# P2P Media Streaming

Amir H. Payberah (amir@sics.se)

# Outline

- Introduction
- P2P media streaming
- Classification of P2P streaming systems
- Connectivity problem
- Security in P2P streaming systems
- Sepidar – a P2P streaming system

# Introduction

# Media Streaming

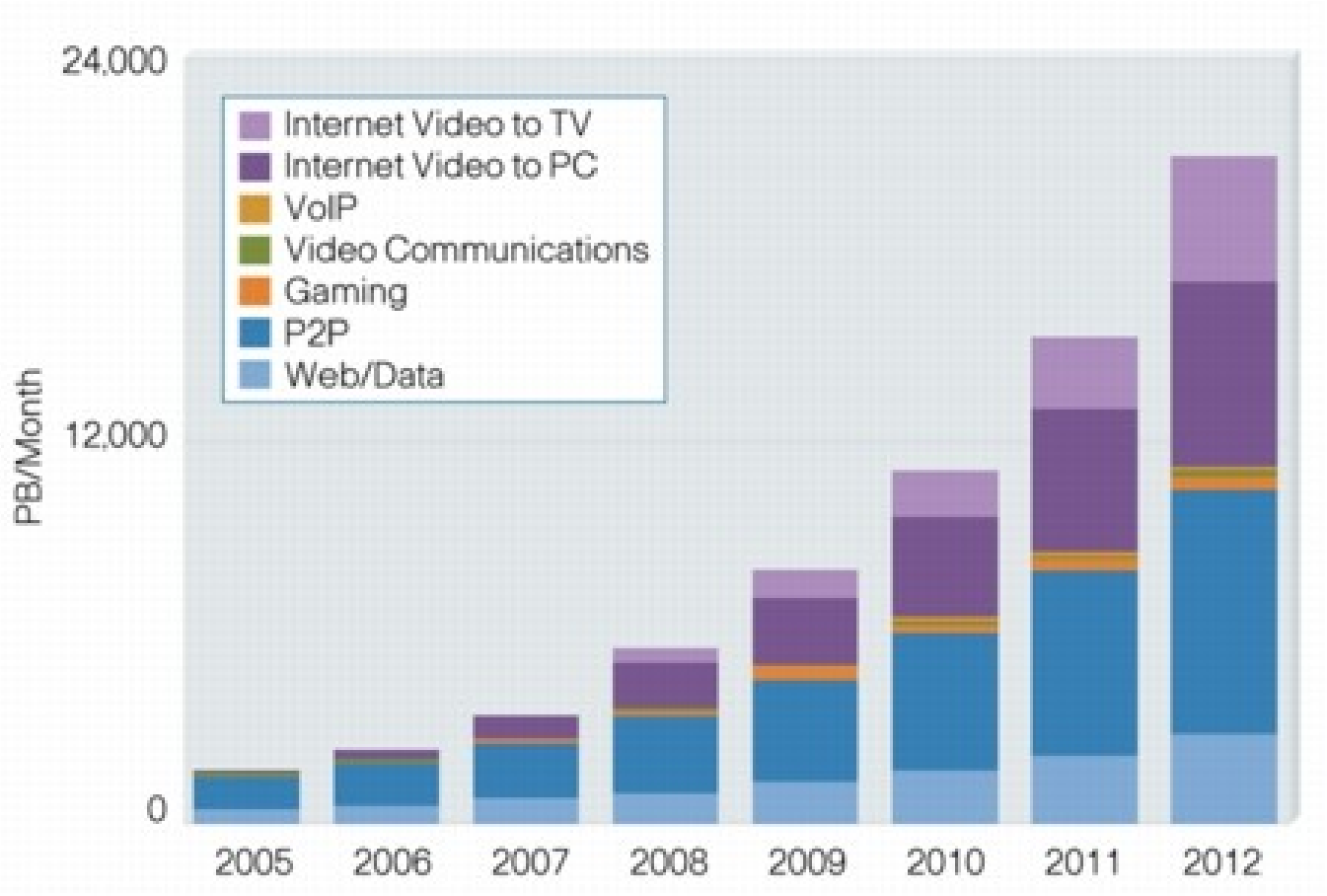
- Media streaming is a multimedia that is sent over a network and played as it is being received by end users.
- Users do not need to wait to download all the media.
- They can play it while the media is delivered by the provider.

# Media Streaming

- Live Media Streaming
  - The streams are only available at one particular 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

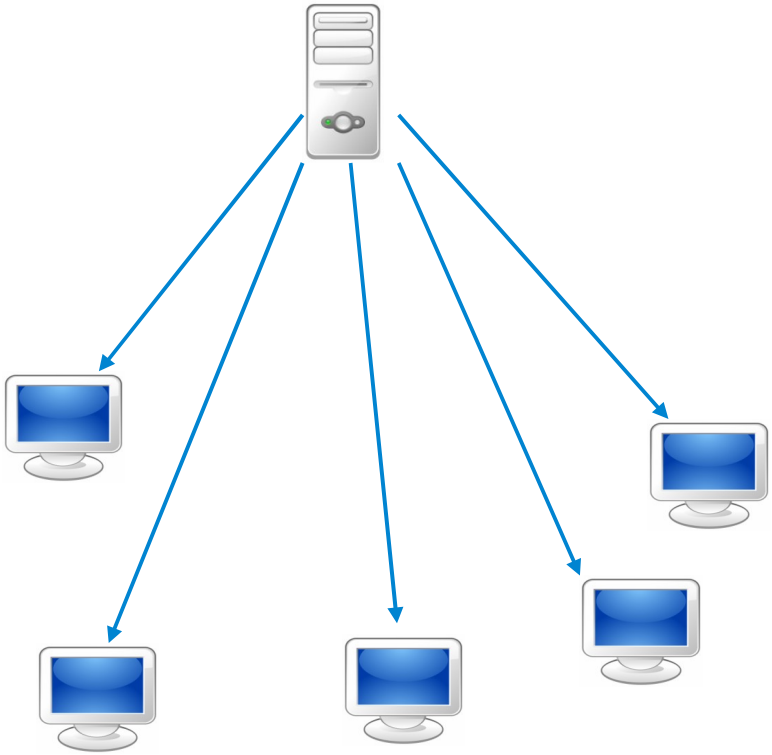



Cisco's global consumer Internet traffic forecast

# Solutions for Media Streaming

- Client-Server solution

# Client – Server



# Client – Server

- What is the problem of Client-Server model? [d]

# Client – Server

- What is the problem of Client-Server model?

- Scalability
- Single point of failure

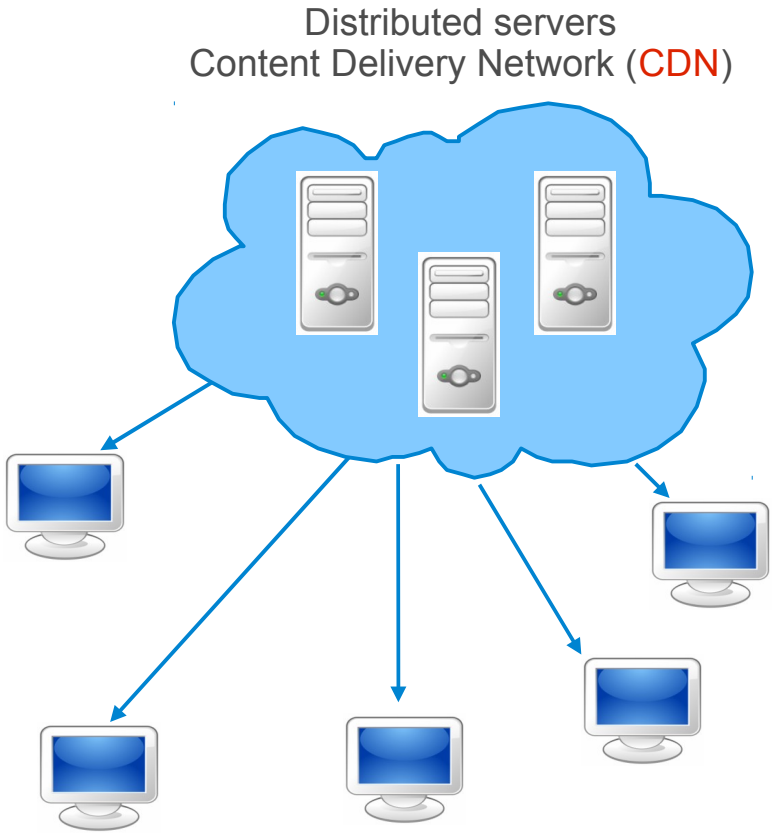
# Client – Server

- What is the problem of Client-Server model?

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

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

# Client – Server

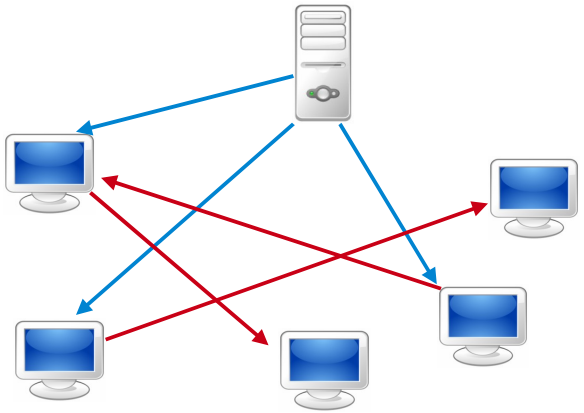



# Solutions for Media Streaming

- 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
- Connectivity problem
- Security in P2P streaming systems
- Sepidar – a P2P streaming system

# 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 overlay topology is built for data dissemination?
- What algorithm is used for data dissemination?
- How to construct and maintain this overlay?

# Main Questions

- What overlay topology is built for data dissemination?
- What algorithm is used for data dissemination?
- How to construct and maintain this overlay?

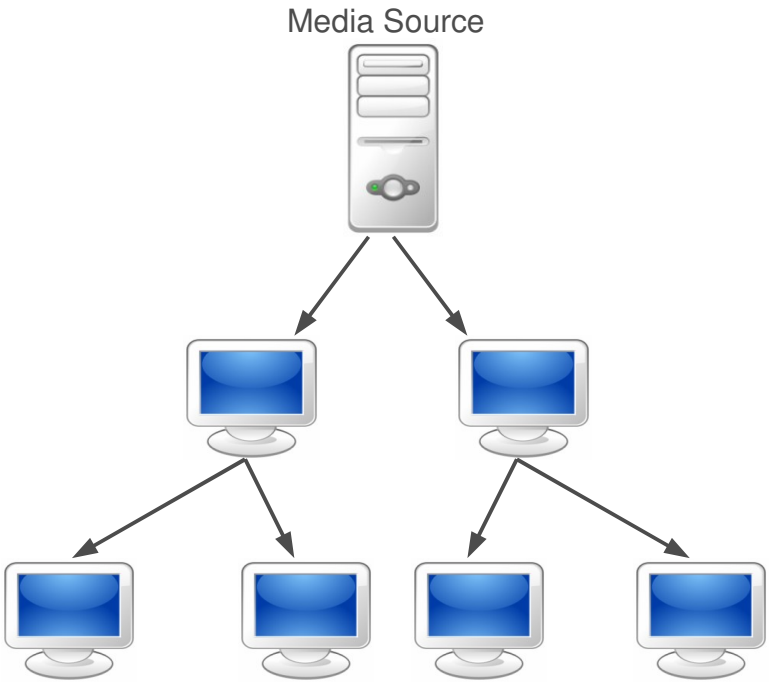
# Data Dissemination Overlay

- What overlay topology is built to distribute data messages.

