**Multicast OSPF**

- **Multicast using link-state based protocol.**
  - Simple augmentation of OSPF. Used for Intra-AS multicast.

- **Edge routers flood mcast group membership information**
  - Similar to LSP flooding.
  - LS database now has group membership info.

- **Locally build source-based, shortest-path mcast tree on each router.**
  - Build tree on-demand (on first packet arrival) for a mcast group to avoid tree calculations for all known groups.
  - Use caching to avoid tree calculation for every packet.
  - Note the mcast trees are already pruned.
  - Data packets are forwarded along tree branches.

---

**MOSPF Performance**

- **Membership information is flooded.**
  - High overhead if group membership changes frequently.
  - Not scalable. Certain divide-and-rule techniques have been suggested.

- **High computational overhead on routers**
  - Need to compute shortest path trees for every active <source,group> after each update on LS database. Cost $O(N^2)$ per source with Dijkstra.

- **Must still maintain forwarding table on a per <source,group> basis.**
Core-Based Tree (CBT)

• **Motivation:**
  – We don’t like maintaining any routing state on a per `<source,group>` basis. Just per group is nice.
  – We don’t like any form of flooding either for group membership (MOSPF) or to trigger pruning (DVMRP).
  – We don’t like non-tree routers doing very much work, if any.

• **Idea:**
  – Build a “shared tree” for the group (same as spanning tree). NOT a source based per source!
  – Tree starts with a “core” router for the group.
  – Use a join/leave protocol to “grow” or “shrink” the tree.