

고품질 실시간 스트리밍 서비스 제공을 위한 네트워크 모니터링 기법

준희원 김 용 술*, 정희원 김 화 성*

Network Monitoring Mechanism For High Quality Real-Time Streaming Service

Yong-sul Kim* *Associate Member*, Hwa-sung Kim* *Regular Member*

요 약

멀티미디어 서비스의 증가는 인터넷 콘텐츠의 새로운 국면으로 나타나고 있다. IP텔레폰, 화상채팅과 같은 멀티미디어 서비스는 대역폭, 지터, 패킷유실, 전송지연 등의 네트워크 파라미터에 민감한 데이터이다. 멀티미디어 서비스의 QoS 보장을 위해서 RTP/RTCP를 이용하여 네트워크 상황을 판단하고 전송률을 조절 하는 기법이 연구되고 있다. 본 논문에서는 최선형(Best-Effort) 서비스를 제공하는 현 인터넷 구조에서 RTP/RTCP 구조를 확장하여 단방향 전송 지연시간을 구하고 이를 이용하여 네트워크 상황을 판단하는 기법을 제안한다. 또한 단방향 전송 지연을 이용하여 급변하는 네트워크 상황에 즉각적으로 대처하기 위한 동적인 RTCP 구조를 제안한다. 모의 실험 결과 제안한 단방향 전송 지연 기법은 GPS를 이용한 단방향 전송 기법과 유사한 성능을 보였으며, 제안한 동적 RTCP 기법은 적은 RTCP RR패킷 전송으로 패킷 손실률이 감소하는 효과를 얻었다.

Key Words : RTP, RTCP, Monitoring, QoS, OWD

ABSTRACT

Recently, there has been explosive growth in Internet service contents, and we have witnessed the emergence of many real-time applications on the Internet. These types of applications require special support from the underlying network such as reliability, timeless, and guaranteed delivery, as well as different levels of service quality. Estimation network status and transcoding data rate are researching for QoS of multimedia service by using the RTP/RTCP. We propose two schemes that the way to calculate the one-way delay by expanding the structure of RTP/RTCP in Internet that provides us "best effort" service and to estimate the network condition by using this in this paper. And we propose the dynamic RTCP mechanism based on one-way delay to deal with rapidly changed network situation immediately. In simulation, proposed monitoring schemes using the one-way delay has similar in performance to monitoring schemes using the GPS, proposed dynamic RTCP scheme obtain the desired result of decreased packet loss rate by sending RTCP RR packet lower than other schemes.

※ 본 연구는 정보통신부 및 정보통신진흥원의 대학 IT 연구센터 지원 사업의 연구결과 및 한국과학재단 특정기초연구 (R01-2006-000-10199-0)지원으로 수행되었음.

* 광운대학교 전자통신공학과 네트워크컴퓨팅 연구실({dragonwine, hwkim}@kw.ac.kr)

논문번호 : KICS2006-01-021, 접수일자 : 2006년 1월 11일, 최종논문접수일자 : 2006년 10월 23일

I. 서론

예전에 비해 인터넷의 속도가 점차 고속화됨에 따라 VoD(Video on Demand)나 IP텔레폰, 화상채팅과 같은 멀티미디어 서비스가 증가 하고 있다. 이러한 멀티미디어 서비스는 일정한 대역폭의 자원을 요구하고 지터나 패킷 유실, 전송 지연시간에 매우 민감한 데이터이다.

멀티미디어 서비스의 QoS(Quality of Service) 보장을 위해 IETF(The Internet Engineering Task Force)에서 자원을 예약하여 QoS를 보장할 수 있는 IntServ(Integrated Services)와 등급에 따라 차등적인 서비스를 제공하여 품질을 보장할 수 있는 DiffServ(Differentiated Services)기술을 제시 하였다^{[1][2]}. 하지만 이러한 기술은 네트워크 라우터를 교체해야 하는 경제적 부담 때문에 널리 사용되진 못하고 있다.

현재 최선형 서비스(Best-Effort-Service)를 제공하고 있는 인터넷 환경에서 멀티미디어 서비스의 QoS를 제공해 주기 위해 지터(Jitter), RTT(Round Trip Time), 패킷유실(Packet Loss)등의 네트워크 파라미터를 이용하여 네트워크 상태를 판단하고 전송률을 조절해 주는 TFRC(TCP-Friendly Rate Control), RAP(Rate Adaptive Protocol), TLFC(TCP Like Flow Control)과 같은 흐름 제어 기법이 널리 사용되고 있다^{[3][4][5]}. 흐름 제어에 사용되는 네트워크 파라미터 값 중에서 지터값은 멀티미디어 서비스에서 품질을 판단하는데 중요한 의미를 갖는 파라미터이다. 그러나 지터값은 이전 패킷과의 지연시간을 상대적으로 판단한 것으로 네트워크 상태를 절대적으로 판단하는데 어려움이 있다. 따라서 실제적으로 네트워크 상태를 판단 할 때는 패킷유실과 RTT 같은 절대적인 값을 이용하여 네트워크 상태를 판단한다. 하지만 흐름 제어 기법에서 사용하고 있는 RTT값은 서버와 클라이언트 사이의 왕복 전송 지연시간을 측정 한 값으로써 서버에서 클라이언트로 단방향으로 데이터가 전송되는 특징을 가지고 있는 스트리밍 서비스에 적용하였을 경우 클라이언트에서 서버로 전송되는 전송 지연시간에 영향을 받아 네트워크 상황을 판단하는데 오차가 발생한다. 또한 주기적으로 전송되는 RTCP(Real-Time Control Protocol) 패킷을 이용하여 RTT 측정 시 전반적인 네트워크 전송 지연시간 측정이 아닌 RTCP 패킷이 전송 되는 순간의 값만이 적용됨으로 특정 네트워크 상황에 국한되어 네트

크 상태를 판단해야 하는 단점이 있다^[6]. RTCP 패킷은 전체 패킷에 5%이내로 주기적으로 전송되기 때문에 네트워크 상황에 신속하게 대처 할 수 없다. 따라서 멀티미디어 서비스의 QoS를 보장하기 위해 이러한 특징을 고려한 확장형 RTP(Real-Time Transport Protocol)/RTCP 구조가 필요하다^[7].

