On-Demand SA 메시지를 이용한 멀티캐스트 송신자 발견

한두현 서창진
숭실대학교 컴퓨터학과
dhhan@kingdom.ssu.ac.kr, cjsuh@computing.ssu.ac.kr

Multicast Source Discovery using On-Demand SA message

Doo-hyun Han Chang-jin Suh
Dept. of Computing, Soongsil University

요 약

여러 도메인에 걸쳐 분포된 멀티캐스트 그룹에게 멀티캐스트 서비스를 제공하는 인터도메인(Interdomain) 멀티캐스트를 지원하기 위하여 IETF에서는 MSDP 방식을 제안하였다. MSDP는 송신자를 기반으로 트리를 구성하기 때문에 수신자는 송신자의 주소를 알고 있어야만 서비스를 받을 수 있는데 진행중인 서비스에 새로이 가입하는 수신자들은 송신자에 대한 정보가 없어서 이 정보를 제공하기 위해서 주기적인 플러딩(flooding) 방법을 사용하고 있다. 이로 인하여 멀티캐스팅의 중요한 성능요소인 확장성이 손상되었다.

이 논문에서는 플러딩을 대신할 새로운 방식으로 On-Demand SA를 제안하였다. 이 방법은 송신자들을 여러 서버에 분산하여 설치하여 멀티캐스트 그룹들은 이 서버에 접속하여 송신자가 누구인지를 확인한다. 이러한 절차는 송신되는 정보량을 줄일 수 있을 뿐 아니라 바뀐 송신자의 정보를 멀티캐스트 그룹이 알기까지의 지연시간을 줄여줄 수 있다.

1 서론

현재 인트라도메인용으로 사용되는 멀티캐스트 라우팅 프로토콜인 DVMRP, PIM-DM, PIM-SM은 인터도메인 영역에서는 다음과 같은 이유로 사용하기가 부적합하다. DVMRP[1]와 PIM-DM[2]은 반복적으로 플러딩이 발생하여 많은 자원을 소모한다는 점과 송신자 기반의 멀티캐스트 트리를 구성하기 때문에 멀티캐스트 송신자의 수와 멀티캐스트 그룹의 평균 회원수의 곱에 비례하는 많은 정보를 교환해야 하는 단점이 있다. PIM-SM[3]에서는 데이터의 플러딩을 없애고 송신자 수에 관계없이 그룹에 한 개의 트리만을 유지하여 확장성을 개선되었으나 각 도메인을 담당하는 라우터인 RP (Rendezvous Point)들 사이에서 교환되는 정보가 크게 증가하고 멀티캐스트와 관련이 없는 RP의 리소스를 많이 소모하는 third-party 문제로 사용하기 어렵다.

IETF에서는 인터도메인 멀티캐스트를 지원하기 위해서 장단기적인 해결방안을 제시하고 있다. 단기적으로는 현재 사용되는 프로토콜을 이용하여 새로운 프로토콜인 MSDP(Multicast Source Discovery Protocol)를 제안하였으며, 장기적으로는 뛰어난 확장성을 자랑하는 BGMP(Border Gateway Multicast Protocol)[5]를 이용하며 아울러 주소를 계층적으로 할당하는 방법인 MASC (Multicast Address Set Claim)[4]를 적용시키고 있다. 그러나 후자는 주소체계의 변경이나 복잡한 구조를 구현하는 문제로 당분간은 단기적인 방법을 사용해야 한다.

본 논문은 MSDP를 개선하는 방법을 다룬다. MSDP는 독립적으로 운영되는 각각의 도메인에서 PIM-SM를 사용하여 만들어진 각각의 트리를 하나로 연결하는 프로토콜이다. 이 과정에서 MSDP는 peer 도메인 사이에 송신자 정보에 관한 메시지를 교환한다. 자신의 도메인 내에서 송신자가 발생하면 그 도메인의 RP는 송신자 주소, 멀티캐스트 주소, RP의 주소를 SA(Source Active)메시지에 실어서 송신이 이루어지는 동안 주기적으로 모든 peer 도메인의 RP로 플러딩한다[6]. 그러나 플러딩으로 인하여 SA메시지가 너무 빈번하게 발생하여서 확장성이 떨어지게 되며, 평균적으로 플러딩 주기의 반만큼 트리 구성이 지연되어 이 시간 동안에 전송된 데이터가 손실된다. 이 본문에서는 이 두 문제를 해결하는 On-Demand SA 방식을 제안하였다.

