Application Level Multicasting

Using SplitStream

Jon Ludwig – jal2656@rit.edu
Brad Israel – bdi8241@rit.edu

Motivation

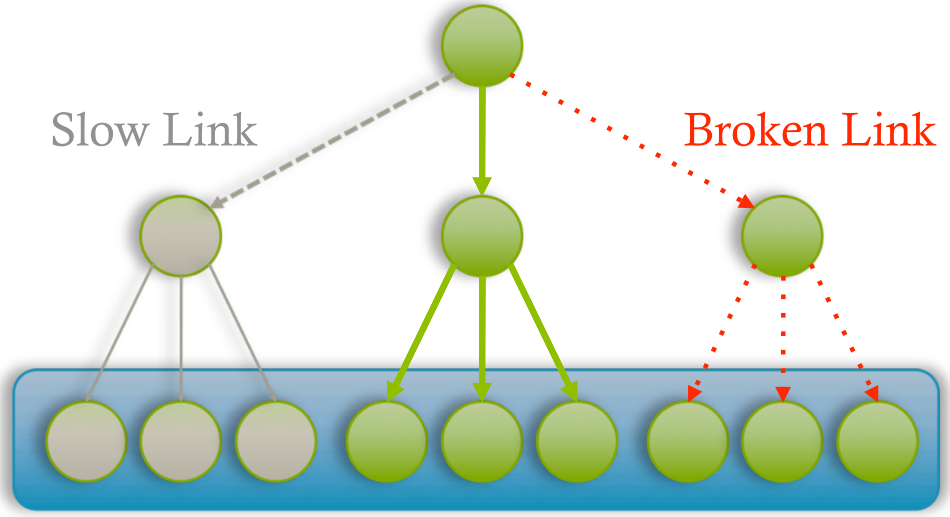
- Why multicast?
  - Streaming multimedia
- Why application level?
  - End-to-end argument
  - Deployment issues
- Why SplitStream?
  - Scalable
  - Fair

Multicast Issues

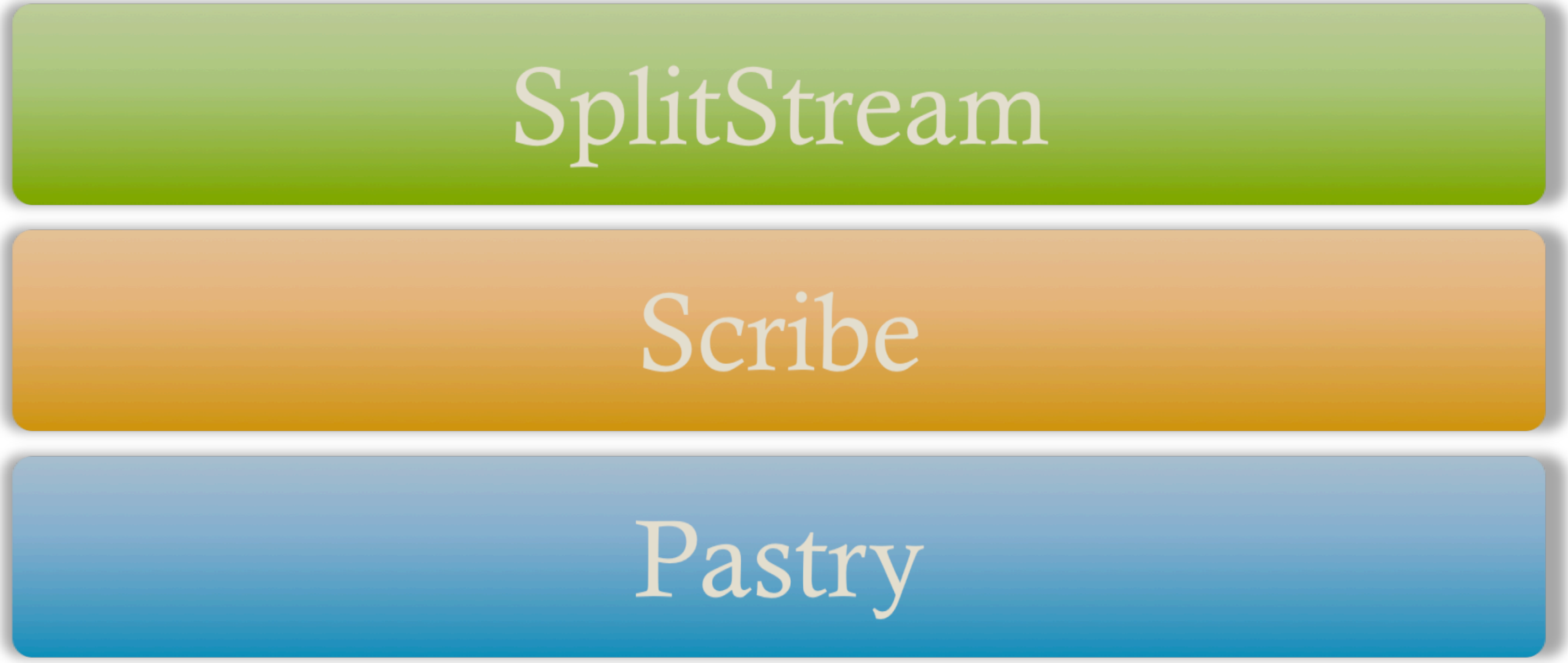
- Scalable
  - Built on DHT for O(log n) performance
  - Uses multicast trees
- Fair
  - Balances forwarding load across all peers
- Low Delay
  - O(log n) hops from source to destination
  - Routing table entries are chosen based on low network delay

