Sepidar: Incentivized Market-Based P2P Live-Streaming on the Gradient Overlay Network

Amir H. Payberah – *Jim Dowling*
Fatemeh Rahimian – *Seif Haridi*

Big Picture

- Sepidar is a P2P solution for live media streaming.
- It uses a distributed market model and the Gradient overlay to construct the streaming overlay.

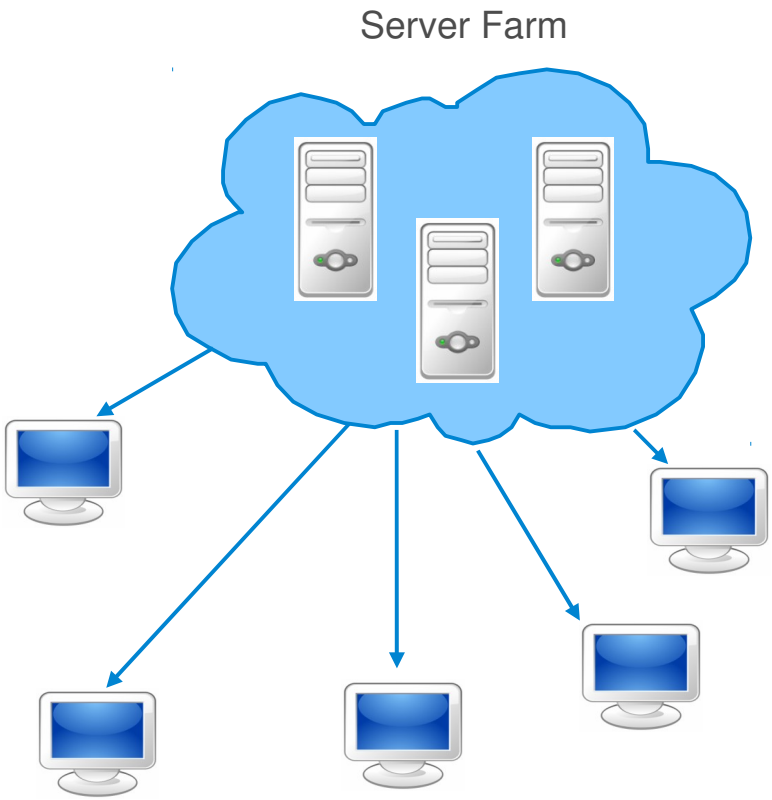
Motivation

Media Streaming

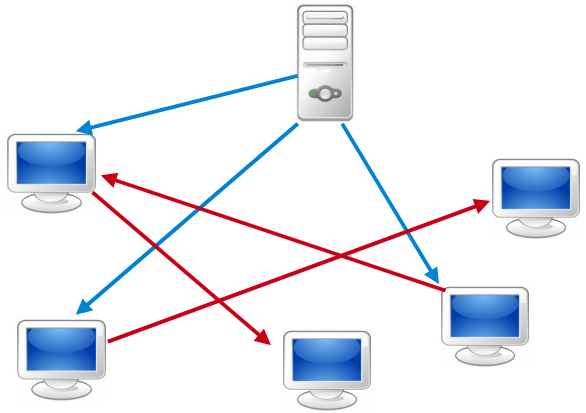
- Media streaming is a multimedia that is sent over a network and played as it is being received by end users.
- Users do not need to wait to download all the media.
- They can play it while the media is delivered by the provider.
- It could be:
  - Live streaming
  - Video on Demand (VoD)

Solutions for Application Level Media Streaming

Client-Server



P2P



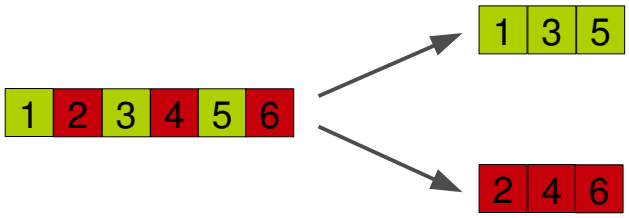
P2P Media Streaming Challenges

- Bandwidth intensive.
- Data should be received with respect to certain timing constraints.
  - A negligible startup delay
  - Smooth playback
  - A negligible playback latency (only for Live Streaming)
- Nodes join, leave and fail continuously.
  - Called churn
- Network capacity changes.

Problem Description

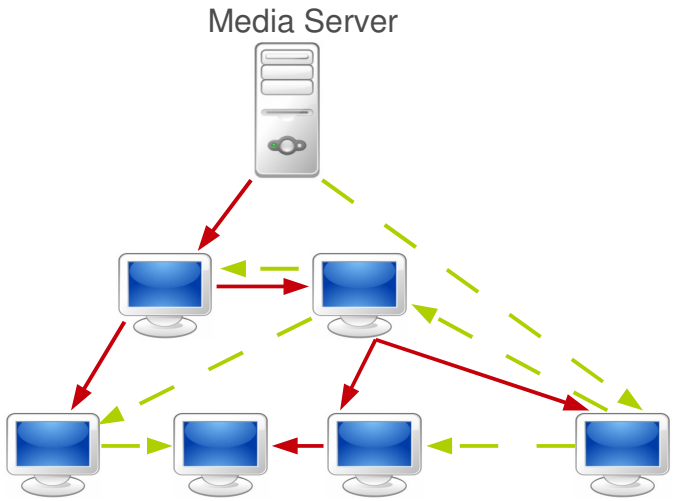
Problem Description (1/3)

- Split the main stream into a set of sub-streams, called stripes, and divides each stripe into a number of blocks.



