Live Streaming in P2P and Hybrid P2P-Cloud Environments for the Open Internet

Amir H. Payberah

Advisors:
Prof. Seif Haridi
Dr. Jim Dowling

ROYAL INSTITUTE
OF TECHNOLOGY

Doctoral Thesis in Information and Communication Technology
Stockholm, Sweden, June 13, 2013, 13:00

Outline

- Introduction
- Contribution 1 (Sepidar and GLive) 1
  - Problem description
  - Solution
- Contribution 2 (CLive) 2
  - Problem description
  - Solution
- Contribution 3 (Gozar and Croupier) 3
  - Problem description
  - Solution
- Wrap Up

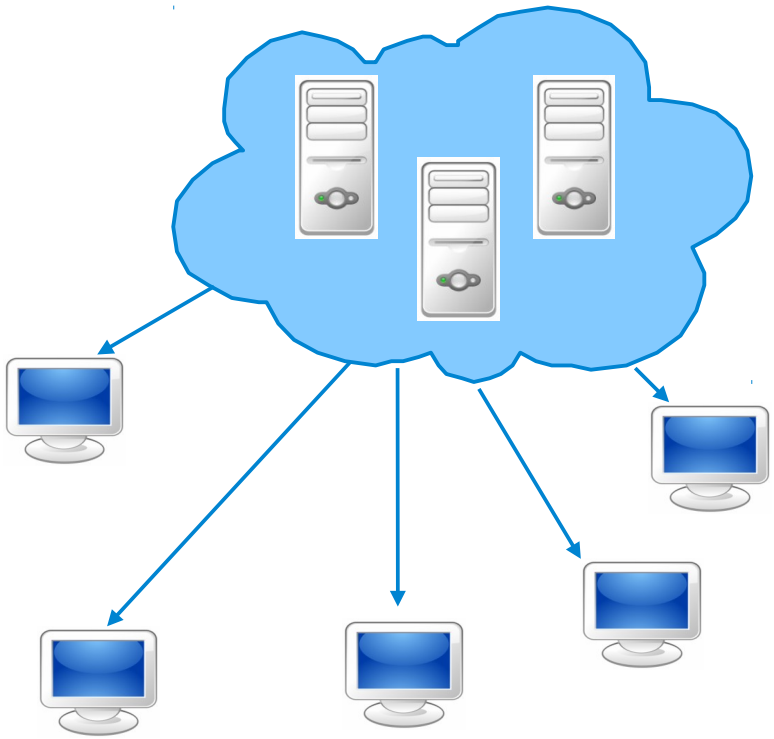
Introduction

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.
- It could be
  - Live
  - On Demand

