

IP-TV가 IP multicast를 부활시킴

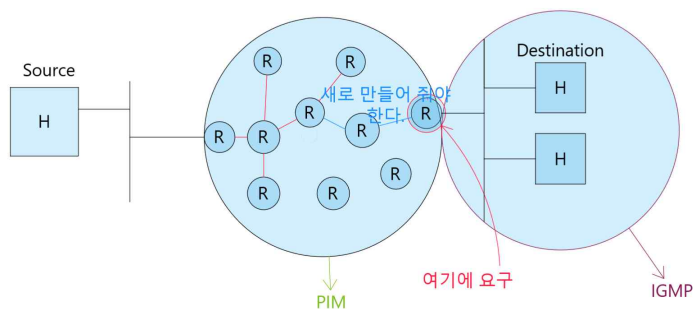
unicast로 방송하려면 100만명을 감당하려면 엄청난 server와 광대역이 필요하다

multicast는 그냥 연약한 smartphone하나로도 방송가능

IP multicast니까, IP 영역에서 일이 일어남

router가 복사기 역할을 해준다.

IP Multicast는 router의 끝에는 Ethernet이 있다고 가정한다.



PIM: tree 관리 protocol

IGMP: 누가 subscriber인가? -> membership 관리 protocol

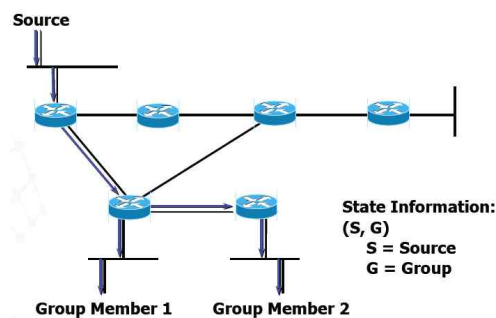
Internet 전체를 사용하는 multicast는 없다.

하나의 network망 안에서만 사용 (고대망, KT망 등)

-PIM-

(1) 다익스트라 알고리즘으로 shortest path tree를 구현한다.

그런데 사실은 Reversed Shortest Path Tree이다.

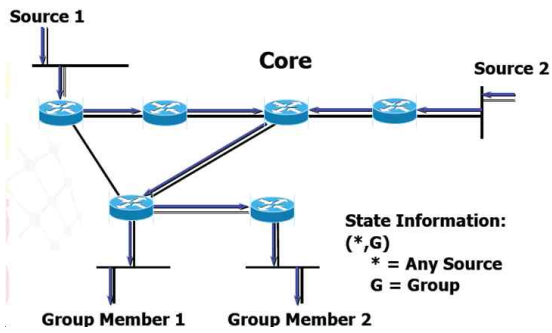


Group Member 1에서 Source까지 간다.

Group Member 2에서도 Source까지 가다가 중간에 길이 있으면 접붙이고 끝이다.

-> 이 방법은 Source의 주소를 알 때만 가능하다.

(2)



-> host가 Source의 주소는 몰라도, Core의 주소를 알아야한다.

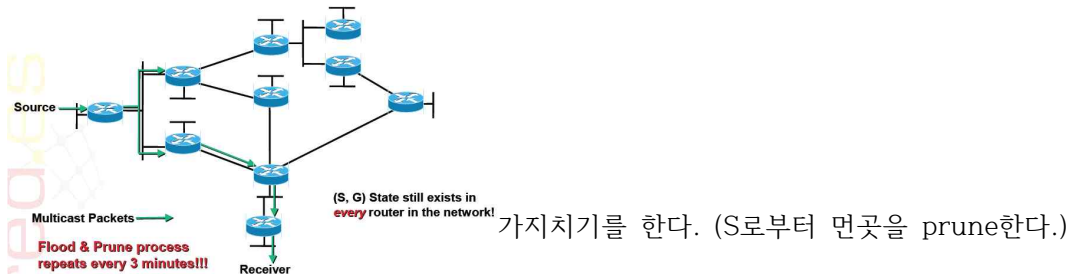
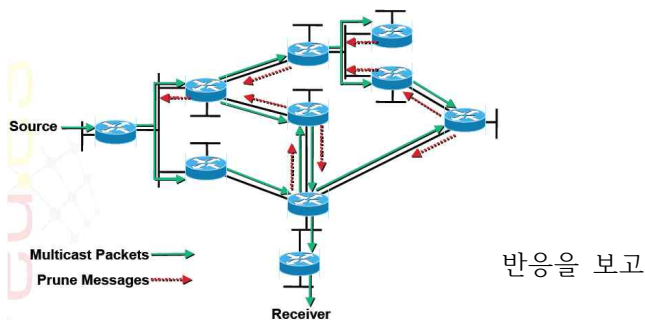
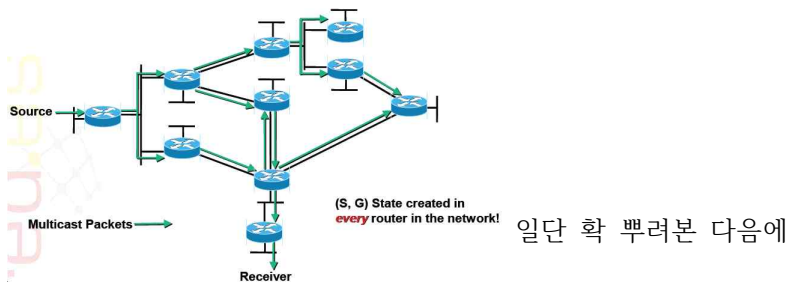
-> IP TV는 두 가지 방식을 모두 사용한다.

