Multicast Address Allocation

- **This will probably also be discussed in IDMR, MboneD and DHC.**
  - **four different IETF areas (TSV, RTG, INT & OPS)!**
- **By next IETF we should have sorted out where the right place for this is!**

Overview

- **Multicast address allocation mechanisms must be robust, scale well, be always available, and must have an appropriate API.**
- **Currently we use (overload?) SAP as part of an address allocation mechanism**
  - **very robust**
  - **need a better API for apps other than SDR**
  - **some scaling issues with SAP**

Proposed Solution

Three layer address allocation scheme:

- **An application asks for an address using DHCP.**
- **The allocation server speaks AAP (address announcement protocol - a variant of SAP)**
  - **to discover which addresses are available**
  - **to request/defend the address it chooses.**
- **MBGP routers speak GUM and "MASC"**
  - **to discover which "prefixes" are available**
  - **to request/defend the prefix it (hierarchically) chooses.**
  - **the most local "prefix" is used by an allocation server to choose its multicast address.**

Informed Partitioned Random Multicast Address Allocation (IPRMA)

Suggested by Van Jacobson in 1992

Requirement: to devise a multicast address allocation scheme that is always available, and gives the minimum probability of an address clash.

- **you can only guarantee no address clash if the allocation mechanism is allowed to be unavailable if failures occur.**

General Idea:

Random Allocation

- always available, scales badly due to “birthday effect”

Informed Random

- scales well, but only if the “informed” mechanism works well
- TTL scoping breaks this!

Informed Partitioned Random

- partition address space into ranges for each TTL range
- *need* one range for each TTL ever likely to be used
- address space packing is not good

Adaptive Informed Partitioned Random

- adapt the partitions based on their usage
- several variants - some work better than others
- can work very well in a wide range of conditions

An Alternative

Get rid of TTL scoping based on thresholds for sessions.

Use administrative scoping instead.

Reasons:

- **simplifies many things**
- **improves multicast pruning**
- **allows non-convex scope zones (when used with GUM)**
- **better security policy management for firewalls**
- **makes multicast address allocation more robust.**

With admin scoping only, we can use Informed Random, which is much simpler than AIPRMA.

Why do we need *three* layers?

- **Informed random with admin scoping scales well only if the announcement mechanism guaranteed very low delay.**
- **packet loss means that the mean delay before receiving and announcement is not so low**
- **this can be improved by making announcements frequently at first, and then backing off, but there are still limits.**

IETF rule no. 27:
flat until it stops scaling then add hierarchy

We can solve this scaling issue by using an upper-layer “prefix” allocation mechanism that works on slower timescales than SAP and so is relatively unaffected by packet loss.

This allows (requires) SAP and AAP to be separate, and so SAP announcements can be global while the equivalent AAP announcement is more local.

- hence we can send it more often
- packet loss is on average lower
- these improve the scaling of AAP

Also helps SAP scale because we can add an announcement hierarchy (as first suggested by Ross Finlayson)

And by the way....

We think we need the prefix mechanism to make multicast routing scale too.

What we need:

- **The AIPRMA solution for TTL scoping doesn't work well in the context of an upper-layer dynamic prefix allocation scheme. This means we really need to use admin scoping for this to work.**
- **We need a prefix announcement mechanism that MBGP can use to communicate prefixes to address allocators.**
- **We need a variant on SAP for address allocation (AAP).**
- **We may need a mechanism for address allocators to talk to MBGP routers to inform them to get more prefix.**
  - **or the routers have to listen to AAP (probably undesirable)**

What we need:

- **We need to carefully specify the interactions between DHCP and AAP to allow sessions to be extended, etc.**
- **We need to carefully specify the interactions between AAP and MBGP, particularly with respect to decreasing a prefix size.**
  - **what we'll end up with is likely to be not vanilla informed random allocation to make this easier.**

So, what do you think?

- **Does this sound like a good architecture?**