논문은 다음과 같이 구성하였다. 다음절에서는 PIM과 MSDP의 동작을 설명하고 3절에서 이를 분석하였다. 그 결과 제안되는 알고리즘을 4절과 5절에서 소개하고 개선점을 제시하였다.

2 PIM과 MSDP의 동작

인터도메인 멀티캐스트는 PIM과 MSDP 프로토콜을 이용하여 수행된다. 트리는 PIM-SM에 의해서 이루어지며 MSDP가 다른 도메인에 위치한 송신자 정보를 전달해주면 PIM-SM를 이용하여 RP는 가입메세지를 송신자에게 전달한다.

2.1 PIM-SM

DVMRP와 PIM-DM 같은 송신자 기반의 트리를 구성하는 프로토콜은 멀티캐스트 그룹내의 송신자 수 만큼의 독립적인 트리를 형성하며 각 송신자의 데이터는 각각의 트리를 통하여 전달된다. 이에 반하여 공유트리를 구성하는 프로토콜은 송신자에 관계없이 그룹 당 단일한 트리를 구성한다. PIM-SM에서는 많은 데이터를 발생하는 송신자에 대해서 만들어진 공유트리를 송신자 기반의 트리로 변환할 수 있게 하였다.

공유트리는 트리의 중심점으로 RP를 사용한다. 각 라우터는 그룹의 주소와 RP를 매핑하는 함수를 이용하여 RP의 주소를 계산하여 알아낸다. 멀티캐스트 그룹에 참여를 원하는 각 수신 라우터는 찾아낸 그룹의 RP로 참가메시지를 전송한다. 참가메시지를 수신한 중계(intermediate) 라우터는 RP로 가는 경로를 설정하면서 트리의 가지를 형성한다.

송신자가 멀티캐스팅을 시작하면 아직 트리가 만들어지지 않은 상황에서 첫 홉 라우터는 레지스터라는 명령어를 유니캐스트 형태로 캡슐화하여 RP까지 터널링시켜서 RP에게 전달하는데 이러한 패킷 형태를 레지스터(register) 패킷이라고 부른다. 새로이 송신자가 발생한 것을 알게된 RP는 송신자에서 RP를 잇는 트리를 만든다. 트리가 형성되면 첫 홉 라우터는 더 이상 레지스터 패킷형태로 데이터를 전송하지 않는다. 한편 RP가 레지스터 형태의 패킷을 수신하면 패킷의 캡슐을 제거하여 다운스트림에 있는 그룹의 수신자들에게 RP와 연결된 공유트리를 통하여 데이터를 전송한다.

2.2 MSDP 동작

RP는 이웃 도메인에 위치한 RP와 TCP연결을 통하여 제어정보를 교환하는 peering 관계를 맺은 후 MSDP를 수행한다. MSDP는 SA 메시지를 두 가지 경우에 발생한다. 하나는 새로운 멀티캐스트 송신자가 발생을 감지한 후 즉시 발생시키는 것이며 다른 하나는 도메인 내에서 데이터를 발생 시키고있는 모든 송신자 정보를 주기적으로 전달하는 경우이다. 전자의 경우 발생하는 SA메시지를 SA_T 후자의 형태를 SA_P라 한다. SA_T는 송신자 정보를 빠르게 전달함으로써 지연을 줄여주는 역할을 하며 SA_P는 늦게 가입의사를 표현하는 수신자들을 위해서 송신자 정보를 제공한다. 그림1에서는 도메인 A에 위치한 송신자 S가 다섯 개의 수신자(R1에서 R5)에게 전달되는 멀티캐스트 트리를 구성하는 과정을 설명하고 있다. 수신자가 존재하는 도메인(도메인 A, B, D)은 PIM-SM 프로토콜을 이용하여 멀티캐스트 트리가 구성되어 있는데 이는 굵은 실선으로 표현됐다. 또한 각 도메인마다 선정된 RP가 일반 라우터와 구별되기 위해서 회색으로 표시되었다. 전체적인 동작은 SA메시지가 만들어지는 시점과 SA메시지를 수신하는 시점을 기준으로 순차적으로 ①, ②, ③ 세 단계로 나뉘어 진다. 이 중 ②는 MSDP 프로토콜에 의해서, ①과 ③은 PIM-SM 프로토콜에 의해 수행된다.