본 논문은 RTT를 이용한 모니터링 기법의 단점을 개선하기 위하여 단방향 전송 지연(One-Way Delay)를 이용한 네트워크 모니터링 기법을 제안하고, 단방향 전송 지연을 이용하여 네트워크 상황에 신속하게 대처하기 위한 동적인 RTCP 구조를 제안한다. 본 논문의 구성은 다음과 같다. 2장에서 표준 멀티미디어 스트리밍 프로토콜과 네트워크 모니터링 기법에 대한 관련 연구에 대해서 기술하고, 3장에서 단방향 전송 지연 기법을 이용한 모니터링과 동적인 RTCP 구조에 대해 기술한다. 4장에서는 모의실험을 통하여 성능평가를 하고, 마지막 5장에서 결론을 맺는다.

II. 관련 연구

2.1 RTP: Real-Time Transport Protocol

RTP는 오디오, 비디오 및 시뮬레이션 데이터와 같은 실시간 데이터를 멀티캐스트 또는 유니캐스트 네트워크를 이용해서 전송하는 응용 서비스에 알맞은 단말-대-단말 네트워크 전송 기능을 제공한다. RTP는 자원 예약을 수행하지 않으며, 따라서 적시 전달, 순차 전달과 같은 서비스 품질도 보장하지 않는다. RTP 데이터 전송 기능은 제어 프로토콜에 의해 확장되는데, RTCP라 불리는 이 제어 프로토콜은 데이터의 전달 상황을 감시하며, 최소한의 제어 기능과 매체 식별 기능을 제공한다. RTCP는 2.2절에서 자세하게 알아본다. RTP와 RTCP는 하위의 전송 및 네트워크 계층에 무관하게 설계되었다. RTP는 별개의 독립 계층으로 구현되기 보다는 특정 응용에서 요구되는 정보를 제공하여 프로토콜의

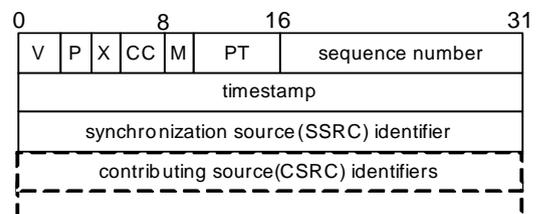


그림 1. RTP 패킷 구조

처리가 응용의 처리 과정으로 통합될 수 있도록 설계되었다. 따라서 기존의 프로토콜들과는 달리 RTP는 응용의 필요에 따라 헤더를 변경하거나 추가하여 응용에 맞는 프로토콜이 될 수 있도록 하는 일종의 맞춤형 프로토콜이다. 그림 1은 RTP 패킷 형태이다.

2.2 RTCP:Real-Time Control Protocol

RTCP는 데이터 송수신 간의 분실된 패킷 수, 지터 간격, 앞의 패킷과의 지연시간 등의 QoS 정보를 교환하여 응용이 적당한 QoS를 평가하여 Adaptive encoding을 제공하도록 한다. 또한 RTCP는 많은 참여자들의 스케일을 위해서 패킷 송신율을 계산하고 사용자 인터페이스의 참여자 ID를 지칭하는 최소한의 세션 제어 정보를 나른다. 또한 데이터의 원천지 식별자가 충돌이 되거나 다시 만들 경우에 변경되어야 하므로 CNAME(Canonical NAME)이라 부르는 영구 트랜스포트 식별자를 나른다. RTCP는 제어 패킷을 주기적으로 모든 참여자에게 전송한다. 그림 2는 RTCP 복합 패킷 양식을 보여준다.

RTCP 패킷은 고정된 헤더 필드와 패킷 타입에 따라서 가변의 길이를 갖는 필드가 붙는다. 효과적인 정보 전송을 위해서 여러 RTCP 패킷들이 한 개의 UDP 패킷으로 전송될 수 있다.

RTP는 수명에서 수천 명의 참가자를 하나의 세션에 참가시킬 수 있도록 설계되었다. 오디오 회의의 경우에 데이터 패킷은 참가자의 수에 상관없이 비교적 일정한 비트율(Bit-rate)을 가지지만(언제나 발언하는 사람은 한 두 사람이므로) 제어 패킷의 경우에는 참가자의 수에 비례하여 비트율이 증가(각 참가자가 나머지 모두에게 일정한 간격으로 제어 패킷을 전송하기 때문에)하게 된다. 따라서 제어 패킷의 전송 간격은 제어되어야 한다. RTCP에 할당되는 대역폭은 세션 대역폭의 5%로 고정되는 것이 바람직하다.

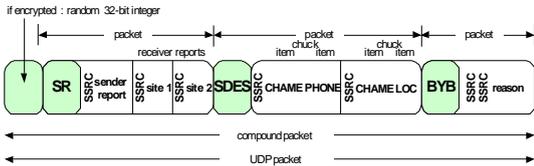


그림 2. RTCP 패킷 전송 형태

2.3 TLFC 알고리즘

TLFC 기법은 패킷유실, RTT와 같은 네트워크

파라미터를 이용하여 네트워크 상태에 적응적으로 전송률을 조절하는 기법이다. 네트워크 파라미터는 RTCP SR패킷과 RR패킷에 의해 파악할 수 있으며 값을 이용하여 전송률을 동적으로 조절한다.

그림 3은 네트워크 상태를 판단하는 기법이다⁵⁾. MaxRTT는 최대 전송 지연 값으로 이 값을 기준으로 상한 값(LoadRTT : $MaxRTT * UpperRatio$)과 하한 값(UnLoadRTT : $MaxRTT * LowerRatio$)을 계산한 후 AvgRTT값이 어느 값 사이에 존재하는지에 따라 네트워크 상태를 CONGESTED, LIGHTCONGESTED, LOADED, UNLOADED로 판단한다. 판단된 네트워크 상태에 따라 전송률을 조절하여 네트워크에 적응적으로 대처한다.

