gradienTv: Market-based P2P live media streaming on the Gradient overlay

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

Big Picture

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

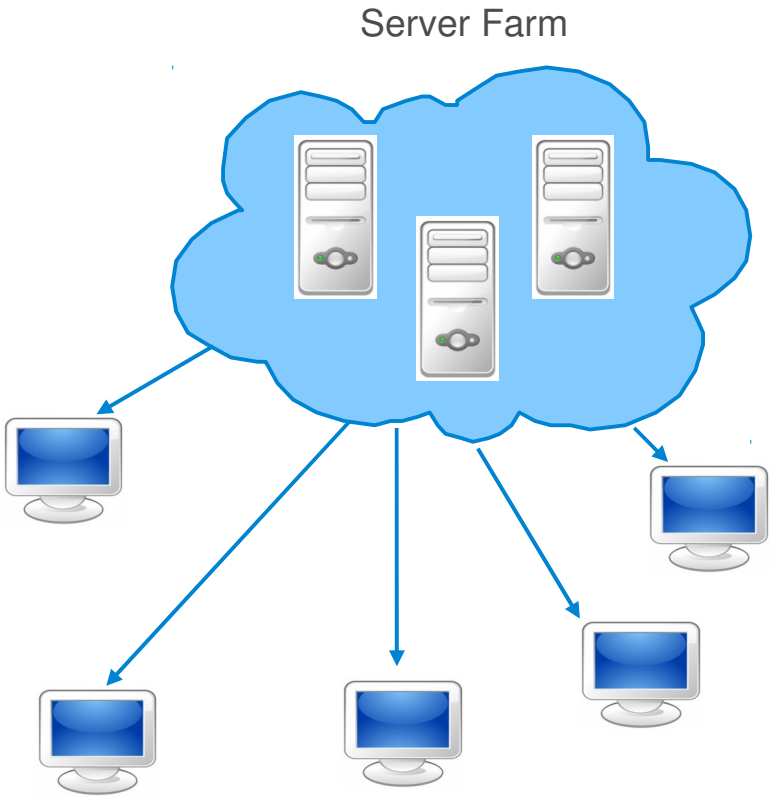
Motivation

Media Streaming

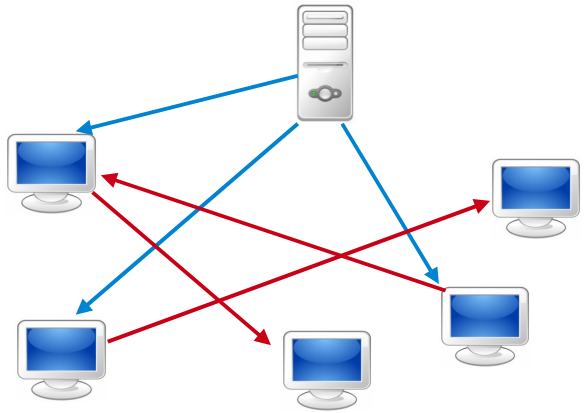
- Media streaming is a multimedia that is sent over a network and played as it is being received by end users.
- Users do not need to wait to download all the media.
- They can play it while the media is delivered by the provider.
- It could be:
  - Live streaming
  - Video on Demand (VoD)

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.

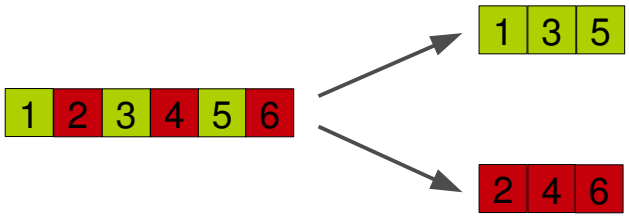
GradienTv

Two Questions on P2P Streaming

- What overlay topology is built for data dissemination (Data Delivery)?
- How to construct and maintain this overlay (Node Discovery)?

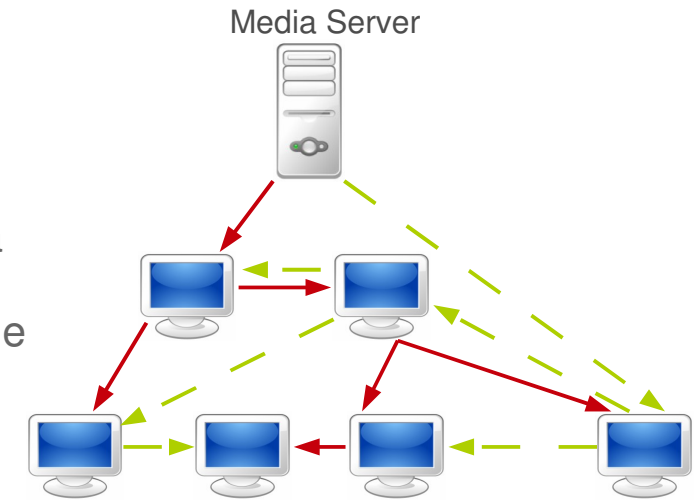
Data Delivery – Multiple-tree

- 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
  - A child node pulls the first block from its parent in a stripe tree.
  - The parent node pushes the rest of the blocks to the child.