- It could be:
  - 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?

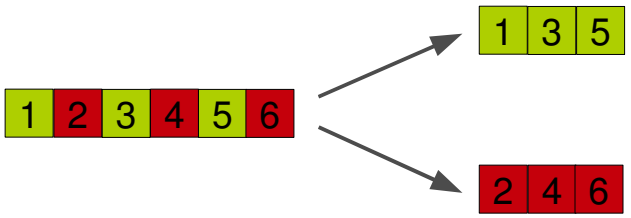
- Advantage/Disadvantage

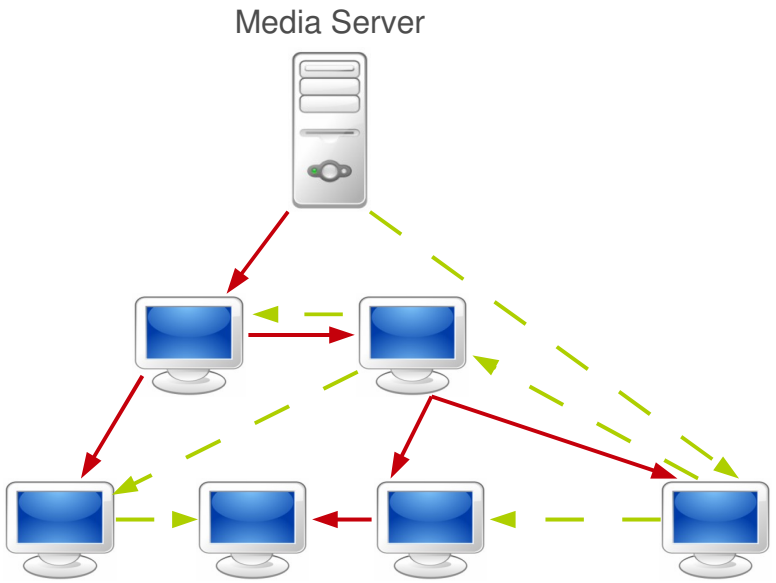
- Advantage
  - The short latency of data delivery.
  - Easy to implement.

- 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?

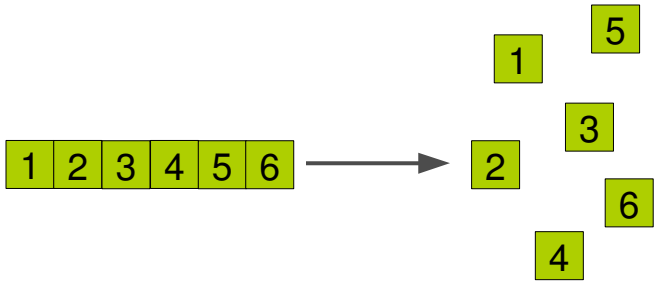
- Advantage/Disadvantage

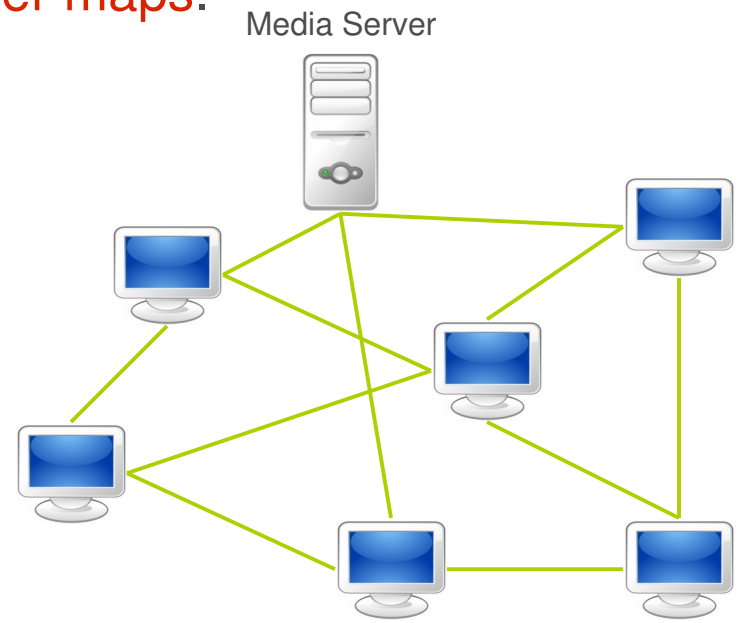
- Advantage
  - Resilient to node failure.
  - Good load balancing

- 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/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 overlay topology is built for data dissemination?
- What algorithm is used for data dissemination?
- How to construct and maintain this overlay?

# 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.

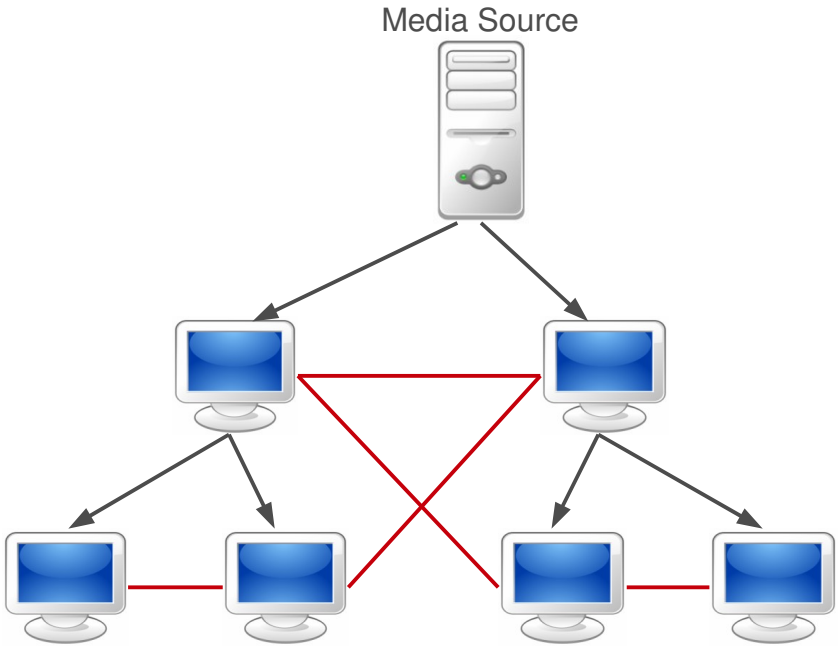
# Pull-based Data Dissemination

- 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

# Push-Pull-based Data Dissemination

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



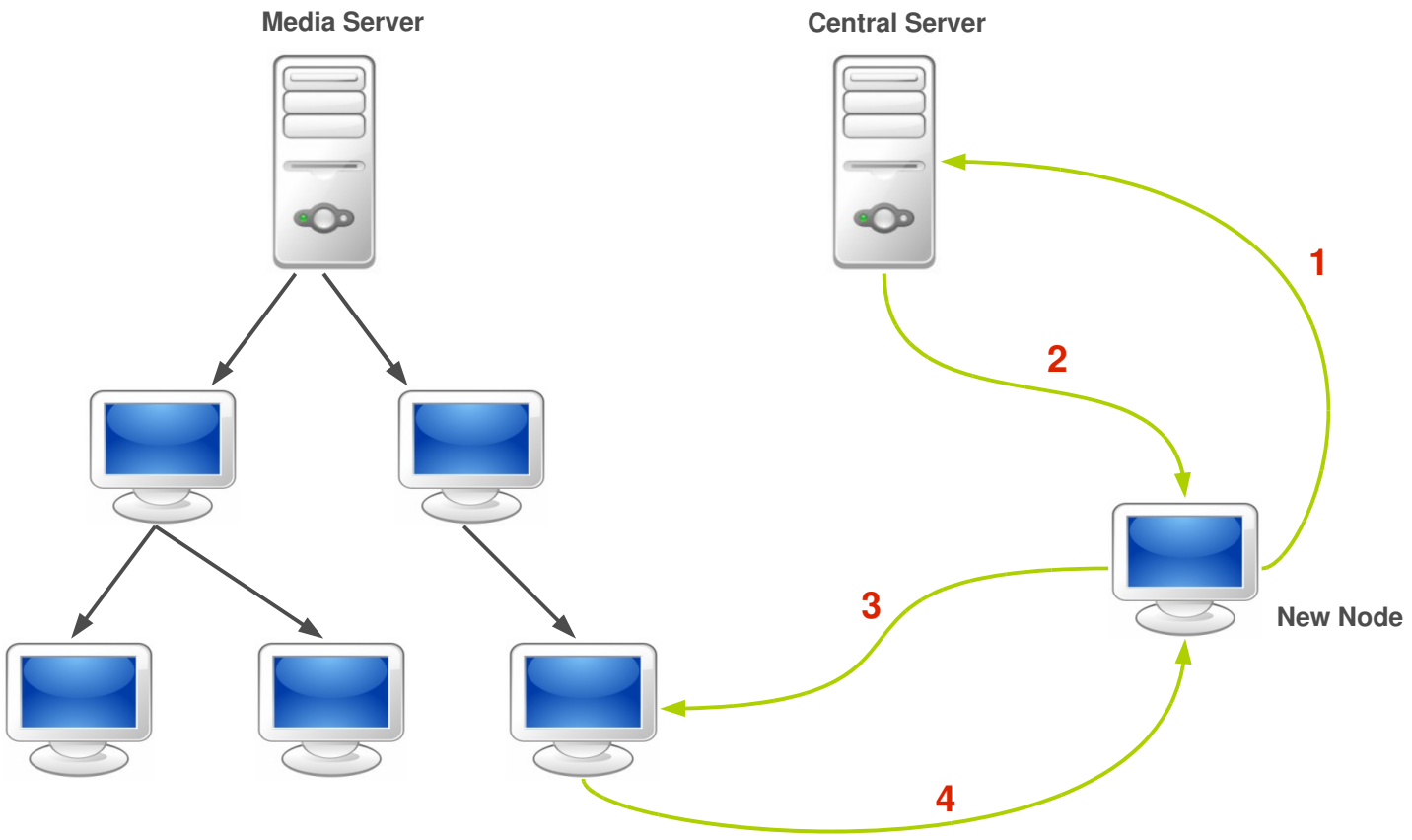
# Main Questions

- What overlay topology is built for data dissemination?
- What algorithm is used for data dissemination?
- How to construct and maintain this overlay?