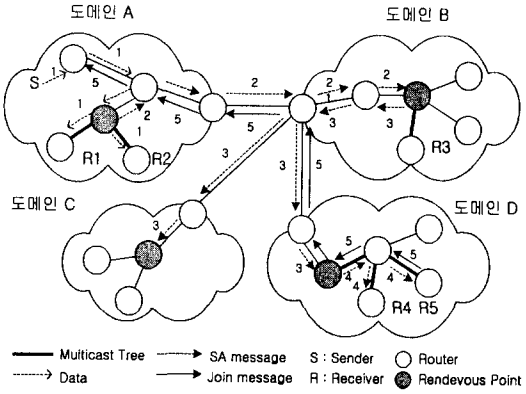

그림1. Source Active 메시지를 포함한 MSDP 동작

① SA메시지의 발생 이전 단계

i. 새로이 발생한 송신자(S)는 도메인 내의 RP로 레지스터 패킷을 전송하고 RP는 이를 멀티캐스트 트리를 통해서 도메인 내의 모든 수신자에게 전달한다.

② SA메시지를 발생 및 전달 단계

ii. 새로운 송신자가 발생한 것을 알게된 RP는 이 사실을 peer관계인 모든 RP에게 방송하기 위해서 SA_T메시지를 플러딩한다. peer 관계에 있는 두 도메인 A, B를 (A,B)로 나타낸다면 그림1에서는 {(A,B), (B,C), (B,D)}가 peer관계이다. 도메인A의 RP인 RP_A는 SA_T메시지를 RP_B로 전달한다.

iii. SA메시지를 전송받은 RP_B는 SA메시지가 루핑되지 않도록 SA메시지를 발생시킨 본래의 도메인으로부터 최단 peer로 SA메시지가 전달됐는지를 점검한다.(peer Reverse Path Forwarding). 만약 올바르게 수신했다면 SA를 RP_C와 RP_D로 SA메시지를 전달한다. 만약 그림1에 도메인 C가 도메인 도메인 B를 통하여 수신한 도메인 A의 SA메시지를 다시 도메인 B로 전달하면 SA메시지가 도메인 A로부터 도메인B로 향하는 최단 peer가 아니기 때문에 무시한다.

③ SA 메시지 수신 이후 단계

iv. SA메시지를 수신한 RP는 그 도메인에 멤버가 있을 경우 SA메시지에 기록된 주소로 가입 메시지를 전송한다.

v. 만약 SA메시지에 데이터가 포함되어 있다면 RP는 멀티캐스트 트리로 데이터를 전송한다. 데이터를 수신한 그룹 멤버는 PIM-SM 송신자 기반 트리의 변환을 결정한다.

이후 RP_A는 도메인 내에 데이터를 발생시키는 송신자가 있는 동안 SA_P를 발생시키며 iii~v 의 과정을 반복한다.

3 MSDP 분석

MSDP에서는 다음과 같은 문제가 발생하고 있다.

3.1 신규가입의 지연

멀티캐스트 서비스가 진행 중에 가입자가 가입하려는데 신규 가입자가 속한 도메인 내에서는 기존의 가입자가 하나도 없어서 트리가 만들어져 있지 않는다고 가정하자. 이 경우 수신자가 속한 도메인을 담당하는 RP는 자신과 신규가입자를 잇는 트리와 RP와 송신자간의 경로를 형성해야 한다. 후자의 경로를 형성하기 위해서 RP는 해당한 멀티캐스트 서비스의 송신자가 어느 도메인에 있는지를 알고 있어야 한다. MSDP에서는 peer 관계에 있는 RP들에게 주기적으로 송신자가 어느 도메인에 위치하는가를 기록한 SA메시지를 모든 라우터에게 플러딩 받아야 알 수 있다. 따라서 신규가입으로 트리를 새로 만들 경우는 SA메시지의 플러딩 주기의 절반 기간 동안가입이 지연되며 이 기간 동안은 데이터도 수신할 수 없다.

또한 MSDP에서는 전체 도메인에서 동시에 플러딩이 발생하지 않도록 각 도메인마다 다른 시점에 플러딩을 실시하기 때문에 도메인에 따라서 송신자 트리가 만들어지는 시간에 차이가 생긴다.

