC 1889, Feb., 1996.
- [5] ITU-T Draft Recommendation H.225.0, "Media Stream Packetization and Synchronization on Non-Guaranteed Quality of Service LANs," May, 1996
- [6] ITU-T Draft Recommendation H.245, "Control

Protocol for Multimedia Communication," Nov., 1995.

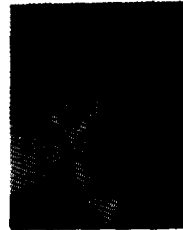
- [7] ITU-T Draft Recommendation H.323, "Visual Telephone Systems and Equipment for Local Area Networks which Provide a Non-Guaranteed Quality of Service," May, 1996.
- [8] J. Bolot, Thierry Turletti, I. Wakeman, "Scalable Feedback Control for Multicast Video Distribution in the Internet," ACM Sigcomm, pp. 58-67, Aug., 1994.
- [9] J. Bolot, H. Crepin, G. A. Vega, "Analysis of Audio Packet Loss on the Internet," Proceedings of NOSSADV95(Network and Operating System Support for Digital Video), pp. 163-174, Durham, NH, April, 1995.
- [10] J. Crowcroft, K. Paliwoda, "A Multicast Transport Protocol", Proc. of ACM Sigcomm, pp. 247-256, Aug. 1988.
- [11] R. Yavakar, L. Manoj, "Optimistic strategies for large-scale dissemination of multimedia information," Proc. ACM Multimedia, pp. 1-8, Aug., 1993.
- [12] Todd Montgomery, "Design, Implementation, and Verification of the Reliable Multicast Protocol," Thesis for the degree of Masters of Science submitted to the Graduate Program in Engineering, West Virginia University, Morgantown, West Virginia, 1994.
- [13] V. Jacobson, S. MacCanne, "vat", Manual pages, Lawrence Laboratory, University of California, Berkeley, CA.
- [14] W. Timothy Strayer, Bert J. Dempsey, and Alfred, "XTP: The Xpress Transport Protocol," Addison-Wesley Pub., 1992.



공 상 환

1977년 숭실대학교 전산학과 졸업
 1977년~1983년 육군 제2군수지원사령부 전산장교
 1983년 고려대학교 경영대학원 전자정보 석사
 1991년~현재 충북대학교 대학원 박사과정

1982년~현재 한국전자통신연구소 책임연구원(실장)
 관심분야: Multicast Transport, Distributed System Architecture 등



강 민 규

1986년 경기대학교 컴퓨터공학과 졸업(공학사)
 1988년 중앙대학교 대학원 컴퓨터공학과 졸업(공학석사)
 1994년~1995년 Stanford Research Institute(International Fellow)

1988년~현재 한국전자통신연구원(선임연구원)
 관심분야: 분산멀티미디어, 실시간 데이터 전송, 인터넷통신 서비스 품질



구 연 설

1964년 청주대학교 상학과
 1975년 성균관대학교 경영대학원 전자자료처리학과(경영학 석사)
 1981년 동국대학교 대학원 통계학과(이학석사)
 1988년 광운대학교 대학원 전자계산학과(이학박사)

1979년 이후 충북대학교 전자계산소장, 자연과학대학장 역임. 한국정보과학회 이사, 전산교육연구회 위원장, 충청지부장, 부회장 역임.

현재 충북대학교 컴퓨터공학과 교수
 관심분야: 소프트웨어공학, 정보통신, 알고리즘 등