
무선 네트워크에서 멀티미디어 서비스를 위한 흐름 제어

김동호* · 이용희* · 안세영**

Flow control for multimedia service in wireless networks

DongHo Kim* · Yong-Hee Lee* · SeYoung Ahn**

요 약

무선 인터넷이 기하급수적으로 증가함에 따라 최근 네트워킹과 멀티미디어 서비스에 대한 요구가 증대되어가고 있다. RTP는 인터넷 위에서 멀티미디어 통신을 지원하고 확장성과 유선환경에서의 적응력을 높였다. 그러나 RTP는 무선환경 안에서 양단간 QoS를 지원하지 못하는 제한을 갖는다. 본 논문은 실시간 멀티미디어 통신 아키텍처를 제안하고 하이브리드 흐름 제어를 설계하고 구현한다. 하이브리드 흐름 제어는 네트워크 상태나 사용자의 특성과 같은 메트릭을 이용한 개선된 AIMD를 기반으로 한다. 멀티캐스트 그룹 관리를 통해서 그룹 기반의 하이브리드 흐름제어를 한다. JMF를 이용하여 제안된 흐름 제어를 구현하여 성능을 분석하였다. 적용한 결과 제안된 흐름 제어가 AIMD보다 좋은 성능을 보여준다.

ABSTRACT

As the wireless internet grows exponentially, the recent trend has an increasing demand for wireless network and multimedia services. RTP is used to support the multimedia communication over the Internet and it supports the flexibility and adaptability over a wide range. However, RTP has a limitation that it cannot support end-to-end QoS guarantee in a wireless home network which has low throughput and high delay. In this paper, we propose the architecture of a real-time multimedia communication and design and implement the hybrid flow control in the architecture. The hybrid flow control mechanism is based on modified AIMD using metrics such as the network state information and the user properties. We implement the proposed flow control using JMF to evaluate the performance of the proposed flow control. The experimental results show that the proposed flow control has better performance than the AIMD.

키워드

wireless network, multimedia communication, flow control, AIMD, JMF

* 한라대학교 컴퓨터공학과

** 고려대학교 전자컴퓨터공학과

접수일자 2009. 01. 28

심사완료일자 2009. 03. 12

I. 서 론

최근 무선 인터넷이 발전함에 따라서 많은 사람들이 오디오 또는 비디오 회의, 인터넷 라디오, 온라인 세미나 또는 VOD, 홈 이나 특정 장소에서의 방송등 실시간 멀티미디어 어플리케이션의 이용 요구가 증대되었다. 홈 네트워크는 데이터 공유 기술과 멀티미디어 무선 통신 기술과 함께 발전해왔다. 그 어플리케이션들은 전송 데이터의 규칙적인 전송간격과 지연과 대역폭 보장이 필요하다.

RTP 프로토콜은 멀티 캐스트와 유니캐스트 망 서비스위에서 실시간 데이터 전송을 위한 어플리케이션에 적합한 양단간의 전송을 담당한다. RTP는 주소의 자원 예약과 실시간 서비스를 위한 QoS를 보장하지 않는다[1][2]. 데이터 전송은 대규모 멀티캐스트 망에서 범용되는 방법으로 데이터 전달의 모니터링을 할 수 있는 RTCP가 추가 된다. 이는 최소의 제어와 기능의 식별을 제공한다. 그러나 RTP/RTCP는 무선 환경에서의 어떠한 이점도 가지고 있지 않다[3]. 첫째, RTP는 유동서 있는 흐름 제어와 세션 대역폭 관리 기능을 가지고 있지 않다. 또 에러 컨트롤과 흐름 제어를 가지고 있지 않고 대역폭을 감시하는 기능이 없다. 두번째, RTCP는 대역폭 컨트롤을 할 수 없다. 초과된 제어 메시지는 망내에 대역폭을 낭비하고 처리능력을 저하시킨다. 셋째, RTP는 무선 환경의 특성을 고려하지 않았다[4][5].

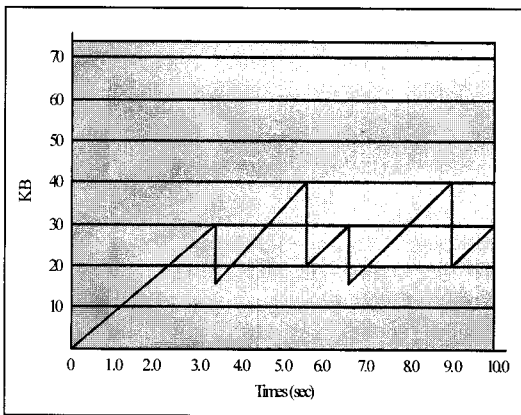


그림 1. AIMD 그래프
Fig. 1 AIMD graph

인터넷은 TCP를 이용하며 혼잡제어를 위해 AIMD (Additive Increase Multiplicative Decrease)를 이용한다. 기본적인 아이디어는 TCP 혼잡제어 뒤에 송신측이 손실 이벤트가 발생했을 때 송신비율을 줄이는 것이다. AIMD는 일반적으로 전송비율을 증가시키고 감소할 상황이 되면 전송률을 배로 감소시킨다. 따라서 AIMD가 작은 값으로 증가한다면 결과적으로는 일시적인 주기에서 낮은 흐름 이용률을 가지며, AIMD의 값이 감소하는 경우 급격하게 감소하여 큰 진동을 갖게 된다. [그림 1]은 이를 보여준다. 멀티캐스트는 한 송신자가 많은 수신자에게 데이터를 복사하여 전달하는 전송 방식으로 여러가지 방법으로 구현될 수 있다[5]. One-to-all 유니캐스트는 송신자가 각각의 수신자들에게 유니캐스트로 전송한다. 송신자는 수신자들의 수보다 작게 복사하여 어플리케이션 계층의 멀티 캐스트는 유니캐스트 전송하고 수신측은 자신들 다른 수신측에게 전달하기 위해 복사를 한다. 따라서 멀티 캐스트는 네트워크 계층에서 지원한다. UDP는 네트워크 계층의 라우터에게 복사되어 수신측에 도착할 때 다중화 링크에 의해 전달된다. IP계층은 멀티 캐스트서비스를 제공하기 위해 동작한다. Richard[6]는 개선된 AIMD를 제안하였고, Cen[7]은 흐름 제어 및 혼잡 제어를 제안하였다. 이들 연구는 흐름 제어로 수신자의 네트워크 상태에만 초점을 두고 있다.

