Video Streaming with Peer-to-Peer Multicast

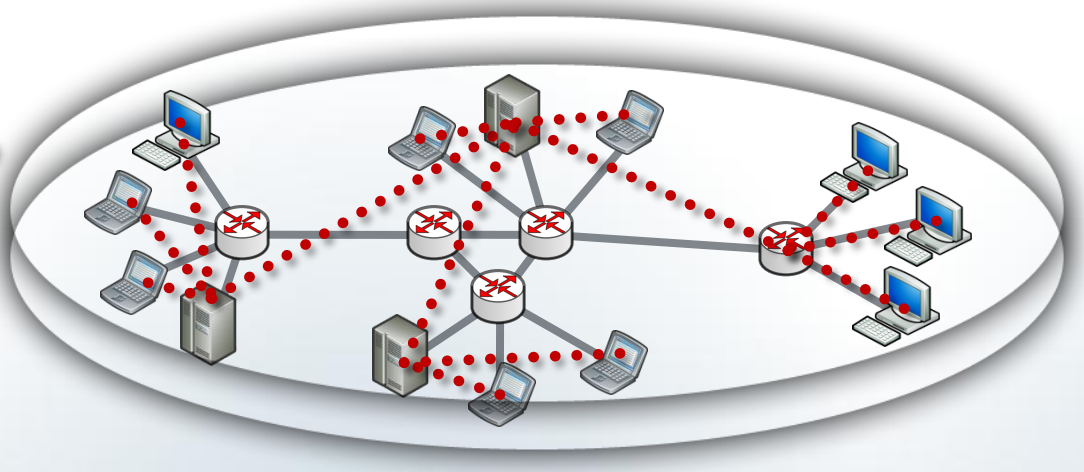
Alex Bikfalvi

P2P Overlays

Peer connections between all users that participate in the network

Application Layer

Physical Layer



- For streaming, the overlay is usually a tree rooted at the source
- The nodes of the tree are the peers rather than the routers
- The challenge is building a reliable multicast tree
- Usually multiple overlay trees are used

Issues and Requirements

Issues

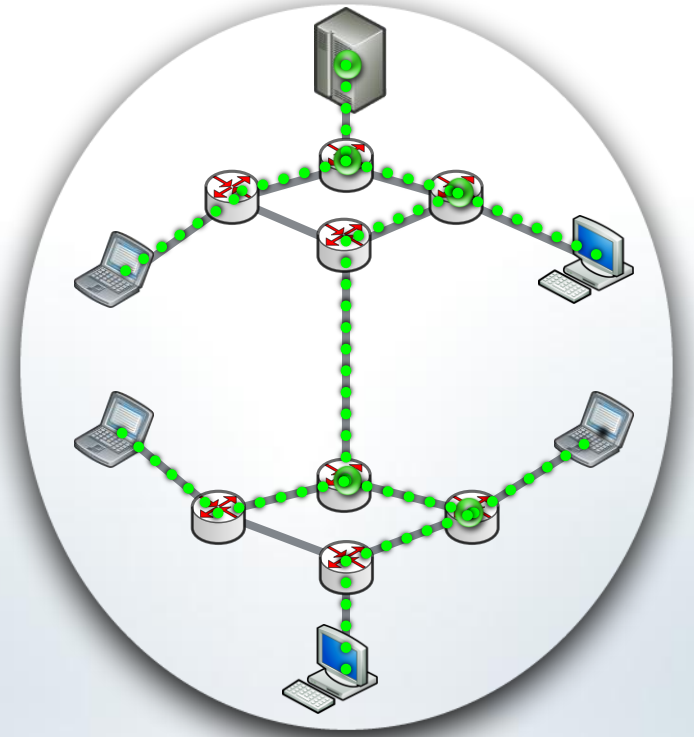
- Peer behavior is unpredictable, they can join and leave at any time
- The users may experience a high delay, since the traffic is routed through several peers
- Requires complex protocol exchanges between peers in order to maintain a reliable overlay fabric

Requirements

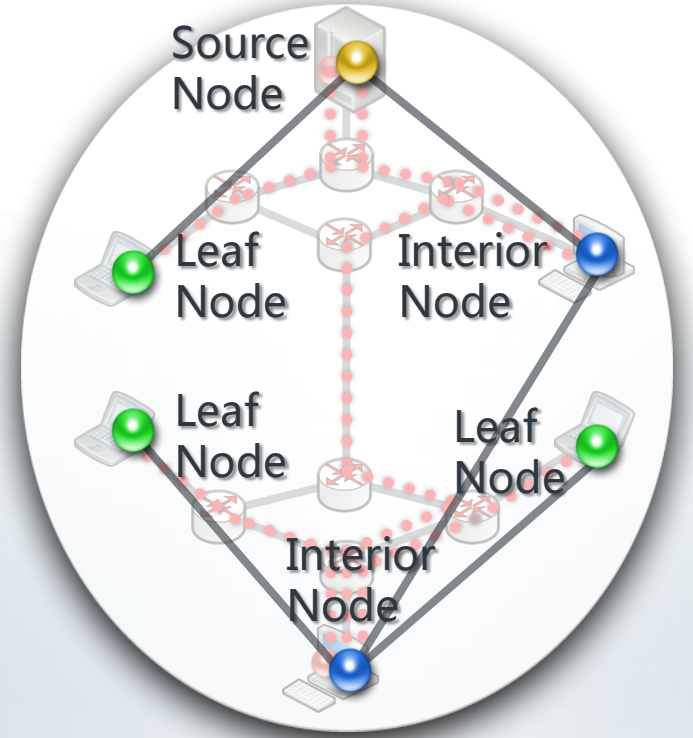
- NAT traversal, especially for a service that targets home users
- Consider the network heterogeneity
- Maintain a quality of experience since QoS is not usually considered in P2P
- Real time transmission, especially in communication services

Multicast Trees

IP Multicast



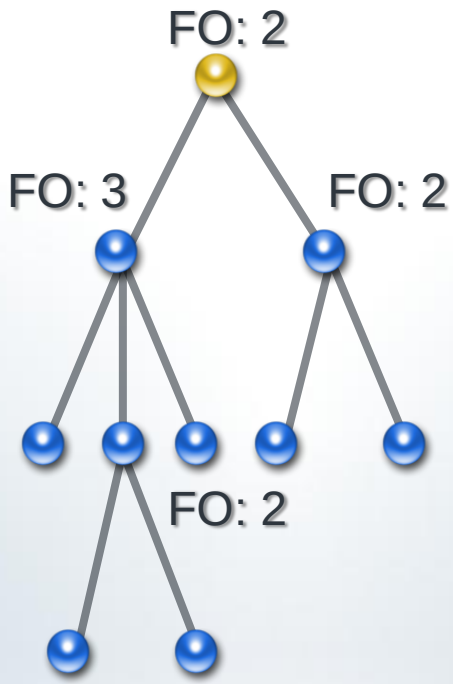
P2P Multicast



- Routing and packet replication shifted to the network edge
- Same traffic traverses some links several times
- Optimized tree construction is very important
- Paramount criteria: parent selection and loop avoidance

