

GLive: The Gradient overlay as a market maker for mesh-based P2P live streaming

Amir H. Payberah - Jim Dowling - Seif Haridi
{amir, jdowling, seif}@sics.se



Introduction

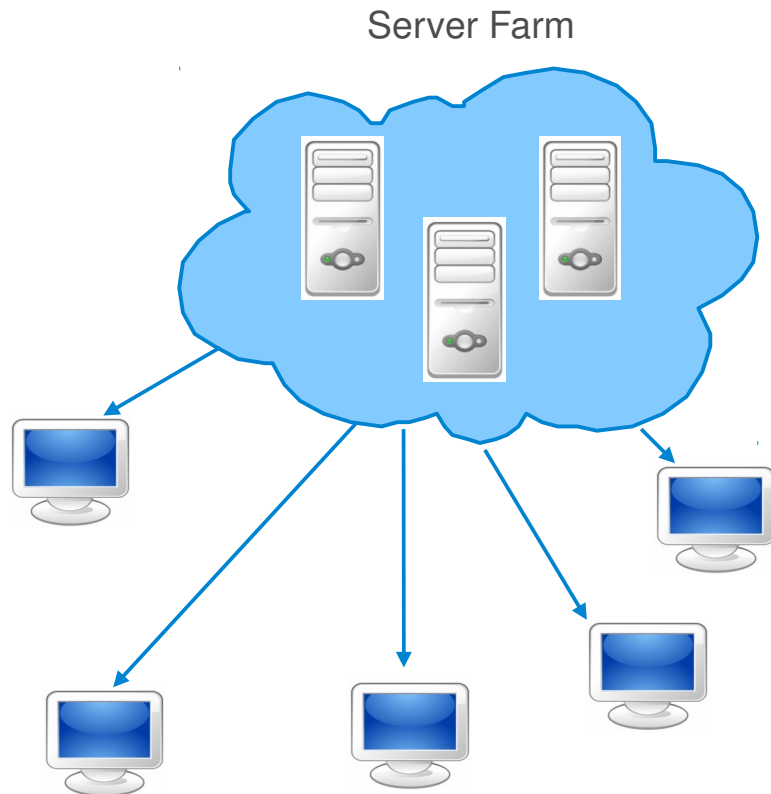
Media Streaming

- **Media streaming** is a multimedia that is sent over a network and played as it is being received by end users.
- Users do **not** need to **wait** to download all the media.
- It could be
 - **Live** Media Streaming
 - Video on Demand (**VoD**)

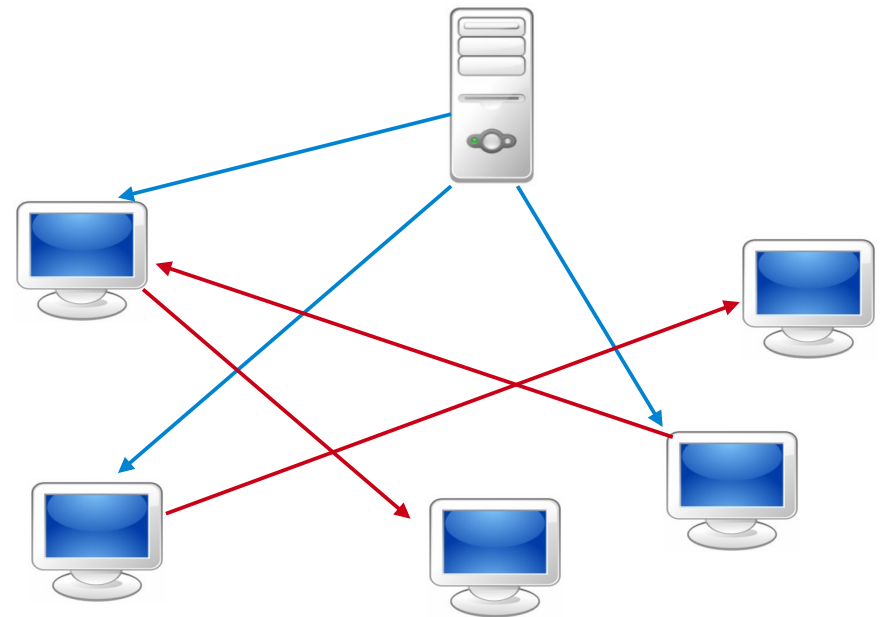


Solutions for Media Streaming

Client-Server

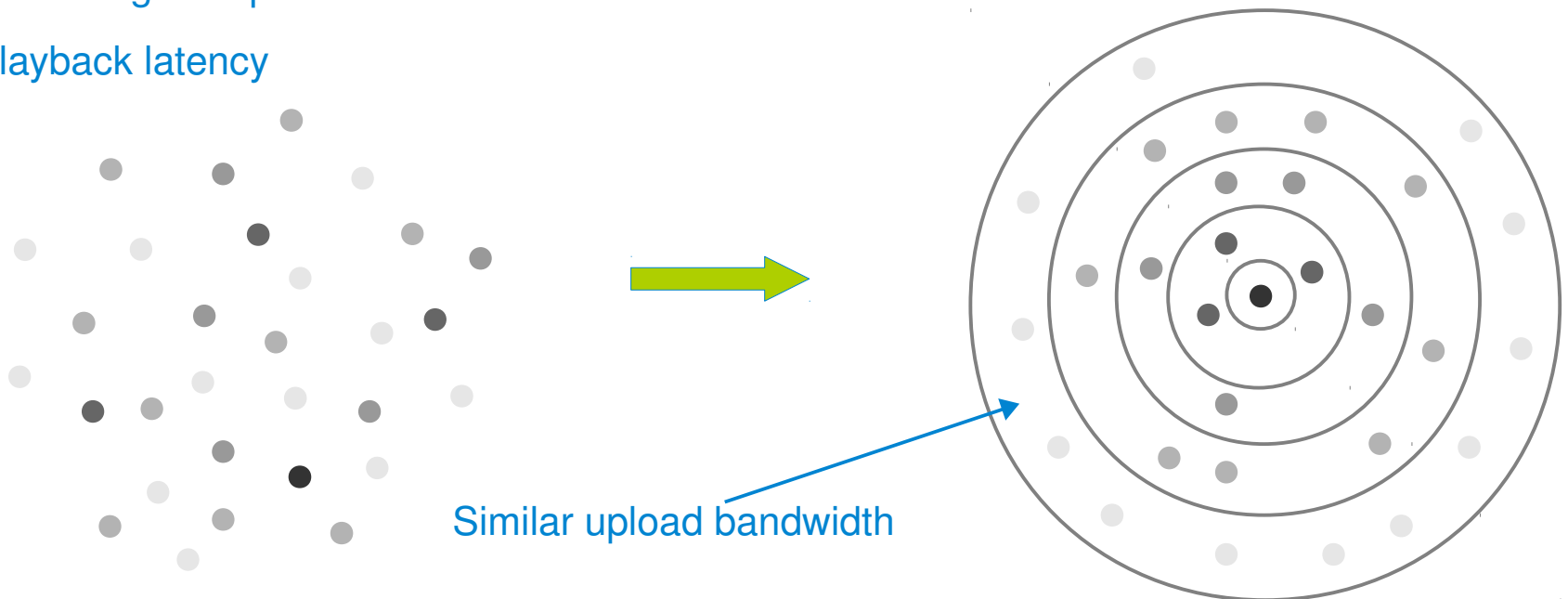


Peer-to-Peer



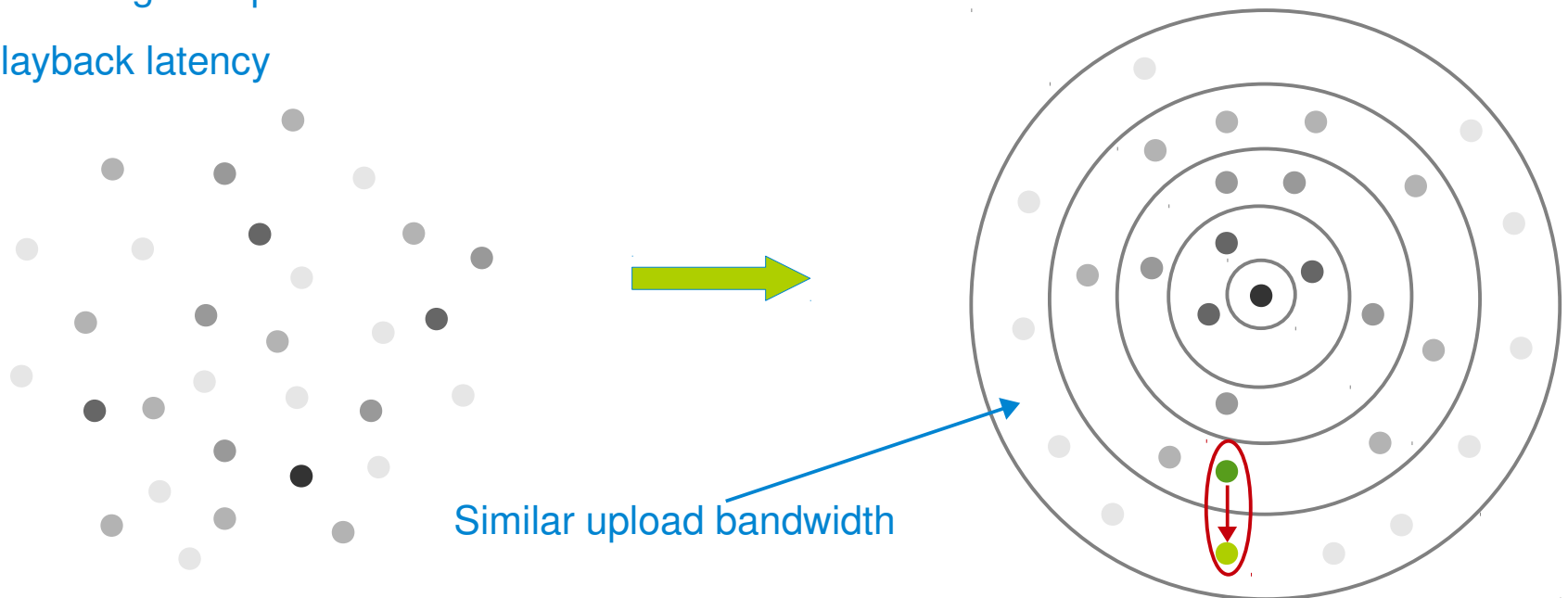
Problem Description (1/4)

