

효과적인 예약 오류 최소화를 위한 RSVP 알고리즘

김 범 석, 김 정 수, 신 봉 식, 정 정 화
한양대학교 전자통신 전파공학과 CAD 및 통신회로 연구실
E-mail : tommmy@netsgo.com

A RSVP Algorithm with Efficient Reservation Error Minimization

Beum-Seok Kim, Jung-Su Kim, Bong-Sik Sihn,
Jong-Wha Chong
CAD & C.C. Lab., Dept. of Electronic Eng., Hanyang Univ.
17 Haengdang-Dong, Seongdong-Gu, Seoul, Korea

요약

본 논문에서는 SM 메시지(Search Message)를 이용하여 효과적인 예약 오류 최소화를 위한 새로운 RSVP(Resource Reservation Protocol) 알고리즘을 제안한다. 기존의 RFC-2205[1]에 제시된 방안은 각각의 예약 요청을 모두 유지해야하기 때문에 과도한 대역폭 낭비와 이로인한 예약 성공 횟수가 감소되는 문제점을 안고 있다.

대역폭 낭비를 줄이기위해 본 논문에서는 SM 메시지를 도입하였다. 수신자는 주기적으로 SM 메시지를 송신자로 전송하여, 전송로 상의 각 라우터의 사용가능한 대역폭을 검사하여 최소 대역폭으로 각 라우터에 블록케이드 값을 생성한다. 각 라우터에서의 병합작업시 SM 메시지를 이용하여 설정된 블록케이드 값을 기준으로 병합작업 여부를 결정한다. 이는 각각의 예약요청을 별도로 유지해야 할 필요가 없어지므로, 기존의 KRP-I 문제의 해결책인 RFC-2205보다 대역폭 낭비를 줄일수 있게되며 이로인해 예약 수락율이 증가하여 보다 많은 예약 성공 횟수를 얻을수 있게된다.

제안한 SM 알고리즘을 모의 실험 해 본 결과, 예약 수락율이 기존의 RFC-2205보다 전체적으로 약 26% 정도 향상되었다.

1. 서론

IP(Internet Protocol)를 기반으로 하는 인터넷은 트래픽의 폭발적인 증가로 인해 전송속도의 저하와 확장성의 한계에 직면하고 있다. 특히, 멀티미디어 통신 시대의 도래에 따라 전송지연에 민감하고 많은 대역폭을 요구하는 화상회의, 원격각의, 주문형 비디오와 같은 응용들이 개발됨에 따라 기존의 인터넷 프로토콜인 최선형 서비스(Best effort Service) 방식만으로는 이들의 요구를 충족할 수 없는 것이 현실이다.

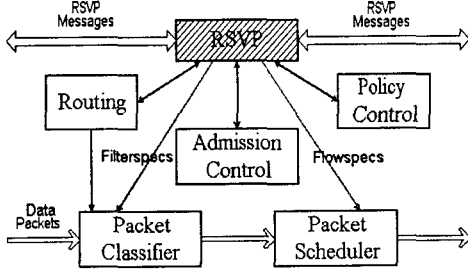
현재 IETF(Internet Engineering Task Force)와 RSVP 포럼 등을 중심으로 다양한 전송기술과 QoS(Quality of Service) 기술이 제안되고 있다. 그 중 하나인 RSVP는 송신자-라우터-수신자간에 자원예약(Resource Reservation)을 하여, 실시간 서비스를 지원하는 자원 예약 프로토콜이다[1]. RSVP 방식은 다수의 수신 측으로부터 오는 자원 예약들을 멀티캐스트(Multicast) 트리 경로상의 분기점인 라우터에서 병합하여 임시상태(Soft State)를 줄이는 방식을

채택함으로써 인해 예약해제 오류(Killer Reservation Problem)를 야기 시키는 단점을 가지고 있다. 기존의 KRP-I의 해결책[1]은 예약된 예약요청을 나중에 도착한 예약요청에 에러가 발생하더라도 먼저 요청을 그대로 남겨두는 방법이다. 그러나, 이 방식은 과도한 대역폭 낭비를 초래한다는 문제점을 가지고 있다.