Multicast Trees

Depth, Spread and Fan-Out

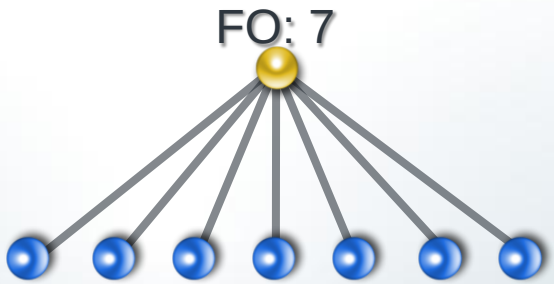


Depth: 3

Maximum Depth

Maximum Latency

Maximum Spread



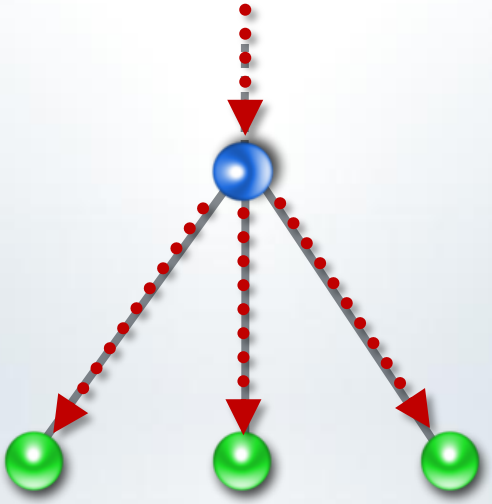
Maximum Source Bandwidth

- Depth: the number of levels in the tree
- Spread: ratio between leaf and interior nodes
- Fan-out: number of children for interior nodes

Multicast Trees

Interior nodes and their fan-out are important

Overlay View



Physical View

- Uplink required bandwidth can be higher than for downlink
- Downstream peer experience depends on upstream peers

Multicast Trees

Balanced Trees

Interior Nodes

$$\frac{f^d - 1}{f - 1}$$

Leaf Nodes

$$f^d$$

Small Fan-Out

Fan-Out of 2

Around **50%** Interior Nodes

Large Fan-Out

Fan-Out of 100

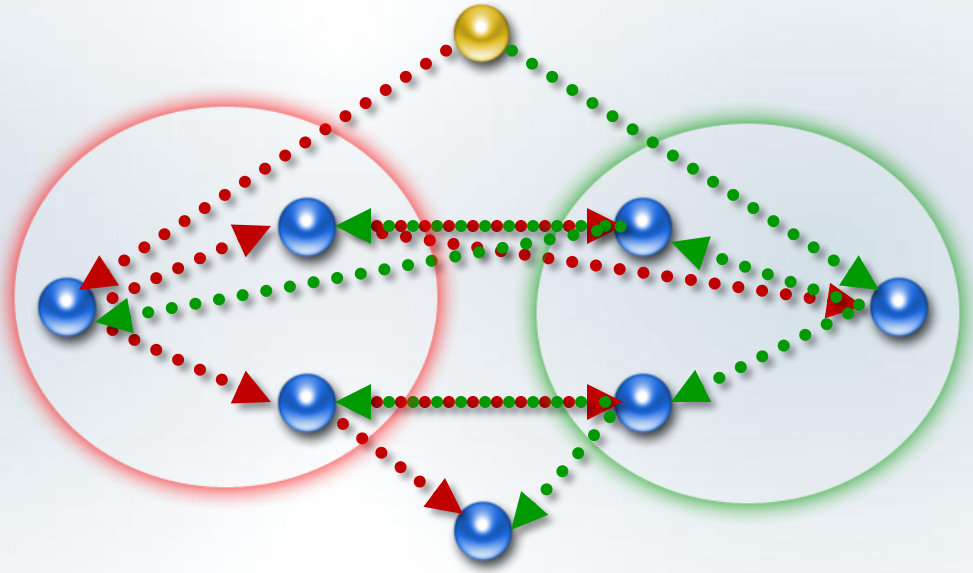
Around **1%** Interior Nodes

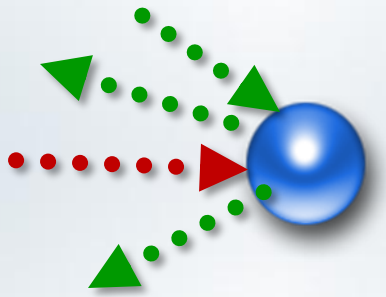
SplitStream

Introduction

Divides the video stream in serveral substreams called stripes

- Use a different multicast tree for each stripe
- Distribute uniformly the peer as interior nodes in each tree
- Ideally, interior nodes in one tree are leaf nodes in all others
- The main tree where a node is interior is called proper tree





Uplink = Downlink

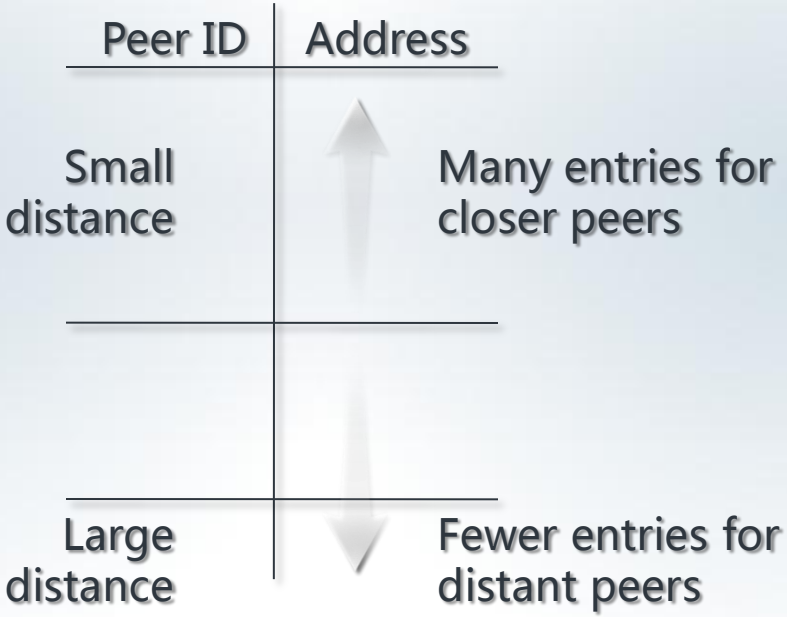
- Downlink-uplink ratio controlled by stripes-fan-out ratio

Routing

Use a DHT-based P2P protocol: Pastry and Stripe

- Assign to each peer an identifier within a range (hash space)
- Distance between two peers is the difference between IDs