# 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



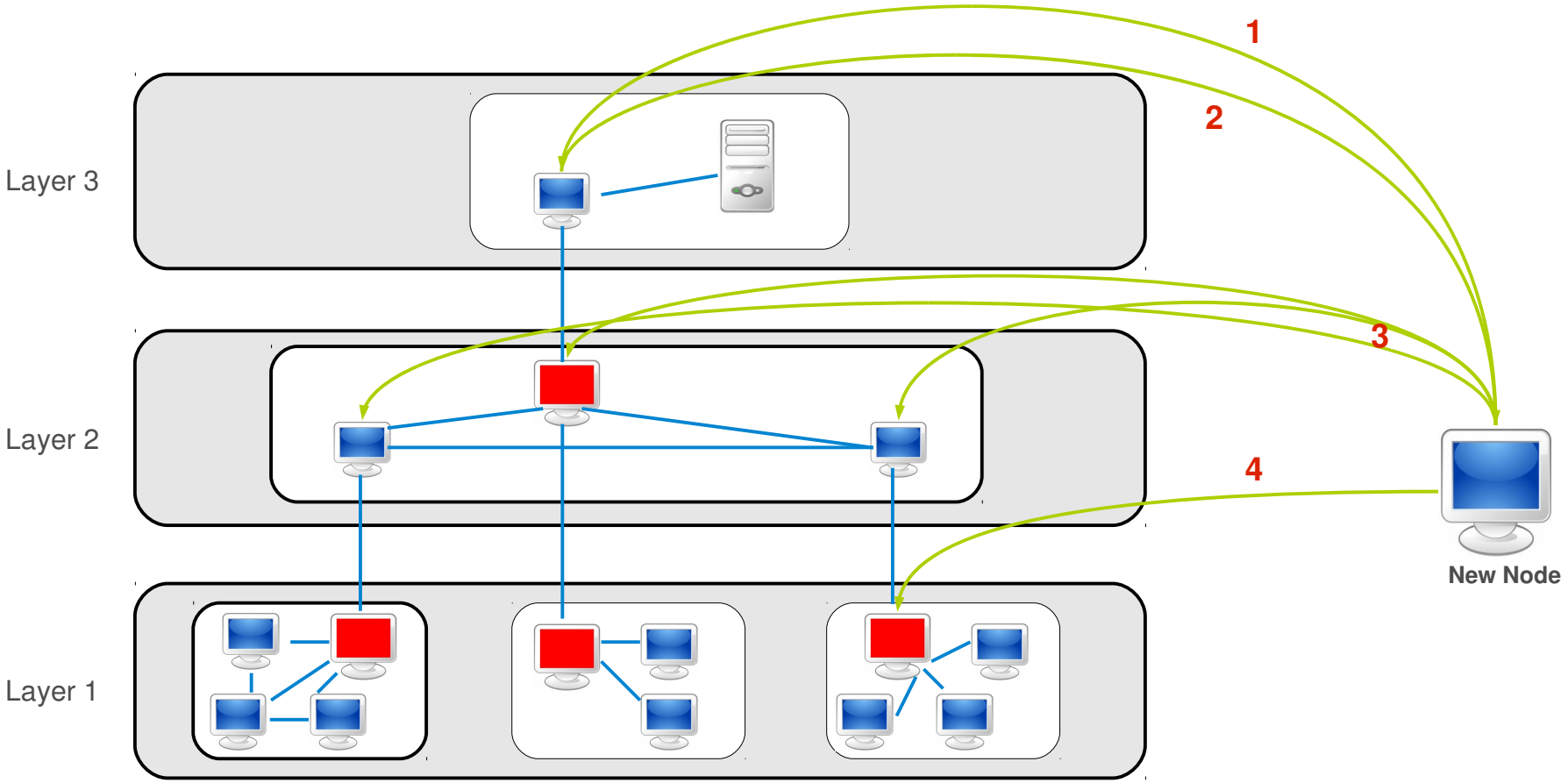
# Centralized Advantage/Disadvantage?

- Advantage/Disadvantage [d]

# Centralized Advantage/Disadvantage?

- Advantage/Disadvantage
- Advantage
  - Fast
  - Easy to apply optimization methods.
  - Easy to implement.
- Disadvantage
  - Not scalable
  - Single point of failure

# Hierarchical Method



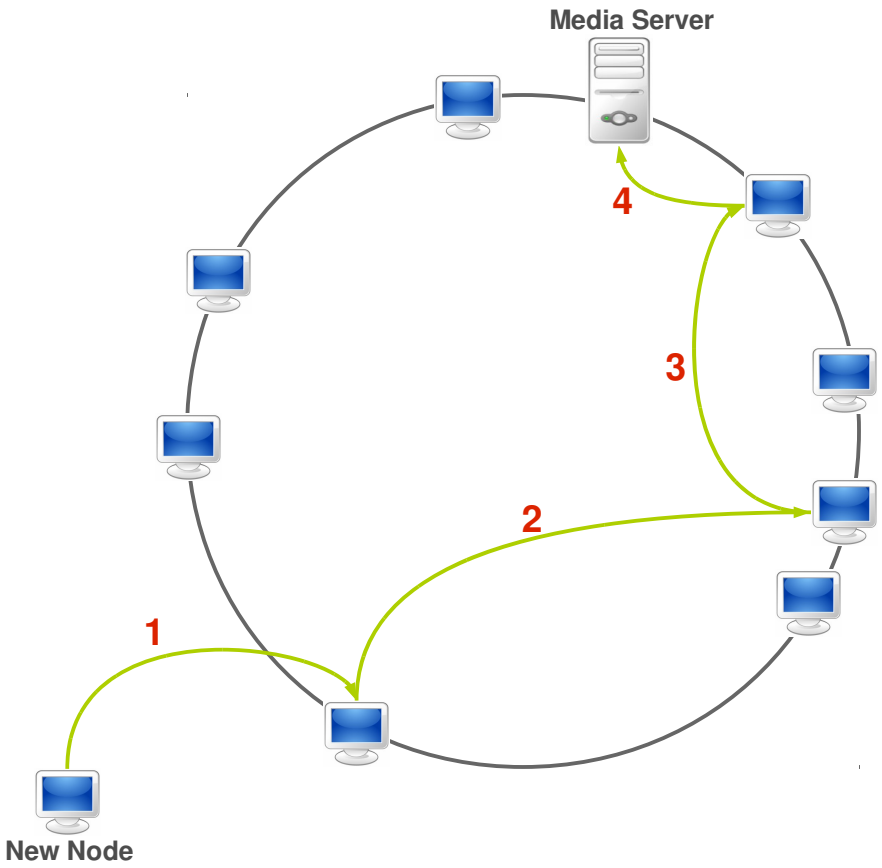
# Hierarchical Advantage/Disadvantage?

- Advantage/Disadvantage [d]

# Hierarchical Advantage/Disadvantage?

- Advantage/Disadvantage
- Advantage
  - Scalable.
  - No single point of failure.
- Disadvantage
  - Slow convergence
  - Difficult to implement

# DHT-based Method



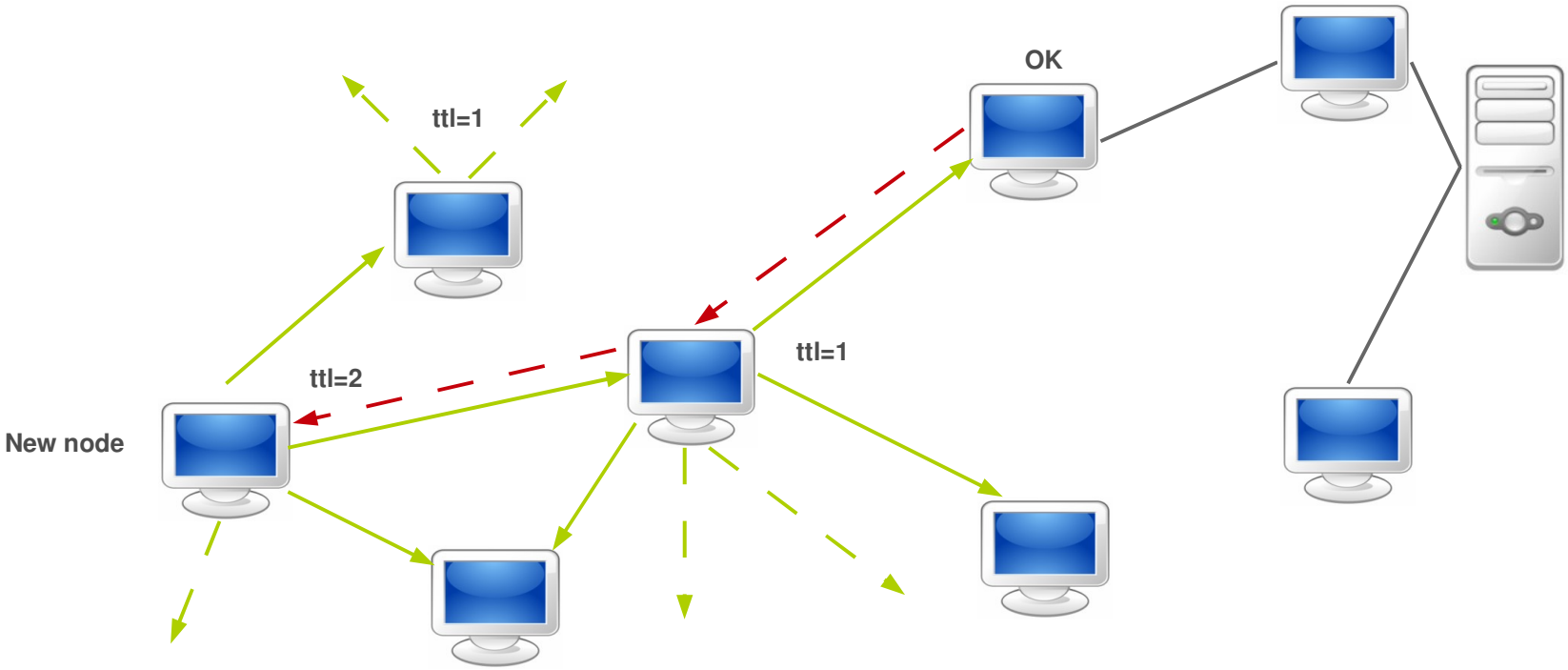
# DHT-based Advantage/Disadvantage?

