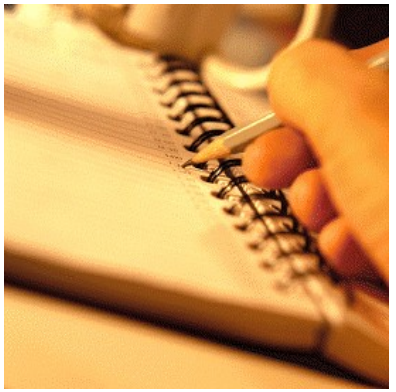
Distributed Optimization of P2P Media Delivery Overlays

Amir H. Payberah
amir@sics.se

Supervisors
Jim Dowling – Seif Haridi

Outline

- Introduction
- Contribution
  - P2P live streaming
  - NAT friendly peer sampling
- Summary and Future work



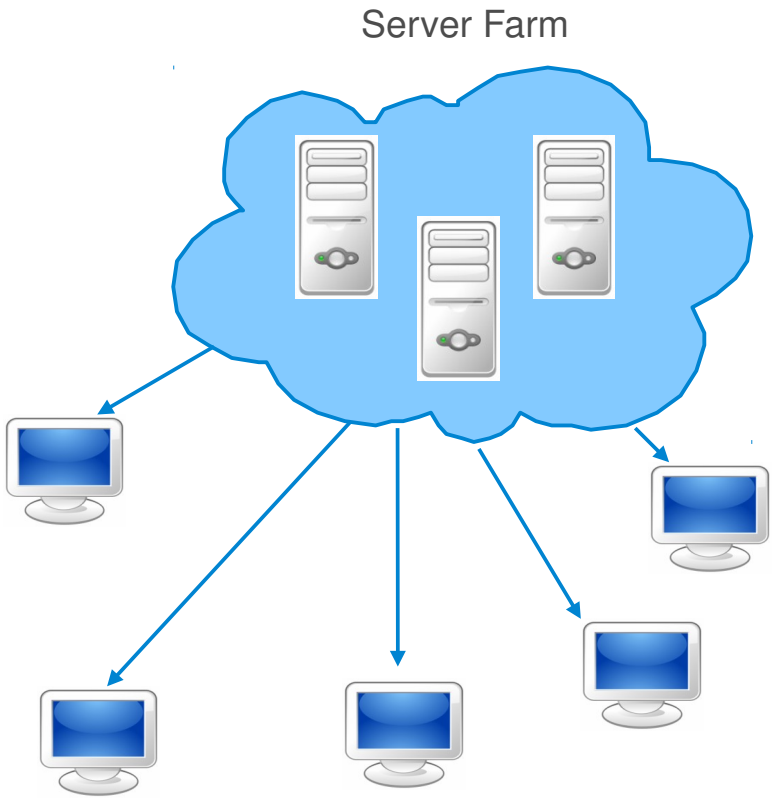
Introduction

Media Streaming

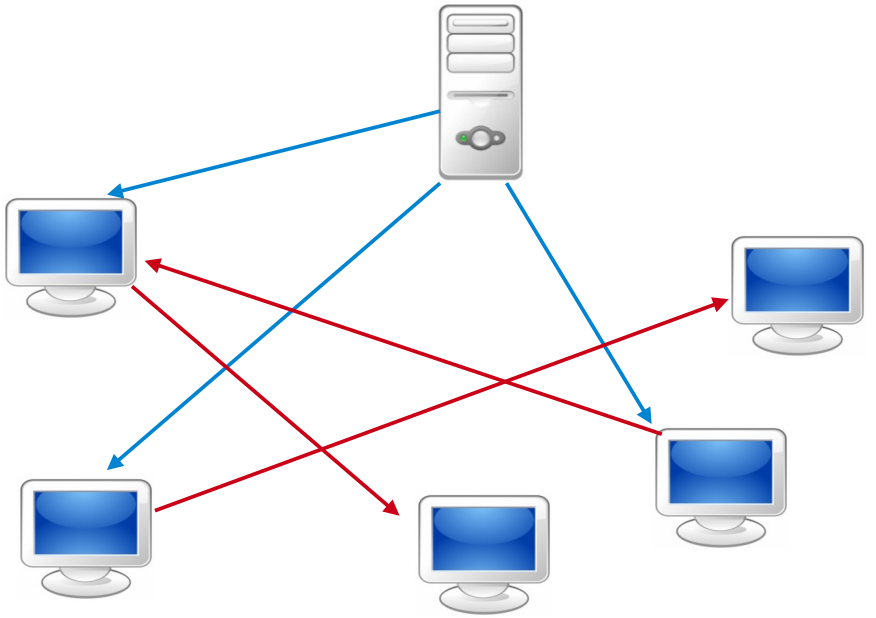
- Media streaming is a multimedia that is sent over a network and played as it is being received by end users.
- Users do not need to wait to download all the media.
- They can play it while the media is delivered by the provider.
- It could be
  - Live Media Streaming
  - Video on Demand (VoD)

Solutions for Media Streaming

Client-Server



Peer-to-Peer



P2P Streaming Challenges

- Data should be received with respect to certain timing constraints.
- Nodes join, leave and fail continuously (churn).
- Network capacity changes.
- Free-riding problem.
- Connectivity Problem.

Contribution

My Contributions

- A distributed market model to construct P2P streaming overlays.
- A NAT-friendly gossip-based peer sampling service.

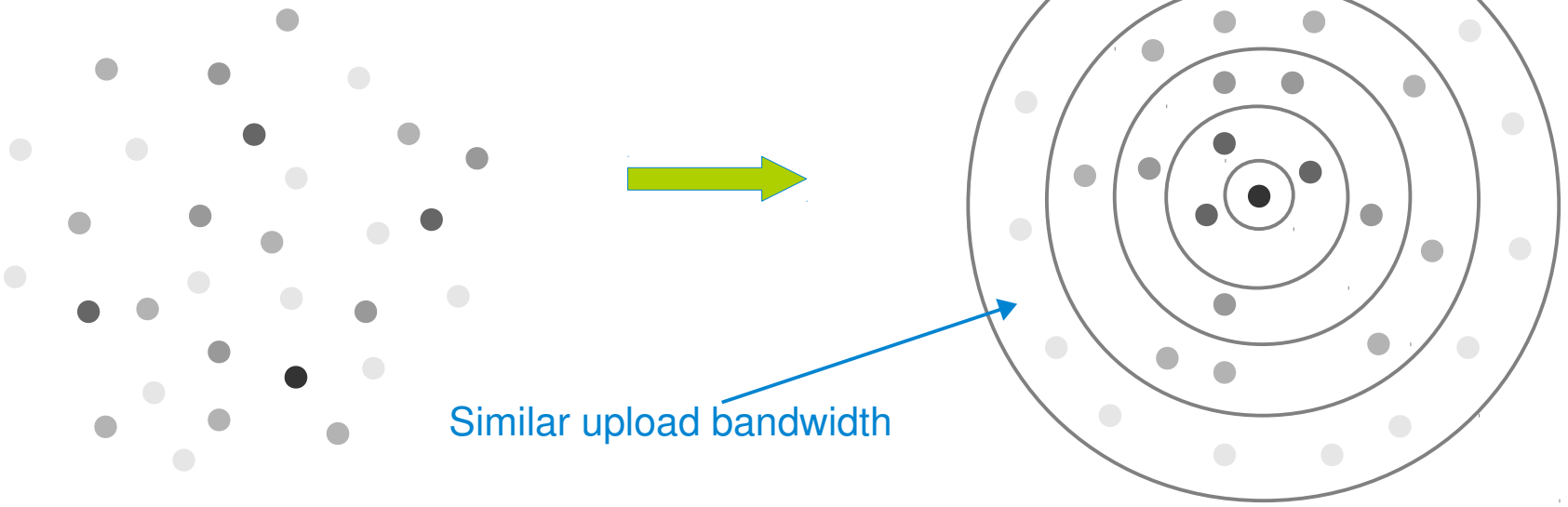
My Contributions

- A distributed market model to construct P2P streaming overlays.
- A NAT-friendly gossip-based peer sampling service.

Problem Description

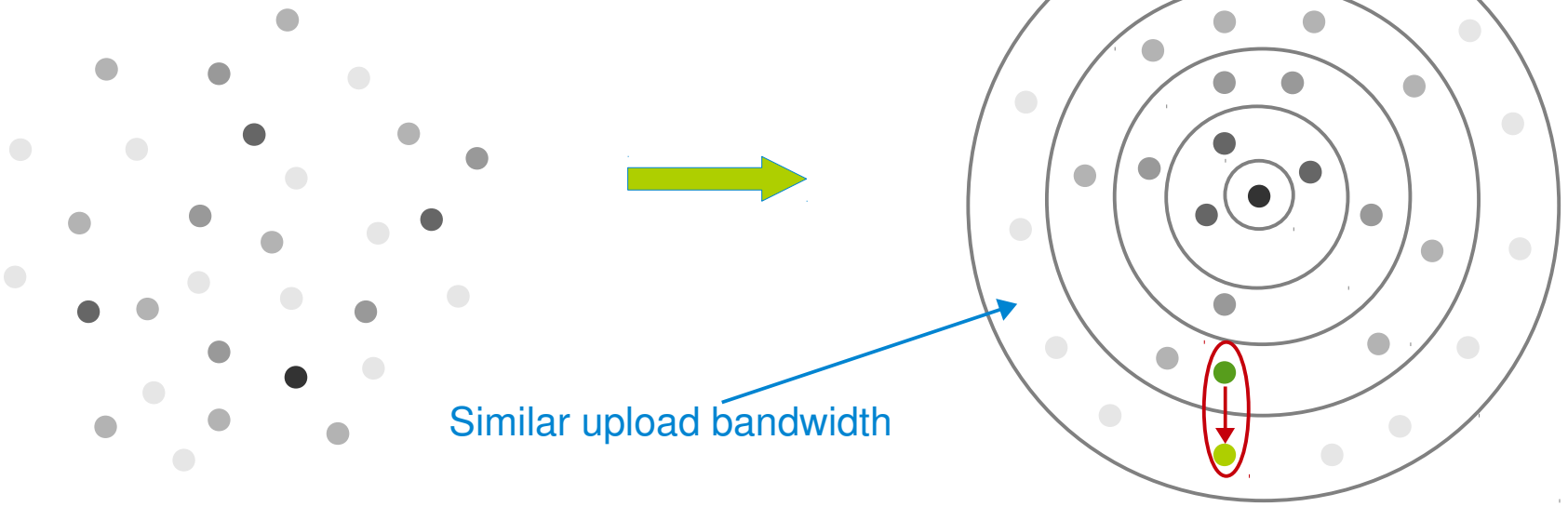
Problem Description (1/4)

- 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)

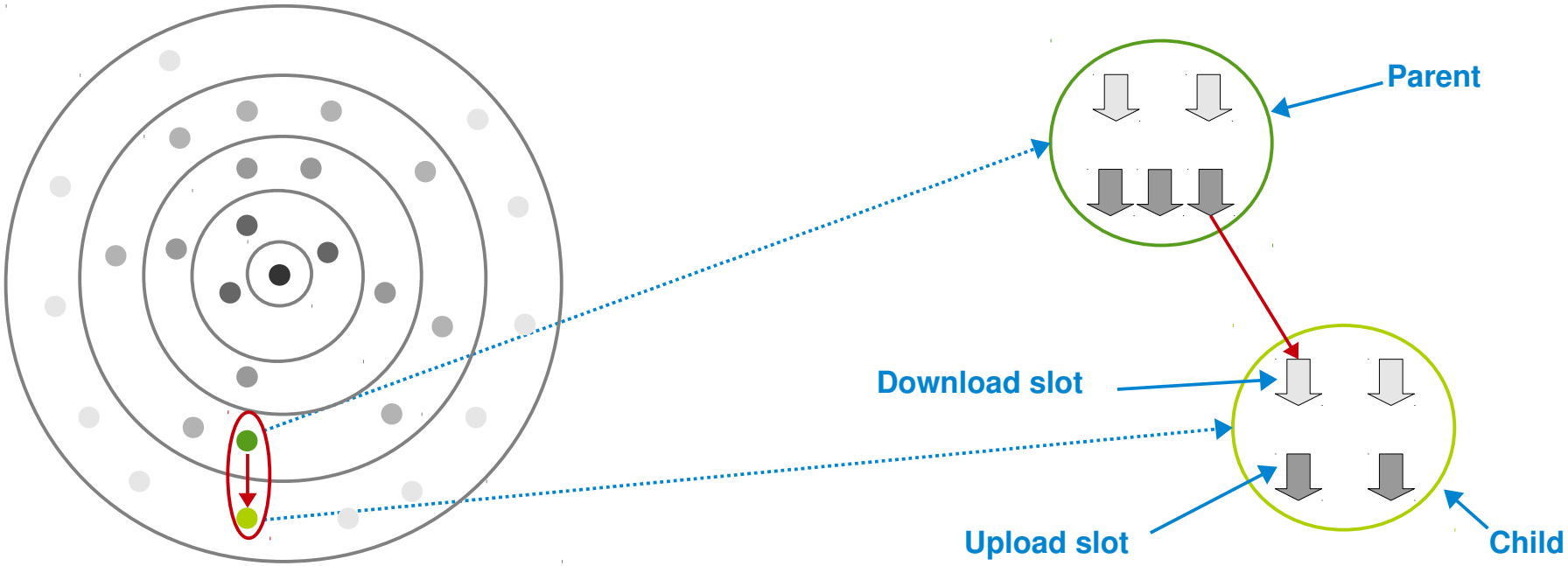
- Building a streaming overlay network, such that:
  - Nodes with higher upload bandwidth are positioned closer to the media source.
  - Nodes with similar upload bandwidth become neighbours.
- Reduces:
  - Average number of hops
  - Streaming disruptions
  - Playback latency



Problem Description (2/4)

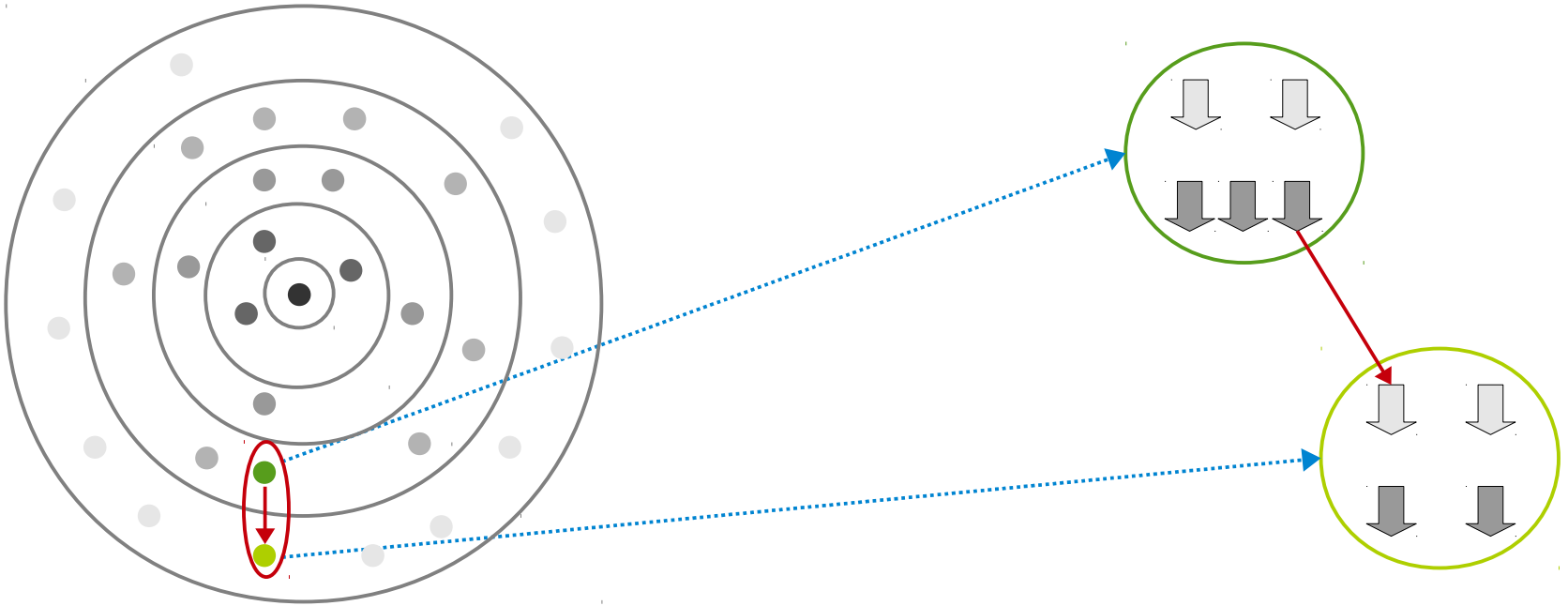
- A node can create a bounded number of download connections, and accept a bounded number of upload connections.

- A parent node sends data block from its upload connection, and a child node receives it from its download connection.



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).

Gradientv and Sepidar

Design Space

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

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

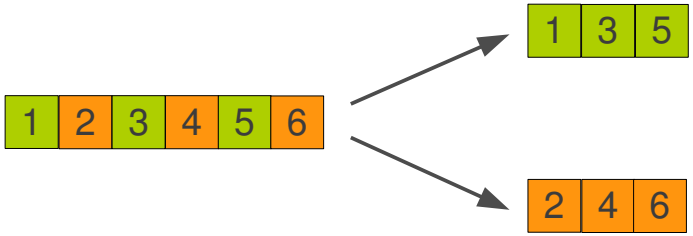
Design Space

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

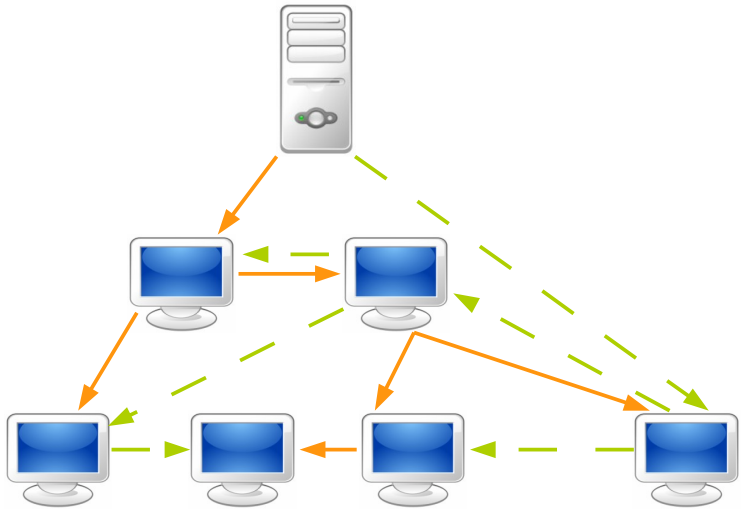
Multiple-tree Overlay

- 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, …

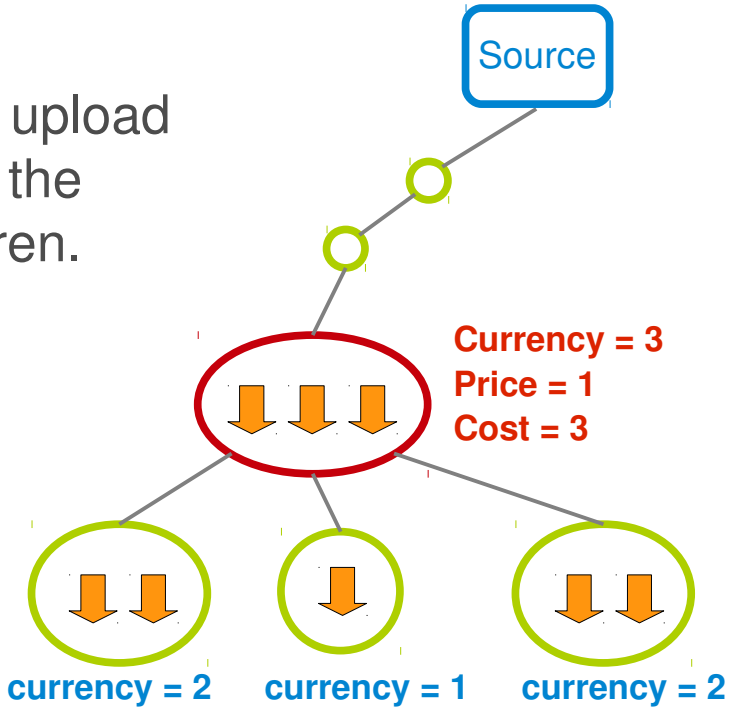


- Construct one tree for each stripe.



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.

