화상회의 시스템에서 RSTT를 이용한 동적 흐름제어 기법에 관한 연구

구하성* · 심종익*

The Study of Dynamic Flow Control Method using RSST in Video Conference System

Hasung Koo* · Jongik Shim*

요 약

본 논문에서는 UDP에서의 동적 흐름 제어기법을 제안하며, 기존의 동적 흐름제어에서 많이 사용되는 측정 요소인 패킷 손실과 왕복시간의 특성에 대해 분석한 후 새로운 측정요소인 RSTT를 제안한다. 제안된 RSTT를 사용한 알고리즘은 전송률을 조절하는데 있어서 현재 측정된 네트워크 상태와 과거의 상태를 고려함으로써 보다 정확한 네트워크 상태측정을 할 수 있었다. 비교실험을 위하여 전송량과 네트워크의 상태를 출력하고 네트워크의 흐름패턴에 대한 상세 정보를 보여주는 네트워크 상태측정 프로그램을 구현하였다. 이를 이용한 성능실험을 통해 제안된 알고리듬이 기존 알고리듬에 비해 낮은 패킷 손실률로 네트워크 상태에 더욱 빠르게 적응함을 보여준다.

ABSTRACT

This study examines dynamic flow control method in UDP, analyzes packet loss which is frequently used element in measuring existing dynamic flow control and characteristics of round trip time, and proposes a new method of measurement, RSST. The algorithm that uses the proposed RSST enables accurate measurementof network status by considering both the currently measured network status and the past history of network status in controlling the transmission rate. For comparison study, a network status measurement software program that displays detailed information about volume of transmission, generation of network status, and flow pattern of network was developed. The performance test shows that the proposed algorithm can better adjust to network condition in terms of low pack loss rate over existing algorithms.

키워드

Dynamic Flow Control, Video Conference, RTT

Ⅰ. 서 론

화상회의 시스템은 인터넷 멀티미디어 서비스의 핵심적인 분야로 자리 잡고 있으며, 관련된 다양한 연구가 활발히 진행되고 있다. 대표적으로 연구 목적으로 개발된 vic[1], IVS[2], nv[3] 시스템 등이 있다. 인터넷 망은 네트워크 트래픽의 변화가 매우 크고 데이터가 일시에 집중되는 체증(congestion) 현상으로 인해 화상회의 시스템에서 요구하는 QoS(Quality of Service)[4]를 보장하지 못하므로 대역폭의 확보를 위해 많은 연구들

* 한서대학교 컴퓨터정보학과

접수일자 : 2005. 10. 27

이 수행되고 있으며, 종단간(end-to-end)의 흐름제어기법이 가장 많이 사용되고 있다. 이 방법은 네트워크상의 라우터와 상관없이 네트워크 양단에 위치한 송수신자의 단말 프로그램만을 조작하므로 구현하기가 쉽다는 장점이 있다. 기존의 흐름제어 기법에서 네트워크 상태 측정에 사용된 주된 요소는 종단 간 왕복시간(RTT)과 패킷 손실률 두 가지다. 왕복시간을 이용하면 네트워크의 상태를 예측하여 패킷손실이 일어나기 전에 미리 대처할 수 있으나, 수신자 측의 시스템 성능이나 과부하에 의해서 발생되는 패킷 손실에 대해서는 대처할 수가 없다. 두 번째 요소인 패킷 손실률을 이용하면 패킷을 손실한 후에 네트워크의 상태를 알 수 있기 때문에 네트워크 트래픽에 대한 대처가 느리고, 패킷 손실이 발생하기 전까지 전송량을 증가하기 때문에 네트워크에 트래픽을 증가시킨다.

본 논문에서 제안된 동적 흐름제어 기법은 네트워크 상태 예측부분으로 RSTT(Relatively Send Trip Time)와 패킷손실을 이용하여 세 가지 상태로 나누며, slow-start 단계와 congest-avoidance 단계를 이용하여 예측된 네트워크 상태에 따라 효과적으로 전송량을 조절하는 알고리즘이다.