문제점은 이질적인 장비나 사용자 환경 망에 제공되는 멀티미디어 통신 서비스에서 일치된 종단간의 QoS를 보장하는 것이 어렵다. 본 논문에서 망 상태와 장비 특성이 고려된 실시간 멀티미디어 서비스에 QoS를 지원하는 하이브리드 흐름 제어(Hybrid Flow Control) 메커니즘과 시스템 아키텍처를 연구한다. 하이브리드 흐름 제어 메커니즘은 망 상태 정보와 사용자 특성에 따라 매트릭스를 이용하는 수정된 AIMD이 기본으로 한다. 또한 홈 환경의 사용자들에 특별한 특성에 따라 그룹으로 분리하여 그룹에 근간을 둔 그룹 기반의 하이브리드 흐름 제어(Group-Based Hybrid Flow Control)를 제안하여 효율적인 멀티 미디어 통신을 실현하고자 한다.

본 논문의 구성은 다음과 같다. 먼저 2장에서 제안하는 시스템의 기본 아키텍처를 살펴보고, 3장에서 하이브리드 흐름 제어를 제안하고 그룹 기반의 하이브리드 흐름 제어를 제안한다. 그리고 4장에서는 JMF를 기반으로

흐름 제어를 구현하고 5장에서는 성능을 분석한 후 6장을 끝으로 결론을 맺고자 한다.

II. 실시간 멀티미디어 통신 구조

멀티미디어 통신 서비스가 무선환경에서 사용자의 QoS를 유지할 수 없을 때 장비의 상태에 따라 자원의 이용 상태나 미디어 데이터의 특성에 따라서 서비스를 재조직해야 한다. 무선망 안에서 QoS를 사용자에게 제공할 목적으로 실시간 멀티미디어 통신의 아키텍처를 제안한다.

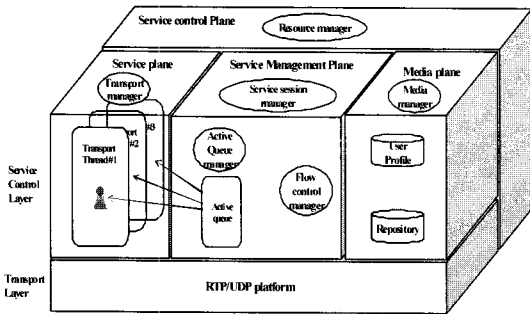


그림 2. 실시간 멀티미디어 통신 시스템 구조
Fig. 2 The Architecture of realtime multimedia communication

[그림 2]는 시간 멀티미디어 통신 시스템 구조를 보여준다. 시스템은 Service control plane, Service management plane, Media plane, Service plane으로 구성된다. 서비스 제어 부분(Service control plane)은 시스템 리소스 및 다른 부분을 제어하고 관리한다. 미디어 부분(Media plane)은 네트워크 상태, 사용자 프로파일, 그리고 장치 프로파일을 저장한다. 멀티미디어 데이터는 미디어 관리를 통해서 트랜스 코드 및 트랜스 레이어가 된다. 서비스 부분(Service plane)은 쓰레드 메커니즘을 이용해 사용자에게 서비스 전송을 제공한다. 쓰레드는 쓰레드 풀을 기반으로 전송 관리를 한다. 서비스 관리 부분(Service management plane)은 사용자 섹션을 유지하고 사용자 QoS를 제공한다. 이는 흐름 제어 메니저를 통해서 서비스 플로우를 제어 한다. 또 미디어

어 전송의 효율성을 위하여 능동적인 큐 관리를 갖는다. 그리고 미디어 데이터 저장소를 갖고, URL을 이용하여 사용자에게 제공한다.

[그림 3]은 사용자 프로파일이나 장치 프로파일 데이터 베이스의 정보 구조를 보여준다. 사용자 정보의 지연과 데이터 손실에 관한 정보는 송신단에 RTCP 레포트를 통하여 얻고 섹션을 통하여 사용자의 프로파일과 장치 프로파일을 얻는다. 가정은 SIP를 통하여 섹션을 맺는다.

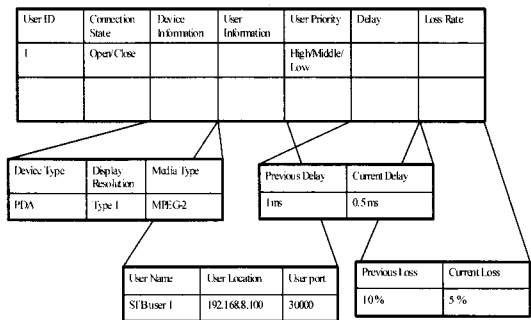


그림 3. 사용자 및 망 정보의 데이터 구조
Fig. 3. Data structure of user and network information

[그림 4]는 사용자에게 보다 좋은 서비스를 제공하고 효율적으로 멀티미디어 데이터를 처리하기 위해 사용되는 능동 큐를 보여준다. 능동큐는 관리자는 선점형 스케줄 정책을 따른다. 우선순위는 사용자 프로파일에 있는 사용자 우선순위와 미디어 타입을 이용하여 결정한다.