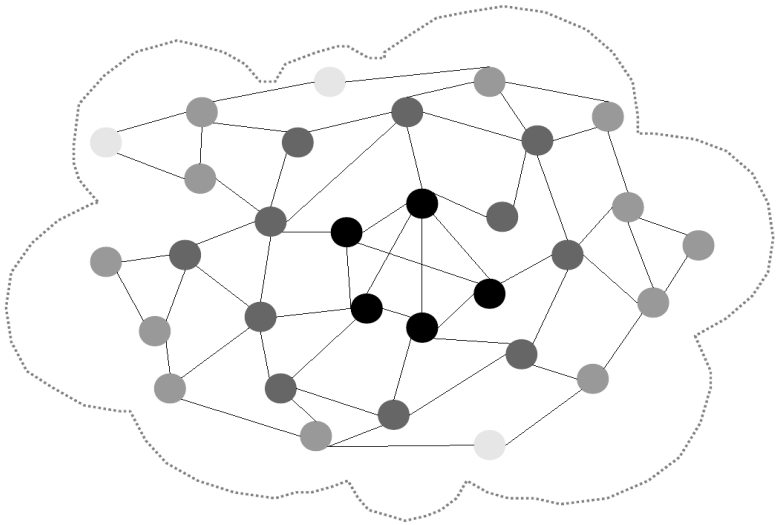


Node Discovery – Gossip-based method

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

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.



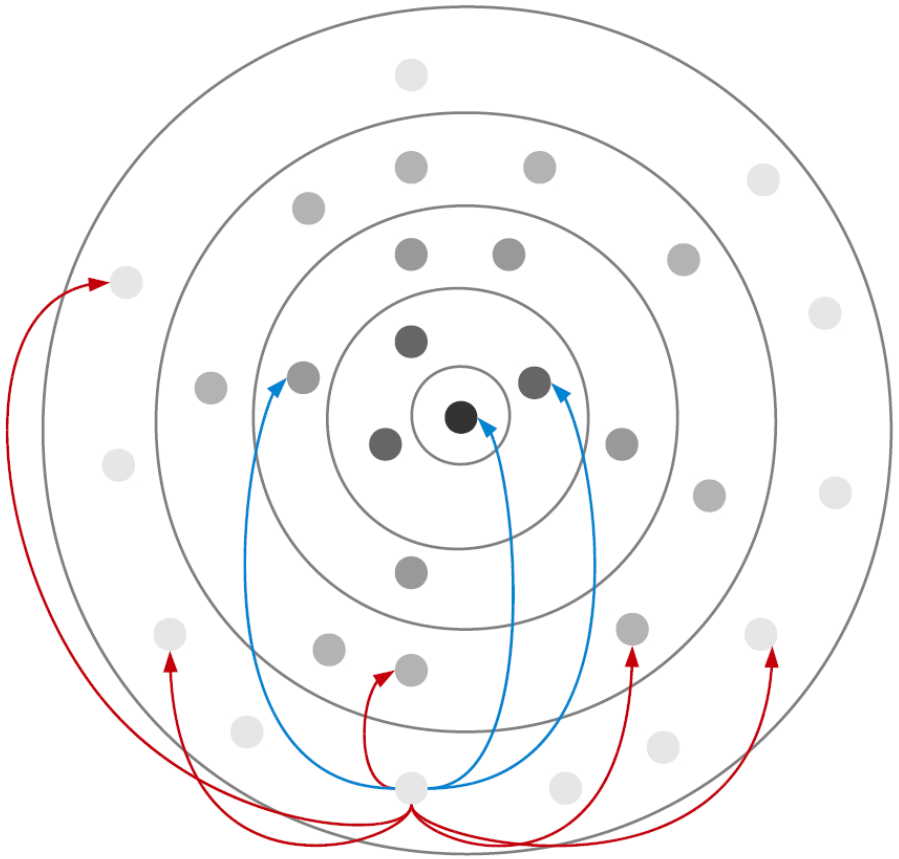
The Gradient Overlay Construction (1/2)

- 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



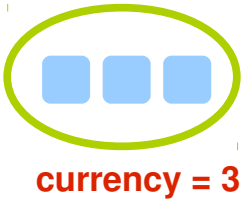
Similar-view pointer

Finger pointer

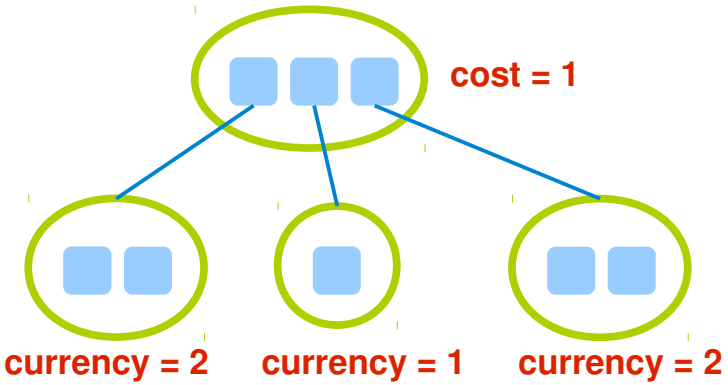
The Streaming Overlay Construction and Maintenance

Node Properties

- Upload capacity: The the number of upload slots at a node.
  - A node uses its upload capacity as currency.