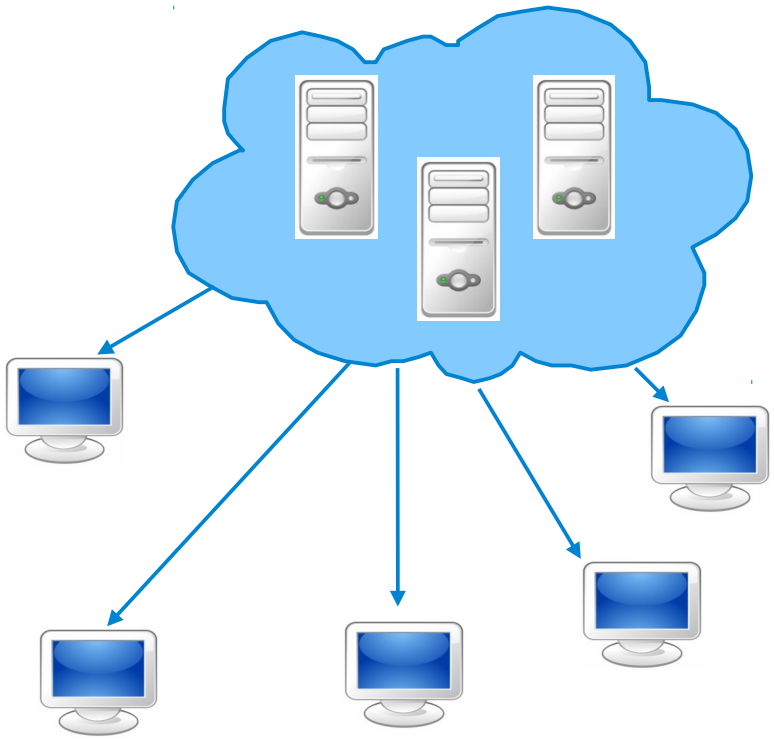
Solutions for Media Streaming

Client-Server

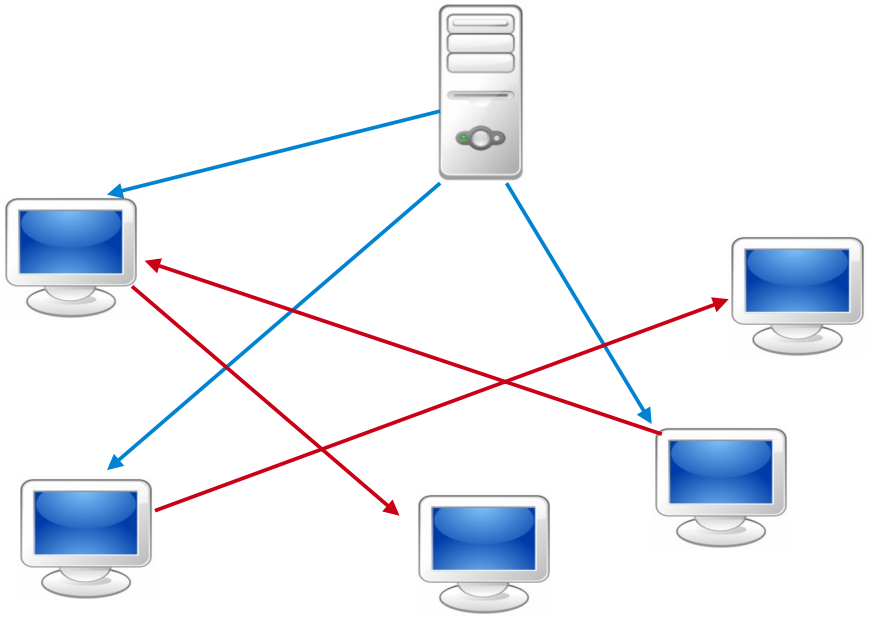


Solutions for Media Streaming

Client-Server



Peer-to-Peer



QoS in P2P Media Streaming Systems

- High playback continuity: Smooth playback
- Short playback latency (only for Live Streaming)

P2P Media Streaming Challenges

- Churn in the system
- Free-riding problem
- Bottleneck in the overlay network
- Connectivity Problem (NAT)

Thesis Contribution

- Churn in the system
- Free-riding problem

Sepidar and GLive — 1

- Bottleneck in the overlay network

CLive — 2

- Connectivity Problem (NAT)