- Building a streaming overlay network, such that:
 - Nodes with **higher upload bandwidth** are positioned **closer** to the media source.
 - Nodes with **similar upload bandwidth** become **neighbours**.
- Reduces:
 - Average number of hops
 - Streaming disruptions
 - Playback latency



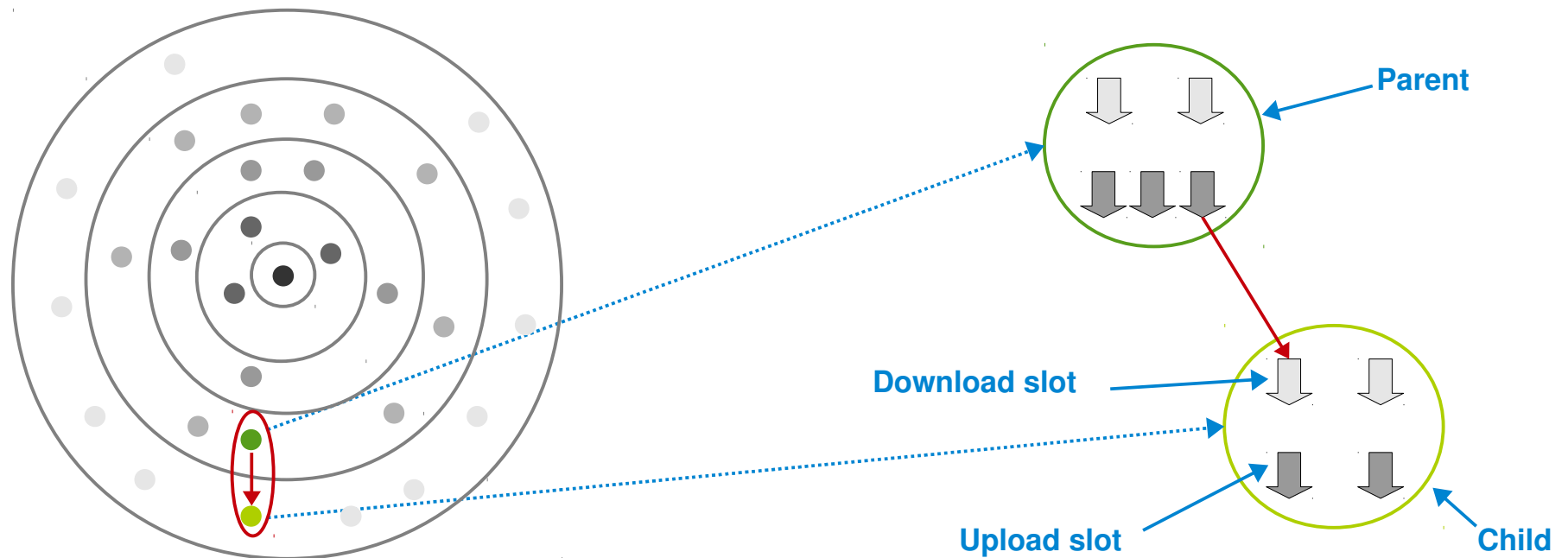
Problem Description (1/4)

- Building a streaming overlay network, such that:
 - Nodes with **higher upload bandwidth** are positioned **closer** to the media source.
 - Nodes with **similar upload bandwidth** become **neighbours**.
- Reduces:
 - Average number of hops
 - Streaming disruptions
 - Playback latency



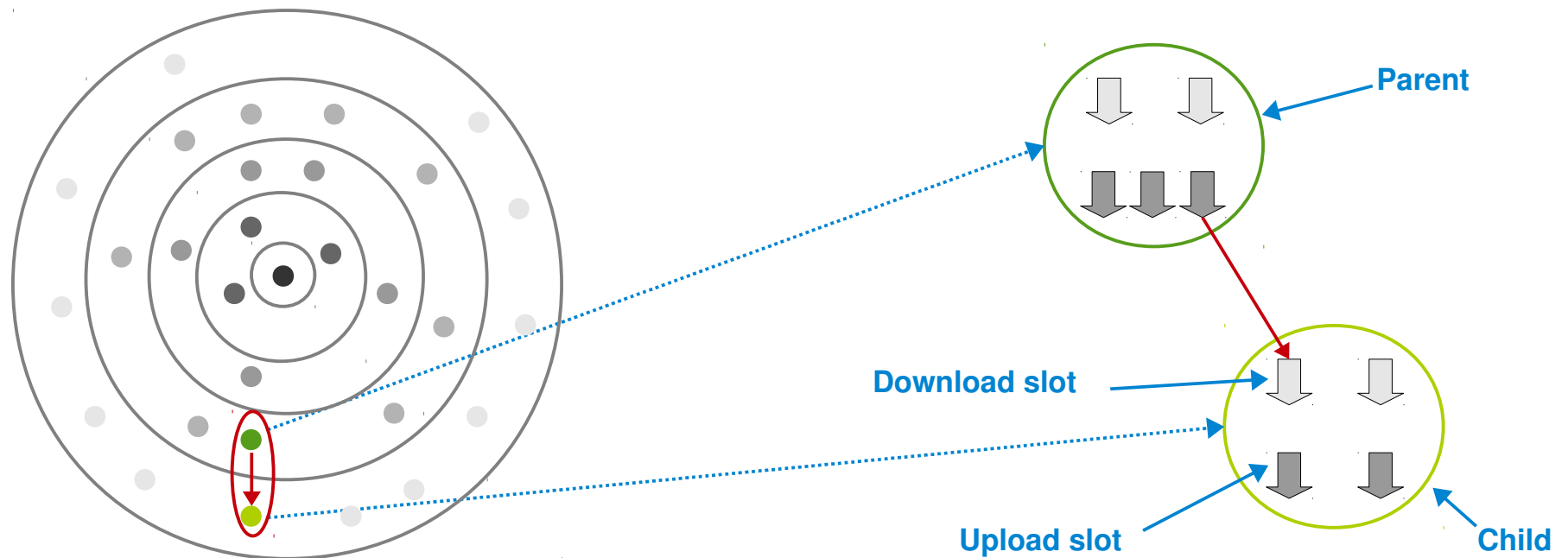
Problem Description (2/4)

- Bounded download/upload connections.
- A parent-child relation.



Problem Description (3/4)

- Problem:
 - How to assign upload slots to download slots?



Problem Description (4/4)

- This can be modelled as an **assignment problem**.
- Centralized solution:
 - Needs **global knowledge**.
 - Possible for **small** system sizes.
- **Distributed market-based** approach:
 - Inspired by **auction algorithms**.
 - Each node knows only a **small number of nodes** in the system (**partial view**).

GLive

Design Space

- What **overlay topology** is built for data dissemination?
- What **algorithm** is used for data dissemination?
- How to **discover** supplying nodes?

Design Space

- What **overlay topology** is built for data dissemination?
 - Tree
 - Multiple-tree
 - Mesh
- What **algorithm** is used for data dissemination?
 - Push
 - Pull
 - Push-Pull
- How to **discover** supplying nodes?
 - Centralized
 - DHT
 - Gossip-based
 - ...

Design Space

- What **overlay topology** is built for data dissemination?
 - Tree
 - Multiple-tree
 - **Mesh**
- What **algorithm** is used for data dissemination?
 - Push
 - **Pull**
 - Push-Pull
- How to **discover** supplying nodes?
 - Centralized
 - DHT
 - **Gossip-based**
 - ...



