An Overlay solution to IP-Multicast Address Collision Prevention

Piyush Harsh
IASTED EuroIMSA 2008
March 18, 2008
Why do we need an allocation service?

- Reduce amount of cross-talk between different applications
- Minimize the probability of address clash in the global scope
- Intelligent allocator could result in improved routing in the network.
Any global service architecture proposal should try to incorporate these design goals -

- Deployment on existing infrastructure
- Scalability
- High Availability
- Resilience against DDoS
- Low bandwidth Usage
Existing Solutions

- ‘sdr’ – session directory tool used in MBONE
  - IRMA (Informed Random)
    - Not scalable globally
    - Depends heavily on control message delays and freq.
    - Performance decreases heavily with packet loss rates
  - IPRMA (Informed Partitioned)
    - Uneven utilization of certain partitions
- MASC / BGMP (Prefix / Hierarchical)
- Cyclic (Contiguous Allocation Scheme)
Hybrid Overlay-Multicast Address Allocator (HOMA)

- HOMA is a hierarchical design
- Root level consists of several global TLDs (Top Level Domain)
- Each global TLD acts as the root of hierarchical tree of regional ISPs and enterprise networks
- All sibling nodes form a peer network among themselves.
- This is facilitated through the parent node (provides the peer network details ex. Multicast channel / key etc.)
Each HOMA Node maintains these internal variables independently of others:

- $\alpha$ – address demand trend parameter
- $\beta$ – address release trend parameter
- $\lambda$ – # of new address requested in a given 5 minute slot
- $\mu$ – # of address released in a given 5 minutes slot
- $\gamma$ – address utilization factor
- $\delta$ – additional address anticipated until lease expires
- $\phi$ – possible disposable address count
- $N$ – # of 5 minutes slots until address lease expires
HOMA Address Allocation Algorithm

\[
\begin{align*}
\alpha_{\text{new}} &= \lambda \cdot p + \alpha_{\text{old}}(1 - p) \\
\beta_{\text{new}} &= \mu \cdot p' + \beta_{\text{old}}(1 - p')
\end{align*}
\]

\[N = \text{lease time} - \text{current time} \div 5\]

\[
\delta = [(\alpha - \beta) \times N] - \#\text{free addresses remaining}
\]

Pseudo-code for address allocator module –

If incoming request is for a new channel address by a multicast application –

- If a free channel address is available then allocate the address to the requesting application after negotiating the address lease time properly.
  - Update γ, λ

- If a free channel address is not available, then allocate a channel address randomly from the parent’s address space.
  - Update λ

If incoming request is to release one of the already allotted addresses by a multicast application –

- If the address belongs to the set owned by this HOMA node, then add it to the free address list.
  - Update γ, μ

- If the address does not belong to the address set owned by the HOMA node, do not add to free address list
  - Update μ

At every 5 minutes interval –

- Recompute α, β
- Set \(\lambda = \mu = 0\)
HOMA Address Allocation Algorithm

After every address allocation / de-allocation check the value of updated $\gamma$.

- If $\gamma < \text{threshold}$: Do nothing.
- If $\gamma \geq \text{threshold}$
  - Compute the anticipated additional address required $\delta$.
  - If $\delta > 0$, initiate a request for $\delta$ number of addresses on the sibling peer network and wait for 2 minutes for responses.
    - If any response comes, add addresses to the free address pool keeping track of the lease associated with those addresses.
    - If no response comes, initiate additional address request to parent HOMA node.

If additional address request is received on the sibling peer network –

- Compute possible disposable address count $\phi$ using the following relation:

  $\Phi = \#\text{free_addresses_remaining} - [(\alpha - \beta) \times N]$

  - If $\phi > 0$, indicate willingness to allocate $\phi$ set of addresses to the sibling node. Treat this allocation just like any other address allocation.
  - If $\phi \leq 0$, then do nothing.
Let $\pi$ be the probability that additional address demand is satisfied by one or more sibling nodes.

Worst case scenario: node must wait for 2 minutes before sending the request to its parent node.

If tree depth is ‘d’, then overall delay could be modeled by a recursive equation:

$$\text{Delay} = 2\pi + (2 + \Lambda_d)(1 - \pi)$$

where $\Lambda_d$ is the delay when the request is made to one's parent.

$$\Lambda_d = 2\pi + (2 + \Lambda_{d-1})(1 - \pi)$$

The value $\pi$ is experimentally determined.
Possible Advantages of HOMA

- Could minimize routing flux because of its hierarchical structure.
- Possibly better address space utilization compared to MASC / BGMP scheme
- Lot better delay characteristics compared to MASC / BGMP which has a 48 hours observation window for address set claim.
- TLDs are well known hosts and their immediate child ISP nodes are also well known, this could be used to prevent DDoS attacks at the top levels
- Algorithm implementable in layer 5, easily deployable on existing infrastructure (possibly as a router OS patch)
References

- Mark Handley – “Session Directories and Scalable Multicast Address Allocation”, SIGCOMM ’98
- Van Jacobson – “Multimedia Conferencing on the Internet”, SIGCOMM ’94
- Satish Kumar, Pavlin Rodoslavov et al. – “The MASC/BGMP Architecture for Inter-domain Multicast Routing”, SIGCOMM ’98