

화상회의 시스템에서 동적 흐름 제어기법에 관한 연구

(A Study on the Dynamic Flow Control Algorithm on Video Conference System)

구 하 성*
(Ha-Sung Koo)

요 약 본 논문에서는 패킷 손실율과 왕복 시간을 사용하여 기존 방식보다 효과적인 네트워크 중단 간 동적 흐름 제어 알고리즘을 제안한다. 패킷손실율이 평균값이하이면 현재 네트워크 상태는 왕복시간을 사용하여 결정된 3가지 상태중 하나로 결정되며, 이것은 심각한 패킷 손실이 발생하기 전에 보다 정확한 대역폭 제어를 할 수 있게 한다. 만약 패킷손실율이 임계값이상이면 현재 네트워크상태는 혼잡상태가 된다. 적당한 전송률은 이들 네트워크의 각 상태에 따라 결정되며, 이때 프레임의 품질 뿐만 아니라 전송 프레임 숫자를 동시에 조절한다. 본 논문에서는 제안된 알고리즘에 대한 실험 결과를 보였다. 정량적 실험 결과를 통하여 제안 방식이 기존 방식보다 1.6배에서 6배의 성능 향상을 나타냄을 보였다. 동시에 총 전송 패킷수 역시 증가되었음을 보였다. 이로써 제안 방식이 기존 방식보다 향상된 대역폭 성능을 나타냄을 알 수 있었다.

핵심주제어 : 동적흐름제어, 화상회의

Abstract This paper proposes a dynamic end-to-end flow control algorithm that is more effective than previous methods considering either the Rate of Packet Loss (RPL) or Round Trip Time (RTT). When the RPL is under normal conditions, the current network status will be in one of three defined states by using the RTT and this makes bandwidth control precise before serious packet loss occurs. If the RPL exceeds a critical level, then the network is considered to be in a fourth state. Suitable transmission rates are determined depending on the network status and are controlled by adjusting not only the number of transmitted frames but also the quality of the frames. In this paper, we present some experimental results of the proposed algorithm. According to our quantitative analysis, the proposed method performs 1.6 to 6 times better than the previous method in terms of the RPL. At the same time, the total number of transmitted packets is increased, which indicates that the proposed method can provide greater bandwidth capacity than previous methods.

Key Words : Dynamic flow control, video conference

1. 서 론

최근 인터넷 화상회의 시스템은 실시간 멀티 미디어 핵심 응용 분야로서 다양한 연구가 활발

하게 진행되고 있으며 대표적으로 Mbone상에서 개발된 vic[1], vat[2], IVS[3], nv[4]와 같은 다 자간 화상 회의 시스템 등이 있다. 인터넷망은 네트워크 트래픽의 변화가 매우 크고 데이터가 일시에 집중되는 체증(congestion)현상으로 말미암아 화상회의 시스템에서 요구하는 만큼 빠르

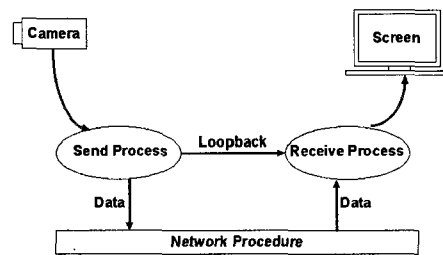
* 한서대학교 컴퓨터정보학과

고 신뢰성 있게 전송하지 못하고 수신자에게 전송되는 영상과 음성 데이터의 품질을 저하시키므로 고품질의 데이터를 제공하지 못하는 원인이 된다. 현재 이 문제점을 해결하고 또한 인터넷에서 QoS(Quality of Service)[5]를 제공하기 위해 많은 연구들이 수행되고 있다. 인터넷상에서 고품질의 데이터를 제공하기 위해 가장 많이 쓰이는 방법은 네트워크 트래픽을 위한 종단간(end-to-end)의 흐름제어이다. 이 흐름제어 기법은 연속적으로 변화하는 네트워크 상태에 맞추어 사용 가능한 대역폭을 예측하고, 예측된 대역폭만큼의 데이터를 전송하여 네트워크를 효율적으로 사용하는 흐름제어 기법이다. 이 방법은 네트워크상에서 새로운 알고리즘을 구현하지 않고, 네트워크 양단에 위치한 송수신자만을 변화시키므로 구현하기가 쉽다는 장점이 있다. 본문에서는 기존의 흐름제어 기법의 측정 요소로서 사용된 왕복시간이 수신자 측의 시스템 성능이나 과부하에 의해서 발생하는 패킷 손실에 대해 신속하게 대처하지 못하고, 패킷 손실은 또한 송신자와 수신자 사이에 패킷 손실이 발생하기 전까지 전송량을 증가함으로써 네트워크 트래픽을 증가시키는 문제점을 제기할 것이다. 그리고 제안된 흐름제어 기법에서는 이 두 가지 측정 요소의 문제점을 상호 보완하여 네트워크 상태 측정 부분에서 효과적으로 이용할 수 있음을 보일 것이다. 제안된 흐름제어 알고리즘의 구성은 다음과 같다. 먼저, 네트워크 상태 예측 부분으로 패킷의 왕복시간(Round Trip Time)을 이용하여 네트워크의 상태를 네 가지 상태로 나누고, 패킷 손실을 이용하여 수신자 측의 시스템 성능이나 과부하를 추정하여 네트워크 상태를 예측하는 알고리즘을 제안한다. 그리고 전송률 조절부분으로 영상 압축알고리즘의 특성을 이용하여 압축된 데이터의 품질별로 전송량을 조절할 수 있게 하고, TCP의 혼잡제어 메커니즘의 slow-start 단계와 congest-avoidance 단계를 이용하여 예측된 네트워크의 상태에 따라 효과적으로 전송량을 조절할 수 있음을 보인다.