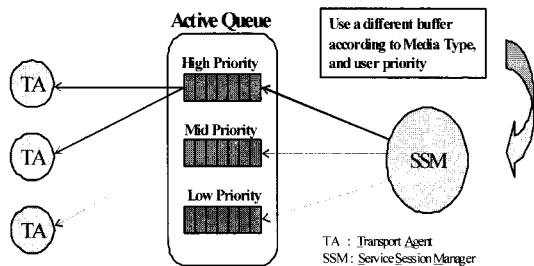


그림 4. 능동큐 관리 구조
Fig. 4. Management structure of Active Queue

RTP는 실시간 데이터를 전송하는데 이용하고 RTCP는 멀티미디어 데이터를 전송하기 위하여 네트워크 상태를 보고하고 모니터링하기 위해 사용한다.

III 흐름 제어

3.1 하이브리드 흐름 제어

무선 환경에 AIMD가 적용되었을 때 흐름률의 큰 진동을 갖는 문제로 멀티미디어 서비스 지원과 사용자의 서비스 질을 보장하는 것이 어렵다. 이 장에서 우리는 AIMD의 값을 줄이고 증가시키는 흐름 제어 메커니즘을 연구한다.

무선 홈 환경에서 다양한 장비의 특성과 멀티미디어 데이터를 전송한다는 것은 네트워크 대역폭 측면에서 효율성이 떨어진다. 하이브리드 흐름 제어(HFC)의 기술은 동적인 파라미터와 정적인 파라미터로 구분하여 사용자의 요구사항들과 네트워크 상태에 따라 전송률을 송신측에서 조절할 수 있는 적응형 방식이다. [그림 5]는 모든 노드들이 동일한 상태이고 여러 개의 송신단과 여러 개의 수신 단을 갖는다는 가정의 흐름 제어 시스템 모델이다.

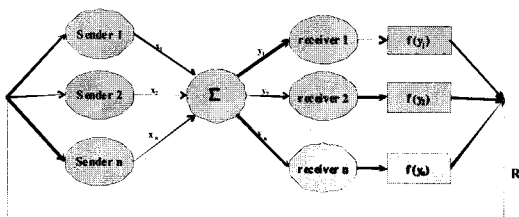


그림 5. 하이브리드 흐름 제어 시스템 모델
Fig. 5. Hybrid flow control model

$$X(t) = \sum x_i(t)$$

$$Y(t) = \sum (y_i(t) + e_i(t))$$

$$R(t) = \begin{cases} 1 & \text{if } F(y_i(t)) > 0 \\ 0 & \text{otherwise} \end{cases}$$

Flow Control Relation:

$$x_i(t+1) = \begin{cases} (\alpha aI + (1-\alpha) bI) x_i(t) & \text{if } (R(t)=1) \\ (\beta aD + (1-\beta) bD) x_i(t) & \text{otherwise} \end{cases} \quad (1)$$

Control Function:

$$aI = W_{min}, I(pstatic, pdynamic)$$

$$bI = W_{max}, I(pstatic, pdynamic)$$

$$aD = W_{min}, D(pstatic, pdynamic)$$

$$bD = W_{max}, D(pstatic, pdynamic)$$

$$\alpha = UI(pstatic, pdynamic) \quad \beta = UD(pstatic, pdynamic)$$

Boundary Condition:

$$1 \leq aI < bI < 2$$

$$0.5 \leq aD < bD < 1$$

방정식의 정의는 다음과 같다. X(t)는 소스로부터 총 트래픽을 보낸다. Y(t)는 목적지로부터 총 트래픽을 받는다. e(t)는 에러 요소이다. 무선 망의 인터페이스와 장비의 특성이 고려되었을 때 R(t)는 수신단들로부터 피드백이다. 만약 R(t)가 "1"이면 네트워크의 상태가 좋다는 것을 의미하고 흐름률이 증가한다. "0"이면 흐름률을 감소시킨다. a나 b 값은 증가율과 감소율의 경계를 의미하며 이는 제어 함수 W를 이용하여 계산된다.

```

//Receiver's cost method check the data loss and delay
//and then send the report to sender

struct rtp_packet{
    RTPHeader header;
    char * payload
}; rtp_data;

Func F (rtp_data){
    START;
    LOOP {
        if (Delay condition){ // delay_error
            Update the last timestamp p from rtp_data;
            make RR packet;
            send ERROR; //R(t)=0
        }
        else if (Loss condition){ // data_loss_error
            Update the last sequence number from rtp_data;
            make RR packet;
            send ERROR; //R(t)=0
        }
        else { //no problem
            make RR packet;
            send Success; //R(t)=1
        }
    }
    END;
}
    
```

그림 6. 수신단 보고 알고리즘
Fig. 6. Receiver's report algorithm

[그림 6]을 보면 제어함수 F는 수신단의 네트워크 상태를 RTP의 타임스탬프와 시퀀스 번호를 이용하여 보

고한다. 그리고 이것은 지연과 손실에 있어 경계값과 비교하여 송신측에게 이 상태(성공또는 에러)를 알린다. 따라서 네트워크 상태가 좋지 않다면 에러값을 반환 받고, 반대라면 성공값을 반환 받는다. 지연 경계값과 손실 경계값은 사용자의 필요에 따라서 손실 이벤트나 지연 이벤트를 통해서 유동적으로 결정된다. [그림 7]은 정적 파라미터와 동적 파라미터를 이용하여 흐름률의 경계를 결정하는 알고리즘이다.

```
//determine the parameter of the flow control
struct rtcp_packet{
    int delay;
    int data_loss;
    int status; //SUCCESS ERROR
} rtcp_data;
enum priority{low=1, middle=2, high=3};
struct static_parameter{
    int resolution; //high, middle, low
    int bandwidth; //high, middle, low
    int media_type; //high, middle, low
    int device_type;
} static_info;
struct dynamic_parameter{
    int delay;
    int data_loss;
} dynamic_info;
int min_inc, max_inc; //1<=min_inc<max_inc<2
int min_dec, max_dec; //0.5<=min_dec<max_dec<1

Func W(static_info, dynamic_info, rtcp_data){
    START;
    if(rtcp_data.status==SUCCESS){
        determine min_inc and max_inc
        from static_info and dynamic_info;
    }
    else if(rtcp_data.status==ERROR){
        determine min_dec and max_dec
        from static_info and dynamic_info;
    }
} END;
```