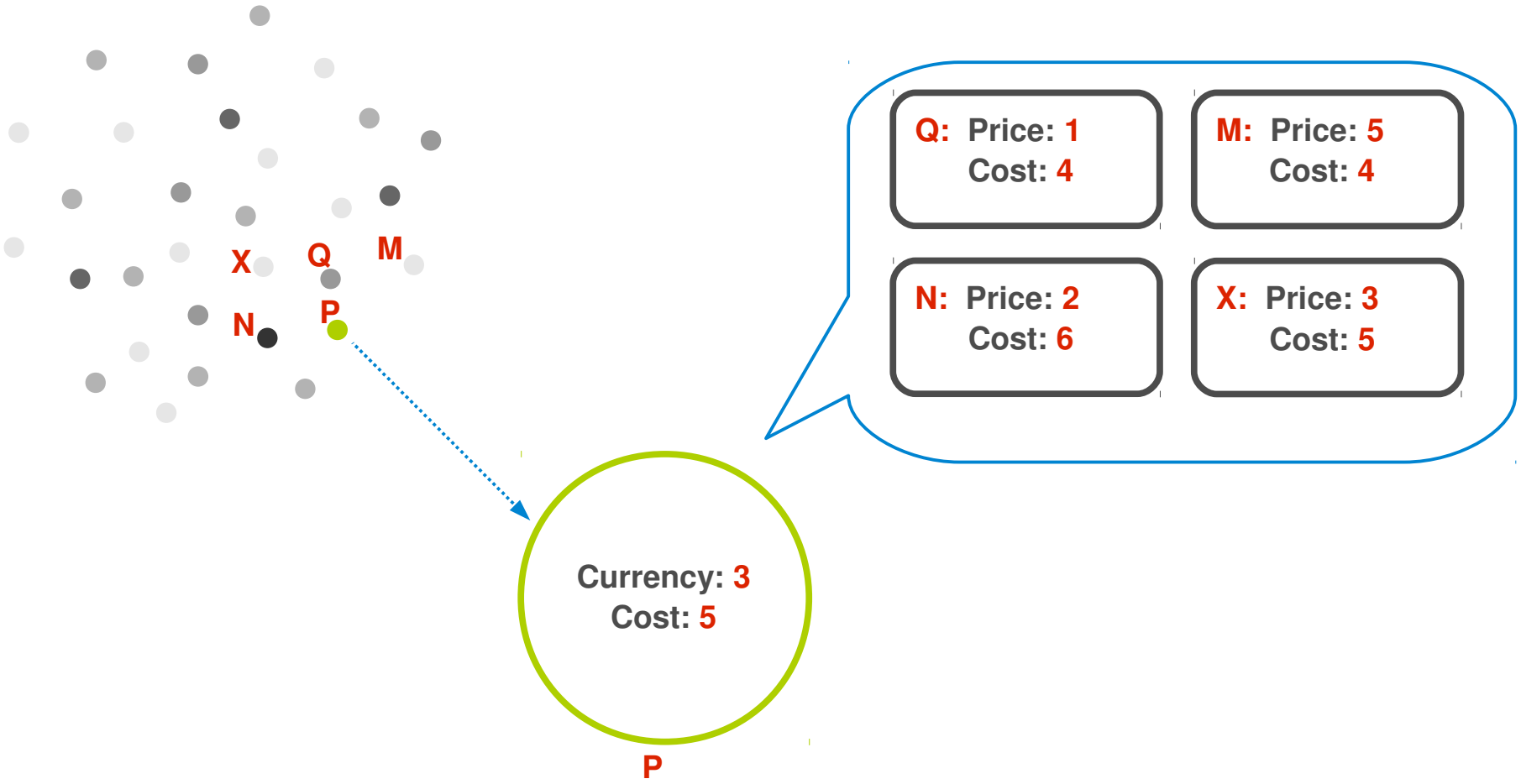


The Market Model - Streaming Overlay Construction

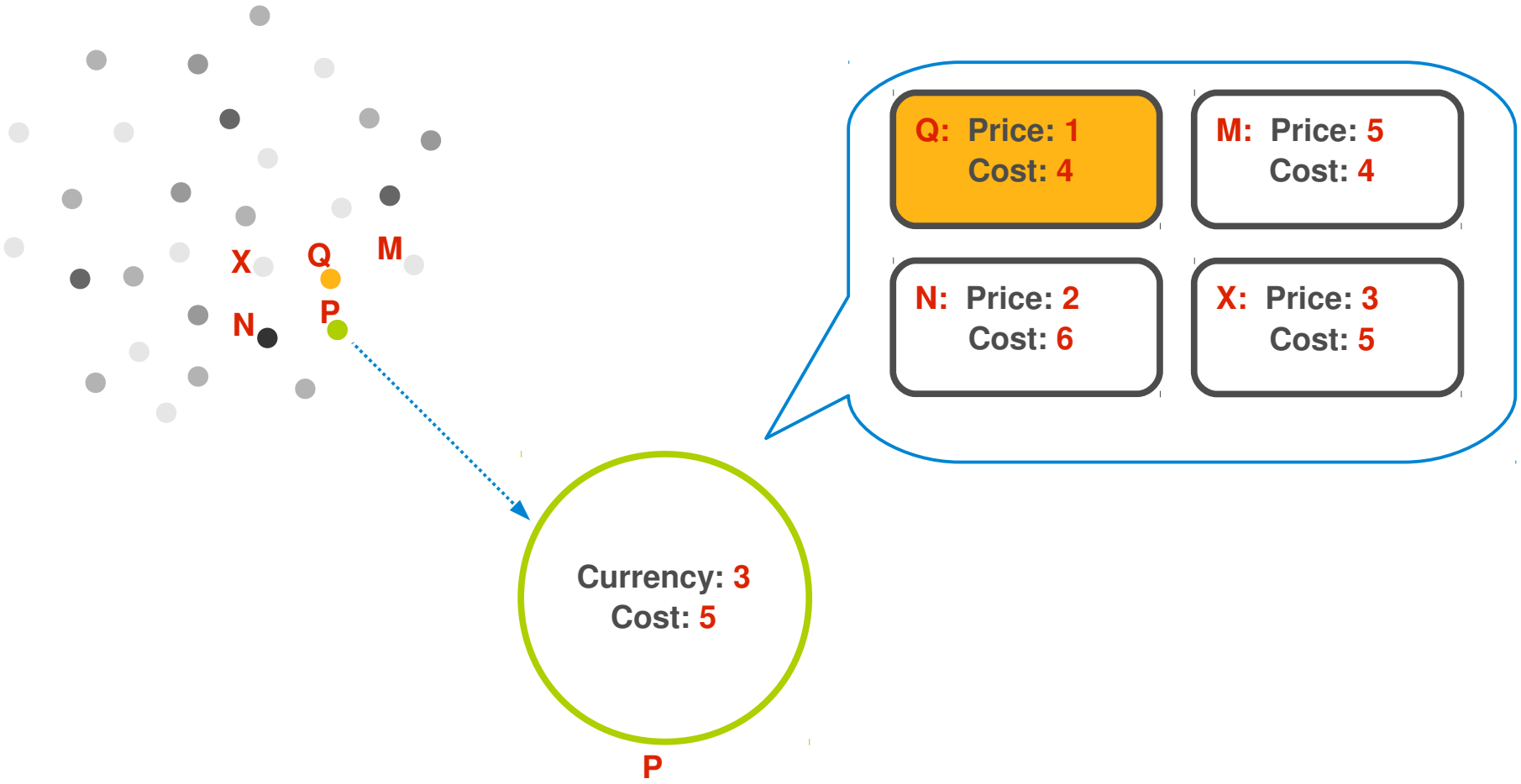
- Our market model is based on minimizing costs through nodes iteratively bidding for upload slots.

- The depth of a node in each tree is inversely proportional to its currency.

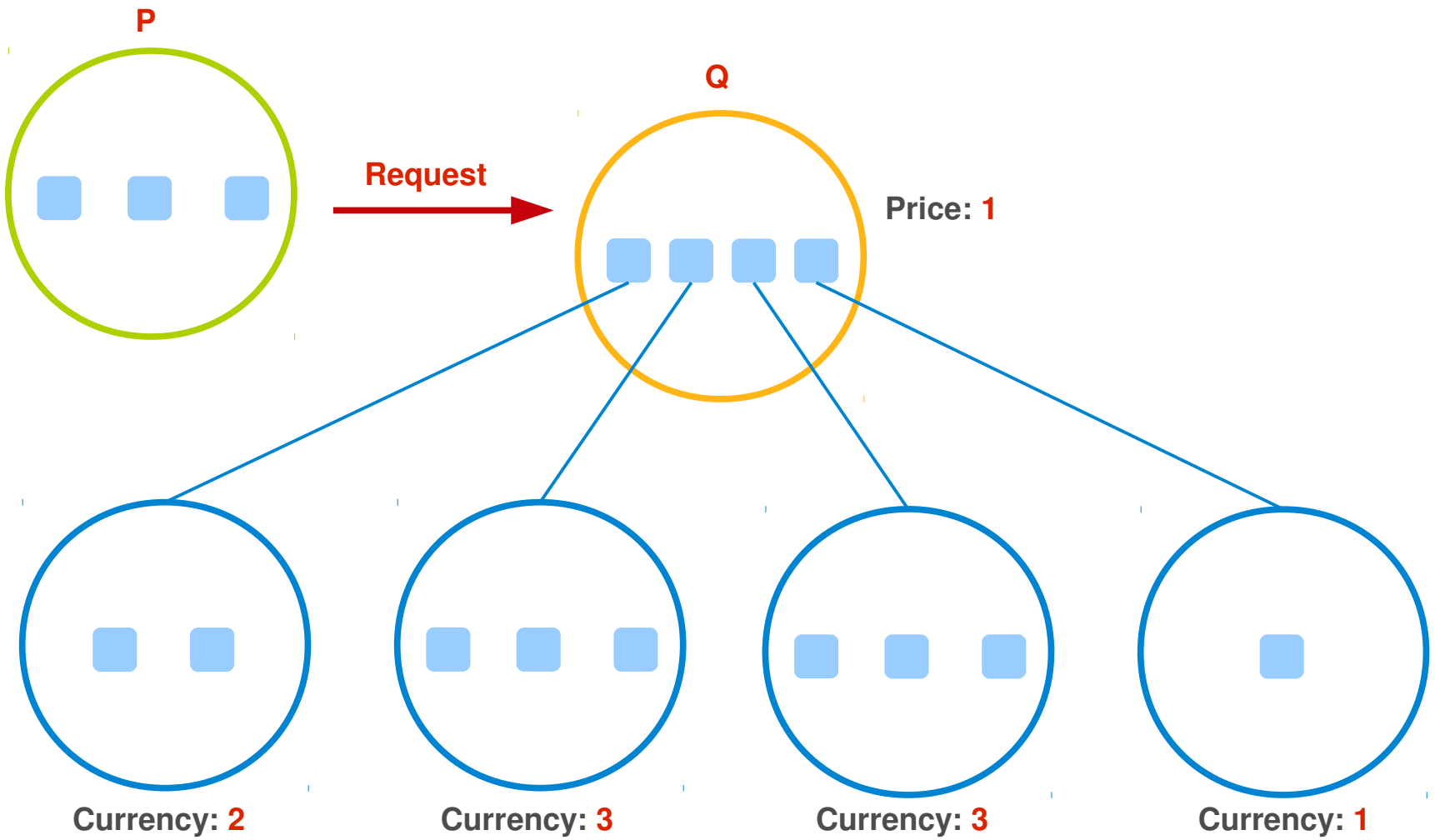
The Market Model – Child Side



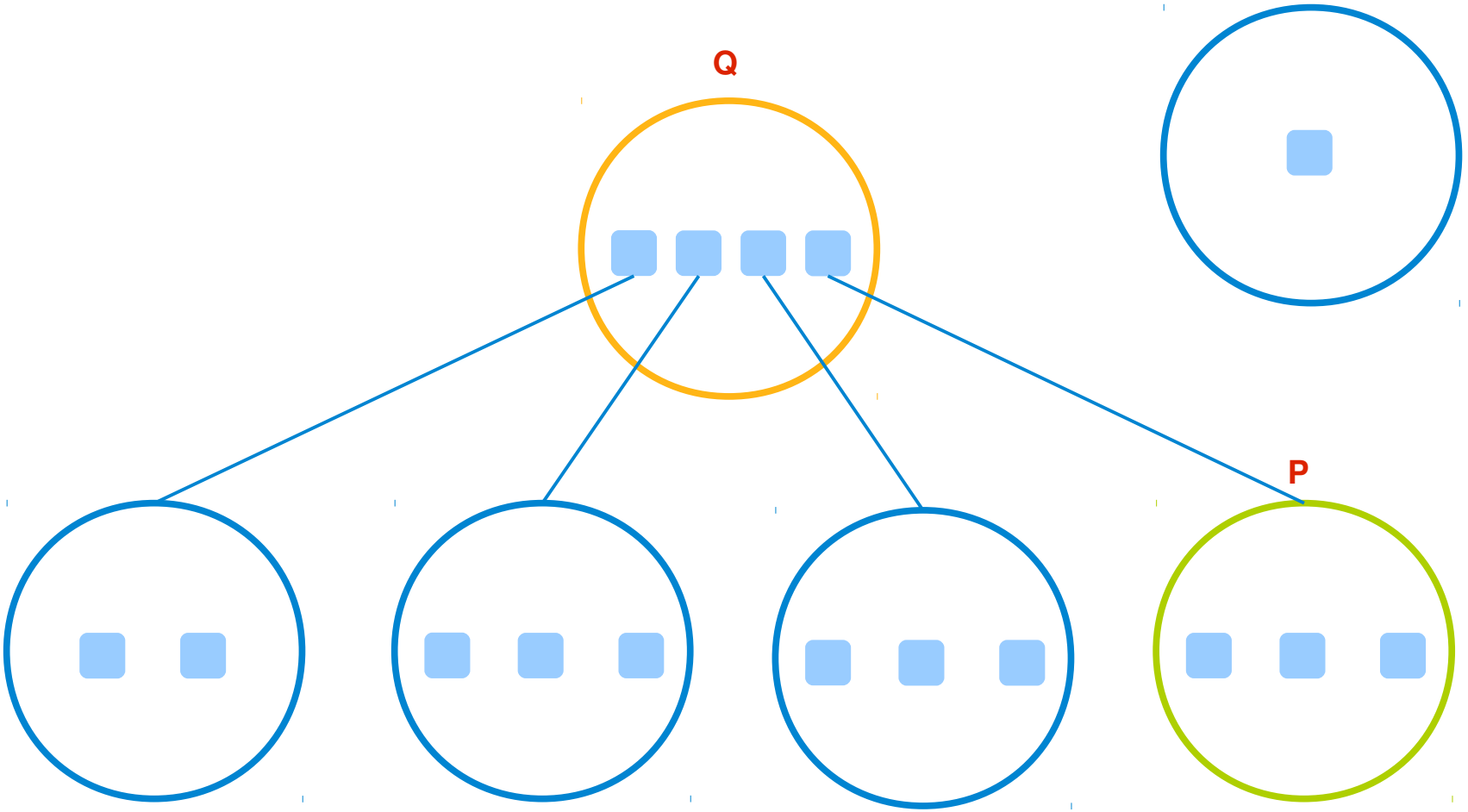
The Market Model – Child Side



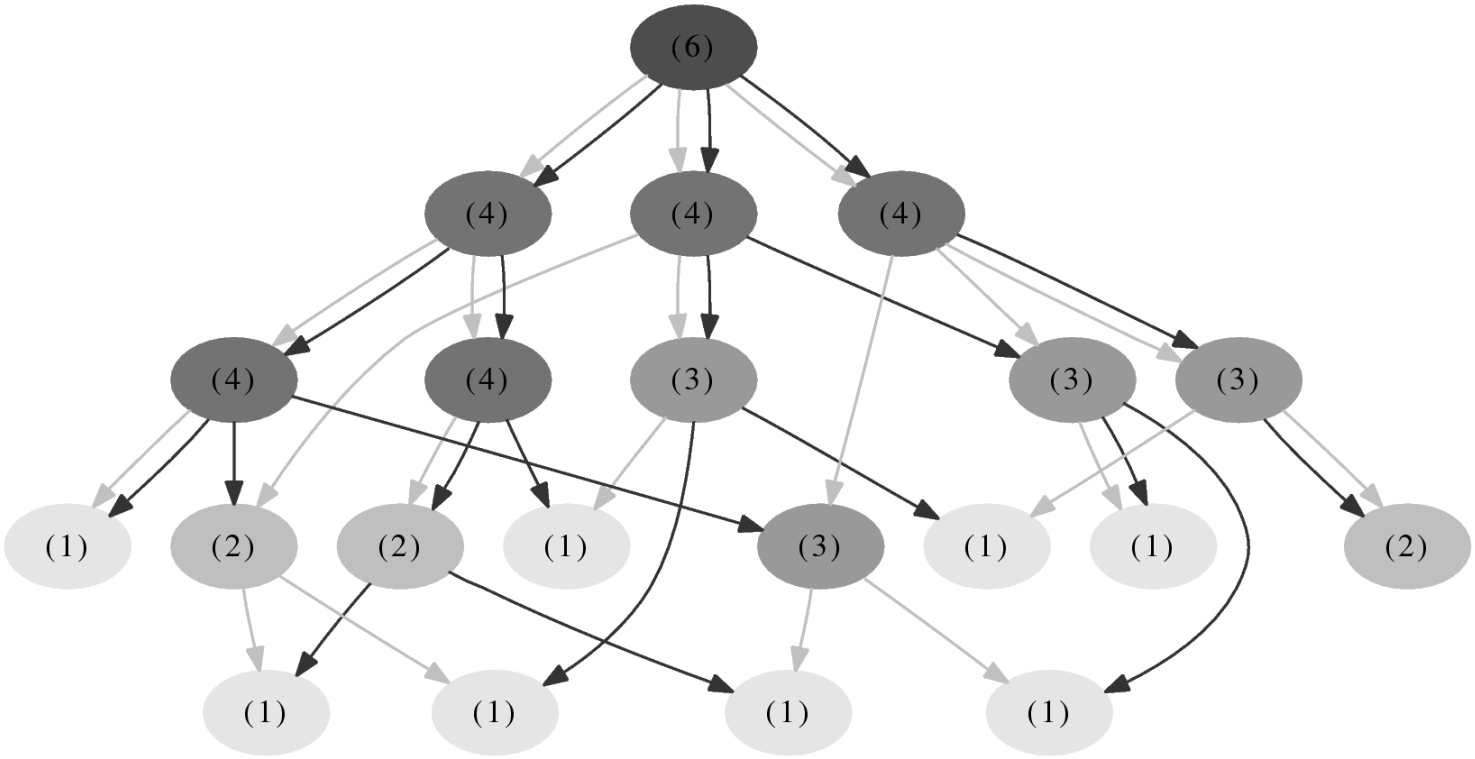
The Market Model – Parent Side



The Market Model – Parent Side



Constructed Streaming Overlay

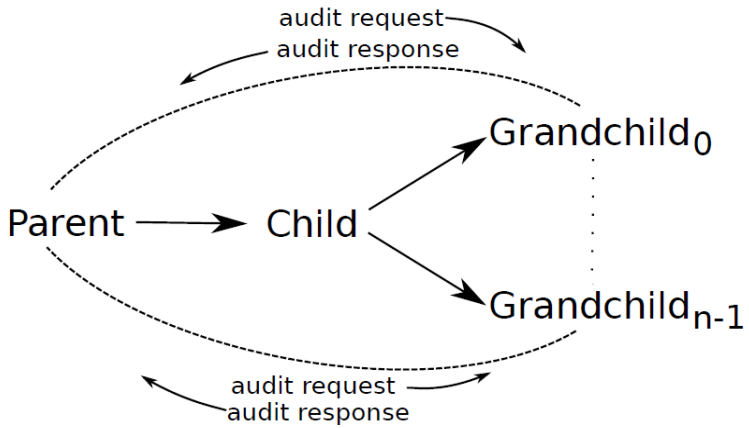


- Constructed 2-tree overlay.
- Darker nodes have more upload capacity than lighter ones.

Freeriders

Freerider Detector

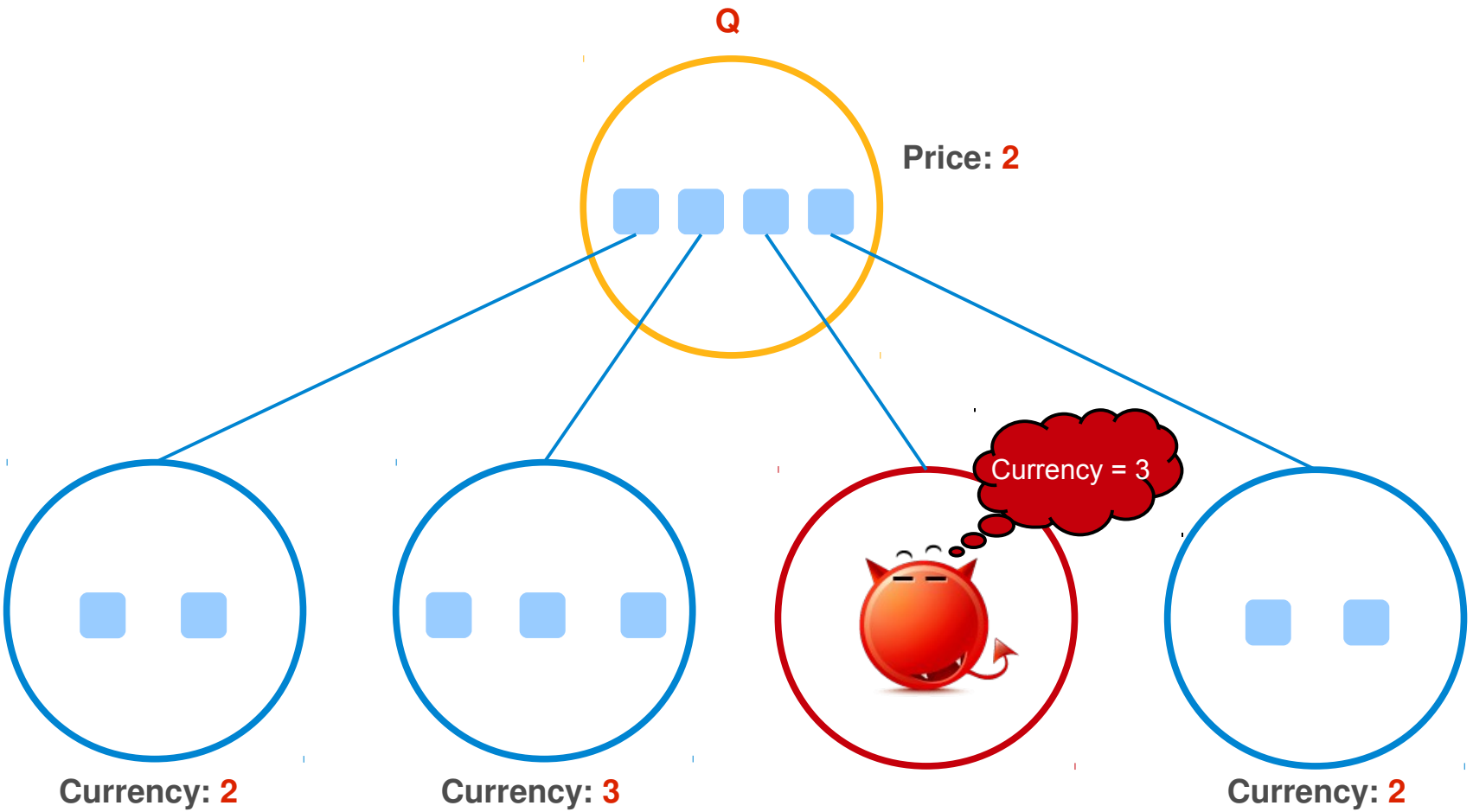
- Freeriders are nodes that supply less upload bandwidth than claimed.
- Nodes identify freeriders through transitive auditing using their children's children.