- Advantage/Disadvantage [d]

# DHT-based Advantage/Disadvantage?

- Advantage/Disadvantage
- Advantage
  - Scalable.
  - No single point of failure.
- Disadvantage
  - Difficult to implement

# Controlled Flooding Method



# Flooding Advantage/Disadvantage?

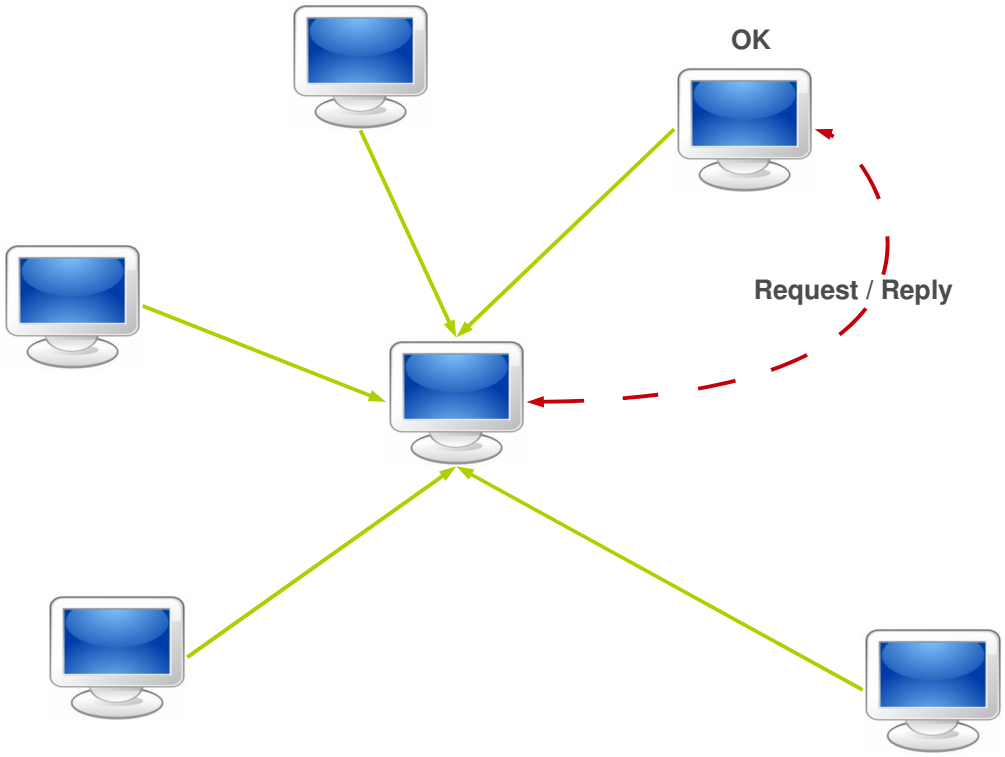
- Advantage/Disadvantage [d]

# Flooding Advantage/Disadvantage?

- Advantage/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/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
- Connectivity problem
- Security in P2P streaming systems
- Sepidar – a P2P streaming system

# 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


# Two Main Classifications

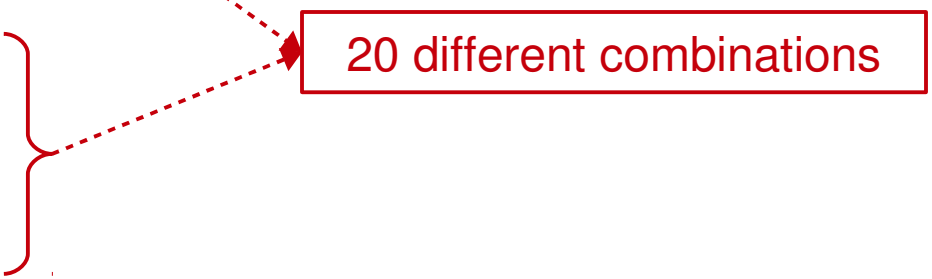
- Data dissemination overlay and its construction.
- How to manage data messages.

# Two Main Classifications

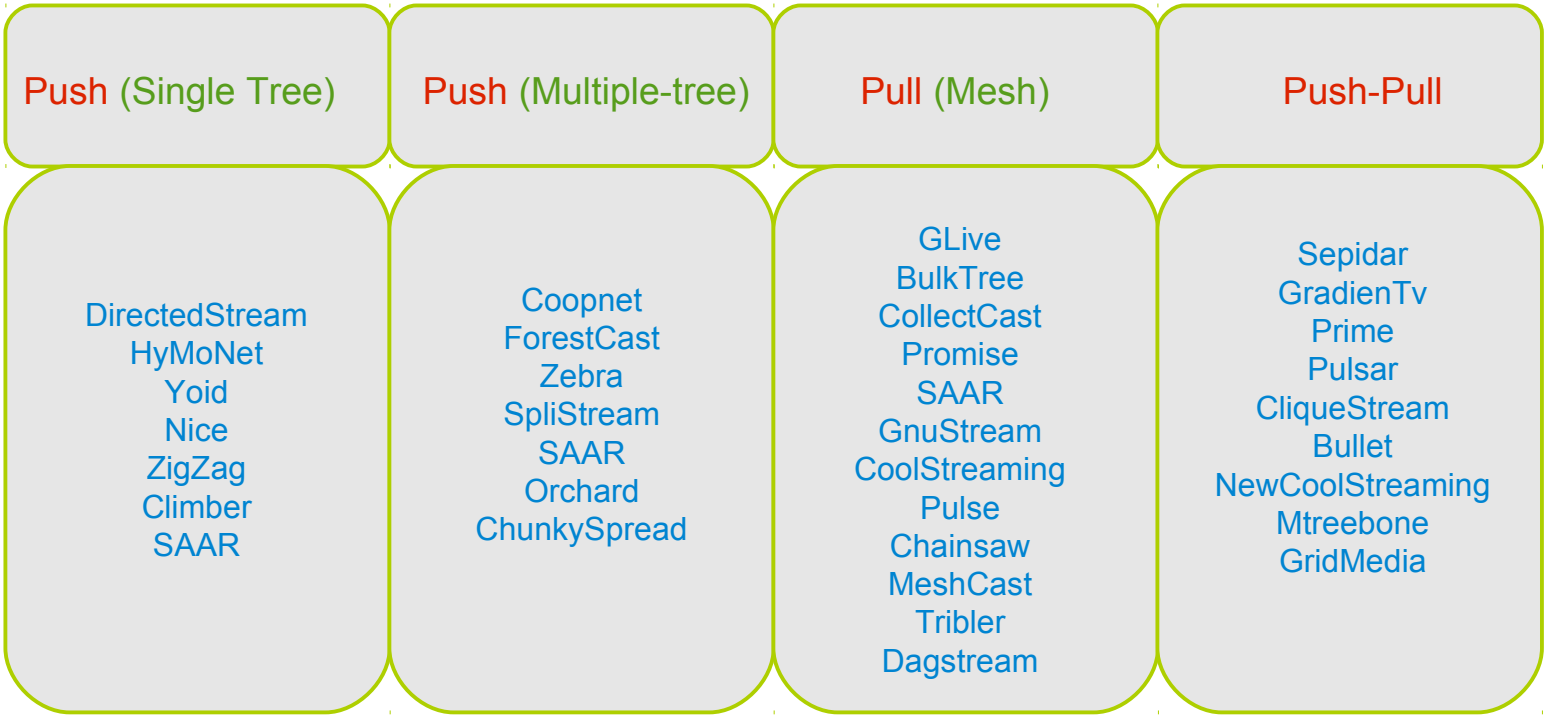
- Data dissemination overlay and its construction.
- How to manage data messages.

# 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



# Data Dissemination Overlay



# Overlay Construction and Maintenance Methods

| | |
|---|---|
| Centralized | DirectedStream, HyMoNet, Yoid, CoopNet<br>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<br>MeshCast, Tribler, DagStream, Bullet, mTreebone, GridMedia |

# All Together

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

# Two Main Classifications

- Data dissemination overlay and its construction.
- How to manage data messages.

# Managing Data Messages

- Source-driven approach
- Receiver-driven approach
- Data-driven approach

# Source-driven Approach

- Use tree(s) rooted at the source to distribute data messages.
- Data messages are pushed.

# Receiver-driven Approach

- Use tree(s) rooted at the receivers.
- Receivers organizes resources (other nodes) that they can obtain the stream.
- Data messages are pushed or pulled from the other nodes.

# Data-driven Approach

- Nodes exchange their buffer maps.
- Data messages are pulled from the other nodes.

# All Together

| | |
|---|---|
| Source-driven | PeerCast, Narada, ZigZag, Nice |
| Receiver-driven | Sepidar, gradienTv, Promise, CollectCast, SplitStream, PeerStreaming, PALS, CoopNet, NewCoolStreaming, ChunkySpread |
| Data-driven | Glive, CoolStreaming, ChainSaw, Pulse |

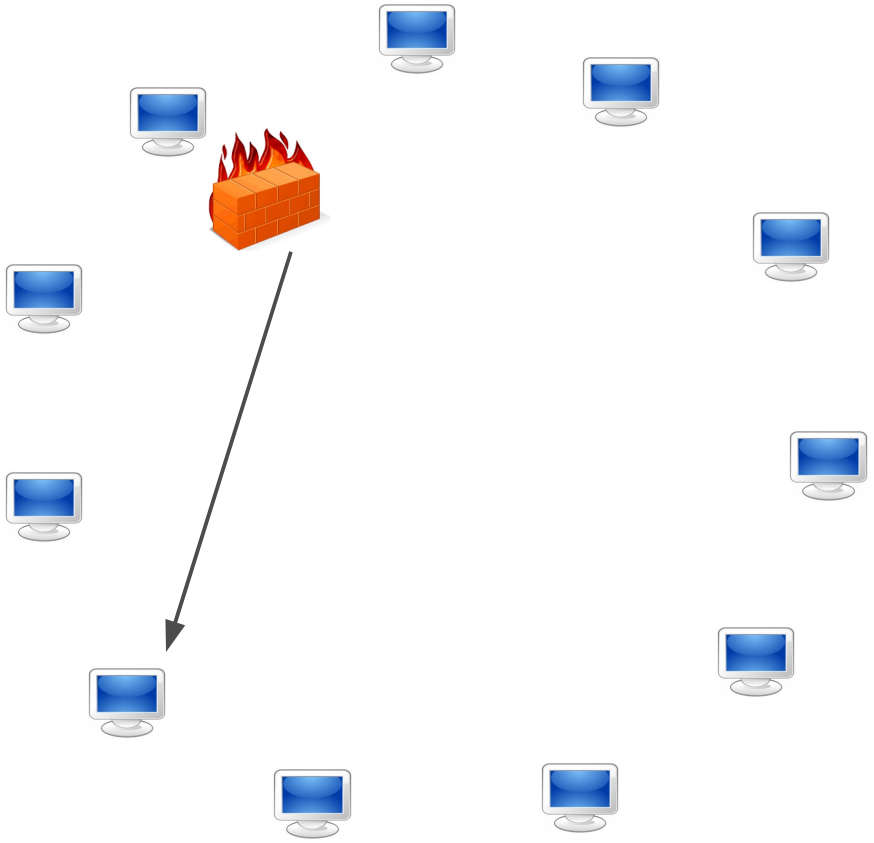
# Outline

- Introduction
- P2P media streaming
- Classification of P2P streaming systems
- Connectivity problem
- Security in P2P streaming systems
- Sepidar – a P2P streaming system

# Connectivity Problem

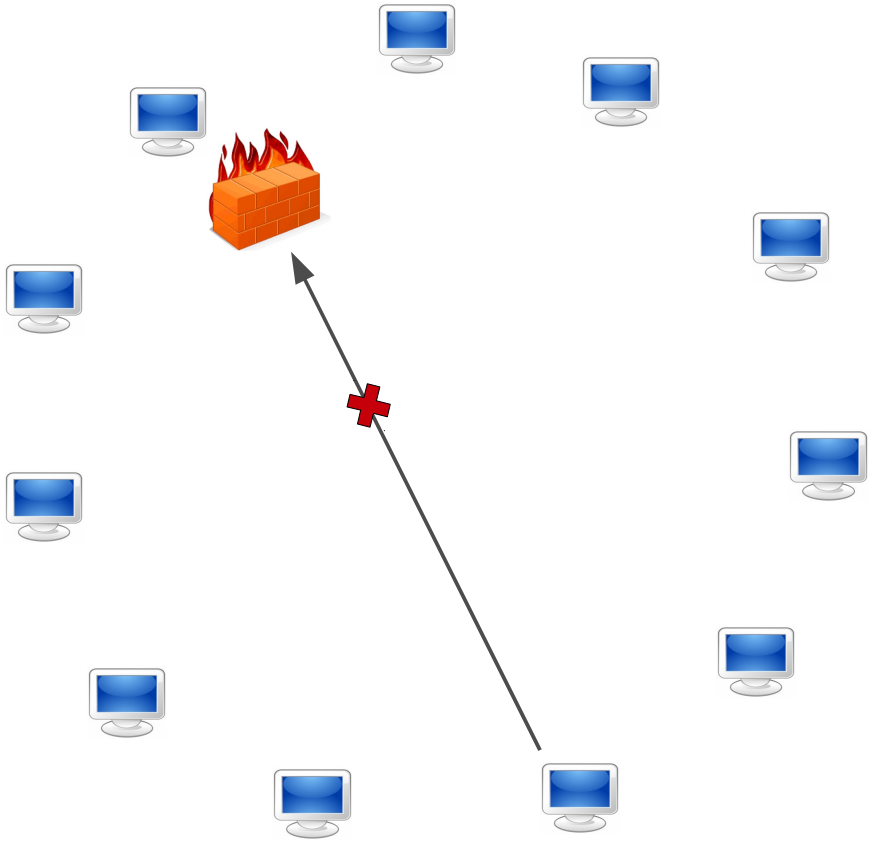
# Connectivity Problem

- In the Internet, a large fraction of the nodes are behind NAT.