그림 7. 흐름률 경계 알고리즘
Fig. 7. Flow rate boundary algorithm

정적 파라미터는 해상도, 대역폭, 미디어 타입, 장비 타입을 갖고 세 가지 우선순위로 나뉜다. 동적 파라미터는 지연과 데이터 손실을 정의한다. 제어 함수 W는 RTCP 패킷의 상태의 조건 안에서 최대 흐름값과 최소 흐름값을 결정한다. 또한 제어 함수 U는 동적 파라미터와 상태 파라미터에 의존하여 계수값을 결정한다. α 는 흐름률 증가에 이용되는 계수이고 β 는 흐름률 감소에 이용되는 계수이다. 그리고 각각의 값은 [그림 8]처럼 RTCP 상태의 조건에 따라서 결정된다. 따라서 흐름률

결정을 위해 사용자의 정보뿐만 아니라 네트워크의 상태 역시 이용되어 AIMD 보다 더 작은 하이브리드 흐름률의 편향을 갖는다. 그래서 HFC는 사용자의 서비스 질을 쉽게 보장할 수 있다. 사용자가 실시간 멀티미디어 서비스를 받을 때 흐름률의 빠르게 감소하는 것은 정적인 정보와 사용자 서비스 질의 보장을 위해 피할 수 있다.

```
//determine the flow rates
int inc_ratio;
int dec_ratio;

FUNC U(static_info, dynamic_info, rtcp_data){
    START;
    if(rtcp_data.status==SUCCESS){
        calculate inc_ratio from static_info
        and dynamic_info;
    }
    else if(rtcp_data.status==ERROR){
        calculate dec_ratio from static_info
        and dynamic_info;
    }
} END;
```

그림 8. 흐름률 결정 알고리즘
Fig. 8. Algorithm for determining the flow rate

3.2. 그룹 기반 하이브리드 흐름 제어

무선 홈 환경은 많은 장비로 구성되며 그 중 전력 처리, 대역폭, 사용자의 서비스 등이 유사한 특성을 갖게 되는 장비도 있다. 효율적인 흐름 제어를 위해 그룹을 만들어 관리 하고자 한다. 보통 홈 환경에서 하나의 미디어 소스를 갖게 된다는 점에서 착안한 가정이다.

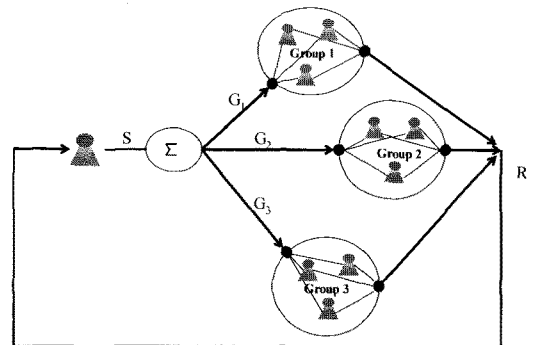


그림 9. 그룹 기반 하이브리드 흐름 제어 모델
Fig. 9. Group-based hybrid flow control model

[그림 9]는 HFC에 그룹 개념을 도입한 그룹 기반 하이브리드 흐름 제어(GHFC)의 시스템 모델을 보여준다. 가상 멀티캐스트 그룹은 각 수신단이 같은 멀티미디어 타입, 같은 데이터 그리고 같은 장비의 특성을 갖고 소스 노드가 중앙에 있게 된다. 따라서 같은 그룹에서 몇 개의 노드들은 같은 정적 파라미터를 포함한다. 소스 노드는 정적인 정보를 이용해 어떤 수신단 노드들을 멀티캐스트 그룹을 만든다. 정적 정보가 변한다면 그룹 역시 변하게 된다. 따라서 그룹 기반 흐름 제어를 위한 방정식은 다음과 같다.

$$S = \sum Gi(t)$$

$$Gi(t+1) = (\alpha aI + (1-\alpha)bI)Gi(t) \text{ if } (R(t)=1) \quad (2)$$

$$(\beta aD + (1-\beta)bD)Gi(t) \text{ otherwise}$$

G(t)는 멀티캐스트 그룹의 트래픽 능력을 의미하는 함수이다. 위 방정식 (2)는 방정식 (1)과 유사하고 같은 특성을 갖는다. GHFC에서 흐름 제어는 멀티캐스팅 그룹관리를 통하여 그룹을 유일하게 만든다. 비록 네트워크 계층에 각 멀티미디어 서버는 유일한 주소로 일반적인 멀티캐스트는 전체적인 멀티 캐스트 주소를 서버에게 할당하는 것으로 결정되지만 그룹관리와 그룹주소는 섹션 계층에서 얻을 수 있다. 섹션 계층에서 그룹주소를 지역적으로 할당하고 송신측 노드가 관리될 때 각 노드에서 송신측 노드의 정보가 정적으로 설정된다. 새로운 그룹의 멀티캐스트 능력만큼이나 멀티캐스팅의 그룹의 멤버의 업데이트는 멀티캐스트 섹션 유지에 필요하다. [그림 10] 처럼 멀티캐스트 그룹의 처리 및 생성을 보여준다.