본 연구에서는 이러한 RSVP의 단점을 보완하기 위하여 전체검색(Full Search)을 하여 수신단과 송신단 간에 탐색 메시지를 보내 라우터의 용량을 검색하고 차단상태를 생성하는 탐색 메시지(Search Message: SM) 알고리즘을 제안하였다. 논문의 구성은 다음과 같다. 2장에서는 RSVP 모듈에서의 메시지 처리에 대해 요약하고, 3장에서는 예약해제와 에러발생에 대해 다루고, 4장에서는 차단 상태(Blockade State) 메시지의 처리에 대해서, 5장에서는 검색 메시지(Search Message:SM) 알고리즘을 제안하며, 6장과 7장에서는 가상망을 통한 모의실험 결과를 제시하며 결론을 내린다.

2. RSVP에서의 메시지 처리

RSVP는 수신자 주소, 트랜스 포트 계층의 프로토콜 타입, 그리고 수신자의 포트 번호로 통신 세션을 인식한다. 각 RSVP의 동작은 특정 세션의 패킷에 대해서만 적용된다. 따라서 각 RSVP 메시지는 세션의 세부 사항에 관한 정보가 있어야 한다[2]. 다음 그림은 RSVP에서의 메시지 처리모습을 나타낸다.



[그림 1] RSVP에서의 메시지 처리

[그림 1]에서 수신 측 호스트의 응용 프로그램이 요구하는 서비스 품질 제어 요청은 일단 호스트의 RSVP 데몬 프로세스로 전달된다. RSVP는 이 요청을 데이터가 전달되는 역 방향 경로의 모든 노드(호스트, 라우터)에게 전달한다. 서비스 품질 제어를 할 수 있는 각 라우터는 입력 패킷을 패킷 분류자가 받은 후 경로와 서비스 품질 클래스(Packet Classifier)를 결정한다. 노드의 출력에서는 패킷 스케줄러(Packet Scheduler)에 의해서 예정된 서비스 품질을 얻을 수 있도록 패킷의 전달을 결정한다. 각 노드에서 RSVP 제어 요청은 '수락 제어부(Admission Control)'와 '정책 제어부(Policy Control)'의 두 부분에서 전달된다. 수락 제어부는 해당 라우터가 요청된 서비스 품질을 만족할 만한 충분한 자원이 있을 경우, 이 요청을 수락한다. 만약 충분한 자원이 없으면 요청을 거절하고, 이를 수신자에게 알린다. 정책 제어부는 예약을 요청하는 사용자가 충분한 자격을 가지고 있는지를 결정한다. 만약 이 두 검사가 성공하면, 패킷 분류자와 패킷 스케줄러의 파라미터 값들을 수정해서 원하는 서비스 품질을 얻을 수 있게 된다. 반면, 두 검사가 실패할 경우 예약 실패가 발생했다는 것을 수신 측에 알린다[2].

RSVP의 가장 큰 특징은 송신자가 아닌 수신자가 QoS를 명시한다는 점이다. 다음의 예를 생각해 보면 송신자가 3Mbps밖에 처리할 수 없는 CPU를 가진 수신자에 6Mbps의 비디오 스트림을 보낼 필요는 없는 것이다. 송신자와 수신자간에 전송속도를 미리 정의하는 방안도 고려할 수 있으나, 이 방안은 멀티캐스트 통신의 경우 각각의 수신자와 전송속도를 정의한다는 것은 비효율적이다.

RSVP의 예약 메커니즘을 간략히 살펴보면 송신자는 주기적으로 Path 메시지를 보내면 수신자는 QoS 요구를 명시한 Reserve 메시지를 송신자로 보

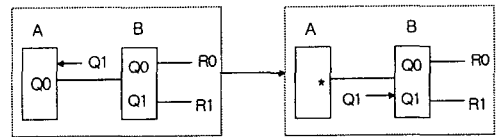
냄으로써 응답한다. 그러면 네트워크는 역 방향을 따라 대역폭을 예약해 나간다[3]. 이와 같은 RSVP 서비스는 IP상에서 QoS를 보장하여 실시간 서비스를 이용하는 경우 안정적인 전송을 가능하게 한다.

3. 예약 해제와 에러 발생

RSVP가 갖는 가장 큰 문제점은 예약해제 오류이다. 이 문제의 언급을 위해 RSVP에서 사용되는 대표적인 메시지를 살펴보면 다음과 같다.