- Connection cost: If a node has an unused upload slot its connection cost is zero, otherwise the node's connection cost is equal to the lowest upload capacity of its currently connected children.

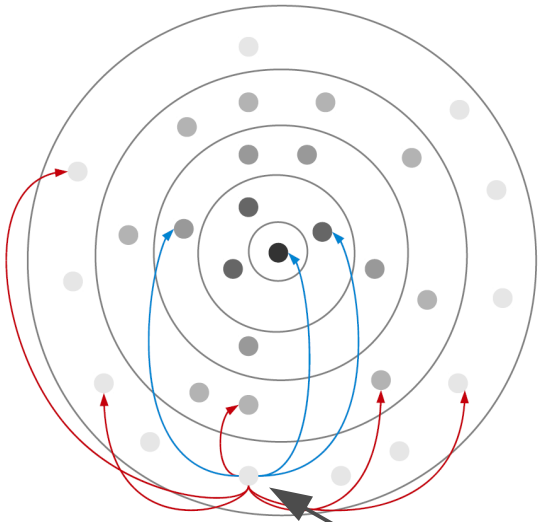


- Depth: The length of its path to the root.

Streaming Overlay Construction

- GradienTv uses a market-based approach to construct the overlay trees.
- One separate tree is created for each stripe.
- The depth of a node in each tree is inversely proportional to its currency.
  - Nodes with higher upload bandwidth end up closer to the media source, at the root of each tree.