Ⅱ. 관련연구

2.1 화상회의 시스템

Vic는 Mbone에서 사용되는 화상회의 시스템으로 Lawrence Berkeley National Laboratory의 Network Research Group에서 연구 목적으로 개발된 응용프로그램이다. vic은 처음에는 UNIX 환경 하에서 개발 되어 점차 Windows 95 등 여러 운영체제하에서 동작하도록 이식되었다. Intra-H.261, JPEG, cellb 등의 다양한 압축 알고리즘들을 지원하며, CCD 카메라에서 데이터를 읽어오는 여러 가지 프레임 그래버(frame grabber) 하드웨어 드라이버들도 구현되어 사용자 환경에 맞게 선택하여 사용할 수 있도록 제작되었다.

vic는 RTP[5](Real time Transport Protocol)로 구현되며, 영상 데이터와 함께 송수신자의 정보를 전송할 수 있다. vic는 기본적으로 아래의 [그림 1]과 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.

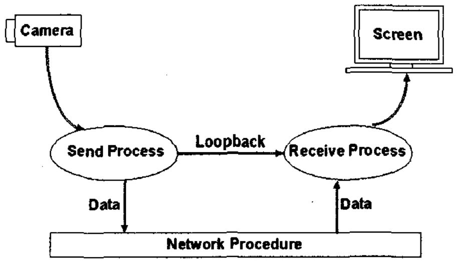


그림 1. Vic의 구조
Fig. 1. Structure of Vic.

2.2 실시간 전송 프로토콜

RTP[5]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. RTP는 음성, 영상 또는 시뮬레이션 데이터와 같은 실시간성 데이터를 전송하기 위한 프로토콜이다. 그러나 RTP는 RSVP[6]와 같이 자원을 예약하지 않으므로 사용자가 요구하는 QoS를 보장해 주지 않는다. RTP는 UDP 위에서 구성되는데 이것은 UDP가 제공하는 멀티플렉싱과 체크섬의 기능을 이용하기 위해서다. 그러나 RTP가 트랜스포트 계층과는 독립적으로 설계되었기 때문에 반드시 UDP위에서 구성되어야 하는 것은 아니다. RTP 자체로는 실시간적 데이터 전송과 QoS에 대한 어떠한 보장도 해 주지 않는다. 또한 손실 없는 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 프로토콜이 간단하여 수행시간이 적기 때문에 실시간 응용프로그램에 알맞다. 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 처리

될 때는 다른 미디어와 동기화(synchronization)를 할 수 있게 한다. 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.

Ⅲ. 기존의 동적 흐름제어 기법

Ingo Busse[7]는 패킷 손실에 의한 동적 흐름제어 기법을 제안하였다. 그는 수신자가 받은 송신자 보고를 이용하여 패킷 손실률과 지연 패킷의 지터(jitter)를 측정하며, [그림2]와 같이 네트워크의 상태를 두 개의 임계값을 적용하여 네트워크 상태를 세 가지 상태, 즉 혼잡, 정상, 무부하로 나눌 수 있다. [그림 2]에서 λ_C는 혼잡 상태의 임계값을 나타내며, λ_U은 무부하 상태의 임계값을 나타낸다. 전체 송신된 데이터의 개수와 수신자가 받은 전체 데이터의 개수를 비교하여 그 차이를 임계값으로 설정한다. Ingo Busse는 비디오 소스에 의해서 가장 적당한 임계값을 λ_C = 4% 그리고 λ_U = 2%라고 제안했다. 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 관찰되었다.