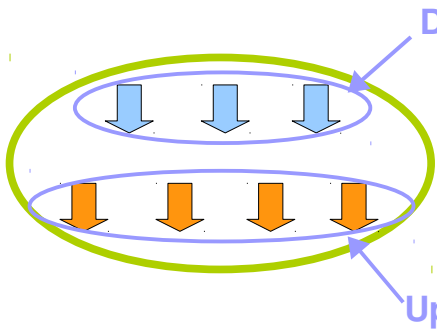
- In case of having 2 stripes:
  - Stripe 0: 0, 2, 4, 6, …
  - Stripe 1: 1, 3, 5, 7, …

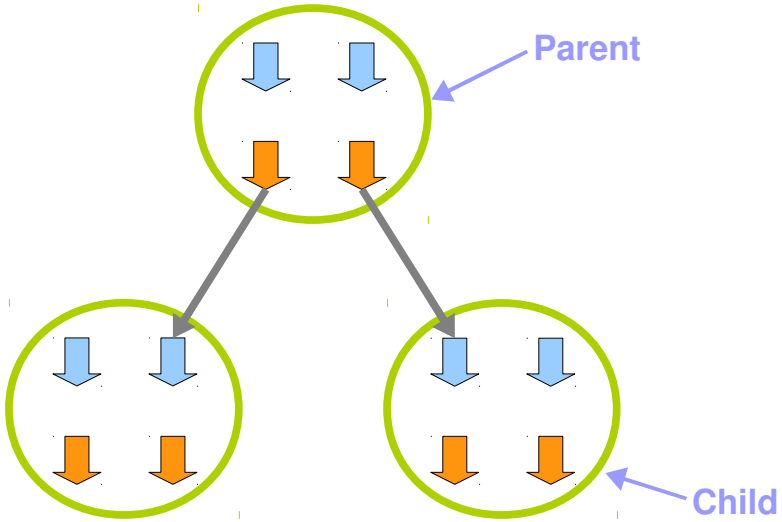
- Construct a tree for each stripe:
  - Multiple-tree



Problem Description (2/3)

- Upload slots: the number of copies of stripes that nodes are willing and able to forward.
- Download slots: equal to the number of stripes.





Problem Description (3/3)

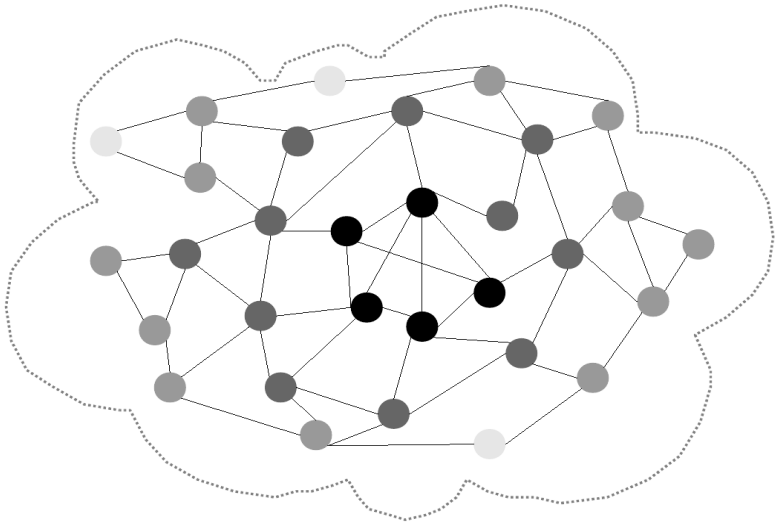
- The problem: how to deliver a video stream from a source as multiple stripes over multiple approximately minimal height trees.
  - assignment problem
- We use a market-based approach to construct the overlay trees.
  - Inspired by auction algorithms
- Centralized solution:
  - Needs global knowledge.
  - possible for small system sizes.
- Distributed solution:
  - Each node has only a partial view of the system.

Node Discovery

Node Discovery

- Nodes use the Gradient overlay to construct and maintain their partial view of the system.

- 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.