GLive – Design Space

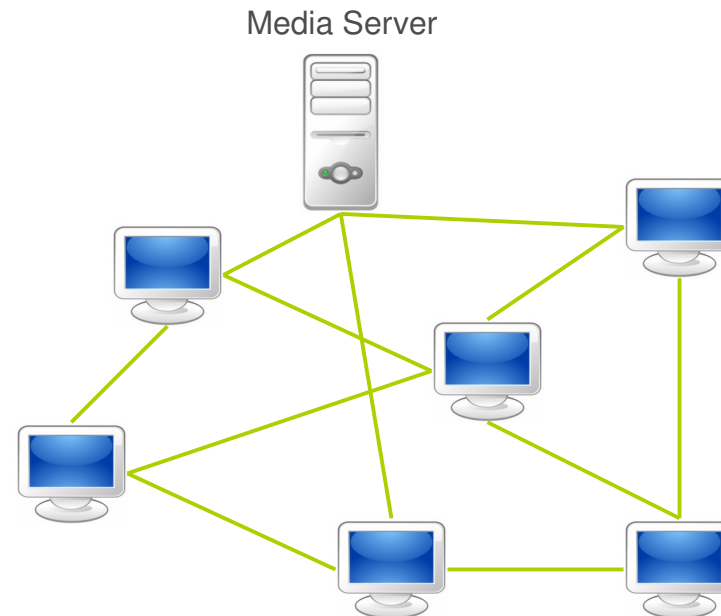
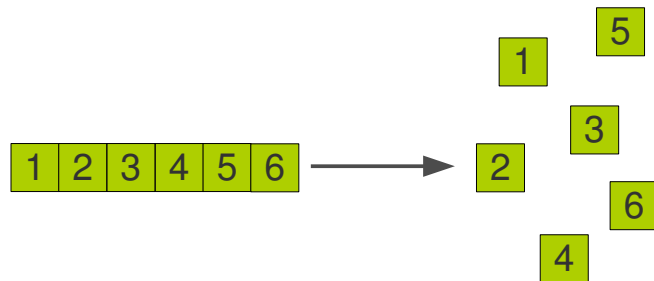
What overlay topology is built for data dissemination?

What algorithm is used for data dissemination?

How to discover supplying nodes?

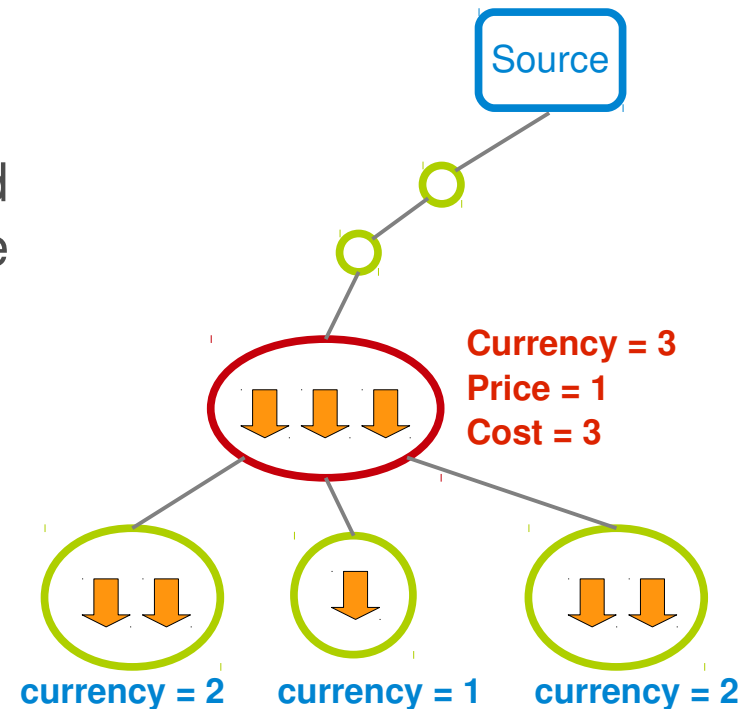
Mesh Overlay

- Divide the main stream into small **blocks**.
- Nodes are connected in a **mesh-network**.