The Market Model – Child Side



Q: Cost: 1
Depth: 4

M: Cost: 5
Depth: 4

N: Cost: 2
Depth: 6

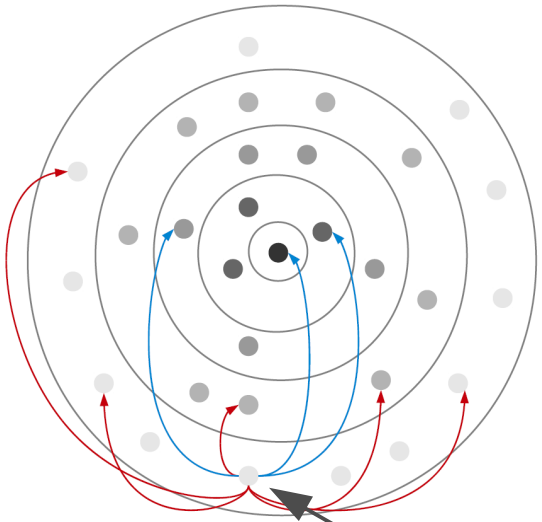
X: Cost: 3
Depth: 5

P's similar view and fingers

Currency: 3
Parent Depth: 5

P

The Market Model – Child Side



Q: Cost: 1
Depth: 4

M: Cost: 5
Depth: 4

N: Cost: 2
Depth: 6

X: Cost: 3
Depth: 5

P's similar view and fingers

Currency: 3
Parent Depth: 5

P

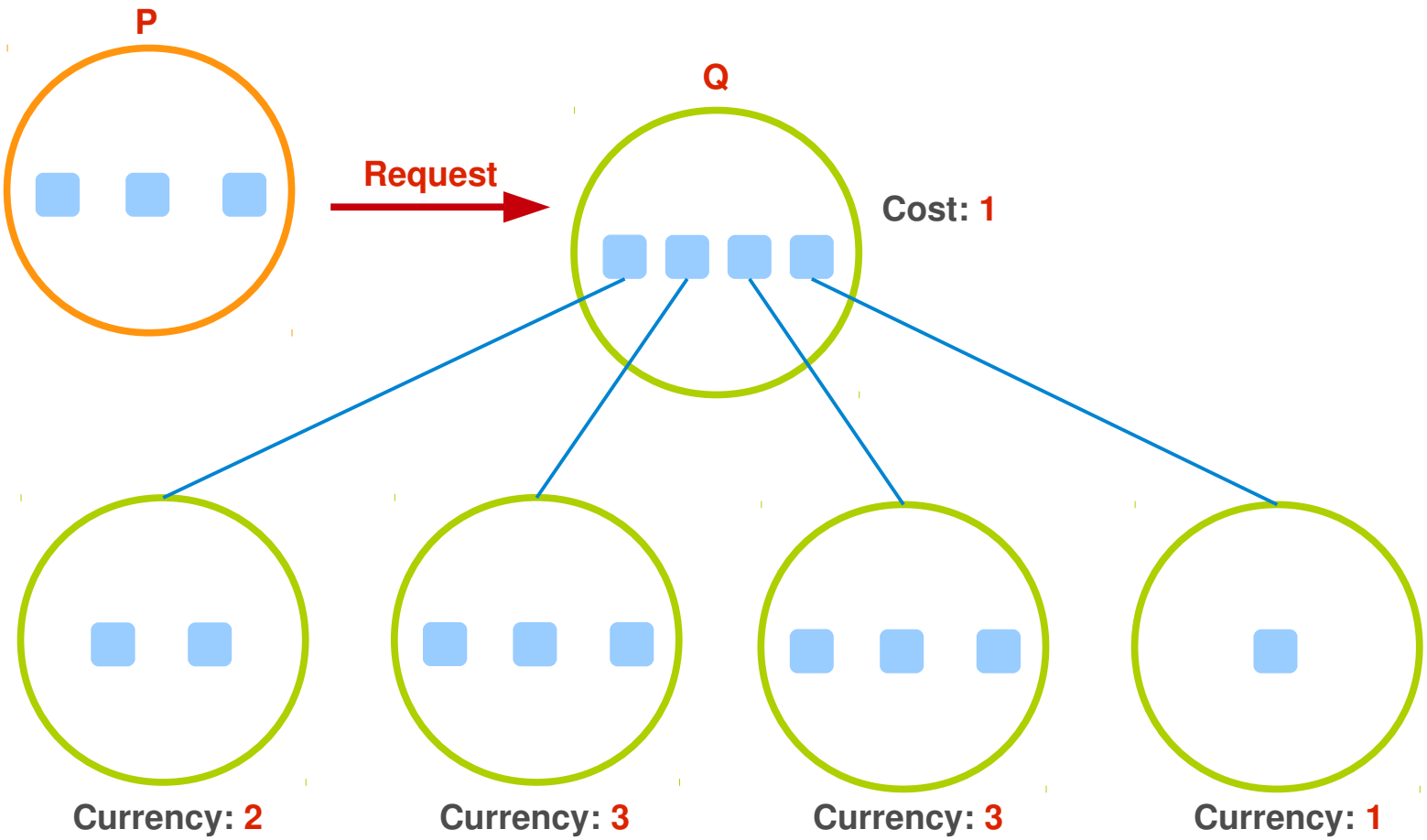
The Market Model – Child Side

- Node P periodically checks if it has a stripe who has not been assigned a parent or if it has a node in its similar-view and fingers that has lower depth than its current parent.

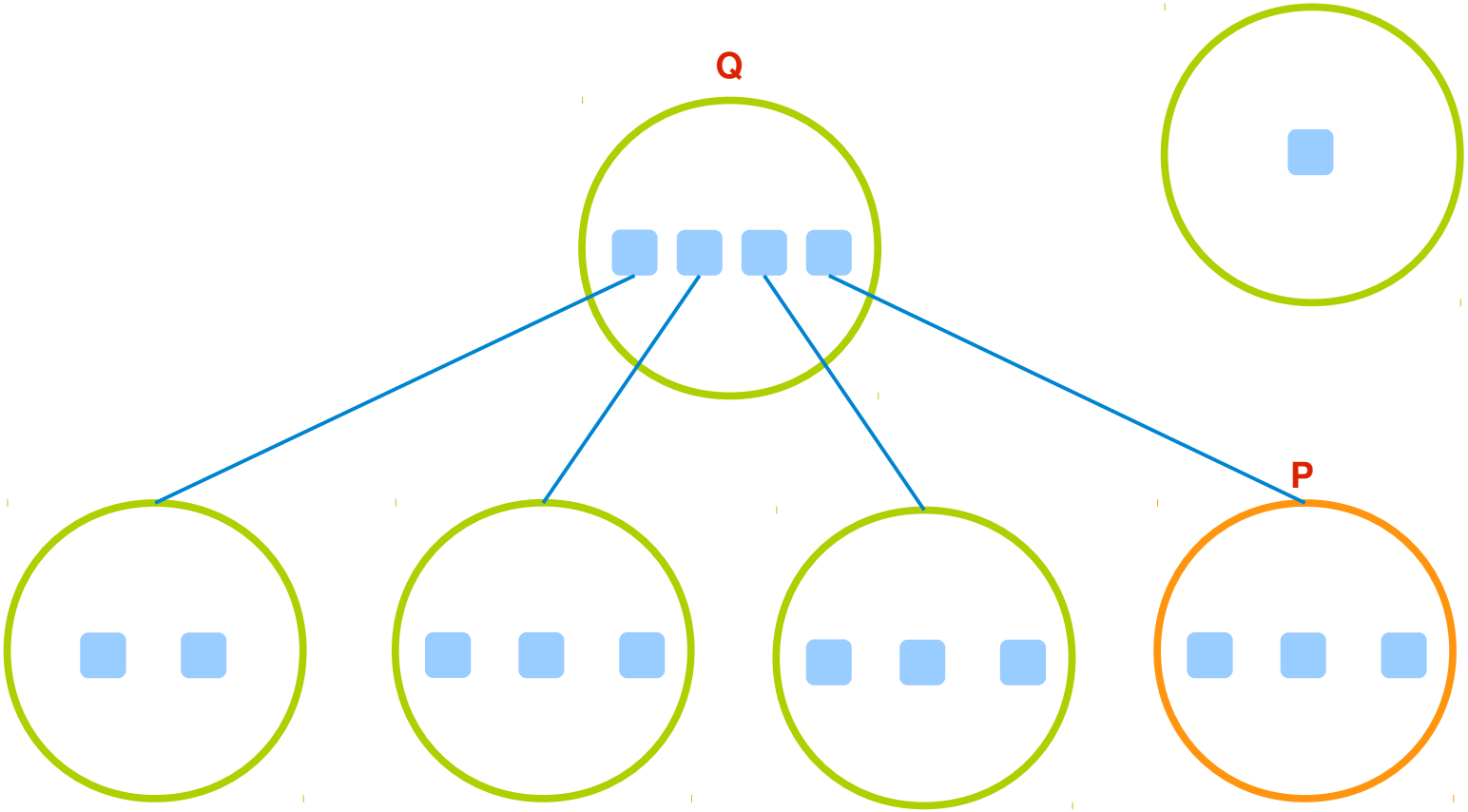
- If any of these conditions is satisfied then P selects the nodes from its similar-view and fingers whose depth for stripe i is lower than its current parent's depth and where P's currency is greater than the found node's connection cost.