3.2 MSDP 확장성

도메인간의 멀티캐스트를 구현하는 서비스에서 가장 중요한 점은 확장성이다. MSDP는 SA메시지를 주기적으로 플러딩하므로서 스스로 확장성을 훼손하였다. 특히 플러딩은 모든 도메인에서 발생하여 모든 도메인으로 퍼져 나가기 때문에 도메인의 수의 제곱에 비례하는 패킷이 플러딩하는 과정에서 소요된다.

4 MSDP의 개선

앞 장에서 언급한 MSDP의 문제들을 해결할 수 있는 방안을 제안하고, 그 결과 어떠한 개선이 이루어지는지를 살펴본다.

4.1 On-Demand SA 방식

MSDP에서는 새로운 송신노드(N_S)가 발생했을 때 N_S가 속한 도메인의 RP인 RP_S가 변경된 N_S만의 주소 정보를 담은 SA메시지를 모든 RP에게 플러딩하며, 또한 자신의 도메인 내에서 데이터가 발생하여 전송되는 멀티캐스트 서비스의 송신자 주소를 가지고 있는 주기성를 가진 SA를 약 1분 간격으로 다른 모든 RP에게 플러딩한다. 이 중 후자 부분을 제안된 On-Demand SA방식에서는 플러딩 대신에 다음과 같은 방법을 사용한다.

특정한 RP인 RP_D을 선택하여 이곳에 주어진 멀티캐스트 주소의 송신자의 수소에 대한 정보를 보관한다. 즉 새롭게 수신자가 발생하여 송신자로부터 트리를 연결해야 하는 RP인 RP_R는 우선 송신자의 주소를 알아야 한다. 만약 자신의 도메인에 이미 수신자가 있다면 이 수신자가 수신할 수 있도록 트리가 만들어져 있을 것으므로 신규가입자와 이 트리를 연결시킨다. 만일 만들어진 트리가 없다면 RP_R은 이러한 정보를 RP_D에게 문의한다. RP_D는 자신이 간직한 데이타베이스를 참조하여 문의한 멀티캐스트 그룹의 송신자가 $N_{S1}, N_{S2}, \ldots, N_{Si}$임을 RP_R에게 알려준다. RP_R은 $N_{S1}, N_{S2}, \ldots, N_{Si}$에게 가입메시지를 전송하여 수신을 원하는 멀티캐스트 그룹의 모든 송신자와 트리를 형성하여 송신자들이 보낸 멀티캐스트 데이터를 수신하고 이를 자신의 도메인 내의 수신자에게로 전송한다.

제안된 방법에서 핵심적인 부분은 RP_R와 RP_S가 사전에 정보 교환을 하지 않은 상황에서 어떻게 동일한 RP 즉 RP_D에게 관련된 정보를 주거나 문의하느냐는 점이다. 우선 전체 인터넷에 한 개의 RP_D만을 설정하면 간단히 해결할 수 있으나 신뢰성이나 부하의 집중도를 생각할 때 비현실적인 발상이다. 따라서 모든 RP는 자체 송신자 데이타베이스를 구축하여 분산된 서비스를 제공해야 한다 이를 위해서 사전에 모든 RP에서 공히 사용하는 해쉬(hash)함수를 정하고 송신자 도메인이 새로운 송신

자가 발생한 경우 RP_S와 RP_R은 동일한 해쉬 함수를 이용하여 RP를 계산하여 그 결과 해당 PR가 RP_D임을 알아낸다. 결과 해당 RP가 RP_D를 알아낸다. 해쉬함수는 추후에 자세히 설명한다.

4.2 개선효과

제안된 방법으로 다음과 같은 긍정적인 효과를 얻을 수 있다.

4.2.1 플러딩 감소

MSDP는 송신자가 바뀐 경우 SA메시지를 플러딩하며, 또한 주기적으로 대개 1분 간격으로 송신자가 자신의 도메인에 위치한 모든 멀티캐스트 어드레스를 플러딩하였으나 On-Demand SA에서는 전자의 경우에만 플러딩이 발생하기 때문에 플러딩의 수를 대폭적으로 줄인다. 특히 보편적인 멀티캐스트 방법인 수신자가 많지 않은 sparse-mode 멀티캐스트의 경우 대부분을 차지하는 수신을 원하지 않는 도메인에게로의 SA 메시지 전송을 억제할 수 있다. 또한 후자의 메시지는 전자에 비해서 매우 길이가 긴 메시지이므로 큰 효과를 거둘수 있다.