Routing Table

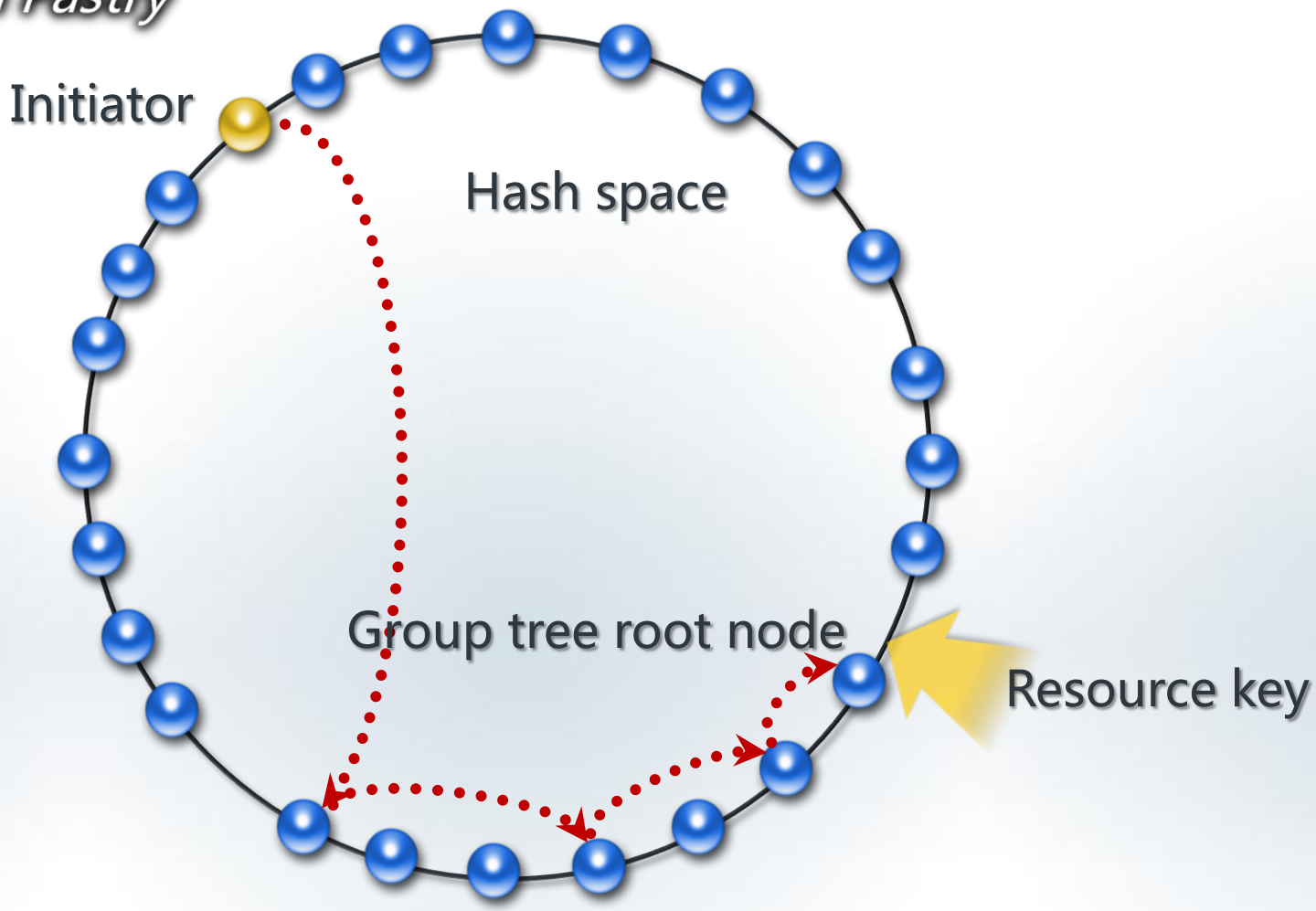


The Idea

- Assign to resources key identifiers in the hash space
- Publish the resources at peers closest to the keys
- Locate those peers using DHT routing

Routing

Routing in Pastry



Multicast Groups

The resources in P2P multicast are the multicast groups

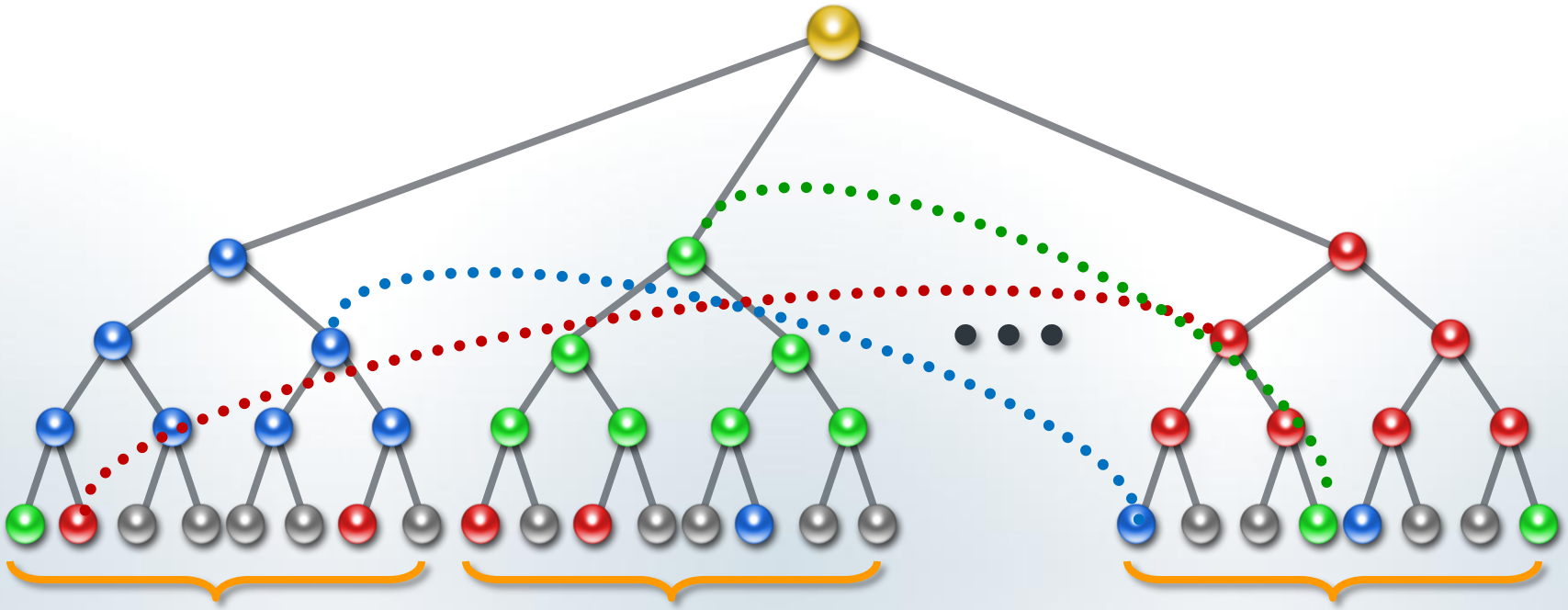
- Groups are assigned unique hash keys
- The node closer to each group key is the root for that group

How to create interior-node disjoint multicast trees?

- Create group keys different in most significant digits
- A group interior node should have an ID with same significant digits

Multicast Groups

Example: Hash space of 4 hexadecimal digits (16 bits) with 16 groups



Group: 0xxx Group: 1xxx Group: Fxxx

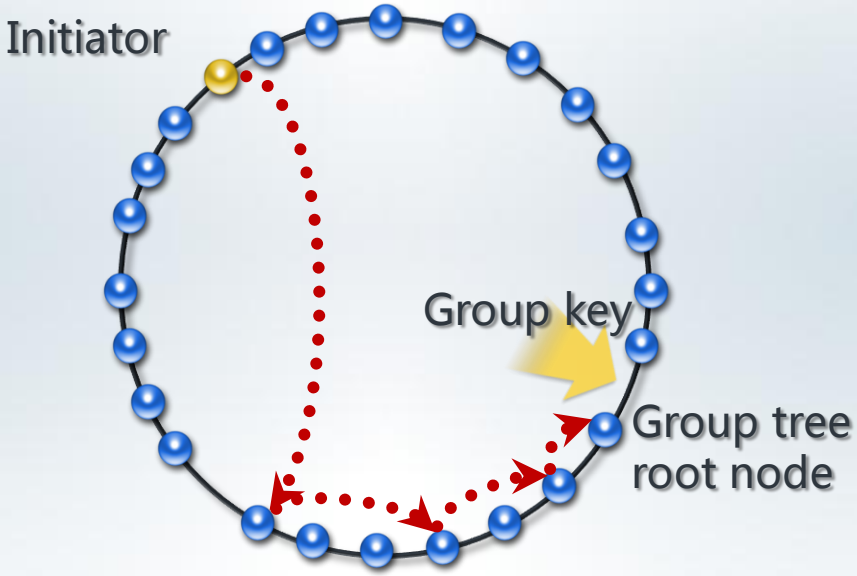
- Peers with node ID: 0000 to 0FFF
- Peers with node ID: 1000 to 1FFF
- Peers with node ID: F000 to FFFF
- Peers with node ID: 2000 to EFFF

Joining and Rejection

How to join a group?