- 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



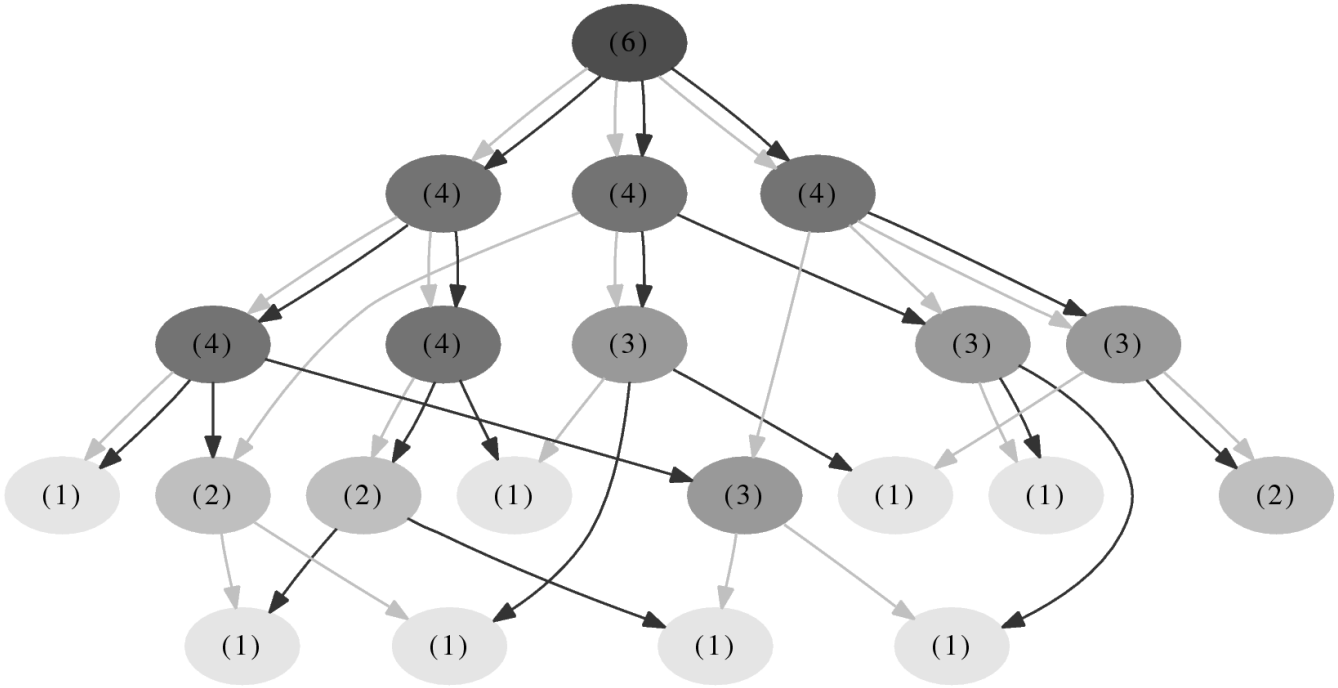
The Market Model – Parent Side



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 cost 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 cost 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.

Evaluation

Evaluation Metrics

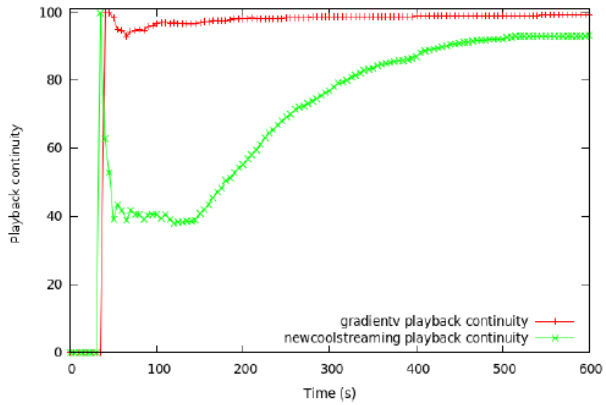
- Playback continuity: the percentage of the segments, which are received before their playback time.
- Bandwidth utilization: the ratio of the total utilized upload slots to the total demanded download slots.
- Playback latency: the difference between the playback point of a node and the playback point at the media source.
- Path length: the minimum distance between the media source and a node for a stripe.