네트워크 상태에 적응적으로 대처하기 위해 측정된 네트워크 상태에 따라 전송률을 조절한다. 그림 4는 전송률 조절 기법을 보여준다⁵⁾. 전송률 조절은 RTCP RR 패킷을 수신하였을 때 ① 수행 된다. LOADED 상태일 때는 ②와 같이 StableRate를 구한다. StableRate는 네트워크 상태가 UNLOADED일 때 네트워크 상태가 슬로우 스타트(Slow-start) 단계인지 혼잡 회피(Congestion avoidance)단계인지 판단하는 기준이 된다. ③은 네트워크 상태가 CONGESTED일 때 전송률을 낮추는 부분이다. 전송률을 낮출 때 MINRATE(최저전송률)보다 전송률이 낮지 않게 조절한다. 혼잡이 발생 가능한 상태를 LIGHTCONGESTED라고 하며 그림 4의 ⑤ 부분이다. 혼잡 가능상태일 때는 혼잡상태와 마찬가지로 전송률을 조절하여 혼잡상태에 대비한다.

UNLOADED상태일 때는 네트워크에 부하가 걸리지 않은 상태로 전송률을 증가시킨다. TCP의 흐름제어 방식과 마찬가지로 슬로우 스타트를 하며

```

AvgRTT = (1-0.75) * AvgRTT + 0.75 * CurRTT;
If( CurRTT > MaxRTT ){
    MaxRTT = CurRTT;
    LoadRTT = UpperRatio * MaxRTT;
    UnloadRTT = LowerRatio * MaxRTT;
}

If( LossRate > 0 )
    NetworkState(CONGESTION);
else{
    If(AvgRTT <= UnloadRTT)
        NetworkState(UNLOADED);
    else if((AvgRTT > UnloadRTT) &&
            (AvgRTT <= LoadRTT))
        NetworkState(LOADED);
    else if(AvgRTT > LoadRTT)
        NetworkState(LIGHTCONGESTED);
}
    
```

그림 3. 네트워크 상태 판단 기법 슈도코드

```

1 if(Received RTCP RR Packet)
2 {
3     if(NetworkState == LOADED){
4         StableRate = StableRate * (1-0.25) + CurRate * 0.25;
5     }else if(NetworkState == CONGESTED){
6         CurRate = max(CurRate * 0.975, MINRATE);
7         StableRate = StableRate * 0.75;
8     }else if (NetworkState == LIGHTCONGESTED){
9         CurRate = max(CurRate * 0.95, MINRATE);
10    }else if (NetworkState == UNLOADED){
11        if(CurRate < StableRate){
12            CurRate += (InterRRTime/AvgRTT)*MSS;
13        }else{
14            CurRate += ((InterRRTime / AvgRTT) * MSS * MSS) / CurRate;
15        }
16    }
17 }
    
```

그림 4. 전송률 조절 슈도코드

슬로우 스타트의 기준은 *StableRate*이다. ⑥과 ⑦에서 *InterRRTime*은 RR Packet의 간격이다. *InterRRTime*값을 *AvgRTT*값으로 나누어 RR 전송 주기 동안 얼마만큼의 데이터를 전송하였는지를 가시적으로 계산 후 *MSS*(Maximum Segment Size)값을 곱해줌으로써 전송률을 계산한다. ⑥은 슬로우 스타트 단계이고 ⑦은 혼잡 회피 단계이다. 스트리밍 서비스를 제공하는 서버는 클라이언트에서 주기적으로 전송하는 RTCP RR 패킷이 도착 할 때 마다 전송률을 조절 하여 네트워크 상태에 적응적으로 대처 하지만 RTT값을 이용한 네트워크 모니터링 기법은 단방향으로 전송되는 스트리밍 서비스에 적용하였을 경우 피드백 전송 지연까지 적용되어 오차가 발생할 수 있으며, RTT 값을 구할 때 보내고 있는 데이터의 전송 지연이 아닌 RTCP SR 패킷과 RR 패킷 사이에 전송 지연만을 이용함으로 특정 네트워크 상황에 흐름제어 알고리즘이 반응 할 수 있다는 단점을 가진다. 또한 주기적으로 정해진 (일반적으로 5초) RTCP 패킷은 급변하는 네트워크 상황에 신속히 대처할 수 없다는 단점을 가지고 있다.

III. 단방향 전송 지연을 이용한 모니터링 기법

3.1 연구의 필요성

멀티미디어 서비스는 거의 RTP, RTCP 프로토콜을 이용하여 전송 된다. 멀티미디어 서비스의 QoS 제공을 위하여 RTCP패킷을 이용한 네트워크 모니터링 기법이 사용되고 있다. RTCP패킷은 총 5가지 패킷으로 구분되며 이중 RTCP SR(Sender Report), RTCP RR(Receiver Report)을 이용하여 RTT, 지터, 패킷유실 등의 네트워크 파라미터 값을 얻을 수 있다. 이 값을 이용하여 네트워크 상태를 판단 전송률을 조절 하게 된다. 하지만 RTCP 패킷을 이용하였을 경우 크게 두 가지 문제가 존재한다.

첫째, 현재 네트워크 상태를 판단하기 위한 모니터링 기법은 주로 RTT와 패킷유실 파라미터를 이용하고 있다. 하지만 멀티미디어 서비스는 대부분 데이터가 서버에서 클라이언트로 전송되기 때문에 왕복 시간을 이용한 모니터링 기법 사용 시 피드백 전송 지연에 대한 값이 반영되어 오차가 발생할 수 있다. 이러한 문제를 해결하기 위한 기법은 단방향 전송 지연을 이용한 모니터링 기법이 있다. 단방향 전송 지연 값을 구하기 위해선 서버와 클라이언트 시스템의 시간 동기화 작업이 필요하며 가장 어려운 문제이다. 이러한 문제를 해결하기 위해 TCP의 서비스를 받는 환경에서 RTT를 이용한 네트워크 모니터링 기법과 GPS(Global Positioning System)를 이용한 시스템 동기화 기법이 제안되었다⁹⁾. 하지만 RTT를 이용한 기법은 서버와 클라이언트에서 모든 데이터에 피드백을 전송해야 한다는 오버헤드가 존재하고, GPS를 이용한 기법은 모든 클라이언트, 서버 시스템에 GPS가 장착되어있어야 한다는 경제적 부담이 있다.