- Lookup the group key
- The routing should stop at first peer along the route that belongs to the group

What happens when a node inside a group is found?

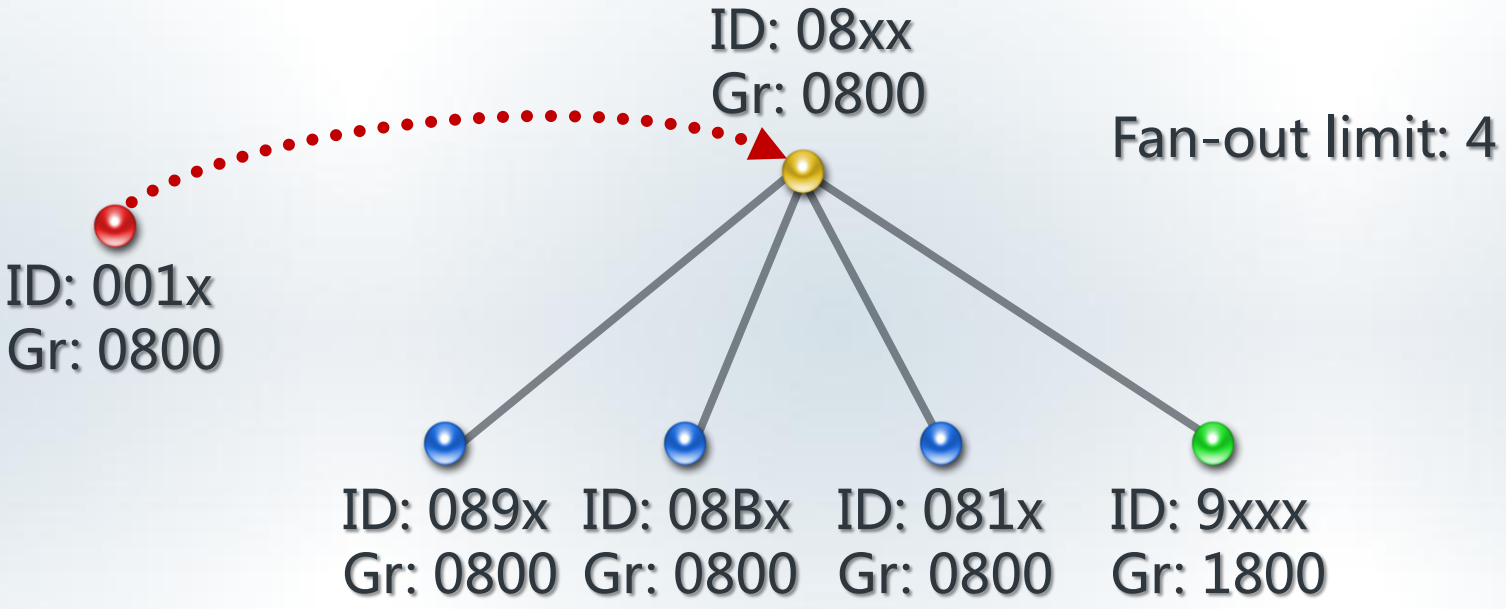


- If the fan-out limit of the candidate parent is not reached, adopt the new node
- Otherwise... reject one node

Joining and Rejection

What node to reject?

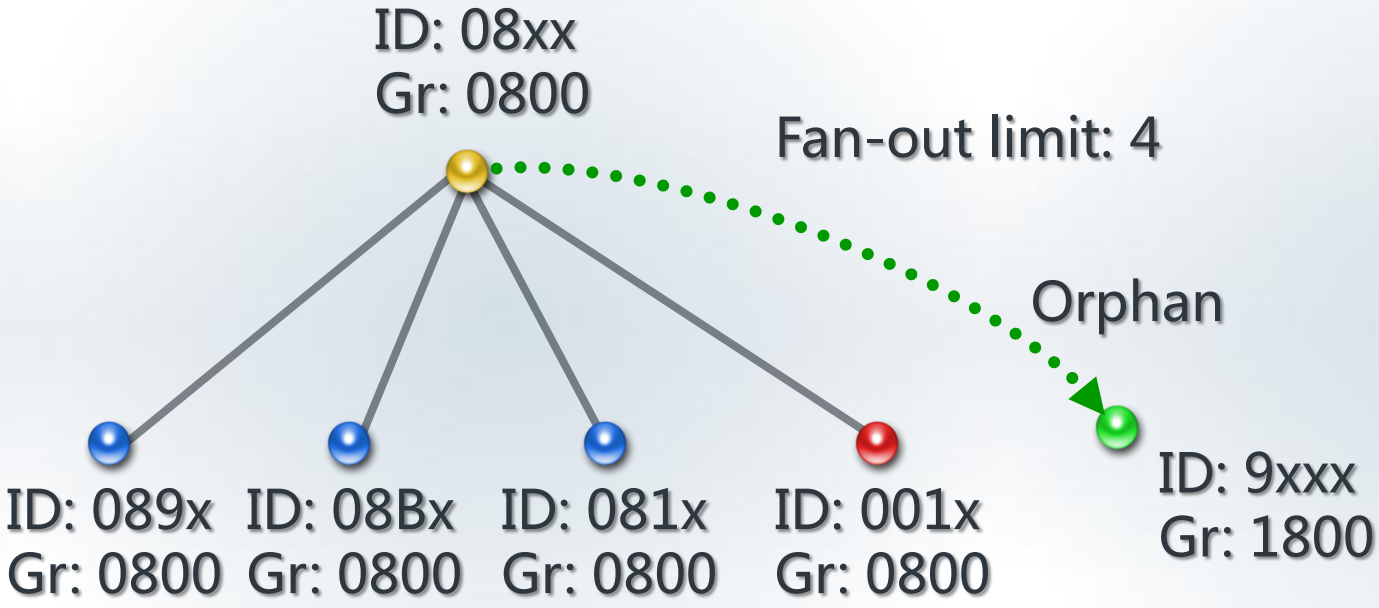
- Match criteria in the following order:
  - Children belonging to other groups
  - Children belonging to the same group and having the greatest distance to the group key



Joining and Rejection

What node to reject?