Experimental Setting

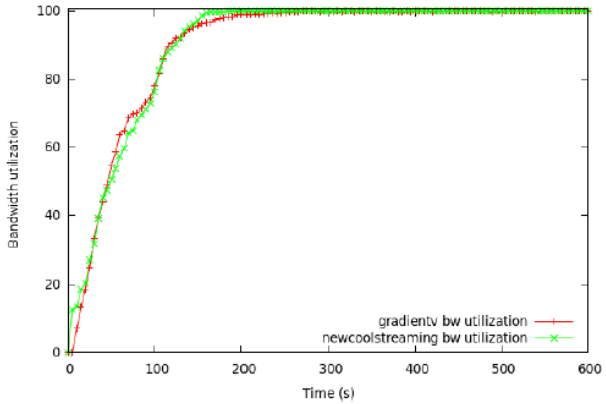
- We have done the experiments on the Kompics as a simulator platform.
- Latencies between nodes are modelled using a latency map based on the King dataset.
- The streaming rate to 512 Kbps, and it is split into 4 stripes, and each stripe is divided into a sequence of 128 Kb blocks.
- Nodes start playing the media after buffering it for 30 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.

Evaluation – Join Only Scenario

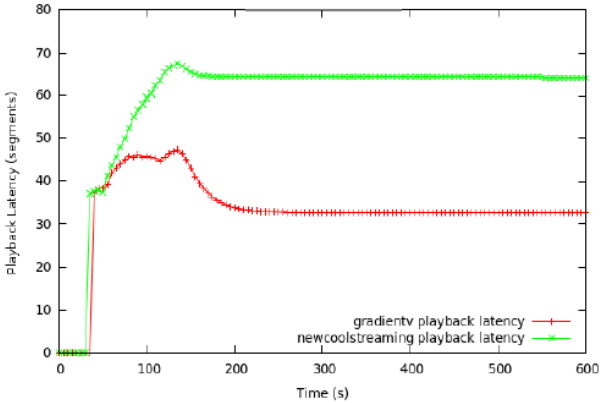
- 1000 nodes join the system following a Poisson distribution with an average inter-arrival time of 100 milliseconds.



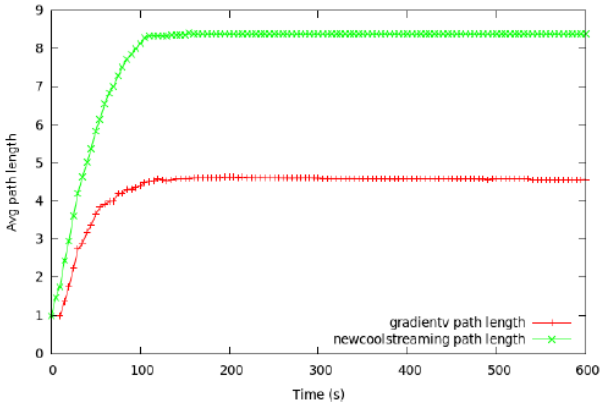
(a) Playback continuity against time



(b) Bandwidth utilization against time



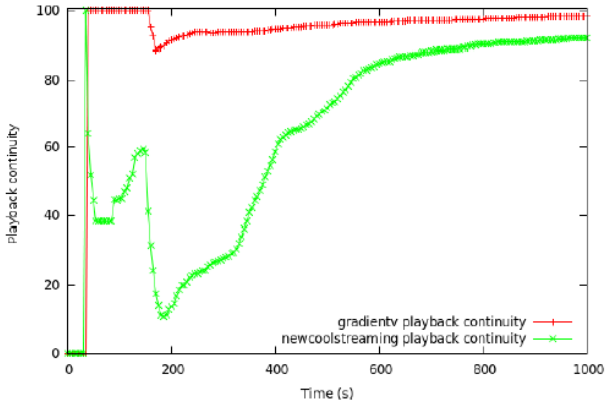
(c) Playback latency against time



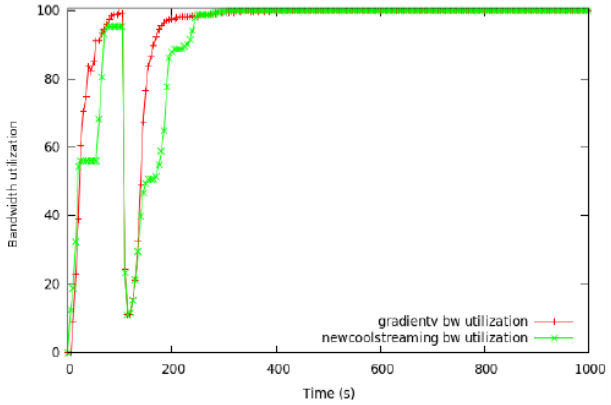
(d) Path length against time

Evaluation – Flash Crowd

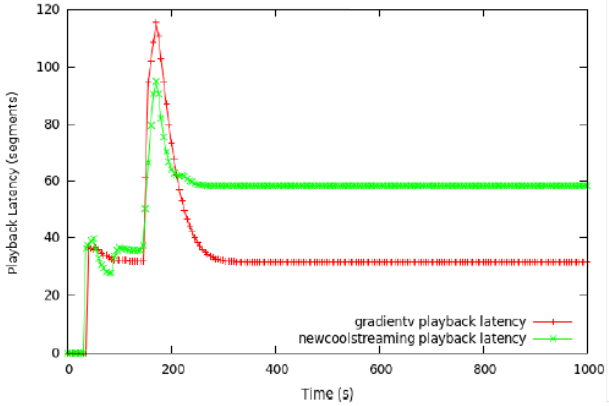
- 100 nodes join the system following a Poisson distribution with an average inter-arrival time of 100 milliseconds, and then 1000 other nodes join the system following a Poisson distribution with an average inter-arrival time of 10 milliseconds.