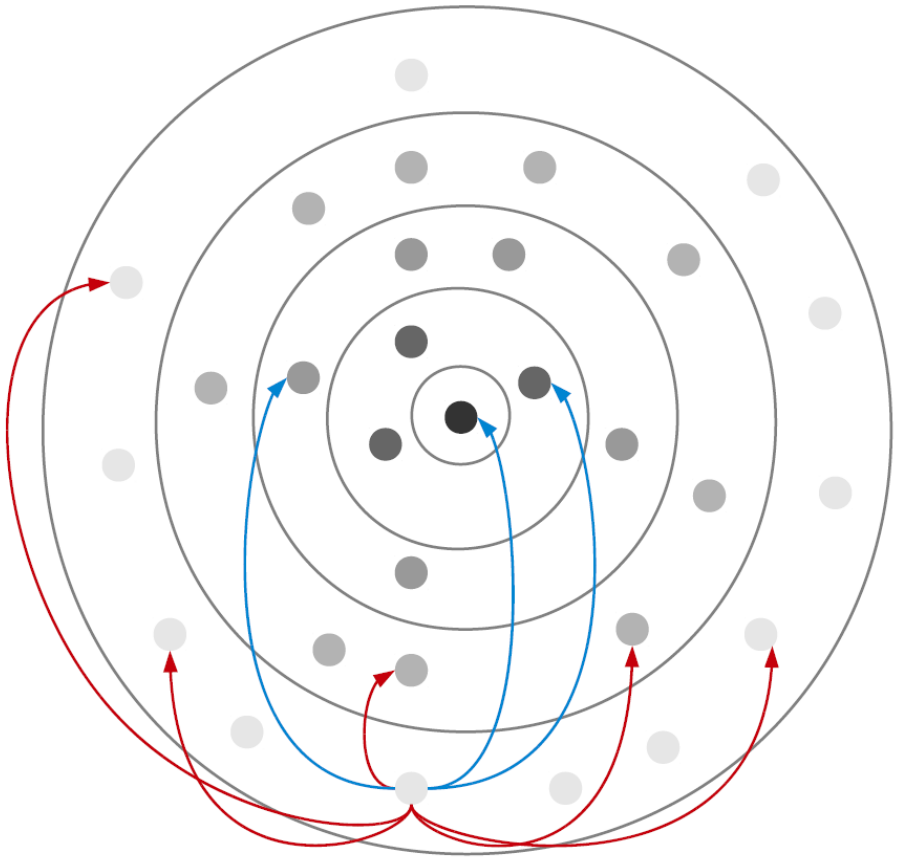
The Gradient Overlay Construction (1/2)

- A node maintains two sets of neighbours: random-view and similar-view.
- Random-view: a random sample of nodes in the system.
- Similar-view: a partial view of the nodes whose utility values are close to the utility value of this node.

The Gradient Overlay Construction (2/2)

- To construct the random-view we are using cyclon.
- To construct the similar-view, nodes periodically exchange their similar-views. Upon receiving a similar-view, a node updates its own similar-view by replacing its entries with those nodes that have closer (but higher) utility to its own utility value.
  - In the GradienTv we consider upload bandwidth for constructing the Gradient overlay.

Peers Partners



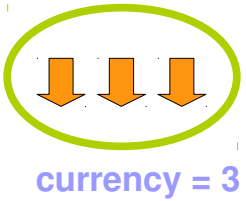
Similar-view pointer

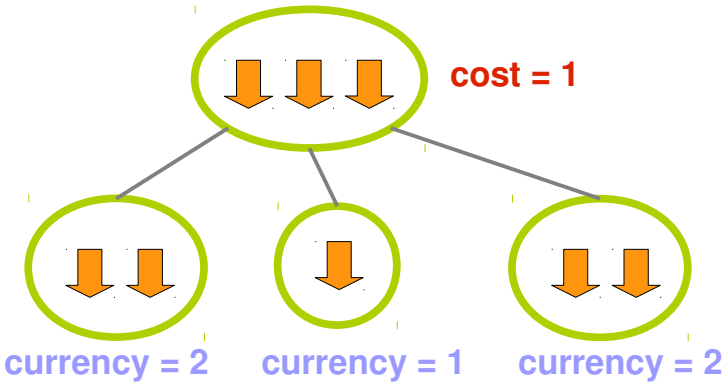
Finger pointer

Construction and Maintenance of The Streaming Overlay

Node Properties

- Currency: The the number of upload slots at a node.
- 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.



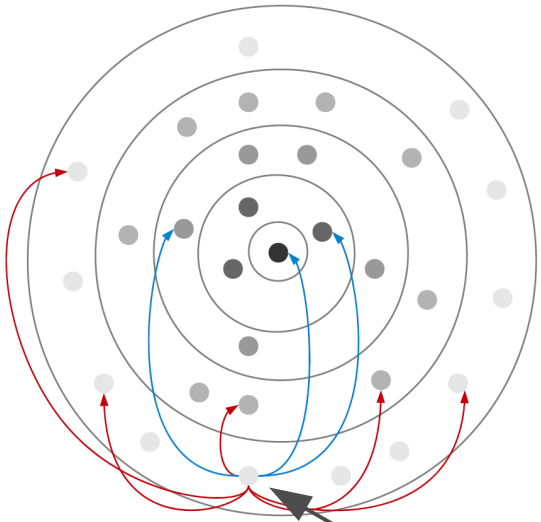


Streaming Overlay Construction

- Our market model is based on minimizing costs through nodes iteratively bidding for upload slots.

- The depth of a node in each tree is inversely proportional to its currency.
  - Nodes with higher currency end up closer to the media source, at the root of each tree.