2. 관련 연구

2.1 Vic 화상 회의 시스템

vic[1]은 Mbone에서 사용되는 화상회의 시스템으로 Lawrence Berkeley National Laboratory의 Network Research Group에서 연구 목적으로 개발된 응용프로그램이다. vic은 Intra-H.261, JPEG, cellb 등의 다양한 압축알고리즘들을 지원하며, CCD카메라에서 데이터를 읽어오는 여러 가지 프레임 그래버(frame grabber) 하드웨어 드라이버들도 구현되어 사용자 환경에 맞게 선택하여 사용할 수 있도록 제작되었다. 또한 vic는 RTP[7](Real time Transport Protocol)를 구현하여, 영상 데이터와 함께 송수신자의 정보를 전송할 수 있다. vic는 기본적으로 아래의 그림과 같이 두 개의 프로세스로 이루어졌으며, 이것을 세분하면 6개의 모듈(프레임 그래버 모듈, 압축 모듈, 송신 모듈, 수신 모듈, 복원 모듈, 상영 모듈)로 나누어지게 된다.



[그림 1] vic의 구조

2.2 실시간 전송 프로토콜(RTP)

RTP[6]는 영상 또는 음성과 같은 실시간 특성을 갖는 데이터를 전송하기에 적합하도록 IETF(Internet Engineering Task Force)의 AVT(Audio-Video Transport) 그룹에서 제안하여 RFC1889로 표준화된 프로토콜이다. RTP는 데이터를 전송하는 RTP와 데이터 전송을 제어하는 RTCP로 나뉜다.

가. RTP(Real time Transport Protocol)

RTP는 음성, 영상 또는 시뮬레이션 데이터와 같은 실시간 데이터를 전송하기 위한 프로토콜이다. RTP는 UDP 위에서 구성되는데 이것은 UDP가 제공하는 멀티플렉싱과 체크섬의 기능

을 이용하기 위해서다. RTP 자체로는 RSVP[8]와 같이 자원을 예약하지 않으므로 실시간 데이터 전송과 QoS에 대한 어떠한 보장도 해주지 않는다. 또한 손실없는 데이터 전송도 보장해 주지 않으며 수신자에게 데이터가 순차적으로 전달될 것도 보장해 주지 않는다. 단지 RTP는 타임스탬프(timestamp), 일련번호(sequence number)등 실시간과 순서에 관계되는 정보만을 전달해 주고 송수신 상태에 대해서는 RTCP 보고를 받을 뿐이다. RTP는 실시간 응용프로그램을 위해 두 가지 메커니즘을 제공한다. 첫째, 일련번호를 제공하여 데이터를 순차적으로 처리할 수 있게 한다. 둘째, 타임스탬프 필드에 데이터가 생성된 시각 정보를 제공함으로써 데이터가 처리되어야 하는 시각(playback time)을 알려줌으로써 다른 미디어에서 생성된 데이터와 같이 처리될 때는 다른 미디어와 동기화(synchronization)를 할 수 있게 한다.

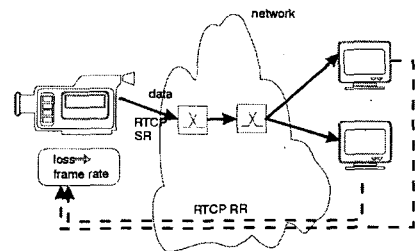
나. RTCP(Real time Transport Control Protocol)

RTCP는 제어 프로토콜로써 회의 또는 세션(Session)에 참가한 모든 참석자에게 피드백 정보를 주기적으로 전송한다. RTCP 프로토콜은 데이터를 전송하는 RTP와는 다르게 세션에 관계된 제어 패킷을 송수신하는 데 그 목적이 있다. RTCP 제어 메시지를 이용할 경우 세션에 관한 정보는 물론이고 에러 복구와 흐름제어까지도 할 수 있다. RTCP는 다양한 제어 정보를 전송하는데 SR(Sender Report), RR(Receiver Report), SDES(Source Description), BYE(goodBYE), APP(Application defined)의 5가지 메시지 형식을 가지고 있다. 기본적으로 SR과 RR을 이용하여 송신자와 수신자의 전송 상태를 파악할 수 있게 한다. SDES는 추가적인 정보를 전송할 때, BYE는 자신이 세션을 떠난다는 것을 다른 참여자에게 알릴 때, 그리고 APP는 응용 프로그램에서 새로운 종류의 메시지들을 이용한 실험과 테스트를 위해 사용한다.