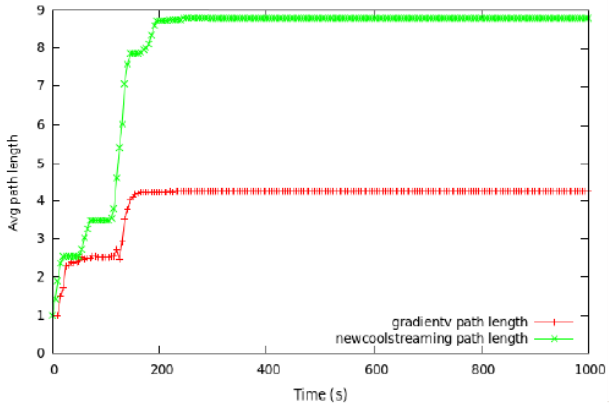
(a) Playback continuity against time



(b) Bandwidth utilization against time



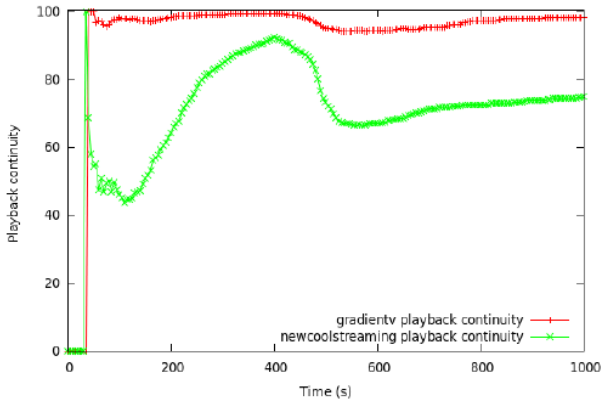
(c) Playback latency against time



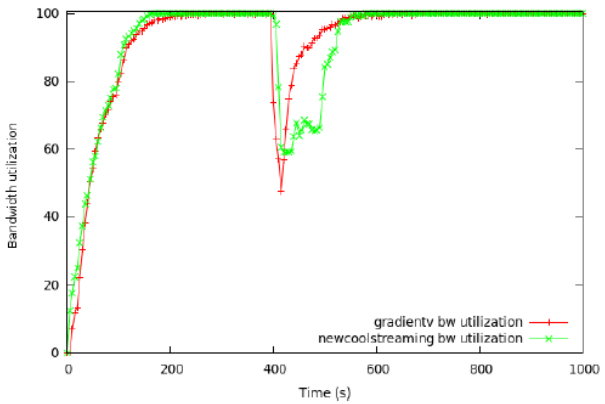
(d) Path length against time

Evaluation – Catastrophic Failure

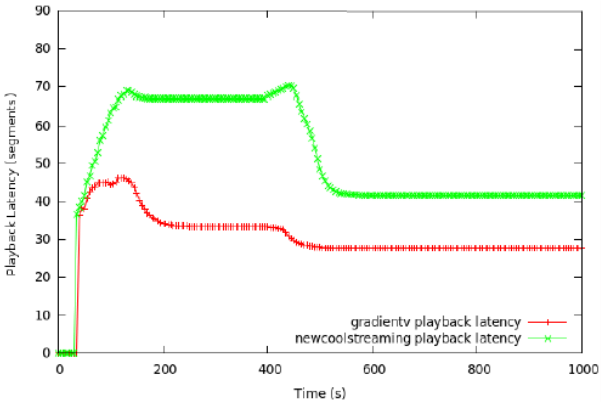
- 1000 other nodes join the system following a Poisson distribution with an average inter-arrival time of 100 milliseconds, and then 400 nodes fail following a Poisson distribution with an average inter-arrival time of 10 milliseconds.