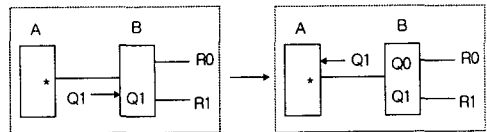
1. 제거 메시지(ResvTear): RSVP의 예약을 해제할 때 사용된다.
2. 경로 에러 메시지(PathTear): 경로 메시지로 인한 에러가 발생되었다는 것을 알리는 메시지로써, 경로 메시지를 보낸 송신자에게 전달된다.
3. 예약 에러 메시지(ResvErr): 예약을 요구한 수신자에게 예약이 설정되지 못했다는 사실을 알린다.
4. 모든 메시지는 홉(Hop)단위로 전달된다.

KRP는 라우터가 처리할 수 없는 큰 대역폭 요청과 처리 가능한 작은 대역폭 요청이 병합되어 작은 대역폭 요청이 희생되는 문제로서 작은 대역폭 요청의 경우 실패하게 된다. KRP는 KRP-I, KRP-II 두 가지가 있는데, 다음 그림은 KRP-I을 나타낸다.



[그림 2] 예약해제 오류 - I

[그림 2]에서 볼 수 있듯이 이미 수신자(R0)의 예약요청(Q0)의 예약이 되어 있는 상태에서, 다른 수신자가 Q0보다 큰 Q1만큼의 예약을 신청하였다. 라우터B는 두 개의 자원요청(Resource Request)을 병합하여 라우터A로 보낸다. 이런 경우, Q1의 예약이 상향 라우터에 의해 거절되면, 이미 설정되어 있던 Q0의 예약까지도 해제되는데 이것을 KRP-I이라 부른다. 다음 그림은 KRP-II을 나타낸다.



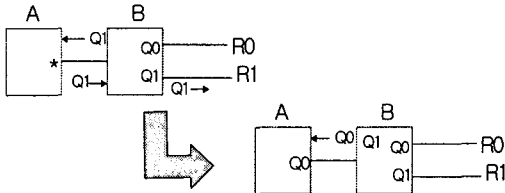
[그림 3] 예약해제 오류 - II

[그림 3]에서 볼 수 있듯이 수신자(R1)로부터의 예약요청이 라우터B를 통해 A에서 계속 라우터 용량 이상의 예약요청으로 인해 거절당한다. 이때, 다른 수신자(R0)가 Q1보다 작은 Q0의 예약을 요구한다면 중간 라우터(B)에 의해 Q0는 Q1에 병합되어

결과적으로 Q0의 예약도 거절된다. 이것을 KRP-II라 부른다. 이런 문제를 해결하는 가장 일반적인 방법은 차단상태(Blockade State) 방법이다[4].

4. 차단상태 방법에 의한 메시지 생성 및 처리

문헌[4]에서 KRP를 해결하기 위해 차단상태 방법이 제안되었다. 다음 그림은 차단상태 방법을 나타낸다.



[그림 4] Blockade state 방법

[그림 4]의 차단상태 방법에서 메시지 처리는 다음과 같다. 수신자 1(R1)의 예약요청(Q1)이 라우터(A)에서 거절되면 Q1에 대한 예약 에러메시지가 라우터(B)로 보내어지며 B의 차단상태에는 예약요구가 너무 커서 거절당한 Q1 값이 저장되어 B에서 Q1값 이상의 예약요청이 다시 병합되는 것을 막아준다.

차단상태를 생성하는 기본적인 규칙은 각각의 예약요구의 플로우스펙 LUB(Least Upper Bound)를 계산하여 병합하는 것이다. 수락제어 실패에 의해 생긴 예약 에러를 받았을 경우, 플로우스펙은 국부 차단의 요소를 생성하거나 수정하는 데 쓰인다. 기본요소로는 플로우스펙(Qb)과 타이머(Tb)로 이루어지며, Tb가 완료되면 지워진다. Explicit 예약 방식의 경우 차단상태 방법에서 고려할 요소는 (Qb(S), Tb(S))로 각각의 송신자(S)에 대해 존재하고, Wildcard의 경우는 이전 휴(P)에 따라 존재한다. 한 노드가 이전 휴에서 수락 제어 실패에 의해 생긴 예약 에러 메시지를 받았을 경우 다음과 같은 일이 발생한다.

