Status of IPv6 SSM

Stig Venaas
UNINETT
venaas@uninett.no
Introduction

- IPv6 and multicast
- What is SSM
- Why SSM
- SSM and PIM routers
- SSM and MLDv2
- SSM support in operating systems
- SSM applications
- SSM and applications with dynamic sources
- Emulating ASM with SSM
- SSM testing

http://domen.uninett.no/~venaas/ssmstatus.pdf
IPv6 and multicast

- IPv6 gives a new chance at deploying multicast
  - All IPv6 hosts support multicast
  - No NATs

- Broadband user can e.g. stream video to thousands of people
  - Requires both IPv6 and multicast

- More content and services available to everyone on the Internet

- Multicast has generally been difficult to deploy and manage
  - SSM helps
What is SSM

- Traditional multicast (Any-Source Multicast)
  - Receivers join a group $G$
  - Sources send packets addressed to $G$
  - All members receive them
  - Network takes care of the rest
  - Independent "sessions" should use different $G$

- Source-Specific Multicast
  - Receivers explicitly join sources, $(S_1, G), (S_2, G), ...$
  - Sources send packets addressed to $G$
  - Receiver only receives from the sources it joined
  - Independent "sessions" can use same $G$, but not same $(S, G)$
  - Receiver needs to learn what sources to join, not just $G$

- Some IPv4 and IPv6 group ranges are for SSM only
- Other ranges can be used for both ASM and SSM
  - Some can join $(S, G)$ and only receive $S$
  - Others can at the same time join $G$ and receive all sources
Why SSM

- Removes complexity from network
  - No Rendezvous-Points, hence no need for BSR, MSDP etc.
  - Always use PIM Shortest Path Trees, easier to debug

- Hence easier to deploy and manage

- More secure
  - Only receive packets from specified sources
  - For e.g. tv broadcast you join a specific source
  - You will not receive packets other sources send to group

- Hence better suited to typical broadcast use

Why not SSM

- More complexity in hosts/applications (source discovery)
- Always per-source state in routers
SSM and PIM routers

- SSM works with PIM-SM routers as is
  - Shortest Path Trees part of PIM-SM

- However, additional requirements for SSM group ranges
  - Must not accept (*,G) joins for G in SSM range
    - No shared tree
  - Routers must not send PIM register to RP
  - An RP must not accept PIM register
  - Or, simply not allow group-to-RP mapping for SSM G
    - With new PIM spec the above then follows

- SSM still works without requirements
  - But, want to enforce SSM-only use for SSM group ranges

- I think all current PIM-SM routers meet the requirements

- In addition edge routers need to support MLDv2 (IGMPv3)
SSM and MLDv2

- Edge routers need to support MLDv2
  - IGMPv3 for IPv4

- Many routers now support this with recent software
  - Cisco, Juniper, *BSD with pim6sd
  - Not yet XORP I think
  - Not sure of others

- MLDv2 allows listener to specify either
  - Interest in only specific sources for a group
  - Interest from all but specific sources
    - Blocking specific sources

- Some issues seen with routers and hosts using different ICMPv6 protocol numbers for MLDv2
  - Now assigned by IANA and specified in RFC 3810
SSM support in operating systems

- MLDv2 needs to be implemented in host; available in:
  - Recent Linux kernels
  - FreeBSD with KAME patches
  - Solaris 10 pre-release
  - Others?

- Hosts need to provide API to applications, RFC 3678
  - More or less ok for all the above
  - Linux only part of API and only in kernel header files
  - Sample Linux code: http://www.uninett.no/testnett/multicast/mctest/

- API allows both joining and blocking sources
- Contains protocol independent functions
  - Useful also for protocol independent code joining G

- No support needed for sending multicast
SSM applications

- Very few IPv6 applications available, a few more with IPv4
  - Any multicast application can send SSM, nothing special

- KAME utilities, mcastsend, mcastread

- PSNC did some tests and have some patches
  - dtms & dtmc, dvts & xdvshow, mad flute

- MAD Flute - http://www.atm.tut.fi/mad/
  - Reliable (FEC) multicast content delivery
  - Daily used with SSM in 6NET and M6Bone

- ssmsdpifier - http://clarinet.u-strasbg.fr/~hoerdt/libssmsdp/
  - Traditional multicast application turned into an SSM application
  - Works without changing code or binary
  - Tested with NLANR multicast beacon
SSM and applications with dynamic sources

- SSM easy for applications with a few static sources
  - In particular with 1
  - All sources are announced in advance
  - Receiver joins all sources

- SSM is difficult with dynamic sources
  - That is, sources that come and go during session
  - Needs to do source discovery in application
    - Replacing the RP functionality

- One solution is SSMSDP
  - Source Specific Multicast Source Discovery Protocol
  - Not standardized
  - Used by ssmsdpifier

- Another similar proposal is draft-lehtonen-mboned-multissm-00.txt
The basic idea is as follows:

- There is a controller C, and a control channel (C,G)
- (C,G) is announced and receivers join it
- Sources S regularly unicast a source announcement to C
- C regularly sends list of active sources on (C,G)
- Receivers receive announcements on (C,G) and join each (S,G)
Emulating ASM with SSM

- One can emulate ASM with SSMSDP or multissm
  - ssmsdpifier does this

- This works well for most applications

- One problem is that a session needs an owner
  - The one running the controller
  - Somehow make controller dynamic?
  - How to cope with long lasting or permanent sessions with no natural owner?

- ASM useful for some embedded use, service discovery etc
  - Group address can be fixed, not depending on local network addresses
  - It might be enough to do intra-domain ASM
SSM testing

☐ To take part in SSM tests together with 6NET and others you need multicast connectivity. This may require a tunnel to go through routers not supporting IPv6 multicast and a multicast BGP peering.

☐ Forum for collaboration http://www.m6bone.net/
  ○ mailing list, see http://www.m6bone.net/article.php3?id_article=8