802.1Qbp – ECMP
Multicast Load Spreading

Ben Mack-Crane
(ben.mackcrane@huawei.com)
Observations on Multicast ECMP

• Multicast traffic cannot use the same load spreading mechanism used for unicast traffic
  – FDB has multiple forwarding ports (cannot select just one)
  – Random selection & replication can lead to duplication & loops

• ECMP for unicast traffic makes congruence (unicast-multicast and bi-directional) either easy or impractical (depending on how the definition is adjusted)
  – In either case congruence is not a concern in ECMP path calculations

• Multicast traffic must be constrained to a tree (to avoid loops and duplicate frames)
  – However, different multicast addresses may use different trees
Spreading Multicast Traffic

• In SPBM each service instance (I-SID) has its own set of group addresses used to carry client multicast/broadcast traffic
  – Group addresses composed from SPSourceID & I-SID
  – # multicast flows = #service instances * #edge nodes
• Assign each flow to an ECT using a standard hash algorithm
  – so all nodes will agree on assignment and produce consistent forwarding state
• Each multicast flow can be independently assigned to an ECT
  – Potentially large calculation (random tree per I-SID)
One Approach

- Select “random” tree from ECT set for each root node
  - Select from all ECTs, not just those selected by std tie-breakers
- Use this tree for all flows from that node
  - All I-SID multicast from root node use same tree
  - But I-SIDs can have varied endpoints, so still some spreading
- Use hash (e.g., FNV) to select one “parent” from set of equal cost parents calculated for unicast ECMP
  - Modest addition to route calculation
  - Include root node MAC address in hash to create variation
- Tried this out in an SPB simulator…
Unicast SPB, e.g. between 26 and 32

SPB selects a single path using an ECT tie-breaking function.
Unicast ECMP, e.g. between 26 and 32

ECMP load spreading utilizes all links on equal cost paths for unicast traffic.
SPB Multicast Tree, e.g. I-SID 255 from 26

Multicast selects links from one equal cost tree using ECT tie-breaker.
Multicast load spreading selects links from all equal cost paths using a hash function (in this case FNV).
#define C1AQ_SYSTHASH_PARENT(result, syst, r, n)
{
    register tUINT32 hash = 0x811C9DC5;
    register tUINT64 fodder;
    register tUINT32 fnvPrime = 0x01000193;
    register tUINT32 best = 0;
    register int k, m, np = syst->node[n].np;
    for (m=0; m<np; m++)
    {
        fodder = syst->node[r].sysIdMac[0];
        for (k=0; k<7; k++)
        {
            hash = hash ^ (fodder & 0x000000ff);
            hash = hash * fnvPrime;
            fodder = fodder >> 8;
        }
        fodder = syst->node[syst->node[n].parent[m]].sysIdMac[0];
        for (k=0; k<7; k++)
        {
            hash = hash ^ (fodder & 0x000000ff);
            hash = hash * fnvPrime;
            fodder = fodder >> 8;
        }
        if (hash > best)
        {
            best = hash;
            result = m;
        }
    }
    result = (m==0 ? -1 : syst->node[n].parent[result]);
}"}
All SPB Multicast Trees, e.g. I-SID 255

Set of multicast trees are congruent.
Multicast load spreading selects links from all equal cost paths using a hash function (in this case FNV). Different trees are selected for each root by including root MAC address in hash.
Observations on this Approach

- Spreads multicast traffic and unicast traffic using common route calculation (all ECMP).
- Multicast spreading using a standard hash (pseudo-random).
- No selection or configuration of tie-breaker needed!
- Propose further study of spreading performance and selection of a standard hash algorithm for use in multicast route calculation.