# Connectivity Problem

- In the Internet, a large fraction of the nodes are behind NAT.

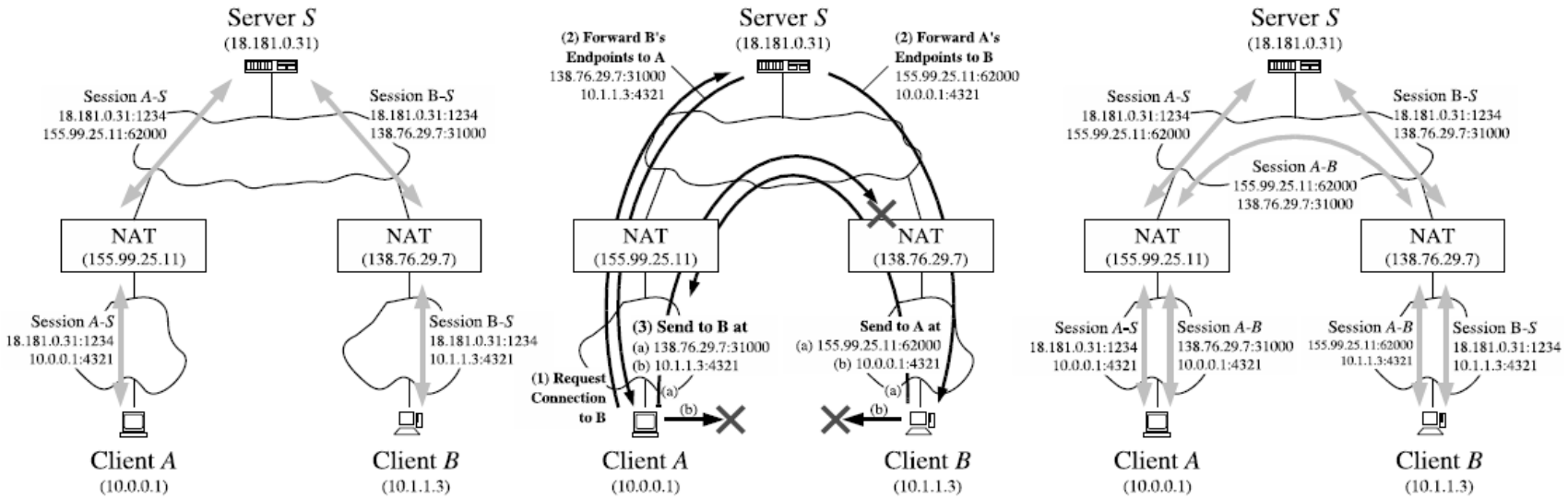


# Common Solutions

- Hole punching
- Relaying

# Hole Punching

- Enables two nodes to establish a direct connection over intermediary NATs with the help of a third party rendezvous server.
  - STUN (UDP) and STUNT (TCP)

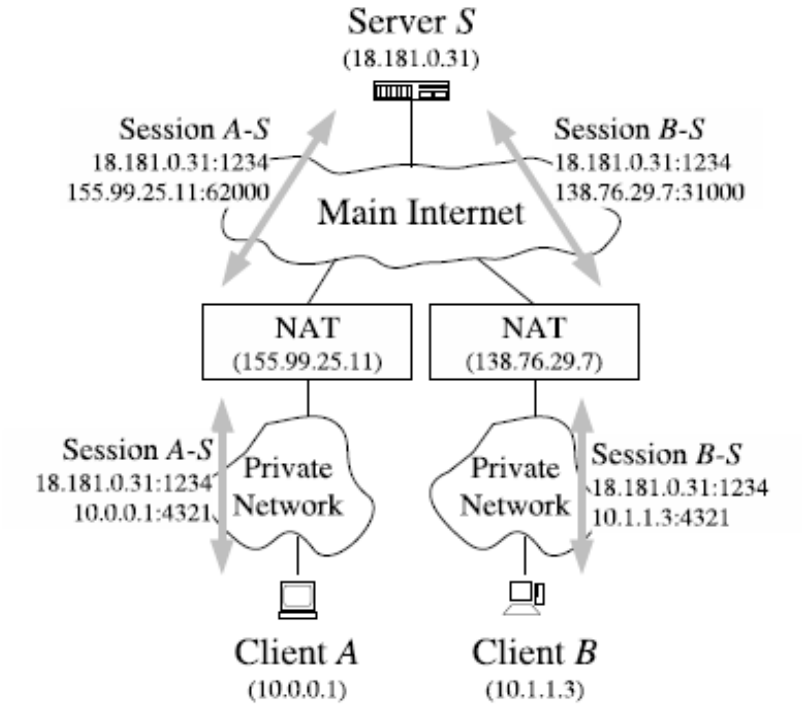



Before Hole Punching | The Hole Punching Process | After Hole Punching

[B. Ford – RFC 5128]

# Relaying

- Relaying can be used either where hole punching techniques do not succeed or where hole punching takes too long to complete.
  - TURN




[B. Ford – RFC 5128]

# Outline

- Introduction
- P2P media streaming
- Classification of P2P streaming systems
- Connectivity problem
- Security in P2P streaming systems
- Sepidar – a P2P streaming system

# Security in P2P Streaming Systems

# 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.

# 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

- Each of the presented attacks can exacerbate by collusion.
- A collection of nodes conduct correlated attack.

# Outline

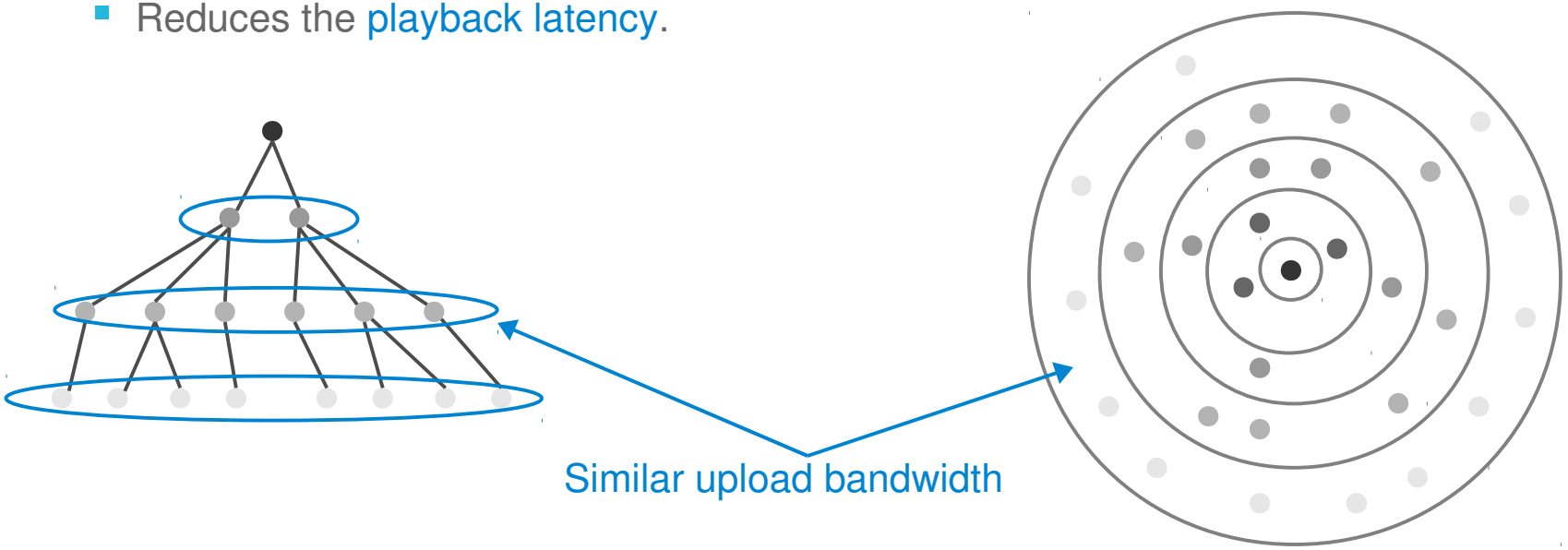
- Introduction
- P2P media streaming
- Classification of P2P streaming systems
- Connectivity problem
- Security in P2P streaming systems
- Sepidar – a P2P streaming system

# Sepidar

# Problem Description (1/3)

- Building an overlay network, such that:
  - Nodes with higher upload bandwidth are positioned closer to the media source.
  - Nodes with similar upload bandwidth become neighbours.

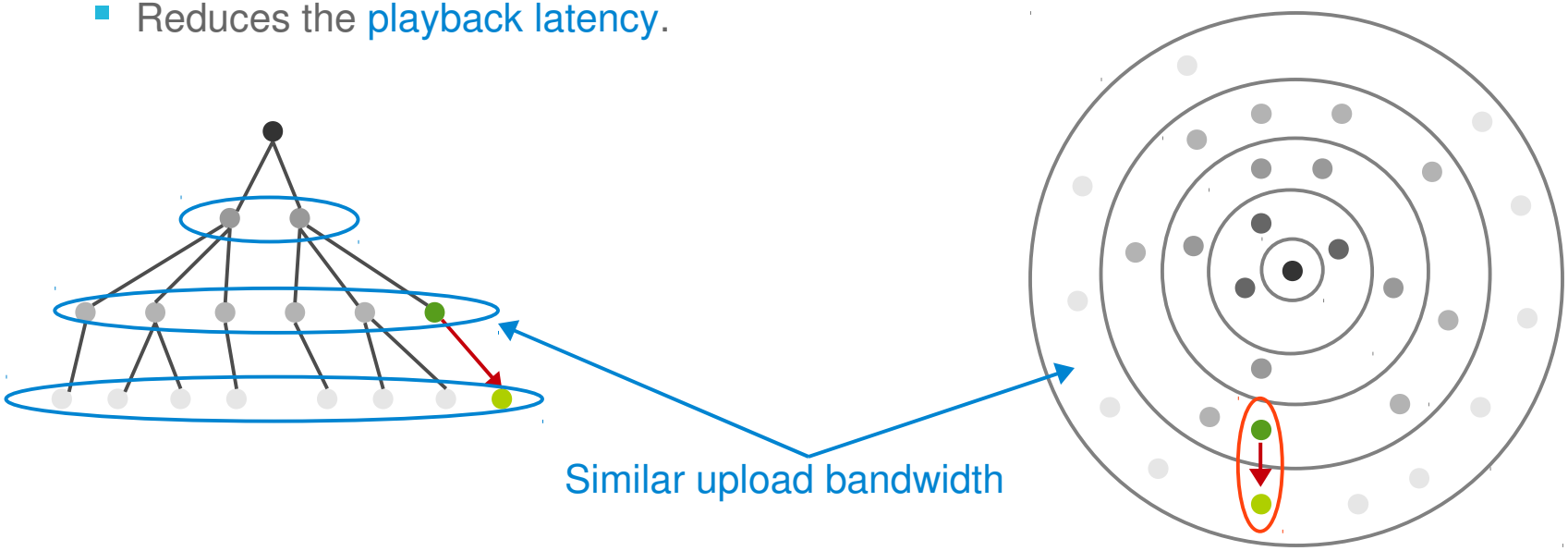
- Results:
  - Reduces the average number of hops from nodes to the media source.
  - Reduces the probability of streaming disruptions.
  - Reduces the playback latency.



