P2P Media Streaming

Amir H. Payberah (amir@sics.se)

Outline

- Introduction
- P2P media streaming
- Classification of P2P streaming systems
- Security in P2P streaming systems
- Sepidar/GLive – two P2P streaming systems

Outline

- Introduction
- P2P media streaming
- Classification of P2P streaming systems
- Security in P2P streaming systems
- Sepidar/GLive – two P2P streaming systems

Introduction

Media Streaming

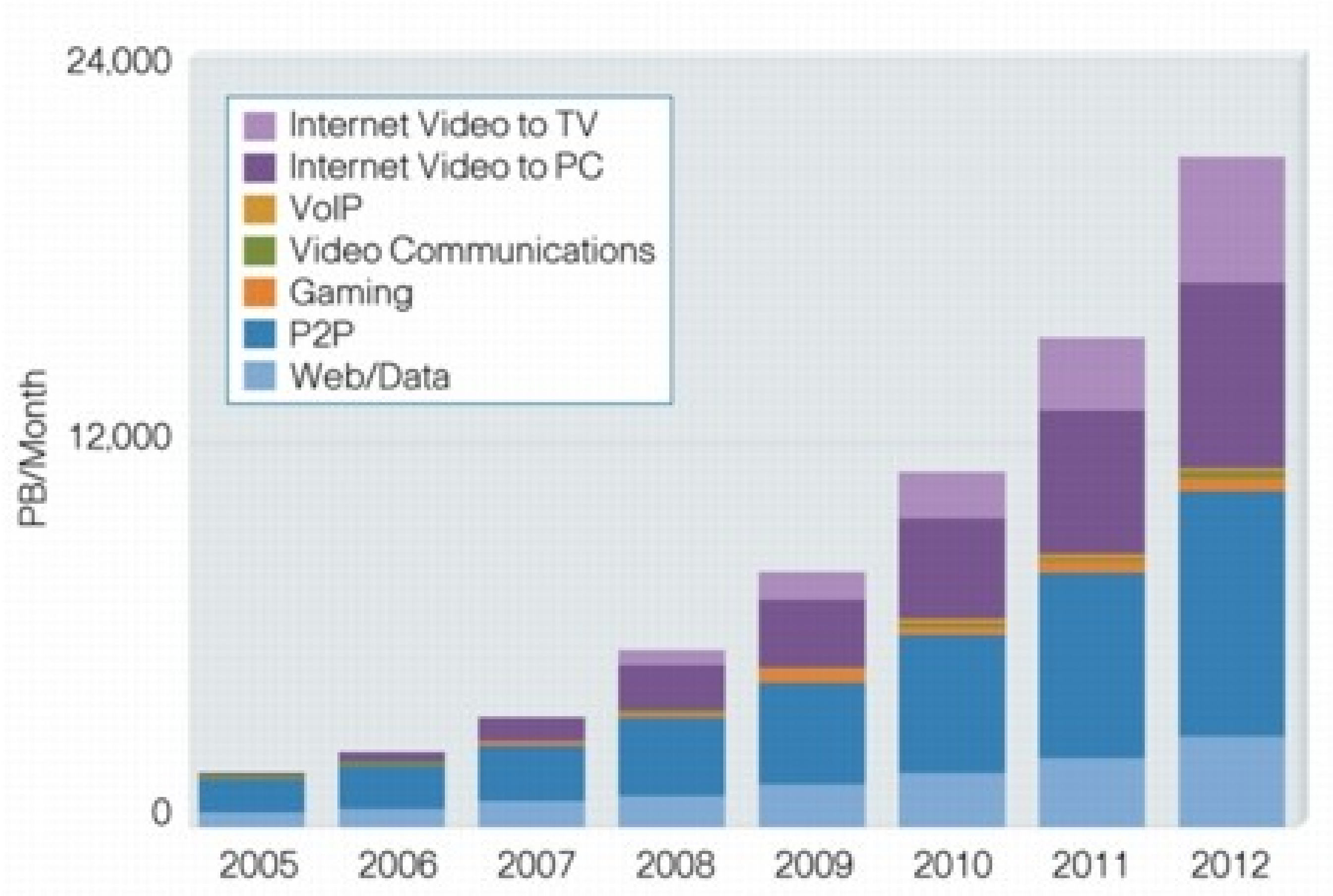
- Media streaming is multimedia that is sent over a network and played as it is being received by end users.
- Users do not need to wait to download all the media.
- They can play it while the media is being delivered by the provider.

Media Streaming

- Live Media Streaming
  - The streams are only available at a particular instant in time.

- Video on Demand (VoD)
  - The streams are stored on a server and are available to be transmitted at a user's request.
  - It provides a large subset of VCR functionality, e.g., pause, fast forward, fast rewind and ...

Media Streaming Trend

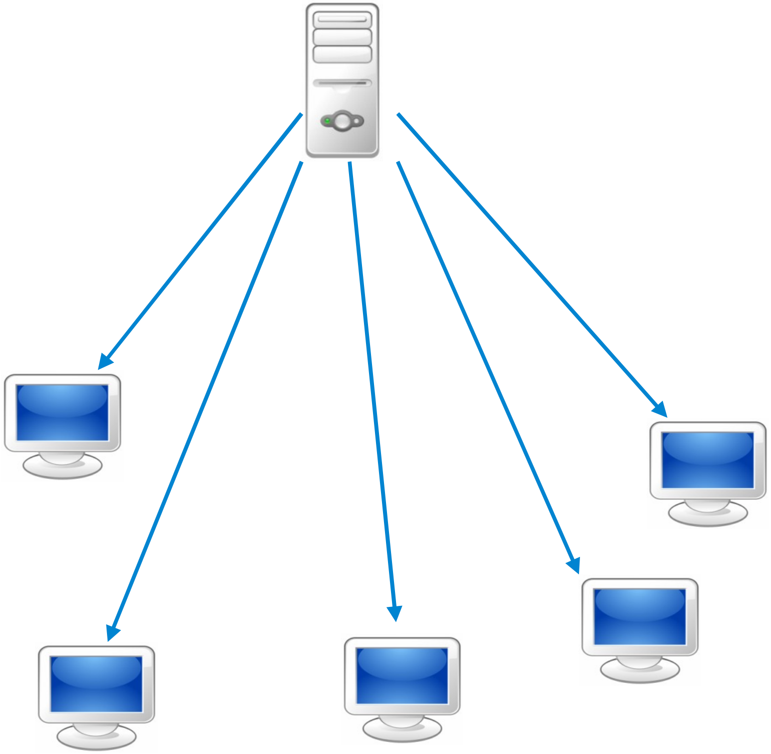


Cisco's global consumer Internet traffic forecast

Solutions for Media Streaming

- Client-Server solution

Client – Server



Client – Server

- What is the problem with the Client-Server model for media streaming? [d]

Client – Server

- What is the problem with the Client-Server model for media streaming?

- Scalability
- Single point of failure

Client – Server

- What is the problem with the Client-Server model for media streaming?

- ~~Scalability~~
- ~~Single point of failure~~

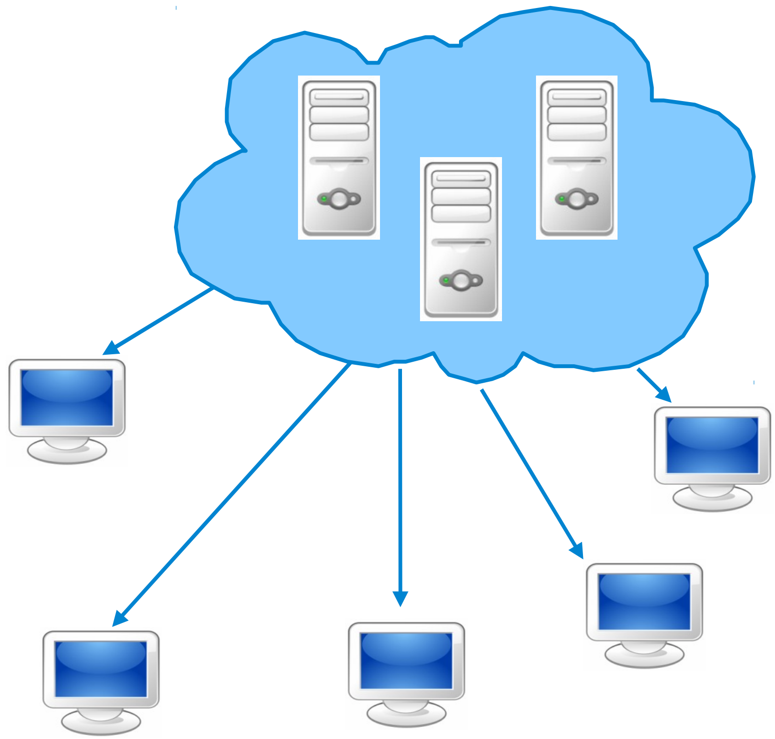
- Providing a scalable service, which is resistant to failure is very expensive.

Client – Server

Akamai

YouTube

Distributed servers
Content Delivery Network (CDN)

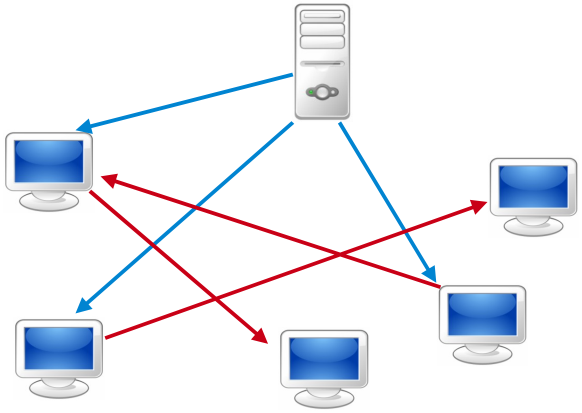


Solutions for Media Streaming

- Client-Server solution
- Peer-to-Peer solution

Peer-to-Peer

- The peers can help each other.
- The peers who have parts of the data can forward it to other requesting peers.
- The capacity increases with the number of peers.



Outline

- Introduction
- P2P media streaming
- Classification of P2P streaming systems
- Security in P2P streaming systems
- Sepidar/GLive – two P2P streaming systems

P2P Media Streaming

P2P Media Streaming Challenges

- Data should be received with respect to certain timing constraints.
  - A negligible startup delay
  - Smooth playback
  - A negligible playback latency (only for Live Streaming)

P2P Media Streaming Challenges

- 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

P2P Media Streaming Challenges

- 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.

P2P Media Streaming Challenges

- 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.
- Free-riding problem.

P2P Media Streaming Challenges

- 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.
- Free-riding problem.
- Connectivity Problem.
  - NAT problem.

Main Questions

- What type of overlay topology is useful for data dissemination?
- What algorithm is used for data dissemination?
- How do we construct and maintain this target overlay topology?

Main Questions

- What type of overlay topology is useful for data dissemination?
- What algorithm is used for data dissemination?
- How do we construct and maintain this target overlay topology?

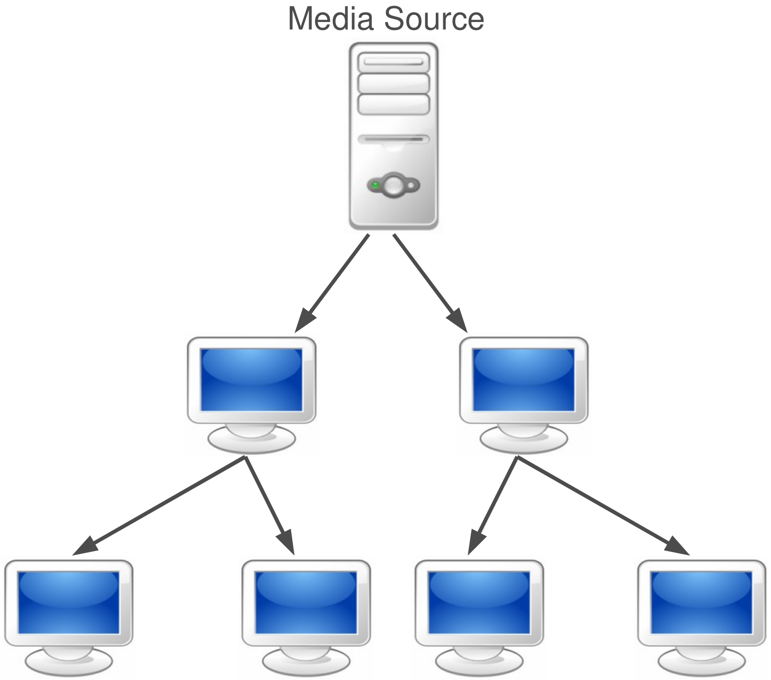
Data Dissemination Overlay

- What overlay topology do we build to distribute data messages?

- Some possibilities include:
  - Single tree
  - Multiple tree
  - Mesh

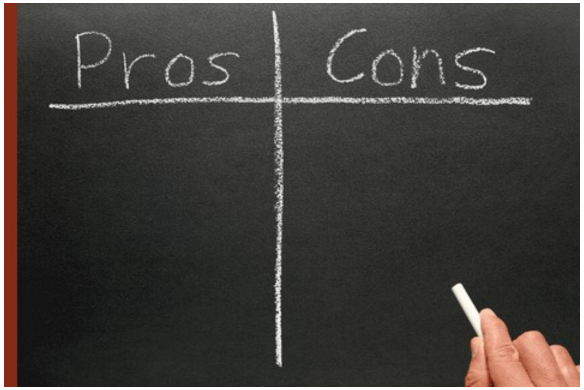
Single Tree Structure

- Build a single multicast tree, in which the root is the media source and the interior nodes and leaves are peers.