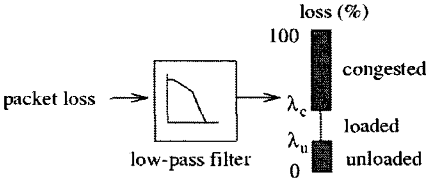


그림 2. 네트워크 상태분류
Fig. 2. Classification of Network Status

왕복 시간에 근거하여 네트워크 상태를 측정하고 측정된 네트워크 상태에 따라 전송률을 변화시키는 기법을 설명한다. 이 기법은 패킷 손실에 근거한 흐름제어 기법에 비해 네트워크에서 패킷 손실이 발생하기 이전에 전송률을 조정하여 네트워크 사용 효율을 증가시킬 수 있으며, 지금 사용하고 있는 네트워크의 대역폭을 효과적으로 이용할 수 있는 장점이 있다.

논문[9]에서는 패킷의 왕복시간(RTT)과 패킷손실을 적용하여 네트워크 상태를 결정하고 결정된 상태에 따라 전송률을 조절하는 기법을 설명한다. 이 기법은 TCP의 흐름제어 알고리즘과 유사하게 전송률을 조절하기 위한 기법으로써 네트워크 상태측정 모듈과 전송량 조절 모듈로 나누어진다.

측정한 RTT에 따라서 임계 값에 대한 비율을 동적으로 변화시켜 네트워크 상태를 정확하게 파악해야 하는데, 대역폭이 좋은 상황에서도 약한 혼잡이 지속되는 현상이 실험을 통해 발견되었다. MaxRTT값이 초기에 너무 낮게 설정이 되면 네트워크 상태결정에서 약한 혼잡이 발생하게 되고 전송량은 낮아지게 된다. 전송량이 낮아지면 일반적으로 이와 비례하여 RTT값이 작아지게 되므로 기존에 설정된 MaxRTT값이 재설정되지 않는다. 이로 인해 RTT에 의해 재설정된 임계값의 비율마저 적용되지 않으므로 결과적으로 약한 혼잡 상태에서 회복하지 못하고 최소 전송률을 유지하게 된다. 위의 그래프에서 보면 잘못된 네트워크 상태 결정에 의해 약한 혼잡상태에서 빠져 나오지 못하고 전송량이 최소 프레임을 계속 유지하게 되는 현상을 발견할 수 있다. 본 논문의 제안된 알고리즘에서는 측정한 전송시간의 흐름에 따라 과거의 상태까지 고려하여 보다 정확한 네트워크 상태 측정을 통해 이러한 문제점을 해결하였다.

Ⅳ. 제안된 동적 흐름제어 기법

4.1 전송시간(Relatively Send Trip Time) 측정

제안된 동적 흐름제어 기법에서는 RTT가 아닌 RSTT를 네트워크 상태 측정요소로 사용하였다. RTT는 앞서 설명했듯이 전송 왕복시간에 대한 절대적인 값을 나타내지만 RSTT는 수신시간이 포함되어 있지 않은 송신시간에 대한 상대적인 값을 나타낸다. 다시 말하면 과거의 송신시간과 현재의 송신시간의 차이를 상대적인 수치를 이용하여 계산한 값이 RSTT이다. 송신자는 다음 수식 (1)을 이용하여 송신자와 수신자 사이의 전송 시간(RSTT)을 계산할 수 있다.

$$RSTT = (Last\ LSR - Cur\ LSR) - (Last\ DLSR - Cur\ DLSR) \quad (1)$$

RSTT는 RTCP의 RR 패킷과 SR 패킷을 이용하여 측정하게 된다. 송신자는 RTCP의 SR 패킷의 LSR에 전송시간을 기록하여 전송한다. 수신자는 SR 패킷에 기록된 SR 전송 시간과 수신자가 SR 패킷을 받은 시간을 RR 패킷의 LSR, DLSR에 각각 기록하여 다시 송신자에게 전송 한다. 송신자는 RR패킷을 받아서 과거의 SR 패킷과 현재의 SR 패킷 송신 시간의 차와 과거의 RR 패킷과 현재의 RR 패킷 수신 시간의 차를 구할 수 있다. 여기서 두 가지 결과 값의 관계는 아래의 [그림 3]과 같다.