둘째, TFRC, RAP, TLFC와 같은 모니터링 기법은 클라이언트로부터 RTCP RR패킷을 수신된 정보를 이용하여 네트워크 상태를 판단한다. RTCP 패킷은 전체 송신되는 패킷량의 5%이내로 제한되어 있지만 실제로는 1~5초 정도 간격으로 전송되고 있다. 때문에 급변하는 네트워크 상태를 정확하게 판단할 수 없으며, 적절한 전송률을 결정할 수 없다.

본 논문에서는 두 가지 문제점을 해결하기 위해 단방향 전송 지연 값을 구하기 위한 동기화 기법과 이를 이용한 동적 RTCP 구조를 제안한다.

3.2 단방향 전송 지연을 이용한 모니터링 기법

본 논문에서 제안하는 알고리즘은 세션이 설정된 후 RTT를 이용하여 서버와 클라이언트의 시간을 동기화시킴으로써 상대적인 단방향 전송 지연시간을 구하고 네트워크 상태를 판단함으로써 효율적이고 경제적인 알고리즘을 제안한다.

그림 5는 단방향 전송 지연을 구하기 위한 동기화 순서를 도식화 한 것이다. 서버를 동기화하기 위해 RTT를 구한다. RTT값을 구하기 위해 Server시간을 샘플링한 시점의 시간을 *ServerTime*이라고 하고 클라이언트의 시간을 *ClientTime*이라고 한다. *ClientTime*은 현재 알 수 없는 미지수이다.

서버에서 클라이언트까지 걸리는 시간은 α 이며 클라이언트에서 서버까지 걸리는 시간을 β 라고 한다. 서버에서 *ServerTime*을 클라이언트로 보냈을 때 패킷

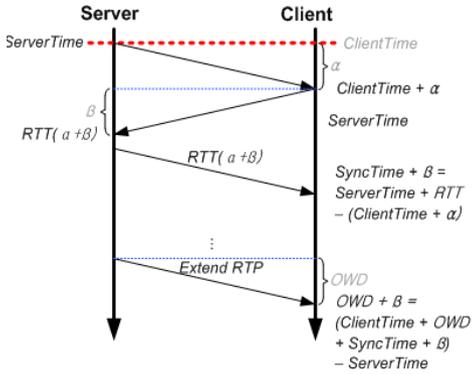


그림 5. 서버/클라이언트 동기화 순서

표 1. 서버/클라이언트 동기화 알고리즘

Ideal case	$SyncTime = ServerTime - ClientTime$ ①
	$(ClientTime + SyncTime) - ServerTime = 0$ ②
	$(ClientTime + OWD + SyncTime) - ServerTime = OWD$ ③
Proposed case	$SyncTime = ServerTime + \alpha - (ClientTime + \alpha)$ ④
	$SyncTime + \beta = ServerTime + RTT - (ClientTime + \alpha)$ ⑤
	$(ClientTime + SyncTime + \beta) - ServerTime = \beta$ ⑥
	$(ClientTime + OWD + SyncTime + \beta) - ServerTime = OWD + \beta$ ⑦

이 도착하는 시점의 클라이언트 시간은 $ClientTime + a$ 이며 $ServerTime$ 의 시간을 저장한 후 서버로 되돌려준다. 클라이언트에서 패킷을 수신한 서버는 왕복 지연시간 RTT 를 구할 수 있으며 $RTT = a + \beta$ 이다. RTT 값을 클라이언트로 전송하면 클라이언트는 $ServerTime$ 과 $RTT(a + \beta)$ 값을 알 수 있다. 표 1에서 서버와 클라이언트의 동기화가 완벽한 경우를 이상적인 경우라 하며 이때 $ServerTime$ 에서 $ClientTime$ 을 뺀 시간을 $SyncTime$ 이라 했을 때, 단방향 전송 지연 값을 구하기 위한 공식은 식 ③과 같다. 여기서 OWD (One-Way Delay)는 단방향 전송 지연을 나타낸다. a 는 OWD 와 같은 값이며 $Client + OWD$ 값은 $ServerTime$ 이 도착했을 때 시간이다. 이런 이상적인 경우(서버와 클라이언트의 동기가 맞을 경우) 아닌 실제 상황에선 $ClientTime$ 을 알아낼 수 있는 방법이 없다. 그렇기 때문에 정확한 $SyncTime$ 을 구할 수 없다. 그러므로 본 논문에서 제안하는 알고리즘은 다음과 같다. 식 ④는 식 ①에 a 값을 더하고 뺌으로 성립함을 알 수 있다. 하지만 위 식은 식 ①과 같이 동시에 성립할 수 없으므로 양변에 β 값을 더해줌으로써 식 ⑤를 구할 수 있다. 식 ⑤의 우측에 있는 식은 우리가 알 수 있는 값이다. 이를 통해 $SyncTime + \beta$ 를 구할 수 있다. 식 ⑥은 식 ②의 양변에 β 값을 더해 줌으로써 성립이 되며 이

표 2. β 에 의한 오차율

Case	Error Rate
LoadOWD	$SOWD += \beta * (1 - UpperRatio)$
UnLoadOWD	$SOWD += \beta * (1 - LowerRatio)$

표 3. 오차율 감소를 위한 증가 값

Case	Increase Threshold
LoadOWD	$LoadOWD += RTT * (1 - UpperRatio) / \alpha$
UnLoadOWD	$UnLoadOWD += RTT * (1 - LowerRatio) / \alpha$

를 통해 식 ⑦이 성립함을 알 수 있다. 위 식을 사용하였을 경우 항상 $OWD + \beta$ 의 값을 구할 수 있으며 β 값은 고정된 상수로써 2.3절에서 설명한 네트워크 상태 판단 알고리즘에 적용하였을 경우 β 값에 따른 네트워크 상태 판단의 오차가 발생한다.

표 2는 2.3절에서 설명한 각 해당 값과 비교하였을 경우 발생하는 오차율을 나타낸다. $SOWD$ (Standardization OWD)는 클라이언트에서 구한 OWD 의 구간 평균값으로 표2와 같은 만큼의 지연시간이 더 크게 된다. 이와 같은 오차를 줄이기 위해 GPS를 이용한 OWD 측정 시의 네트워크 상태 판단 기법의 파라미터 값보다 $UpperRatio$ 와 $LowerRatio$ 값을 크게 해야 한다. 표 3은 서버, 클라이언트 동기화시 알게 되는 $RTT(a + \beta)$ 를 이용한 Ratio 증가 기법이다. 여기서 a 는 상수 값으로 2이상의 값을 가진다. 위 식을 적용하였을 때 보다 GPS를 이용한 단방향 전송 지연 값과 유사하게 네트워크 상태를 판단 할 수 있다.