Single Tree Advantage/Disadvantage?

- Advantage/Disadvantage [d]

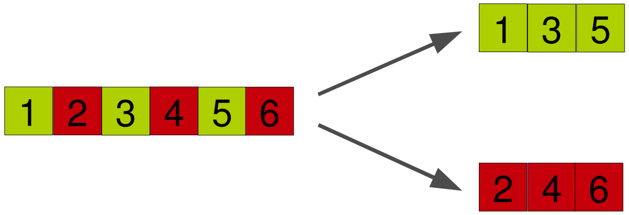


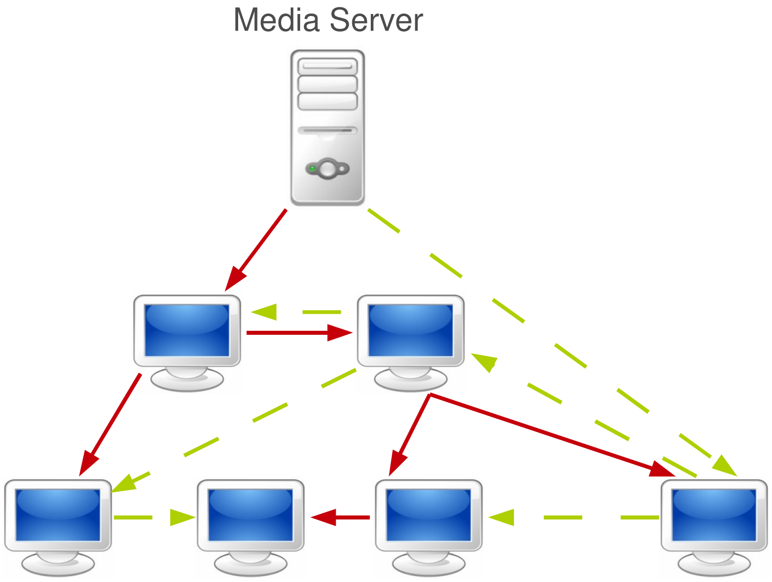
Single Tree Advantage/Disadvantage?

- Advantage
  - The short latency of data delivery
  - Easy to implement
- Disadvantage
  - The fragility of the tree structure upon the failure of nodes close to the root
  - All the traffic is only forwarded by the interior nodes

Multiple-Tree Structure

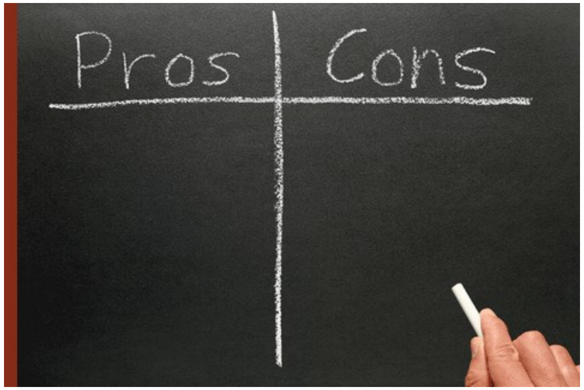
- The media source splits the stream into a set of sub-streams.
- A single tree is created for each sub-stream.
- A peer to receive the whole media should join all trees.





Multiple-Tree Advantage/Disadvantage?

- Advantage/Disadvantage [d]



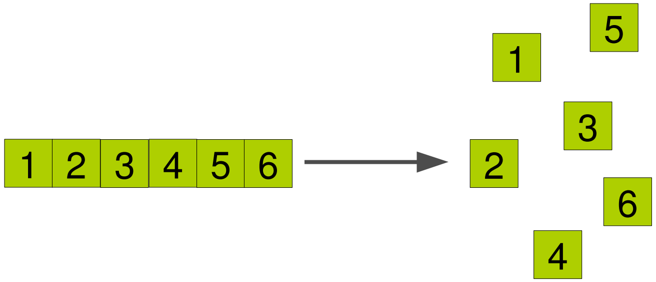
Multiple-Tree Advantage/Disadvantage?

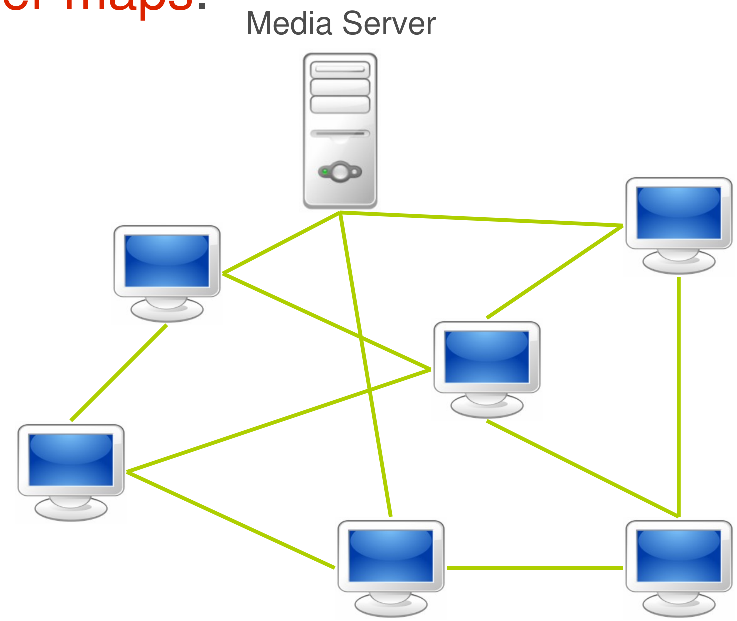
- Advantage
  - Resilient to node failure
  - Good load balancing

- Disadvantage
  - Difficult to implement
  - If a node fails, the sub-tree rooted at that node does not receive data, while they rejoin the system again

Mesh-based Structure

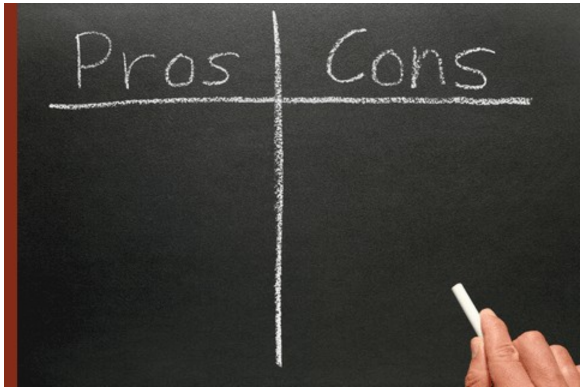
- The media source into small blocks.
- Nodes are connected in a mesh-network.
- Nodes periodically exchange their buffer maps.





Mesh Advantage/Disadvantage?

- Advantage/Disadvantage [d]



Mesh Advantage/Disadvantage?

- Advantage
  - Resilient to node failure
  - Good load balancing
  - Easy to implement

- Disadvantage
  - Unpredictable latencies due to the frequent exchange of notifications and requests

Main Questions

- What type of overlay topology is useful for data dissemination?
- What algorithm is used for data dissemination?
- How do we construct and maintain this target overlay topology?

Data Dissemination Algorithms

- How to distribute data messages.
- It could be:
  - Push-based
  - Pull-base
  - Push-Pull-based

Push-based Data Dissemination

- A node actively pushes a received block to its neighbours.
- Mostly used in tree-based overlays.
- What about mesh-based overlays? [d]

Push-based Data Dissemination

- A node actively pushes a received block to its neighbours.
- Mostly used in tree-based overlays.
- What about mesh-based overlays?
  - Redundant messages: a node might blindly push a block to a node already has that block.

Pull-based Data Dissemination

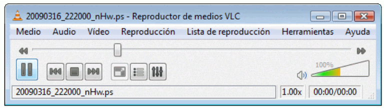
- Nodes periodically exchange data availability (buffer maps).
- After receiving a buffer map, a node can decide and schedule to pull which block from which node.
- Mostly used in mesh-based overlays.

Pull-based Data Dissemination

- Nodes periodically exchange data availability (buffer maps).
- After receiving a buffer map, a node can decide and schedule to pull which block from which node.
- Mostly used in mesh-based overlays.

In order
Rarest first
Hybrid

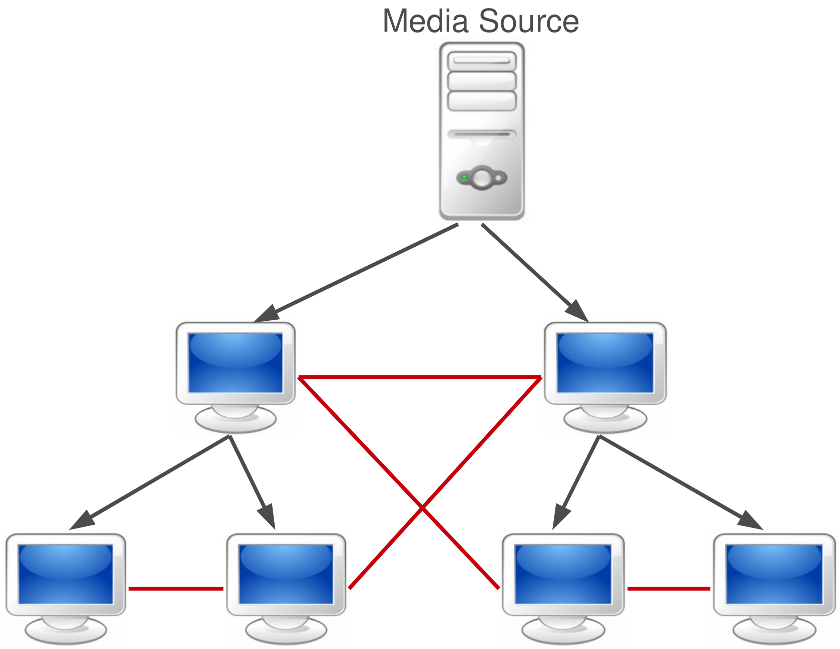
Will be back to this



20090316_222000_nHw.ps 1.00x 00:00/00:00

Push-Pull-based Data Dissemination

- Usually blocks are pushed through a tree and missed blocks are pulled from the mesh neighbours.



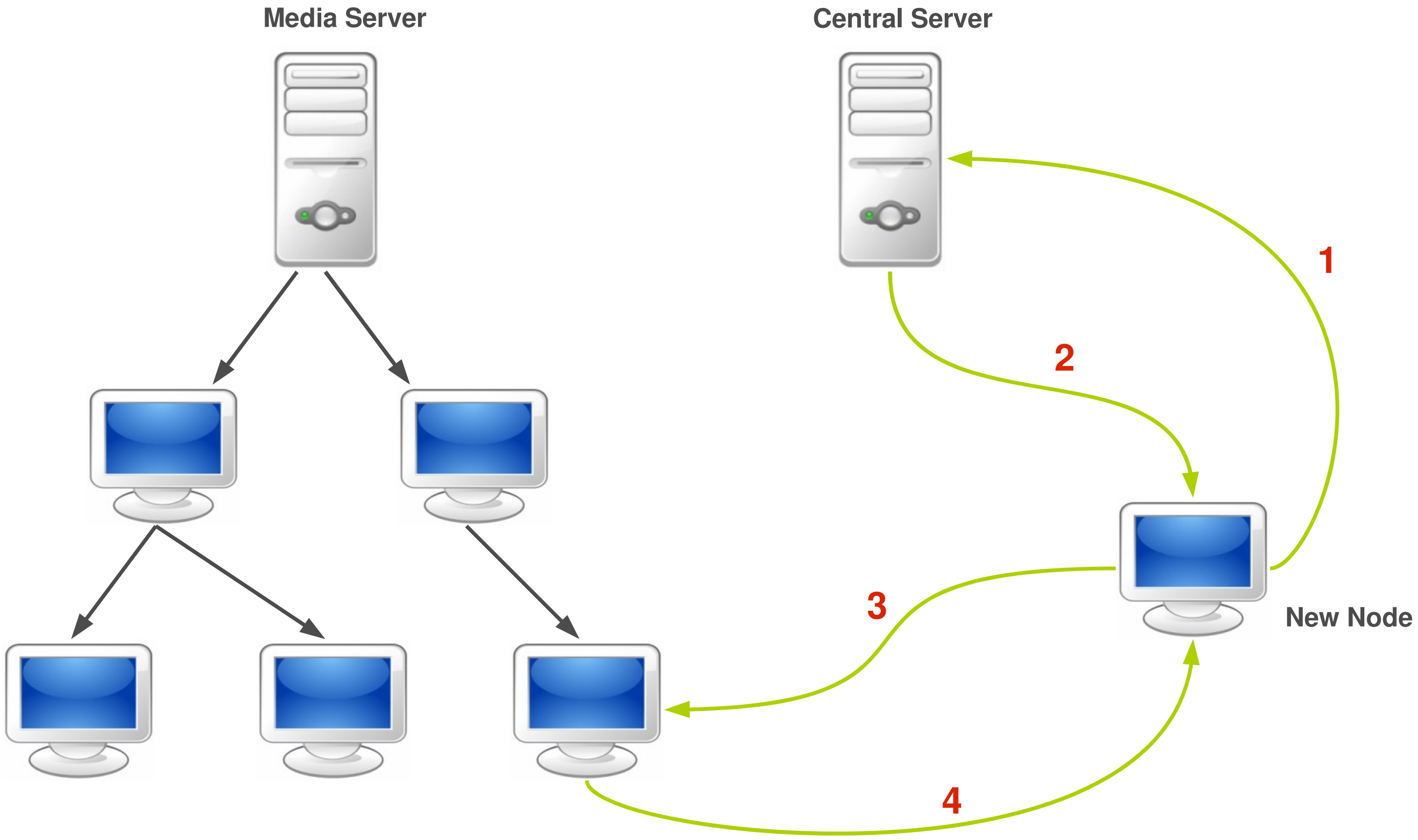
Main Questions

- What type of overlay topology is useful for data dissemination?
- What algorithm is used for data dissemination?
- How do we construct and maintain this target overlay topology?