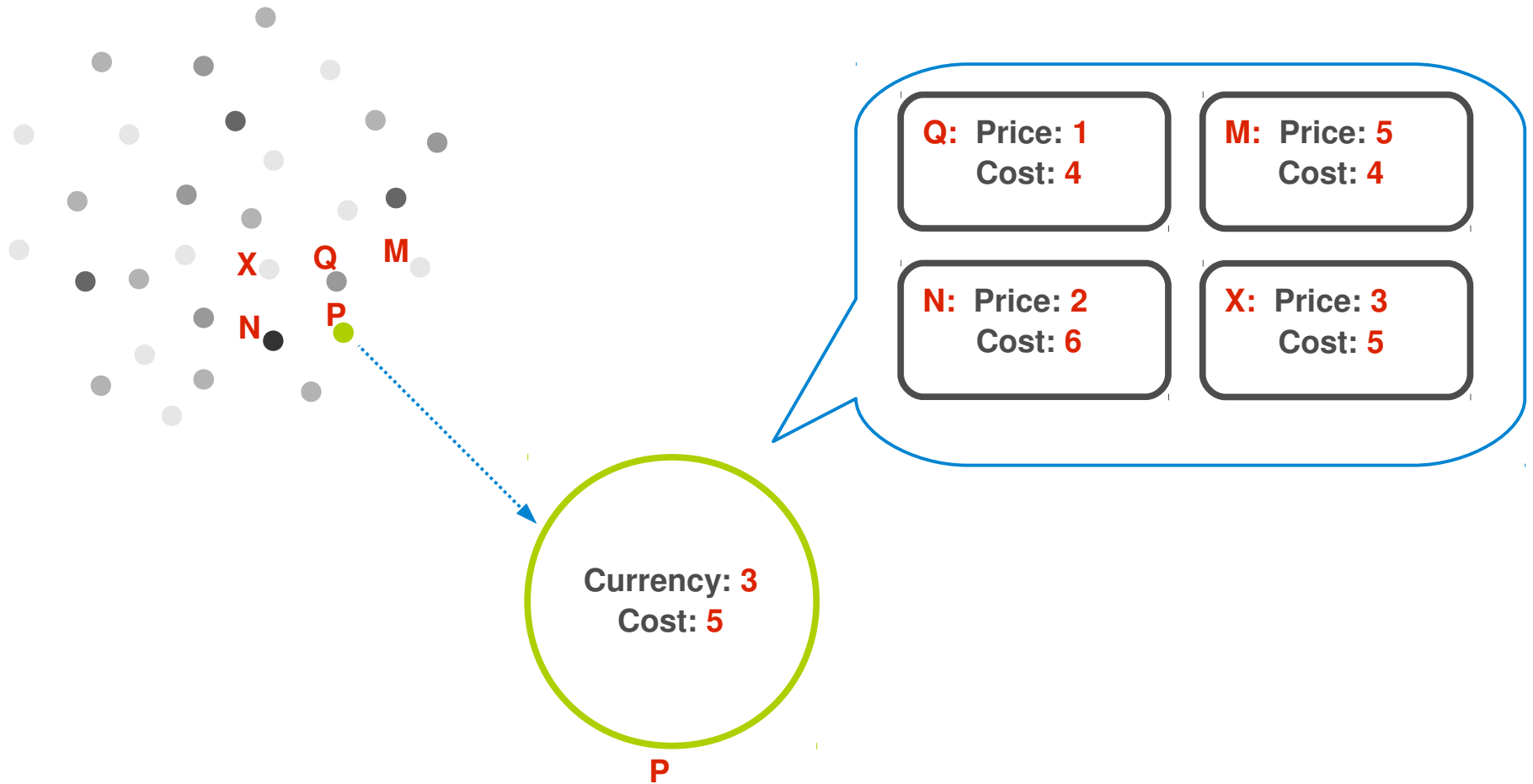


The Market Model - Node Properties

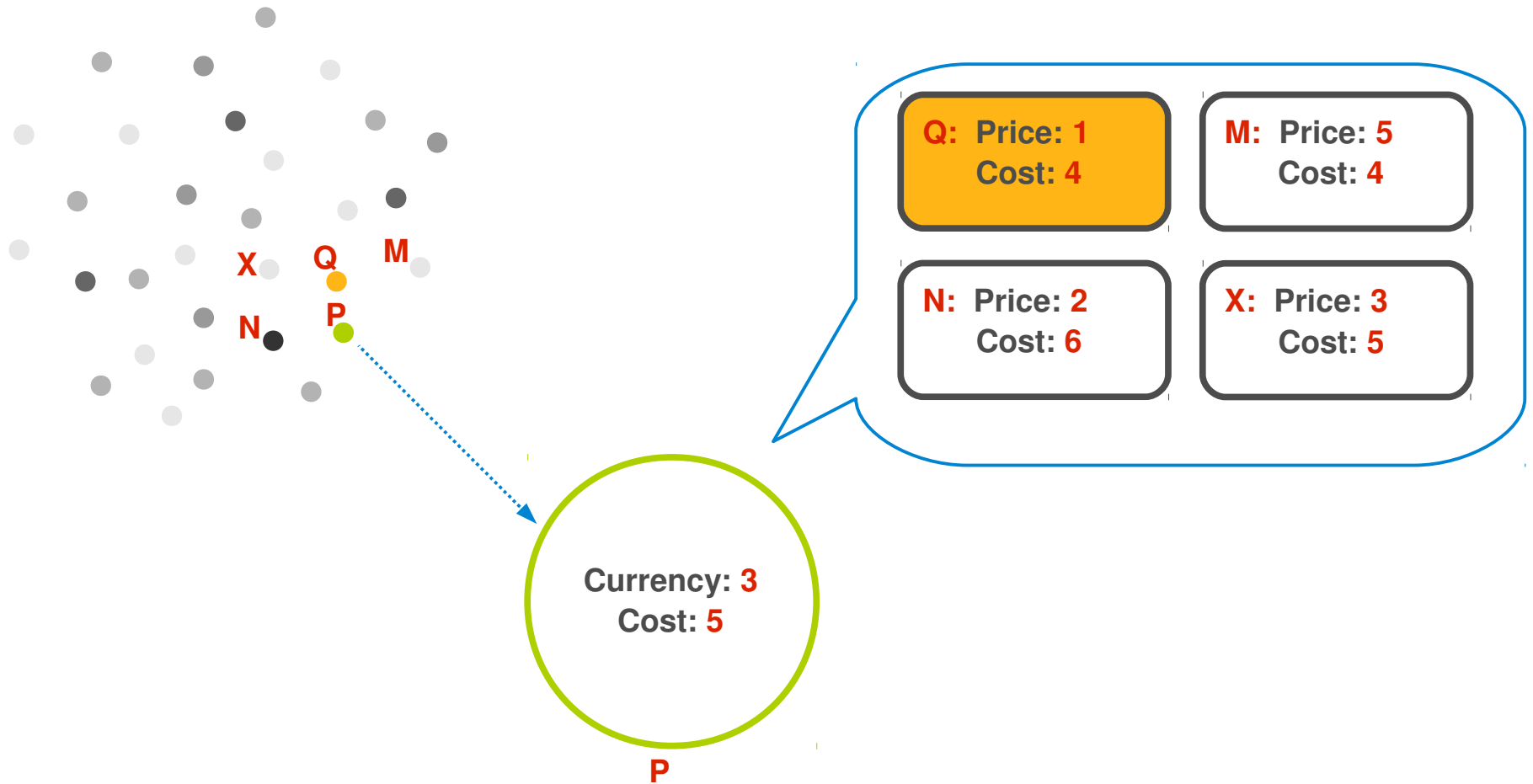
- **Currency**: The total number of blocks uploaded to children during the last 10 seconds.
- **Price**: The price of a node that has an unused upload slot is zero, otherwise the node's price equals the lowest currency of its already connected children.
- **Cost**: The length of its path to the root via its shortest path.