Gozar and Croupier — 3

Sepidar / GLive

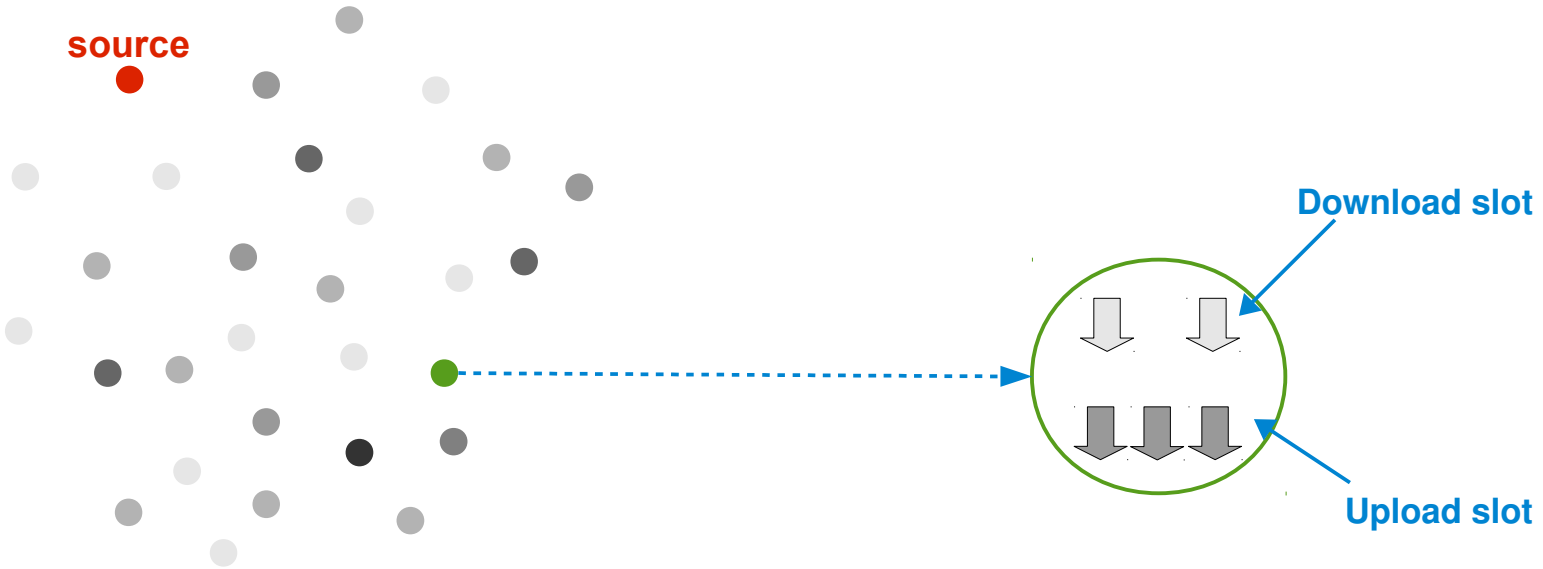
P2P Solutions for Live Media Streaming

Problem Description

Problem Description (1/5)

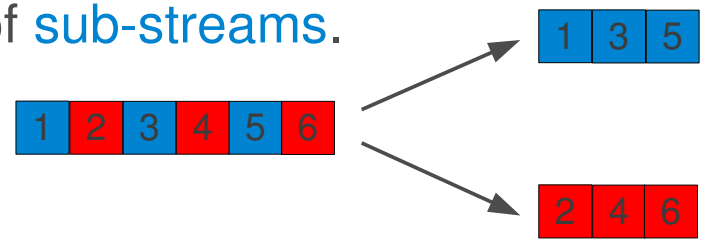
- Building and optimizing a P2P overlay for live media streaming.
  - High QoS

- Bounded number of download connections and upload connections.

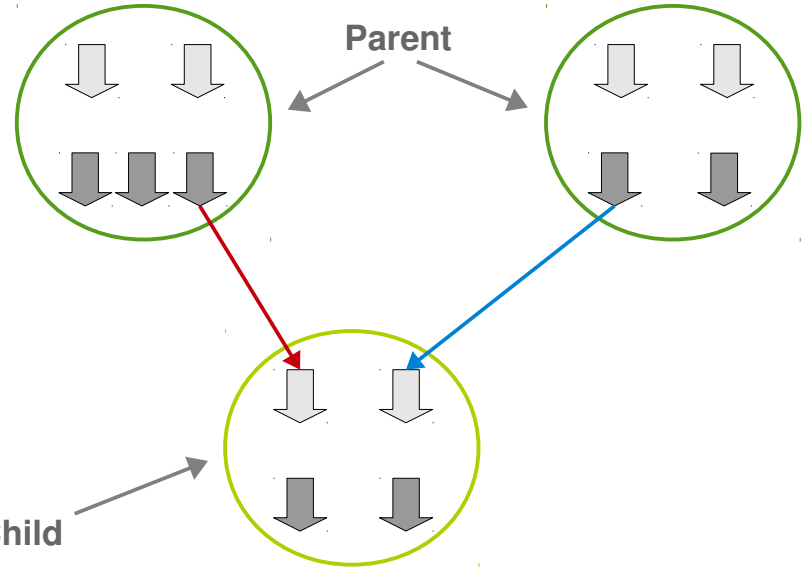


Problem Description (2/5)