The Market Model – Child Side



Q: Price: 1
Cost: 4

M: Price: 5
Cost: 4

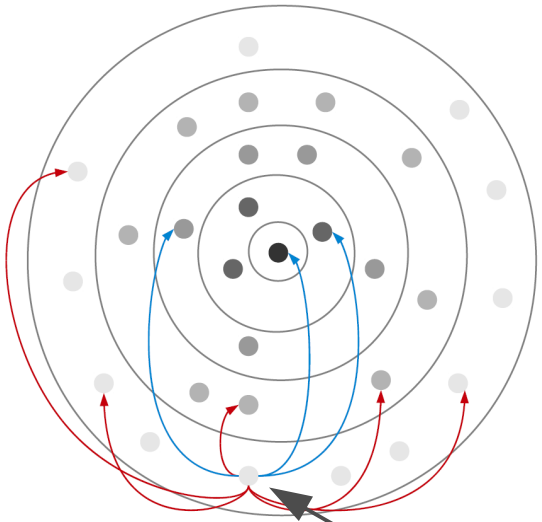
N: Price: 2
Cost: 6

X: Price: 3
Cost: 5

Currency: 3
Cost: 5

P

The Market Model – Child Side



Q: Price: 1
Cost: 4

M: Price: 5
Cost: 4

N: Price: 2
Cost: 6

X: Price: 3
Cost: 5

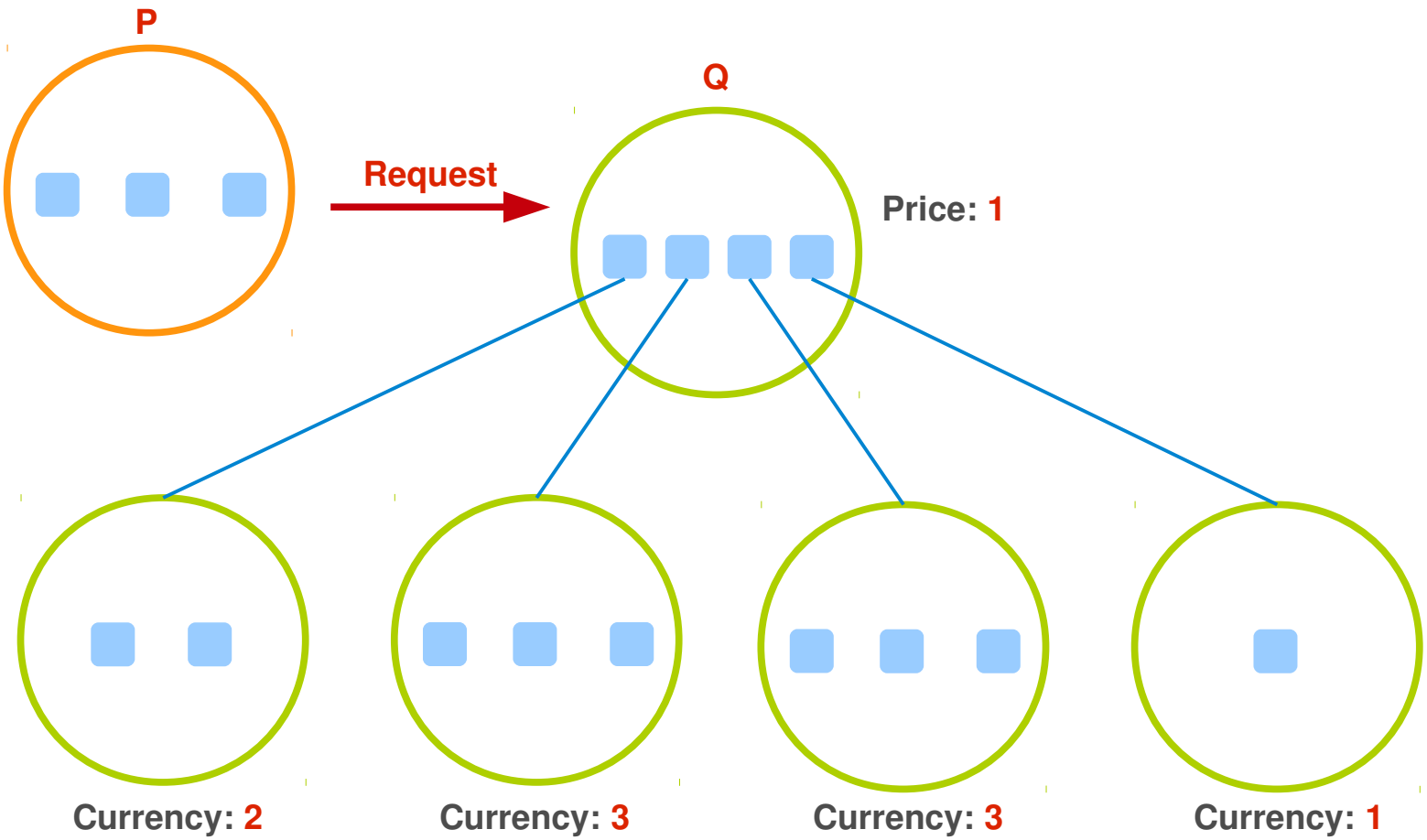
Currency: 3
Cost: 5

P

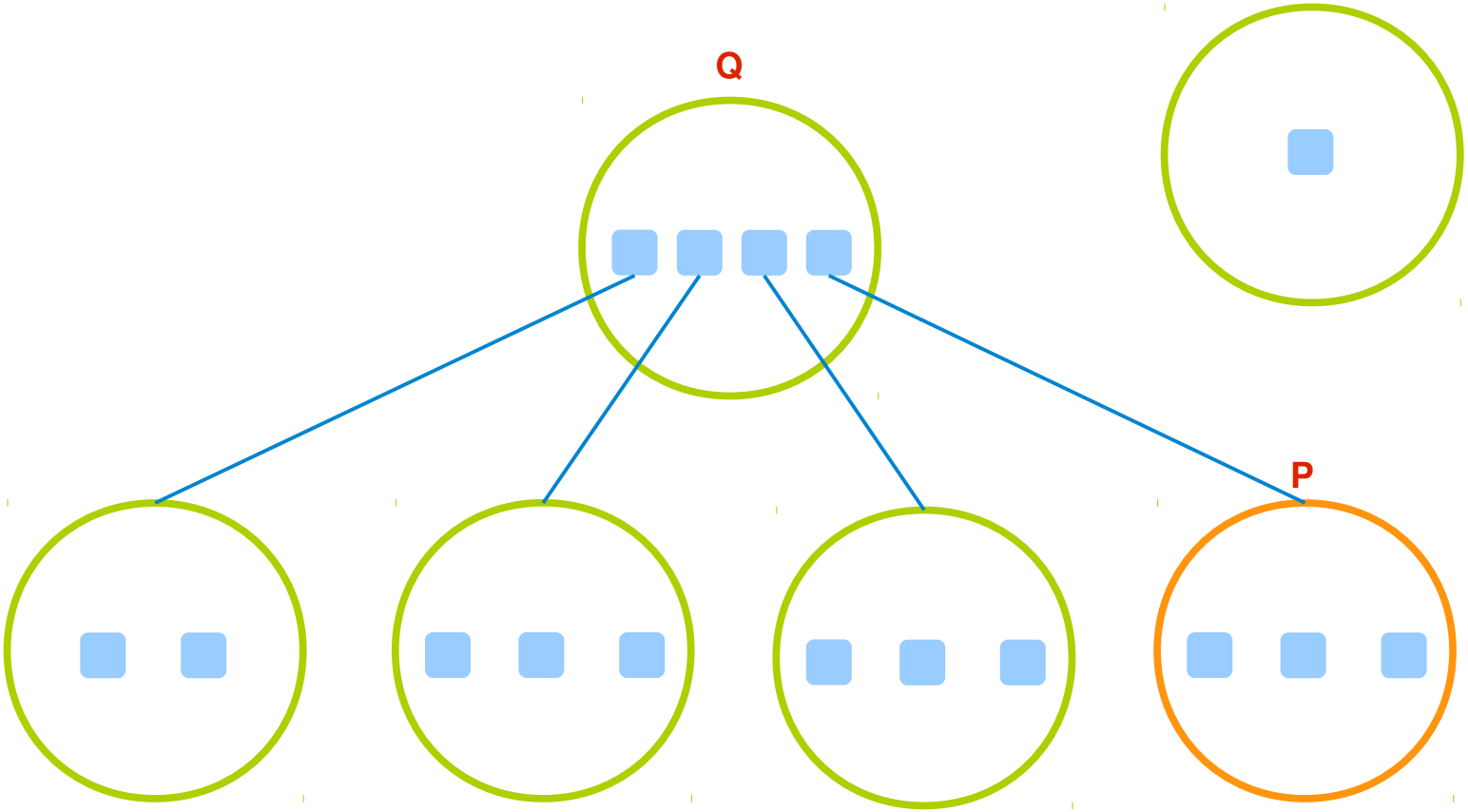
The Market Model – Child Side

- Node P periodically checks if it has a stripe who has not been assigned a parent or if it has a node in its view that has lower cost than its current parent.
- If any of these conditions is satisfied then P selects the nodes from its view whose cost for stripe i is lower than its current parent's cost and where P's currency is greater than the found node's price.
- P then uses a random policy to select a node from the candidate parents, and sends a request to it.