# Problem Description (1/3)

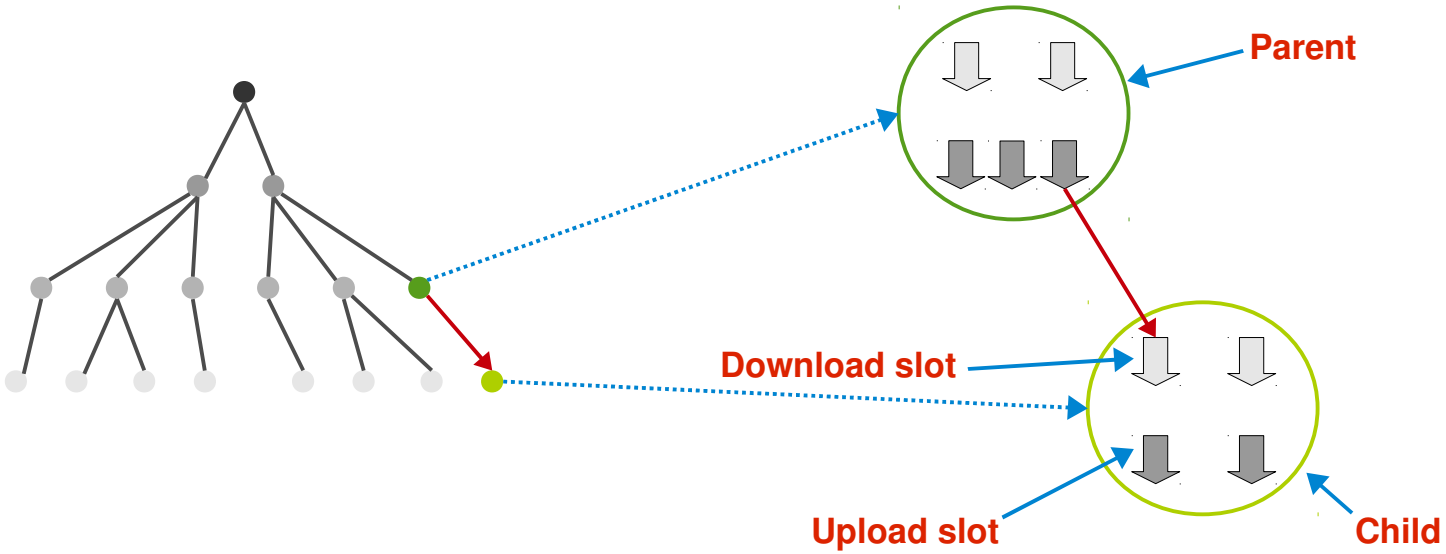
- Building an overlay network, such that:
  - Nodes with higher upload bandwidth are positioned closer to the media source.
  - Nodes with similar upload bandwidth become neighbours.

- Results:
  - Reduces the average number of hops from nodes to the media source.
  - Reduces the probability of streaming disruptions.
  - Reduces the playback latency.



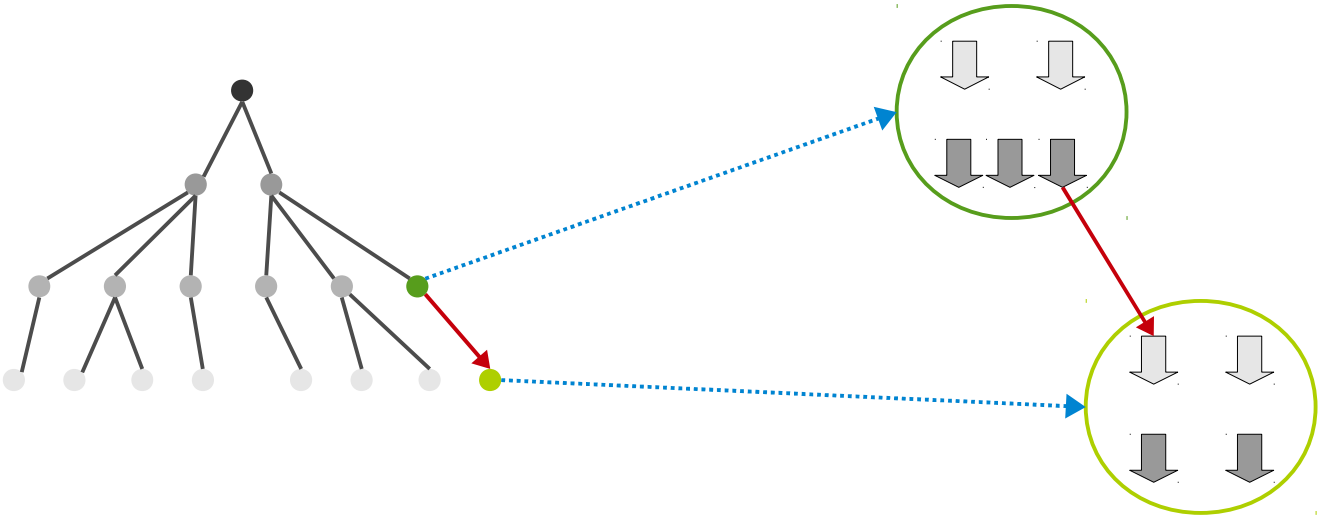
# Problem Description (2/3)

- A node can create a bounded number of download connections, and accept a bounded number of upload connections.
- A parent node pushes data block from its upload connection, and a child node receives it from its download connection.



# Problem Description (3/3)

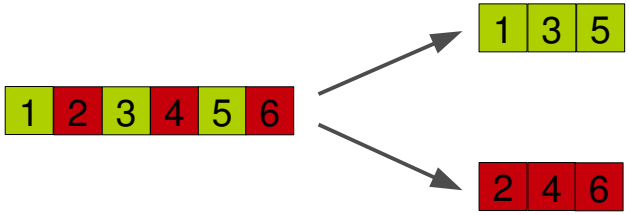
- Problem:
  - How to assign upload slots to download slots?



# The Naïve Solution

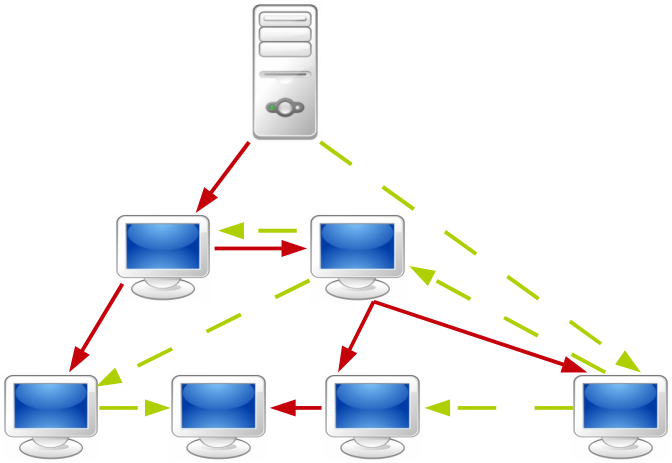
# Multiple-tree Overlay

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




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

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



# How to Assign Upload Slots to Download Slots?

- This can be modelled as an 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.

# How to Assign Upload Slots to Download Slots?

- This can be modelled as an 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 knows only a small number of nodes in the system (partial view).
  - The nodes of partial view are selected randomly.
  - We used Cyclon in our implementation.

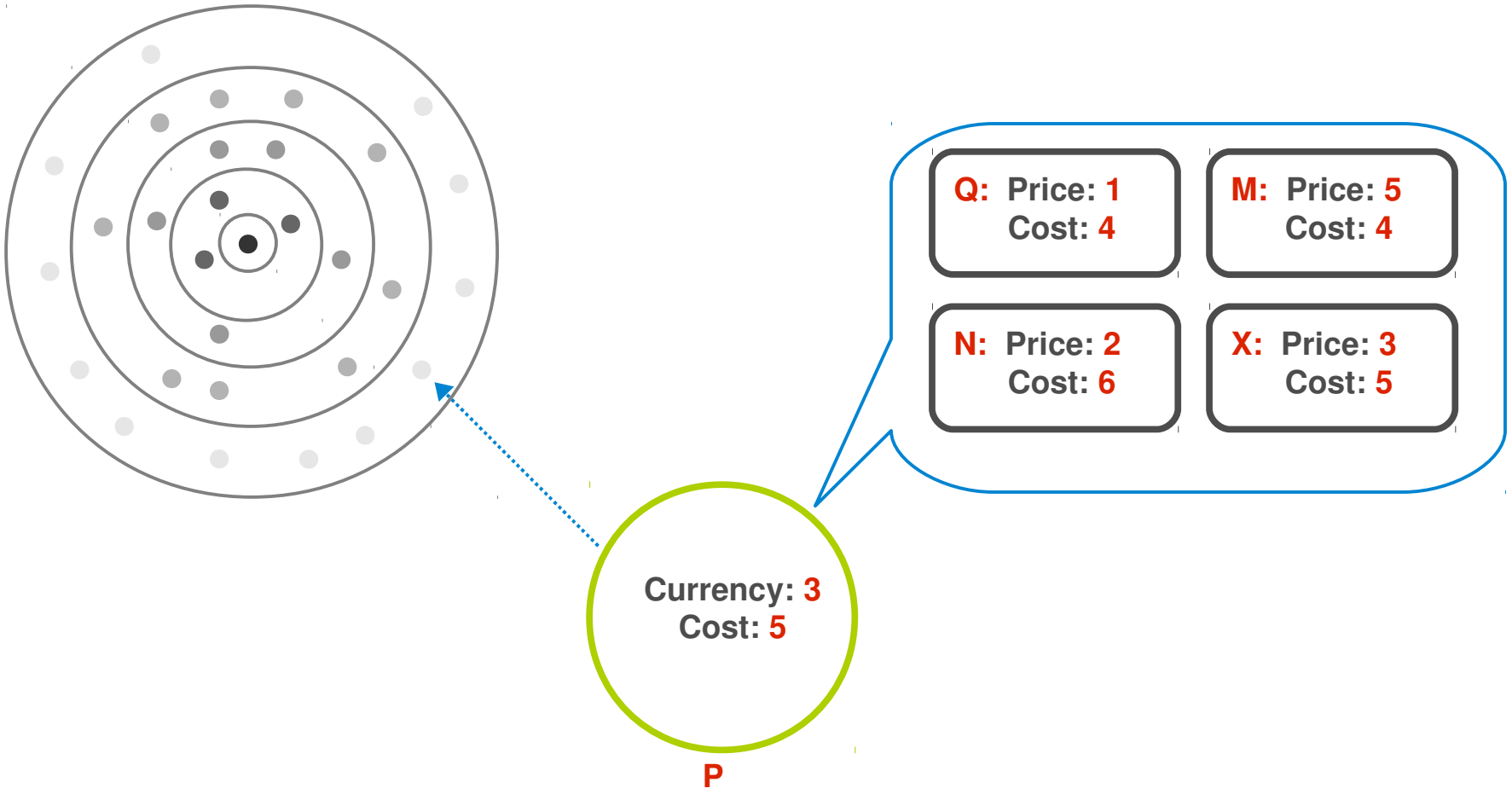
# Node Properties

- 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.