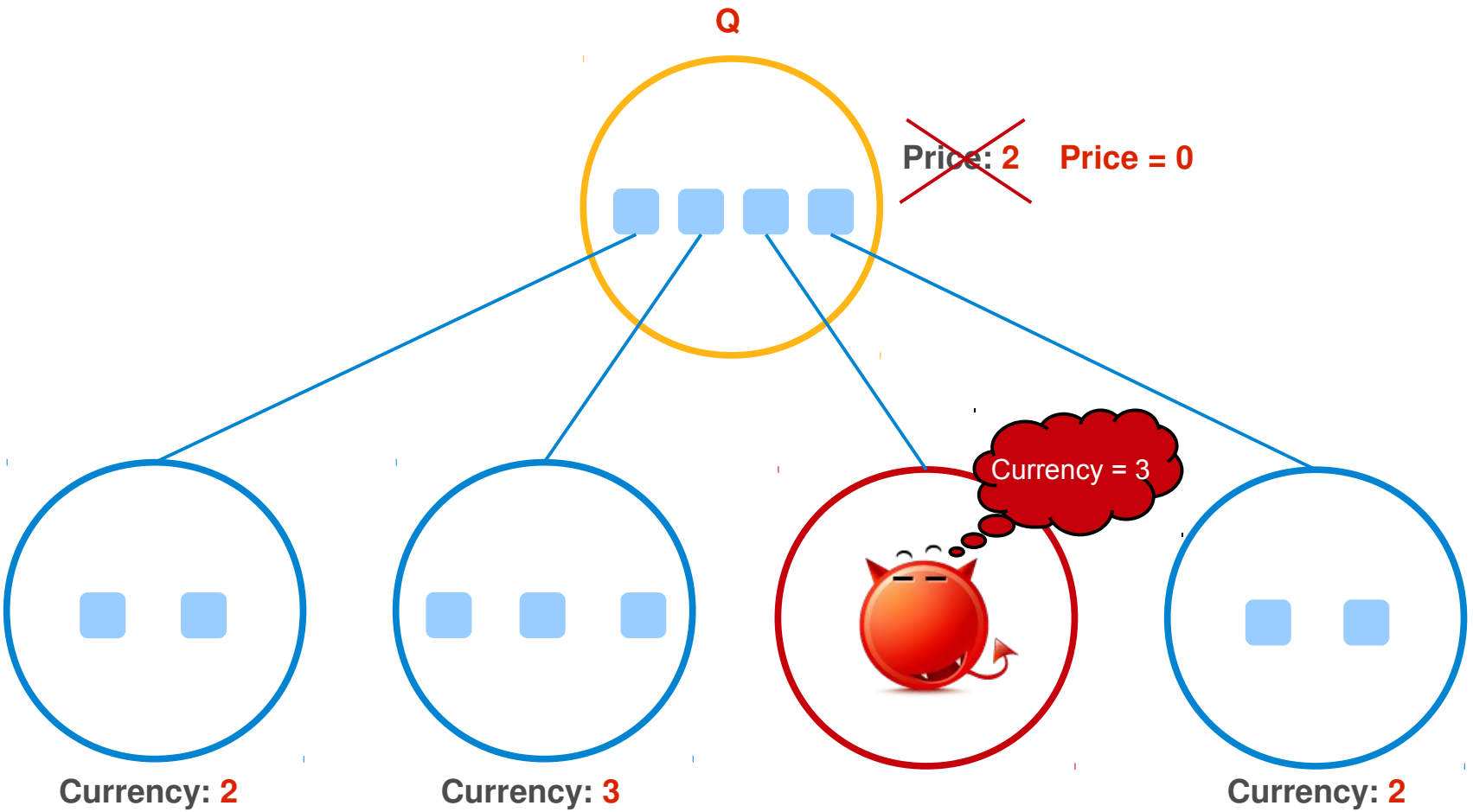
Detecting Freeriders

- 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.

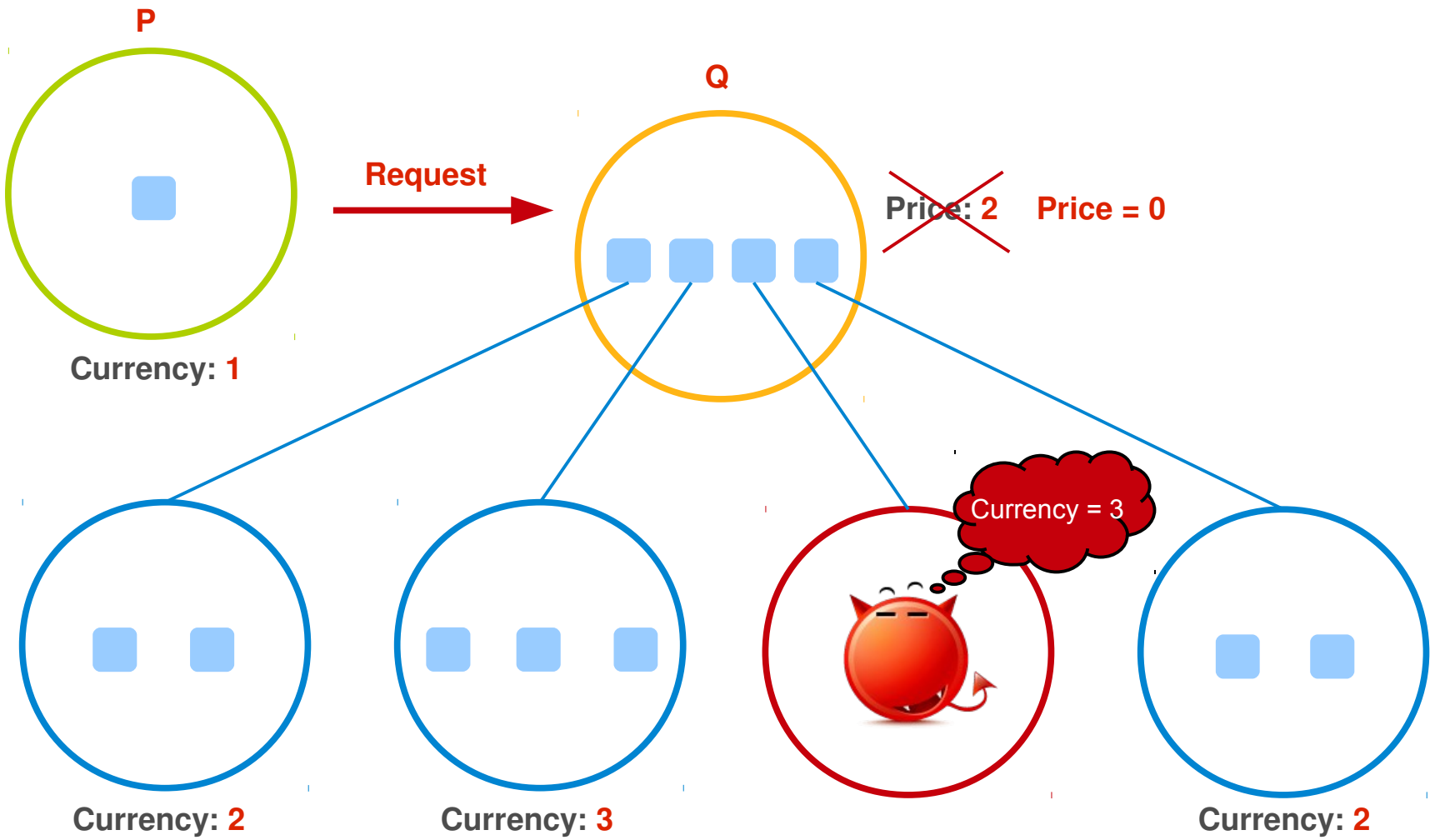
Freerider – Punishment



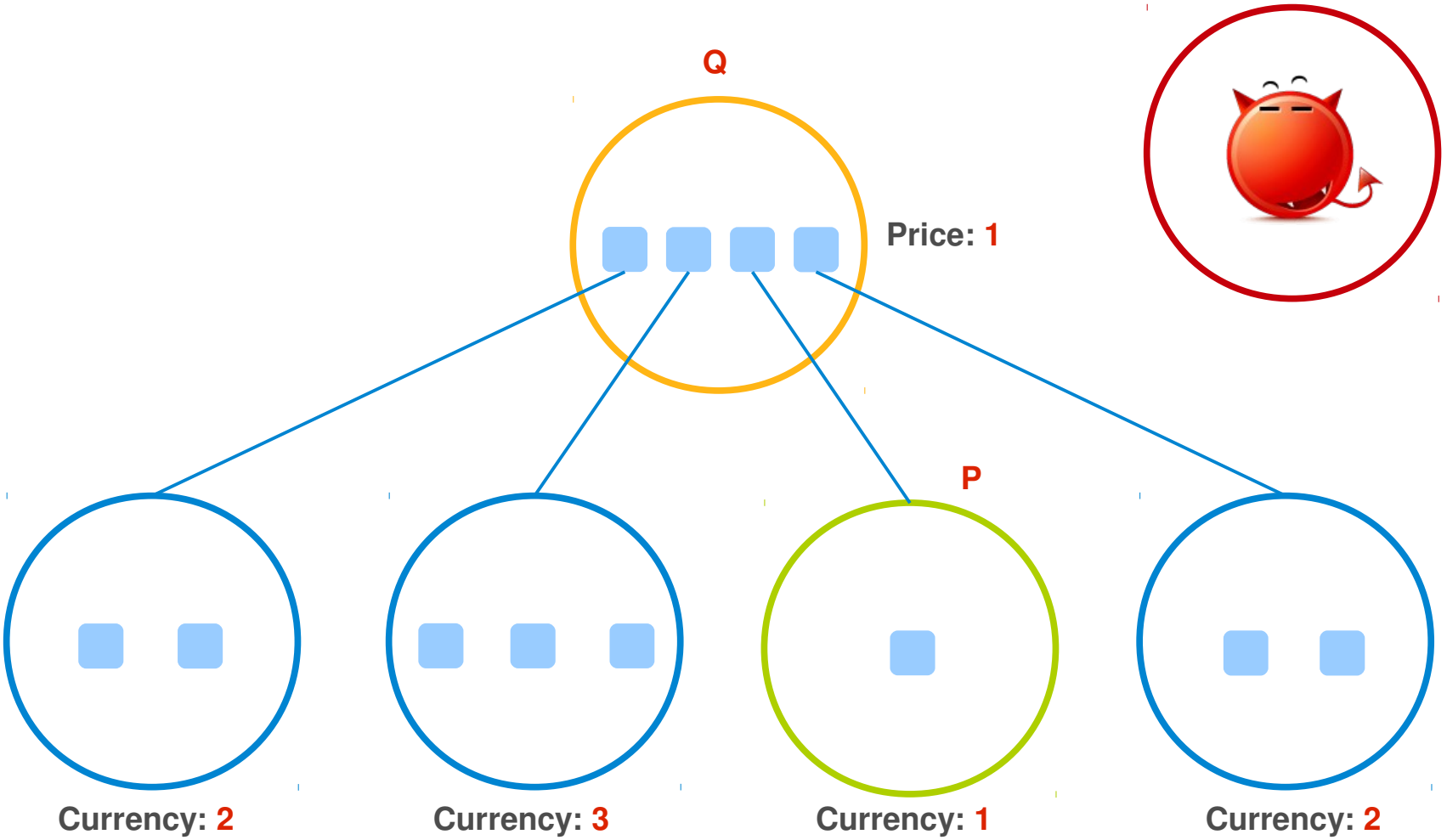
Freerider – Punishment



Freerider – Punishment



The Market Model – Parent Side

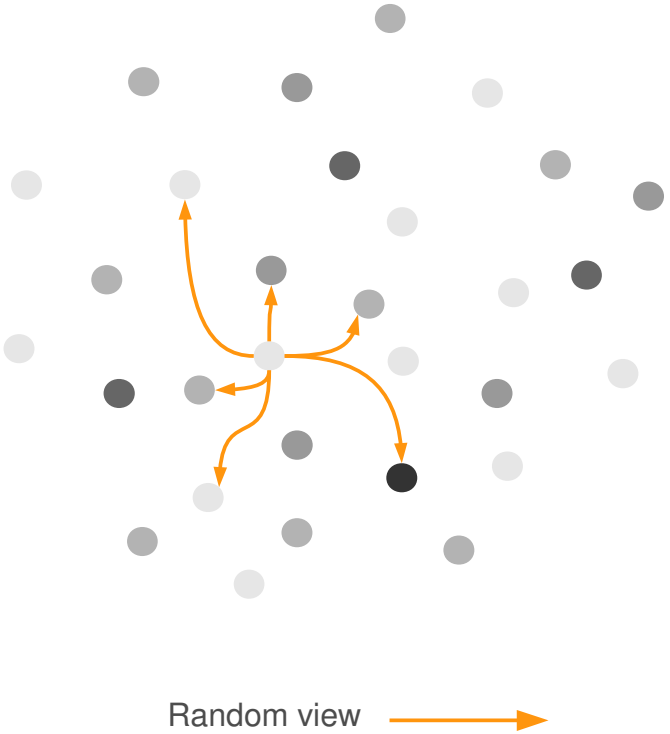


Optimization

Node Discovery

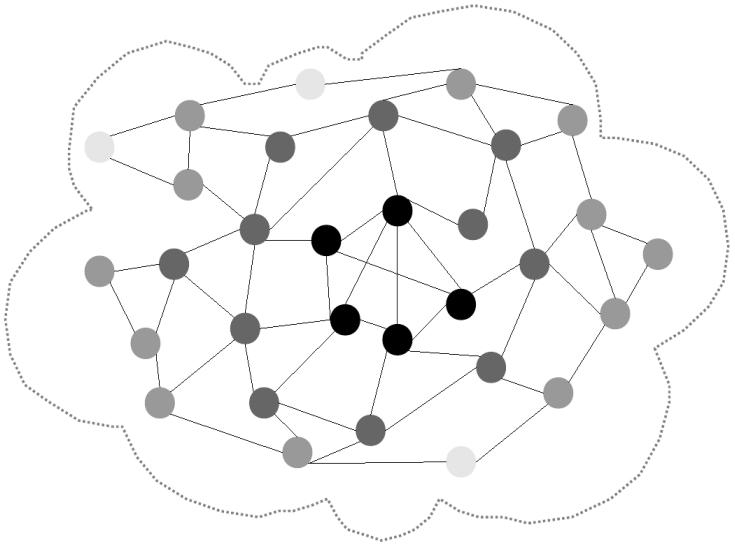
- Naïve solution: nodes in partial views are selected randomly from all the nodes.

- Optimization: nodes use the Gradient overlay to construct and maintain their partial view of the system.



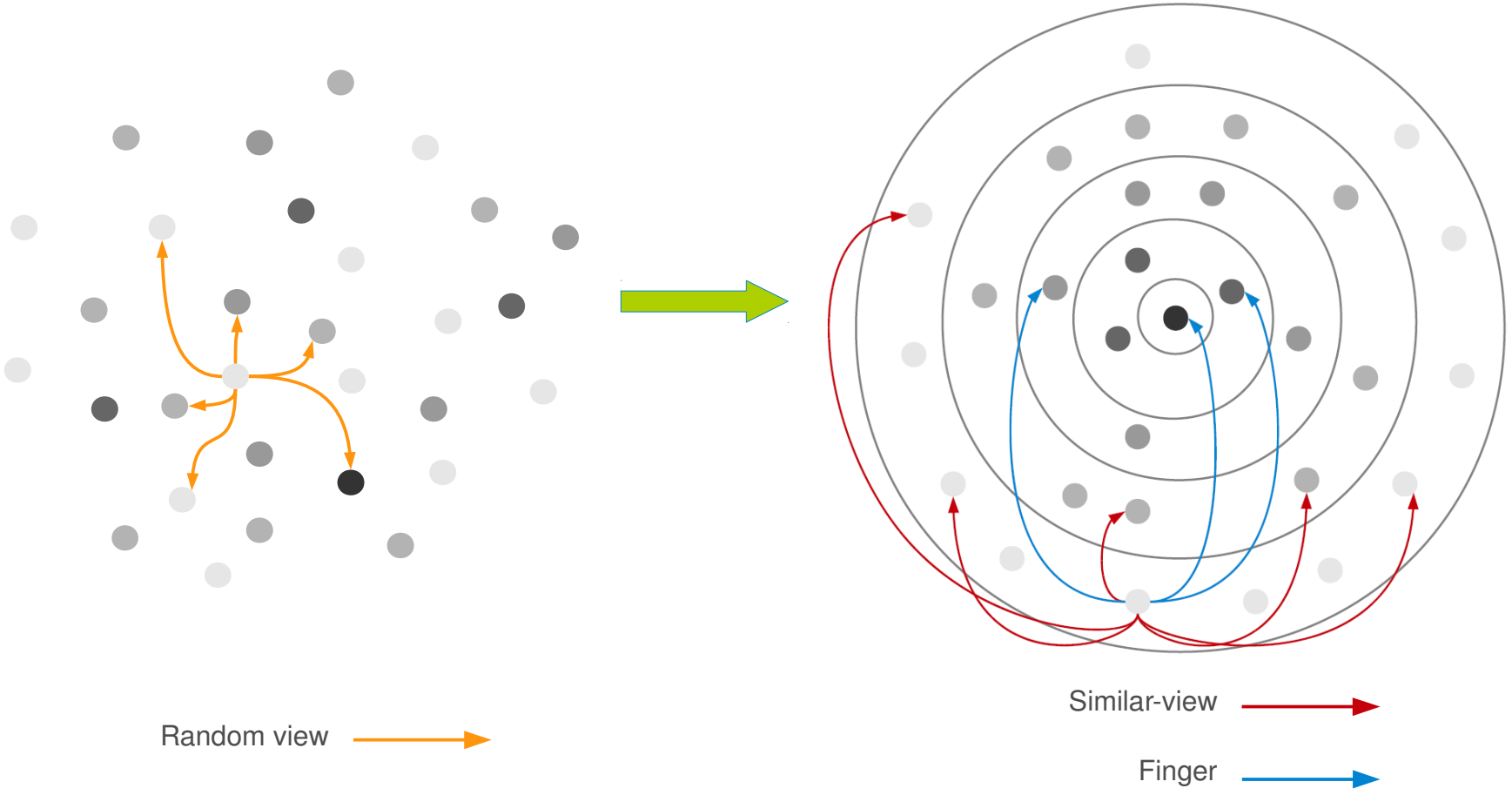
The Gradient Overlay

- 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.



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 the First Solutions

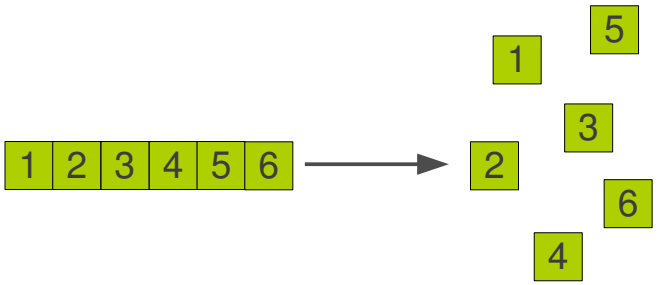
- Tree structure
- Fragile in massive failures

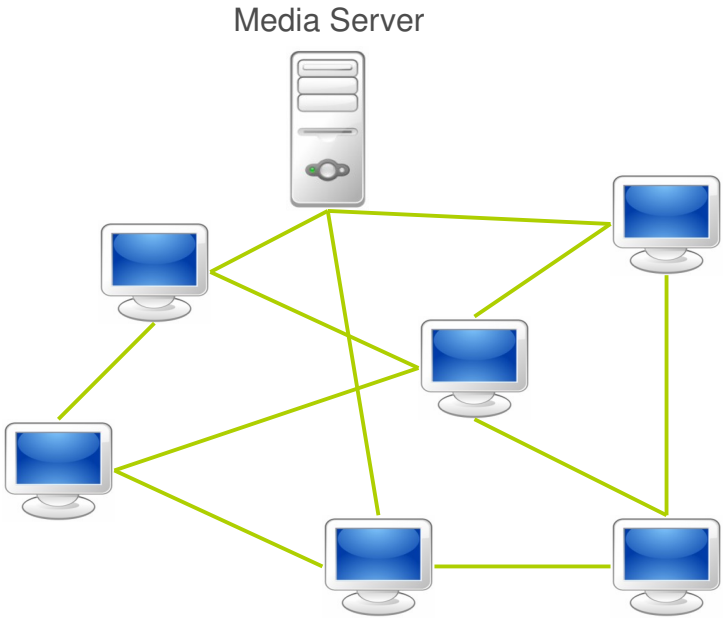
Design Space

- 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 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.



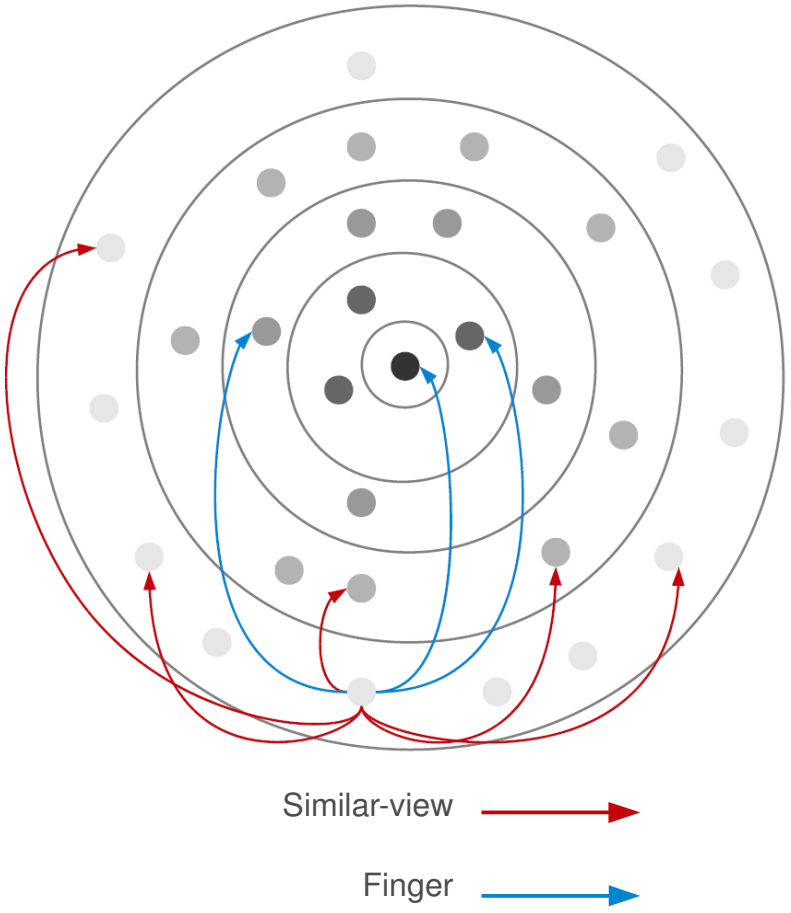


The Market Model - Node Properties

- 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 Relation

- The same as Sepidar.