3. 기존의 동적 흐름제어 기법

3.1 패킷 손실에 근거한 흐름제어 기법

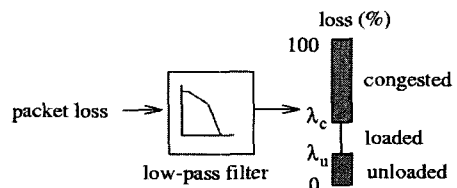
Ingo Busse[7]은 패킷 손실에 의한 동적 흐름 제어 기법을 제안하였다. 그는 수신자가 받은 송신자 보고를 이용하여 패킷 손실률과 지연 패킷의 지터(jitter)를 측정하였다. TCP가 혼잡상태로 변하면 비트 에러에 의해서 패킷의 손실이 발생하게 된다. 또한 버퍼 오버플로우가 발생하면 혼잡 연결 상태나 시스템의 네트워크 인터페이스에 의해서 패킷 손실이 일어난다. 아래의 그림은 Ingo Busse가 사용한 QoS 피드백 제어를 나타낸 그림이다.



[그림 2] 중단 대 중단 제어 메커니즘

가. 네트워크 상태 예측

수신자가 측정한 패킷 손실률은 두 개의 임계값을 적용하여 네트워크 상태를 세 가지 상태, 즉 혼잡, 정상, 무부하로 나누었다. 다음 그림에서 λ_c 는 혼잡 상태의 임계값을 나타내며, λ_u 는 무부하 상태의 임계값을 나타낸다. 전체 송신된 데이터의 개수와 수신자가 받은 전체 데이터의 개수를 비교하여 그 차이를 임계값으로 설정한다. Ingo Busse는 비디오 소스에 대해서 가장 적당한 임계값을 $\lambda_c = 4\%$ 그리고 $\lambda_u = 2\%$ 라고 제안하였다.



[그림 3] 네트워크 상태분류

나. 대역폭 조절

Ingo Busse는 네트워크 대역폭 조절에 있어서 두 가지의 알고리즘을 이용하여 전송되는 양을 조절한다. 혼잡 상태의 수신 데이터 개수와 총 수신 데이터 개수의 비율이 초기 혼잡값보다

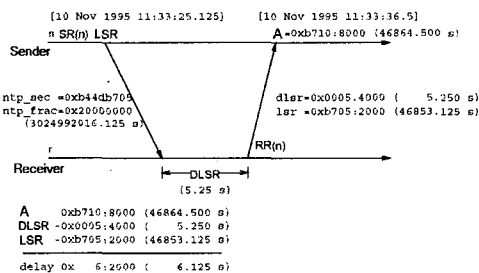
크거나 같으면 대역폭을 감소시키고, 정상 상태의 수신 데이터 개수와 총 수신 데이터개수의 비율이 초기 정상값보다 크거나 같으면 대역폭을 유지시키며 그 밖의 경우에는 대역폭을 증가시킨다. 이때 대역폭의 조절은 TCP의 알고리즘을 적용하여 전송상태에 따라서 실제적인 전송율의 수치를 결정하게 된다. 즉 전송상태의 값이 '증가'로 설정되면 대역폭을 선형적으로 증가시키고, 전송상태의 값이 '감소'로 설정되면 대역폭을 지수적으로 감소시킨다[7].

다. Ingo Busse 알고리즘의 문제점

논문[8]의 실험에서는 Ingo Busse가 제안한 알고리즘은 매우 불안정한 것으로 관찰되었다. 즉 네트워크 상태에 따라 전송률을 동적으로 변화시켜 네트워크 상태가 정상상태로 접근해야 하는데, 선형적인 증가율의 크기에 네트워크 상태가 혼잡과 무부하 상태를 반복하는 진동(oscillation)현상이 발생하는 것이다.

3.2 왕복 시간에 근거한 흐름제어 기법

왕복 시간(RTT)에 근거하여 네트워크 상태를 측정하고 측정된 네트워크 상태에 따라 전송률을 변화시키는 기법이다. 이 기법은 패킷 손실에 근거한 흐름제어 기법에 비해 네트워크에서 패킷 손실이 발생하기 이전에 전송률을 조정하여 네트워크 사용 효율을 증가시킬 수 있으며, 네트워크의 대역폭을 효과적으로 이용할 수 있는 장점이 있다. RTCP에서 RR 패킷의 도착시간을 A라고 하면 이들 세 변수의 관계는 아래의 그림과 같으며, 송신자는 다음 식을 이용하여 송신자와 수신자사이의 왕복시간을 계산할 수 있다.