3.3 단방향 전송 지연을 이용한 모니터링 기법

단방향 전송 지연을 적용한 모니터링 기법은 2.3절에서 설명한 TLFC 알고리즘을 기반으로 한다. RTT 값 대신 OWD 값을 사용하여 네트워크 상태를 판단한다. 클라이언트에서 OWD 를 구하는 슈도코드는 그림 6과 같다. 앞에서 설명하였던 TLFC 알고리즘은 $MaxRTT$ 와 $AvgRTT$ 를 서버에서 구하였지만 단방향 지연을 이용한 모니터링 알고리즘은 클라이언트에서 각 해당하는 값을 계산하여 최종적으로 OWD 를 구한다.

①은 RTP 패킷을 받은 경우 수행되는 루틴이다. ②는 현재 시간을 측정하는 것이다. ③식을 이용하여 OWD 를 구한 후 현재 값과 이전 값을 누적하여 $SOWD$ 에 저장하며 이 값이 RTT 를 이용한 TLFC의 $AvgRTT$ 에 해당하는 부분이다. ⑤, ⑥에서 패킷유실과 최대 OWD 값을 구하고 RTCP RR Timer가 만

```

1 if(Received RTP Packet)
2 {
3   ClientTime = GetTime();
4   OWD = (ClientTime + SyncTime) - ServerTime;
5   SOWD = SOWD * (1 - α) + OWD * α;
6   Loss += CurSeq - LastSeq - 1;
7   if(OWD > MaxOWD)
8   {
9     MaxOWD = OWD;
10  }
11 }
12 if(Expired RTCP RR Timer)
13 {
14   Send RTCP RR Packet();
15   Loss = 0;
16 }
    
```

그림 6. 클라이언트 슈도코드

로 되었을 경우 ⑦ SOWD, MaxOWD, Loss값 들을 서버로 보낸다.

3.4 동적 RTCP 기법

기존의 RTCP RR 패킷은 RTCP RR Timer가 만료되었을 경우에만 전송되었기 때문에 네트워크 상황에 빠르게 대처 할 수 없었다. 본 논문에서 제안하는 동적인 RTCP는 단방향 전송 지연을 이용하였을 경우 클라이언트에서 네트워크 상태를 판단할 수 있다는 이점을 이용한 RTCP RR 전송 주기 조절 기법이다. 네트워크 혼잡이 발생 했을 경우에는 바로 서버에게 OWD를 보낸다. 하지만 네트워크 상태가 혼잡한 경우 RTCP 패킷을 계속 보내는 것을 막기 위하여 MIN Timer를 사용한다.

RTCP RR 패킷을 전송하는 시점은 다음과 같다.

1. RTCP RR 패킷 Timer 만료
2. 패킷유실 개수 / 수신패킷개수 > CongestionRate
3. 네트워크 상태 == UNLOADED && 현재 전송률 < MAX Transmission Rate

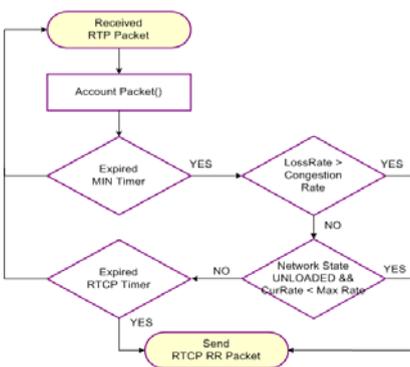


그림 7. 동적인 RTCP 순서도

제한한 패킷 전송 주기인 두 번째와 세 번째 경우는 급격한 RTCP RR 패킷 개수의 증가를 방지하기 위해 MIN Timer를 동작시켜 Timer가 만료된 이후 전송 가능하다. MIN Timer는 전체 패킷 송신량의 5%이내가 되도록 주기를 조절 한다. 즉 제한한 기법은 MIN Timer < 전송가능 시기 < RTCP RR Timer 에서 전송이 가능하다. 그림 7은 동적인 RTCP 구조 순서도이다.

IV. 모의실험 및 성능 분석

본 장에서는 단방향 전송 지연을 이용한 모니터링 기법에 대한 모의실험과 성능분석을 하고, 제한한 동적인 RTCP 기법을 적용한 네트워크 모니터링 기법에 대한 성능을 측정한다. 성능평가를 위해 LBNL(Lawrence Berkely National Laboratory)의 NS-2(Network Simulator)를 사용하였다^[10].

4.1 단방향 전송 지연을 이용한 모니터링 기법

단방향 전송 지연에 대한 모의실험을 수행하며, RTT를 이용한 모니터링 기법과 단방향 전송지연을 이용한 모니터링 기법의 네트워크 상태 판단 오차를 평가한다.

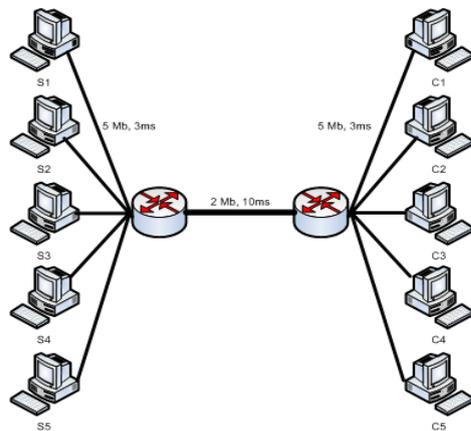


그림 8. 모의실험 환경

표 4. 단방향 전송 지연 모의실험 트래픽 특성

Server-Client	Protocol	Application	Transmission Rate
S1 - C1	RTP	CBR	300kbps
S2 - C2	TCP/Reno	FTP	Window size 15
S3 - C3	UDP	Traffic generator	100kbps
C4 - S4	TCP/Reno	FTP	Window size 15
C5 - S5	UDP	Traffic generator	100kbps