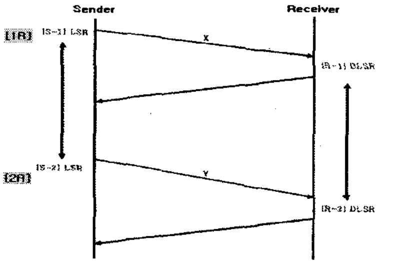


그림 3. RSTT를 계산하기 위한 변수들의 관계
Fig. 3. Computation for RSTT using Variable Relationship

위의 그림에서 RSTT는 x - y와 같다. RSTT가 양수이면 네트워크 상태가 과거보다 나빠졌음을 의미하고, 음수이면 네트워크 상태가 과거보다 좋아졌음을 의미한다. RSTT는 과거와 현재의 RTCP 라운드에 의해 계산되므로, 처음 1R에서는 RSTT값을 측정할 수 없다. 다시 말해서 RSTT가 측정되는 시기는 2R부터 시작된다.

4.2 네트워크 상태 측정

본 절에서는 정확한 네트워크의 상태를 측정하기 위해서 패킷 손실률과 전송시간(RSTT)를 네트워크 상태 파악요소로 사용한다. [그림 4]는 송신자에게 RTCP의 RR 패킷이 도착할 때마다 적용하는 네트워크 상태 측정 알고리즘의 가상코드(pseudo code)이다. 제안된 네트워크 상태 결정 알고리즘은 두 가지 단계를 거쳐 네트워크의 상태를 결정한다. 첫 번째 단계는 (2)~(7) 라인으로서 네트워크 상태를 대역폭 증가, 유지, 감소 세 가지 상태로 분류 한다. 두 번째 단계인 (8)~(12)라인은 첫 번째 단계에서 결정된 네트워크 상태를 재설정 한다.

```
(1)   LONG lCurSTTime = 1000 - ((lCurSRTime - lLastSRTime) - (lCurRRTime - lLastRRTime));
(2)   if(lCurSTTime < lLastSTTime)
      {
(3)       dCurIncreaseFlat = NET_WINDOW_INCREASE;
      }
(4)   else if(lCurSTTime == lLastSTTime)
      {
(5)       dCurIncreaseFlat = NET_WINDOW_MAINTAIN;
      }
(6)   else
      {
(7)       dCurIncreaseFlat = NET_WINDOW_DECREASE;
      }

(8)   if(lPacketLossRate)
      {
(9)       if(dCurIncreaseFlat <= 0)
          {
(10)          dCurIncreaseFlat -= 0.15;
          }
(11)      else
          {
(12)          dCurIncreaseFlat = NET_WINDOW_CONGESTED;
          }
      }
```

그림 4. 제안된 네트워크 상태 측정 알고리즘
Fig. 4. Proposed Network Status Measurement Algorithm

첫 번째 단계에서는 네트워크 상태를 RSTT 값을 기준으로 결정 하게 된다. (1)라인에서는 RSTT를 계산하고 (2)~(7)라인에서는 현재의 RSTT와 과거의 RSTT를 비교하여 1차적으로 네트워크의 상태를 결정하게 된다.

현재의 RSTT가 과거의 RSTT보다 작으면 대역폭이 좋아졌음을 의미 하므로 N_W_INCREASE(증가) 상태로 설정하며, 두 값이 같으면 N_W_MAINTAIN(유지) 상태로 설정 하고, 나머지의 경우 N_W_DECREASE(감소) 상태로 설정 하게 된다.

두 번째 단계에서는 측정된 패킷 손실에 따라서 첫 번째 단계에서 결정된 네트워크 상태를 재설정 하게 된다. (8)~(10)라인은 패킷 손실이 있고 1차적으로 결

정된 네트워크 상태가 증가 또는 유지 일 경우이다. 네트워크 상태는 좋지만 송, 수신자의 시스템 부하에 의한 패킷 손실로 판단하여, 네트워크 상태를 1단계씩 하향 조정을 한다. (11)~(12)라인은 패킷 손실이 있고 1차적으로 네트워크 상태가 감소일 경우로서 네트워크 상태가 혼잡한 것으로 판단하여, N_W_CONGESTED (혼잡)로 재설정 한다.