The Overlay Construction and Maintenance

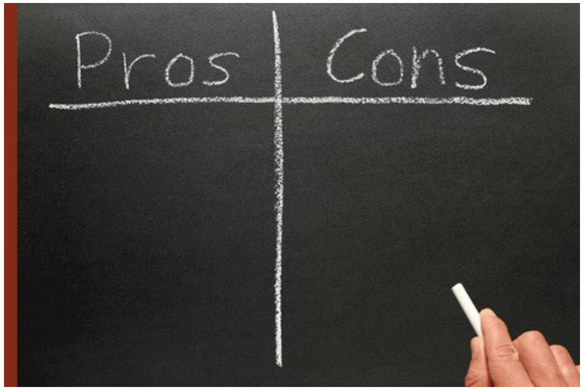
- How to build and maintain the data distribution overlay.
- Using the control messages for this purpose.
- It could be:
  - Centralized
  - Hierarchical
  - DHT-based
  - Control flooding
  - Gossip-based

Centralized Method



Centralized Advantage/Disadvantage?

- Advantage/Disadvantage [d]

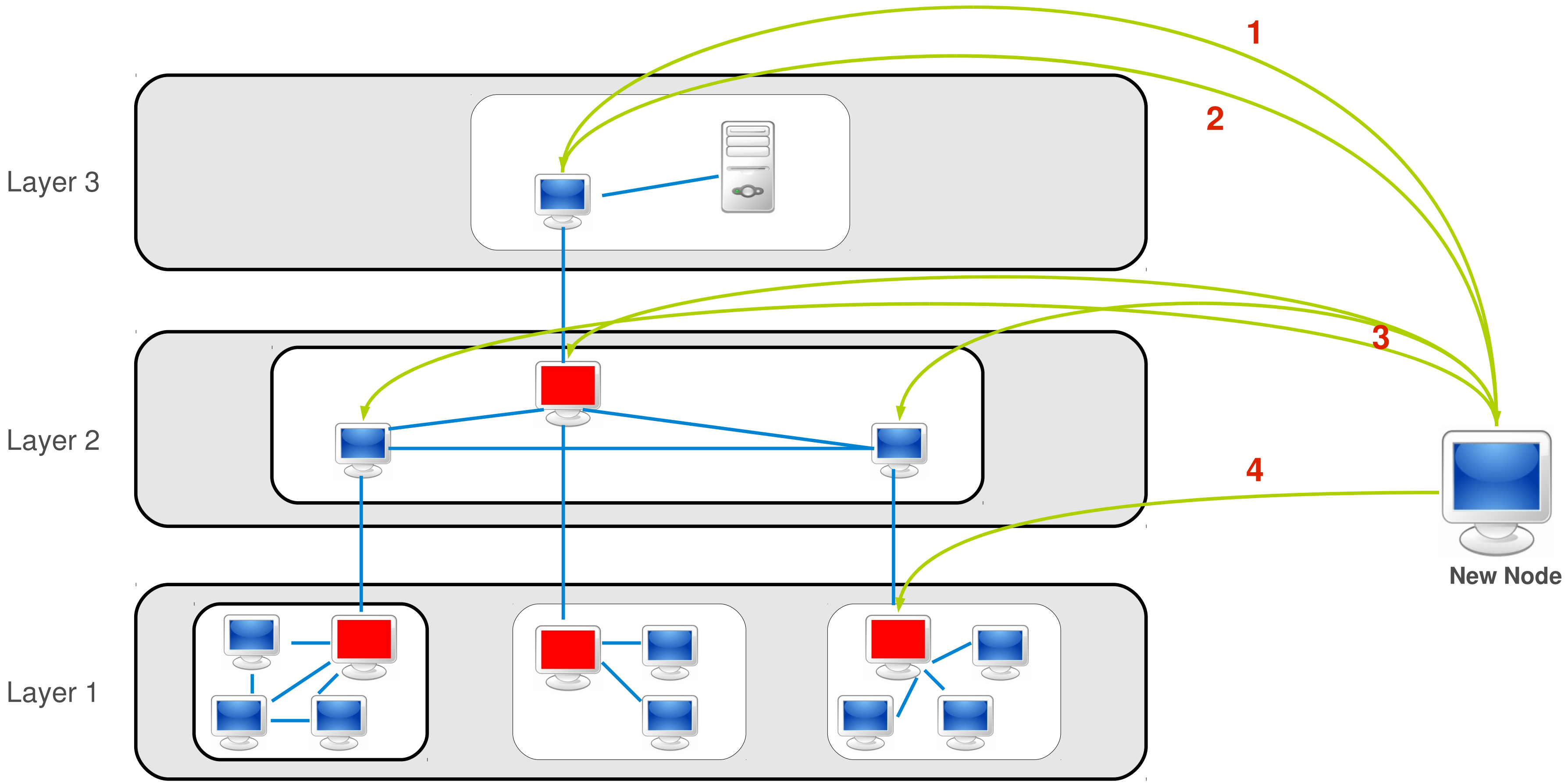


Centralized Advantage/Disadvantage?

- Advantage
  - Fast
  - Easy to apply optimization methods
  - Easy to implement

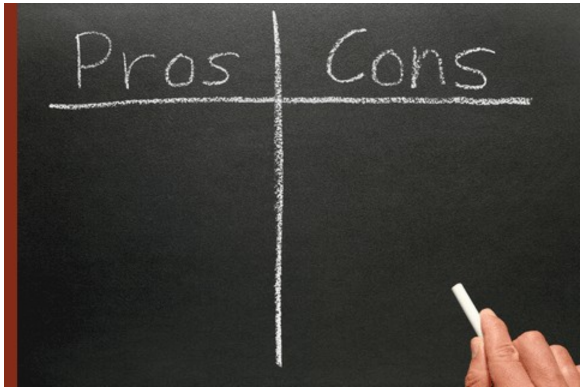
- Disadvantage
  - Not scalable
  - Single point of failure

Hierarchical Method



Hierarchical Advantage/Disadvantage?

- Advantage/Disadvantage [d]

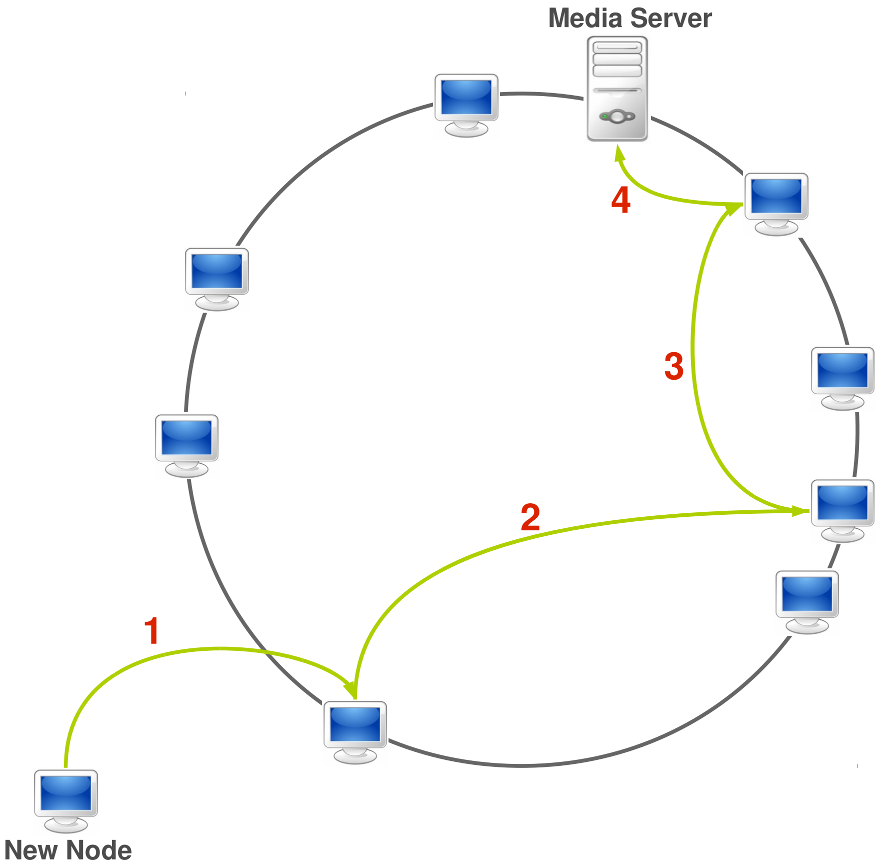


Hierarchical Advantage/Disadvantage?

- Advantage
  - Scalable
  - No single point of failure

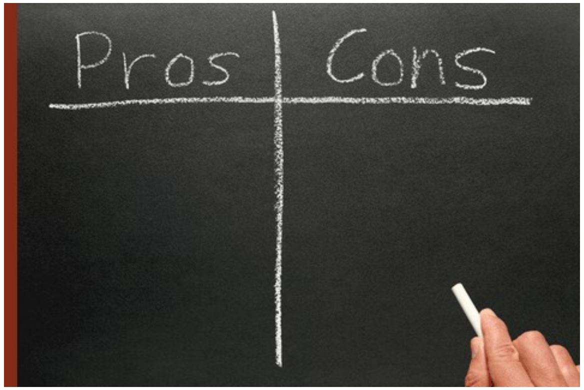
- Disadvantage
  - Slow convergence
  - Difficult to implement

DHT-based Method



DHT-based Advantage/Disadvantage?

- Advantage/Disadvantage [d]

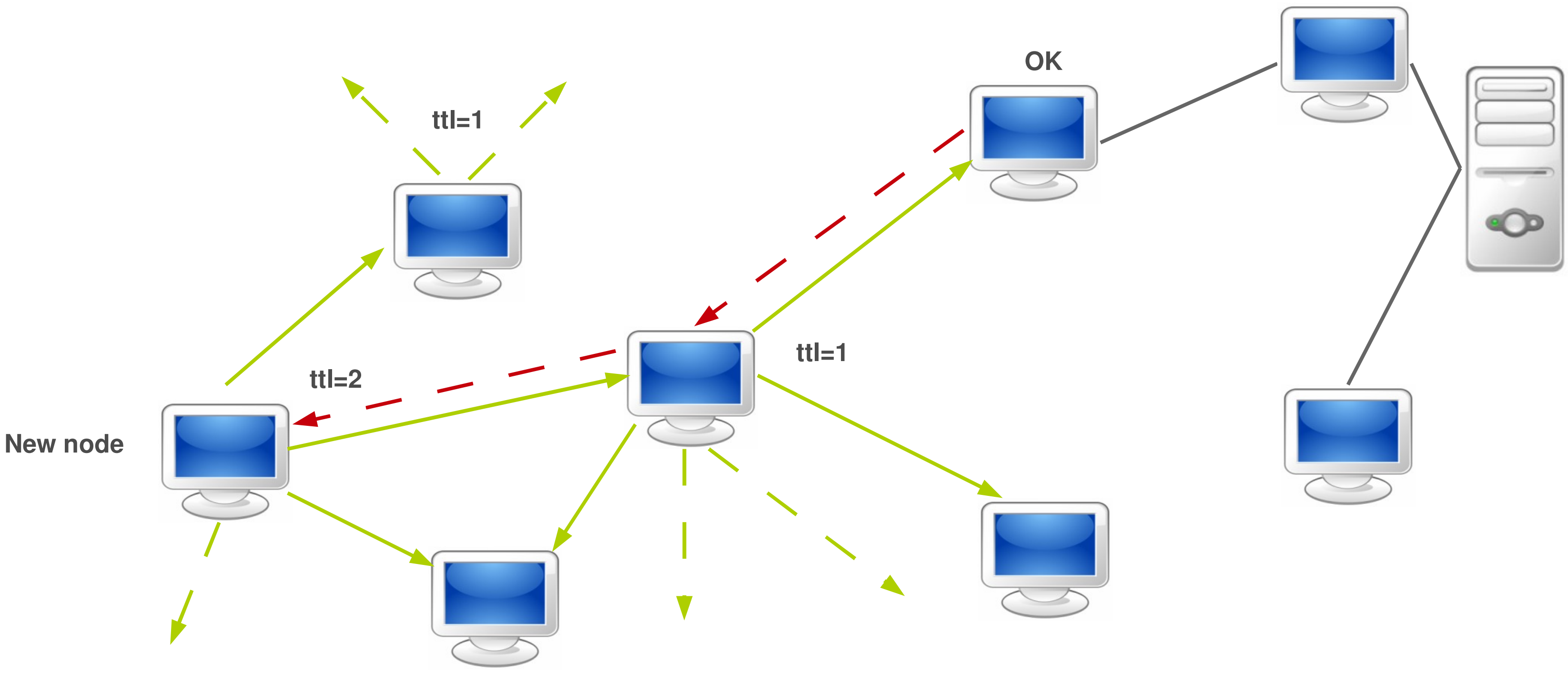


DHT-based Advantage/Disadvantage?