$$RTT = A - lsr - dlsr$$

[그림 4] RTT 계산에 필요한 RTCP 변수들

4. 제안된 동적 흐름제어 기법

4.1 영상 압축알고리즘의 압축률 분석

화상회의에서는 여러 명의 사람이 동시에 대화하거나 영상을 보기위해서 여러 개의 버퍼를 두고 각각의 프레임을 저장하여 동적인 중복성을 제거하기 힘들다. 따라서, JPEG의 압축특성을 변화시킴으로써 네트워크 특정 상황에 적응시킬 수 있음을 보인다. RTP의 페이로드(Payload)에서 Video 카메라영상의 JPEG 압축을 지원한다.[9] 본 논문에서는 네트워크의 대역폭에 따라 데이터의 양을 동적으로 조절하기 위해 JPEG 압축 특성을 사용하였다. JPEG 표준에서는 Q백터를 이용하여 압축률을 조절하게 된다. 이 압축율의 범위는 1 - 100까지의 값으로 나타나게 된다. Q백터가 1이면 최대의 손실을 내면서 가장 많은 압축을 하며, 100이면 영상에 손실은 적게 주면서 압축은 가장 적다. 본 논문에서는 JPEG의 Q백터에 따라 실제적으로 압축되어지는 데이터의 크기를 측정하기 위해서 Intel사의 JPEG 알고리즘을 사용하여 실험을 하였다. 영상회의 실험에서는 USB카메라로 160 * 120 크기의 RGB24포맷의 데이터를 입력 받아 Q백터를 인자로 하여 영상데이터를 JPEG 부호화를 하였다. JPEG 알고리즘의 품질 레벨을 설정하기 위해서 Q백터의 값을 85, 75, 65, 55, 45까지 각각 지정하여 각 Q백터에 따라 생성되는 데이터의 양을 측정하였다.

<표 1> Q백터별 품질레벨 및 압축률

Q백터의 값	JPEG 품질레벨	압축데이터의 평균	압축율
85	Maximum	4884.118	0.08
75	high	3607.521	0.06
65	Medium	3125.879	0.05
55	-	2888.406	0.05
45	Low	2442.606	0.04

위 실험의 결과를 보면 약 0.04 ~ 0.08까지의 압축율의 변화를 보여주고 있는데, 네트워크 대역폭을 적용시키면, 초당 15프레임의 전송 속도가 요구될 때, 약 36.63Kbyte/sec에서 약 73.26Kbyte/sec

까지의 대역폭 변화에 적응할 수 있음을 의미한다. 예를 들면 73.26Kbyte/sec의 네트워크 대역폭이 접속자의 증가로 15%정도 감소되었다면 Q백터의 값을 45로 두어서 압축률을 0.04%로 유지하면 전송량은 약 63Kbyte/sec로 감소되어 네트워크 과부하를 방지할 수 있다. 본 실험을 통하여 품질 레벨 변화를 이용하여 변화하는 네트워크의 대역폭에 따라 영상의 화질을 향상시키거나 저하시킴으로써 일정한 프레임 수를 유지할 수 있는 근거를 마련하였다.

4.2 네트워크 상태 측정 알고리즘

본 절에서는 패킷 손실율과 왕복시간(RTT : Round Trip Time)을 네트워크 상태 파악요소로 고려하여 동적으로 네트워크 상태를 측정하는 방법을 제안한다. 아래의 그림은 송신자에게 RTCP의 RR 패킷이 도착할 때마다 적용하는 네트워크 상태 측정 알고리즘의 가상코드(pseudo code)이다. 제안된 알고리즘에서는 (1)라인에서와 같이 현재까지 보고된 패킷의 개수와 전체패킷의 개수를 적용하여 패킷 손실률 즉 loss_Rate를 계산한다.

```

// Packet loss-rate measurement
(1) loss_Rate = (1 - (Packet_cnt / Maximum_cnt)) * 100;

// smoothed average RTT measurement
(2) Avg_RTT = (1 - 0.75) * Avg_RTT + 0.75 * Cur_RTT;
(3) if((Old_RTT / Cur_RTT) > 1.0)
(4) {
(5)   Loaded_Ratio += 0.025;
(6)   Congested_Ratio = 0.5;
(7) }
(8) else
(9) {
(10)   Congested_Ratio -= 0.025;
(11)   Loaded_Ratio = 0.2;
(12) }
(13) if(Cur_RTT > Max_RTT)
(14) {
(15)   Max_RTT = Cur_RTT;
(16) }
(17) Congested_RTT = Congested_Ratio * Max_RTT;
(18) Loaded_RTT = Loaded_Ratio * Max_RTT;

(19) if(loss_Rate >= 0 && loss_Rate < 4)
(20) {
(21)   if(Avg_RTT >= Loaded_RTT) // Unloaded state
(22)     NetworkState(UNLOADED);
(23)   else if(Avg_RTT < Loaded_RTT && Avg_RTT >= Congested_RTT) // Loaded state
(24)     NetworkState(LOADED);
(25)   else if(Avg_RTT < Congested_RTT) // Light congested state
(26)     NetworkState(LIGHTCONGESTED);
(27) }
(28) else if(loss_Rate >= 4 && loss_Rate < 100)
(29) {
(30)   NetworkState(CONGESTED); // Congested state
(31) }