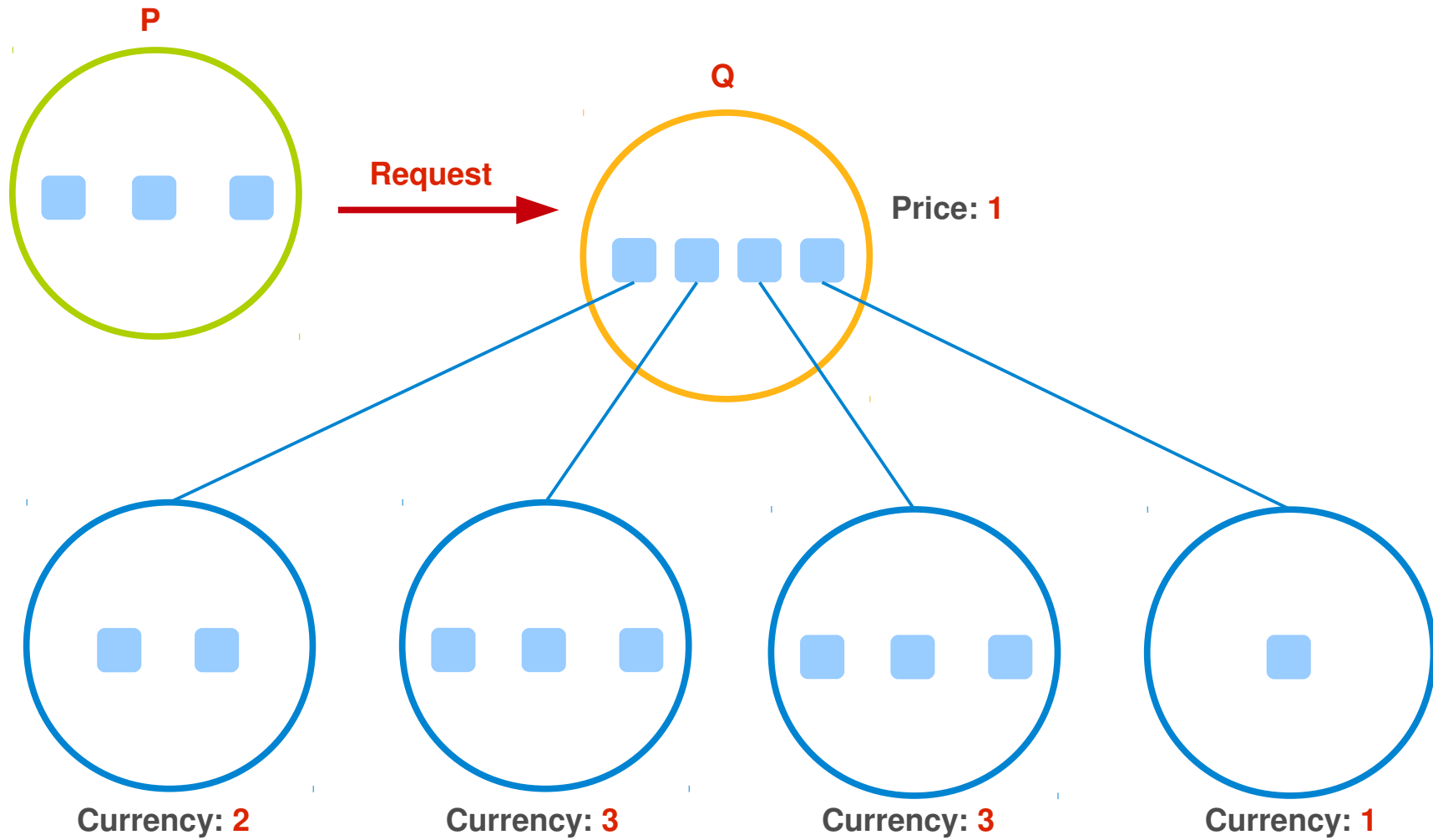
The Market Model – Child Side



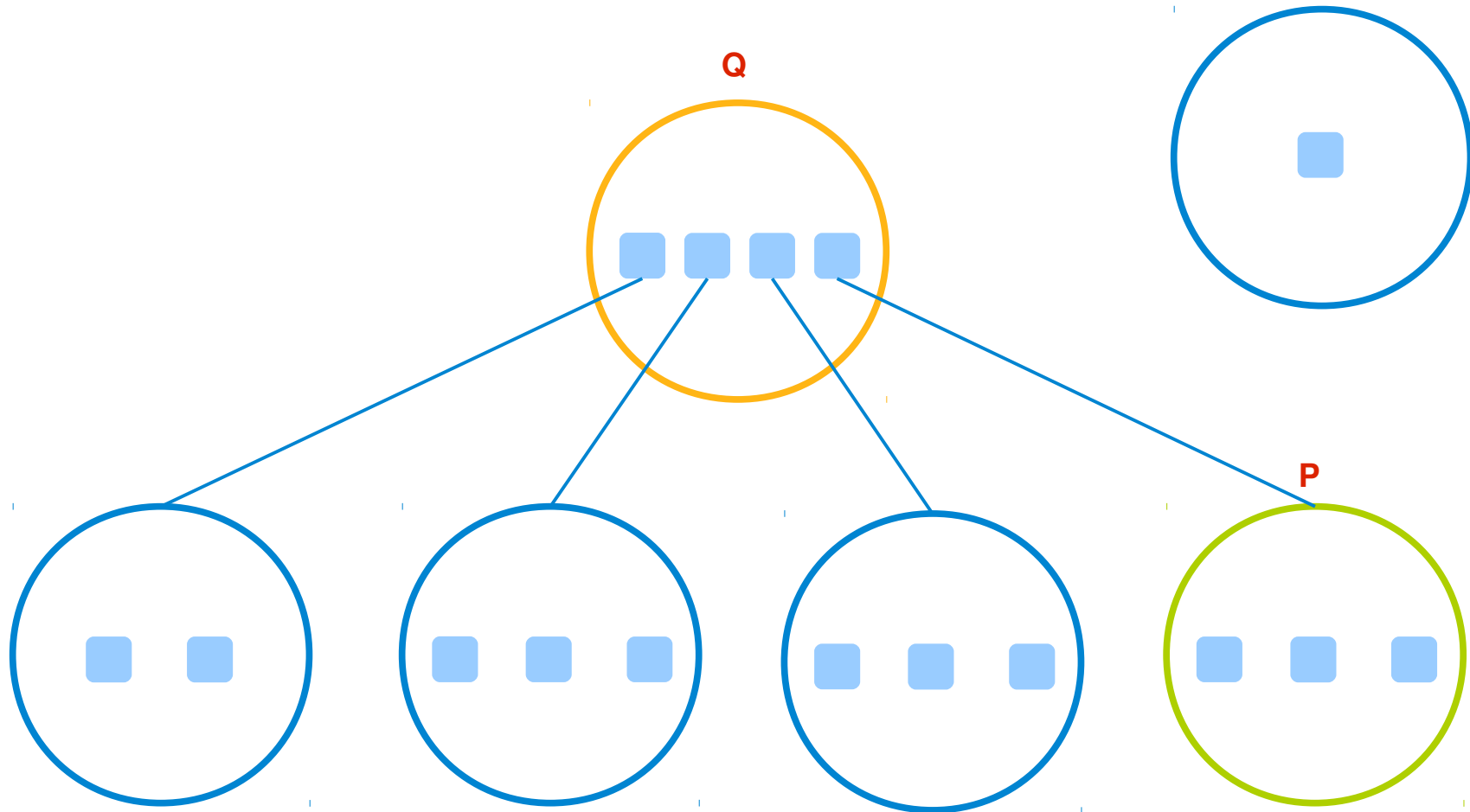
The Market Model – Child Side



The Market Model – Parent Side



The Market Model – Parent Side



GLive – Design Space

What overlay topology is built for data dissemination?

What algorithm is used for data dissemination?

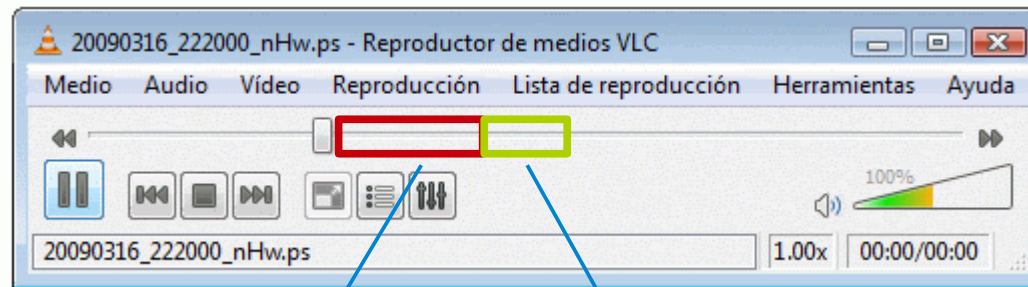
How to discover supplying nodes?

Data Dissemination (1/2)

- Each parent node periodically sends its **buffer map** and its **load** to all its assigned children.
- A child node, **pull** the required blocks using the received information.

Data Dissemination (2/2)

- Sliding window



In-order set

Rare set

GLive – Design Space

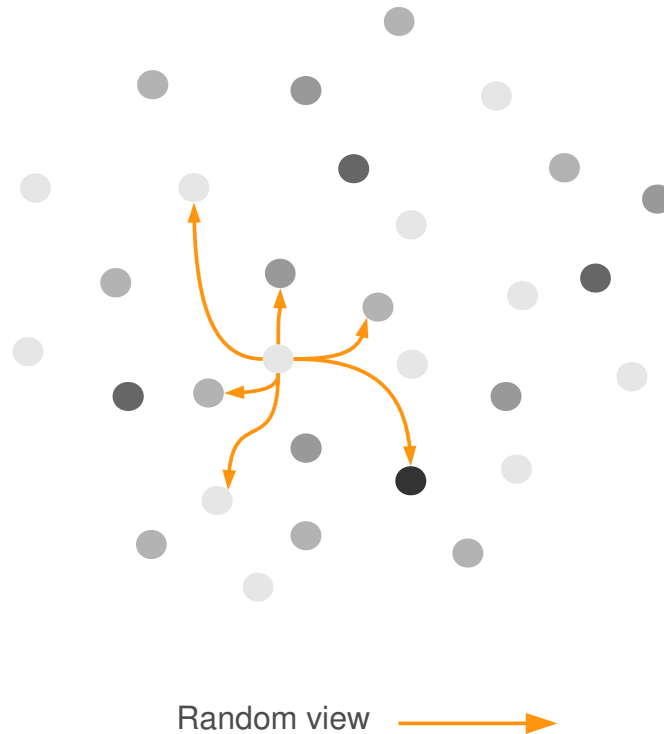
What overlay topology is built for data dissemination?

What algorithm is used for data dissemination?

How to discover supplying nodes?

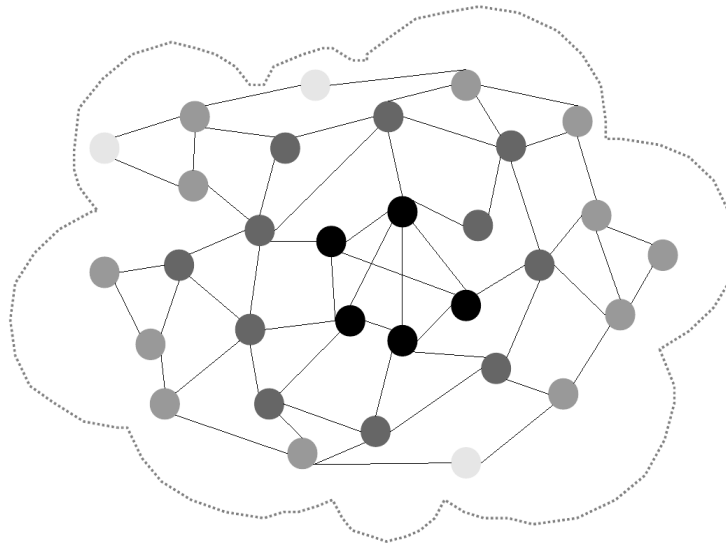
Node Discovery (1/2)

- **Naïve solution:** nodes in partial views are selected **randomly** from all the nodes.
- **Optimization:** nodes use the **Gradient overlay** to construct and maintain their partial view of the system.