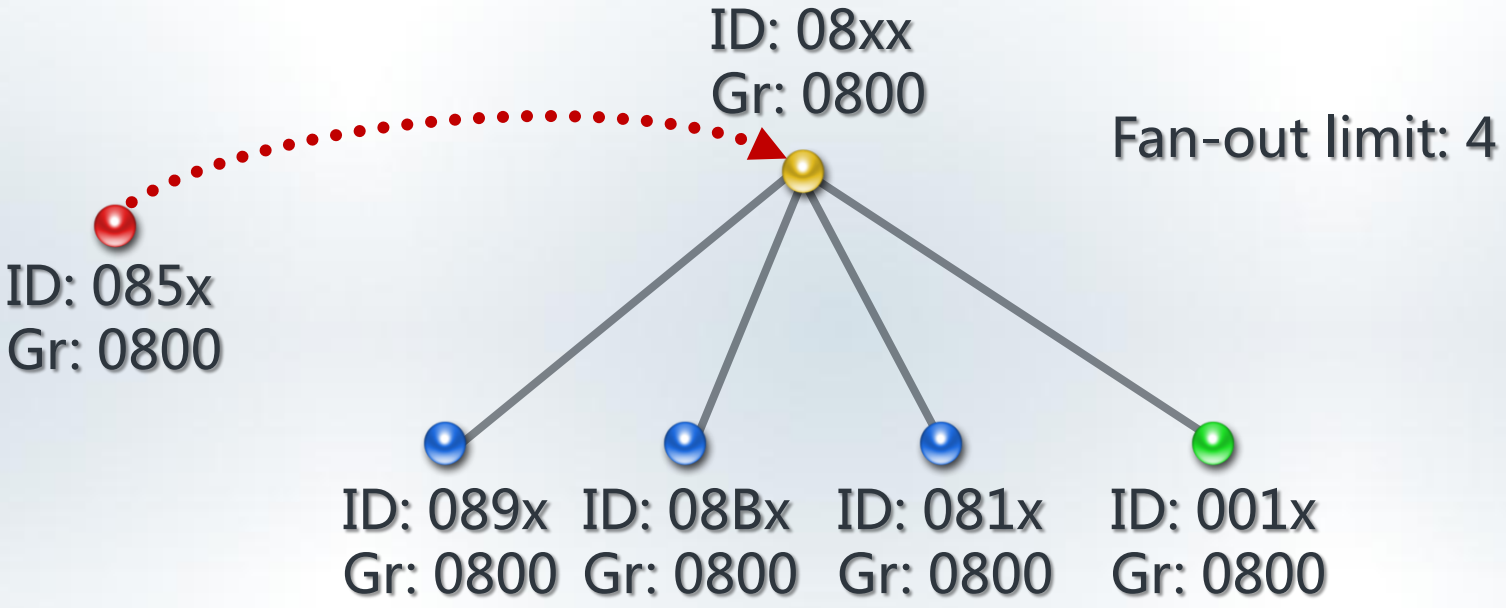
- Match criteria in the following order:
  - Children belonging to other groups
  - Children belonging to the same group and having the greatest distance to the group key



Joining and Rejection

What node to reject?

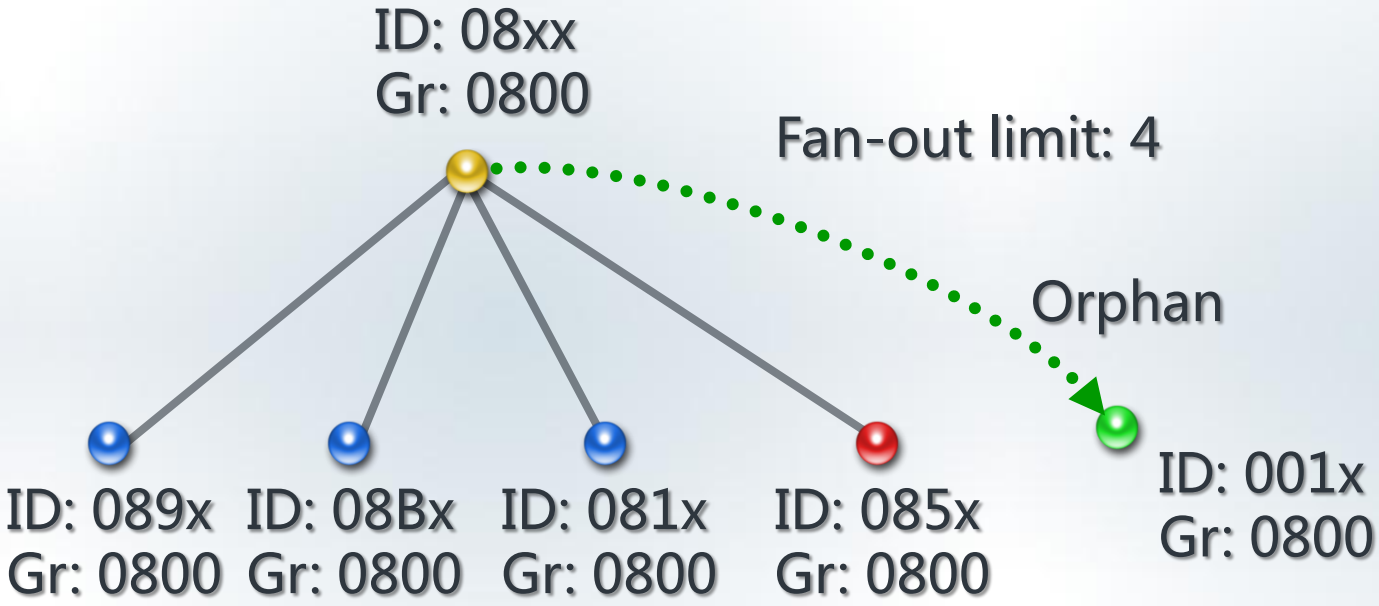
- Match criteria in the following order:
  - Children belonging to other groups
  - Children belonging to the same group and having the greatest distance to the group key



Joining and Rejection

What node to reject?

- Match criteria in the following order:
  - Children belonging to other groups
  - Children belonging to the same group and having the greatest distance to the group key



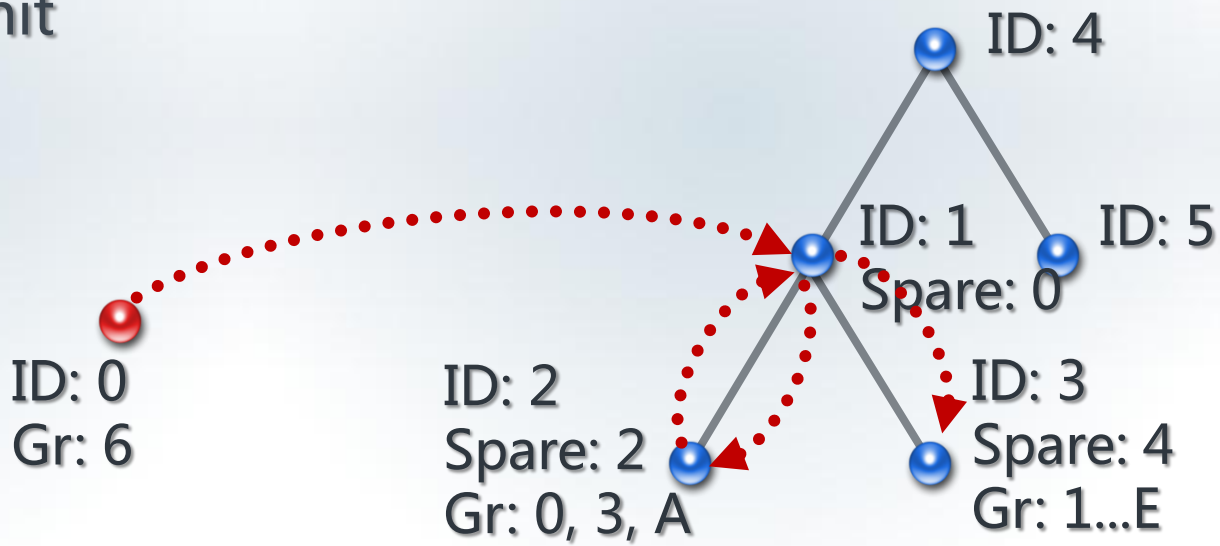
Joining and Rejection

What do orphans do?