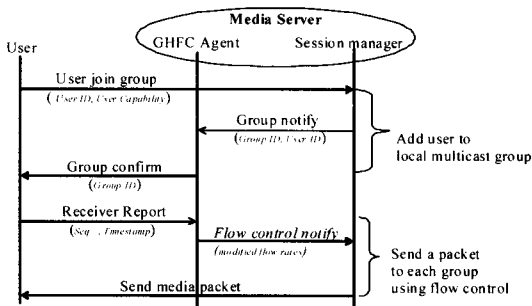


그림 10. 멀티캐스트 그룹 가입 절차
Fig. 10. Multicast group join procedure

사용자는 섹션 관리를 위해 멀티캐스트 그룹에 가입한다. 섹션은 사용자 데이터안에 사용자 정보의 저장하고 사용자 능력을 기반으로 어떤 송신측 노드들을 그룹 관리한다. 흐름 제어를 위해 흐름제어 관리는 그룹 상태를 확인하고 사용자를 확인하고 그것을 그룹에 전송한다. 그리고 사용자 상태는 흐름 제어 관리에 보고되고 이는 섹션 매니저에 흐름을 수정을 한다. 이런 식으로 미디어 전송을 제어한다. 이 논문에서 흐름률은 미디어 매니저안에서 미디어 전환을 제어 하고, 전송 매니저에서 전송률을 전환한다. 비록 GHFC는 그룹 등록 처리를 하기 때문에 등록에 대한 트래픽과 보안 및 서비스 정책에 대한 결점이 있을 수 있지만 가상 멀티캐스팅을 통하여 트래픽의 흐름을 조절을 통하여 그룹 QoS에 따라 멀티미디어 서비스를 한다. 따라서 GHFC는 홈 네트워크 장비의 특성에 따라 흐름제어 효율적으로 관리하고 사용자에게 보다 좋은 서비스를 제공하고 소스 노드의 성능이 향상되었다.

IV. JMF를 이용한 흐름제어 구현

4.1. 서버 구조

[그림 11]은 서버 구조를 보여준다. 서버의 구조는 크게 Session Acceptor와 Flow control Manager 와 Media Server 그리고 data base로 구성된다.

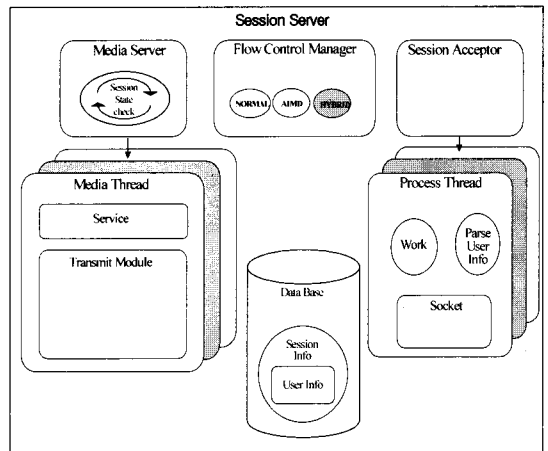


그림 11. 서버 구조
Fig. 11. The server structure

Session Acceptor는 다중 클라이언트의 요청을 받아 각각의 Process Thread를 생성한다. 이는 java Server socket를 이용하여 구현하였고 간단한 구조의 섹션 프로토콜을 정의한다. 다중 클라이언트와 주고 받은 데이터를 data base에 저장한다. Flow Control manager는 본 논문에서 제안한 하이브리드 흐름 제어, AIMD 알고리즘을 구현하였다. Data base에 control 정보를 보고 다중 클라이언트의 Media Thread를 제어한다. Media Server는 멀티미디어를 전송하는 상태를 확인하면 Media Thread를 생성한다. 이는 JMF(Java Media Framework)[8]를 이용하여 구현하였다. Media Thread는 data base를 보고 각각의 클라이언트가 선택한 멀티미디어 데이터를 선택하여 이를 RTP를 이용하여 전송할 수 있는 단위로 처리하여 각각의 클라이언트에게 데이터를 전송하게 하는 전송 모듈(Transmit module) 부분이 있다.

[그림 12]는 전송 모듈 부분을 보여준다. 전송 모듈은 멀티미디어 데이터를 RTP가 전송가능한 포맷으로 만드는 processor부분과 전송을 담당하는 RTP Manager부분으로 나뉜다. Processor에서 멀티미디어 데이터의 bit rate를 제어할 수 있도록 com.sun.media.controls.Bitrate Adapter와 javax.media control.BitRateControl의 JMF 라이브러리인 setChangnedBitRate 메소드를 이용하여 제어하였다. mpg 파일을 입력 데이터 소스로부터 프로세서 읽어와 파싱부분에 지연을 주도록 코드를 수정하여 setChangnedBitRate에 인수범위는 0.5부터 1까지 범위를 넘겨주어 원하는 제어를 할수 있다. 1은 기본bit rate이며 0.5로 내려갈수록 버퍼에 있는 미디어 데이터를 빠르게 전송할 수 있도록 한 것이다.

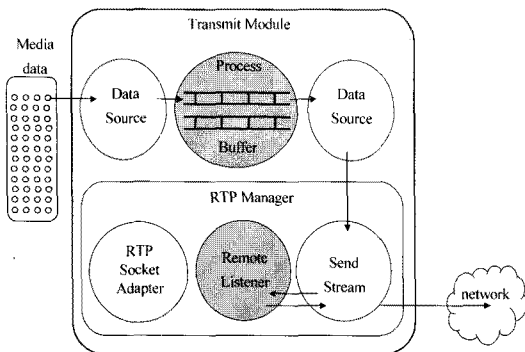


그림 12. 서버의 전송 모듈
Fig. 12. Transmit module of server

RTP Manager에서 Remote Listener를 통하여 RTCP리시버 레포트(RR)와 센터 레포트(SR)를 확인한다. 또 이를 log 파일로 남겨 성능을 측정하는 데이터를 얻는다. RTP Socket Adapter는 멀티미디어 데이터를 전송하기 RTP로 전송하기 위하여 지원하는 부분이며 Send Stream을 통하여 멀티미디어 데이터를 통하여 클라이언트에 전송된다.