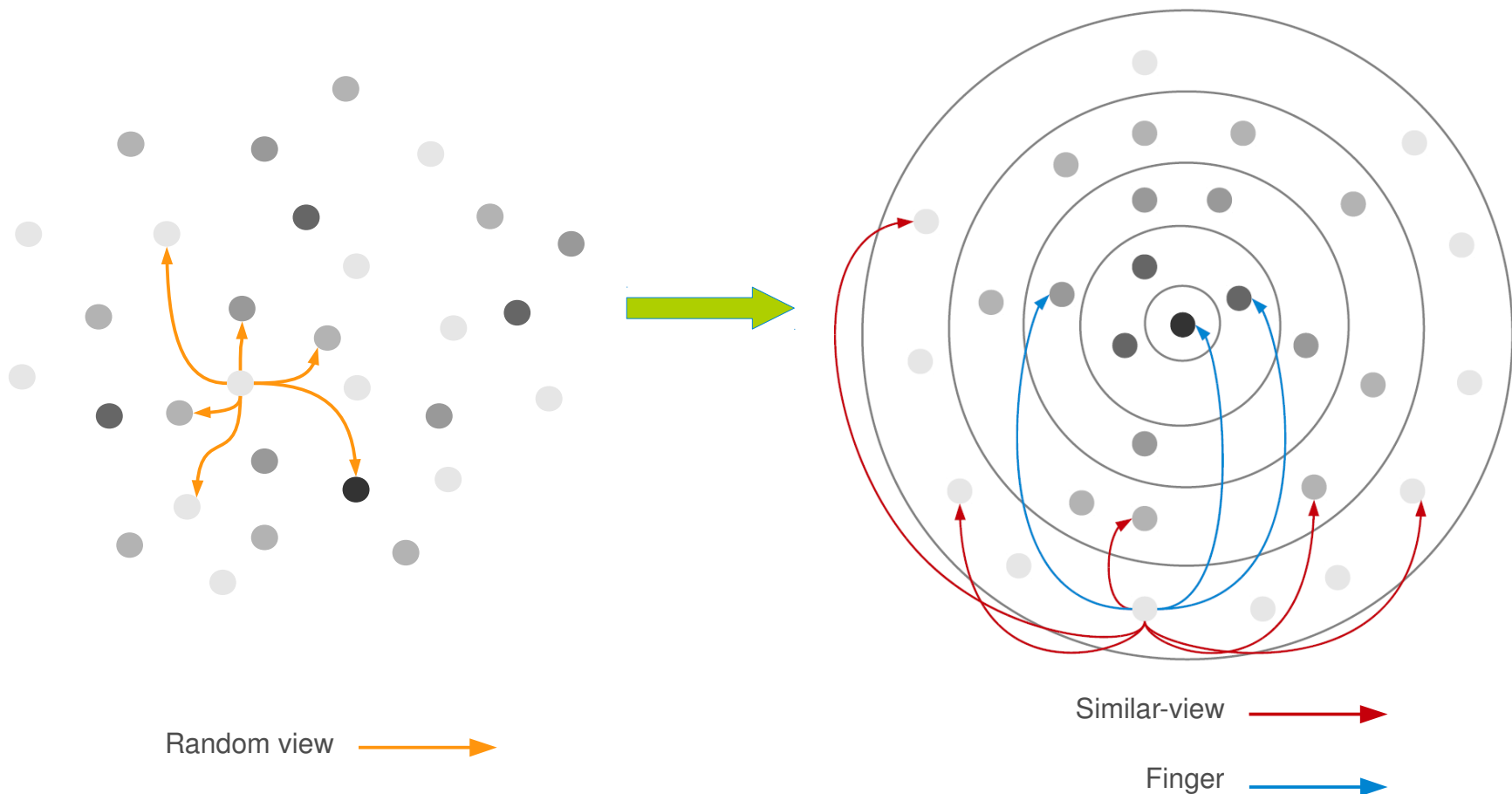
The Gradient Overlay

- The **Gradient overlay** is a class of P2P overlays that arranges nodes using a **local utility function** at each node, such that nodes are ordered in descending utility values away from a **core** of the **highest utility** nodes.



Node Discovery (2/2)

- Rather than have nodes **explore the whole system** for better parents, the Gradient enables nodes to **limit exploration** to the set of nodes with a similar number of upload slots.



Conclusion

Conclusion

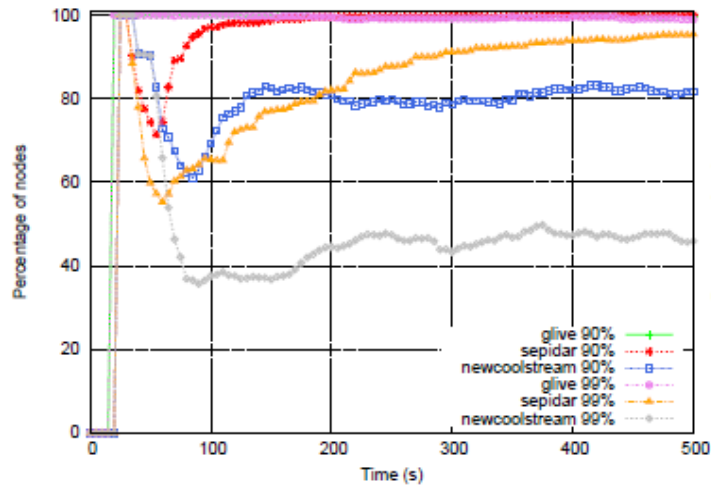
- **GLive** → P2P solution for live media streaming.
- Distributed market model
- Design space:
 - Overlay topology: **Mesh**
 - Data dissemination: **Pull**
 - Discover supplying nodes: **Gossip – Gradient**

Question?

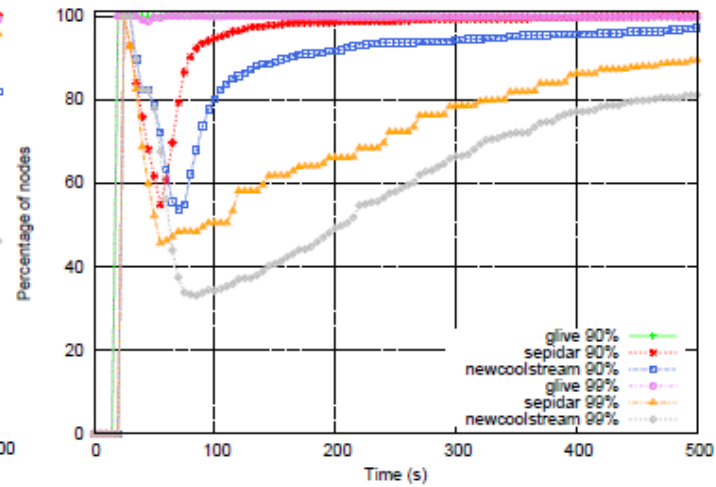
Experiment Setup

- Using the [Kompics](#) as a simulator platform.
- [King dataset](#) is used to model the latencies between nodes.
- The streaming rate to [512 Kbps](#), and it is split into [8](#) stripes (in sepidar). The stream/stripes is divided into a sequence of [16 Kb](#) blocks.
- Nodes start playing the media after buffering it for [15](#) seconds.
- The number of upload slots for the non-root nodes is picked randomly from [1](#) to [10](#).
 - bandwidths from [128 Kbps](#) to [1.25 Mbps](#).
- Compare with [NewCoolstreaming](#).

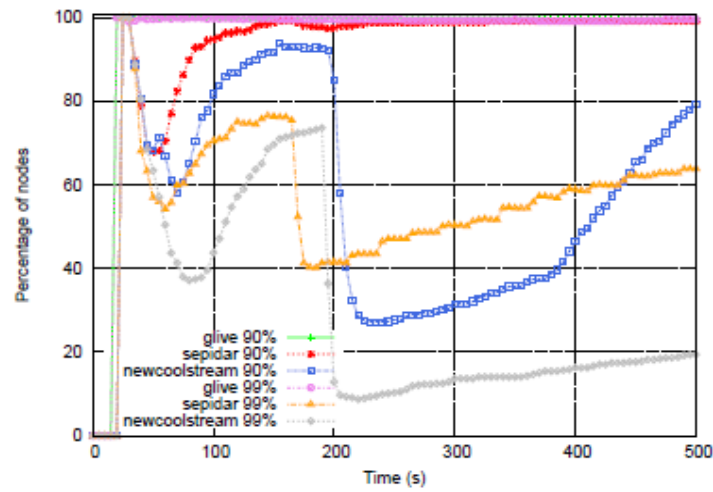
Playback Continuity



(a) Churn.

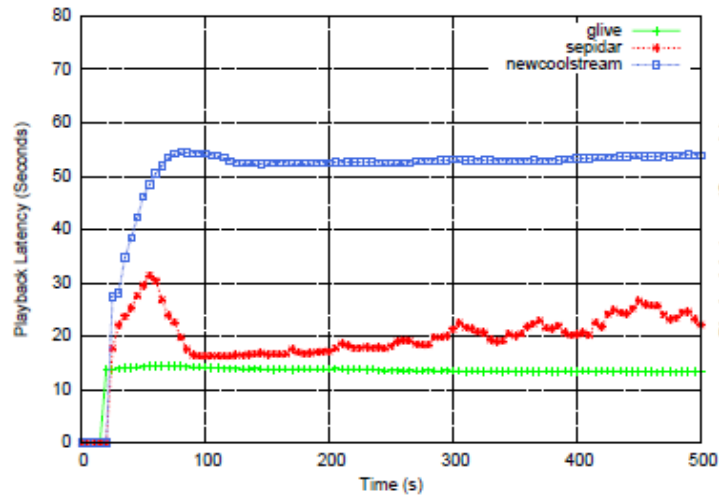


(b) Flash Crowd.

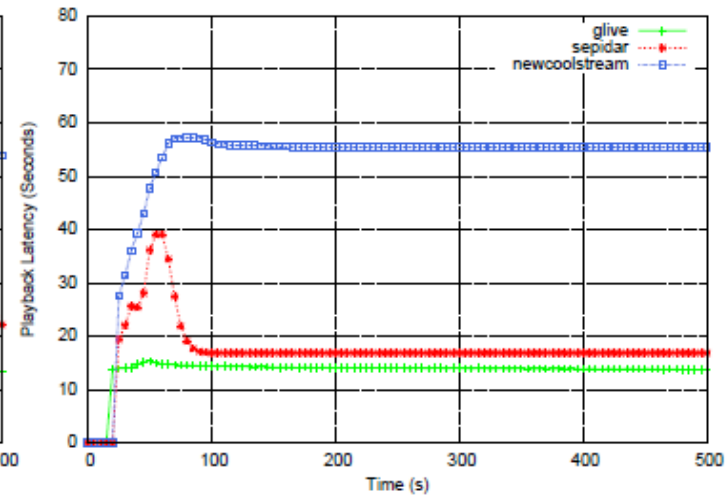


(c) Catastrophic failure.