- Search for another parent
- If none is found, search for parent in the spare capacity tree
- If none is found... bad luck

Spare Capacity Tree

- A tree with all nodes that have not reached their fan-out limit



P2PCast

Introduction

Same principles like in SplitStream

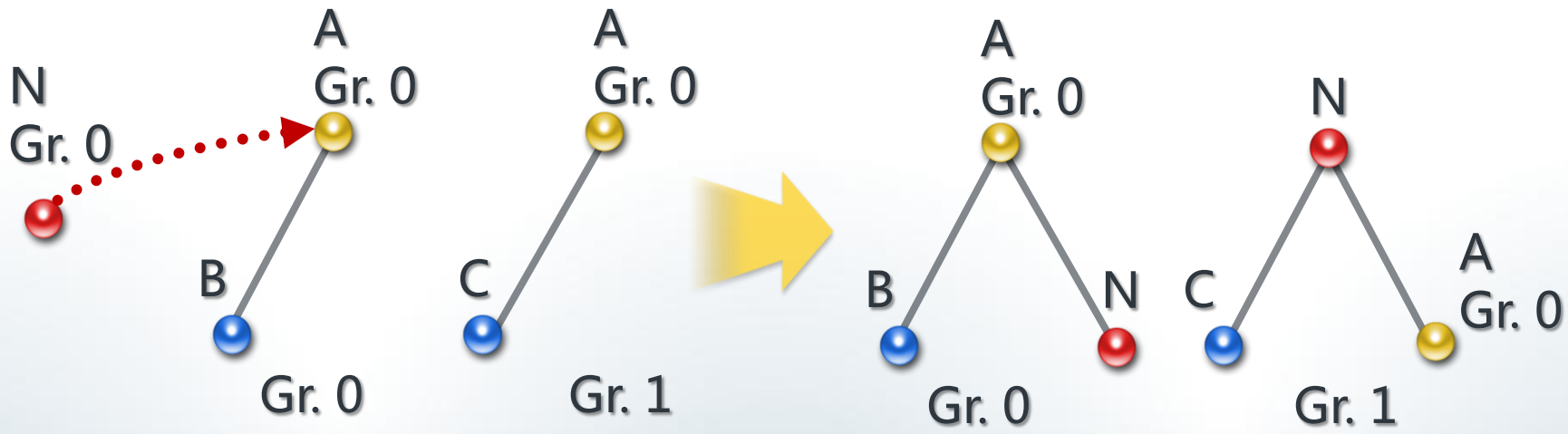
What is Different?

- Classify the nodes in
  - Incomplete: fan-out not reached in the proper tree
  - Complete: fan-out limit reached in the proper tree
  - Only-child: a leaf node

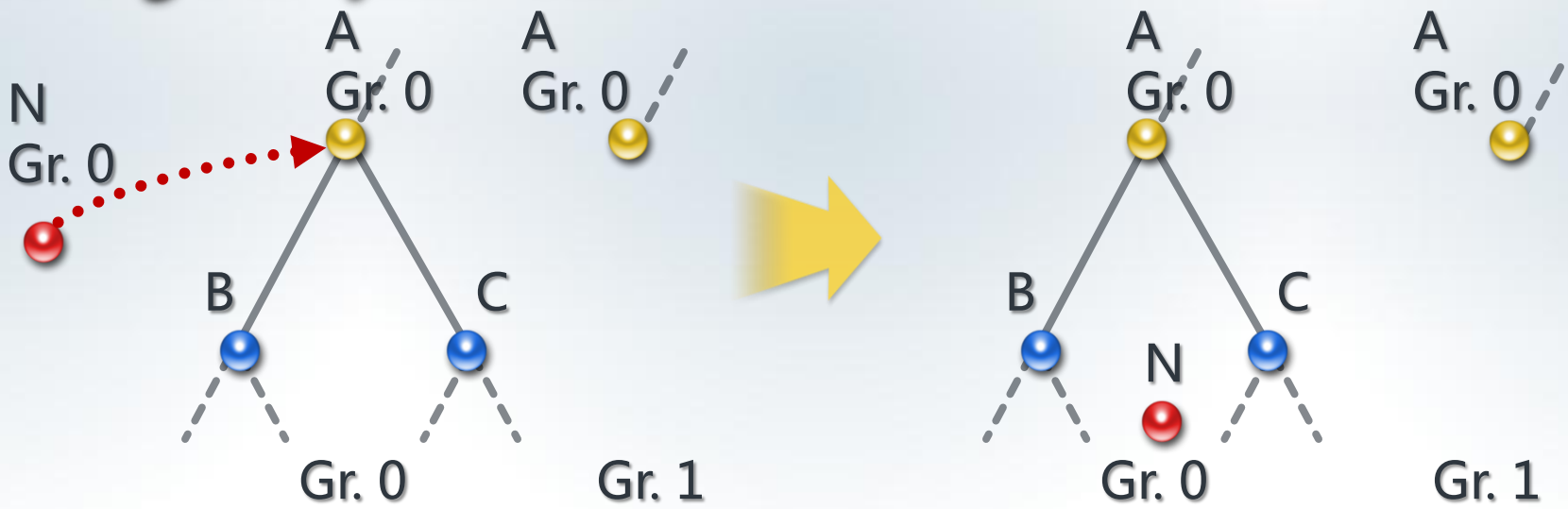
Let's assume an overlay with 2 tree groups and fan-out 2 for all peers