MBores: host끼리 복사기 역할을 한다. (이게 최초의 multicast같은거 였다.)

-> 전부 unicast를 사용한다.

분배구조를 어떻게 가지나?

DVMRP: flood-and-prune



2분마다 다시해서 반응을 본다. 0->X, X->0가 될 수 있다!

PIM: 라우터가 지원하기 시작.

PIM Dense Mode = flood-and-prune

flood-and-prune은 대부분의 사용자가 그 트래픽을 받아보길 원할 때 효율적이다.

-> 그 반대 상황은 비효율적.

-Source or Shortest Path Tree(SPT)- :가장 효율적

문제1: 소비자는 생산자가 누구인지 알아야 한다.

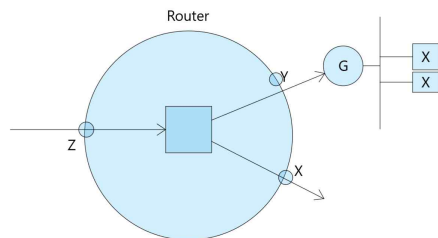
IP 주소로 class D를 준다.

Source Class D를 받으면 내부 이더넷에 뭐가 있는지도 모른다.

-> IP multicast는 멤버가 어떻게 될지를 신경쓰지 않는다.

문제2: 너무 많은 Routing table이 필요하다.

(사실 multicast용 Routing table이 따로 존재한다.)



S->G일 때 X, Y로 보내라

S->G | X, Y

만일 Y에 아무도 원치 않으면,

S->G | X

으로 Y를 라우팅 테이블에서 제거한다.

라우팅 테이블은 처음에 flood-and-prune를 할 때 만들어진다.

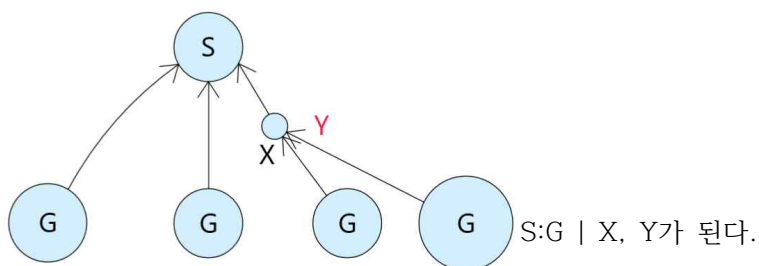
최악의 경우, $O(SG)$ 를 일일이 관리 해야한다.

-> 라우터가 관리해야 할 table이 너무 많다.

“문제1: 소비자는 생산자가 누구인지 알아야 한다.”

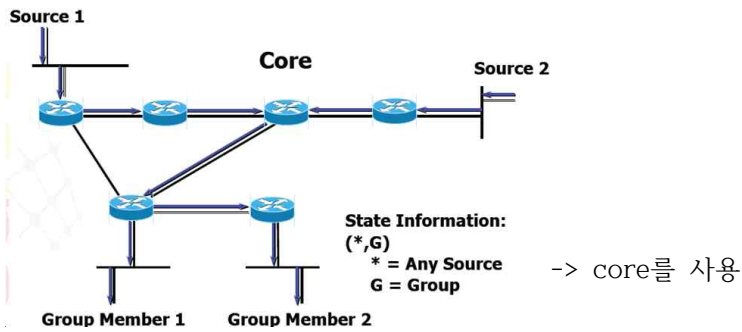
-> Reversed Shortest Path Tree로 해결한다.

G가 S를 찾아감.



한 라우터씩 올라가면서 올라온 interface들을 추가해라! 라고 말한다.

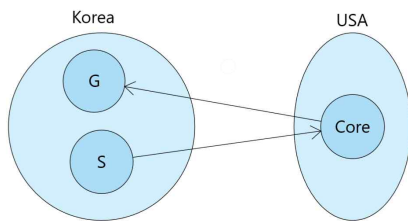
“문제2: 너무 많은 Routing table이 필요하다.”