4.2. 클라이언트 구조

[그림 13]은 클라이언트 구조를 보여준다.

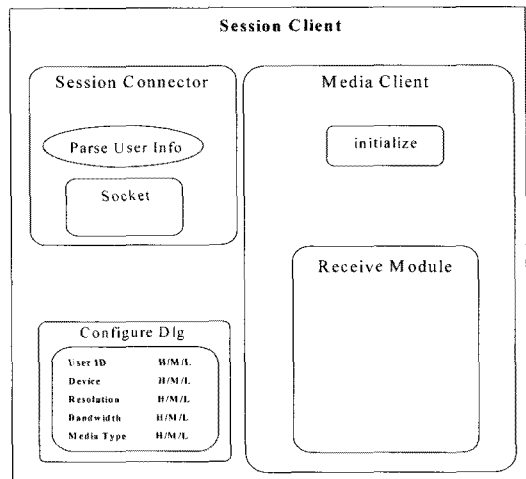


그림 13. 클라이언트 구조
Fig. 13. The client structure

클라이언트 구조는 크게 Session connector와 Media client 로 구성된다. 사용자 인터페이스에서 사용자의 정적인 정보는 UserID, Device, Resolution, Bandwidth, Media type을 셋팅 할 수 있으며 각각의 등급을 High, middle, low로 나누어 셋팅한다. Session connector는 다음에 설명한 간단한 섹션 연결 프로토콜을 이용하여 서버에게 섹션 연결을 요청한다. TCP 위에서 동작하며 사용자의 정적 정보를 서버에 보내고 서버로부터 오는 메시지를 파싱하여 채널 정보를 얻는다. Media Client는 서버로부터 오는 멀티미디어 데이터를 받는 Receive Module이 있으며 이는 JMF를 이용하여 구현하였다. [그림 14]는 클라이언트에서 수신 모듈(Receive Module) 부분을 보여준다. 수신 모듈은 크게 RTP Manager와 Player 부분이 있으며 서버로부터 모든 데이터는 Receive Stream을

통하여 들어오고 이 중 RTCP 정보는 Remote Listener에 의해서 선별된다. 또 RTCP의 리시버 보고(RR) 메시지는 Send Stream을 통하여 전송한다. 동적인 사용자 정보를 이 RTCP 정보를 통하여 얻어 서버의 database에 저장하고, 이를 AIMD와 HFC 알고리즘이 파라미터를 이용하고, log 파일로 남겨서 성능을 측정한다.

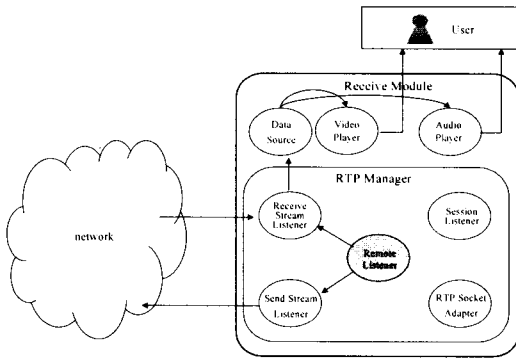


그림 14. 클라이언트에서 수신 모듈
Fig. 14. Receive module in client

V 성능 측정 및 결과 분석

5.1. 시뮬레이션 환경

[그림 15]는 실험환경을 보여준다. 실험환경은 1대의 서버와 3대의 무선 클라이언트와 3대의 유선 클라이언트로 되어 있다.

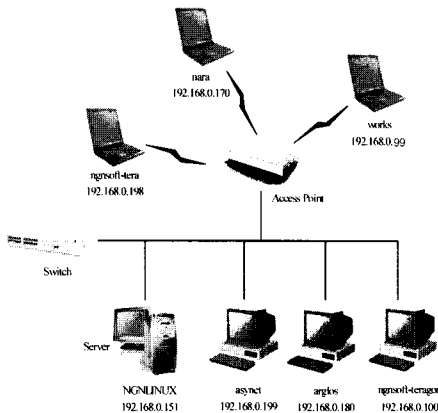


그림 15. 실험 환경
Fig. 15. The test bed environment

5.2. 시뮬레이션 시나리오

1대의 서버가 6대의 유무선 클라이언트에 같은 멀티미디어 데이터를 전송할 때 AIMD, HFC의 알고리즘에 대한 loss, jitter, delay를 기록하고 이를 측정한다. 기록 방법은 JMF method중 getDLSR(returns the delay since last SR: DLSR), getJitter(returns the interarrival jitter), getNumLost(returns the cumulativenumber of packets lost)를 이용한다. [표 1]은 정적인 정보와 동적인 정보의 셋팅 값을 보여준다. 또 이 실험은 30분 동안 mpg 파일에 대상으로 실험한다.

표 1. 시뮬레이션에서 정적 정보와 동적 정보
Table 1. Static information and dynamic information in the simulation

condition	contents
Static information	Device : Middle(2) Resolution : Middle(2) Bandwidth : Middle(2) Media Type : Middle(2)
Dynamic information	Coefficient : 50 DelayTH : 300000 LossTH : 10 JitterTH : 3000
Etc	Service time :30 minutes Only mpg file

5.3. 성능 측정

그림들은 비디오 스트림에 대한 AIMD, HFC 각각의 알고리즘에 대한 loss, delay, jitter에 대한 그래프를 보여준다.

[그림 16] 및 [그림 17]은 AIMD 및 HFC에서 패킷 손실의 누적 계수를 보여 주고 있다.