Joining

Joining an incomplete node

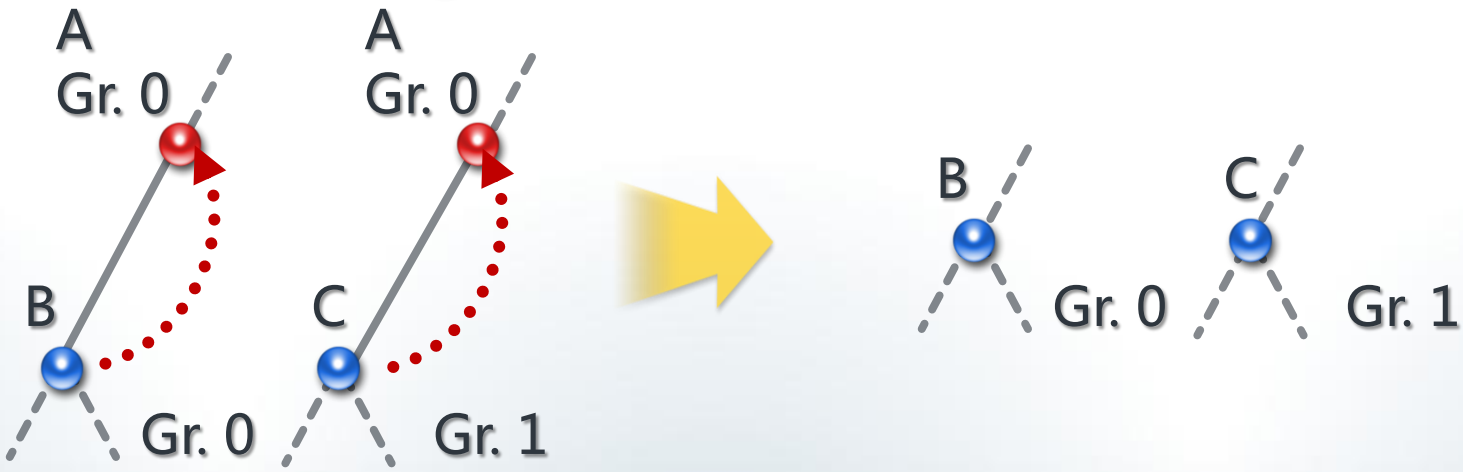


Joining a complete node



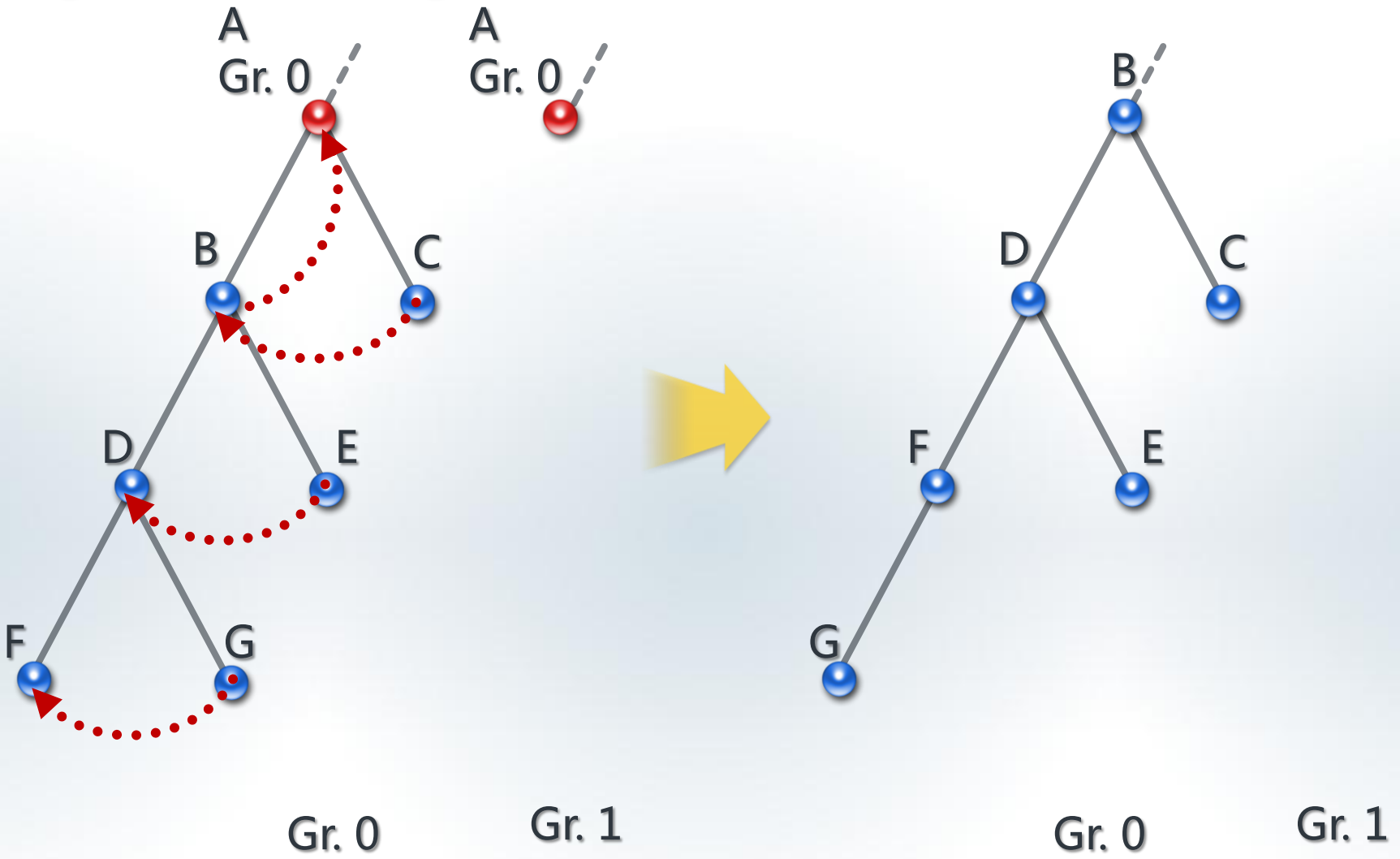
Departures & Failures

Departure of an incomplete node



Departures & Failures

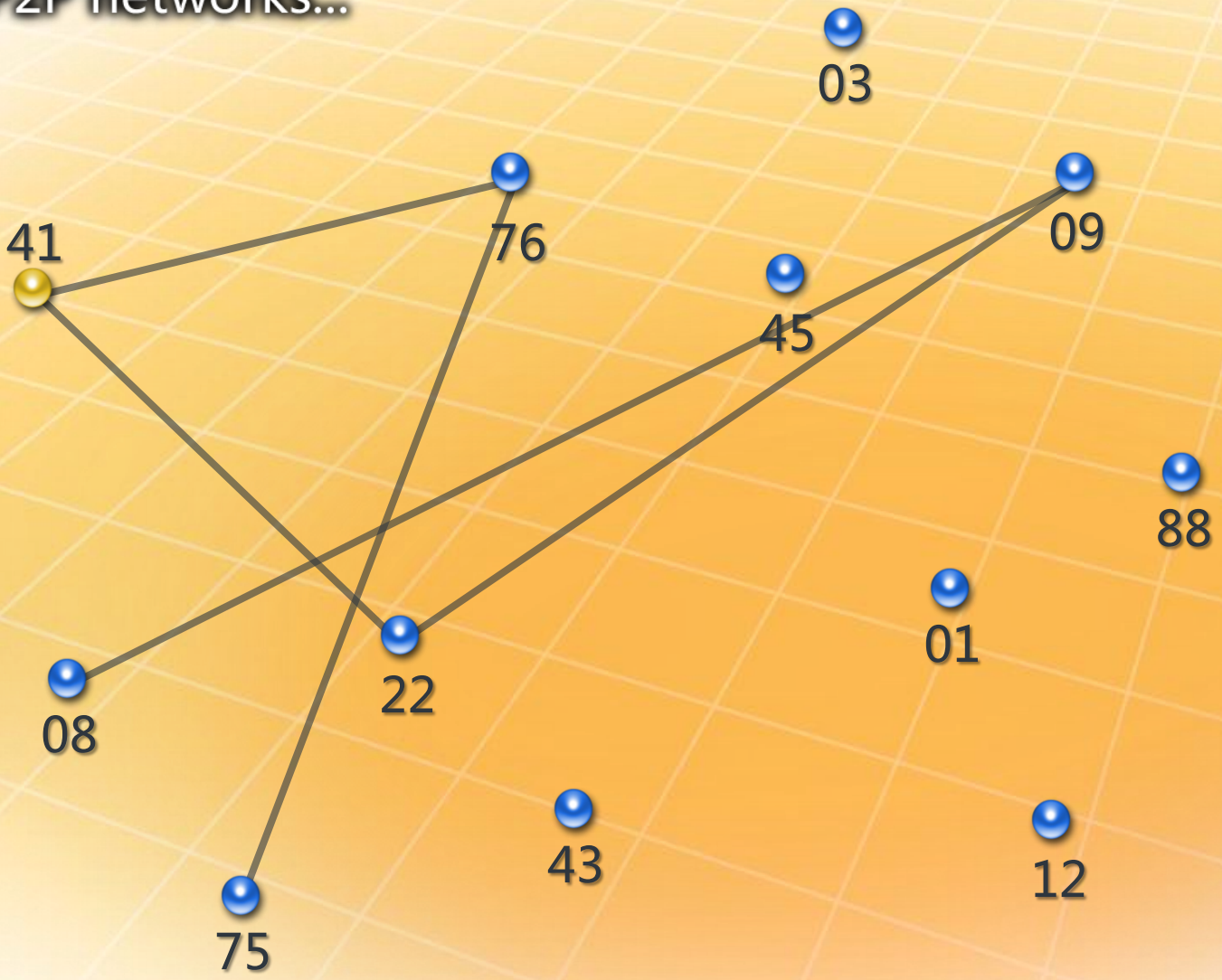
Departure of a complete node



PeerCast

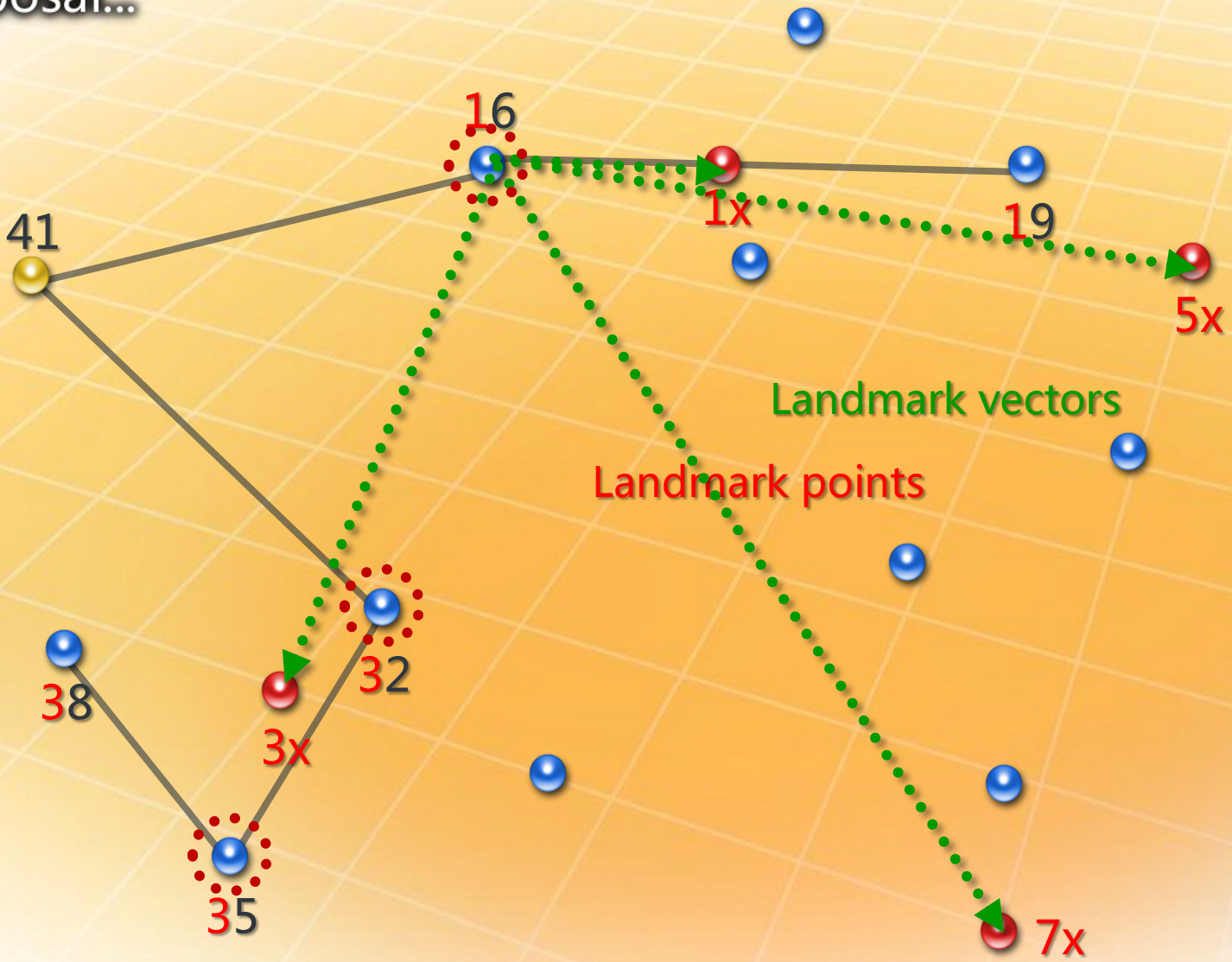
Proximity Awareness