4.3 전송률 조절

본 절에서는 앞에서 결정된 네트워크 상태에 따라 [그림 5]에서와 같이 과거의 상태를 고려하여 전송률을 조절한다.

```
(1) dFrameControlRatio = (dCurIncreaseFlat * 0.5) + (dLastIncreaseFlat * 0.35) + (dDLastIncreaseFlat * 0.15);
(2) dLastFrame          += dLastFrame * dFrameControlRatio;
```

그림 5. 제안된 전송률 조절 알고리즘
Fig. 5. Proposed Transmission Rate Control Algorithm

(1)라인에서는 현재의 상태에 50%의 가중치를 주며 한 단계 전 과거의 상태에 35%의 가중치를 주고 두 단계 전 과거의 상태에 15%의 가중치를 주어 전송량 계산에 사용할 비율을 계산한다. 이때 네트워크의 각 상태에 따른 비율은 [그림 6]과 같이 정의 된다.

```
#define NET_WINDOW_INCREASE      0.2
#define NET_WINDOW_MAINTAIN      0.05
#define NET_WINDOW_DECREASE      -0.1
#define NET_WINDOW_CONGESTED     -0.2
```

그림 6. 네트워크 상태에 따른 전송률 조절 비율
Fig. 6. Transmission Rate Adjustment using Network Status

N_W_INCREASE(증가) 상태일 때 전송률 조절 비율은 20%이며, N_W_MAINTAIN(유지) 상태일때는 5%이고, N_W_DECREASE(감소)와 N_W_CONGES-TED (혼잡)일 때 각각 -10%, -20%로 정의 된다. (2)라인에서는 위의 그림에서 과거와 현재의 패턴정보에 의해 결정된 전송률 조절 비율을 현재의 프레임에 적용시킨다.

V. 실 험

본 절에서는 제안된 동적 흐름제어 기법을 적용하여 화상회의 시스템의 성능을 측정하였다. 실험에서 사용된 화상회의 시스템은 시뮬레이션 프로그램으로 구현하여 전송량과 패킷 손실, 그리고 RSTT에 따른 네트워크 상태를 측정 하였다. 아래의 [그림 7]은 화상회의 시스템의 데이터 추출을 위한 시뮬레이션 프로그램이다. 위의 프로그램은 현재의 전송량을 그래프로 그려주고 RSTT의 변화에 따라 과거와 현재의 네트워크 흐름 상태를 시뮬레이션 해주도록 구현하였다. 시뮬레이션 프로그램의 구성 요소 5가지로 나누어진다. 첫 번째로 Net Slide그래프는 과거와 현재의 RSTT 값을 출력함으로써 RSTT 수치의 변화를 알아 볼 수 있다. 두 번째로 N-2 그래프는 2단계 전의 과거에서부터 현재까지의 RSTT를 Display하여 네트워크의 흐름을 한눈에 파악할 수 있다. 세 번째로 Net Status 그래프는 측정된 값에 따라 네트워크의 4가지 상태(증가, 유지, 감소, 혼잡)를 보여준다. 네 번째로 데이터 송신 그래프는 현재 전송되고 있는 데이터의 양을 1초 마다 보여줌으로써 측정된 네트워크 상태에 따라 전송량 조절이 잘 되고 있는지 알 수 있다. 다섯 번째로 RTP/RTCP 패킷 분석 데이터에서는 네트워크 상태, 그리고 전송률 및 패킷 손실률, STT값등의 상세 데이터를 출력한다. 실험은 한서대학교와 캐나다와의 구현된 화상회의 연결로 하였으며, 이들사이에 14개의 라우터들이 존재한다.