1. P(또는 S)에 대해 차단의 요소가 생성된다.
2. Qb(P)(또는 Qb(S))가 예약 에러 메시지(Qe)와 함께 설정된다.
3. Timer Tb(P)(또는 Tb(S))가 작동되거나 Kb * R로 재시작 된다. (여기서 Kb는 미리 정의된 상수이고, R은 재생 주기이다.)
4. 만약 차단되지 않은 로컬 예약 상태가 있다면 즉시 P나 S로 예약 재생이 이루어진다.
5. 만약 InPlace 비트가 꺼져(off) 있다면, 예약 에러 메시지는 모든 다음 휴에게 전달되고, 켜져(on)있다면 Qb에 의해 차단된 Qi를 갖고 있는 다음 휴에게만 전달된다.

예약 재생 메시지를 생성하기 위해 플로우스펙을 병합하는 수정된 규칙은 다음과 같다. 차단되지 않은 로컬 예약 요구 Qi가 있기만 하면 LUB를 계산

하여 병합되고, 차단된 예약은 무시된다. 이것은 작은 예약을 요구하는 예약 메시지를 보존해 주기 위해서이다. 모든 국부 예약 요구가 차단되어 있다면 GLB(Greatest Lower Bound)를 계산한다[5].

따라서 이와 같은 차단상태 방법은 에러가 발생한 노드를 포함한 하향 라우터에 저장된다. 반면 추가적인 부분이 더 필요하다는 문제와 자원예약을 위해 많은 시간이 소요될 수 있다.

5. 제안한 방법과 구현 알고리즘

기존의 KRP-I 문제의 해결책으로 기존에 제안된 방법은 RFC2205에 나온 방법으로 예약된 예약요청을 나중에 온 예약요청에 에러가 발생하더라도 먼저 요청을 그대로 남겨두는 방법이다. 그러나, 이 방식은 먼저 예약 요청을 보존하기 위해서 과도한 대역폭 낭비를 초래한다는 문제점을 가지고 있다.

본 논문에서는 이런 문제를 해결할 방법을 제안하였는데, 방법은 수신자에서 송신자의 경로로 전체 탐색메시지(Search Message)를 전송하고 라우터의 대역폭 정보를 받아 라우터에 BS(Blockade State) 값을 생성하게 하는 방법으로 SM 알고리즘이라 명하였다.

제안된 방법은 대역폭낭비를 줄이기 위한 SM 알고리즘으로 다음 그림은 SM 예약 방법에 대해 보여주고 있다.

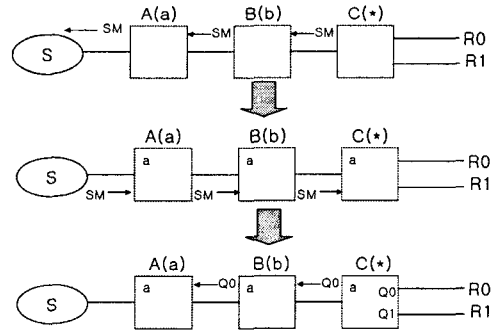


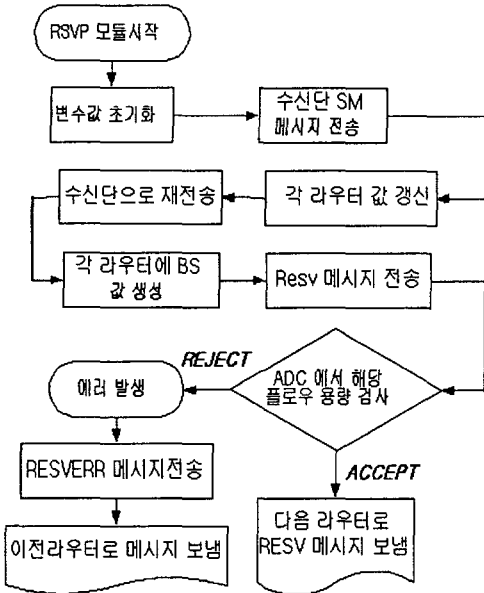
그림 5. SM 예약 구현방법

[그림 5]에서 나타낸 구현 방법은 수신자에서 Resv 메시지를 송신자로 주기적으로 Search Message(SM)를 보내는 방법으로, SM 메시지가 A, B, C라우터를 거쳐가면서 각 라우터의 대역폭 값을 검사하여 가장 최소값 대역폭 정보를 기록한다. 이 값을 기반으로 해당노드에 있는 라우터들에게 차단상태 값을 생성한다.

수신자는 Resv 메시지전송과는 별도로 SM을 주기적으로 송수신하며, 블록케이트 값을 갱신한다. 라우터에 해당 BS(Blockade State) 값이 생성되면, 차단상태 a값 이상의 요청 값은 병합하지 않고 전송하게 되며 라우터0(R0)에서 예약요청(Q0)로 자원 예약시 에러가 발생하지 않는다.