The Market Model – Parent Side



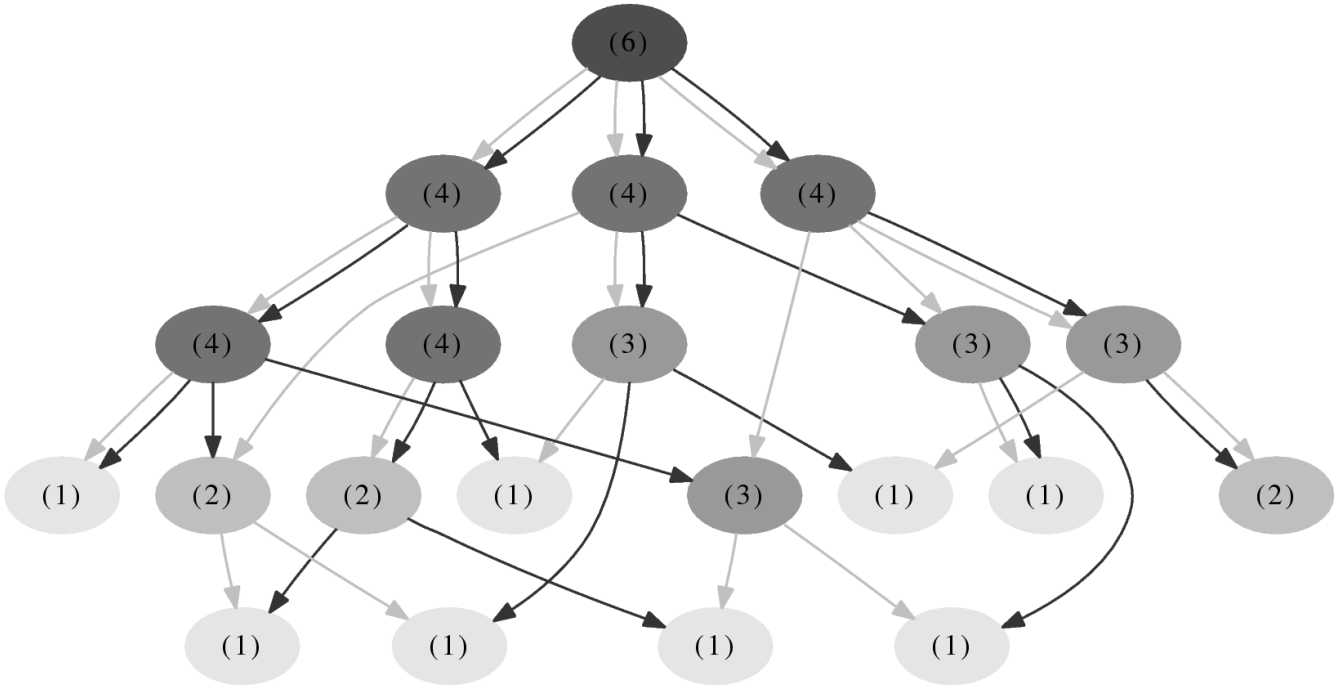
The Market Model – Parent Side



The Market Model – Parent Side

- After receiving a request from P, If Q has a free upload slot, it accepts the request.
- Otherwise if the currency of the requesting node P is greater than the price of Q, Q releases its child node with the lowest currency and accepts P as a new child.
  - In this case, the released node has to find another parent for its stripe.
- If Q's price is greater than P's currency, Q sends a not accepted message back to the P, and P has to find another parent in the next iteration.

Constructed Streaming Overlay

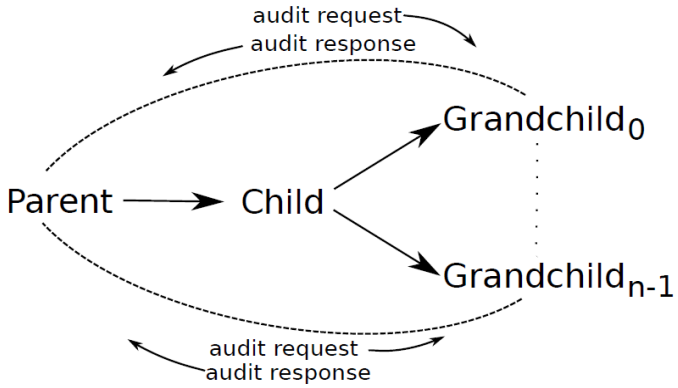


- Constructed 2-tree overlay.
- Darker nodes have more upload capacity than lighter ones.

Freeriders

Freerider Detector

- Freeriders are nodes that supply less upload bandwidth than claimed.
- The freerider detector component.
- Nodes identify freeriders through transitive auditing using their children's children.



Detecting Freeriders

- F is the sum of
  - the number of audit responses not received before a timeout.
  - the number of negative audit responses.
  - the free upload slots.
- If F is more than M% of claimed upload slots, Q is suspected as a freerider.
- If Q becomes suspected in N consecutive iterations, it is detected as a freerider.
- The higher the value of N, the more accurate but slower the detection is.

Punishment

- After detecting a node as a freerider, the parent node P, decreases its own price (P's price) to zero and considers the freerider node Q as its child with the lowest currency.
- On the next bid from another node, P replaces the freerider node with the new node.
- Freeriders can use the extra resources in the system without any punishment if they just join as a member of market-level one or two.

Evaluation

Evaluation Metrics

- Playback continuity: the percentage of the segments, which are received before their playback time.
- Playback latency: the difference between the playback point of a node and the playback point at the media source.

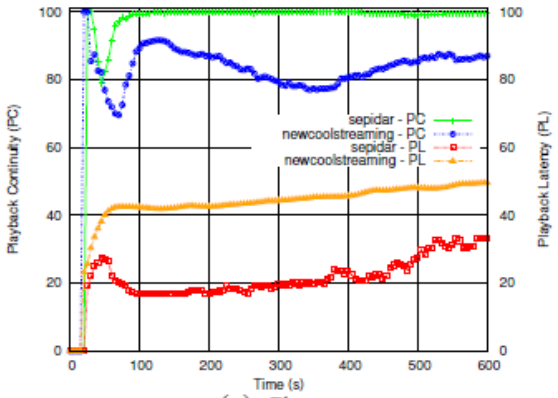
Experimental Setting

- We have done the experiments on the Kompics as a simulator platform.
- Latencies between nodes are modelled using a latency map based on the King dataset.
- The streaming rate to 512 Kbps, and it is split into 8 stripes, and each stripe is divided into a sequence of 128 Kb blocks.
- Nodes start playing the media after buffering it for 15 seconds.
- The upload bandwidth of nodes are from 128 Kbps to 1.25 Mbps.