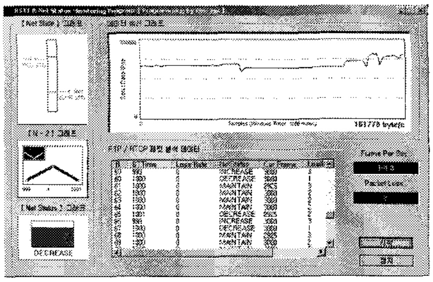


그림 7. 성능실험을 위해 구현한 시뮬레이션 프로그램
Fig. 7. Network Simulation Program

제안된 동적흐름제어 기법의 성능 측정과 기존의 동적 흐름제어 기법과의 성능 비교를 위해서 4가지 실험을 하였다. 첫 번째로 [그림 8]은 현재 측정된 RSTT 만을 네트워크 상태 결정에 적용시킨 경우에 대한 성능 측정 그래프 이다. 현재의 RSTT값만 가지고 네트워

크의 상태를 결정하기 때문에 RSTT값의 조그만 변화에도 민감하게 반응하여 진동현상이 일어나는 것을 볼 수 있다. [그림 9]는 과거의 두 단계 전까지의 RSTT를 네트워크 상태 결정에 적용한 실험 결과 그래프이다.

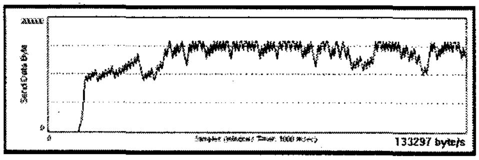


그림 8. RSTT-1 성능측정 그래프
Fig. 8. Performance Measurement Graph (RSTT-1)

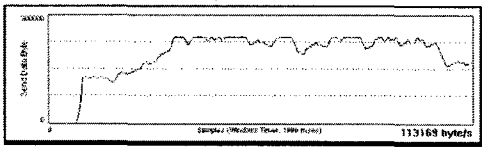


그림 9. RSTT-3 성능측정 그래프
Fig. 9. Performance Measurement Graph (RSTT-3)

그림[9] 실험에서는 현재의 RSTT에 50%의 가중치를 주고 한 단계 전 과거의 RSTT에 35%의 가중치를 주었다. 그리고 두 단계 전 과거의 RSTT에는 15%의 가중치를 적용하였다. 위의 그림은 안정적으로 영상 데이터를 전송하는 것을 볼 수 있으므로 본 논문에서 적용하였다.

다음은 기존 흐름제어 기법인 TLFC 알고리즘과 제안된 알고리즘의 성능 비교 실험이다. 기존의 흐름제어 기법과 제안된 흐름제어 기법의 성능을 비교 실험하기 위해서는 네트워크의 상태를 정확하게 파악할 수 있는 ICMP의 PING을 사용해야 하지만 ICMP패킷 자체가 네트워크에 상당한 부하를 가져오기 때문에 패킷 손실률과 RSTT와 RTT 그래프로써 성능을 비교하였다. [그림 10]은 논문[9]의 TLFC의 알고리즘의 실험 결과를 나타낸 그래프이다. 위의 실험을 통해 앞서 말한 TLFC의 문제점으로 인해 두 가지의 문제가 발생하게 됨을 발견하였다. 첫 번째로 과거의 RTT에 비해 현재의 RTT가 얼마나 증가되었고 감소되었는지에 따라 비율을 조절하는 것이 아니라 단지 과거보다 증가(TRUE) 또는 감소(FALSE)함을 가지고 각 네트워크 상태에 대한 비율을 조절하기 때문에 네트워크 상태의 잘못된 예측이 많다. 두 번째로는 잘못된 MaxRTT값이 설정 되었을 경우이다. MaxRTT가 초기에 낮은 값으로 설정되었을 때는 지속적으로 약한 혼잡상태가 발생하기 때문에 실제 네트워크 상태와는 반대로 전송률이 최저 수치까지 감소하게 되었다. MaxRTT가 실제 대역폭에 비해서 높은 값으로 설정 되었을 때는 무부하와 정상상태에 머물러있는 현상이 발생하였다. 결국 실제로는 혼잡상태임에도 불구하고 전송률을 계속 증가시키거나 유지하게 되어 패킷 손실이 늘어나는 현상을 발견하였다.