- Advantage
  - Scalable
  - No single point of failure

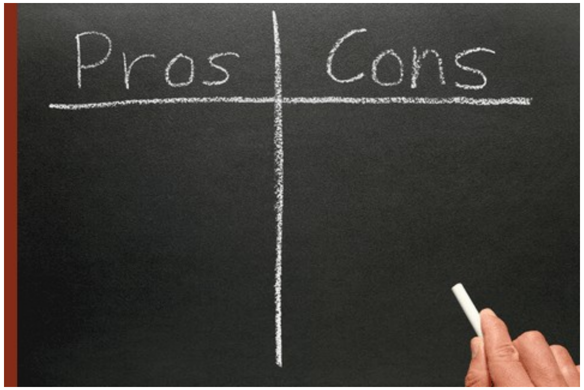
- Disadvantage
  - Difficult to implement

Controlled Flooding Method



Flooding Advantage/Disadvantage?

- Advantage/Disadvantage [d]



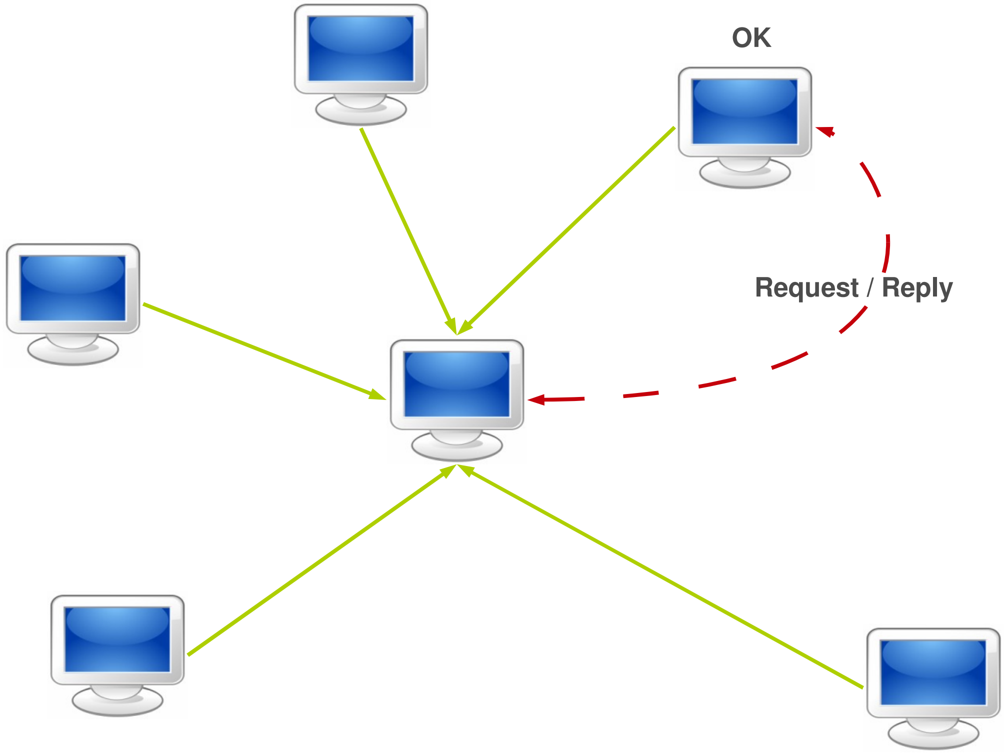
Flooding Advantage/Disadvantage?

- Advantage
  - Scalable
  - No single point of failure

- Disadvantage
  - No guarantee to find supplier node
  - Slow convergence

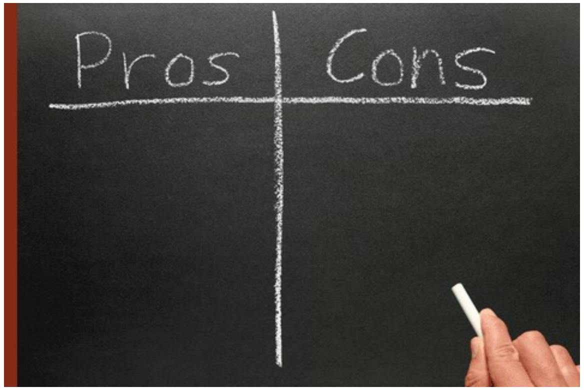
Gossip-based Method

- Peers periodically send their data availability to their neighbours.



Gossip-based Advantage/Disadvantage?

- Advantage/Disadvantage [d]



Gossip-based Advantage/Disadvantage?

- Advantage
  - Scalable
  - No single point of failure
  - Easy to implement

- Disadvantage
  - No guarantee to find supplier node in time

Outline

- Introduction
- P2P media streaming
- Classification of P2P streaming systems
- Security in P2P streaming systems
- Sepidar/GLive – two P2P streaming systems

Classification of P2P Streaming Solutions

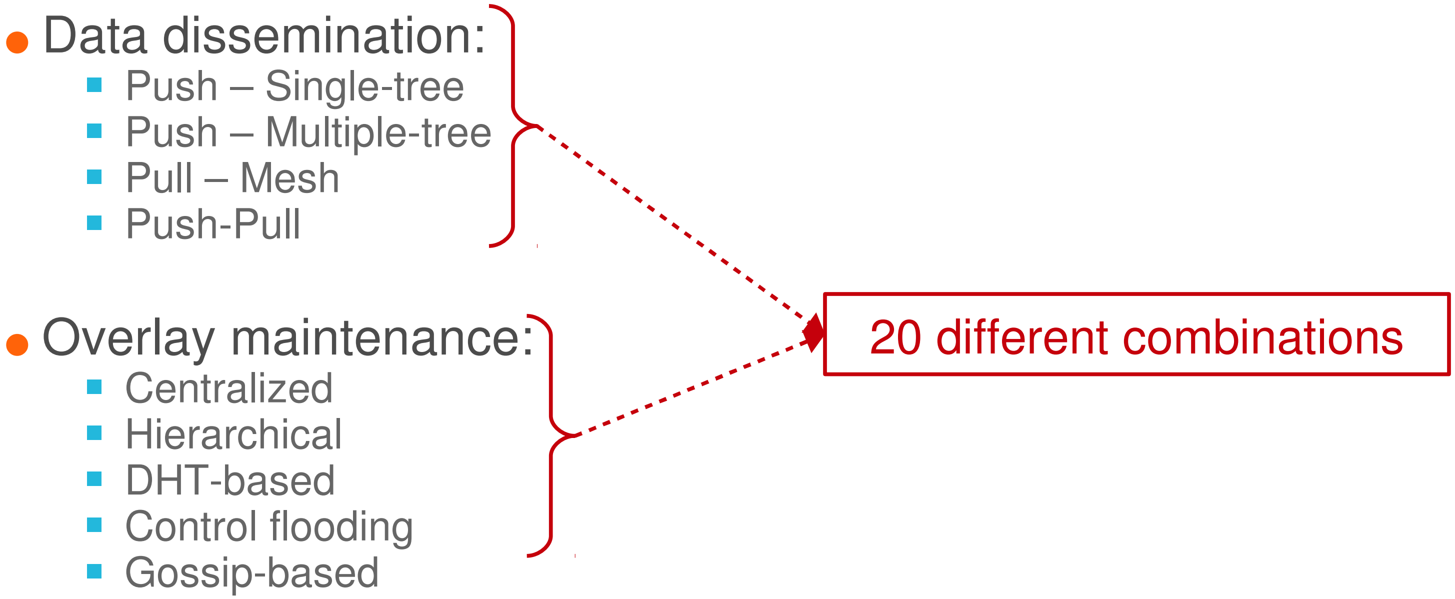
Related Work

- SplitStream
- DONet/Coolsteraming
- CoopNet
- Orchard
- Bullet
- Prime
- Pulsar
- NICE
- Zigzag
- DirectStream
- MeshCast

- mtreeBone
- PULSE
- GnuStream
- SAAR
- ChainSaw
- ChunkySpread
- BulkTree
- ForestCast
- AnySee
- DagStream
- Climber

- CollectCast
- HyMoNet
- GridMedia
- Promise
- Yoid
- Zebra
- Tribler
- CliqueStream
- GradienTv
- Sepidar
- GLive

Data Dissemination Overlay



- Data dissemination:
  - Push – Single-tree
  - Push – Multiple-tree
  - Pull – Mesh
  - Push-Pull
- Overlay maintenance:
  - Centralized
  - Hierarchical
  - DHT-based
  - Control flooding
  - Gossip-based

20 different combinations

Data Dissemination Overlay

Push (Single Tree)	Push (Multiple-tree)	Pull (Mesh)	Push-Pull
DirectedStream HyMoNet Yoid Nice ZigZag Climber SAAR	Coopnet ForestCast Zebra SpliStream SAAR Orchard ChunkySpread	GLive BulkTree CollectCast Promise SAAR GnuStream CoolStreaming Pulse Chainsaw MeshCast Tribler Dagstream	Sepidar GradienTv Prime Pulsar CliqueStream Bullet NewCoolStreaming Mtreebone GridMedia

Overlay Construction and Maintenance Methods

Centralized	DirectedStream, HyMoNet, Yoid, CoopNet ForestCast, Zebra, Prime
Hierarchical	NICE, ZigZag, Climber, BulkTree, Prime
DHT-based	SAAR, SplitStream, CollectCast, Promise, CliqueStream, Pulsar
Flooding	GnuStream
Gossip-based	GLive, Sepidar, GradienTv, Orchard, ChunkySpread, CoolStreaming, Pulse, Chainsaw MeshCast, Tribler, DagStream, Bullet, mTreebone, GridMedia

All Together

	Push (Single tree)	Push (Multiple-tree)	Pull (Mesh)	Push-Pull
Centralized	DirectedStream HyMoNet Yoid	Coopnet ForestCast Zebra		Prime
Hierarchical	NICE ZigZag Climber		BulkTree	Prime
DHT-based	SAAR	SAAR SplitStream	SAAR CollectCast Promise	Pulsar CliqueStream
Flooding			GnuStream	
Gossip-based		Orchard ChunkySpread	Glive - CoolStreaming – Pulse - Chainsaw – MeshCast - Tribler - DagStream	Sepidar - GradienTv Bullet - mTreebone GridMedia

Outline

- Introduction
- P2P media streaming
- Classification of P2P streaming systems
- Security in P2P streaming systems
- Sepidar/GLive – two P2P streaming systems

Security in P2P Streaming Systems

Free-riding Problem

- Free-riders are the nodes that uses the resources in the system, without contributing in data distribution.

- Incentivzing mechanism
  - Tit-for-tat
  - Transitive auditing

Collusion

- The attacks can exacerbate by collusion.
- A collection of nodes conduct correlated attack.

Common Attacks in P2P Streaming Systems

- Forgery and Repudiation attacks
  - Forgery: fabricating or tampering data stream.
  - Repudiation: denying the received data stream or to acknowledge with false information.

Common Attacks in P2P Streaming Systems

- Forgery and Repudiation attacks
  - Forgery: fabricating or tampering data stream.
  - Repudiation: denying the received data stream or to acknowledge with false information.

- Pollution attacks
  - Mixing or substituting junk data into the stream.

Common Attacks in P2P Streaming Systems

- Forgery and Repudiation attacks
  - Forgery: fabricating or tampering data stream.
  - Repudiation: denying the received data stream or to acknowledge with false information.

- Pollution attacks
  - Mixing or substituting junk data into the stream.

- Membership and Eclipse attacks
  - Compromising the underlying overlay or membership protocol, e.g., the routing mechanism.

Common Attacks in P2P Streaming Systems

- Neighbour selection attacks
  - Controlling the neighbour selection mechanism of some nodes.

Common Attacks in P2P Streaming Systems

- Neighbour selection attacks
  - Controlling the neighbour selection mechanism of some nodes.
- Sybil attacks
  - Used when the reputation mechanism established in a P2P system.
  - Creating a large number of entities, which bear the same disguised identifier.

Common Attacks in P2P Streaming Systems

- Neighbour selection attacks
  - Controlling the neighbour selection mechanism of some nodes.

- Sybil attacks
  - Used when the reputation mechanism established in a P2P system.
  - Creating a large number of entities, which bear the same disguised identifier

- DoS attacks
  - Sending excessive amount of requests and ...

Common Attacks in P2P Streaming Systems

- Neighbour selection attacks
  - Controlling the neighbour selection mechanism of some nodes.
- Sybil attacks
  - Used when the reputation mechanism established in a P2P system.
  - Creating a large number of entities, which bear the same disguised identifier
- DoS attacks
  - Sending excessive amount of requests and ...
- Omission attacks
  - Not sending the data according to the protocol.
  - Other extreme than DoS attack.