```

[그림 5] 제안된 네트워크 상태 측정 알고리즘

(2)라인은 현재까지 보고된 RTT인 Cur_RTT에다가 smoothing factor 0.75(3/4)를 적용하여 평

균 RTT 즉 Avg_RTT를 계산한다. Cur_RTT에 높은 가중치를 둔 이유는 RTCP 피드백 정보가 자주 일어나지 않기 때문에 네트워크 변화를 빨리 반영시키기 위함이다. 참고로 RTCP 피드백 정보는 약 5초 주기로 전송된다. (3)~(7)라인은 네트워크 상태변화에 빠르게 적응하기 위해서 측정된 RTT가 증가하는지 감소하는지를 연속적으로 판단한다. 현재의 측정된 RTT인 Cur_RTT가 이전 값 Old_RTT보다 큰지 작은지에 따라 Loaded_Ratio와 Congested_Ratio를 증가시키고 그렇지 않으면 각 상태의 초기 설정값으로 재설정(reset)한다. Loaded_Ratio와 Congested_Ratio는 네트워크 상태가 각각 혼잡상태와 무부하 상태로 되어간다는 것을 나타내며, 각 네트워크 상태에 따라 임계치의 값을 조절하게 된다. (8)~(11)라인은 패킷지연을 줄이기 위해 패킷의 RTT를 일정한 범위 내에 유지시키도록 하기 위한 코드 부분이다. 먼저 (8)~(9)라인은 경험한 RTT 중에서 최대 RTT인 Max_RTT를 계속적으로 갱신하는 부분이다. (10)~(11)라인은 두 개의 임계치인 Loaded_RTT와 Congested_RTT를 Max_RTT의 비율에 의해 설정한다. 이 임계치를 구하기 위해 Loaded_Ratio와 Congested_Ratio를 위에서 설정하였다. 이 두 임계치의 값은 화상회의 시스템의 성능에 중요한 영향을 준다. Loaded_Ratio의 값이 커질수록 Avg_RTT의 값이 Loaded_RTT에 도달할 때까지 전송률을 증가시키므로 네트워크의 대역폭을 많이 사용하게 된다. 반대로 Loaded_RTT와 Congested_RTT의 간격이 커질수록 두 임계치 내로 평균 RTT인 Avg_RTT를 유지하려고 하므로 전송률은 더욱 안정적이게 된다. (12)~(20)라인은 보고된 패킷 손실율과 평균 RTT인 Avg_RTT에 의해 네트워크 상태를 결정하게 된다. 패킷 손실이 4%에서 100%일 경우에는 혼잡상태인 것으로 파악하고 수신자 측의 시스템 성능이나 과부하에 의해서 발생하는 패킷 손실률을 파악하며, 패킷 손실이 0%에서 4%전까지일 경우에는 계산된 Avg_RTT에 의해 Loaded_RTT와 Congested_RTT 임계치로 생기는 세 범위 중 하나로 선택된다. 세 가지 상태는 정상, 무부하, 약간혼잡이다. 한편 네트워크 상태가 정상상태일 때의 대역폭 임계값들은 각각 0.2와 0.5이며 다음절에서 실험을

통하여 최적화된 비율이다.

4.3 전송률 조절 알고리즘

제안하는 전송률 조절 알고리즘은 다음 그림과 같다.

```

#define DELTA_VALUE 0.25 // frame per second
(1) if(Avg_RTT >= Last_time)
    {
    if(NetworkState == LOADED) // loaded state
    (2)     Threshold_fps = (1 - 0.25) * Threshold_fps + 0.25 * Cur_fps;
    (3)     CompressionQuality(QUALITY_HIGH);
    }
    else if(NetworkState == CONGESTED) // congested state
    (4)     Cur_fps = max(Cur_fps * 0.65, MIN_FPS);
    (5)     Threshold_fps = (1 - 0.25) * Threshold_fps;
    (6)     CompressionQuality(QUALITY_LOW);
    }
    else if(NetworkState == LIGHTCONGESTED) // Lightcongested state
    (7)     Cur_fps = max(Cur_fps * 0.75, MIN_FPS);
    (8)     CompressionQuality(QUALITY_MEDIUM);
    }
    else if(NetworkState == UNLOADED) // Unloaded State
    {
    if(Cur_fps < Threshold_fps) //slow-start phase
    (9)     Cur_fps += DELTA_VALUE;
    }
    else // Congestion-avoidance phase
    (10)    Cur_fps += (1 / Cur_fps) * DELTA_VALUE * 0.25;
    (11)    CompressionQuality(QUALITY_MAXIMUM);
    }
    }
    
```

[그림 6] 제안된 전송률 조절 알고리즘

(1)~(2)라인은 보다 안정적인 전송률 측정을 위해 마지막으로 전송률을 결정한 후에 적어도 한 RTT만큼 기다린 후에 결정하도록 하였다. 여기서 Last_time은 최종적인 전송률 결정시간 이후에 경과된 시간을 의미한다. 만약 데이터 패킷이 RTT가 정상 상태의 범위 내에 속한다면 프레임 수를 변화시키지 않고 현재의 프레임 수 Cur_fps를 smoothing factor 0.25를 써서 프레임 수를 threshold_fps로 평균화시킨다. (3)라인은 네트워크 상태가 정상이므로 압축품질을 QUALITY_HIGH로 설정한다. 네트워크 상태 예측 알고리즘에 의해서 결정된 상태가 (4)라인에서처럼 심한 혼잡 상태라고 판단되면 현재의 프레임 수 Cur_fps를 0.65의 비율로 기하급수적으로 감소시킨다. (5)라인에서는 화상회의 시스템이 Cur_fps를 갑자기 증가시키므로 인해 패킷손실이 발생할 수 있으므로 smoothing factor를 적용하여 threshold_fps를 낮춘다. threshold_fps를 낮추어 주는 것은 다음 순간에 패킷손실이 발생하는 것을 막기 위해 무부하 상태에서 Cur_fps를 조심스럽게 증가시키기 위해서이다. (6)라인에