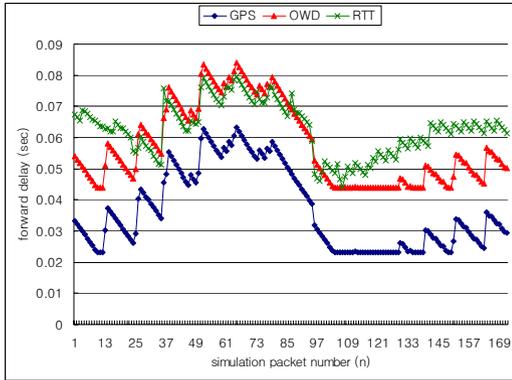


그림 9. 전송 지연 비교 분석

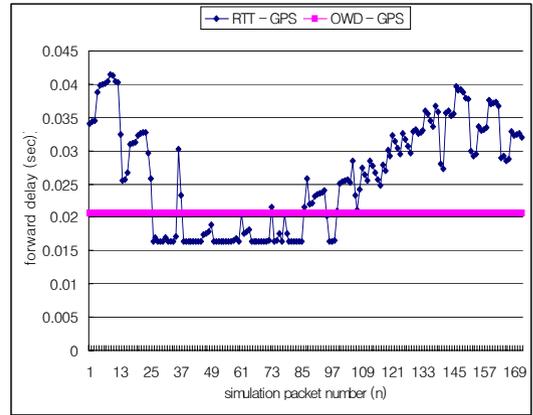


그림 10. 전송 지연

단방향 전송 모의실험은 그림 8과 같은 실험 환경을 구성하여 성능 실험을 수행하였다. 단방향 전송 지연을 이용한 모니터링 기법의 성능 실험을 위해서 경쟁하는 트래픽으로 TCP 트래픽과 UDP 트래픽을 사용하였으며 제안하는 단방향 전송지연을 측정하기 위해서 RTP 트래픽을 사용하였다. 생성되는 트래픽은 표 4와 같다. S1과 C1은 측정하려는 프로토콜이고 S2와 S3은 서버에서 클라이언트 쪽으로 트래픽을 발생시키는 에이전트이고 C4와 C5는 클라이언트 쪽에서 서버 쪽으로 트래픽을 발생시키는 에이전트이다. 혼잡을 발생시키기 위해 각각 다른 시간에 트래픽을 발생시키며 7초간 성능을 측정하였다.

그림 9는 GPS를 이용하였을 경우 전송 지연 값과 단방향 전송 지연과 RTT를 이용하였을 경우의 전송지연 값을 보여준다. 그림에서 알 수 있듯이 단방향 전송 지연 값과 GPS를 이용한 모니터링 기법은 동일한 패턴의 전송 지연 값을 갖는 것을 확인할 수 있으며 이는 그림 10을 통해 더욱 확실하게 알 수 있다. 그림 10은 ‘OWD - GPS’, ‘RTT - GPS’ 값을 보여주는 그래프로 단방향 전송 지연 값은 항상 일정한 상수 값만큼 큰 값을 갖는 것을 알 수 있다. 여기서 상수 값은 3.1절에서 설명한 β 값이다. 그림 9와 10을 통해 RTT 값은 클라이언트에서 서버로 전송되는 트래픽에 영향을 받아 전송 지연 값이 변화됨으로써 서버에서 클라이언트로 전송되는 스트리밍 서비스에서 사용하였을 경우 네트워크 상태를 판단함에 있어 오차가 발생 할 수 있음을 확인할 수 있다.

그림 11은 RTCP RR 패킷을 수신하였을 경우 서버의 AvgDelay 변화를 보여준다. 그림 9, 그림 10과 마찬가지로 OWD는 GPS와 동일한 패턴의 전송

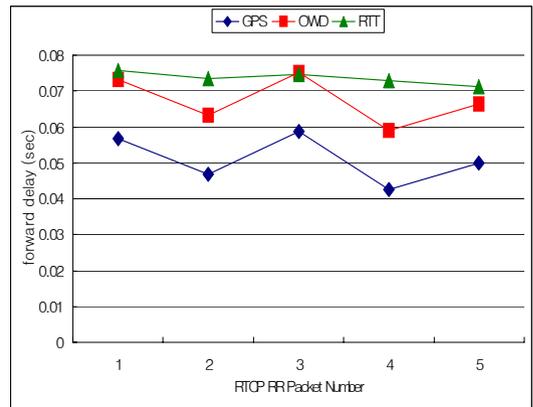
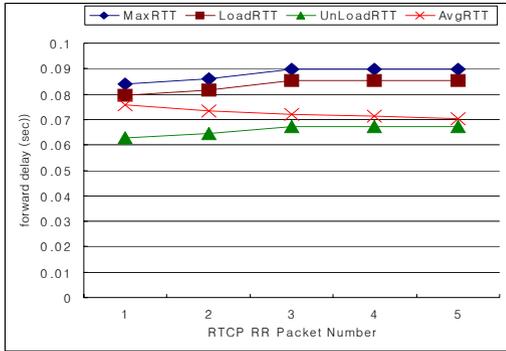


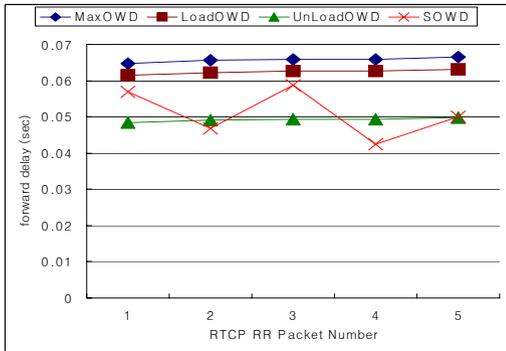
그림 11. RTCP RR 패킷에 따른 AvgDelay

지연 값을 가진 반면 RTT는 거의 동일한 AvgDelay 값을 가짐으로써 네트워크 상태를 판단함에 있어 오차가 발생 할 수 있다는 것을 확인할 수 있다.