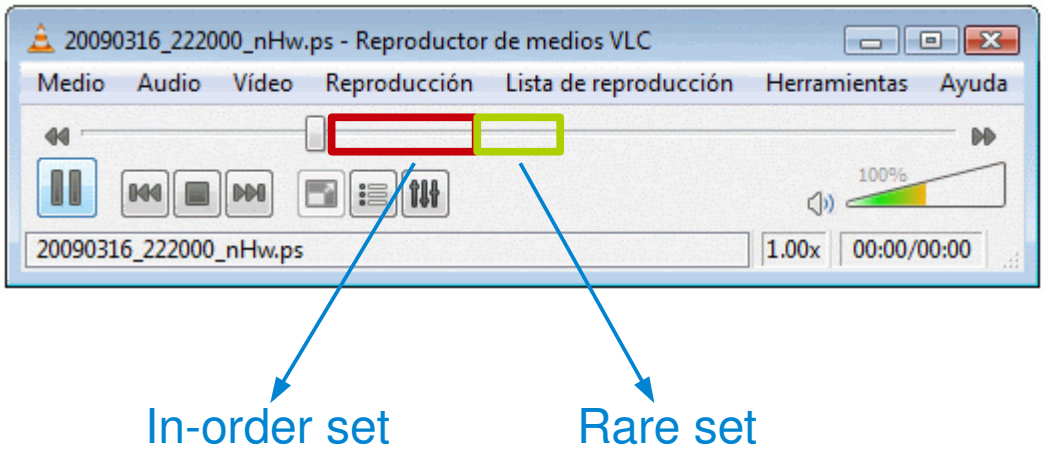
Data Dissemination (1/2)

- 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.

Data Dissemination (2/2)

- Sliding window



Freerider Detection (1/2)

- Each child assigns a score to each of its parents, for a time window covering the last 10 seconds.
  - Increments on receiving non-duplicate blocks from its parent in the last 10 seconds.
- 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.

Experiments

Experiment Setup

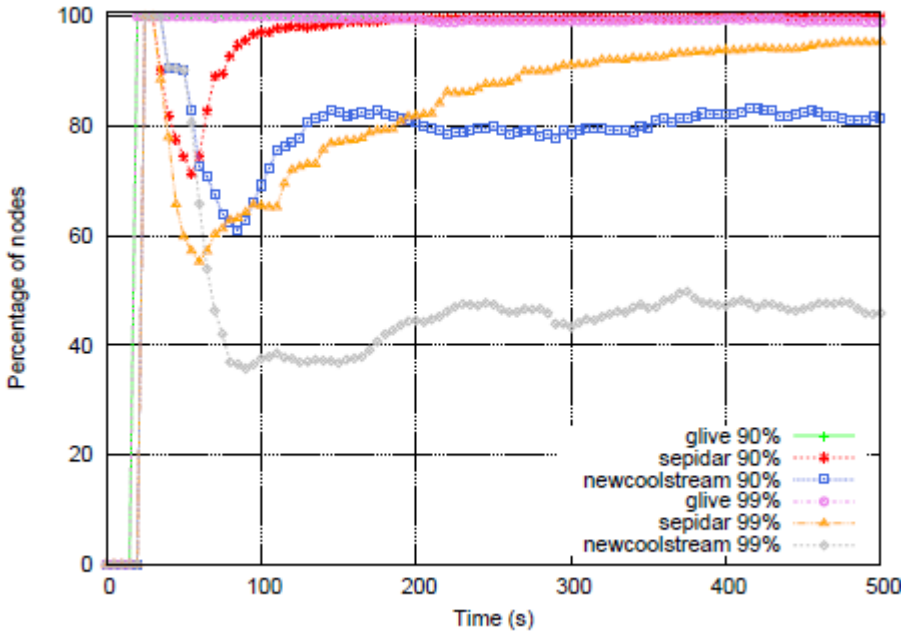
- Using the Kompics as a simulator platform.
- King dataset is used to model the latencies between nodes.
- The streaming rate to 512 Kbps, and it is split into 8 stripes (in sepidar). The stream/stripe is divided into a sequence of 16 Kb blocks.
- Nodes start playing the media after buffering it for 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.

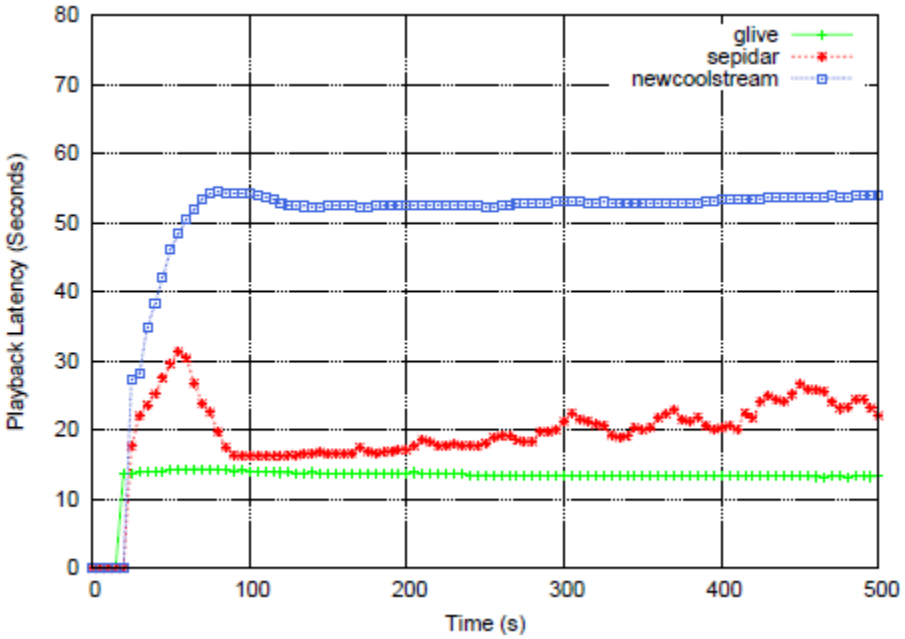
Metrics

- Playback continuity: the percentage of the block received before their playback time.

- Playback latency: the difference between the playback point of a node and the playback point at the media source.

Playback Continuity and Playback Latency






zzzzzzz

My Contributions

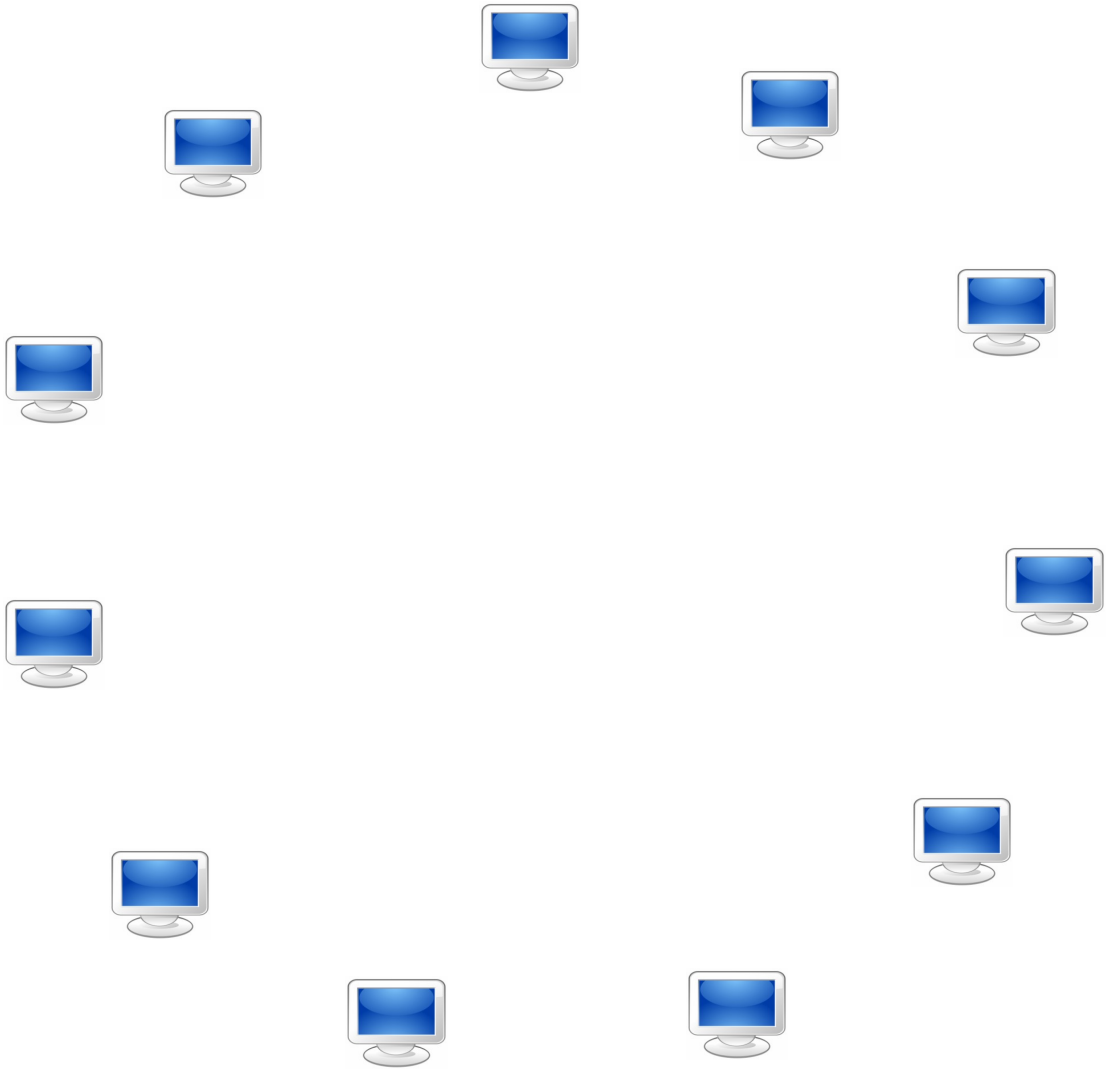
- A distributed market model to construct P2P streaming overlays.
- A NAT-friendly gossip-based peer sampling service.

Problem Description

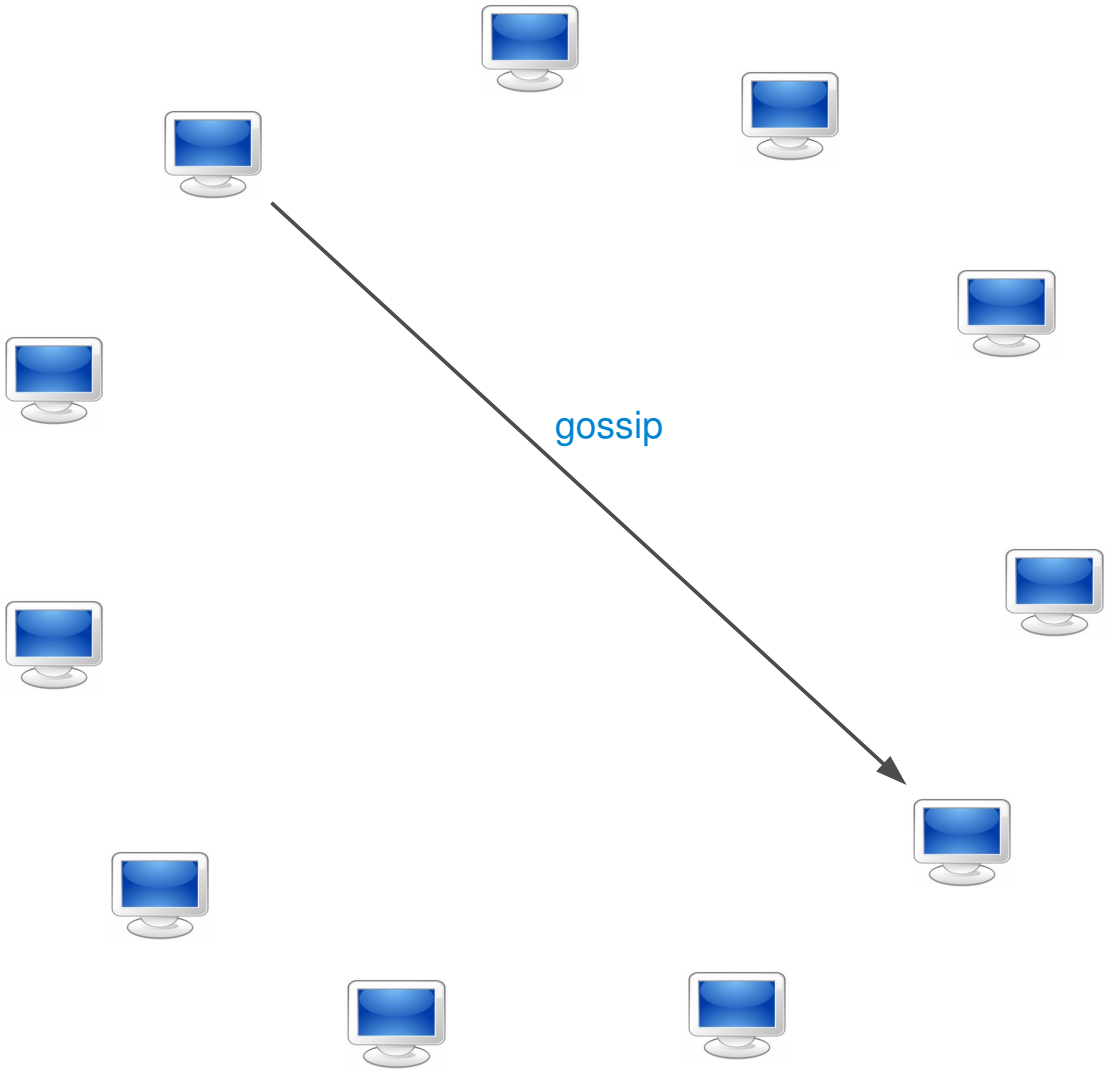
Gossip-based Peer Sampling Service

- It provides a node with a uniform random sample of live nodes from all nodes in the system (partial view).

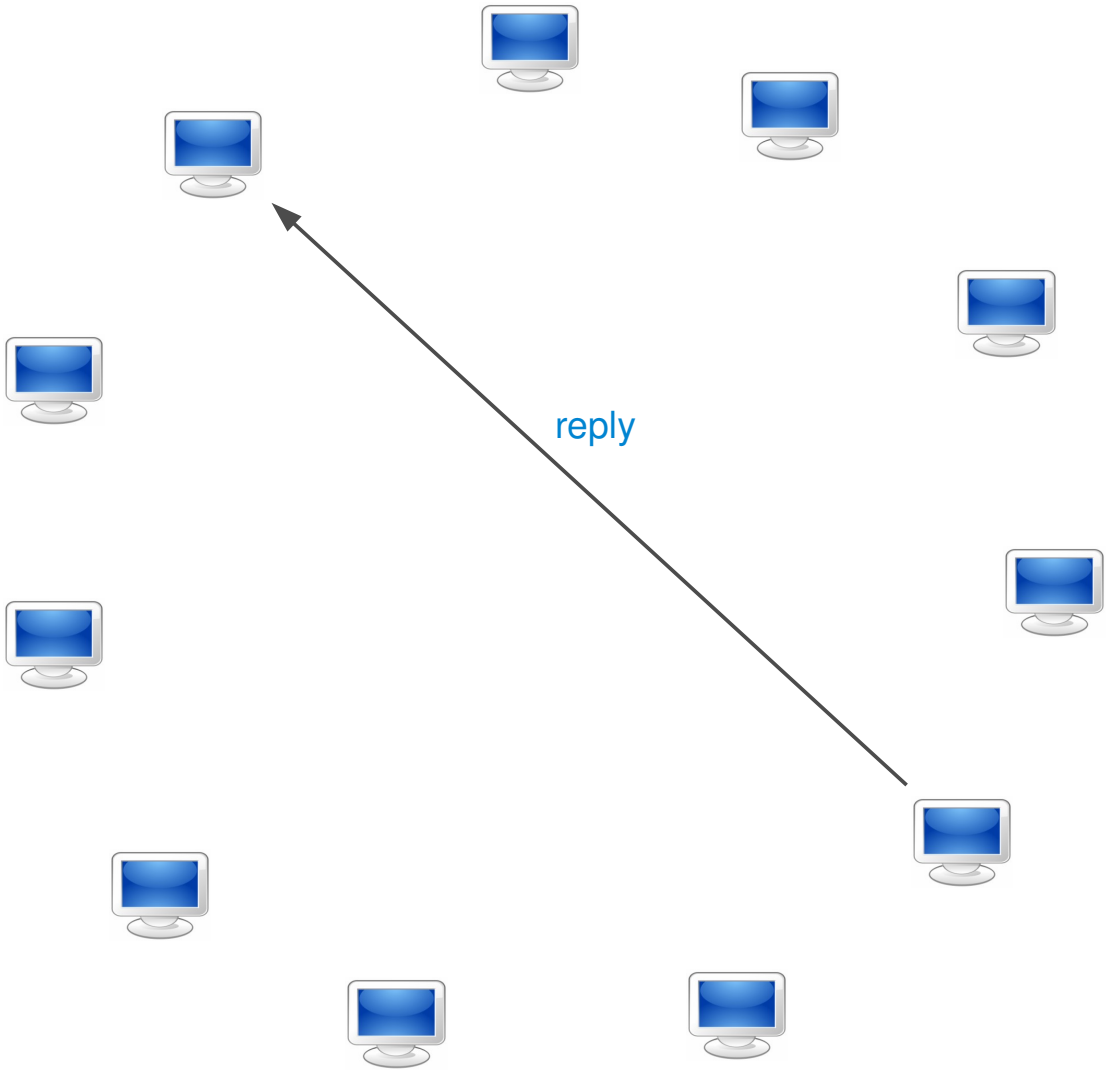
Gossip Protocol (1/4)



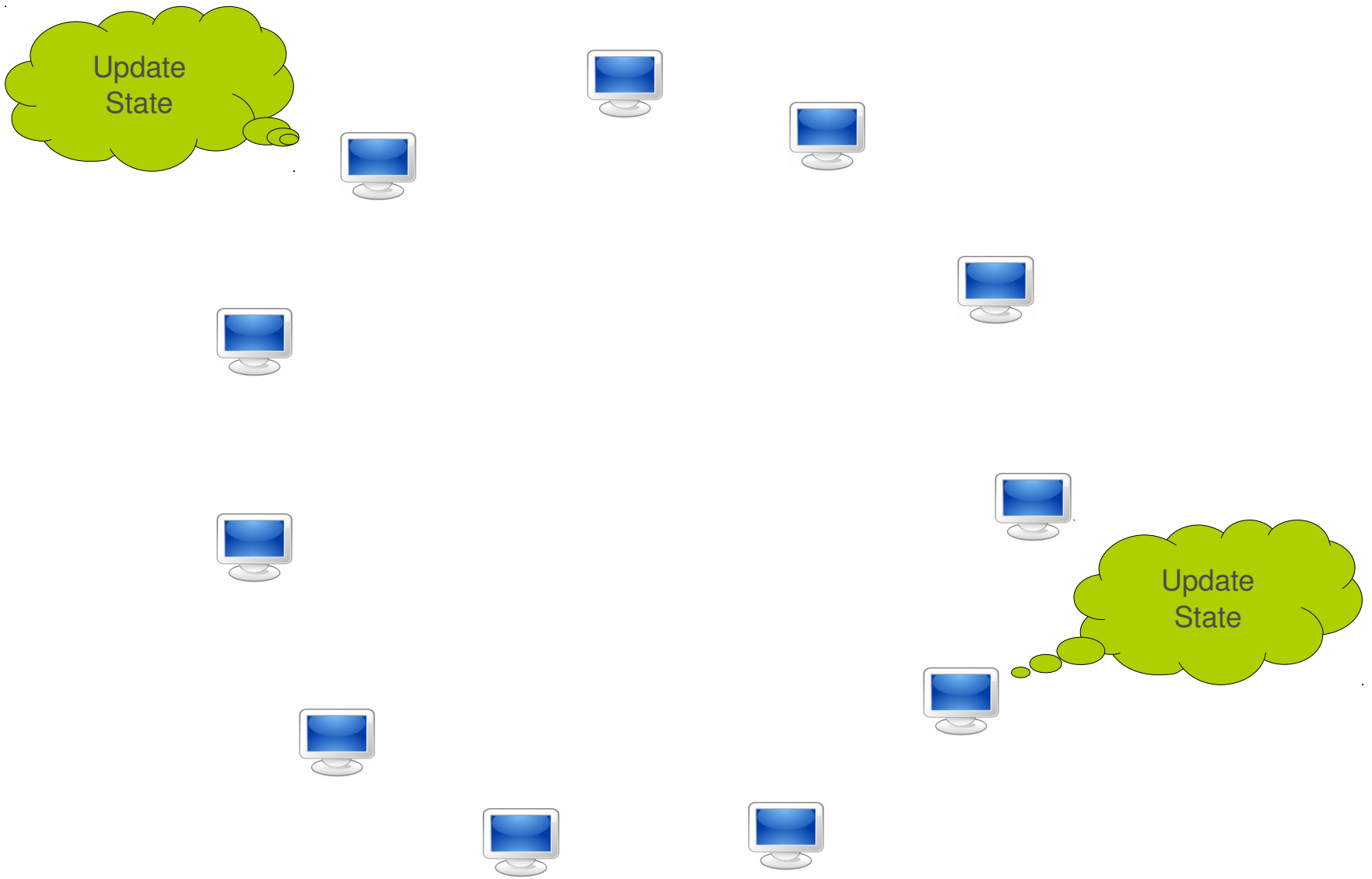
Gossip Protocol (2/4)