서는 혼잡 상태에서 압축품질을 QUALITY_LOW로 설정하여 저품질의 영상을 제공하게 된다. 패킷손실이 발생하는 상황을 미리 예측할 수 있는 약간혼잡 상태에서는 (7)라인과 같이 Cur_fps를 조금씩 감소시킨다. (8)라인에서는 약간혼잡 상태이므로 압축품질을 QUALITY_MEDIUM으로 설정하여 전송되는 데이터의 량을 조절한다. 네트워크 상태가 무부하 상태이고 현재 프레임 수가 threshold_fps 이하일 때 (9)라인에서는 프레임 수를 DELTA_VALUE만큼 증가함으로 지수적으로 전송률을 높인다. 현재 프레임 수가 threshold_fps 이상일 때 (10)라인에서처럼 현재의 프레임 수의 역수에 의해서 증가시킴으로써 전송률을 선형적으로 높인다. (11)라인은 무부하 상태이므로 압축 품질을 최고의 품질인 QUALITY_MAXIMUM으로 설정하여 고품질의 영상을 수신자에게 전송한다. TCP의 윈도우조절 알고리즘은 매 RTT마다 사용되는 반면에 이 알고리즘에서는 새로운 RTCP의 RR 패킷이 들어올 때마다 이 코드 부분을 수행하므로 TCP처럼 하기 위해서는 시간간격을 일정하게 해줄 필요가 있다.

5. 실험

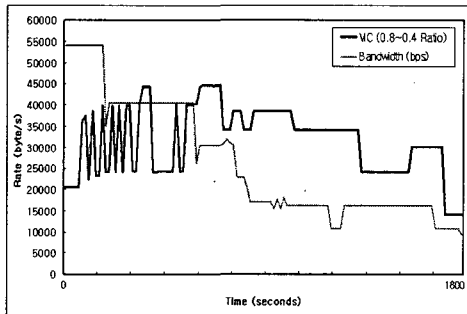
본 절에서는 제안된 동적 흐름제어 기법을 적용하여 화상회의 시스템의 전송량과 패킷 손실을 측정하는 방법을 통하여 실제 성능을 측정 및 비교하였다. 실험에서 사용된 컴퓨터들의 사양은 아래의 표와 같으며, 화상회의 시스템은 vic를 이용하였다. 그리고 화상회의 시스템이 경유하는 네트워크상에는 9개의 라우터가 존재하였다. 제안된 동적 흐름제어 기법의 성능 실험하기 위해서 실제 네트워크 상태를 정확하게 파악할 수 있는 ICMP의 echo를 이용한 PING이라는 프로그램을 작성하여 실험에 사용된 두 기관 사이의 실제 네트워크 대역폭을 측정하였다.

정확한 측정을 위해서는 화상회의 시스템의 RTCP 제어 패킷과 PING의 ICMP echo 패킷을 똑같은 5초 간격으로 설정하고 실험을 하였으며, 먼저 네트워크 상태 측정 알고리즘에서 초기 대역폭 결정 비율의 두 계수인 congested_ratio와 loaded_ratio 그리고 대역폭 비율의 증가와 감소를

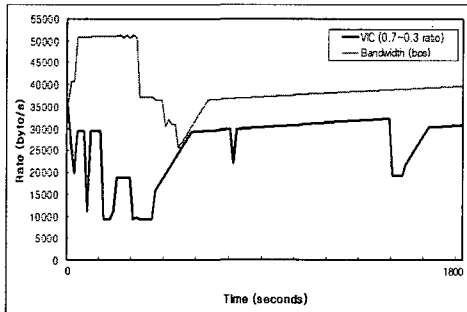
<표 2> 실험에 사용된 컴퓨터 사양

	컴퓨터1	컴퓨터2
CPU	850Mhz	866Mhz
Memory	512Mbyte	128Mbyte
Video Card	Geforce 256	Riva TNT 2
Camera	Logitech	Prochips
OS	Window XP	Window XP

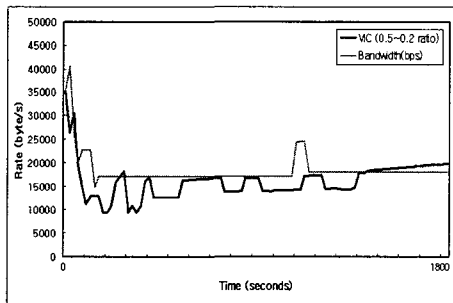
조절하는 계수인 network_ratio 의 결정을 위한 실험을 하였다. 아래의 그림은 congested_ratio 와 loaded_ratio의 계수를 0.5~0.2, 0.7~0.3, 0.8~0.4로 설정하여 실험한 그래프이다.