AIMD에서 최대 손실 누적계수는 무선 호스트 99(IPv4 주소가 192.168.0.99)에서 4600 packet 누적 손실을 보였으며, HFC에서는 무선 호스트 170에서 3490 packet의 최대 손실 누적계수를 보였다. 최소의 패킷 손실 누적계수는 AIMD에서는 무선 호스트 170에서 3100 packet, HFC에서는 유선 호스트 199에서 2990 packet을 보여준다. 최대 패킷 누적을 및 최소의 패킷 누적이 AIMD에서는 무선 호스트에서 발생하여 AIMD가 HFC와 비교하여 무선연결은 환경에 변화에 예민하다는 것을 알 수 있다.

또 AIMD에 비해 HFC는 최대 손실 누적계수에서 1110 packet 만큼, 최소 손실 누적계수에서 110 packet 만큼의 성능 우위를 보여 주고 있다.

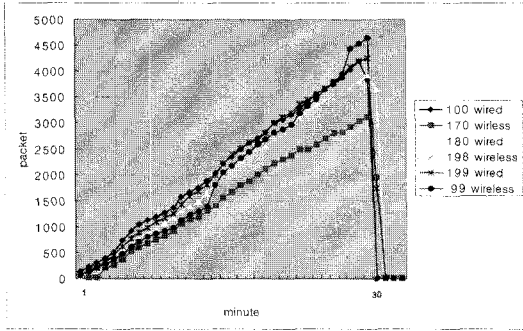


그림 16 AIMD에서 패킷 손실의 누적 계수
Fig. 16. The cumulative number of packets lost on AIMD mode

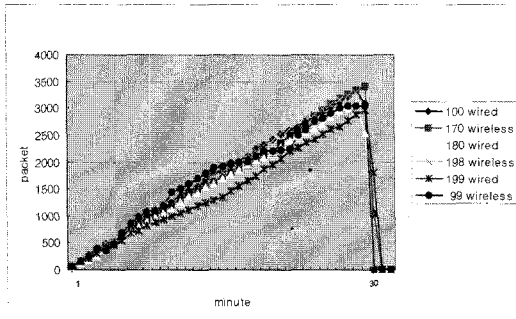


그림 17. HFC에서 패킷 손실의 누적 계수
Fig. 17. The cumulative number of packets lost on HFC mode

[그림 18]과 [그림 19]는 AIMD와 HFC의 지연을 보여 준다. AIMD에서 최대 지연은 유선 호스트 199에서 257000 msec을 보였고 최소지연은 무선 호스트 198에서 80000 msec을 보였다. HFC에서는 최대지연은 무선 호스트 170에서 250000 msec 무선 호스트 99와 유선호스트 198에서 100000 msec 보여 주었다. 유선과 무선의 대한 최대지연은 7000 msec으로 HFC가 우수하고 최소지연은 20000 msec으로 AIMD가 우수하였다. 하지만 실험의 환경에 따라 AIMD는 최대, 최소의 지연의 폭이 크고 각 호스트들의 전체적인 지연율은 HFC가 우수한 성능을 보여 주고 있다.

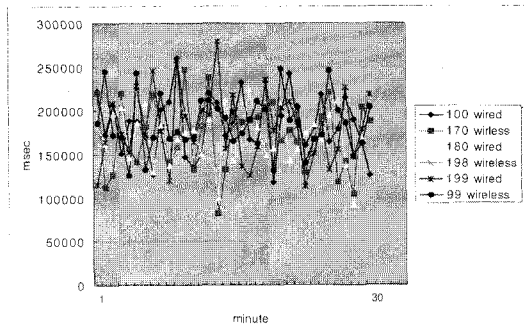


그림 18. AIMD에서 지연
Fig. 18. The delay on AIMD mode

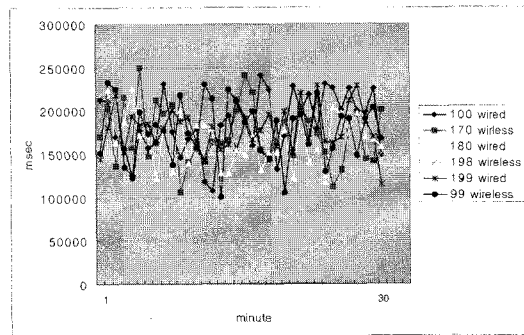


그림 19. HFC에서 지연
Fig. 19. The delay on HFC mode

[그림 20]과 [그림 21]는 AIMD와 HFC의 지터를 보여 준다. 지터 실험은 지연과 유사한 요소로 멀티미디어 서비스의 척도가 될수 있는 실험이다.

지터는 전후 패킷 사이의 수신 시간 차로 AIMD에서 최대 지터는 유선 호스트 170에서 2098 msec, 최소 지터는 무선 호스트 180에서 1190 msec이다. HFC에서 최대 지터는 유선 호스트 180에서 2120 msec, 최소 지터는 유선 호스트 100에서 1080 msec이다. 최대 지터차이는 22msec로 AIMD가 우수하고 과 최소 지터 차이는 110 msec는 HFC가 우위를 보였다. 지터 실험의 전체적인 성능은 근소하기는 하나 HFC가 약간의 우위를 보여 주고 있다. 만약 멀티미디어 서비스를 한다면 기록이 심한 지연을 갖는 패킷보다는 다소 지연이 있을 지라도 기록 작은 지연을 갖는 트래픽을 갖는 땅이 유리하다.