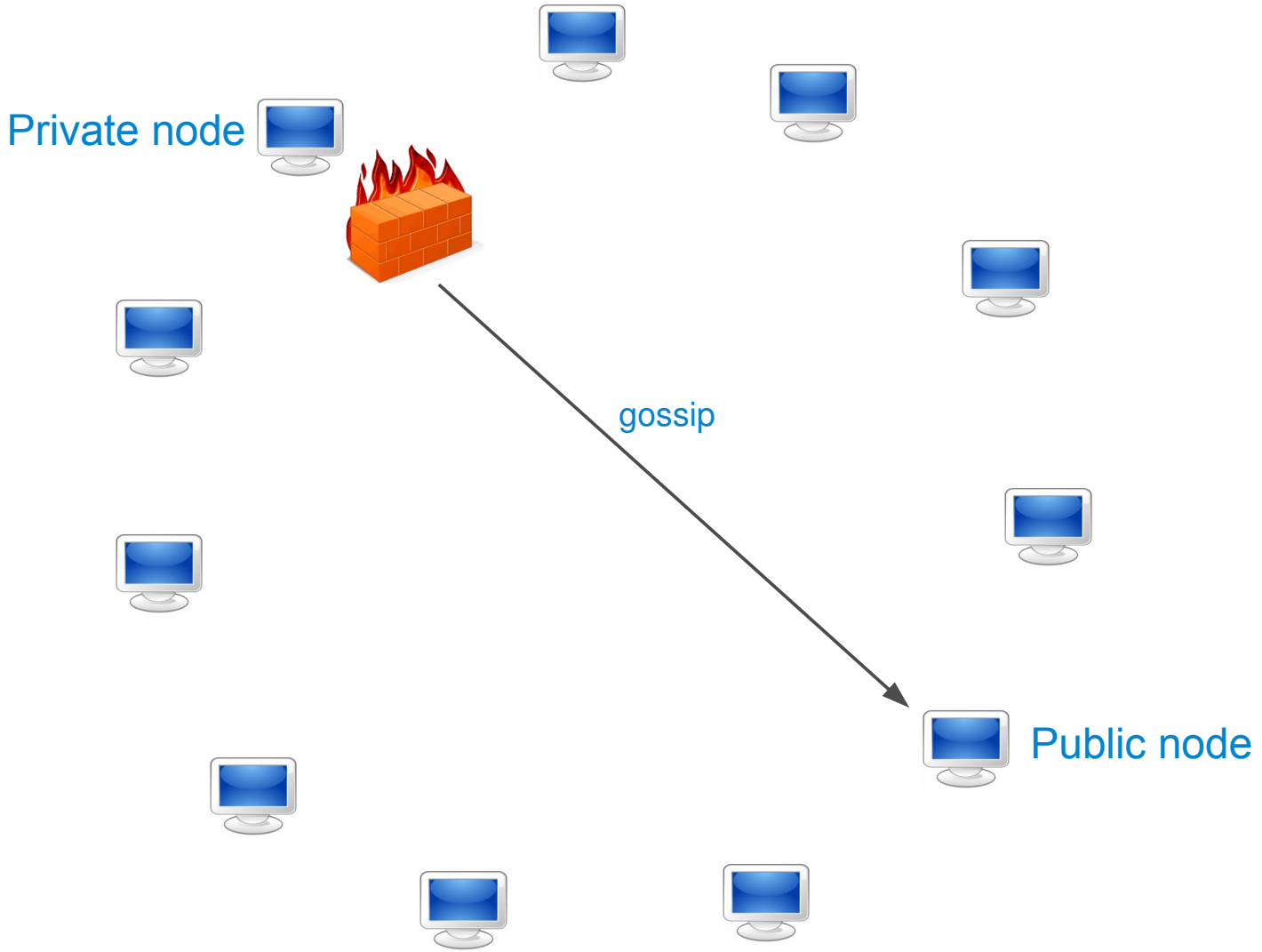
Gossip Protocol (3/4)



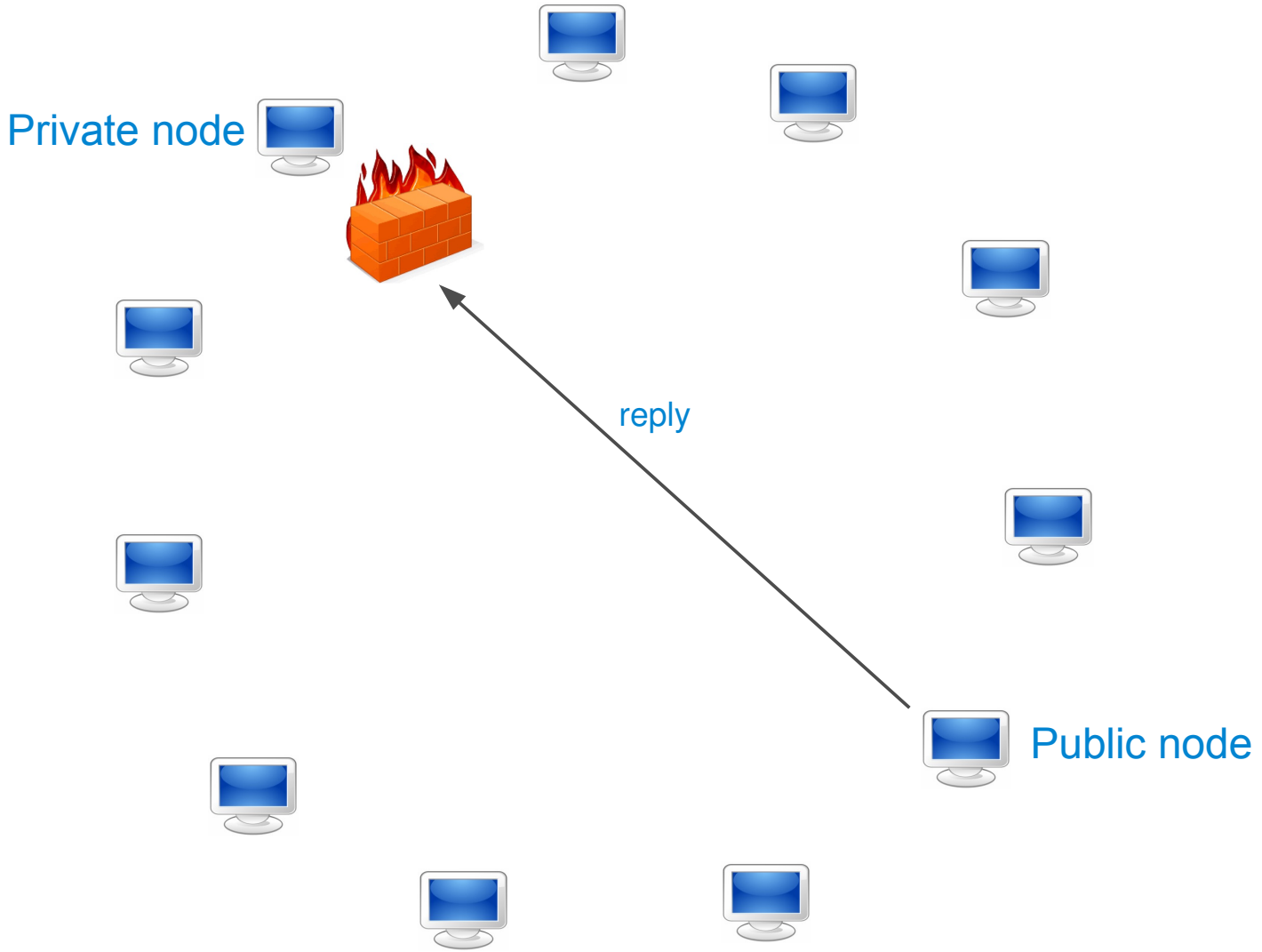
Gossip Protocol (4/4)



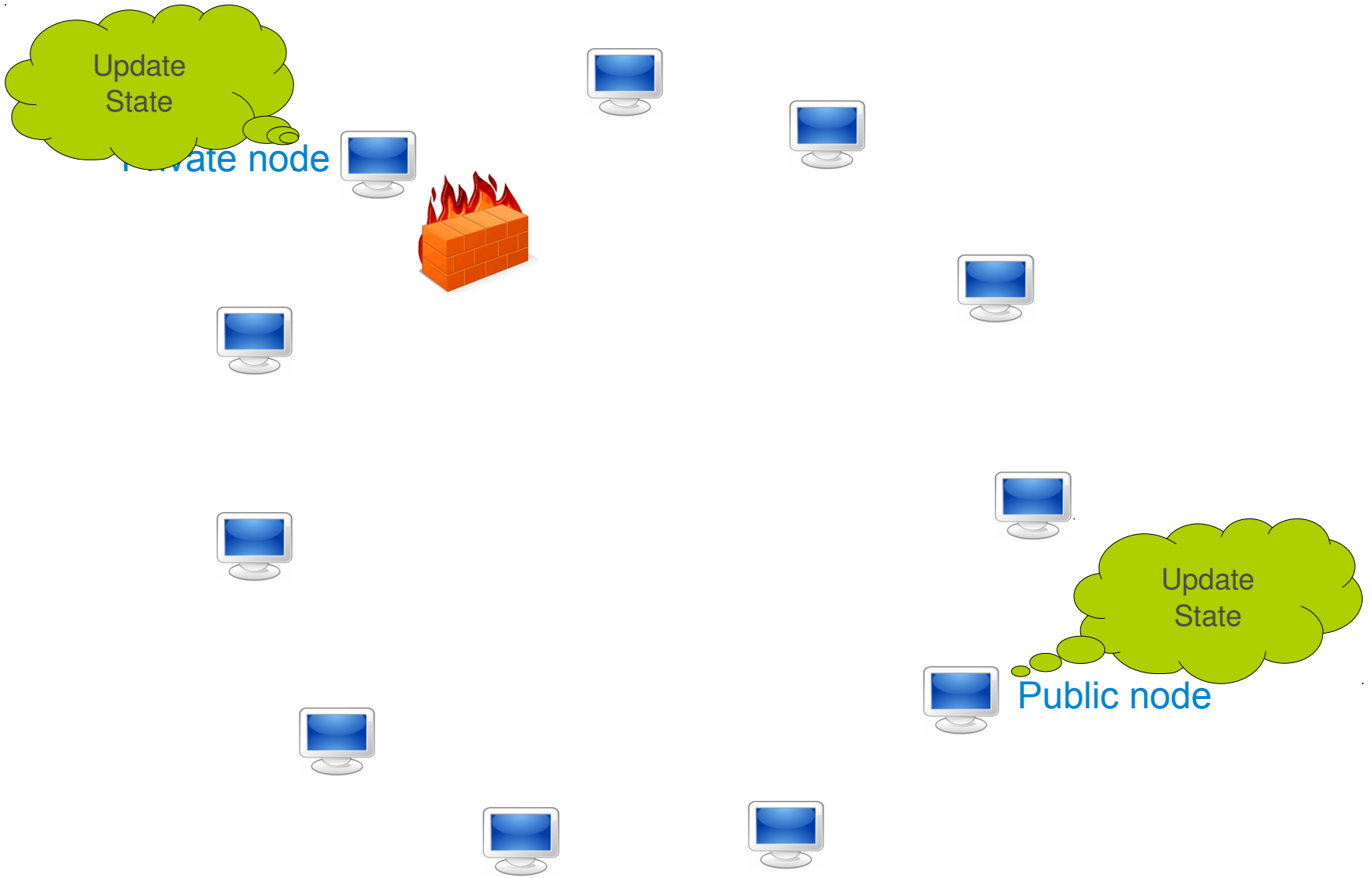
Natted Gossip Protocol (1/4)



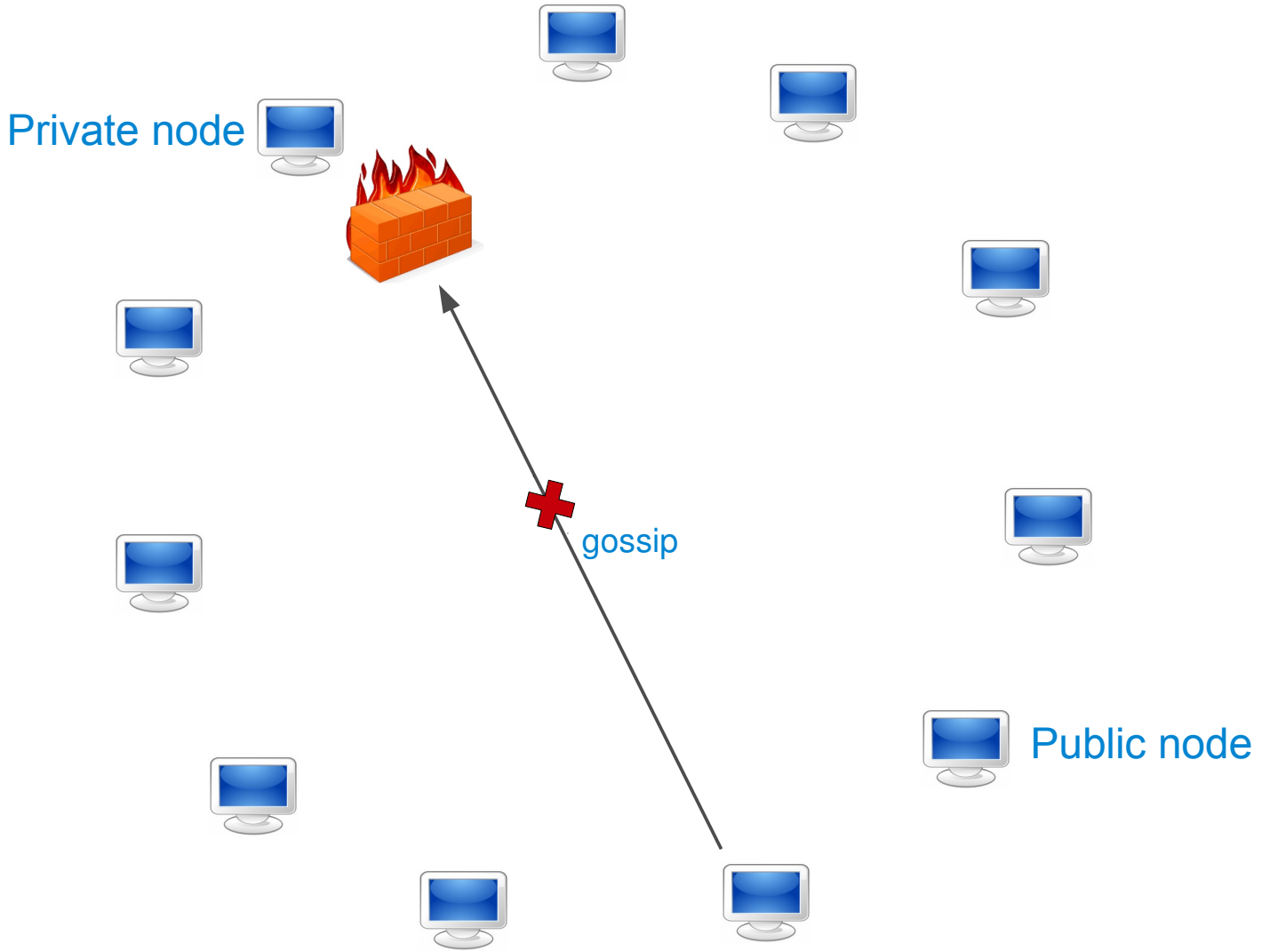
Natted Gossip Protocol (2/4)



Natted Gossip Protocol (3/4)

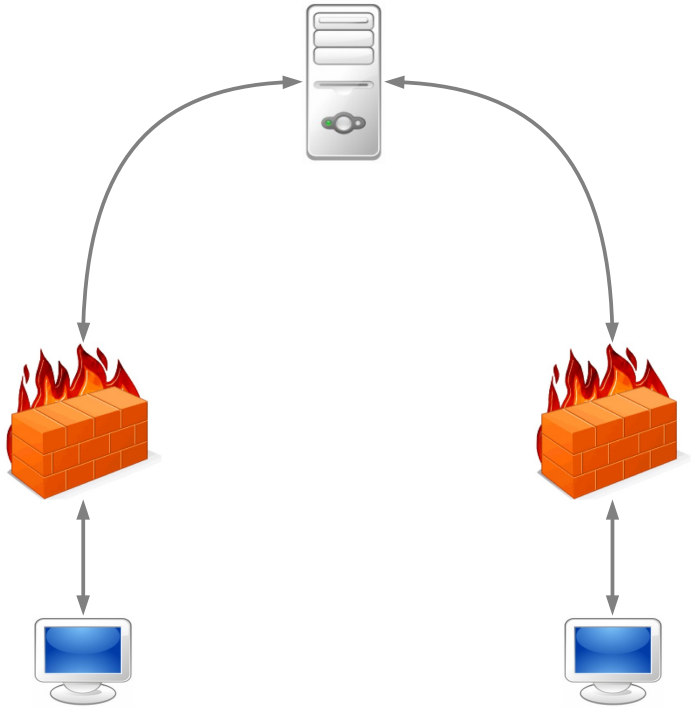


Natted Gossip Protocol (4/4)



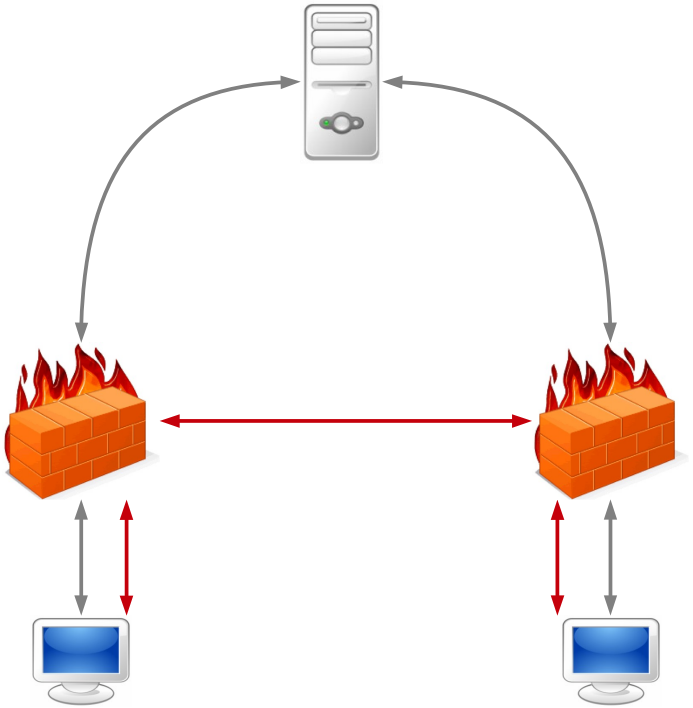
The Solutions to Communicate with a Private Node (1/3)

- Relay communications to the private node using a public relay node.



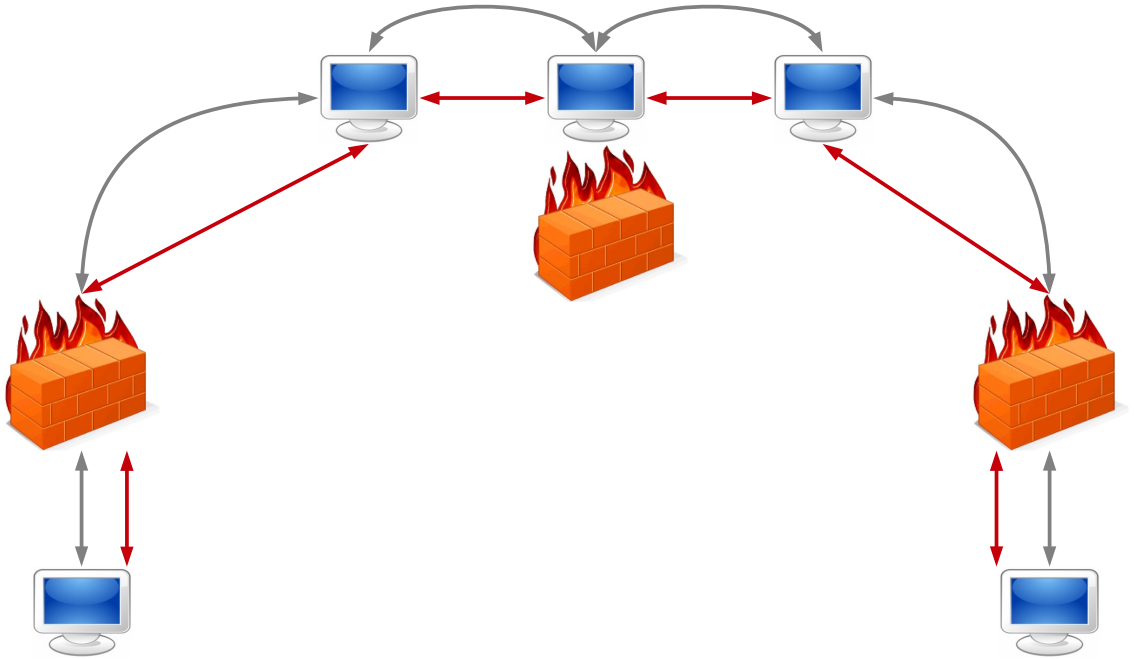
The Solutions to Communicate with a Private Node (2/3)

- Use a NAT hole-punching algorithm to establish a direct connection to the private node using a public rendezvous node.



The solutions to Communicate with a Private Node (3/3)

- Route the request to the private node using chains of existing open connections.



Main Questions

- Discovering which public nodes act as partners for the private nodes?
- How much data will be sent over the connection?
- How fairly should the gossiping load be distributed over public versus private nodes?