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

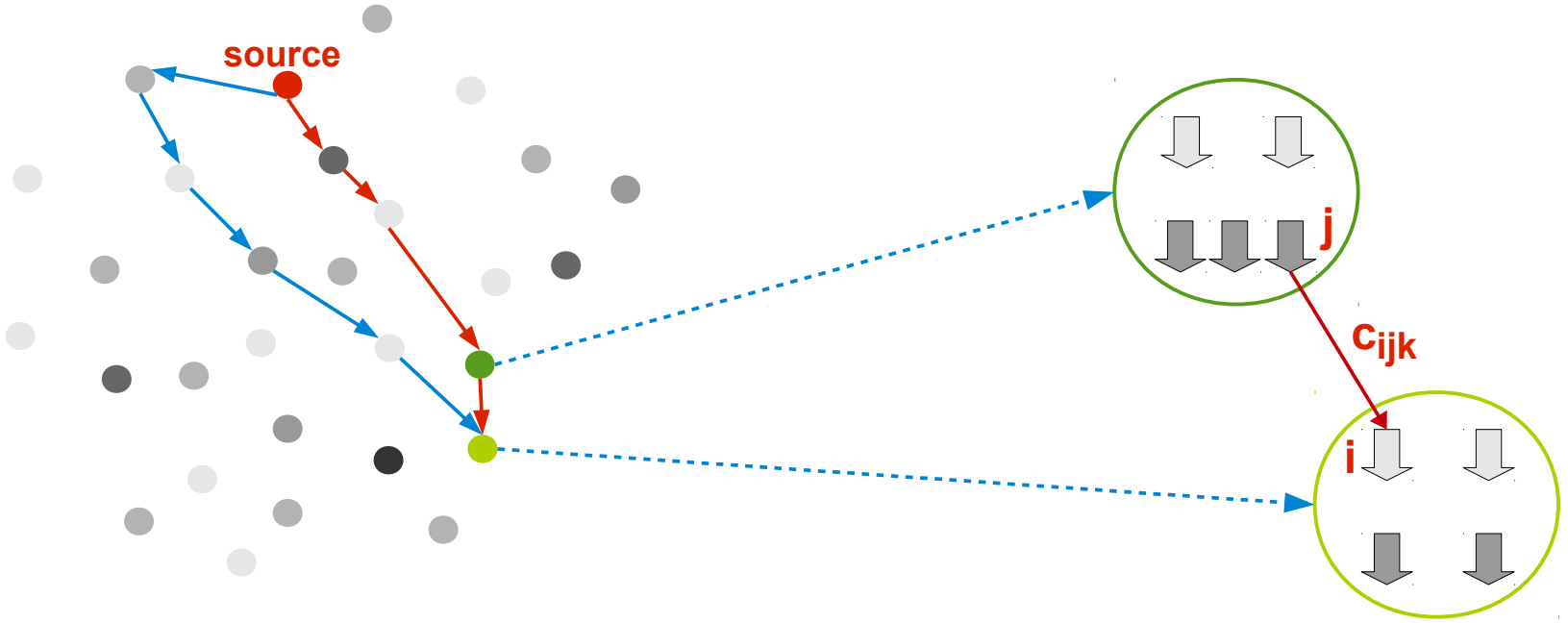


- To provide the full media to all the nodes: complete assignment (A)
  - Assign all download-slots in a node.
  - Download distinct sub-stream.



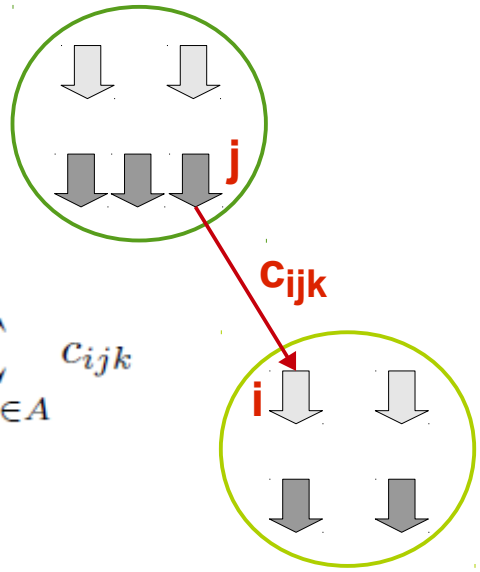
Problem Description (3/5)

- Cost c_{ijk}: the number of hops from the owner of the upload-slots j, to the source for the sub-stream k.



Problem Description (4/5)

- An optimization problem:



- Objective function
  - Find a complete assignment that minimizes the total cos $\sum_{(i,j,k)\in A} c_{ijk}$
- Subject to
  - Each 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 sub-streams.

Problem Description (5/5)

- Centralized solution:
  - Needs global knowledge.
  - Possible for small system sizes.
- Distributed solution:
  - No global knowledge.
  - Inspired by auction algorithms.

Solution

P2P Streaming Overlay Construction 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
  - ...

Sepidar vs. GLive

- 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
  - ...

Sepidar

Multiple-tree
Push
Gossip

Sepidar vs. GLive

- 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
  - ...