위와 같은 순서로 SM 알고리즘에서 예약을 구현하며 제안된 방법을 플로우 차트로 나타내면 다음과

같다.

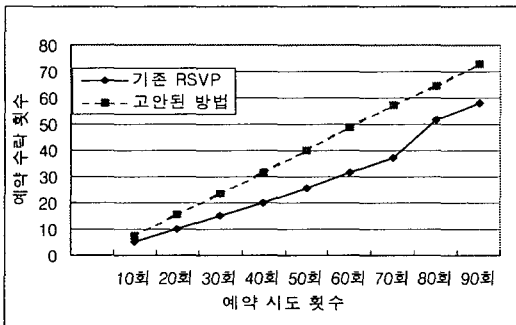


[그림 6] SM 알고리즘 플로우 차트

[그림 6]에 표현되어있는 SM알고리즘은 위에서 설명한 방법대로 순차적으로 진행하며 최종적으로 예약을 설정하여 데이터를 전송한다.

6. 모의 실험

제안된 SM 알고리즘의 타당성을 검토하기 위해 가상 RSVP망을 구축하고 모의 실험하였다. 구축된 망은 한 개의 송신자, 100개의 라우터 한 개의 수신자로 구성된다. 10번에서 90번까지 예약요청을 하였을 경우 요청용량과 예약 수락률을 비교하였다. 동일한 실험을 10회 반복하였으며 예약 수락률에 대한 실험 결과는 다음과 같다.



[그림 7] 가상 RSVP 망에서 수락률 결과비교

[그림7]에서 기존방법은 차단상태(Blockade State) 방법이고 고안된 방법은 SM 알고리즘 방법이다.

[그림7]에서 볼 수 있듯 예약시도를 70회 했을 경우, 기존 RSVP 방법에 비해 고안된 방법이 약 20회정도 더 개선됨을 볼 수 있다. 또한 제안된 방법은 평균적으로 기존방법보다 26% 더 높은 예약 수락률을 보였다. 다음으로 요청용량에 따른 예약용량을 모의 실험하였다.

표 1. 예약용량 비교

시도횟수	요청용량	예약된용량 (기존 RSVP)	예약된용량 (고안된 방법)
10회	206	73.2	127.2
20회	382	129.6	286.0
30회	592	187.0	423.3
40회	792	241.4	551.6
50회	962	296.5	698.4
60회	1132	341.7	825.1
70회	1339	389.4	960.5
80회	1502	437.2	1091.1
90회	1699	494.7	1236.5

표 1로부터 나온 결과를 통해 제안된 방법에 의한 예약 경우 예약 성공률뿐만 아니라 기존의 방법에서의 예약된 용량보다 더 많은 용량을 예약할 수 있음을 확인할 수 있었다. 이상의 실험 결과를 토대로 제안된 SM 알고리즘의 타당성을 확인할 수 있었다.

7. 결론

본 논문에서는 차세대 인터넷 프로토콜로 인정되고 있는 RSVP 예약시 발생하는 문제점 즉, RFC-2205에서 제시된 KRP-I 문제의 해결책으로 SM알고리즘을 제안하였다. 제안된 방법은 기존 방법[5]보다 대역폭 낭비 등의 문제를 줄임으로써, 효과적인 자원 예약을 수행할 수 있다. 모의실험을 통해 제안된 SM 알고리즘의 타당성을 보였다. KRP-I과 KRP-II문제를 동시에 해결할 수 있는 RSVP 방법에 관한 연구가 더 필요하다.

참고문헌

- [1] Braden, R., Ed., Zhang, L., Berson, S., Herzog, S., and S. Jamin, "Resource Reservation Protocol (RSVP) -- Version 1 Functional Specification." RFC 2205, September 1997.
- [2] Braden, R., Ed., Zhang, L., "Resource Reservation Protocol (RSVP) -- Version 1 Message Processing Rules," RFC 2209, September 1997.
- [3] Zhang, L., Deering, S., Estrin, D., Shenker, S., Zappala, D., "RSVP: A New Resource Reservation Protocol," IEEE Network, September 1993.
- [4] Berson, S., "Killer Reservation Problem" RSVP WG 12/30/94
- [5] Talwar, M., "RSVP Killer Reservation," Internet Draft, IETF, January 1999.