Fairness & Performance with Multicast Trees

- Slow or broken links can impact entire sub-trees of the multicast tree
- High churn can disable multicast with frequent tree reconstructions
- The interior nodes of the tree carry all the burden of forwarding packets



Lucky!

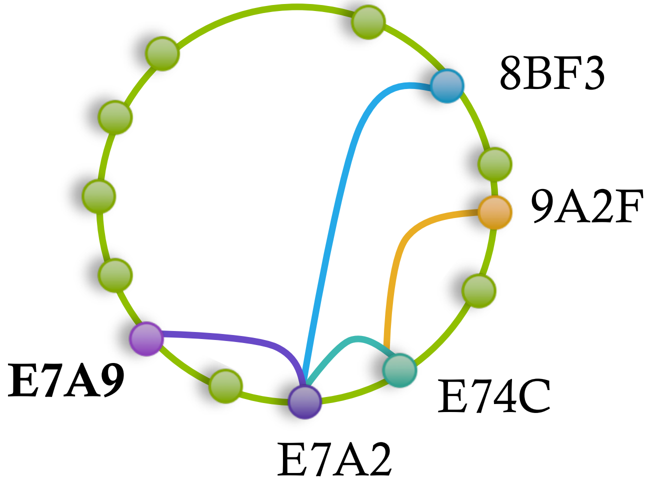


SplitStream

Scribe