Sepidar

Multiple-tree
Push
Gossip

GLive

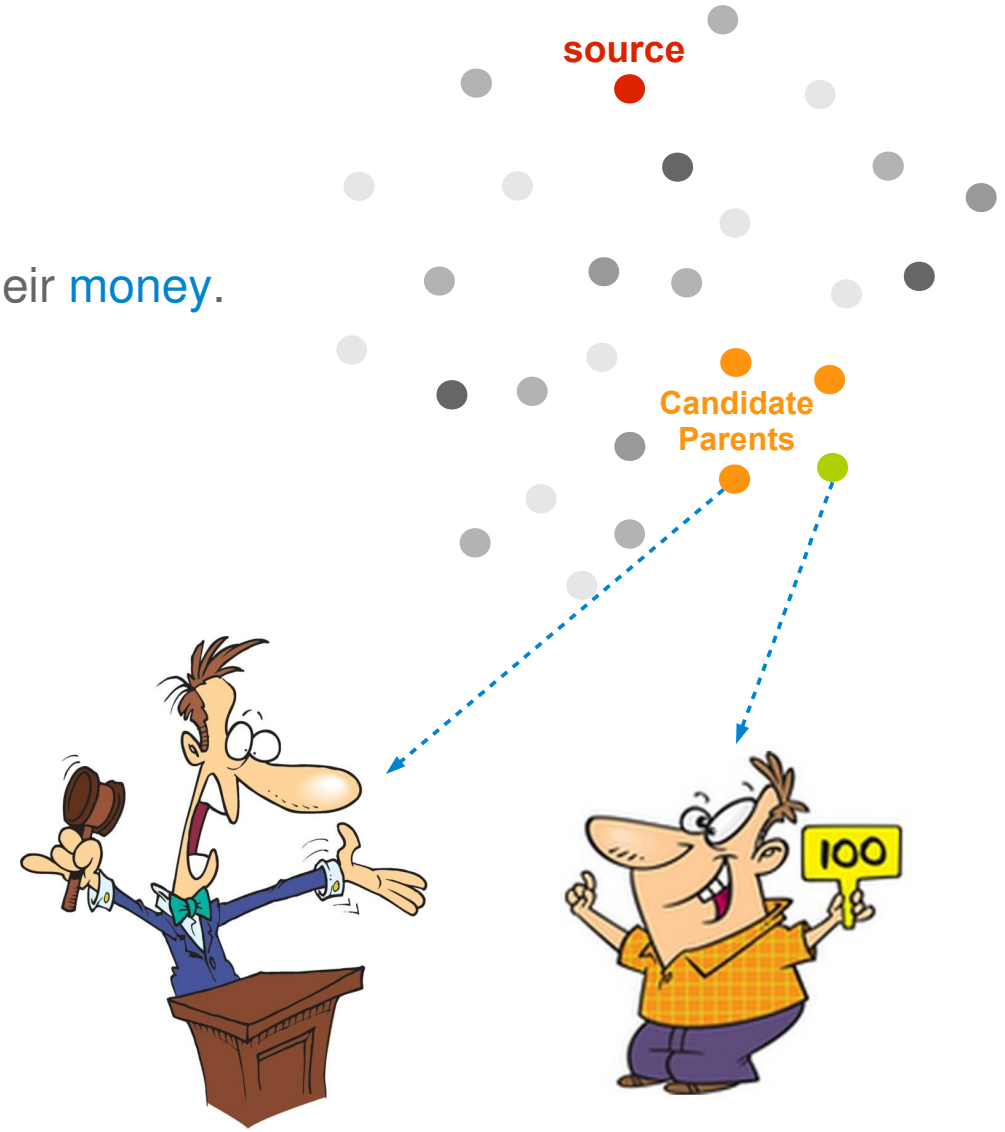
Mesh
Pull
Gossip

Streaming Overlay Construction (1/2)

- Complete assignment that minimizes the costs.
- Each node knows only a small number of nodes in the system.
- Putting the nodes with higher number of upload slots closer to the source.

Streaming Overlay Construction – Auction Model (2/2)

- Child nodes bid for better parents.
  - Closer to the source.
  - Use their number of upload-slots as their money.
- Parent nodes accept the highest bid.

