Some Notes on Multicast Scaling and PIM

by Van Jacobson

(with apologies to Deborah Estrin
for mangling her text and ideas)

30th IETF
Toronto, Canada
25 July 1994

Preliminaries

- We want to scale to a very large number of multicast groups and sources — O(Internet Hosts).

- No new limits on traffic. I.e., ultimate traffic limits same as for unicast: link sharing policy and bandwidth.

- Want to control both the amount of multicast state in routers and the link bandwidth used by multicast routing traffic. (We are slightly more concerned about bandwidth since memory is cheap.)

There are two ways of doing multicast distribution:

- Send the data everywhere; sites that don’t want it prune themselves.

- Sites announce what groups they want to receive; data sent only where wanted.

(These roughly correspond to PIM ‘dense mode’ and ‘sparse mode’.)

Neither way is ‘right’ for all groups and all topologies.

The two styles have very different characteristics:

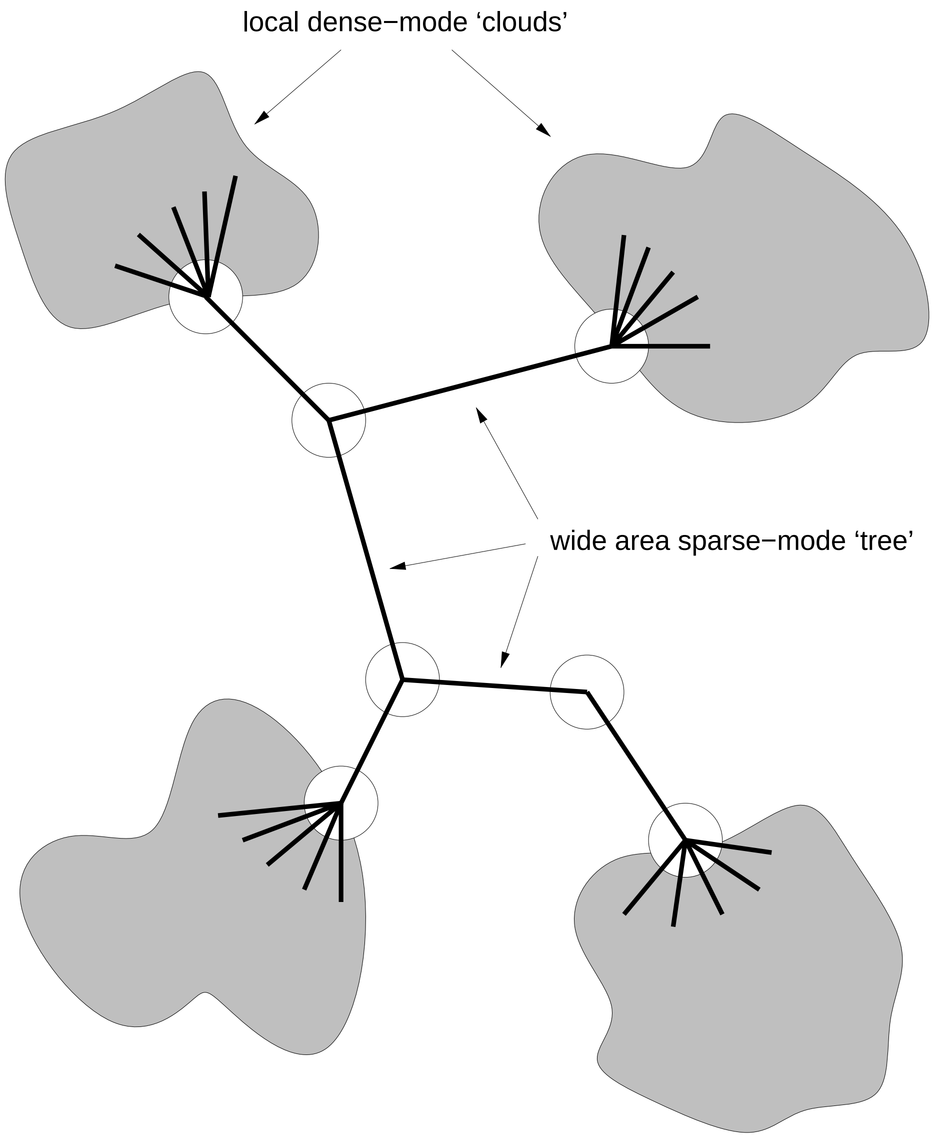
	Dense Mode	**Sparse Mode**
Distribution tree	Implicit (from unicast routes)	Explicit
Prune state	Explicit	None
State created	by data traffic, on demand	by receivers, periodically

And very different state memory and control traffic scaling properties:

	Dense Mode	**Sparse Mode**
Distribution tree	None	$O(G)$ to $O(G \times S)$
Pruning	$O(G \times S)$	None

Where G is the number of ‘active’ groups and S is the number of ‘active’ sources. Note that the $O(G)$ sparse mode scaling holds for CBT and RP-tree PIM. The $O(G \times S)$ scaling is for shortest-path-tree PIM.

Basic model: Local campus/site/domain is dense mode with wide-area, sparse-mode backbone providing glue for interdomain groups:



Basic Model's Scaling Properties

- State and outbound join traffic at border router of each domain scales $O(G_{local})$ (number of inter-domain groups with local members).

- Inbound join traffic scales $O(S_{local})$ (number of local sources sending to interdomain groups).

- Can switch from RP-tree to shortest-path tree triggered by data traffic intensity from some source to get better distribution trees at cost of slightly more state in backbone *while that source is active*.

$O(G_{local})$ scaling, combined with fact that inbound link bandwidth effectively limits number of external groups domain can usefully subscribe to, implies per-domain scaling is independent of size of Internet.

Basic Model's Scaling Properties (cont.)

There's still an $O(G)$ scaling problem in the backbone: Backbone routers see $\bigcup G_l$ joins which results in $O(G)$ state and join traffic.

Can get rid of interior state by assuming that border routers will do the right thing and just do (stateless) reverse-path forwarding check on data traffic to prevent loops.

- This can cause traffic to be delivered to places that don't want it but they will correctly discard it based on their local join state.

- This implies that we are able to do RPF check on traffic sent via an RP (encapsulation or IP option or ???).

- Join's have to be forwarded upstream so proper state gets instantiated at boundaries. So there's still a scaling problem from join traffic.

Bounding Control Traffic

Basic scheme for bounding the amount of control traffic is to say that there is an administratively set, per-link limit on the total bandwidth that can be used for PIM control traffic (joins, prunes, prune probes, etc.). E.g., if the limit is 1% of a 2 Mb/s link, there are 200 groups with local members, and the average join message is 64 bytes, the message rate could be at most:

$2000000 * 0.01/(200 * 64 * 8) =$ .2msgs/s per group

or 5 sec. between each group's join messages.

- This implies that timeouts are per-link, not global.

- Can get better responsiveness by dividing control bandwidth non-uniformly. E.g., half to recently active groups and half to inactive groups.

Next iteration on Basic Model:

- Edges keep state on everything they're interested in and send appropriate joins.
- Center keeps as much state as it can and on overload discards state for groups that have been least recently active.
- Control bandwidth divided such that recently active groups get higher rates.
- Bandwidth limit on Edge → Center links set such that sum matches limit on Center → Center links (so that Center can just pass through joins for inactive groups and not keep the state necessary to rate limit them).