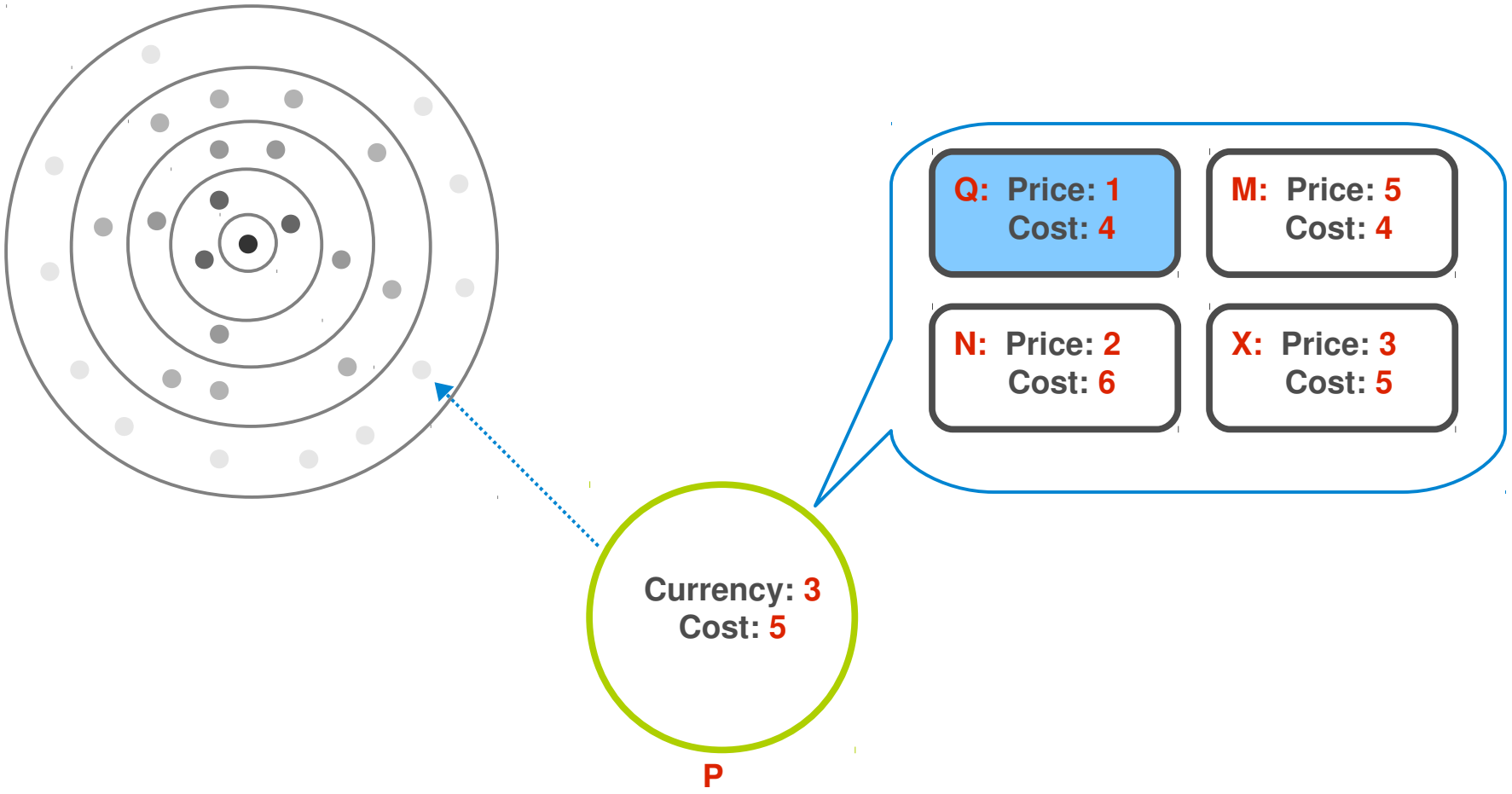
# Streaming Overlay Construction

- 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.