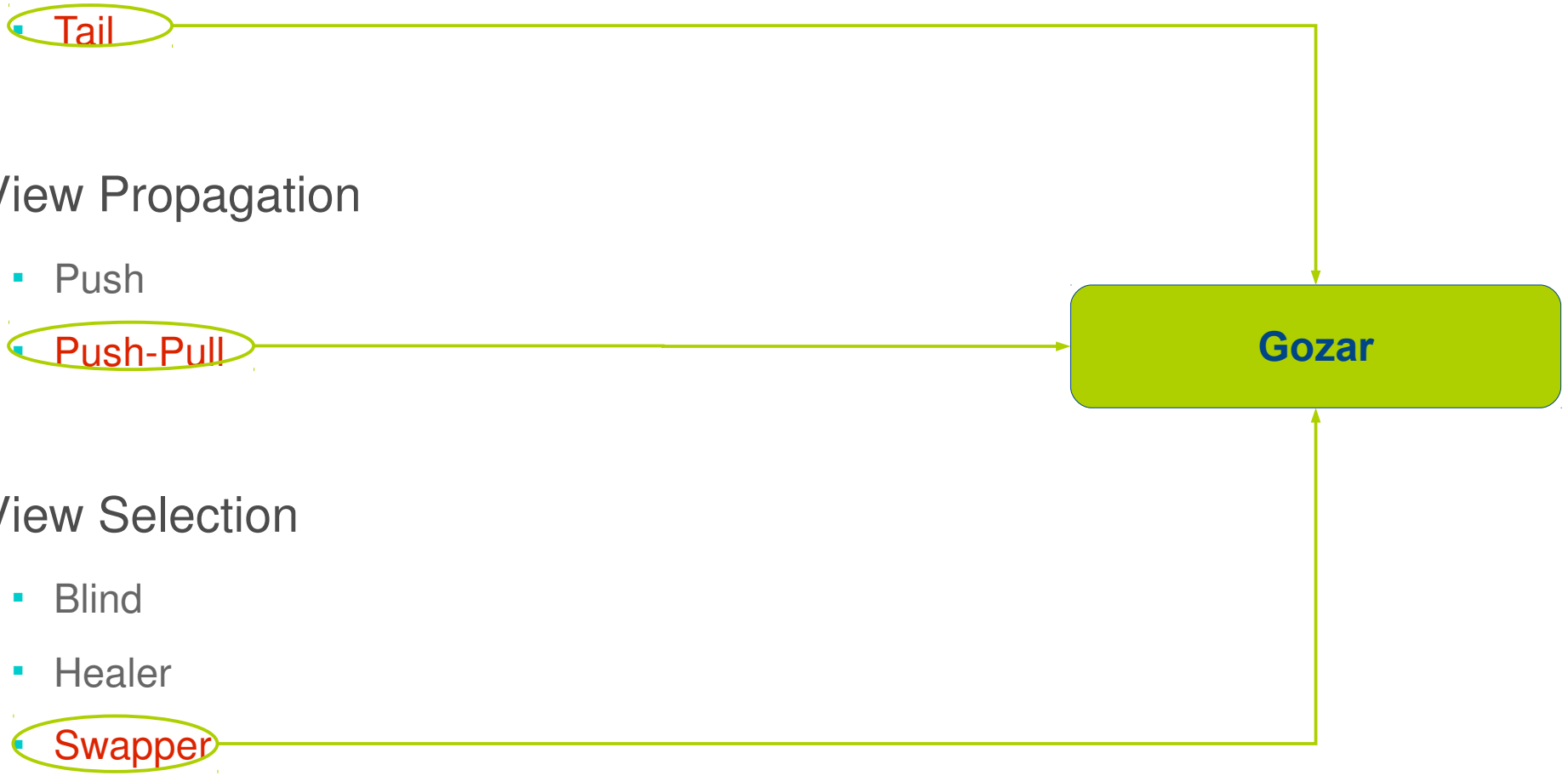
Gozar – NAT friendly Peer Sampling Service

Design Space

- Peer Selection
  - Rand
  - Tail
- View Propagation
  - Push
  - Push-Pull
- View Selection
  - Blind
  - Healer
  - Swapper

Design Space

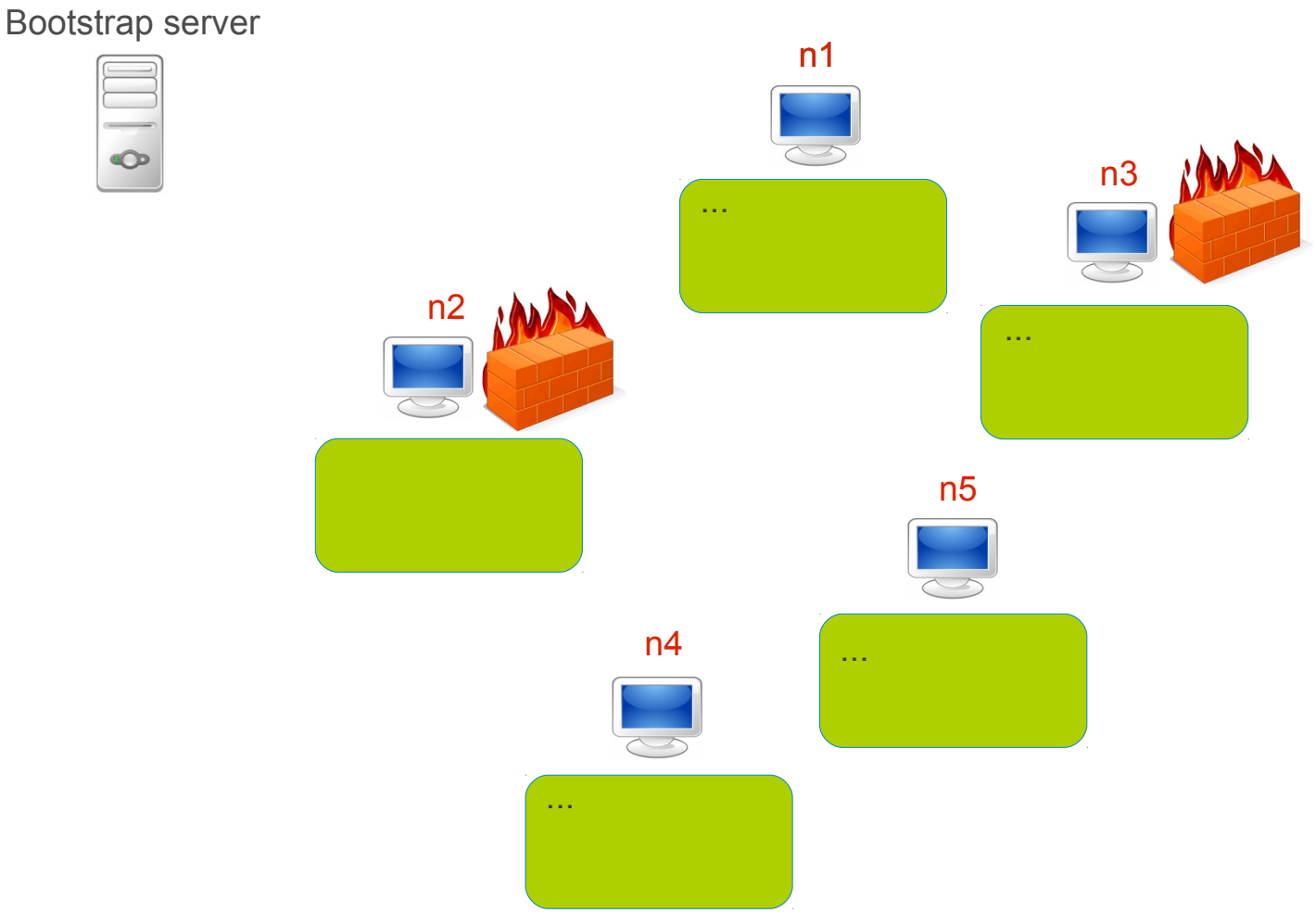
- Peer Selection
  - Rand
  - Tail
- View Propagation
  - Push
  - Push-Pull
- View Selection
  - Blind
  - Healer
  - Swapper



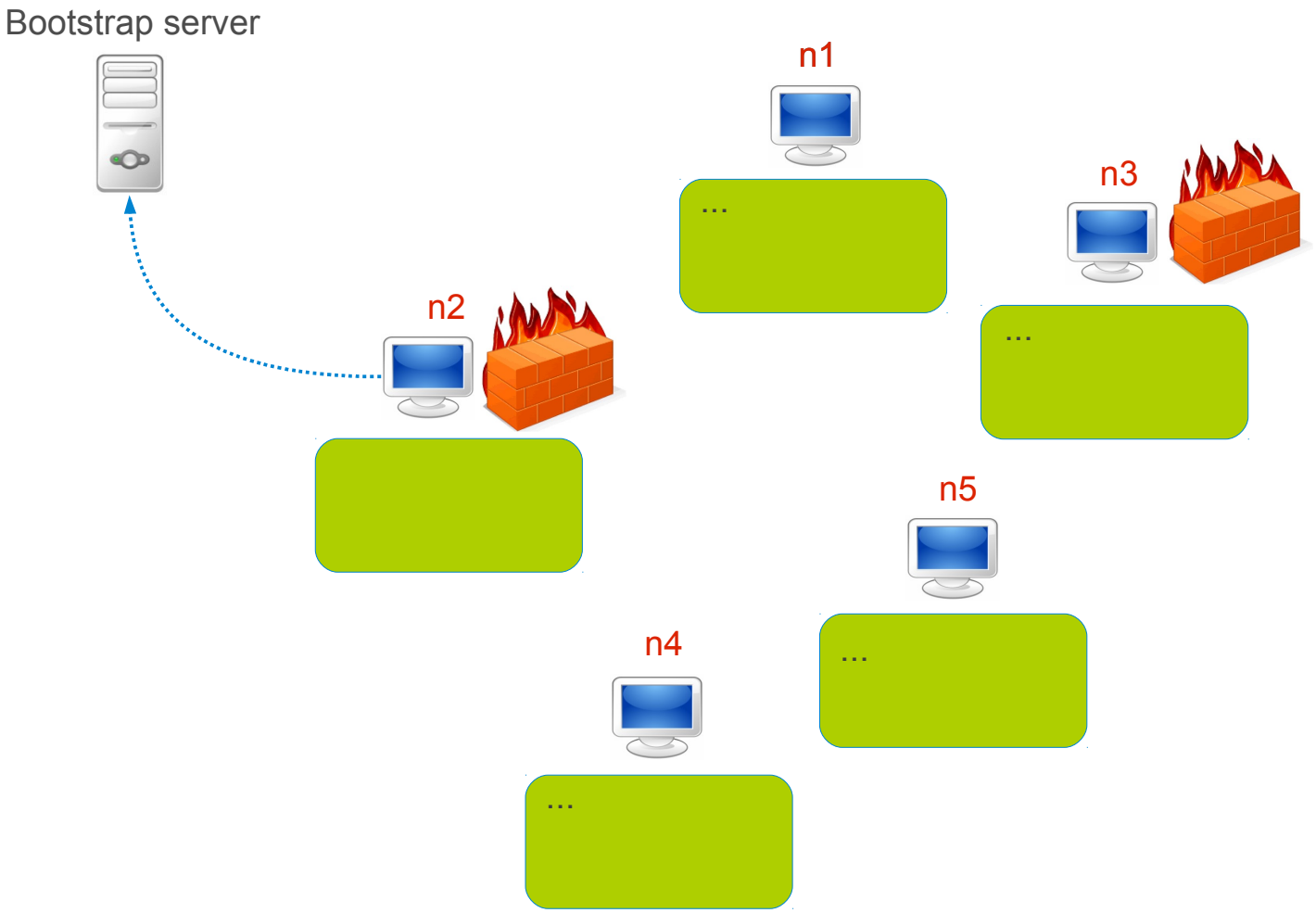
The Main Idea of Gozar

- In Gozar, each private node connects to one or more public nodes, called partners.
- A node spreads its its address, as well as its partners' addresses while gossiping with other nodes.
- A node can communicate with a private node using one of its partners as a relay or rendezvous node.

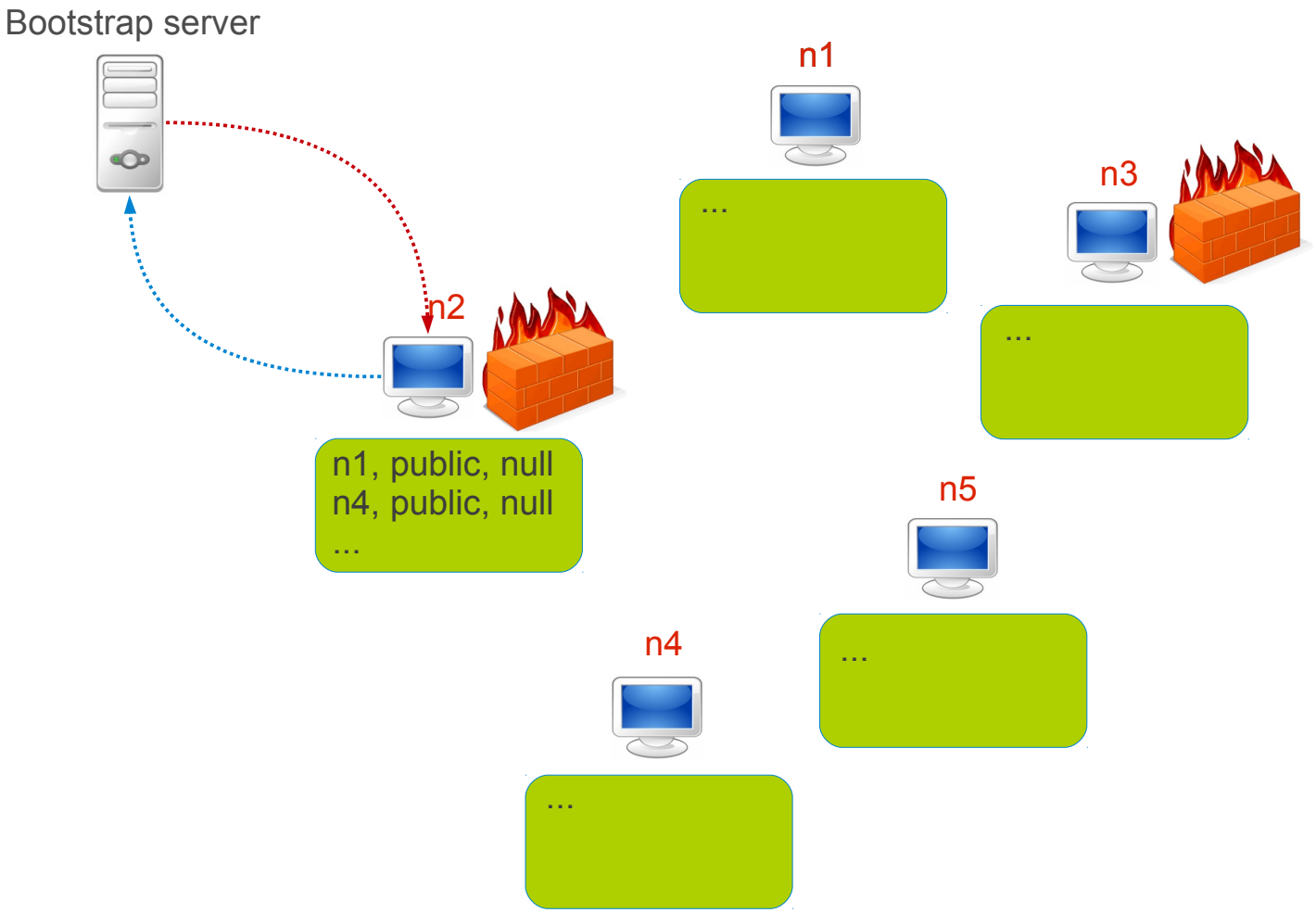
Partnering (1/10)



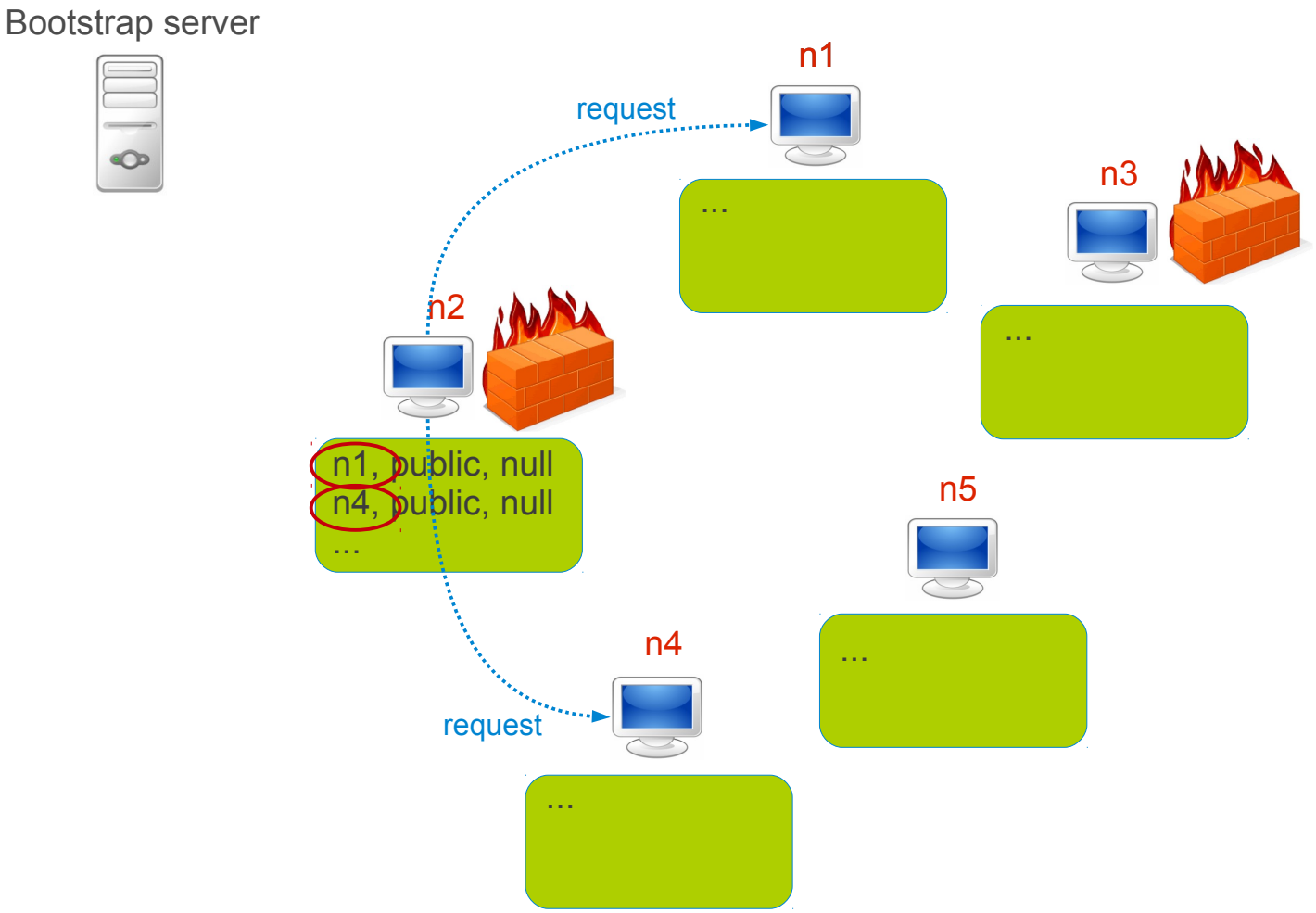
Partnering (2/10)



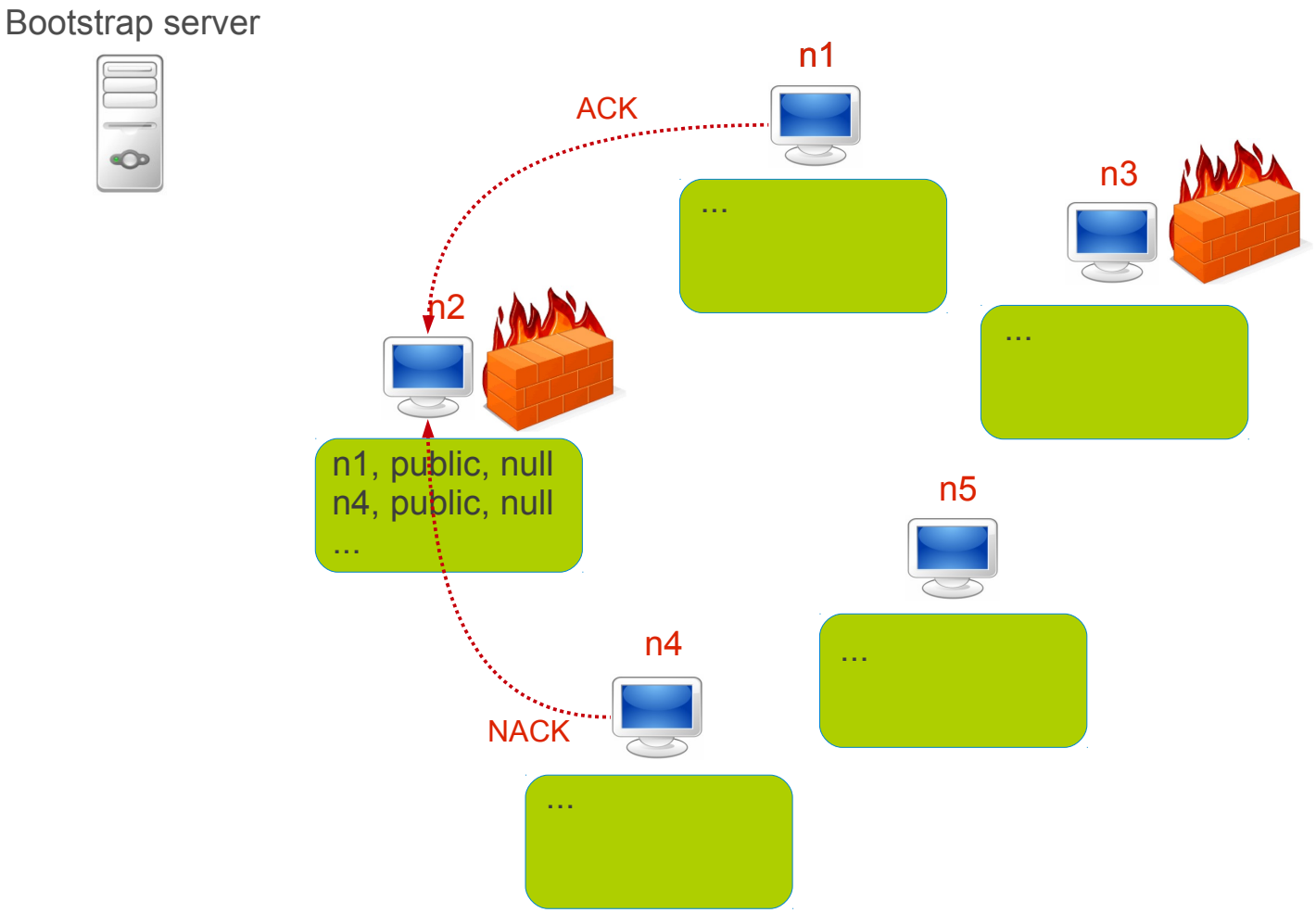
Partnering (3/10)