4.2.2 가입지연의 감소

인터넷을 배회하는 사용자들은 대개 10초 동안 응답이 없으면 다른 site로 옮겨가 버린다. 이처럼 인터넷 상에서 상업적으로 멀티캐스팅 서비스를 제공하는 업체들에게 고객의 초기 가입 지연은 고객을 놓치는 치명적인 결점이 된다. 제안된 방식은 원하는 멀티캐스트 서비스의 송신자가 속한 도메인을 알아보기 위해서는 특정 RP에게 문의하고 이의 응답을 받는 처리시간이 소요된다. 반면 MSDP의 경우는 관련 정보가 주기적으로 플러딩될 때까지 수동적으로 기다려야 하는데 플러딩의 주기는 발생되는 데이터의 양에 반비례하므로 함부로 짧게 할 수 없으므로 상당한 시간을 대기해야 한다. 이처럼 On-Demand SA는 가입지연 시간을 상당히 감소시켰다.

5 On-Demand SA 방식의 구현

5.1 RP-Set 구성

평소 RP들은 다른 도메인의 RP들과 제어정보를 교환하기 위하여 TCP 연결기반으로 peering 관계를 맺고 있다. On-Demand SA에서는 모든 RP가 전체 RP-Set을 관리해야 한다. 새로이 추가될 RP는 기존의 RP에게 자신을 신고한다. 또한 기존의 도메인에 고장이 발생했다면 이웃의 RP는 곧 이 사실을 알게된다. 이처럼 전체 RP-set에 변화가 생길 경우 이를 처음 접수한 RP는 모든 RP에게 플러딩한다.

일반적으로 RP들은 매우 안정되게 동작하고 있으므로 RP의 고장과 신규등록은 매우 가끔씩 발생하는 일이다. 그러나 일단 발생하면 짧은 시간동안 각각 RP에 보관된 데이타베이스에게는 불일치하지 않는 부분이 발생할 수도 있다. 약간의 내용이 불일치한 데이타베이스를 사용해도 거의 동일한 우선순위를 산출하는 해쉬함수를 정의하자.

5.2 RP 매핑 함수

모든 RP는 2단계로 구성된 해쉬 함수를 가지고 있다. 단계를 둘로 나눈 이유는 한번 원하는 RP_D를 찾기 위해서 해쉬 함수 f를 반복적으로 사용해야 하는데 총 수행되는 횟수를 줄이기 위해서이다. 첫번째 단계는 *Modular* 연산을 통하여 RP-Set의 소집합의 인덱스 i를 구하고 두번째 단계에서는 선택된 소집합 i에 속한 전체 RP에 대해서 주어진 룰에 따라 순위를 매긴다. SA정보를 제공할 RP는 계산된 순위에 따라 응답하는 두 RP에게 관련된 내용을 보관시키며, 한편 SA정보를 받아와야 할 RP는 계산된 순서로 필요한 SA 내용을 자신에게 보낼 것을 요구한다.

1단계

$$i = f(k) \bmod n \qquad (1)$$

식(1)에서 f는 해쉬함수이고, n은 상수로서 f와 n은 모든 RP에서 모두 알려져 있는 고정된 함수와 숫자이다. k는 멀티캐스트 주소이며, i는 소집합의 인덱스이다. 위의 연산을 통하여 RP-Set은 0부터 (n-1)의 번호를 가지는 n 개의 소집합으로 나뉘지며 각각의 소집합의 크기(cardinality) 즉 속한 RP의 수는 해쉬함수의 성질에 의해 거의 비슷하게 분배된다.

2단계

HRW(Highest Random Weight)알고리즘을 이용하여 소그룹 i에 속한 모든 RP에 대해서 h를 계산하여 이들 사이에 우선순위를 정한다.

$$h = i_p \ : \ (f(k,\ l(i_p)) > (f(k, l(i_q)),\quad 0 \le i_p,\ i_q < \lfloor \frac{RP의\ 수}{n} \rfloor \qquad (2)$$

식(2)은 다음과 같이 동작한다. 멀티캐스트 주소와 RP의 주소가 임의의 가중치(random weight)를 구하기 위해 사용된다. 각 RP는 가중치에 의해서 정렬된다. $l(i_p)$는 i번째 소집합의 p번째 RP의 유니캐스트 주소이며 f는 k와 $l(i_p)$를 입력 값으로 하는 pseudo-random 함수이다 [7].