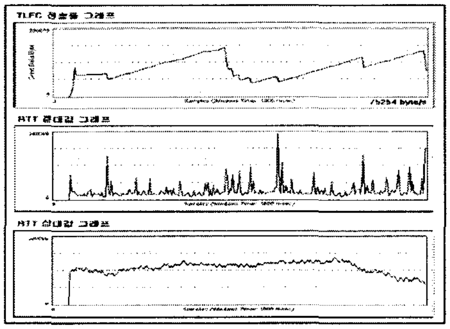


그림 10. 논문9의 TLFC의 성능측정 그래프
Fi. 10. Performance Measurement Graph (Reference Paper[9])

[그림 11]은 제안된 흐름제어 기법의 성능측정 그래프이다. 위의 그래프를 보면 제안된 흐름제어 기법은 미리 정의된 네트워크의 상태가 아닌 RSTT의 패턴에

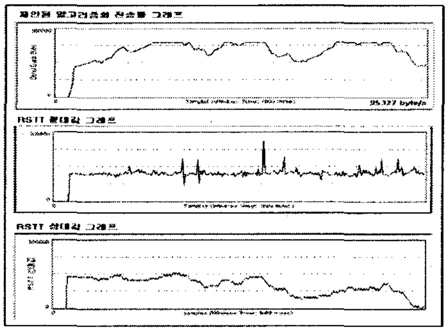


그림 11. 제안된 알고리즘의 성능측정 그래프
Fig. 11. Performance Measurement Graph (Proposed Algorithm)

따라 전송률을 조절하기 때문에 실제 대역폭에 민감하게 반응함을 알 수 있다. 패킷 손실이 발생한 곳을 보면 급격히 감소한 부분과 약간 감소한 부분이 있는데 급격히 감소한 부분은 연속적인 DECREASE 상태와 패킷 손실에 의한 것이고 약간 감소한 부분은 RSTT가 좋은 상태로 측정되었음에도 불구하고 패킷 손실이 발생하였기 때문에 시스템에 의한 부하로 예상된다.

아래의 <표 1>은 TLFC와 제안된 흐름제어 기법의 패킷손실에 대한 실험 결과를 나타낸 것이다.결과의 정확성을 위해 두 번의 성능 실험을 하였고 제안된 흐름제어 기법이 TLFC 기법에 비해 전송량도 많고 패킷 손실률도 적다는 것을 볼 수 있다. 이는 왕복시간을 이용하여 각 네트워크 상태에 따른 비율을 조절하는 것보다 RSTT의 패턴을 적용하는 것이 성능 측면에 있어서 더욱 우수하다는 것을 나타낸다. 본 실험에서는 RTT를 적용한 것 보다 RSTT의 패턴을 적용한 알고리즘이 변화하는 네트워크 상태와 패킷 손실에 더욱 빠르게 대처함을 보였다.

표1. 패킷 손실 비교
Table 1. Comparison of Packet Loss

	TLFC 기법	제안된 흐름제어 기법
총 전송 패킷의 개수 (평균)	17249	22716
총 손실된 패킷의 개수 (평균)	118	32
패킷 손실 비율	0.684 %	0.140%
실험시간	40분	40분