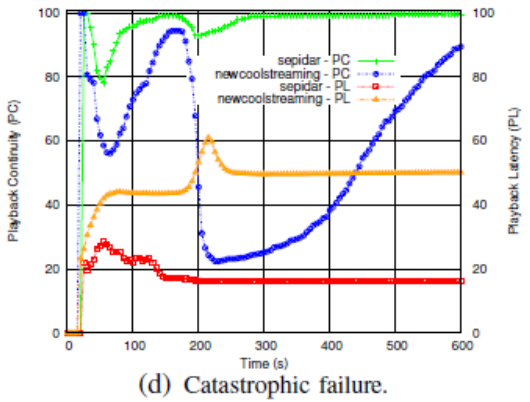
Scenarios

- Join only: 1000 nodes join the system following a Poisson distribution with an average inter-arrival time of 100 milliseconds;
- Flash crowd: first, 100 nodes join the system following a Poisson distribution with an average inter-arrival time of 100 milliseconds. Then, 1000 nodes join following the same distribution with a shortened average inter-arrival time of 10 milliseconds;
- Catastrophic failure: 1000 nodes join the system following a Poisson distribution with an average inter-arrival time of 100 milliseconds. Then, 500 existing nodes fail following a Poisson distribution with an average interarrival time 10 milliseconds.
- Churn: 500 nodes join the system following a Poisson distribution with an average inter-arrival time of 100 milliseconds, and then till the end of the simulations nodes join and fail continuously following the same distribution with an average inter-arrival time of 1000 milliseconds;
- Freerider: 1000 nodes join the system following a Poisson distribution with an average inter-arrival time of 100 milliseconds, such that 20% of the nodes are freeriders.

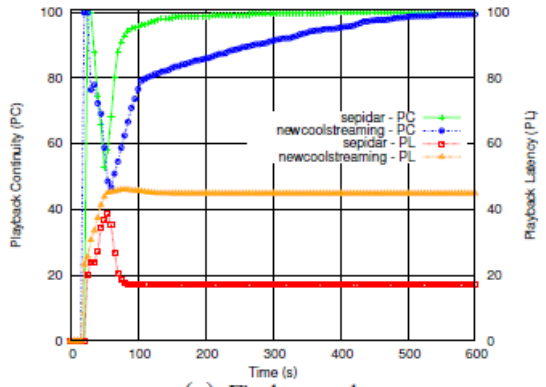
Sepidar vs NewCoolstreaming



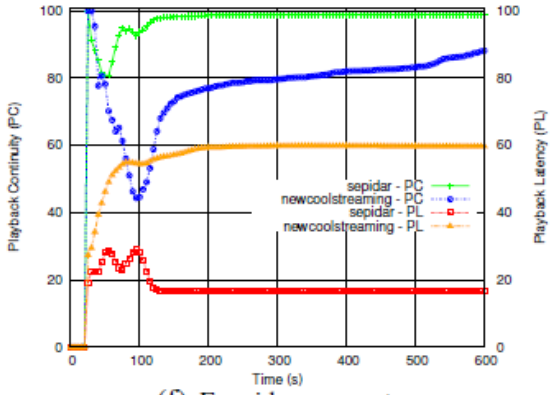
(c) Churn.



(d) Catastrophic failure.

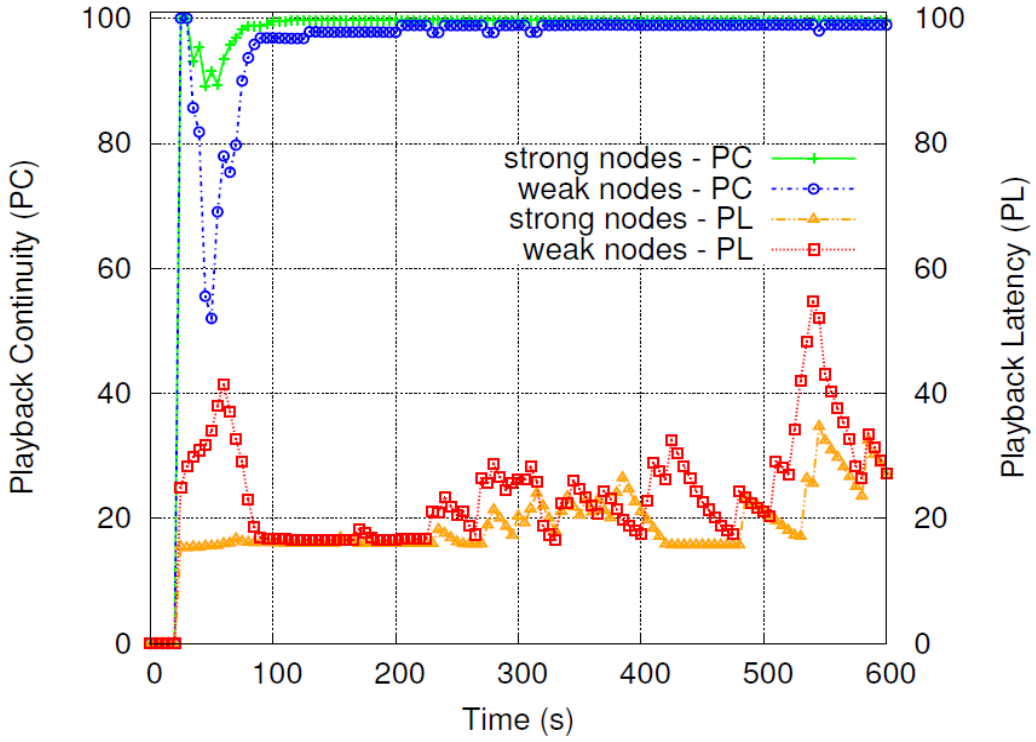


(e) Flash crowd.