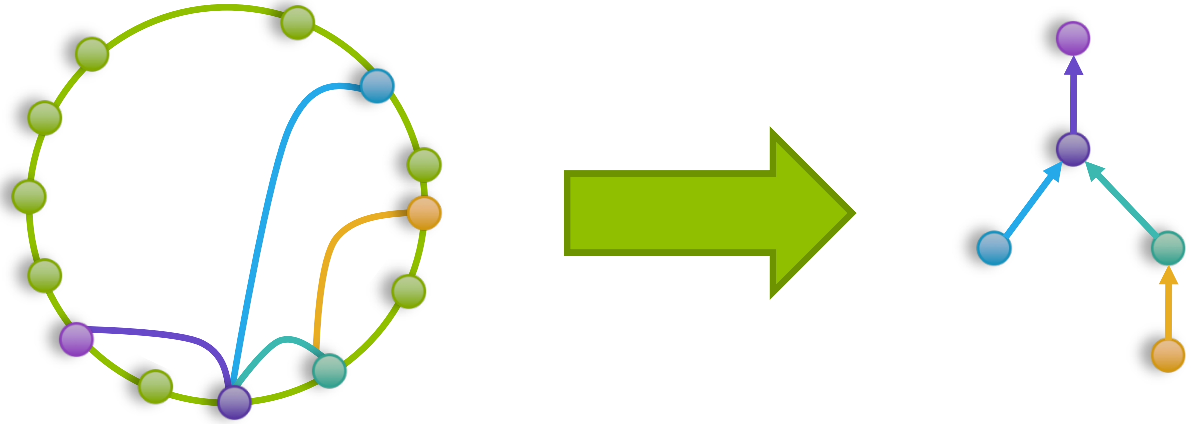
Pastry

Pastry



Scribe

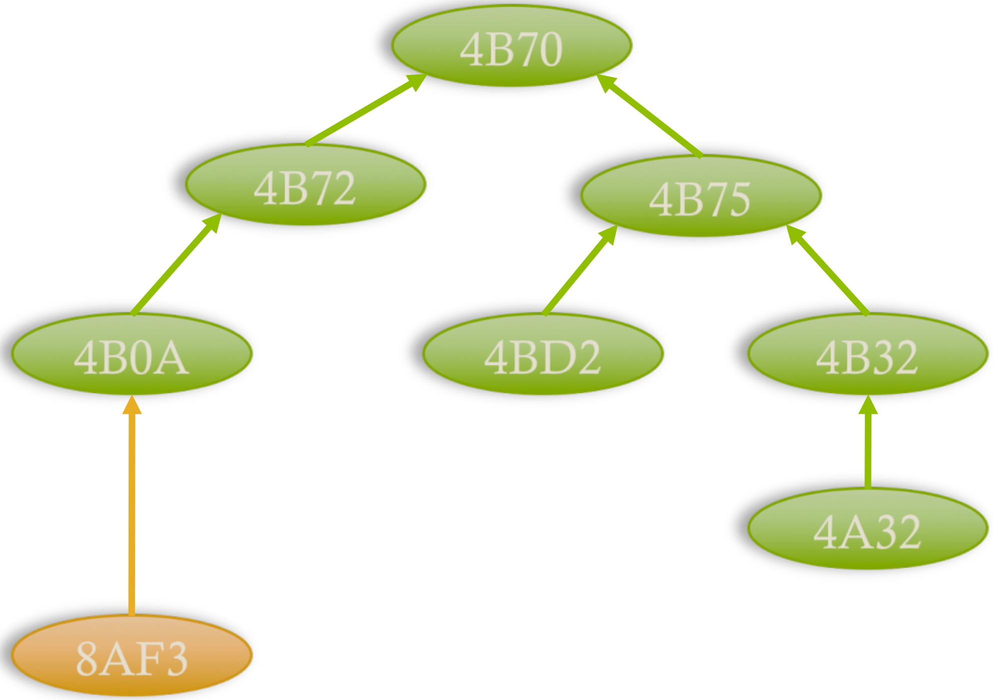
- A node routes a JOIN message to the groupId
  - Nodes along the route add the source as a child for that group and forward the message to the next node
- A node routes a MULTICAST message to the groupId
  - The tree root multicasts this message through the tree