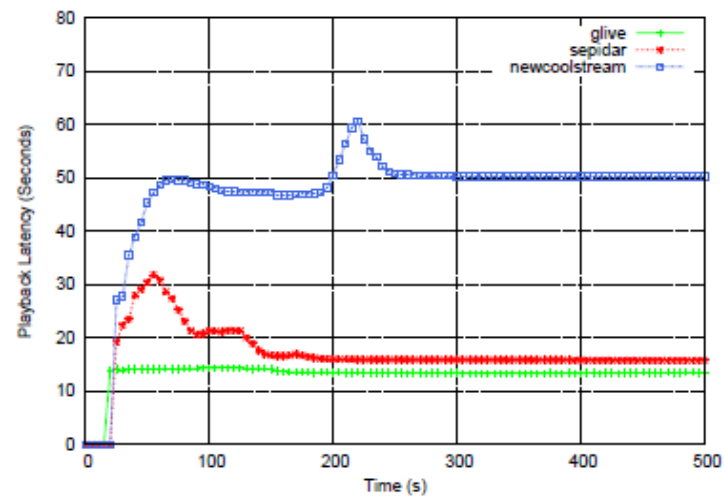
Playback Latency



(a) Churn.

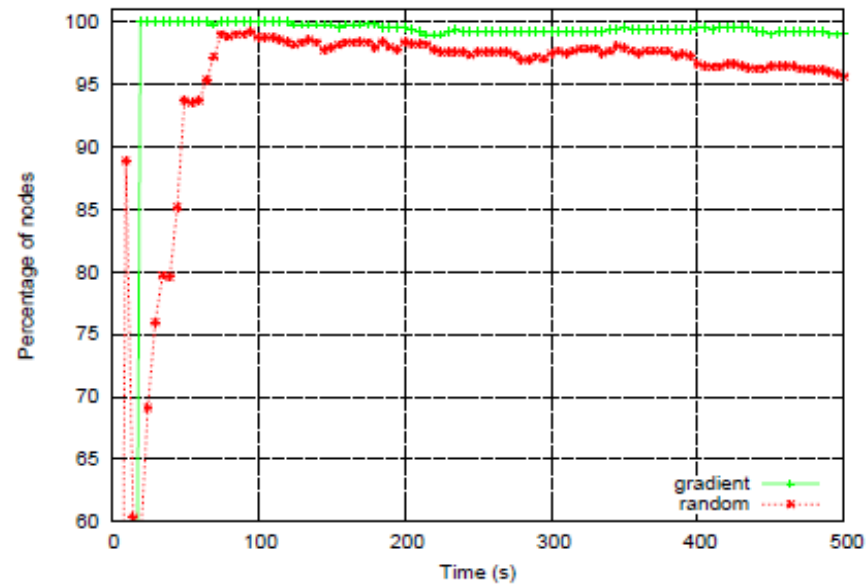


(b) Flash Crowd.

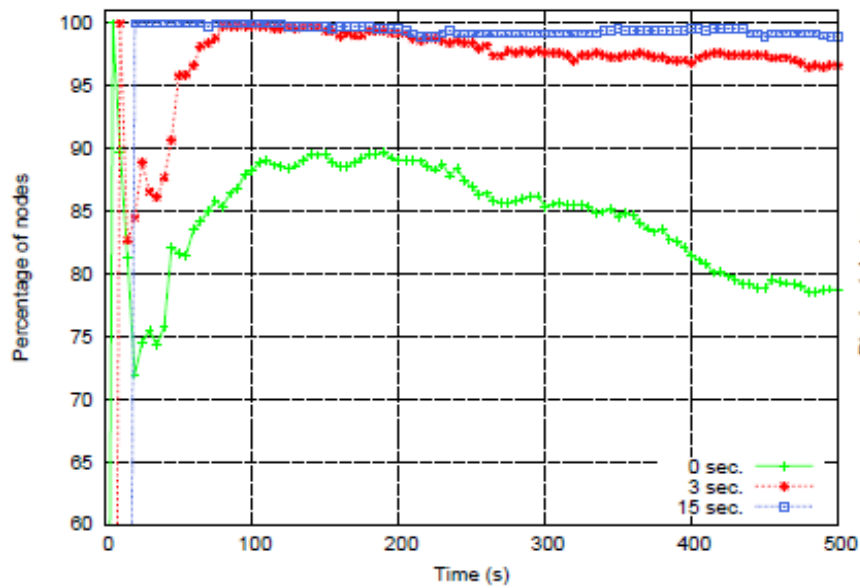


(c) Catastrophic failure.

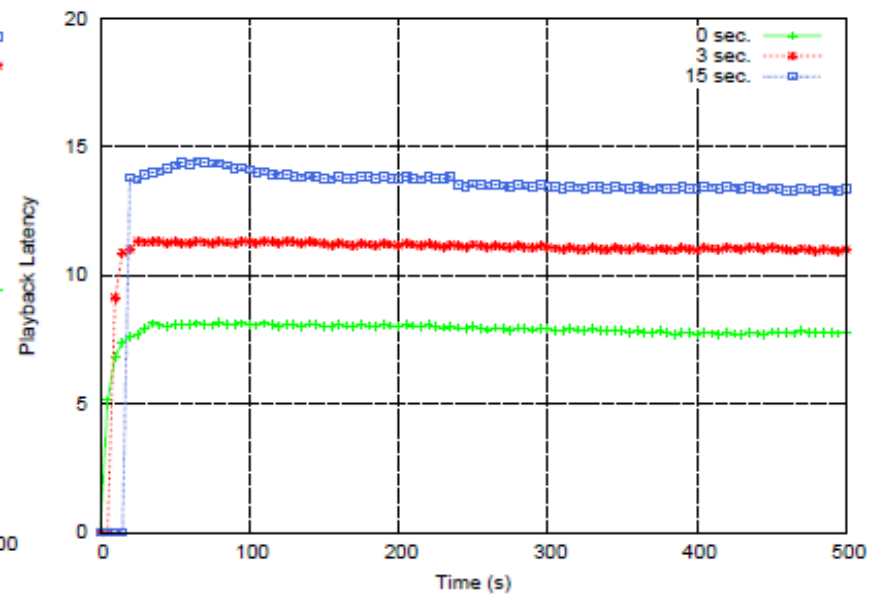
Gradient overlay vs. Random overlay



Buffering Time

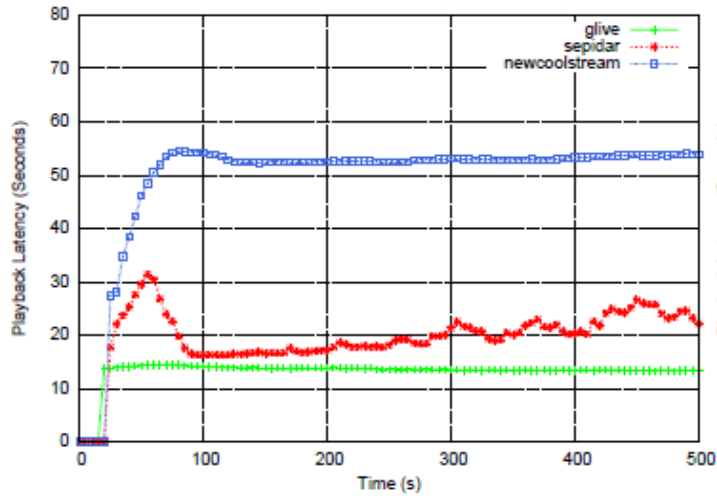


(a) 99% of playback continuity.

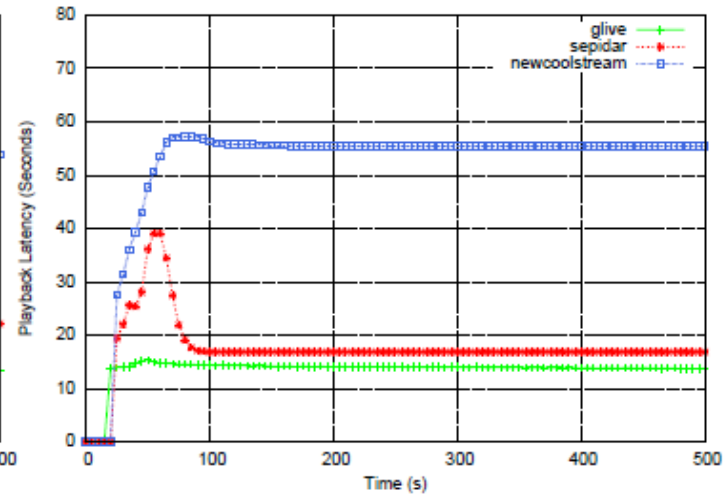


(b) Playback latency.

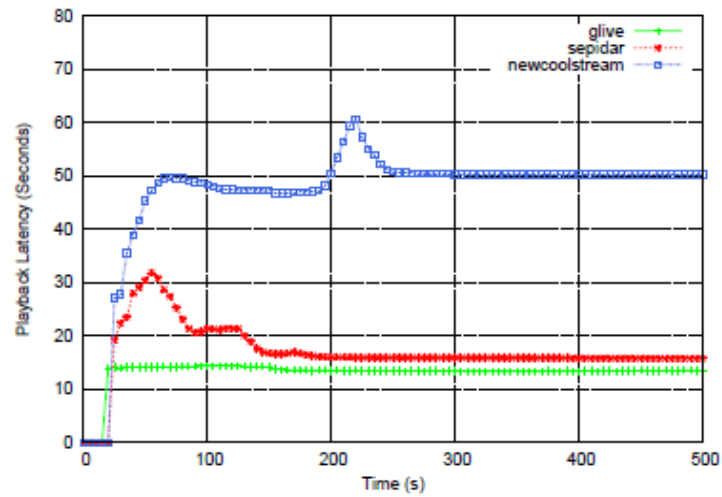
Playback Latency



(a) Churn.