5.3 송신자 정보 검색 절차

송신자 정보를 제공하는 RP_S와 송신자 정보를 받고자 하는 RP_R의 데이터베이스가 동일할 경우는 양측에서 RP그룹들이 동일한 우선순위로 배열된다. 따라서 두 개의 RP에 송신자 정보가 보관되며 이 두 RP 모두가 down된 경우가 아니라면 RP_R는 원하는 정보를 받아올 수 있다. 그러나 이 두 RP가 일치하지 않을 경우는 일치되지 않은 부분에 대해서 서로 다른 순서가 매겨질 수 있다. 이처럼 RP의 고장이나 신규가입으로 야기되는 약간의 혼란을 다음과 같은 방법으로 극복한다.

RP_S가 식(2)를 사용하여 얻어진 RP의 순서를 RP_1, RP_2, RP_3, RP_4라고 칭하자. RP는 RP_1과 RP_2에 송신자 정보를 제공하고 이의 응답을 기다린다. 잘 보관되었다는 응답이 두 곳에서 오면 수행이 종료된다. 만일 한 개가 오면 RP_3에게 두 곳에서 공히 응답이 없으면 RP_3과 RP_4에게 보낸다. 그러나 양쪽에서 응답이 없는 경우는 RP_1과 RP_2가 동시에 down되었음을 의미하므로 현실적으로는 발생하지 않는다.

RP_R는 식(2)를 이용하여 얻은 RP_1, RP_2 양쪽에게 수신자정보를 요구한다. 만일 RP_1과 RP_2 중 한쪽에서 회신하면 검색을 종료한다. 만일 양쪽 모두에서 회신이 전달되지 않았다면 RP_3, RP_4에게 다시 한번 문의한다. 이때 문의에 실패한 RP명단(RP_1, RP_2)을 메시지에 포함시킨다. 받은 RP_3 혹은 RP_4는 정보가 가지고 있으면 이를 회신한다. 만일 자신에게 정보가 없으면 식(2)를 이용하여 자체적으로 점검한다. 그 결과 자신보다 우선순위가 더 높지만 실패한 명단에는 없는 RP가 있다면 이 RP에게 동일한 질문을 하고, 그 회답을 RP_R에 게 할 것을 요구한다. 이때 실패한 명단에 자신을 포함시킨다.

6 결론

본 논문에서는 MSDP에서 사용되는 송신자 확인 방법을 주기적으로 플러딩하여 방송하는 방법 대신 여러 RP에 관련정보를 분산하여 보관하고 이 정보가 필요한 RP는 즉각 이 정보를 찾아갈 수 있는 방법을 제시하였다.

이러한 방법은 네트워크 운영자측면에서는 많은 부하를 초래하는 플러딩을 사용하지 않음으로서 확장성을 높여주며, 정보 제공자측면에서는 새로이 가입하는 가입자들에게 신속하게 서비스를 개시하게 하여서 고객의 불만을 줄이는 효과를 얻을 수 있다.

참고문헌

[1] D. Waitzman et al., "Distance Vector Multicast Routing Protocol," RFC1075, Nov. 1988

[2] S. Deering et al., "Protocol Independent Multicast Version 2 Dense Mode Sepcification," *Internet draft*, dratf-ietf-pim-v2- dm.txt, Nov. 1998

[3] D. Estrin et al. "*Protocol Independent Multicast-Sparse Mode (PIM-SM)*: Protocol Specification," RFC2117, June 1997

[4] Deborah Estrin et al., "The Multicast Address-Set Claim (MASC) Protocol", *Internet draft*, draft-ietf-malloc-masc-05.txt, July 2000

[5] D. Thaler, "Border Gateway Multicast Protocol (BGMP): Protocol Specification", *Internet draft*, draft-ietf-bgmp-spec-01.txt, Mar. 2000

[6] Dino Farinacci et al. "Multicast Source Discovery Protocol (MSDP)," *Internet dratf*, draft-ietf-msdp-spec-05.txt. Feb. 2000

[7] D. Estrin et al., "A Dynamic Bootstrap Mechanism for Rendezvous-based Multicast Routing," *In Proceedings of IEEE INFOCOM'99*, New York, pages 1090-1098, Mar. 1999