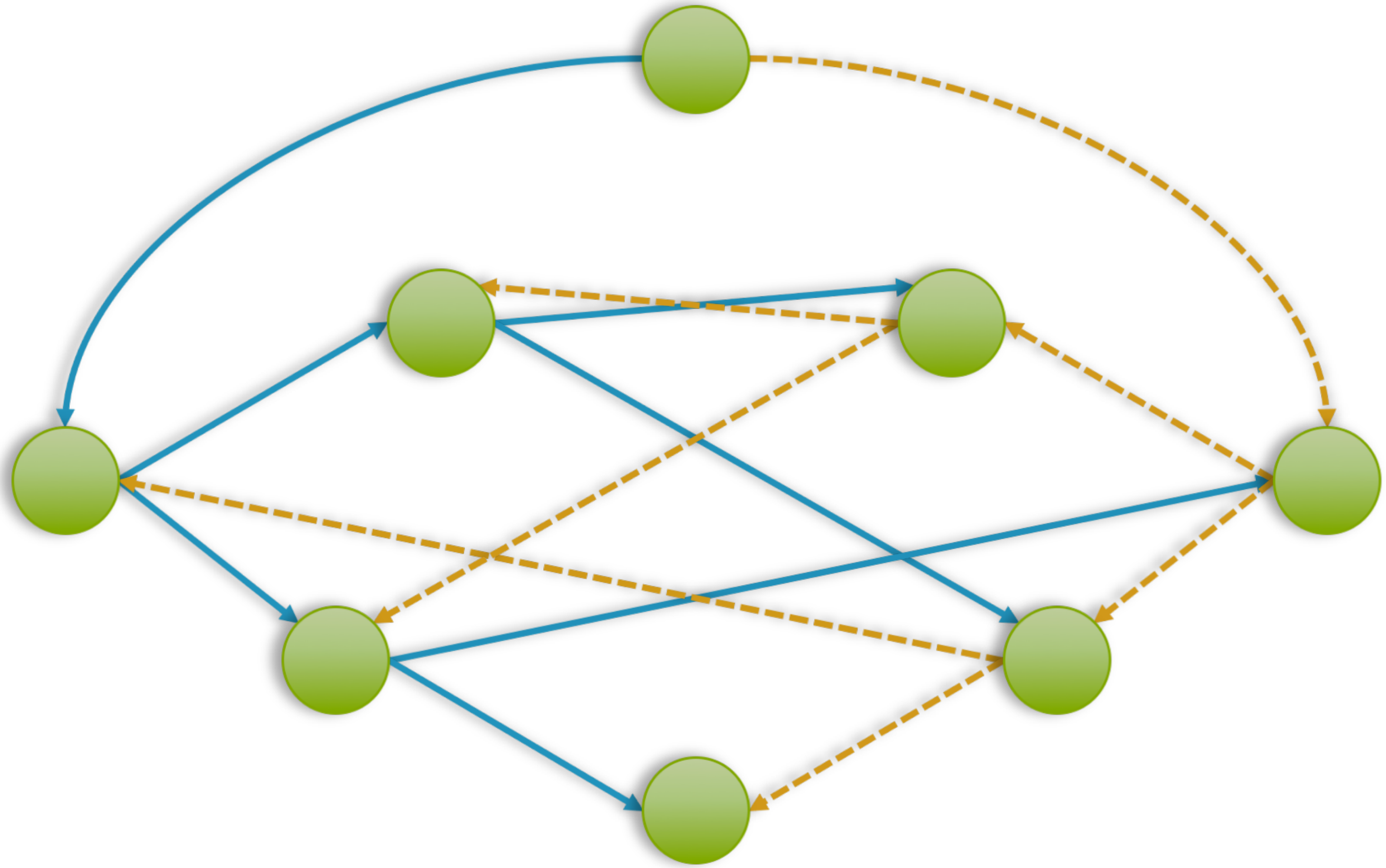
SplitStream

- Split data stream into *k* stripes
- Construct a multicast tree for each stripe
- Ensure that each node is an interior node of only one multicast tree
  - groupId for each tree has a different MSB
  - Nodes with MSB *k* are only interior nodes for groupId *k*

SplitStream



No node is an interior node of both trees



Project

- Implement multimedia streaming in a SplitStream environment
- Multicast a playlist of media to subscribers
- Visualize and monitor multicast trees
- Test SplitStream performance and fault tolerance
    - Compare SplitStream to Scribe

Questions?