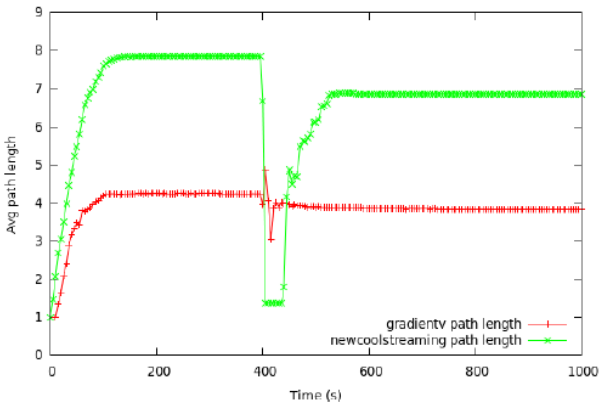
(a) Playback continuity against time



(b) Bandwidth utilization against time



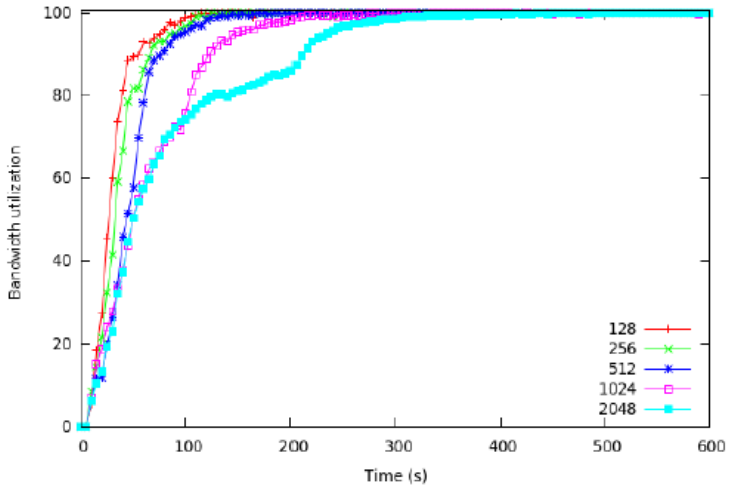
(c) Playback latency against time



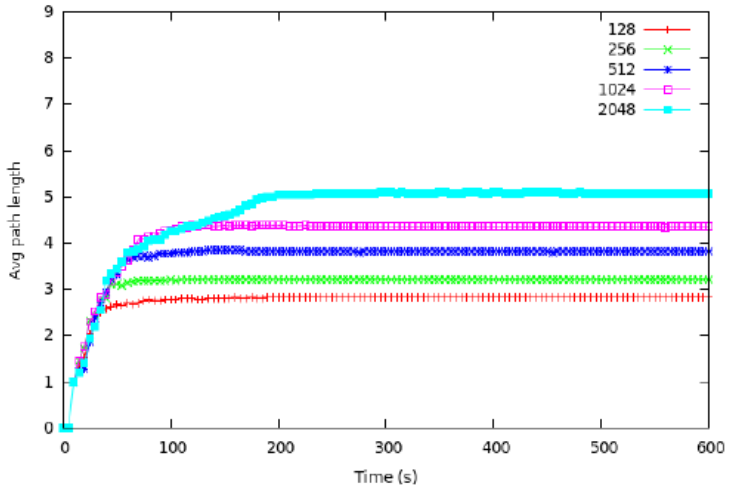
(d) Path length against time

Evaluation – Different Number of Nodes

- Different number of nodes join the system following a Poisson distribution with an average inter-arrival time of 100 milliseconds.



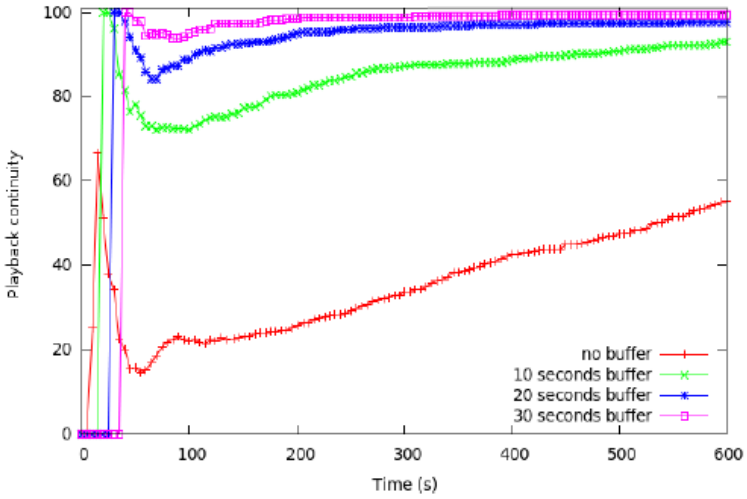
(a) Bandwidth utilization against time



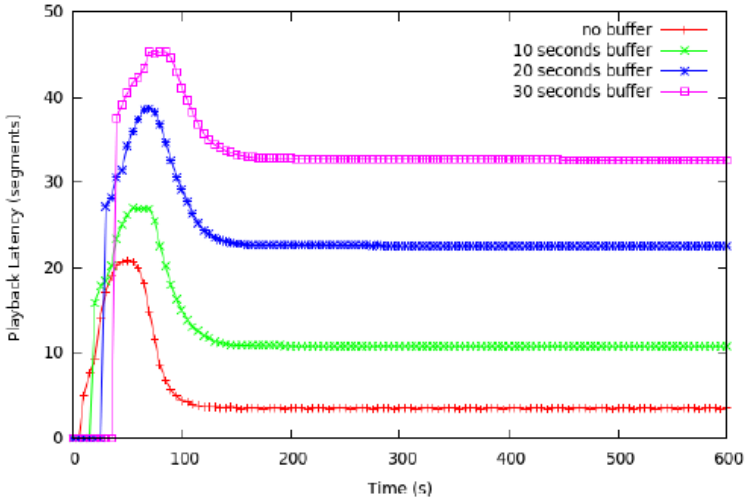
(b) Path length against time

Evaluation – Different Buffering Time

- 500 nodes join the system following a Poisson distribution with an average inter-arrival time of 100 milliseconds.



(a) Playback continuity against time



(b) Playback latency against time

Summary and Future Work

- Here, we presented gradienTv, a P2P live streaming system that uses both the Gradient overlay and a market-based approach to build multiple-tree streaming overlay.
- The constructed streaming trees had the property that the higher a node's upload capacity, the closer that node is to the root of the tree.
- Future work:
  - How to prevent free-riding?
  - How to deal with the dynamic number of upload slots (dynamic currency).

Question?