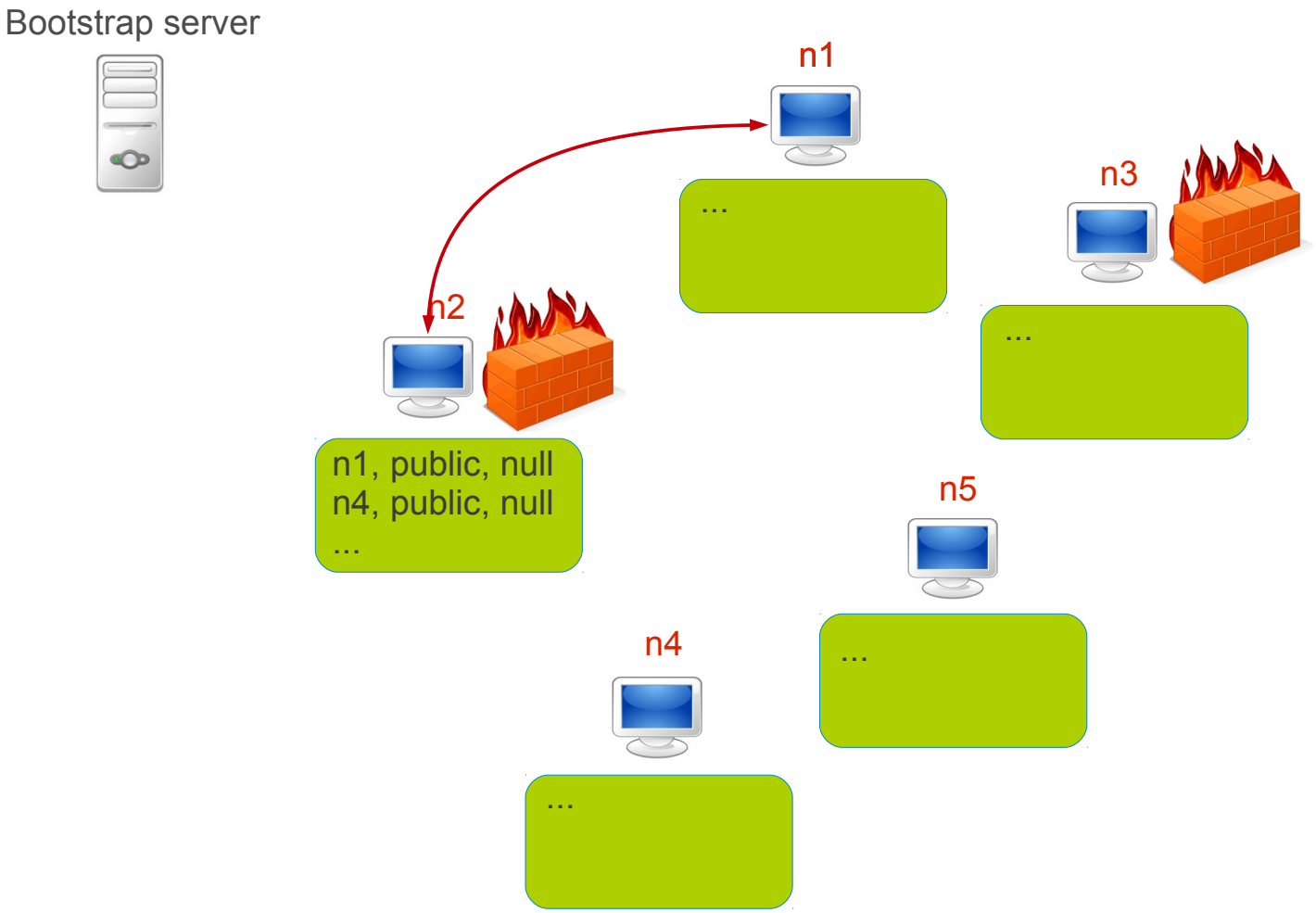
Partnering (4/10)



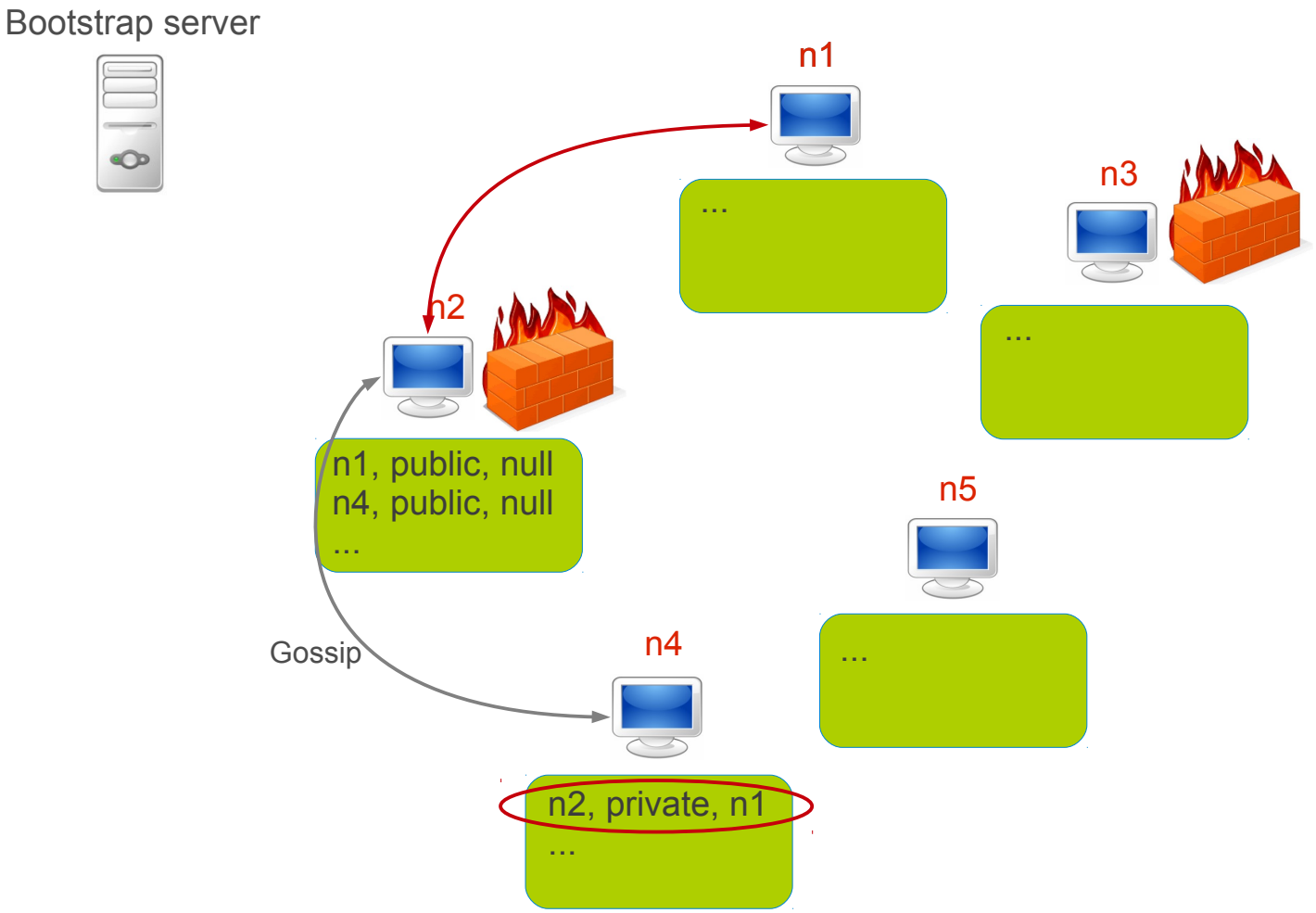
Partnering (5/10)



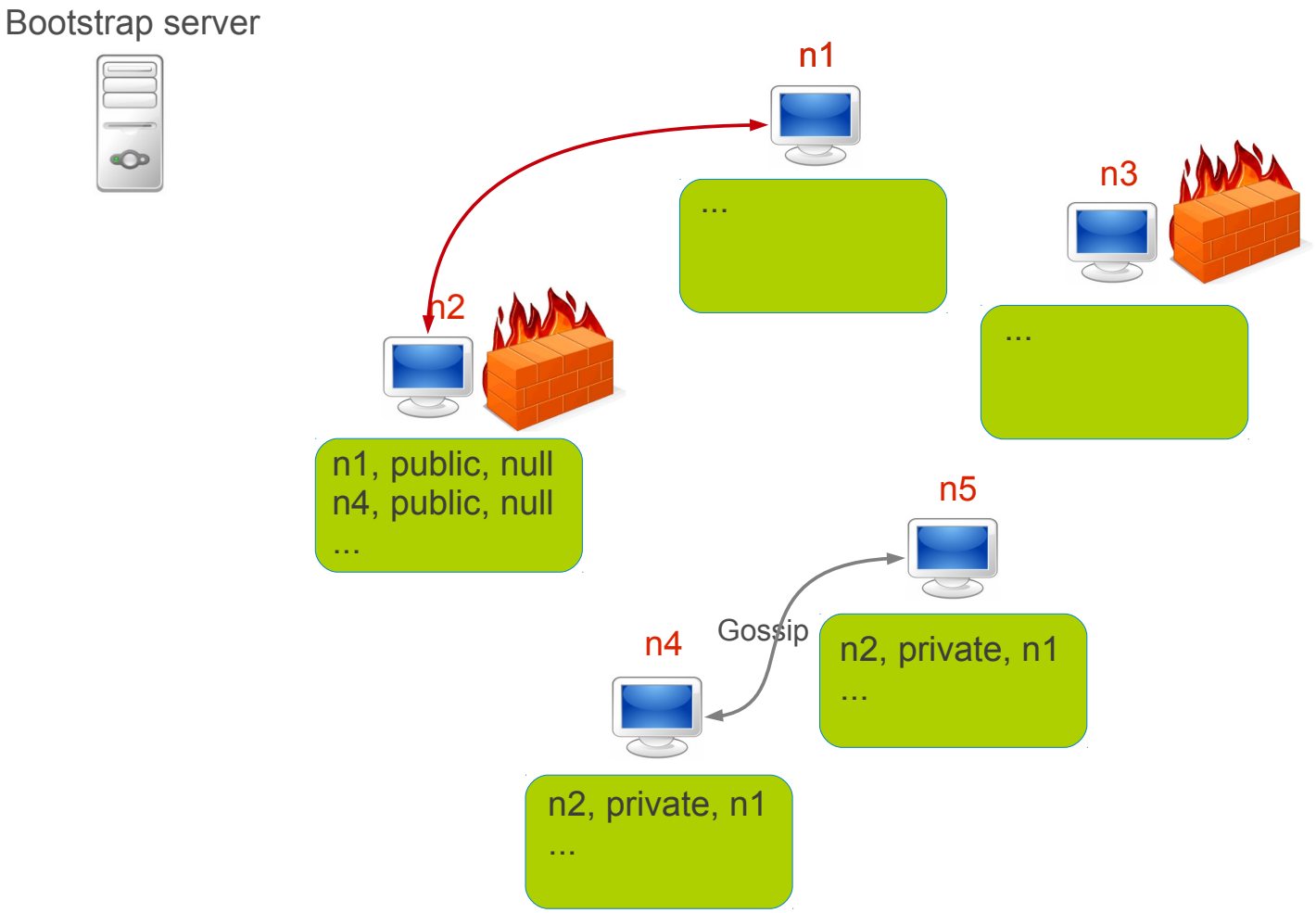
Partnering (6/10)



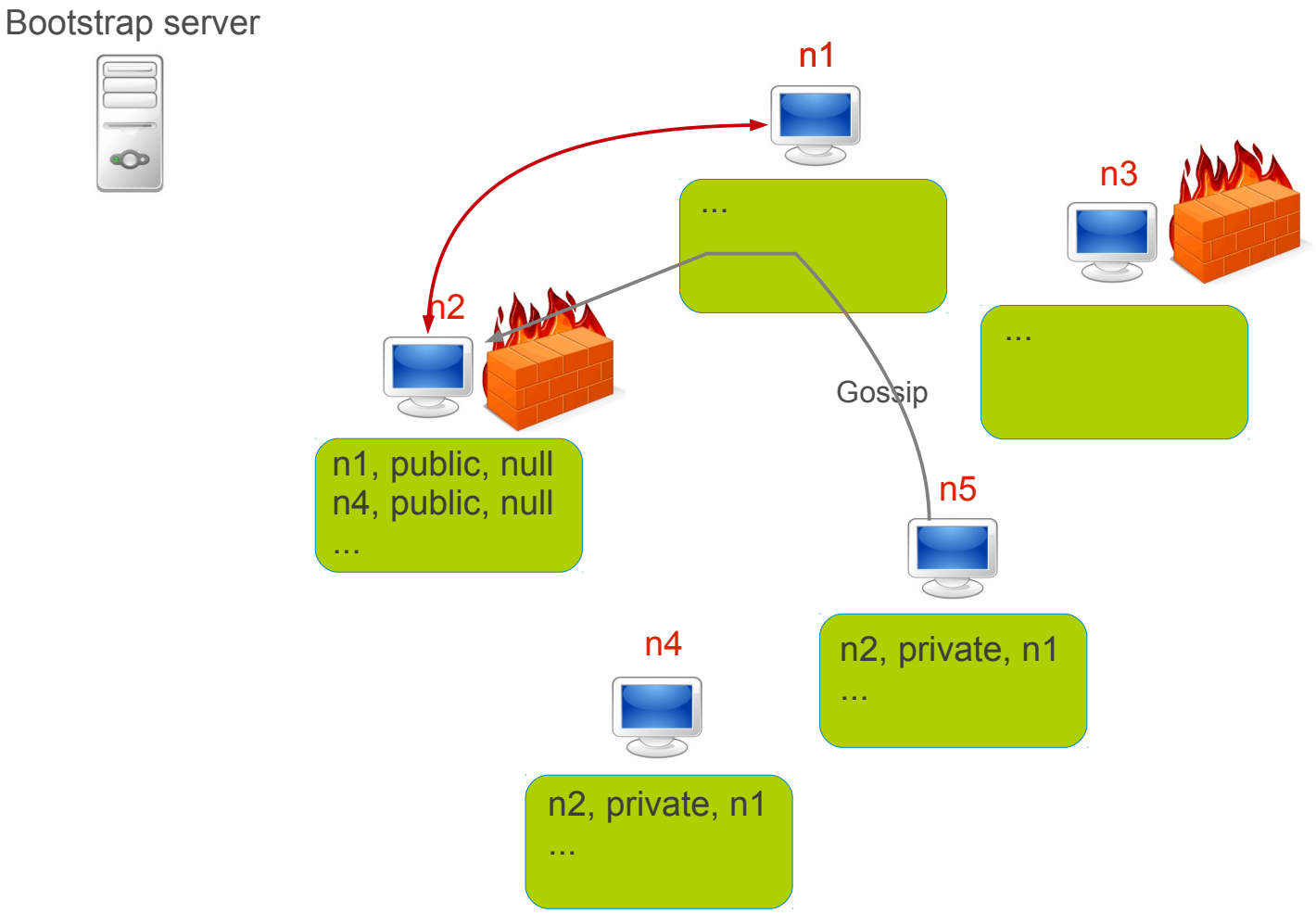
Partnering (7/10)



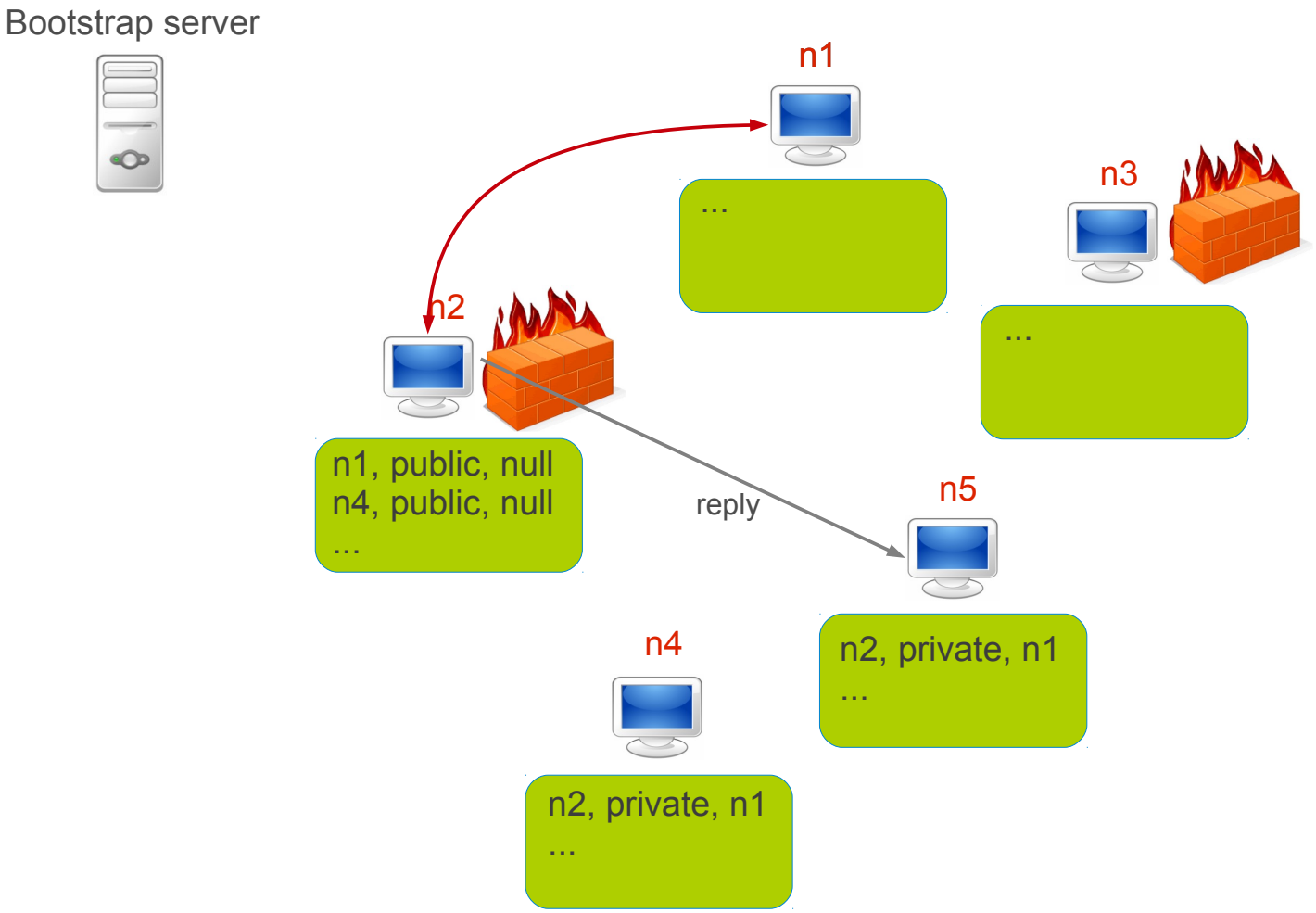
Partnering (8/10)



Partnering (9/10)



Partnering (10/10)



Relaying or Hole Punching?

- Relaying?
  - Enables faster connection establishment.
  - Allowing for shorter periodic cycles for gossiping.
    - Necessary in dynamic networks
    - Improve convergence time
- Hole punching?
  - Decreases load on public nodes.

Experiments

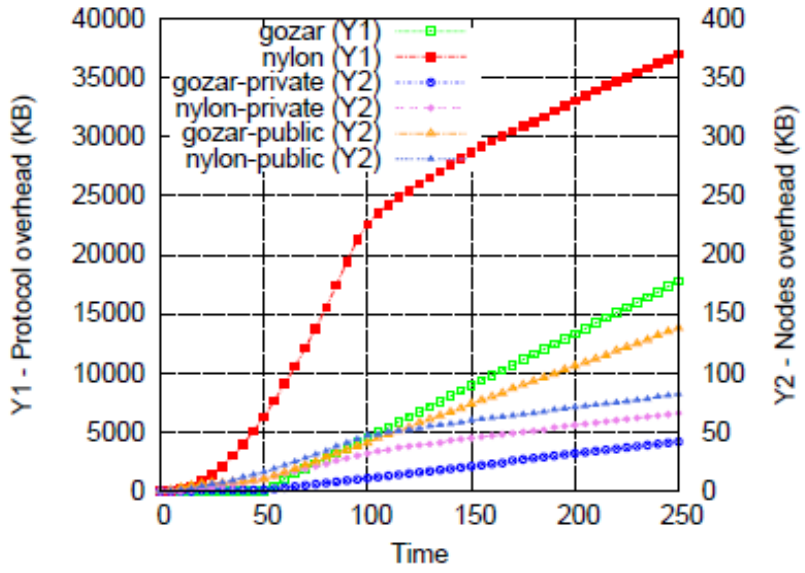
Experiment Setup

- Using the Kompics as a simulator platform.
- King dataset is used to model the latencies between nodes.
- 80% of nodes are private and 20% are public.
- Compare with Nylon and ARRG.
- Cyclon is used as a baseline.

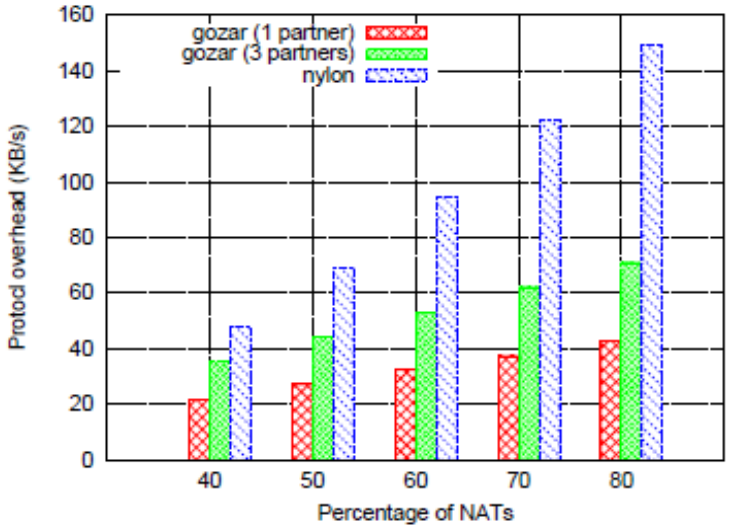
Metrics

- Randomness properties:
  - Local randomness
  - In-degree distribution
  - Clustering coefficient
  - Avg. path length
- Protocol overhead.
- Fairness and connectivity in catastrophic failure.

Protocol Overhead



(a) Protocol overhead of Gozar vs. Nylon.



(b) Overhead traffic of Gozar vs. Nylon for varying percentages of private nodes.

Summary and The Future work

Summary

- Distributed market model
- GradienTv and Sepidar → multiple-tree/push/gossip
- Glive → mesh/pull/gossip
- Gozar → NAT friendly tail/push-pull/swapper peer sampling service

Future Work

- Upload bandwidth as the only influencing parameter in the overlay construction.
  - Extend to include other important characteristics, such as node uptime, load, reputation, and locality.
- The collusion problem.
- Integrate our existing streaming systems Gozar and implement it in the open Internet.