Common Attacks in P2P Streaming Systems

Attack	Target
Forgery	Data
Pollution	Data
Eclipse	Overlay, Protocol
Neighbor	Protocol
Sybil	Protocol
DoS	Peers
Omission	Peers, Data

Outline

- Introduction
- P2P media streaming
- Classification of P2P streaming systems
- Security in P2P streaming systems
- Sepidar/GLive – two P2P streaming systems

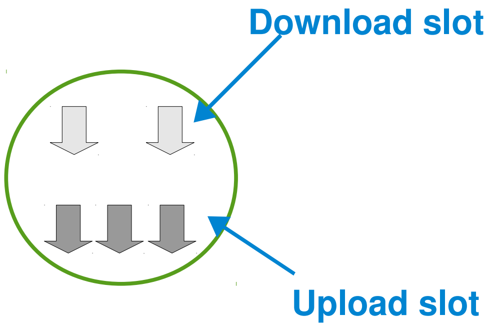
Sepidar

Problem Description (1/5)

- Building and optimizing a P2P overlay for live media streaming
  - minimize playback latency
  - improve timely delivery of the stream

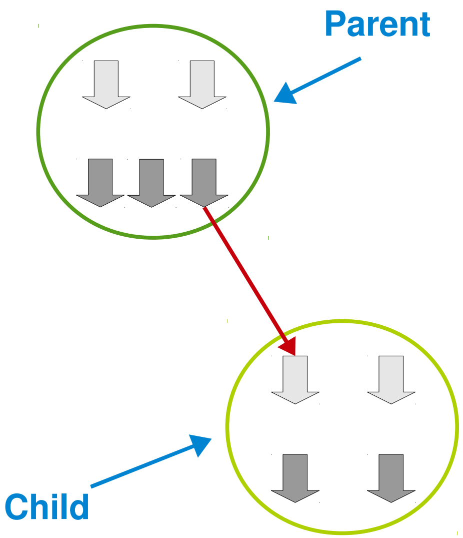
- The media stream is split into a number of sub-streams or stripes.

- A node can create a bounded number of download connections, and accept a bounded number of upload connections.



Problem Description (2/5)

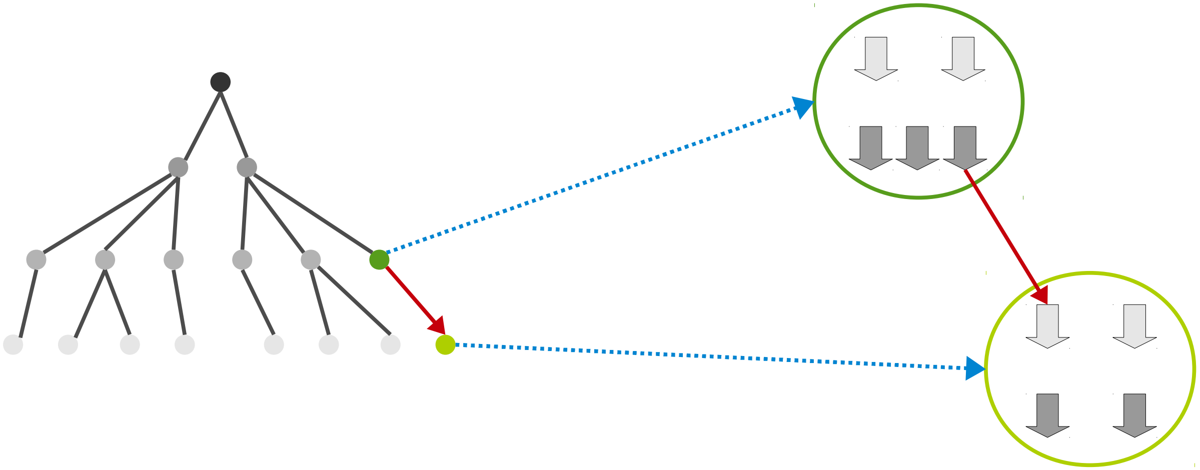
- In order to provide the full media to all the nodes
  - every download-slot needs to be assigned to an upload-slot.
  - download-slots at a node must download different stripes.
- This problem can be defined as an assignment problem.



Problem Description (3/5)

- A connection between a download-slot i and an upload-slot j for a stripe k is associated with a cost cijk, which is the number of hops from the owner of the upload-slots j, to the media source for the stripe k.

- A complete assignment, A, is an assignment that each download-slot is assigned to an upload-slot.



Problem Description (4/5)

- Formulating as an optimization problem:

- Objective function
  - We want to find a complete assignment over all the complete assignments that minimizes the total cost:

$$\sum_{(i,j,k)\in A} c_{ijk}$$

- Subject to
  - Every download-slot is assigned to exactly one upload-slot.
  - Each upload-slot is assigned to at most one download-slot.
  - The download-slots owned by the same node download distinct stripes.

Problem Description (5/5)

- Centralized solution:
  - Needs global knowledge.
  - Possible for small system sizes.
- Our distributed market-based approach:
  - Inspired by auction algorithms.
  - Each node knows only a small number of nodes in the system (partial view).

Designing a P2P Media Streaming System

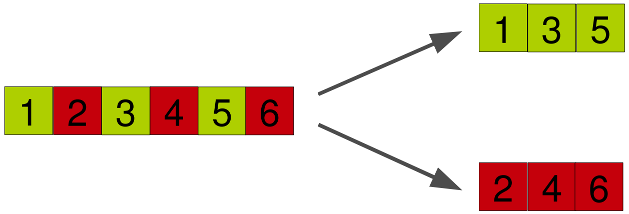
- What overlay topology is built for data dissemination?
  - Tree
  - Multiple-tree
  - Mesh

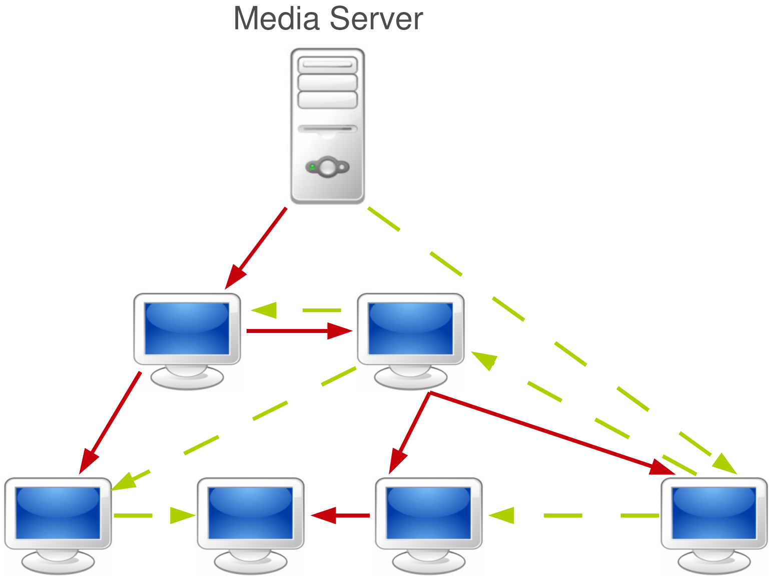
- What algorithm is used for data dissemination?
  - Push
  - Pull
  - Push-Pull

- How to construct and maintain this overlay?
  - Centralized
  - DHT
  - Gossip-based
  - ...

Multiple-Tree Structure

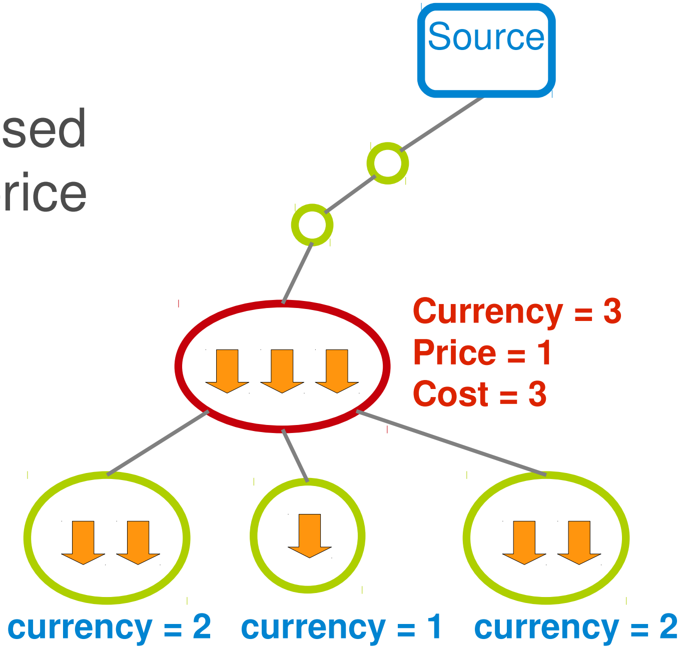
- The media source splits the stream into a set of sub-streams.
- A single tree is created for each sub-stream.
- A peer to receive the whole media should join all trees.





The Market Model - Node Properties

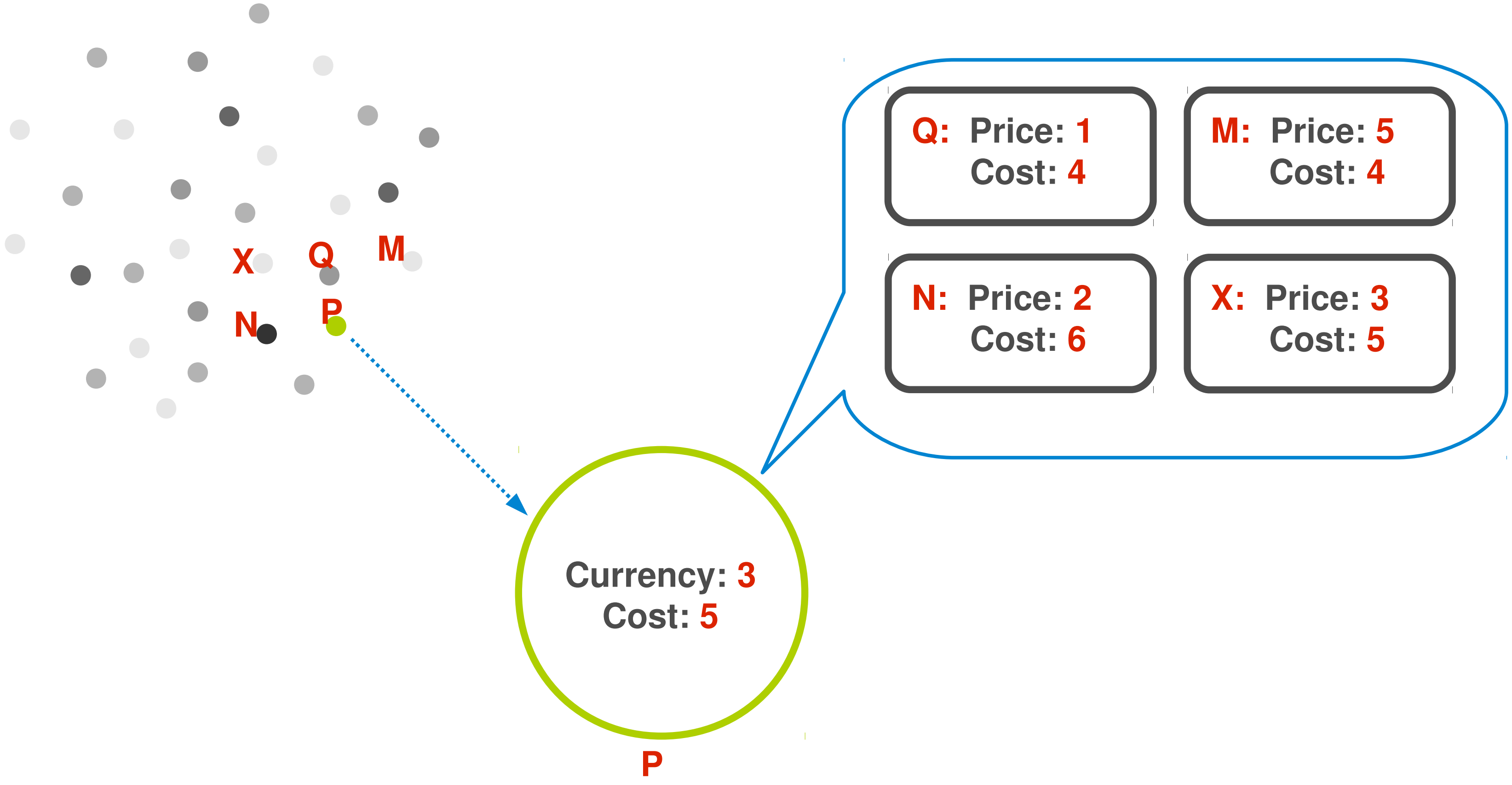
- Currency: The the number of upload slots at a node.
- Price: The price of a node that has an unused upload slot is zero, otherwise the node's price equals the lowest currency of its already connected children.
- Cost: The length of its path to the root.



The Market Model - 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.