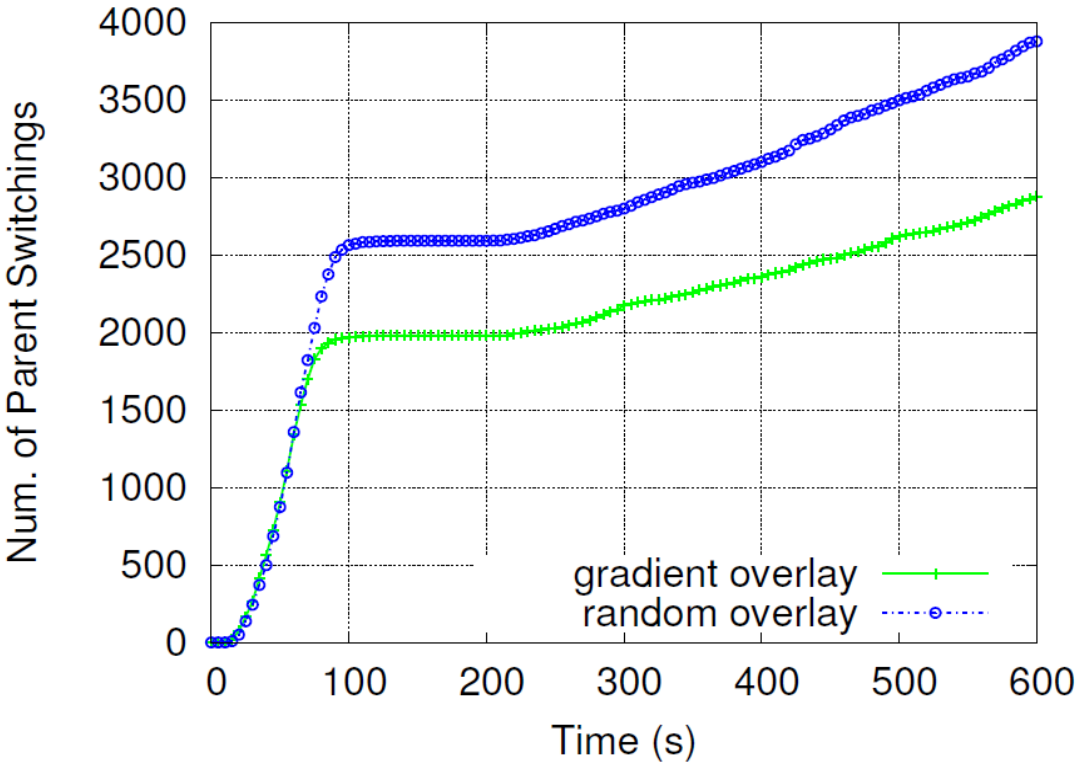


(f) Freeriders present.

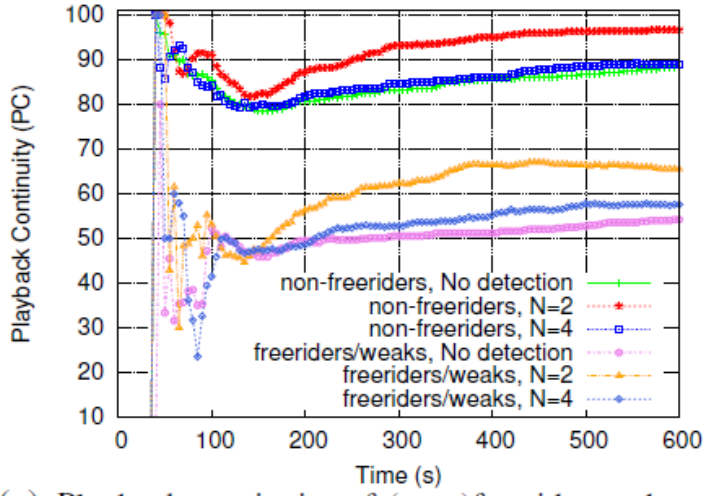
Incentivizing Nodes



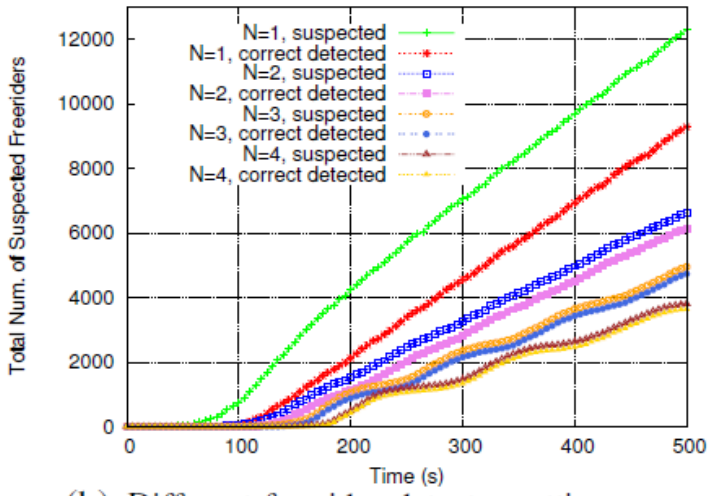
Gradient vs Random Network



Freerider Setting



(a) Playback continuity of (non-)freerider nodes.



(b) Different freerider detector settings.

Summary and Future Work

- Here, we presented Sepidar, a P2P live streaming system.
- It uses both the Gradient overlay and a distributed market-based approach to build multiple minimal height trees, where nodes with higher available upload bandwidth are positioned higher in the tree.
- We addressed the problem of free-riding through parent nodes auditing the behaviour of their children nodes by querying their grandchildren.

Question?