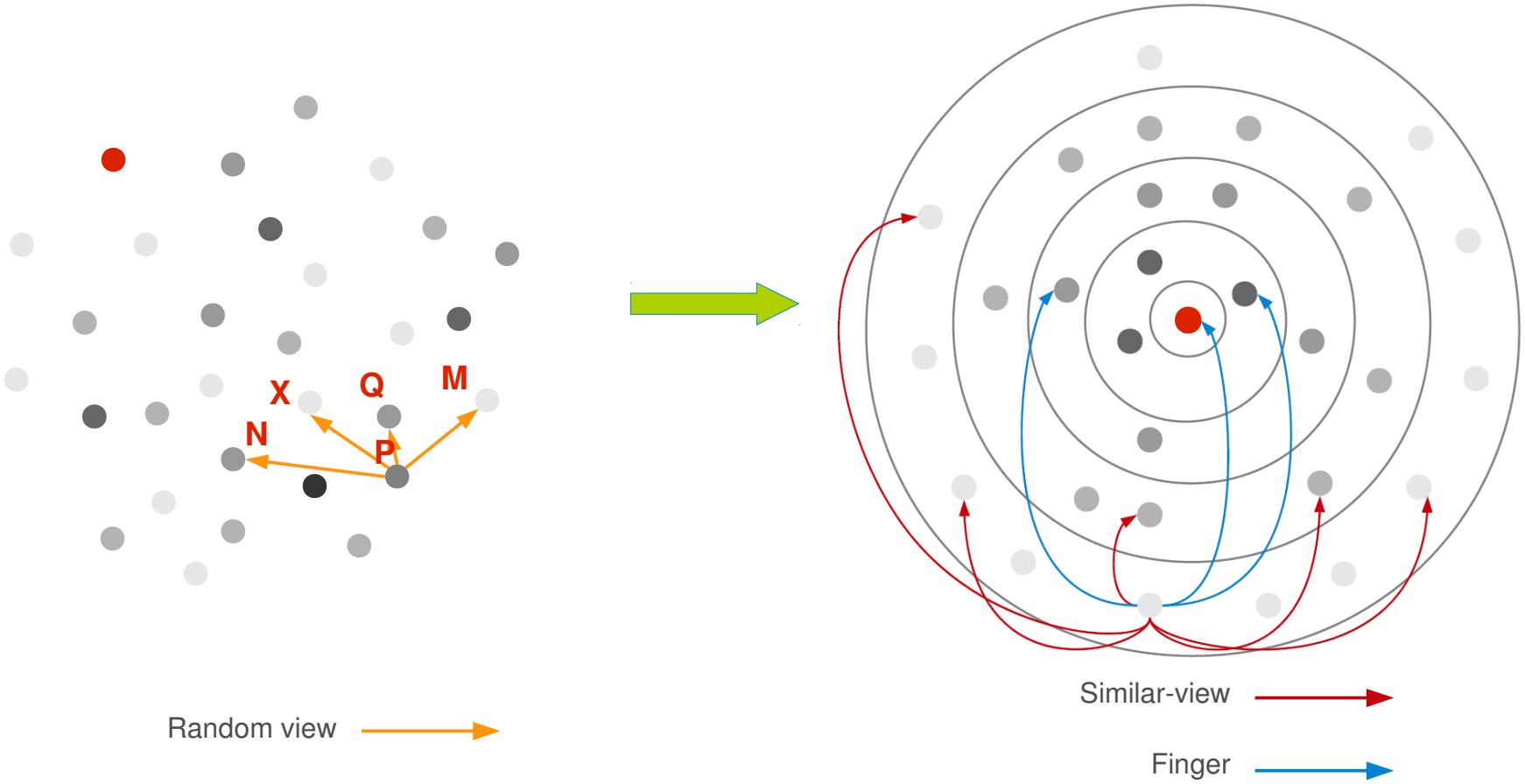


Two More Questions?

- How to make the partial view at each node?
- How to handle free-riding nodes?

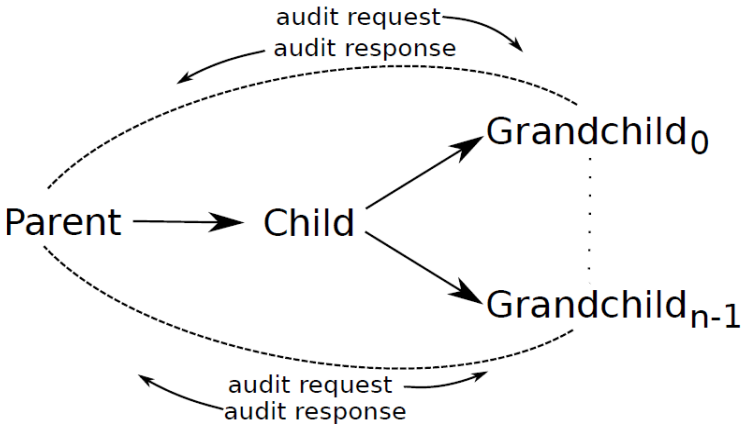
How to Build Partial View at Each Node?

- The Gradient overlay.
- Limit exploration to the set of nodes with a similar number of upload-slots.