The Market Model – Child Side



Q: Price: 1
Cost: 4

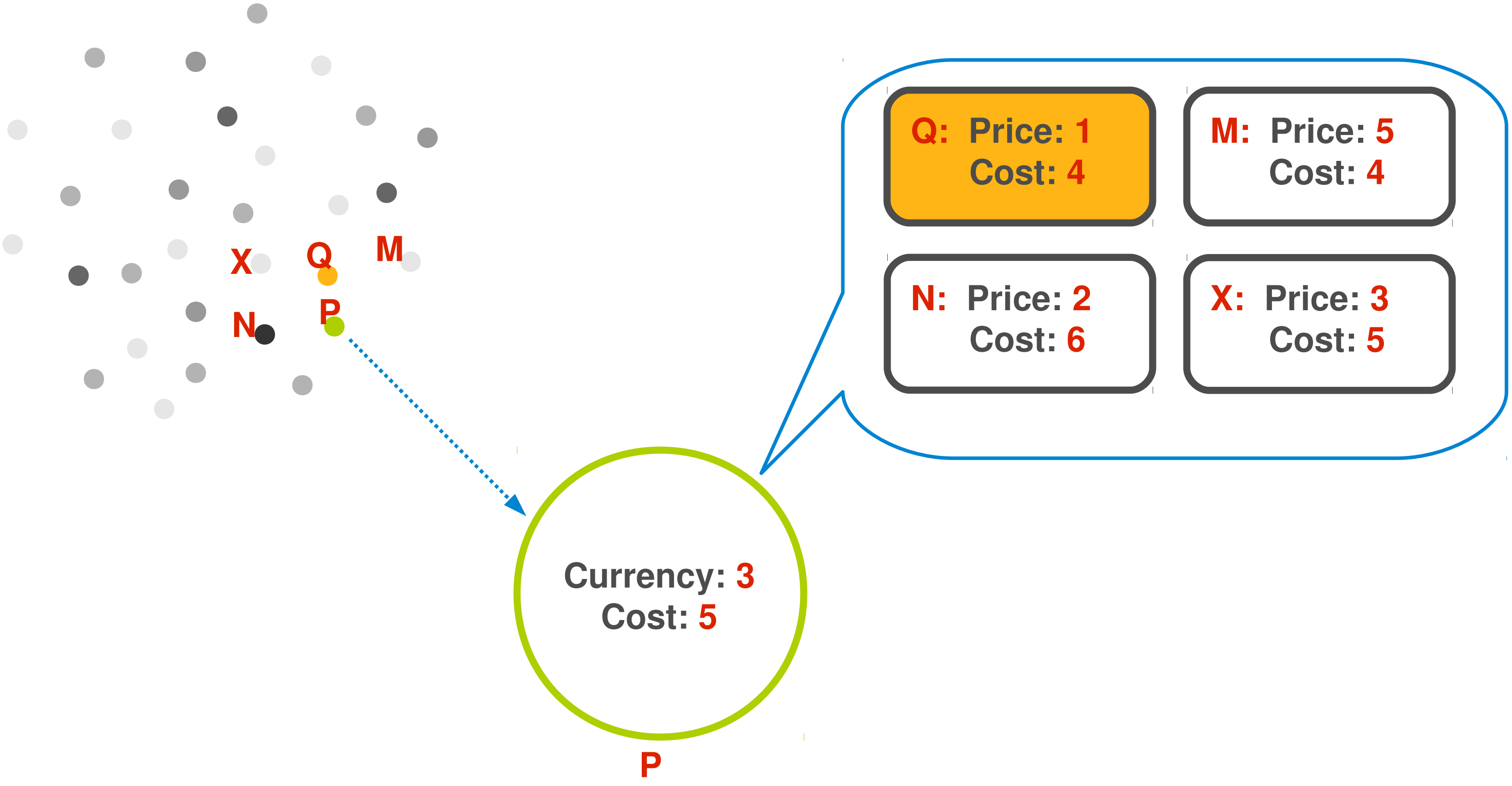
M: Price: 5
Cost: 4

N: Price: 2
Cost: 6

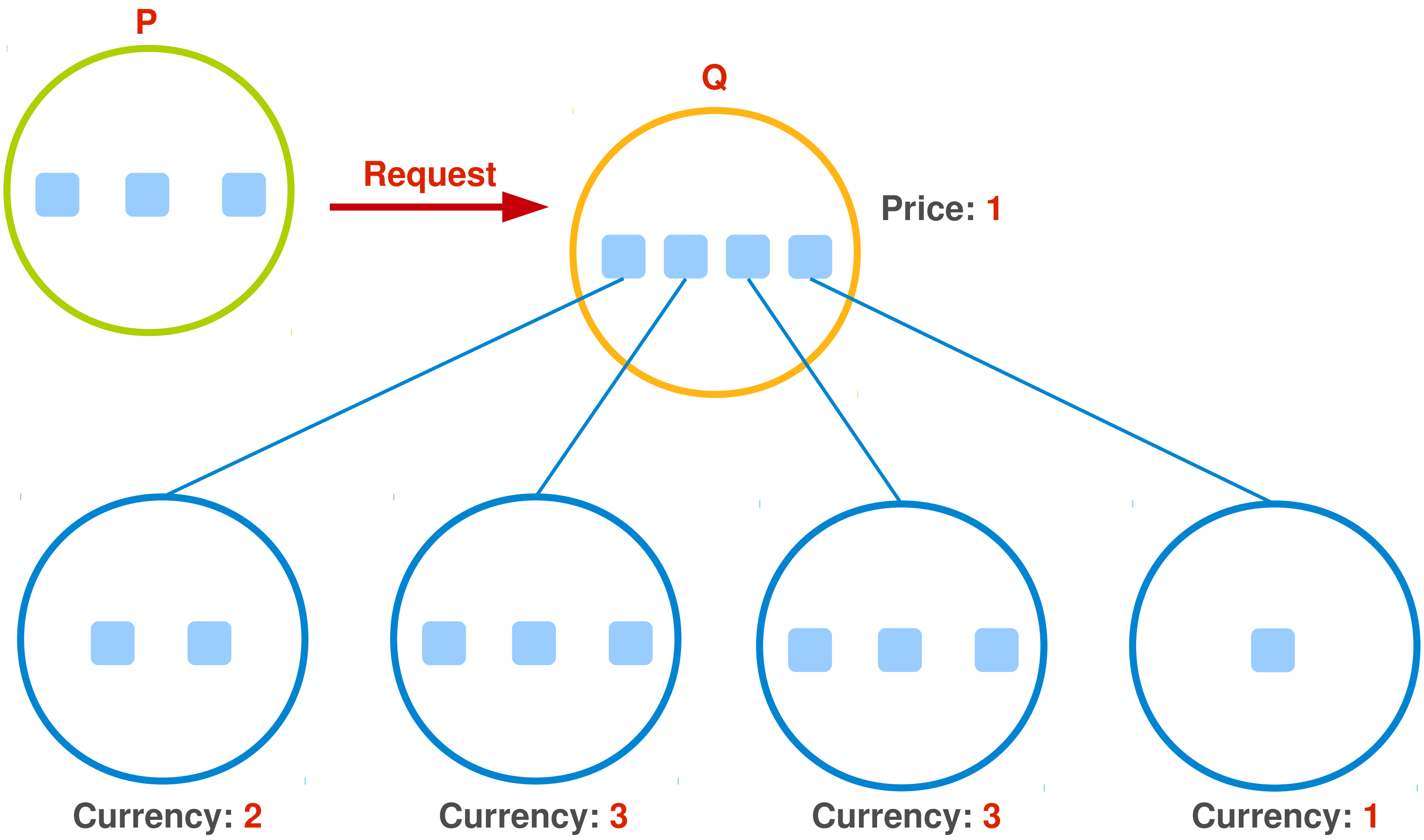
X: Price: 3
Cost: 5

Currency: 3
Cost: 5

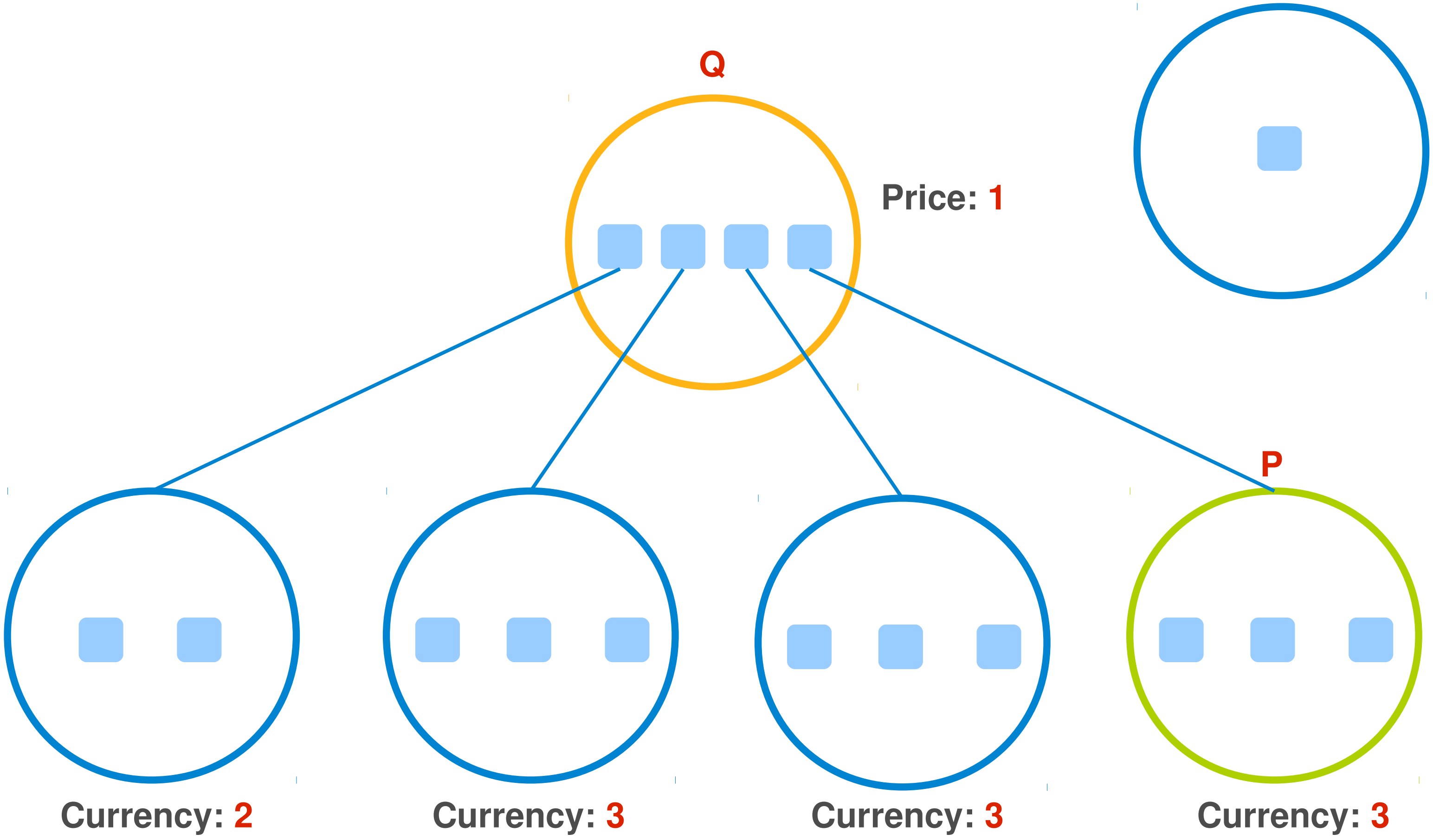
The Market Model – Child Side



The Market Model – Parent Side

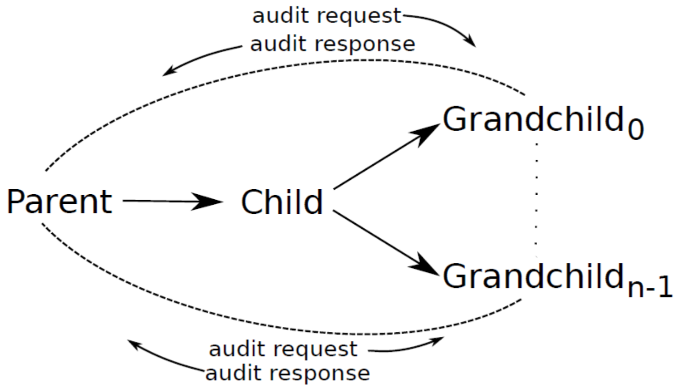


The Market Model – Parent Side



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

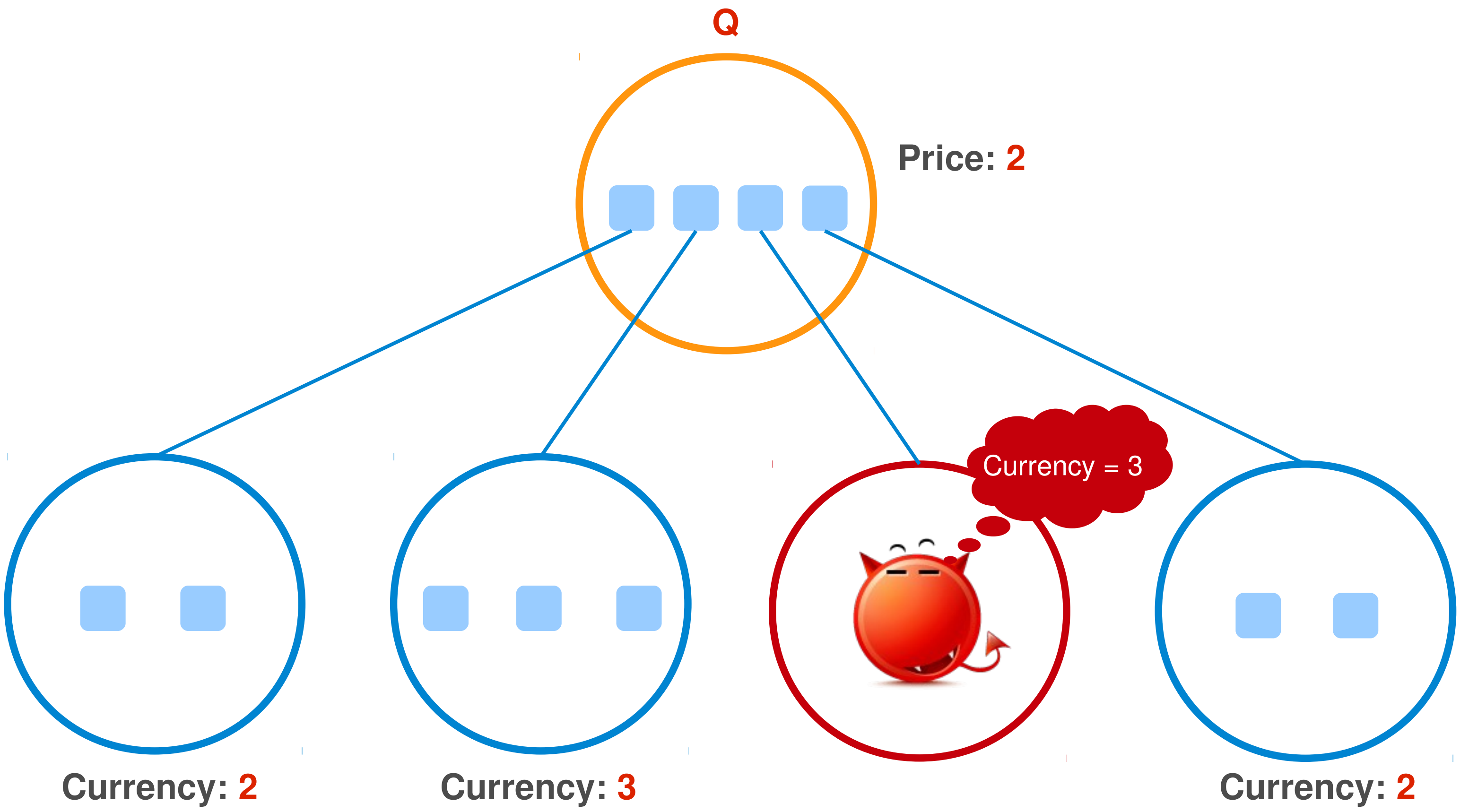
- F is the sum of