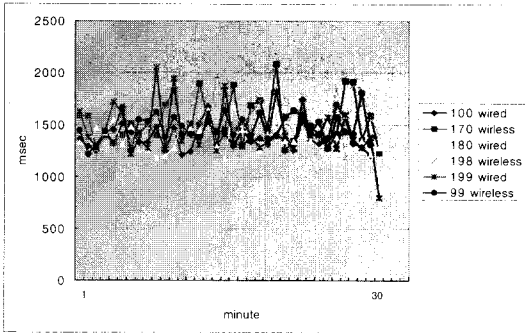


그림 20. AIMD에서 지터
Fig. 20. The jitter on AIMD mode

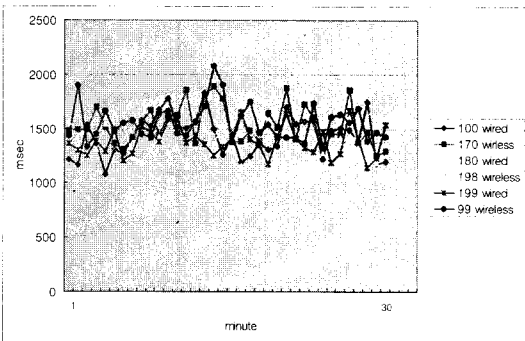


그림 21. HFC에서 지터
Fig. 21. The jitter on HFC mode

5.4. 실험 결과 분석

HFC는 loss부분에 대해서 개선되었음을 알 수 있고 delay, jitter에 대해서도 성능 개선은 있었다.

그래프가 delay와 jitter에 대해서는 유사한 성능을 보여 주는데 실험을 할 때 내부 망을 만들고 임의의 불규칙한 트래픽을 유발하면 HFC가 AIMD보다 좋은 성능을 보여주었다. HFC는 사용자의 정보 만큼이나 네트워크의 상태 역시 이용하여 AIMD 보다 더 작은 하이브리드 흐름들의 편향을 갖는다. 그래서 HFC는 사용자의 서비스 질을 쉽게 보장할 수 있다. 하지만 실험에 있어 임의의 트래픽을 배제하여 실험하였다. 실제 인터넷상에 멀티미디어 데이터를 전송할 때는 성능이 개선될 것이다. 즉 HFC는 사용자들에게 QoS를 만족시킬 수 있는 알고리즘이라는 결과를 실험을 통해서 알 수 있다.

VII 결론 및 향후 계획

무선 네트워크안에서 멀티미디어 서비스를 제공을 위해 많은 연구가 진행되었다. 무선 네트워크 특성을 고려하여야 하고 무선 기술이 멀티미디어 통신을 지원하는데 높은 지연율, 낮은 처리율, 낮은 QoS등 문제점들있다. 이 논문은 사용자 서비스 질의 만족할 수 있는 HFC를 제안했다. HFC는 사용자 특성 만큼이나 네트워크 상태에 기본이 된다. 그리고 트래픽 흐름 제어를 위해 AIMD 매커니즘을 수정하여 구현하였다. 또한 무선 홈 환경에서 많은 홈 기기들의 장비는 유사한 처리능력과, 디스플레이 그리고 대역폭을 갖는다. 이런 특성을 이용하여 소스노드는 멀티캐스트그룹의 단위안에서 흐름들을 제어하는 단 시스템의 멀티캐스트의 형태가 되는 GHFC를 본 논문에서 제안하였다. 하이브리드 흐름 제어는 무선홈 환경에서 홈 방송, 온라인 게임, 비디오/오디오 서비스, 회의 서비스 등에 적용가능하며 이기종 무선망에서도 확장가능하다. 향후 계획은 멀티 캐스트 그룹주소의 관리 연구와 멀티캐스팅 그룹으로부터 섹션 프로토콜의 기반인 SIP연구를 이어가고 하이브리드 흐름 제어를 이용한 무선 멀티 시스템을 구현하는 것이다. 또 이기종 무선 망 안에서 멀티캐스팅 적용 방안에 대해서 연구할 계획이다.

참고문헌

- [1] RTP: A Transport Protocol for Real-Time Applications (RFC 1889)
- [2] RTP Profile for Audio and Video Conferences with Minimal Control (RFC 1890)
- [3] Zhang Zhanjun, Han Chengde, A RTP-based Architecture of Multimedia Communications for Wireless networks, info-tech and info-net, 20001, proceedings, ICII 2001- Beijing, 2001 International conference, IEEE, 2001, p345-350 vol.5
- [4] Alan B.Johnston, Understanding the session Initiation Protocol, 2000.
- [5] Dharma Prakash Agrawal, Qing-An Zeng, Introduction to wireless and Mobile Systems, Brooks/cole, 2003

- [6] Yang Richard Yang and Simon S. Lam. General AIMD Congestion Control, Proceedings of ICNP 2000, Nov. 2000. Page(s):187 - 198
- [7] Shawnwei Cen, Calto Pu, and Jonathan Walpole. Flow and congestion control for Internet streaming applications. Proceeding of Multimedia Computing and Networking 1998, January 1998, Vol. 3310, p. 250-264
- [8] JMF <http://java.sun.com/http://java.sun.com/javase/technologies/desktop/media/jmf/2.1.1/guide/index.html>

저자소개



김동호(DongHo Kim)

1986년 2월 고려대학교 전자공학과
(공학사)

1996년 2월 고려대학교 전자공학과
(공학석사)

2000년 8월 고려대학교 전자공학과(공학박사)
2001년 3월~현재 한라대학교 컴퓨터공학과 조교수
※ 관심분야: 정보보호, 컴퓨터네트워크, RFID



이용희(Yong-Hee Lee)

1991년 2월 한양대학교 전자공학과
(공학사)

1993년 2월 한양대학교 전자공학과
(공학석사)

1998년 2월 한양대학교 전자공학과(공학박사)
1999년 3월~현재 한라대학교 컴퓨터공학과 교수
※ 관심분야: 임베디드시스템, 생체신호처리



안세영(SeYoung Ahn)

1999년 2월 고려대학교 자연과학
대학 정보공학과 (공학사)

2004년 2월 고려대학교 공과대학원
통신시스템기술 협동학과
(공학석사)

2007년 2월 고려대학교 공과대학원 전자컴퓨터공학과
박사 수료
※ 관심분야: 컴퓨터네트워크, 센서네트워크