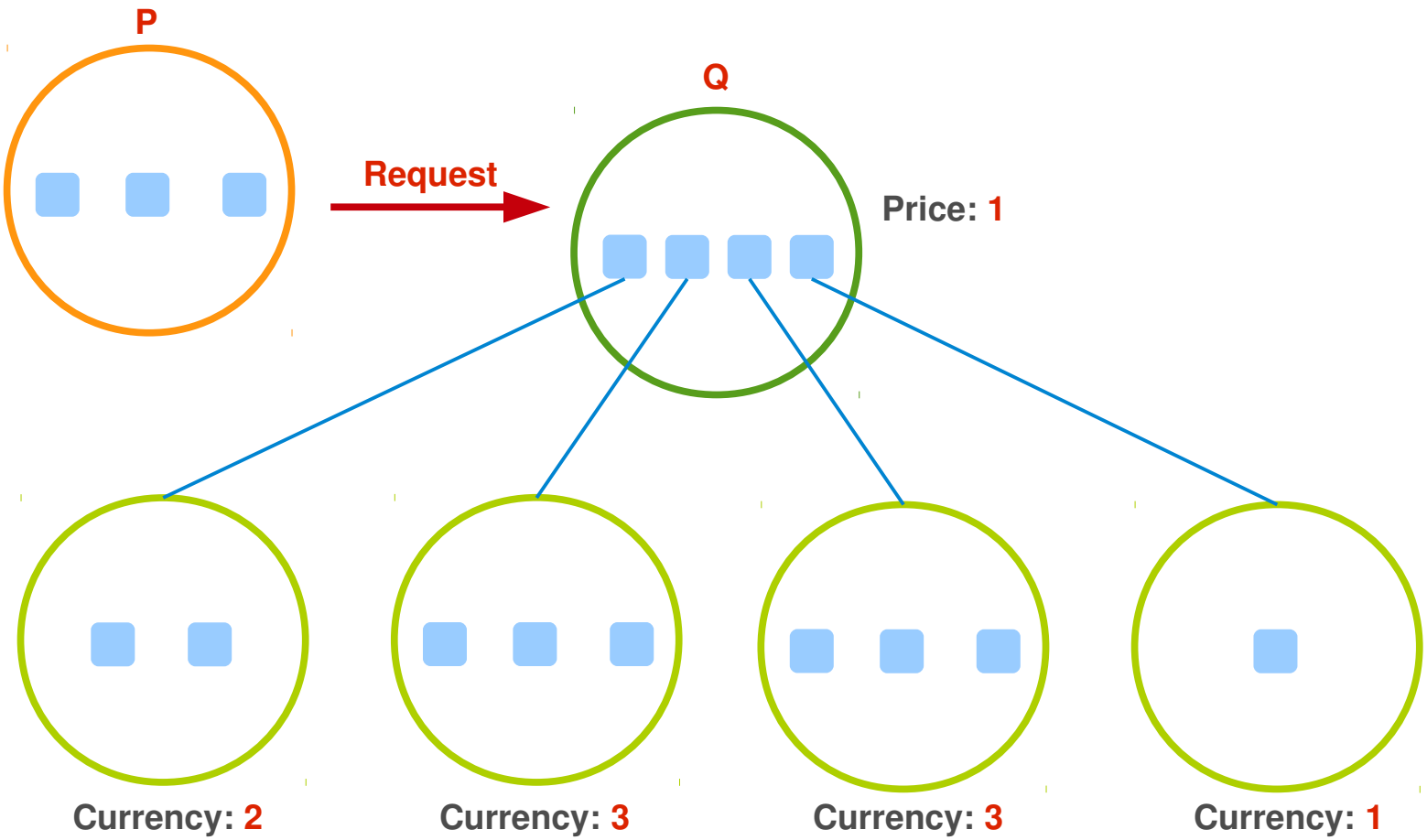
# The Market Model – Child Side



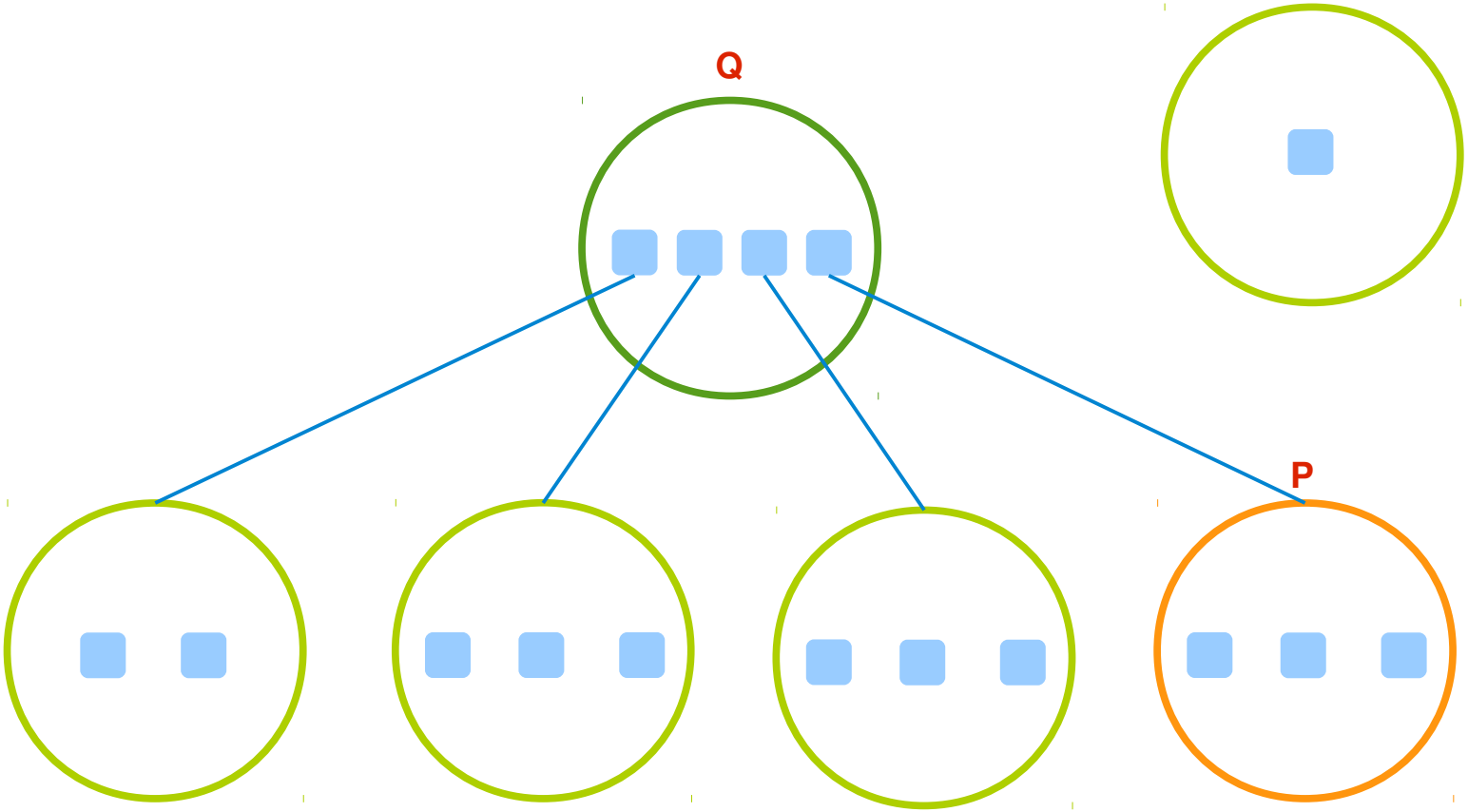
# The Market Model – Child Side



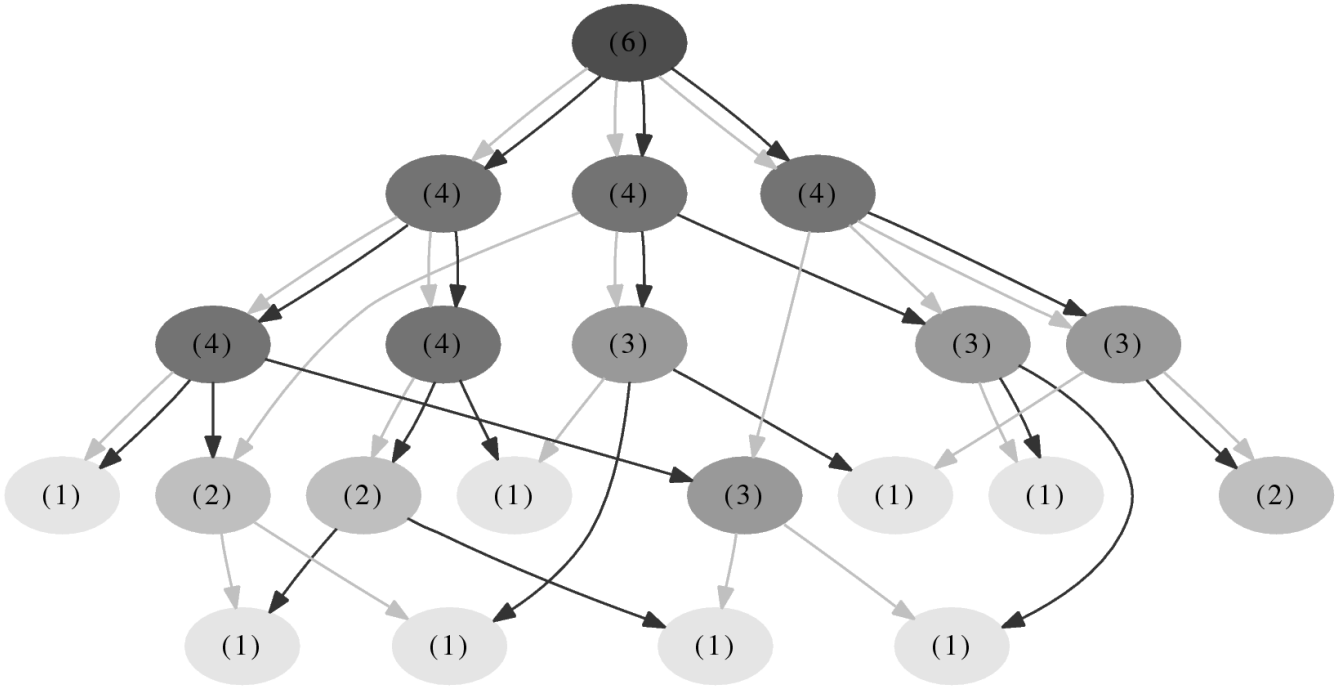
# The Market Model – Parent Side



# The Market Model – Parent Side



# Constructed Streaming Overlay




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

# Optimization

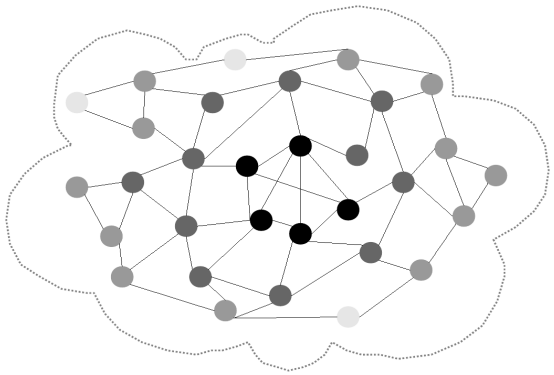
# Node Discovery

- 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.

# The Gradient Overlay

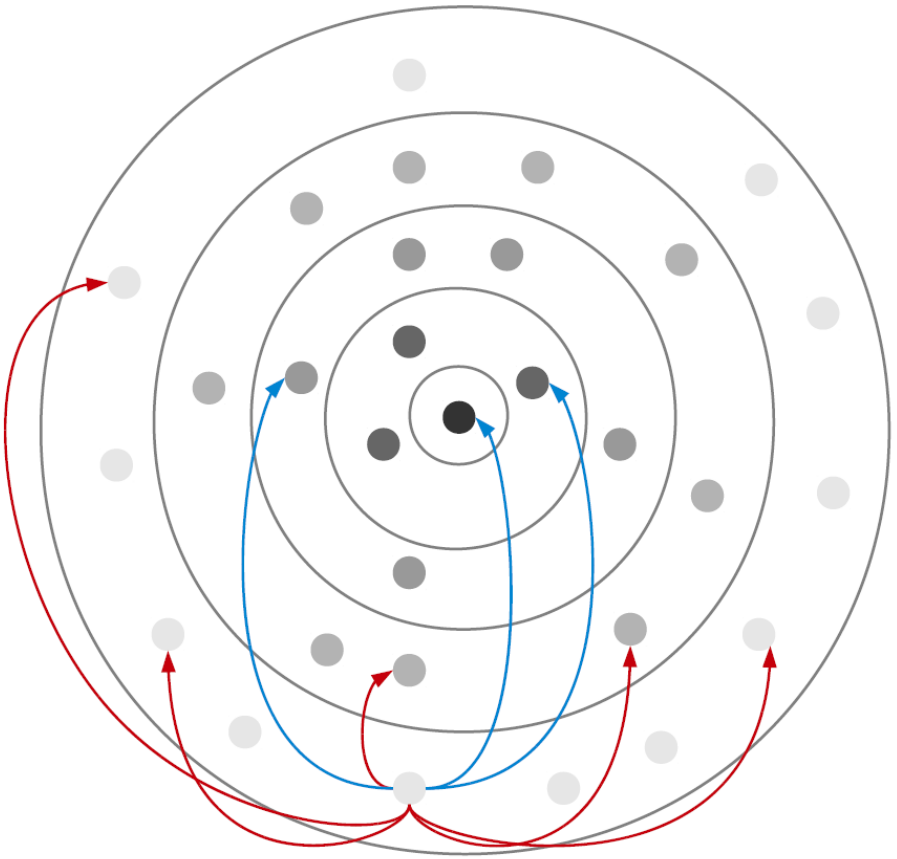
- The Gradient overlay is a class of P2P overlays that arranges nodes using a local utility function at each node, such that nodes are ordered in descending utility values away from a core of the highest utility nodes.

- 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.



The Gradient overlay

# Partial Views Using The Gradient Overlay

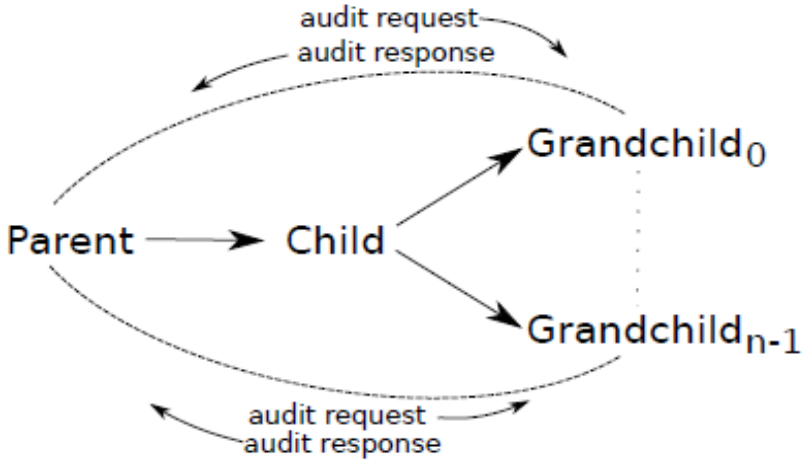


Similar-view pointer

Finger pointer

# Handling Free-riders

# Handling Free-riders



# Handling Free-riders

- After detecting a node as a free-rider, the parent node p decreases its own price (p's price) to zero and as a punishment considers the free-rider node q as its child with the lowest currency.
- On the next bid from another node, p replaces the free-rider node with the new node.

# DONE!

# A Page To Remember

- Media Streaming
  - Live
  - VoD
- Client-Server mode
  - Expensive

- P2P model
  - The peers can help each other and the capacity increases with the number of peers.

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

# Question?

# References


- [1] W. -P. Ken Yiu, Xing Jin, and S. -H. Gary Chan. 2007. **Challenges and Approaches in Large-Scale P2P Media Streaming**. IEEE MultiMedia 14, 2 (April 2007), 50-59.
- [2] Y. Liu, Y. Guo, and C. Liang, **A survey on peer-to-peer video streaming systems**, in Journal of Peer-to-Peer Networking and Applications, by Springer New York, Feburary, 2008.
- [3] Thomas Silverston and Olivier Fourmaux. 2006. **Source vs Data-driven Approach for Live P2P Streaming**. In Proceedings of the International Conference on Networking, International Conference on Systems and International Conference on Mobile Communications and Learning Technologies (ICNICONSMCL '06). IEEE Computer Society, Washington, DC, USA, 99-.
- [4] G. Gheorghe, R. Lo Cigno, A. Montresor, **Security and privacy issues in P2P streaming systems: A survey**, Peer-to-Peer Networking and Applications (on-line 23 April 2010), Springer.
- [5] Amir H. Payberah, Fatemeh Rahimian, Seif Haridi, Jim Dowling, **Sepidar: Incentivized Market-Based P2P Live-Streaming on the Gradient Overlay Network**, ism, pp.1-8, 2010 IEEE International Symposium on Multimedia, 2010.