  - the number of audit responses not received before a timeout.
  - the number of negative audit responses.
  - the free upload slots.

- If F is more than M% of claimed upload slots, Q is suspected as a freerider.

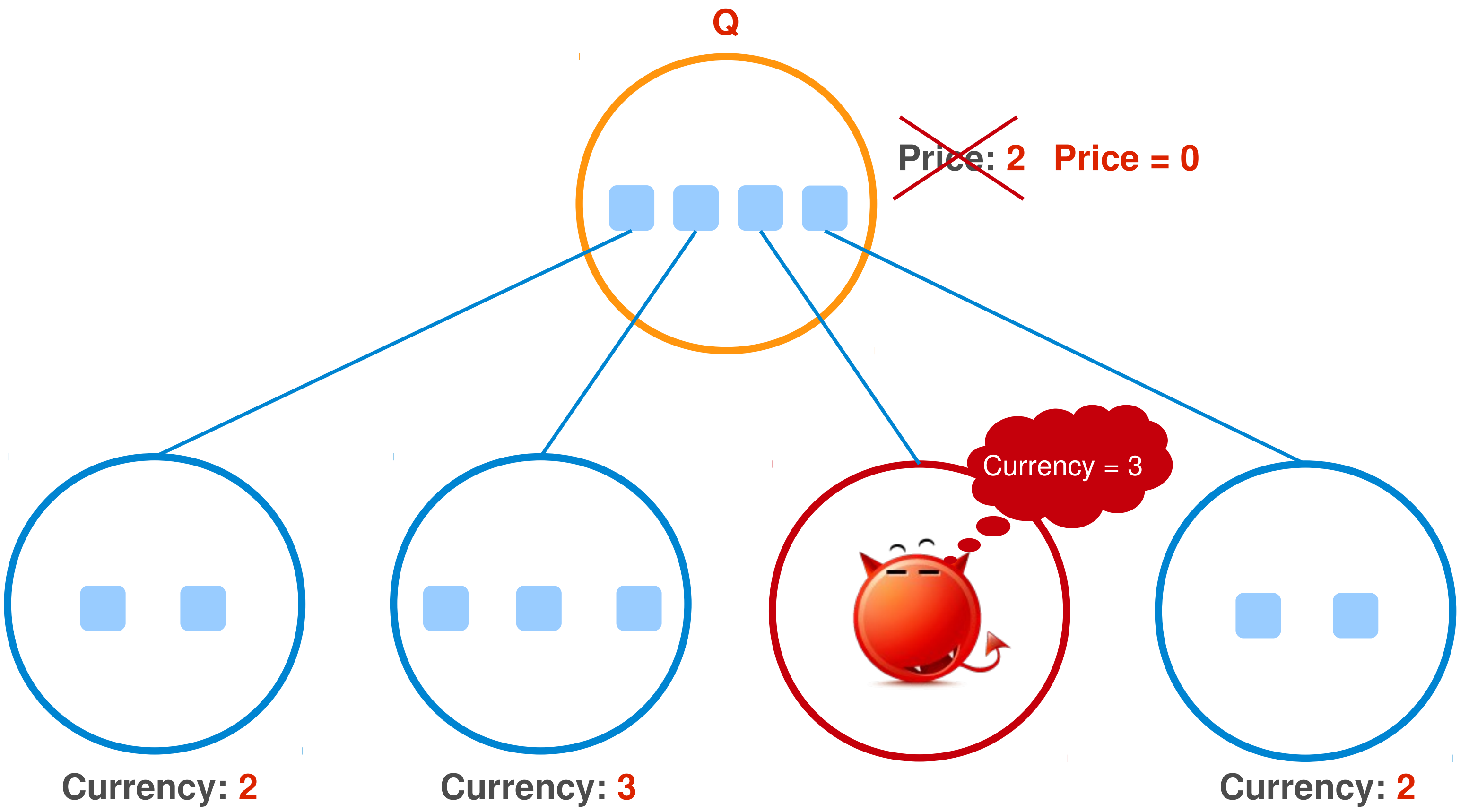
- If Q becomes suspected in N consecutive iterations, it is detected as a freerider.

- The higher the value of N, the more accurate but slower the detection is.

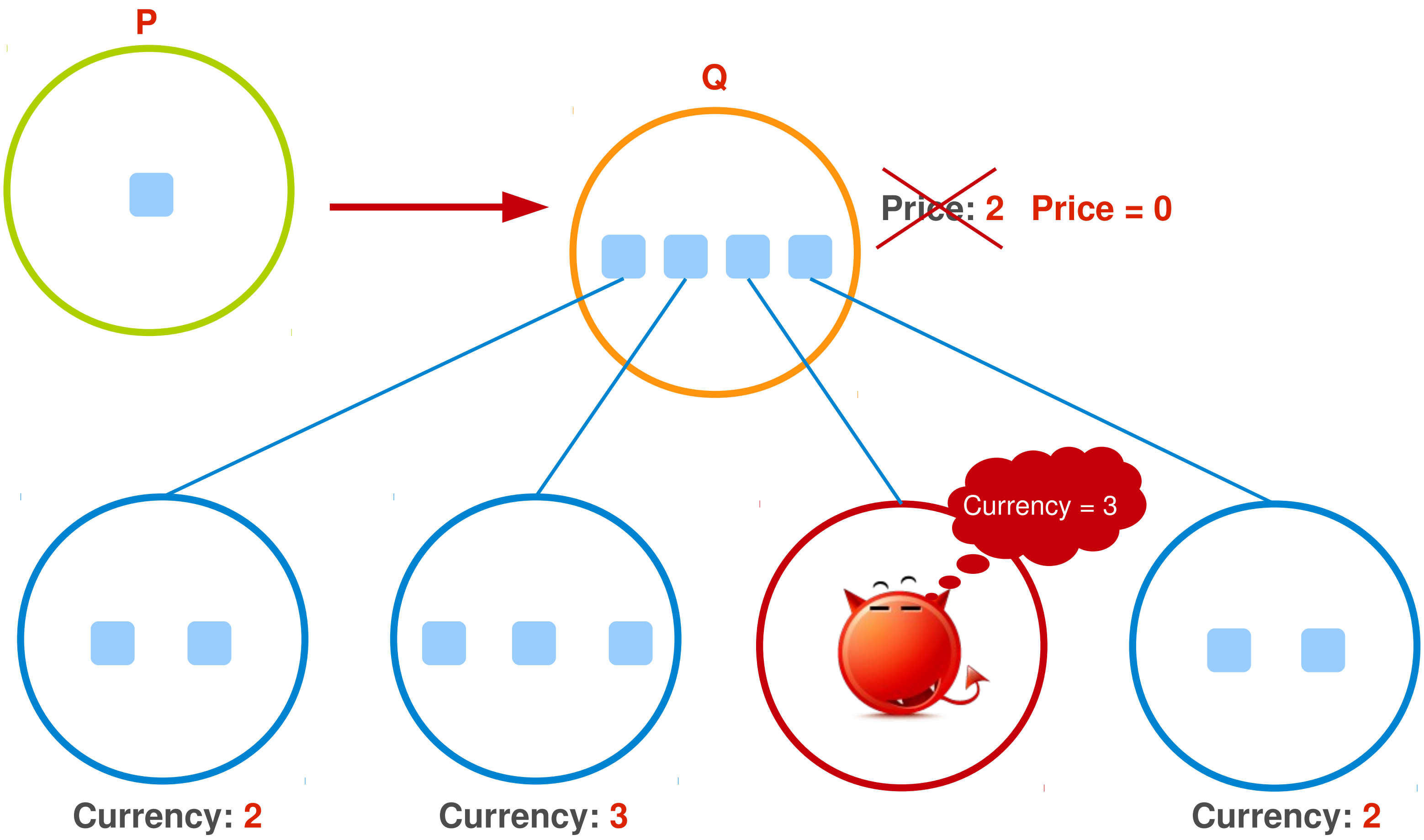
Freerider – Punishment (1/4)



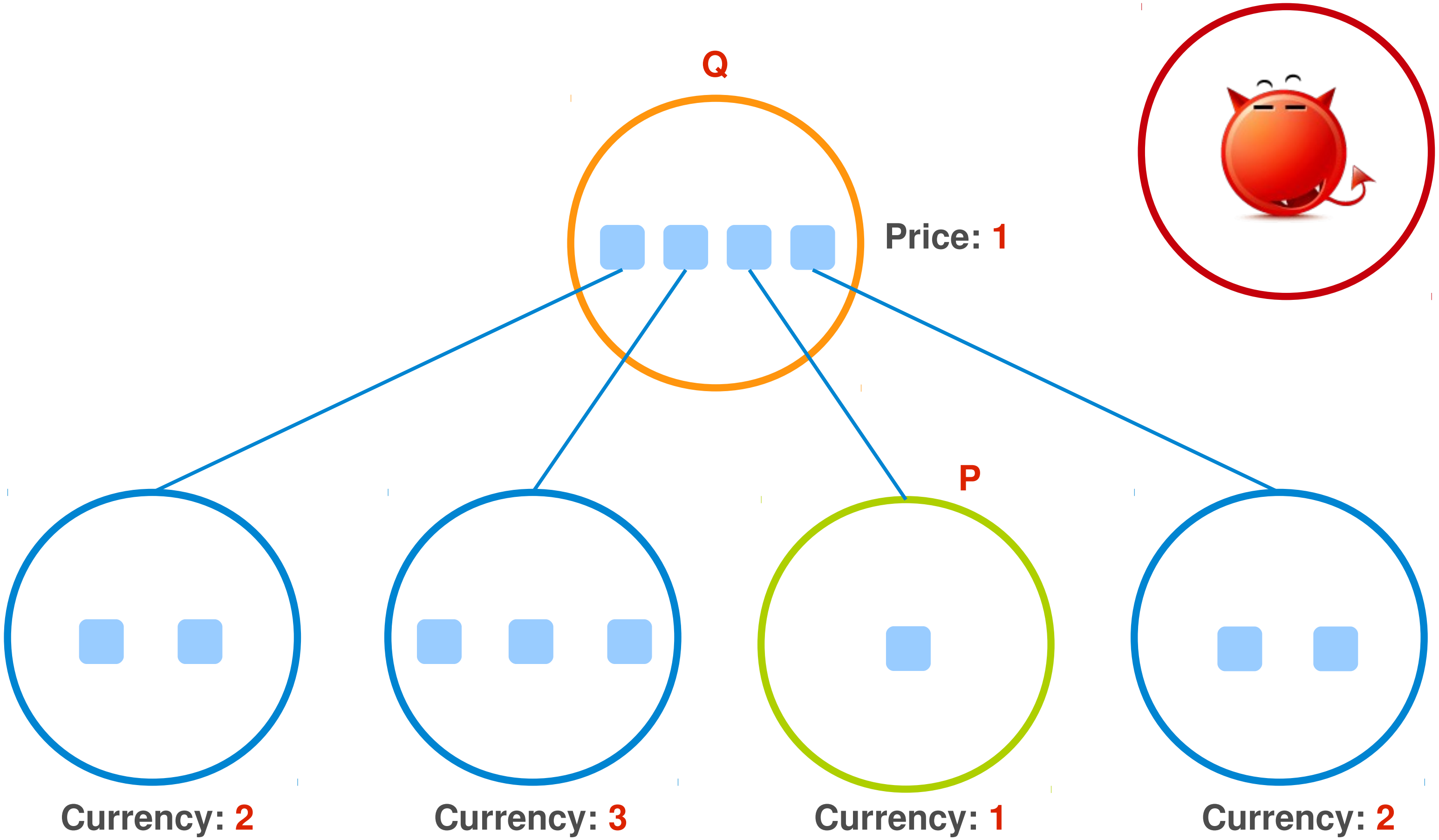
Freerider – Punishment (2/4)