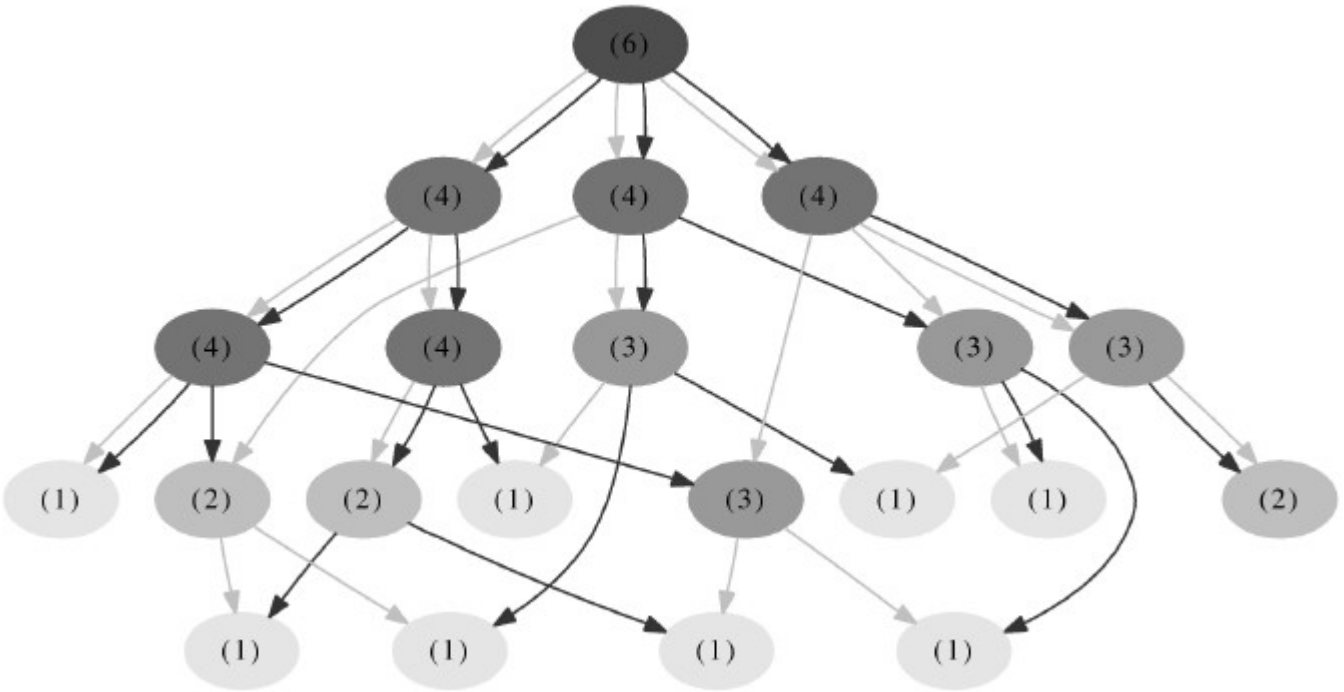


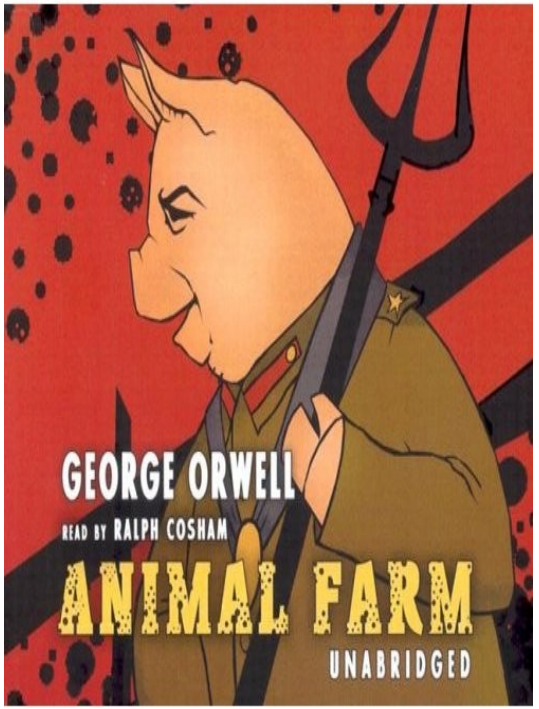
How to Prevent Free-riding?

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



All Nodes are Equal, but Some Nodes are More Equal





Sepidar/GLive Summary

- P2P overlays for live media streaming.
- Distributed market model to construct the streaming overlay.
- The Gradient overlay to speed up the overlay construction.
- Transitive auditing to detect the free-riders.

CLive

A Hybrid P2P-Cloud Solution for Live Media Streaming

Problem Description

Problem Description (1/3)

- Bottlenecks in P2P video streaming systems: upload bandwidth

- A potential solution: P2P network is assisted by a cloud computing.