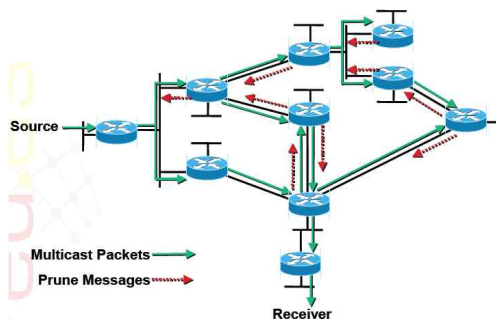
core가 생기면 source를 생각하지 않아도 된다. core만 알면된다.
 -> $O(G)$ 이 된다.

core는 효율이 낮아질 수 있다.

만일 S와 G가 한국에 있는데, Core가 미국에 있으면 효율이 낮아진다.



-RPF Check-



-만약에 2개의 interface로부터 S의 신호가 온다면?

-> S로부터 먼곳을 향해 있는 interface를 사용하지 않는다.

RPF Check는 unicast를 사용한다.

unicast routing table이 따로 있다. (unicast routing table은 잘 안 바꾼다.)

-> RPF Check를 할 때, unicast routing table lookup만을 해보고, 없는 건 입구를 닫아버림

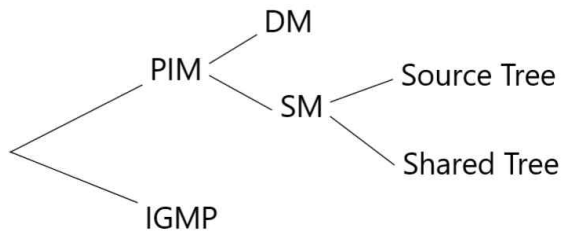
If multicast path is different from unicast path, then a multicast table will exist

어차피 unicast routing table은 하나로 이미 최적화되어 있을 것이다.

그래서 만약에 unicast로 보냈는데, 응답이 없으면 그 interface를 닫는 것이다.

-> 2분마다 다시하기도 한다.

-정리-



PIM: multicast, router 존재

IGMP: 첫 번째 홉에만 사용

DM(Dense Mode): Flood-and-Prune, RPF Check

Source Tree: 배달효율↑, 라우터 상태 정보↑

Shared Tree: 일종의 “RP(랑데부 포인트)”가 있다. 배달효율↓, 라우터 상태 정보↓

Shared Tree는 Source Tree로 옮겨간다.

PIM은 wireshark로 안 잡힌다.

가입자들이 처음에는 S가 누구인지 알지 못한다. 오직 Core만 안다.

-> 처음에는 C에게 traffic을 보고 싶다고 이야기한다.

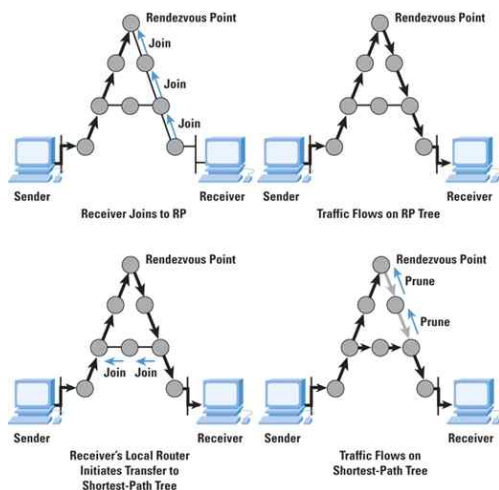
-> S를 알게 된다.

-> S에게 직접 G가 traffic을 달라고 요구

Shared Tree는 의무이다.

Source Tree는 선택이다.

(Core Router는 unicast, multicast 모두 해줄 수 있다.)



-IP multicast building blocks-

Class D (224.0.0.0~239.255.255.255) 중에서 하나를 선택하는 문제이다.
multicast는 정말 주인이 없다.

global multicast는 불가능하다

-> 단 하나의 network라도 협조를 안 하면 그냥 안 된다.

multicast의 scope -> 망 안에서만 사용가능

-> 239/8: multicast 버전의 사설주소

TTL=1->라우터를 넘을 수 없다.

-IP multicast services-

1. Any Source Multicast (ASM)
2. Source Specific Multicast (SSM)

-Any Source Multicast (ASM)-

: 어디서 왔는지 전혀 신경 쓰지 않는다.

Group G에 뿌리면 그냥 무조건 다 받아야 함. -> 보안의 문제

1. G만 일치하면 그냥 다 받아야 한다.
2. G주소로 coordination하는 문제가 있다.

-> multicast는 주인이 없다.

-> 서로 다른 콘텐츠가 동일한 G주소를 잡음

-> 화면이 까맣게 나오거나 하면서 망함

-Source Specific Multicast (SSM)-

누구로부터 왔는지까지 다 따져줌

이걸 쓰려면 PIM, IGMP의 버전 업이 필요했다.

SSM은 coordination이 쉽다.

