Embedding Priorities in Addresses is Problematic

Keith McCloghrie
Cisco Systems

Overview

Two issues:

- **Address fields in the MAC header are intended to hold addresses.**
- **Having the specification limit the number of priorities to two is too restrictive.**

Address fields should only contain addresses

- Burying a priority indication inside an address is architecturally questionable.
- Today's bridges/switches:

  —won't know that priority is not part of the address,

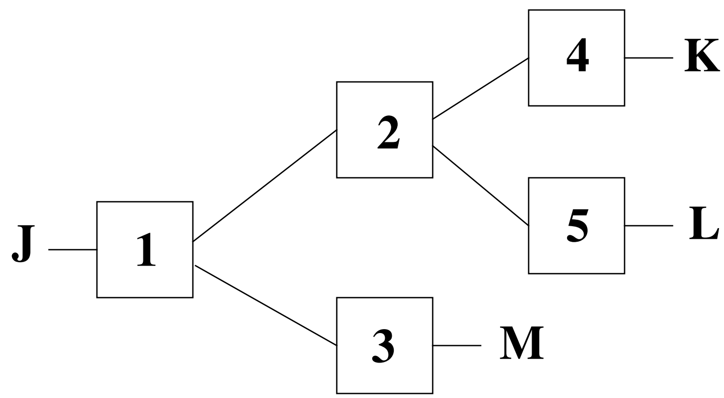
  —will think that different priorities are different addresses,

  —will learn/age each priority independently.
- 802.1D relies on:

  —source address learning to avoid unicast flooding.
- source address learning relies on:

  —bi-directional exchange of (unicast) traffic between a pair of addresses.

Bi-directional unicast traffic is needed



Four end-systems: J, K, L, M,
Five of today's bridges: 1, 2, 3, 4, 5,
Addresses for priorities: J1,...,Jn, K1,...,Kn

1. Priority-n traffic from J to K: .... DA=Kn, SA=J1
To learn Kn, need other traffic: .... DA=J1, SA=Kn

==> have to generate non-priority traffic (for each Kn in use)

Using Priority Addresses in Source

2. Priority-n traffic from J to K: .... DA=K1, SA=Jn
To learn K1, need other traffic: .... DA=J1, SA=K1
To learn J1, need other traffic: .... DA=K1, SA=J1

==> need to generate non-priority traffic

3. Priority-n traffic from J to K: .... DA=K1, SA=Jn
To learn K1, need other traffic: .... DA=Jn, SA=K1

==> need to generate non-priority traffic

Using Priority Addresses in Source & Dest.

4. Priority-n traffic from J to K: .... DA=Kn, SA=Jn
To learn Kn, need other traffic: .... DA=Jn, SA=Kn

==> for each priority at which traffic is sent, it must be sent in both directions, i.e., priority traffic must be bi-directional.

The I/G bit ?

- **So, confusing MAC address semantics with priorities is problematic - what about the I/G bit ?**

- **The architecturally semantics of the I/G bit are different from the other address bits, but:**
  - **—Can’t use Destination I/G bit (for priority in unicast addresses), since it indicates multicast.**
  - **—802.1D says: don’t learn from frames with I/G bit set in source address, so can’t use Source I/G bit.**

Other Implications

- **Forwarding Table size:**
  - **—With each priority for each station being learned as a separate address, the size of the forwarding table increases.**

- **Additional flooding:**
  - **—Would like to learn addresses for all priorities from the first packet, but**
  - **—with bridges treating each priority's address independently, only one address can be learned from the first packet;**
  - **—the addresses for other priorities must be learned from later packets.**
  - **—Thus, the first packets at each priority will be flooded.**

Conclusion

Having different addresses for different priorities is architecturally questionable:

- **To avoid flooding with today's bridges/switches:**
  - **—there must be approriate traffic in the other direction; if none, then perhaps bogus messages should be generated, or**
  - **—applications must generate bi-directional traffic at each priority level.**
- **Increases in forwarding table size and flooding.**

Also note that it doesn't help today's bridges to add (to 802.1p) an explicit definition of priority bit(s) within addresses.

How Many Priorities ?

However, 802.1p is right on the number of priorities:

- **At least two priorities are needed:**
  - **—"real-time",**
  - **—"best-effort".**
- **Other possible needs:**
  - **—network/layer management.**
  - **—IETF's Integrated Services has multiple classes of "real-time".**
- **802.1p's combination of allowed/required is good:**
  - **—up to 8 priorities are defined,**
  - **—implementation of 2 is required.**

Recommendation

Choose a mechanism for carrying priority which:

- **allows for more than two priorities, and**
- **doesn’t embed priority within the MAC header’s address fields.**