(a) 대역폭 비율 계수를 0.8~0.4로 설정



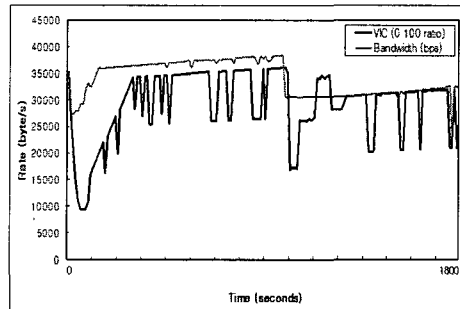
(b) 대역폭 비율 계수를 0.7~0.3으로 설정



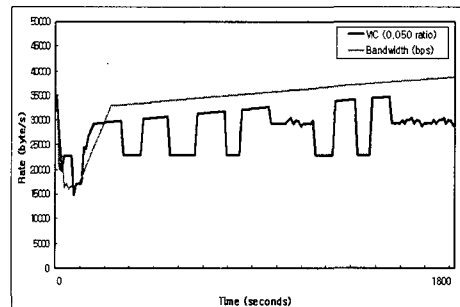
(c) 대역폭 비율계수를 0.5~0.2로 설정

[그림 7] 대역폭 결정 비율 계수에 따른 성능 실험 그래프

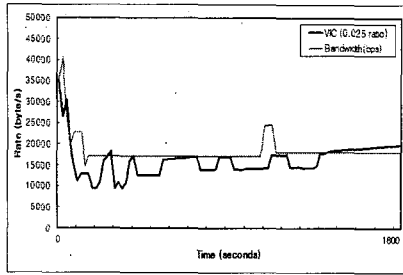
위의 실험 그래프에 의하면 congested_ratio와 loaded_ratio 계수의 간격이 크고 congested_ratio의 값이 1.0에 가까울수록 실험 초기의 실제 대역폭에 민감하게 반응하여 네트워크 상태가 혼잡과 무부하 상태로 반복하는 진동현상이 심하게 나타나게 된다. 진동현상은 congested_ratio 계수가 높은 값을 가지고 대역폭 비율의 간격이 크기 때문에 현재의 네트워크 상태를 제대로 파악하지 못한다. congested_ratio 계수가 클수록 정확한 네트워크 상태 측정에 많은 시간이 소요되며, 작을수록 네트워크 상태 측정에 적은 시간이 소요됨을 볼 수 있다. 위 실험을 통하여 congested_ratio와 loaded_ratio 계수의 최적의 값을 0.5와 0.2로 설정하였다. 아래의 그림은 기 설정된 대역폭 비율에 대하여 network_ratio 계수를 0.025, 0.050, 0.100으로 각각 설정하여 실험한 그래프이다. 아래의 실험 그래프에서는 network_ratio 계수가 클수록 대역폭의 비율이 큰 폭으로 변화되므로 네트워크 상태가 혼잡과 무부하 상태를 반복하는 진동현상이 관찰되었다. 진동현상의 빈도수는 network_ratio 계수가 클수록 자주 발생하며 진동의 폭도 크게 일어나며, 작을수록 진동현상의 빈도수와 진동의 크기가 작게 발생함을 관찰할 수 있다.



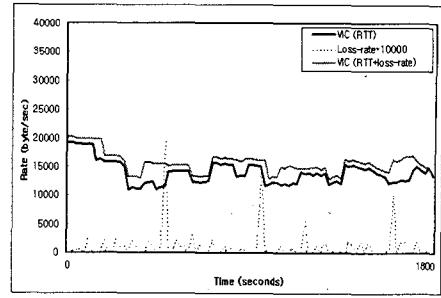
(a) 대역폭 증감조절 계수를 0.100으로 설정



(b) 대역폭 증감조절 계수를 0.050으로 설정

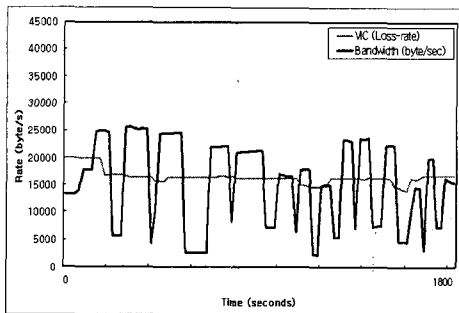


(c) 대역폭 증감조절 계수를 0.025로 설정
[그림 8] 대역폭 증감조절 계수에 따른 성능
실험 그래프



[그림 10] RTT 이용방법과 제안 방법의 성능
실험

위 실험에서는 network_ratio 계수의 값을 0.025로 설정했을 때 최적의 상태를 나타냄을 관찰하였다. 두 번째 실험에서는 네트워크 상태 측정의 측정 요소인 패킷 손실과 왕복시간의 특성을 알아보기 위해서 같은 네트워크 환경과 같은 시간대에 성능 실험을 하였다. 아래의 그림은 패킷 손실만을 측정 요소로 설정한 VIC의 전송량과 실제 대역폭의 전송량을 비교 실험한 그래프이다.



[그림 9] 패킷 손실만을 적용한 VIC의 성능
실험

위의 실험 그래프에서는 패킷 손실이 발생한 후에 전송량을 동적으로 변화시키기 때문에 네트워크 패킷 손실이 일어나기 전까지 전송량을 높이기 때문에 과도한 전송량에 의해서 패킷 손실이 일어나고 다시 지속적으로 전송량을 감소하기 때문에 실제 대역폭 전송량과는 상관없이 전송량이 큰 폭으로 상승과 하락만을 한다. 아래의 그림은 왕복시간만을 측정 요소로 적용한 VIC의 전송량(검은색)과 왕복시간과 패킷 손실을 적용한 VIC의 전송량(회색)을 비교한 그래프이다.