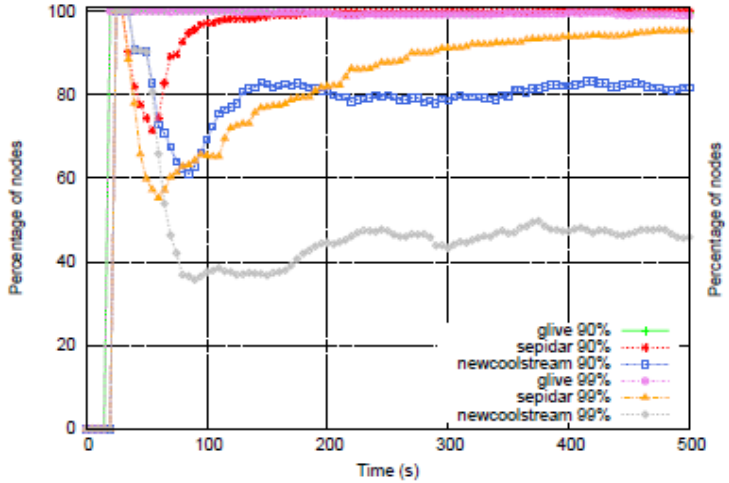
Thank you

Spidar vs. Glive vs. Newcoolstreaming

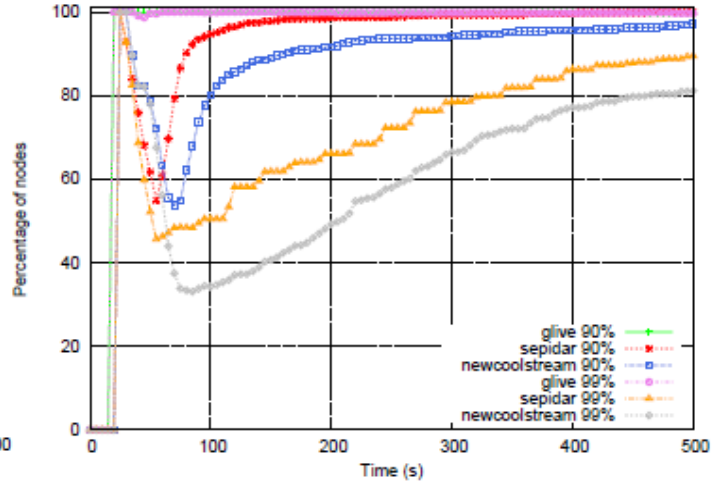
Experiment Setup

- 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/stripe 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.

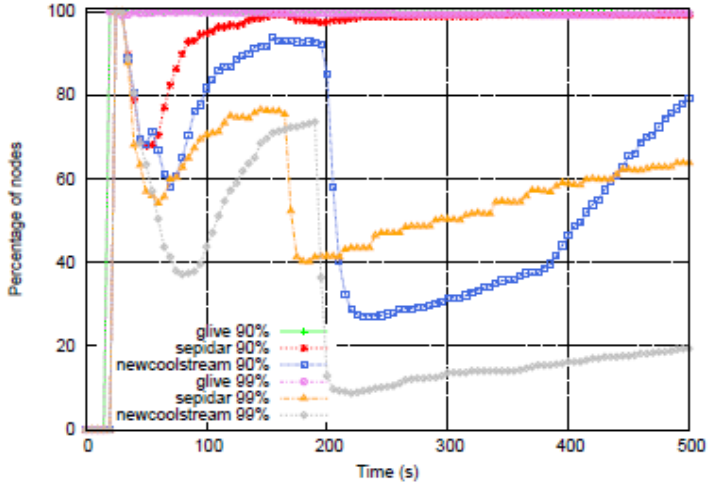
Playback Continuity



(a) Churn.

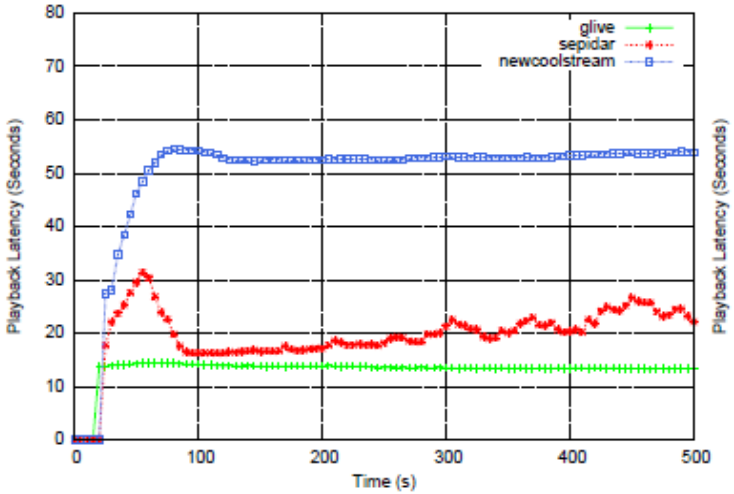


(b) Flash Crowd.

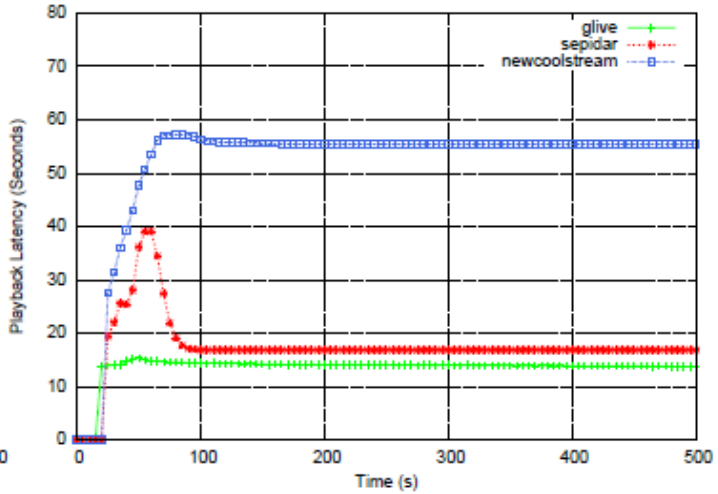


(c) Catastrophic failure.

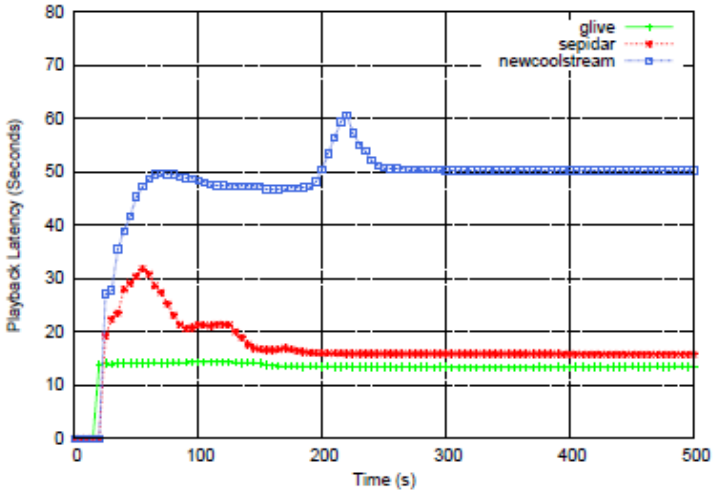
Playback Latency



(a) Churn.

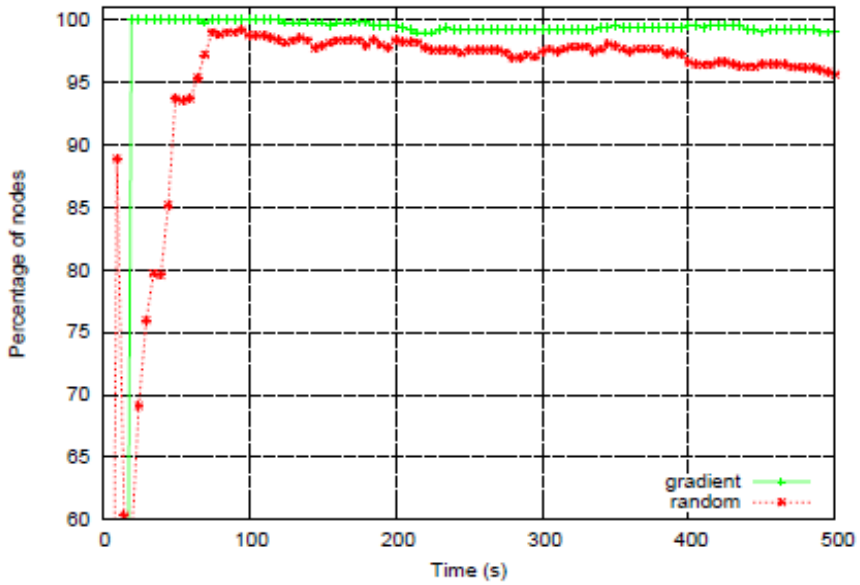


(b) Flash Crowd.

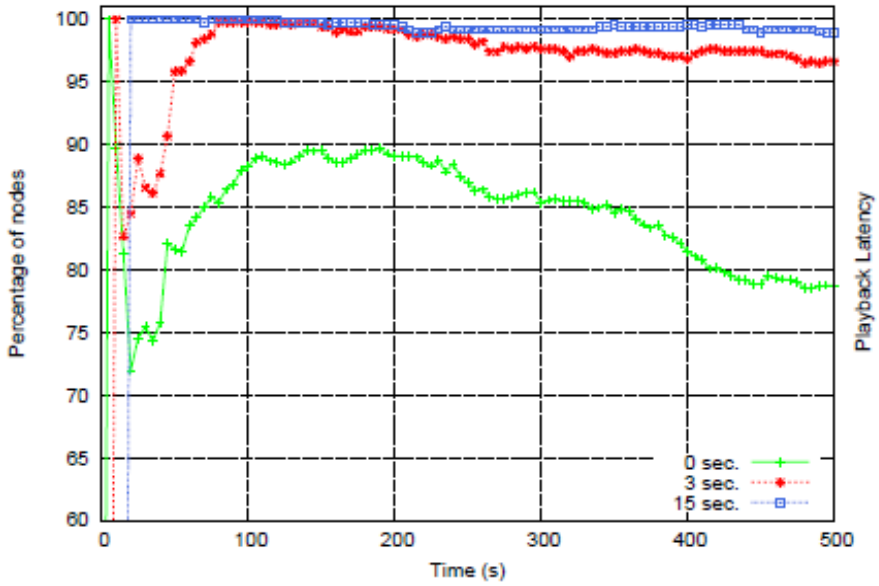


(c) Catastrophic failure.

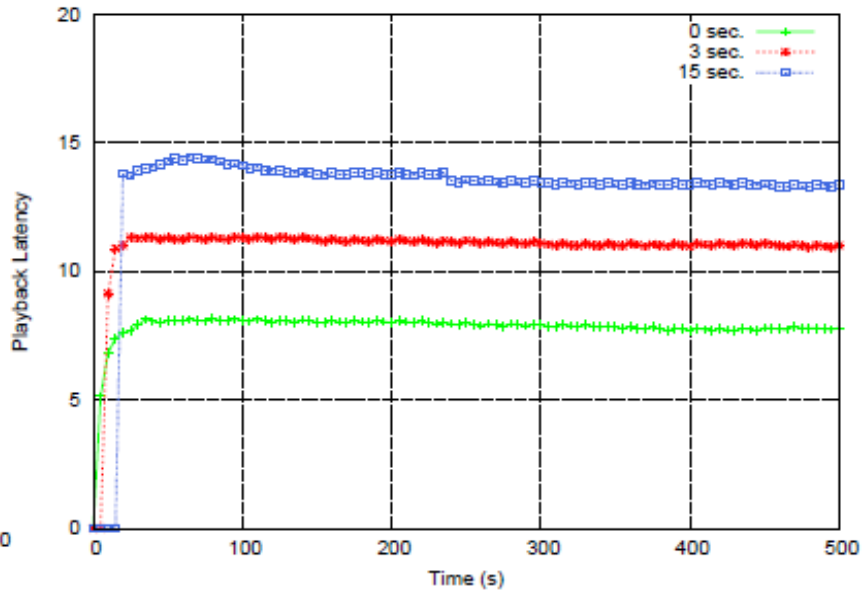
Gradient overlay vs. Random overlay



Buffering Time



(a) 99% of playback continuity.



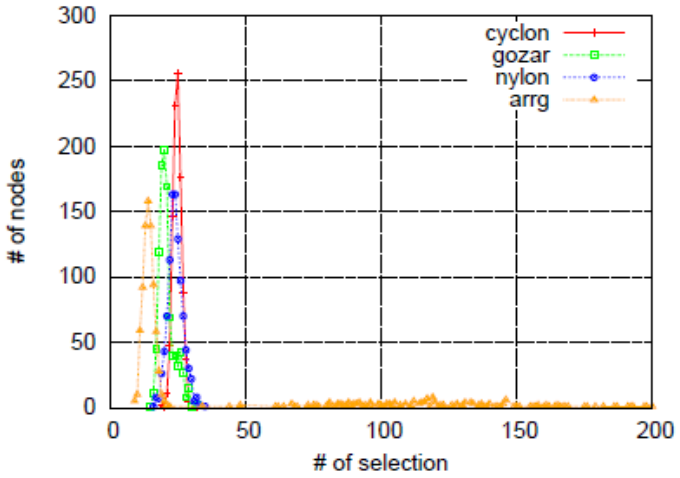
(b) Playback latency.

Gozar vs. Nylon vs. ARRG

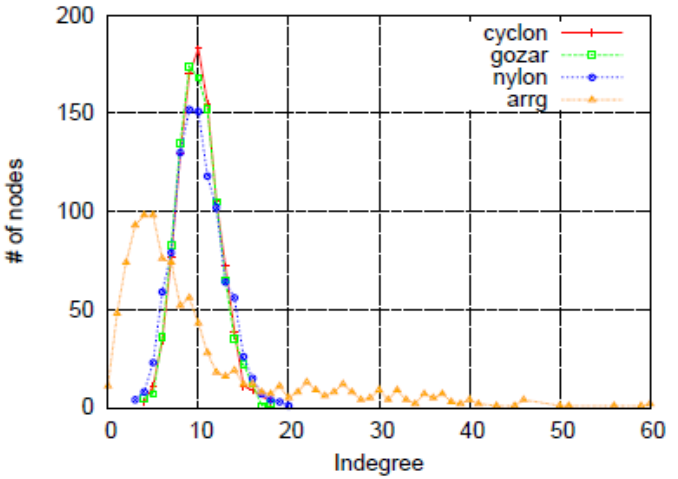
Experiment Setup

- Using the Kompics as a simulator platform.
- King dataset is used to model the latencies between nodes.
- 80% of nodes are private and 20% are public.
- Compare with Nylon and ARRG.
- Cyclon is used as a baseline.

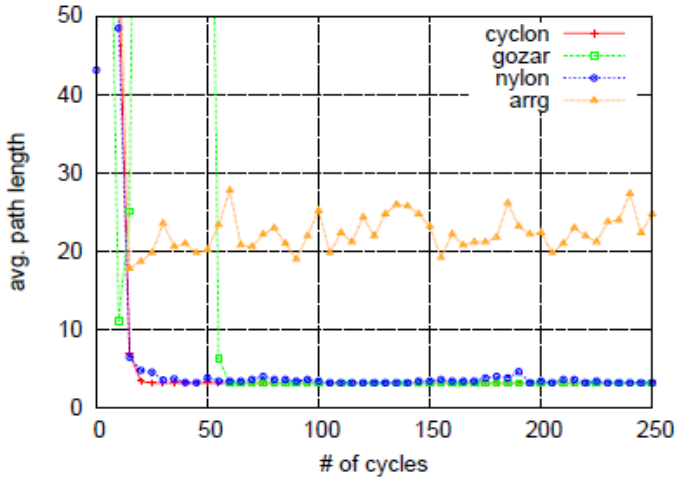
Randomness



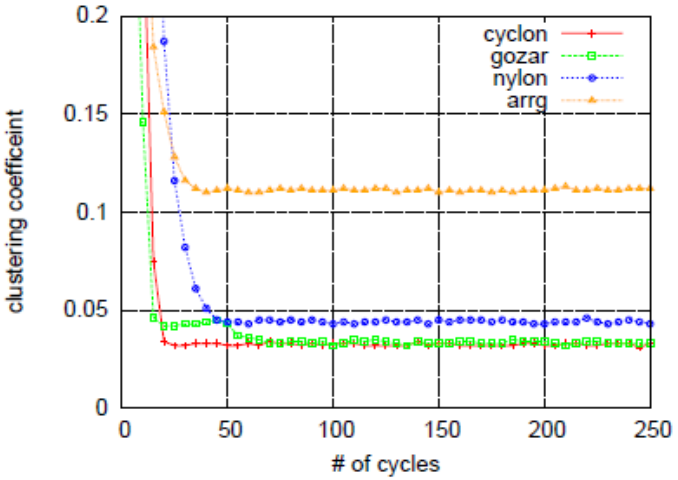
(a) Local randomness.



(b) Indegree distribution.

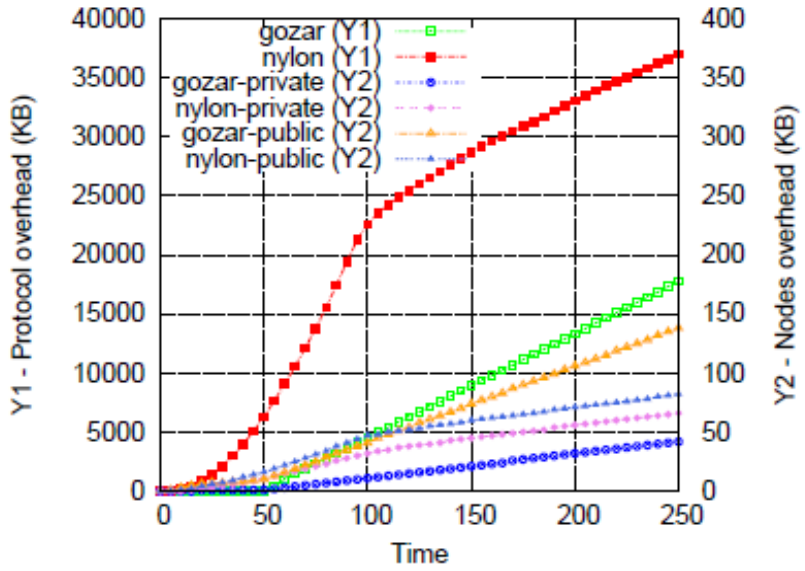


(c) Average path length.

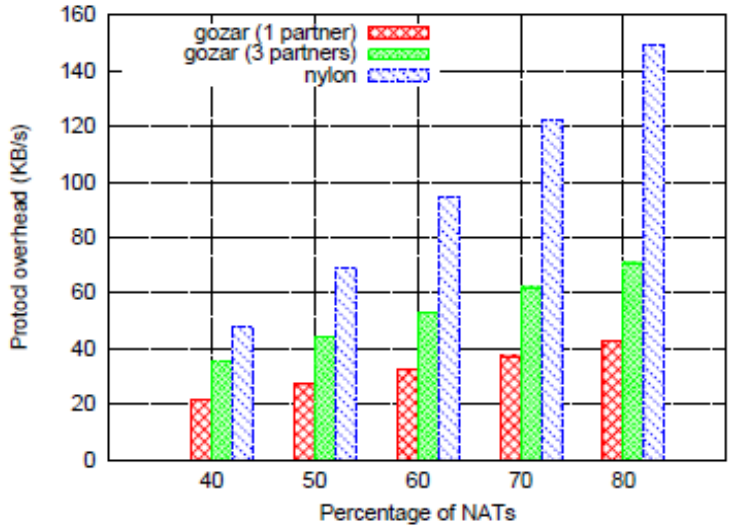


(d) Clustering coefficient.

Protocol Overhead

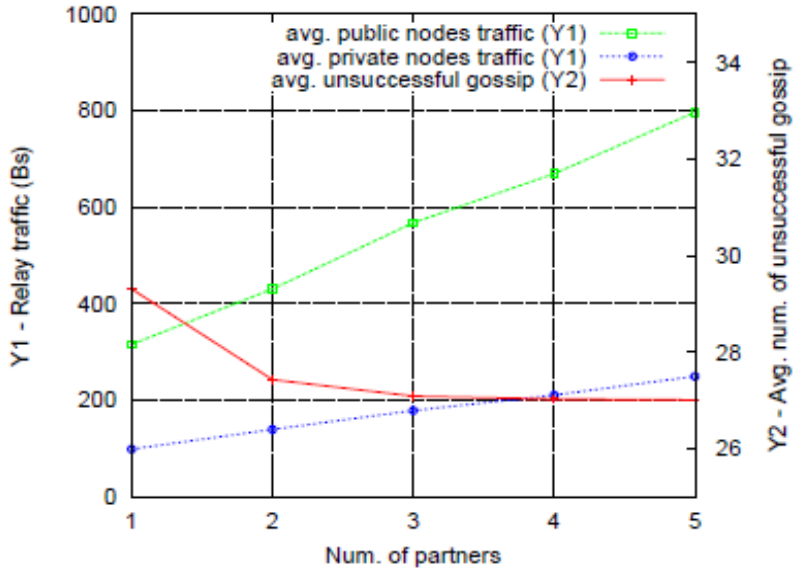


(a) Protocol overhead of Gozar vs. Nylon.

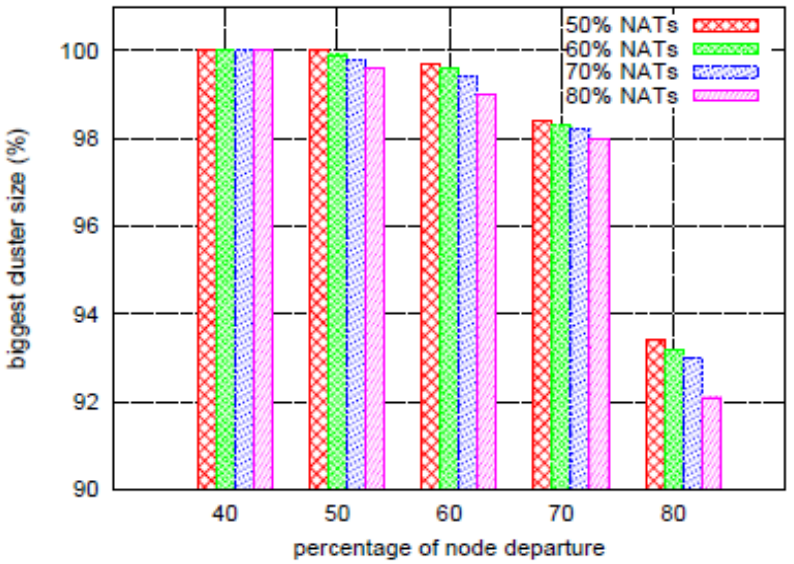


(b) Overhead traffic of Gozar vs. Nylon for varying percentages of private nodes.

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.