V. 결 론

전 세계적으로 멀티미디어 관련 서비스가 늘어나고 화상회의 시스템을 이용하는 사용자의 수가 증가하면서 네트워크 상에서 안정적인 대역폭을 보장받지 못하고 있다. 화상회의 시스템이 변화가 심한 네트워크 상황에 안정적으로 대처하고 품질이 높은 서비스를 제공하기 위해서는 기존의 TCP 기술로는 부적합하다. 본 논문에서는 RTP와 RTCP를 이용하여 RSTT를 측정하고, 과거와 현재 RSTT의 9가지 패턴에 따라 동적으로 흐름제어를 하였다. 또한 패킷 손실률을 사용하여 시스템 부하에 의한 혼잡도 고려하였다. 전송률 조절 부분에서는 TCP의 혼잡제어 메커니즘인 slow-start단계와 congest-avoidance 단계를 적용하여 전송량을 효율적으로 조절한다. 왕복시간과 패킷손실을 적용한 흐름제어와 RSTT를 적용한 제안된 알고리즘을 비교 실험하였고, 시뮬레이션 프로그램을 통해 패킷 손실률과 안정적인 전송량 조절 측면에서 뛰어난 성능을 보임을 증명 하였다.

향후 연구과제로는 다음과 같이 두 가지로 나눌 수 있다. 첫 번째로 RTCP 패킷에 대해 동적으로 대역폭을 할당하고 RTCP 패킷의 빈도수를 조절하여 RTCP 패킷 손실로 인한 타임아웃을 줄일 것이다. 두 번째로 Kalman 필터를 적용하여 수집된 데이터로 보다 정확한 네트워크 예측을 가능케 하도록 연구해 나갈 것이다.

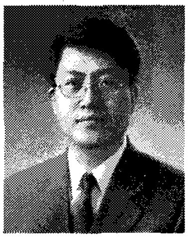
참고문헌

[1] http://www-nrg.ee.lbl.gov/vic/
[2] T. Turletti, "The INRIA Videoconferencing System (IVS)", conneXions - The Interoperability Report, Vol.8, No. 10, pp. 20-24, October 1994.
[3] R.Fredrick, "nv", Manual Pages, Xerox Palo Alto Research Center.
[4] Paul Ferguson and Geoff Huston, "Quality of Service - DeliveringQoS on the Internet and in Corporate Networks", Wiley Computer Publishing pp.24, 1998.
[5] Schulzrinne, Casner, Frederick, and Jacobson, V., "RTP: A transport protocol for real-time applications.", RFC1889, January 1996.
[6] R. Braden, Ed., L. Zhang, S. Berson, S.Herzog, and S. Jamin, "Resource ReSerVation Protocol (RSVP)", RFC2205, September 1997.
[7] Ingo Busse, Bernd Deffner, Henning Schulzrinne, "Dynamic QoS Control of Multimedia Application based on RTP", May 1995, http://www.fokus.gmd.de/step/acontrol/ac.html.
[8] 나승구, 고동환, 안종석, 김승범, "회의 시스템의 성능을 개선하기 위한 두개의 동적 흐름 제어방식의 비교", SoftExpo 97 (p381-390), 12월 1997.
[9] K. Hasung"Dynamic flow control algorithm on teleconferrence system", CISST, pp249-pp252, June 2003

저자소개

구하성(Hasung Koo)

1989년 3월 광운대학교 전자통신공학과 (공학사)
1991년 9월 광운대학교 전자통신공학과 (공학석사)
1995년 8월 광운대학교 전자통신공학과 (공학박사)
1995년 9월부터 1997년 3월 기아정보시스템 알고리듬 개발팀장
1997년 3월부터 현재 한서대학교 컴퓨터정보학과 교수
※관심분야 : 영상처리, 생체인식



심종익(Jongik Shim)

1985년 인하대학교 전자계산학과 졸업(학사)
1987년 인하대학교 대학원전자계산학과졸업(석사)
1998년 인하대학교 대학원 전자계산공학과 졸업(공학박사)
1986년 ~ 1993년 LG산전 연구소 선임연구원
1994년부터 현재 한서대학교 컴퓨터정보학과 교수
※관심분야 : 인터넷 통신, 데이터베이스, 실시간 시스템