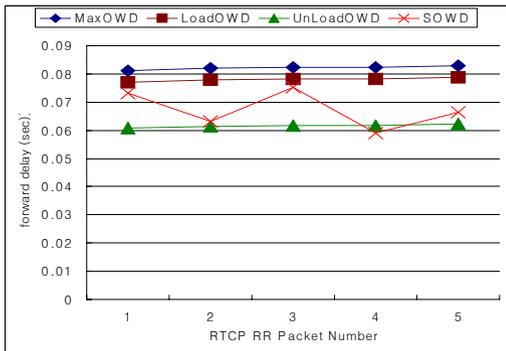
그림 12는 네트워크 상태 판단 알고리즘에 적용하였을 경우 네트워크 상태를 판단한 것이다. UpperRatio는 0.95 LowerRatio는 0.75이다. 각 (a),(b),(c)는 MaxRTT (MAXOWD)를 기준으로 Load값과 UnLoad값을 구하였으며 AvgRTT(SOWD)값이 어디에 위치하느냐에 따라 네트워크 상태가 결정된다. (a)는 네트워크 상태가 모두 LOADED 상태에 있으며 (b)는 LOADED에 3번 UNLOADED에 2번 있는 것을 알 수 있다. (c)는 제안한 알고리즘을 적용하였을 경우 네트워크 상태인데 4번의 LOADED와 1번의 UNLOADED 상태임을 알 수 있다. 제안된 단방향 전송 지연 값은 RTT를 이용한 네트워크 상태 판단 기법에 비해 20%정도 높은 정확도를 보였다. 그림 12의 (d)는 표 3의 α 값을 3으로 적용 하였을 경우 네트워크 상태 판단 그래프로 (c)에 비해 높은 정확도를 보인다.



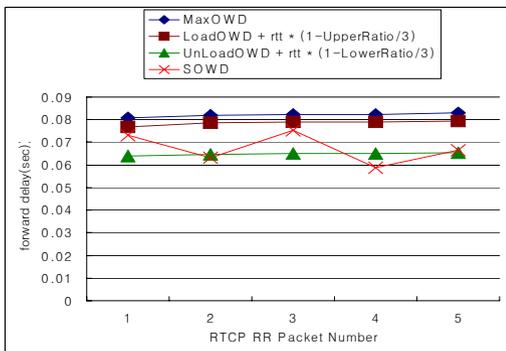
(a) RTT



(b) GPS



(c) OWD



(d) OWD + 오차보상

그림 12. 오차율과 네트워크 상태 판단

4.2 동적 RTCP 모의실험

네트워크 상황에 적응적으로 대처 할 수 있는 동적인 RTCP 기법에 대한 모의실험을 진행 한다. 동적인 RTCP와 RR 패킷 간격에 따른 패킷 손실률과 RTCP RR 패킷 개수에 따른 성능을 측정한다.

그림 8과 같은 환경을 사용하였으며, 트래픽 특성은 표 5와 같다. 네트워크 혼잡 상태를 발생시키기 위해 표 4보다 더 높은 트래픽을 발생 시켰다.

동적 RTCP의 성능을 측정하기 위해 TLFC 알고리즘을 사용하였다. TLFC 알고리즘에 적용하기 위한 파라미터 값은 표 6과 같다. LIGHTCONGESTION값에 LossRate를 적용시킴으로써 LIGHTCONGESTION 상황에서 전송률을 감소시키는 부분을 줄이고 시뮬레이션을 실행 하였다. 시뮬레이션은 총 100초 동안 실행하였으며 RTCP RR 패킷간격을 0.2초, 0.5초, 1초로 고정시켰을 경우와 제안하는 알고리즘인 동적 RTCP(MIN : 0.2~MAX : 1)로 하였을 경우, Period Control RTCP을 네트워크 혼잡 상태 일 때 0.2초 혼잡가능 상태일 때 0.3초 네트워크 부하 상태일 때 0.5초 네트워크 무부하 상태일 때 1초로 적용하였을 경우 패킷유실과 RTCP RR 패킷 개수를 측정하여 성능을 분석하였다^[11].

표 7은 시뮬레이션 결과를 표로 작성한 것이다. 제안한 동적 RTCP는 패킷 전송주기를 0.2초로 하였을 경우와 거의 유사한 패킷 유실률을 나타내고 있지만 전송횟수는 0.2초에 비해 40%정도 밖에 전송되지 않은 것을 확인 할 수 있다.

표 5. 동적인 RTCP 모의실험을 위한 트래픽 특성

Server-Client	Protocol	Application	Transmission Rate
S1 - C1	RTP	CBR	500kbps , 250kbps
S2 - C2	TCP/Reno	FTP	Window size 15
S3 - C3	UDP	Traffic generator	200kbps
C4 - S4	TCP/Reno	FTP	Window size 15
C5 - S5	UDP	Traffic generator	200kbps

표 6. TLFC 파라미터

Maximum Rate	500kbps	CongestionRate	2%
Minimum Rate	10kbps	LightCongestion Rate	1%
UpperRatio	0.95	LowerRatio	0.5

표 7. 시뮬레이션 결과

	0.2 Period	0.5 Period	1 Period	Period Control RTCP	0.2 - 1 Dynamic RTCP
Num of Loss	186	349	374	312	178
Num of RTP	12296	13834	12946	12508	11546
Loss Rate	1.51%	2.52%	2.89%	2.49%	1.54%
Num of RTCP	493	196	97	138	200