In usual DHT P2P networks...



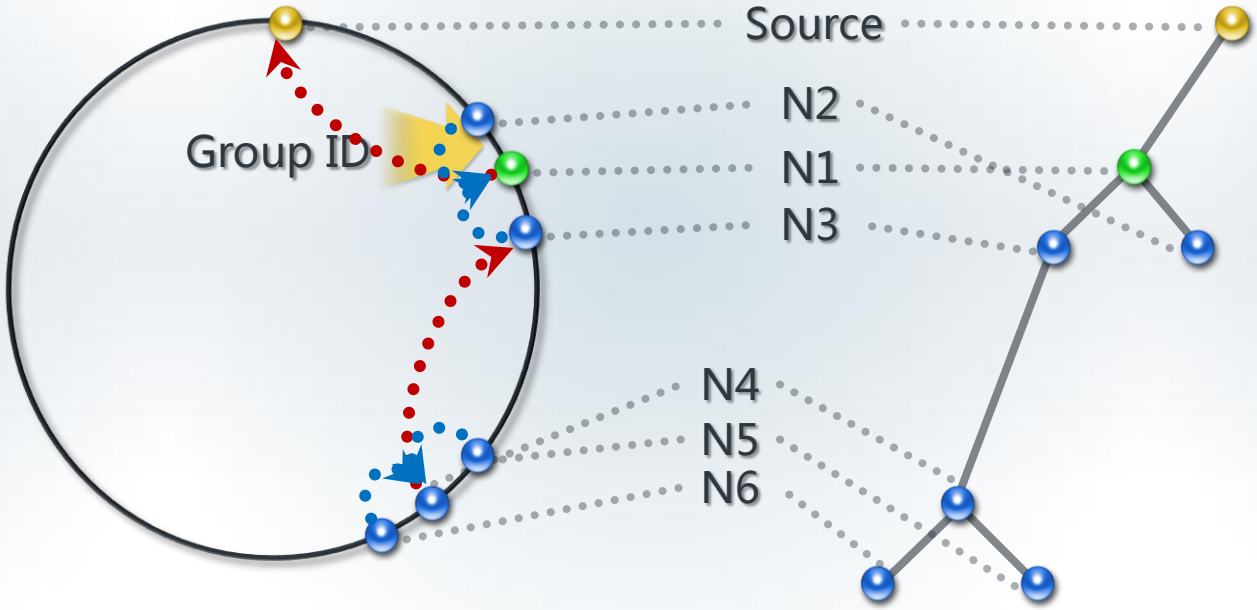
Proximity Awareness

PeerCast proposal...



Multicast Management

- Service (group ID) advertised on an off-band channel
- Peer with closest ID becomes a rendezvous point
- New peers will lookup the neighbors
- Only if none of the neighbors are in the group, a lookup towards the group ID is performed



Q&A