위 그림에서 제안된 알고리즘의 대역폭 특성이 기존 방식보다 향상되었음을 보이며, 여기서 왕복 시간만을 적용한 VIC는 상대방 시스템의 성능 저하로 인한 패킷 손실이 크게 발생한 경우 네트워크 상태 측정에 반영되지 않기 때문에 전송량을 신속하게 감소하지 않으므로 다음 단계에서 혼잡상태가 발생하여 그 지속시간이 왕복 시간과 패킷 손실을 적용한 VIC보다 오래 지속된다는 것을 확인하였다. 아래의 표는 왕복 시간과 패킷 손실을 적용한 VIC와 왕복 시간만을 적용한 VIC 그리고 패킷 손실을 적용한 VIC의 패킷 손실에 대한 실험의 결과를 나타낸 것이다.

<표 3> 3개의 알고리즘에 의한 패킷손실 비교

알고리즘 패킷	제안방식	RTT	Loss-rate
총개수	13915	12072	11986
손실개수	123	179	632
비율(%)	0.883	1.482	5.277
수행시간	11:30~12:00	11:30~12:00	11:30~12:00

위의 표에 의하면 제안된 알고리즘에서의 패킷 손실이 기존 방식에 비하여 약 1.6배에서 6배까지 향상되었고, 데이터의 전체 전송량에 있어서도 향상된 성능을 확인할 수 있다. 이는 왕복시간과 패킷 손실을 따로 적용하는 것보다 같이 적용하는 것이 성능측면에서 더욱 우수하다는 것을 반증한다. 또한 본 실험 결과를 통하여 패킷 손실과 왕복시간은 비례적인 관계가 있음을 관찰하였고, 왕복 시간을 기반으로 네트워크 상태를 측정하면서, 상대방 시스템의 성능저하

와 과부하로 인한 패킷 손실을 측정하고 전송량 조절에 적용함으로써 변화하는 네트워크 상태에 효과적으로 대처함을 보였다.

6. 결론 및 향후 연구방향

본 논문에서는 RTP/RTCP를 이용하여 왕복 시간과 패킷 손실을 측정 요소로 사용하며, 대역폭의 비율을 과거와 현재의 왕복시간의 차이에 의해서 조절함으로써 정확한 네트워크 상태를 예측하는 방법을 제안하였다. 전송률 조절 부분에서는 TCP의 혼잡제어 메커니즘인 slow-start 단계와 congest-avoidance 단계를 적용하여 전송량을 효율적으로 조절하였다. 왕복시간만을 적용한 흐름제어와 패킷 손실만을 적용한 흐름제어를 비교 실험하여 두 개의 측정 요소를 함께 사용하는 것이 패킷 손실율과 전송률 조절측면에서 뛰어난 성능을 보인다는 것을 실험을 통하여 보였으며, 실제 대역폭의 전송량에 근접하여 적용할 수 있음을 실험을 통하여 보였다.

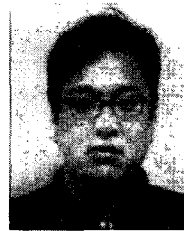
향후 연구과제로는 다음과 같이 세 가지로 나눌 수 있다. 첫째, 제안한 동적 흐름제어 기법을 다양한 TCP 백그라운드 트래픽을 적용하여 실험하는 것과 둘째, 다양한 압축알고리즘을 적용하여 네트워크 상태에 따라 압축품질을 분류하여 제안된 동적 흐름제어 기법에 적용할 것, 마지막으로, RTP 대역폭이 동적으로 할당될 때 RTCP 대역폭의 양과 결정하는 방법에 관하여 연구하는 것이다.

참 고 문 헌

[1] Steven McCanne, Van Jacobson, "vic: a flexible framework for packet video", ACM Multimedia, November 1995.
 [2] Jacobson, V. and McCanne, S., "Visual Audio Tool", Lawrence Berkeley Laboratory.
 [3] T. Turetti, "The INRIA Videoconferencing System (IVS)", onneXions - The Interoperability

Report, Vol.8, No. 10, pp. 20-24, October 1994.

[4] R.Fredrick, "nv", Manual Pages, Xerox Palo Alto Research Center.
 [5] Paul Ferguson and Geoff Huston, "Quality of Service - DeliveringQoS on the Internet and in Corporate Networks", Wiley Computer Publishing, pp.24, 1998.
 [6] Schulzrinne, Casner, Frederick, and Jacobson, V., "RTP: A transport protocol for real-time applications.", RFC1889, January 1996.
 [7] Ingo Busse, Bernd Deffner, Henning Schulzrinne, "Dynamic QoS Control of Multimedia Application based on RTP", May 1995.
 [8] 나승구, 고동환, 안종석, 김승범, "회의 시스템의 성능을 개선하기 위한 두 개의 동적 흐름 제어 방식의 비교", SoftExpo 97 (p381-390), 12월 1997.
 [9] L. Berc, W. Fenner, R. Frederick, S. McCanne, "RTP Payload Format for JPEG-compressed Video", RFC 2035, October 1996.



구 하 성 (Ha-Sung Koo)

정회원

- 1989년 3월 : 광운대학교 전자통신공학과 (공학사)
 - 1991년 9월 : 광운대학교 전자통신공학과 (공학석사)
 - 1995년 8월 : 광운대학교 전자통신공학과 (공학박사)
 - 1997년 3월부터 1999년3월:기아정보시스템 알고리즘개발팀장
 - 1997년 9월부터 현재 : 한서대학교 컴퓨터정보학과 교수
- <관심분야> 영상처리, 생체인식