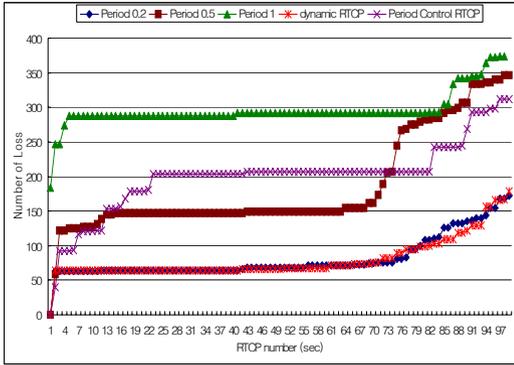


그림 13. RTP 패킷 유실

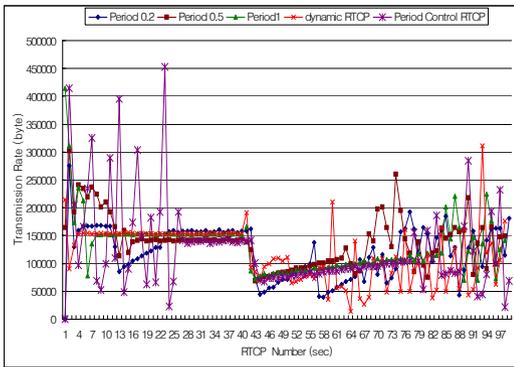


그림 14. 데이터 전송률

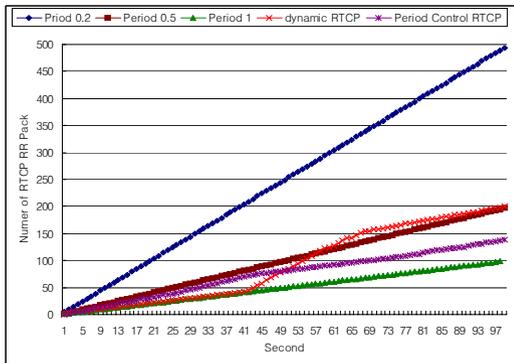


그림 15. RTCP 패킷 전송 횟수

그림 13을 통해 RTCP패킷에 따른 패킷 유실과의 관계를 볼 수 있다. 그림에서 알 수 있듯이 제안하는 동적 RTCP는 네트워크 상태가 혼잡할 경우 0.2초로 RTCP RR 패킷을 전송하는 유형과 비슷한 패턴의 패킷유실 상태를 확인할 수 있다. 그림 14는 데이터 전송률을 나타낸다. 시뮬레이션 환경에서 데이터 트래픽은 40초 부근부터 극심한 경쟁을 시작하며 네트워크 혼잡을 가중시킨 것이다.

그림 15는 초당 RTCP패킷 개수가 증가하는 양을 보여준다. 1~40초 구간은 네트워크 상태가 약간 혼잡한 상태로 유지 되는 구간이며 이때 RTCP RR 패킷 간격 1초일 때와 거의 동일한 값을 보여준다. 하지만 40초 이후 네트워크 상태가 혼잡해지면 네트워크 상태에 적응적으로 대처하기 위해 RTCP RR 패킷 개수가 증가하는 것을 확인할 수 있다.

V. 결과

본 논문에서는 IETF의 RTP/RTCP프로토콜을 사용하는 멀티미디어 스트리밍 서비스 환경에서 네트워크 상태판단의 오차율을 줄이기 위해 단방향 전송 시간을 이용한 네트워크 모니터링 기법과 네트워크 상황에 빠르게 대처하기 위한 동적인 RTCP기법을 제안했다.

단방향 전송을 이용한 모니터링 기법은 정확한 단방향 전송 시간을 구하는 것은 아니지만 송수신 컴퓨터 동기 오차를 고려하여 동일한 패킷을 구하고 이를 이용하여 실제 단방향 지연과 근사한 값을 구하는 기법이다. 모의실험에서 단방향 전송 지연을 이용한 모니터링 기법은 실제 단방향 전송 지연 값과 동일한 패킷을 구할 수 있고 네트워크 상태판단 오차를 줄인 효과를 보였다. 또한 단방향 전송 지연을 이용한 모니터링 기법은 오차율을 줄이기 위한 기법을 적용하였을 경우 GPS를 이용하여 단방향 전송 지연 값을 구한 것과 동일한 성능을 나타냈으며, RTT를 이용한 네트워크 상태판단 기법보다 우월한 성능을 보였다. 제안한 기법은 β^* ($1-UpperRatio$), β^* ($1-LowerRatio$) 만큼의 오차가 발생할 수 있으며 이는 처음 동기화시 RTT값을 이용하여 오차율을 줄이기 위한 기법에 적용하였을 경우 성능이 개선됨을 확인할 수 있었다.

참고 문헌

- [1] R. Braden, D.Clark, and S.Shenker, "Integrated Services in the Internet architecture : An overview," IETF, RFC 1633, June 1994.
- [2] K. Nichols, V. Jacobson, and L.Zhang, "A two-bit differentiated services architecture for the Internet," IETF, RFC 2638, July 1999.
- [3] S. Floyd, M. Handley, J. Padhye, and J. Widmer, "Equation-based congestion control for unicast applications," in Proc. ACM SIGCOMM,

Stockholm, Sweden, August 2000

- [4] R. Rejaie, M. Handley, and D. Estrin, "Rap: An end-to-end ratebased congestion control mechanism for realtime streams in the internet," Proc. IEEE Infocom, March 1999.
- [5] Seung-Gu Na and Jong-Suk Ahn, "TCP-like flow control algorithm for real-time applications" IEEE International Conference, September 2000.
- [6] C. Huitema, "Real Time Control Protocol (RTCP) attribute in Session Description Protocol (SDP)", IETF, RFC 3605, October 2003.
- [7] Audio-Video Transport Working Group, H. Schulzrinne, S. Casner, R. Frederick and V. Jacobson "RTP: A Transport Protocol for Real-Time Applications", IETF, RFC 1889, January 1996.
- [8] Jin-Hee Choi and Chuck Yoo, "Analytic End-to-End Estimation for the One-Way Delay and Its Variation.," IEEE Consumer Communications and Networking Conference 2005, January 2005.
- [9] G. Almes, S. Kalidindi and M. Zekauskas "A One-way Delay Metric for IPPM", IETF, RFC 2679, September 1999.
- [10] URL:<http://www.isi.edu/nsnam/ns/>
- [11] 나승구, 백갑천, 안중석, 김승범 "RTP 트래픽을 위한 효율적인 흐름제어 기법", 한국정보과학회 하계 컴퓨터통신 워크숍, 1998

김 용 술 (Yong-Sul Kim)

준회원



2004년 2월 광운대학교 정보통신공학과 졸업

2006년 2월 광운대학교 전자통신공학과 석사 졸업

<관심분야> 미들웨어환경, QoS, 스트리밍서비스

김 화 성 (Hwa-sung Kim)

정회원



1981년 2월 고려대학교 전자공학과 졸업

1983년 2월 고려대학교 전자공학과 석사

1996년 Lehigh Univ. 전산학 박사

1984년 3월~2000년 2월 ETRI

책임 연구원

2000년 3월~현재 광운대학교 전자통신공학과 교수

<관심분야> Wireless Internet, NGN 미들웨어환경, Streaming QoS,