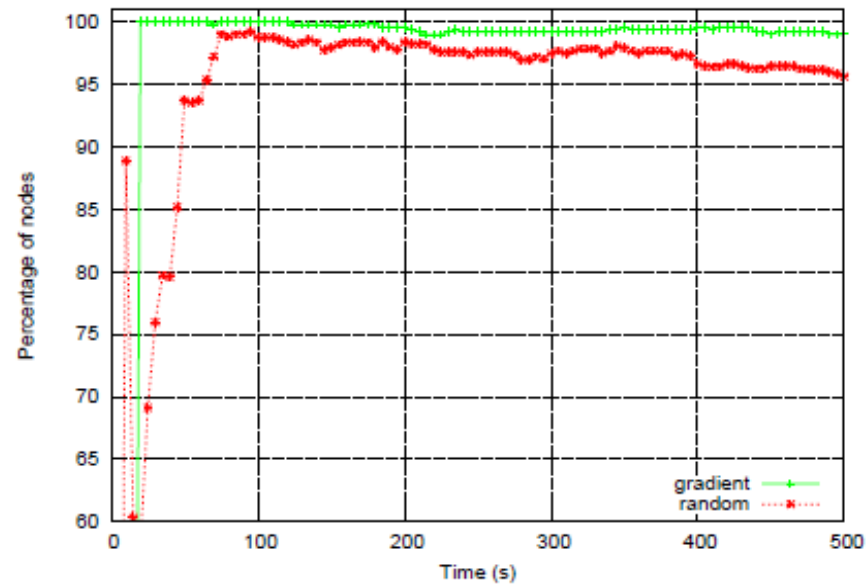


(b) Flash Crowd.

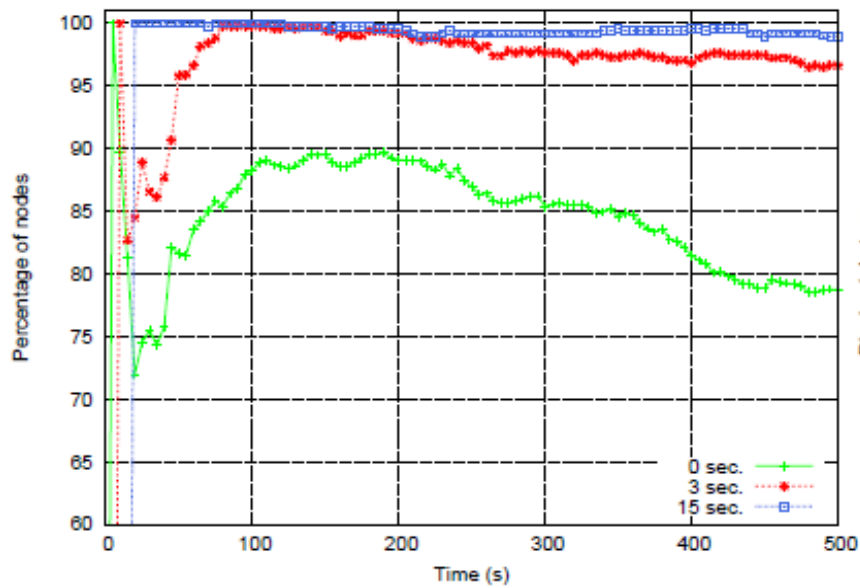


(c) Catastrophic failure.

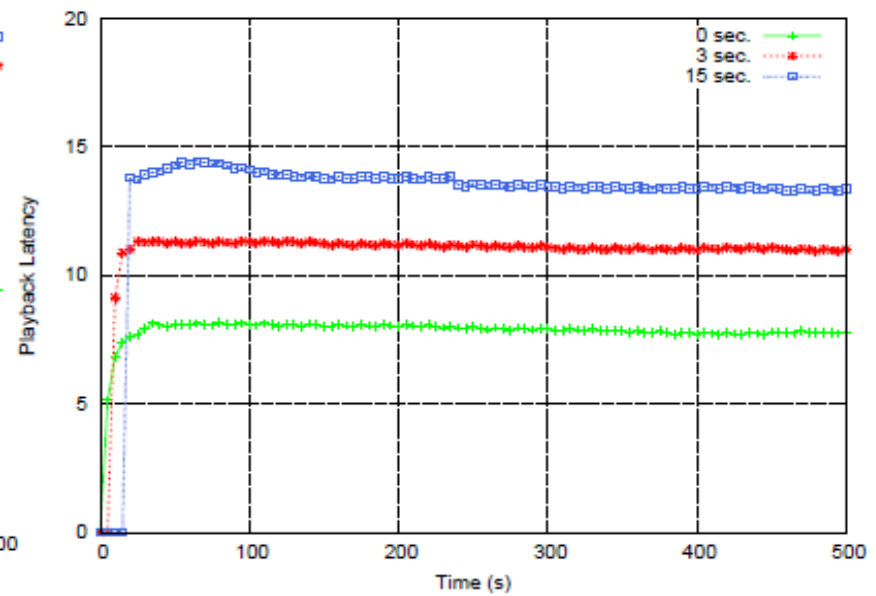
Gradient overlay vs. Random overlay



Buffering Time

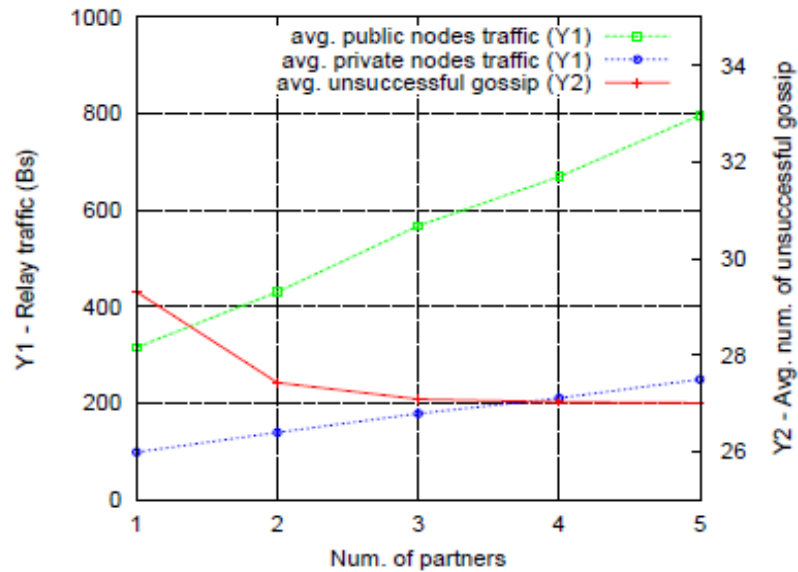


(a) 99% of playback continuity.

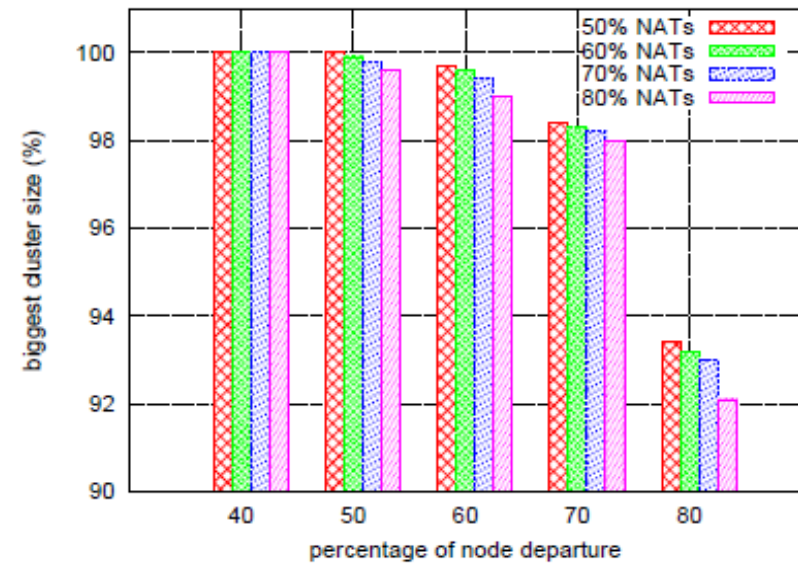


(b) Playback latency.

Fairness and Connectivity in Failure



(a) Fairness after catastrophic failure: overhead for public and private nodes for varying numbers of parents.



(b) Biggest cluster size after catastrophic failures.