Freerider – Punishment (3/4)



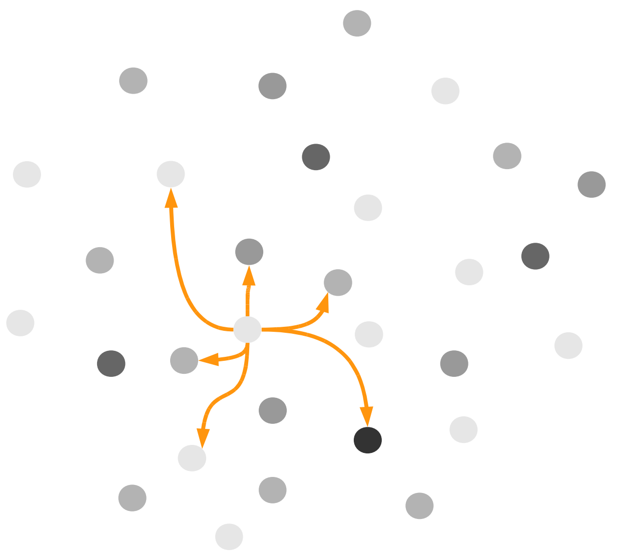
Freerider – Punishment (4/4)



Optimization

Node Discovery

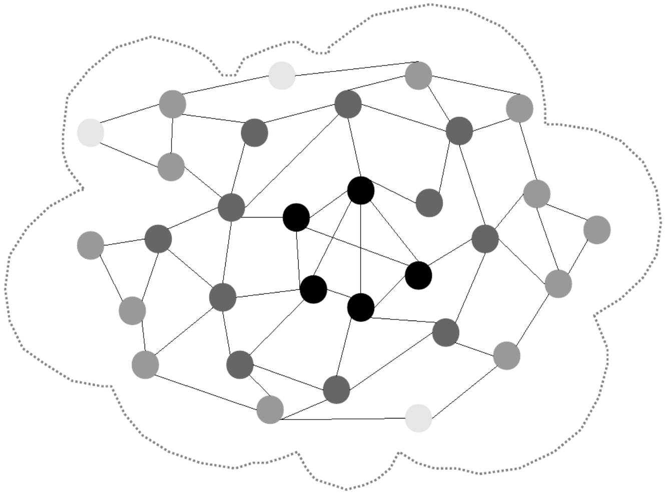
- Naïve solution: nodes in partial views are selected randomly from all the nodes.
- Optimization: nodes use the Gradient overlay to construct and maintain their partial view of the system.



Random view →

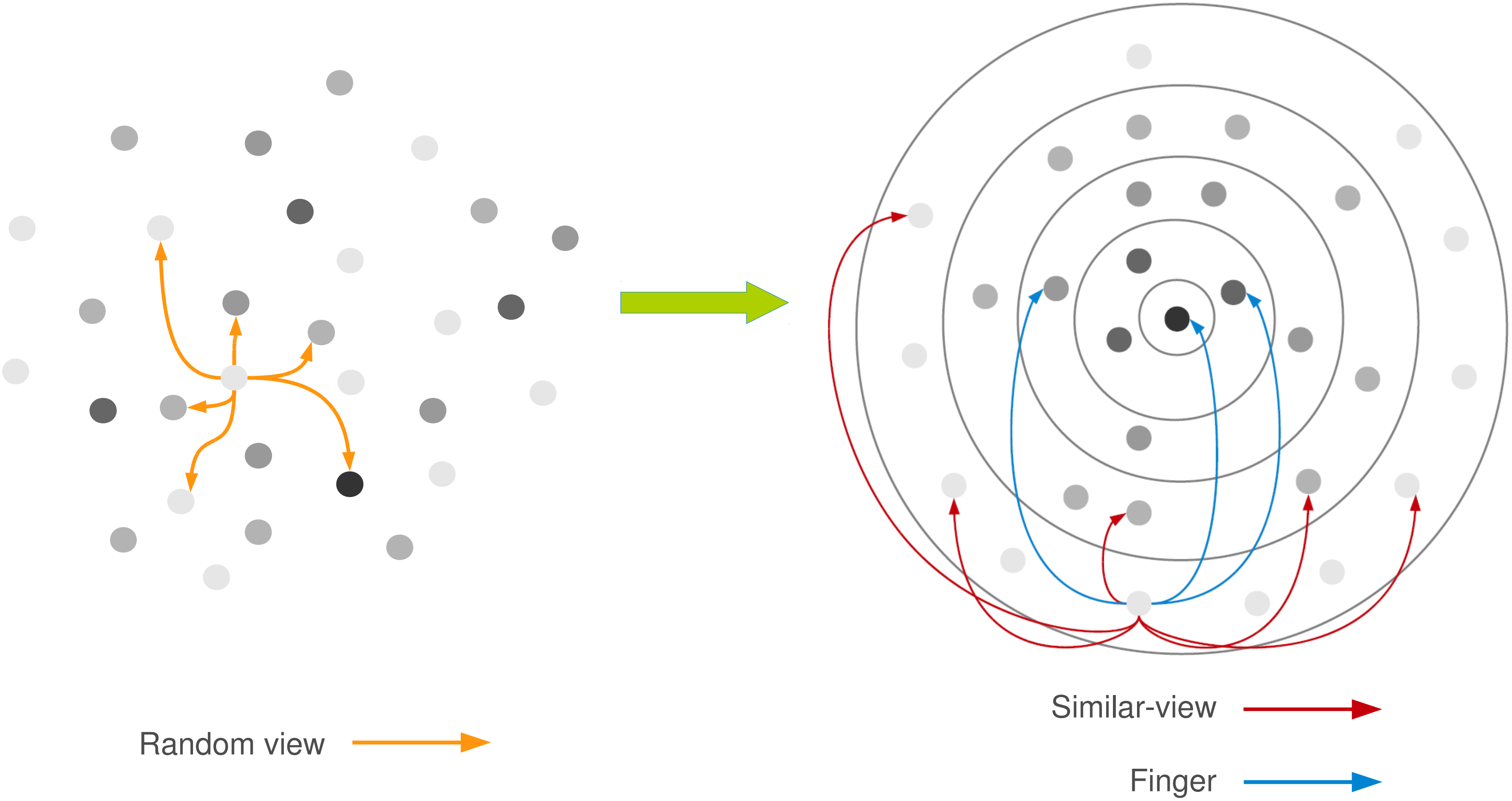
The Gradient Overlay

- 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.



A Peer Partners

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



GLive

Shortcoming of Sepidar

- Tree structure
- Fragile in massive failures

Designing a P2P Media Streaming System

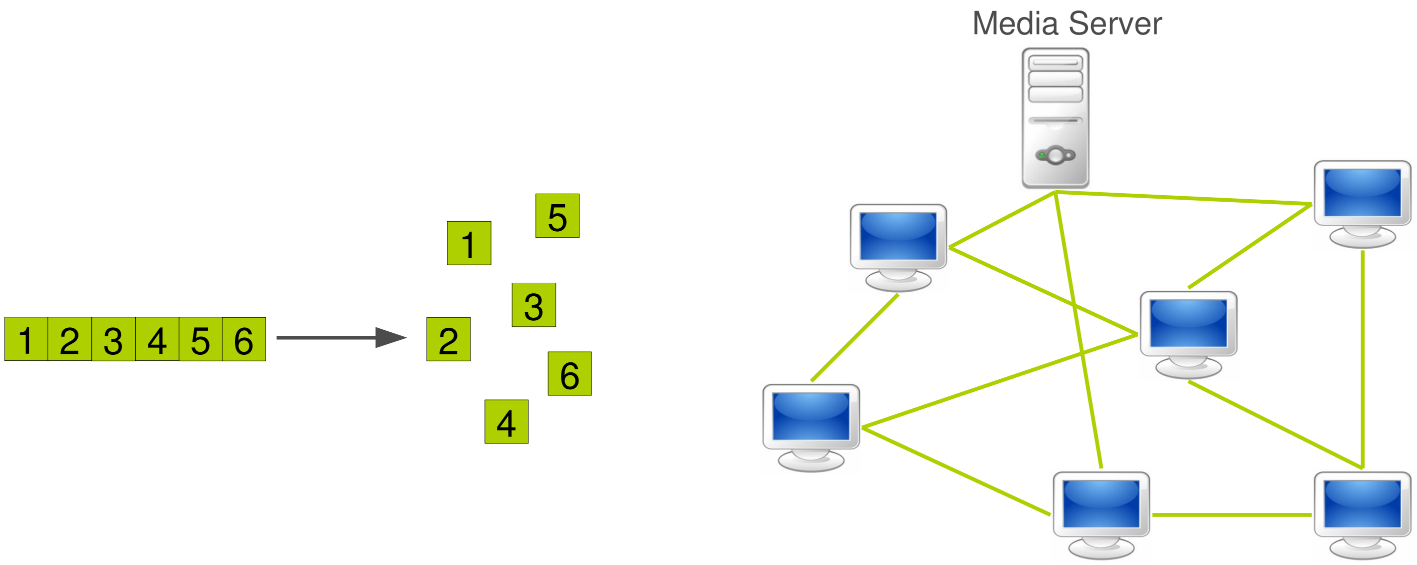
- What overlay topology is built for data dissemination?
  - Tree
  - Multiple-tree
  - Mesh

- What algorithm is used for data dissemination?
  - Push
  - Pull
  - Push-Pull

- How to construct and maintain this overlay?
  - Centralized
  - DHT
  - Gossip-based
  - ...

Mesh Overlay

- Divide he main stream into a 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 – Parent/Child Selection

- The same as Sepidar.

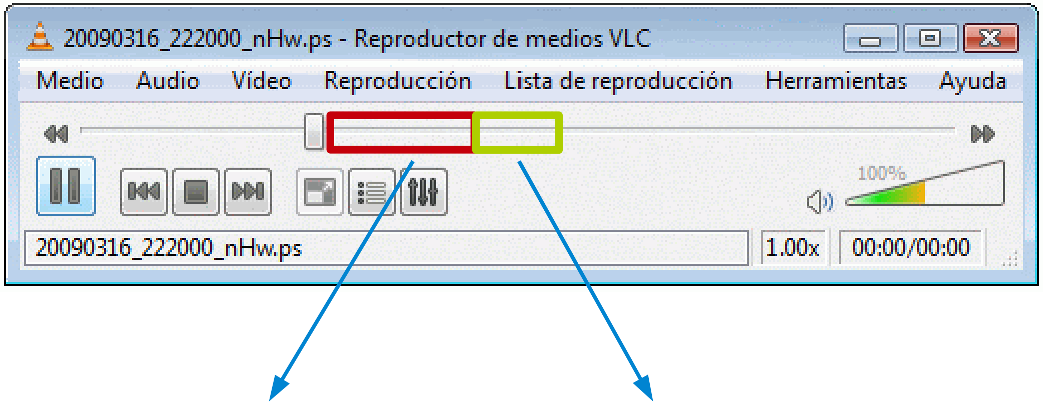
Data Dissemination (1/2)

- 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.

In order
Rarest first
Hybrid
I'm back

Data Dissemination (2/2)

- Sliding window



In-order set

Rare set

Freerider Detection (1/2)

- Each child assigns a score to each of its parents, for a time window covering the last 10 seconds.

- When a child requests and receives a non-duplicate block from a parent within the last 10 seconds, it increments the score of that parent.

- A node periodically sends a score request to its grandchildren.

Freerider Detection (2/2)

- Threshold s to detect freeriders.

- When a node with no free upload connection receives a connection request, it sorts its children based on their latest scores.
  - If there exist children with score less than s, the lowest score child is abandoned.
  - Otherwise, accepts if the new node offers more money than the lowest money of its existing children.

DONE!

Summary

- Media Streaming
  - Live
  - VoD

- Client-Server model
  - Expensive
- P2P model
  - The peers can help each other and the capacity increases with the numbe of peers.

- Challenges
  - Time constraint
  - Churn
  - Connectivity
  - Security
- Main questions
  - What overlay topology?
  - What algorithm for data dissemination?
  - How to construct the topology

Any Questions?