Problem Description (2/3)

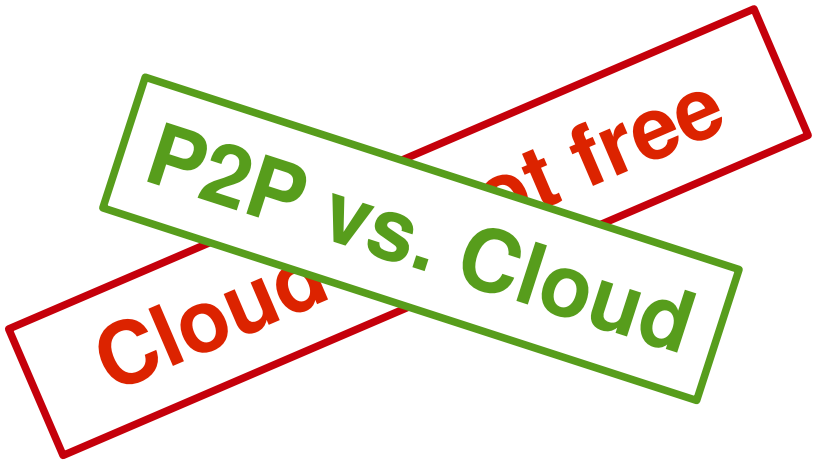
- Bottlenecks in P2P video streaming systems: upload bandwidth

- A potential solution: P2P network is assisted by a cloud computing.

Problem Description (3/3)

- Bottlenecks in P2P video streaming systems: upload bandwidth

- A potential solution: P2P network is assisted by a cloud computing.



P2P vs. Cloud

- P2P
  - P2P resources are cheap
  - Churn may compromise availability
- Cloud
  - Superior availability
  - Cloud resources are not free

We Cannot Beat Them, Let's Restrain Them

- The cloud as a support group for P2P.

- Reduce the number of cloud interactions as much as possible.

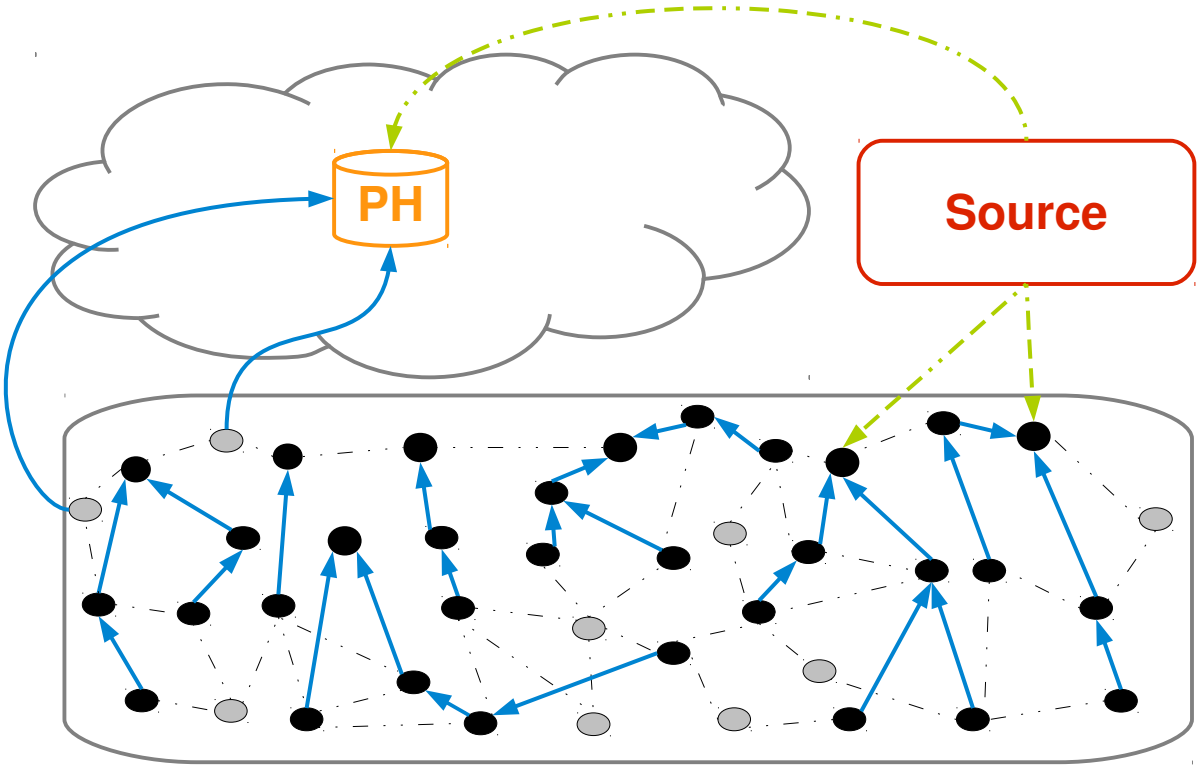
Occupy Wallstreet

Solution

Baseline Solution

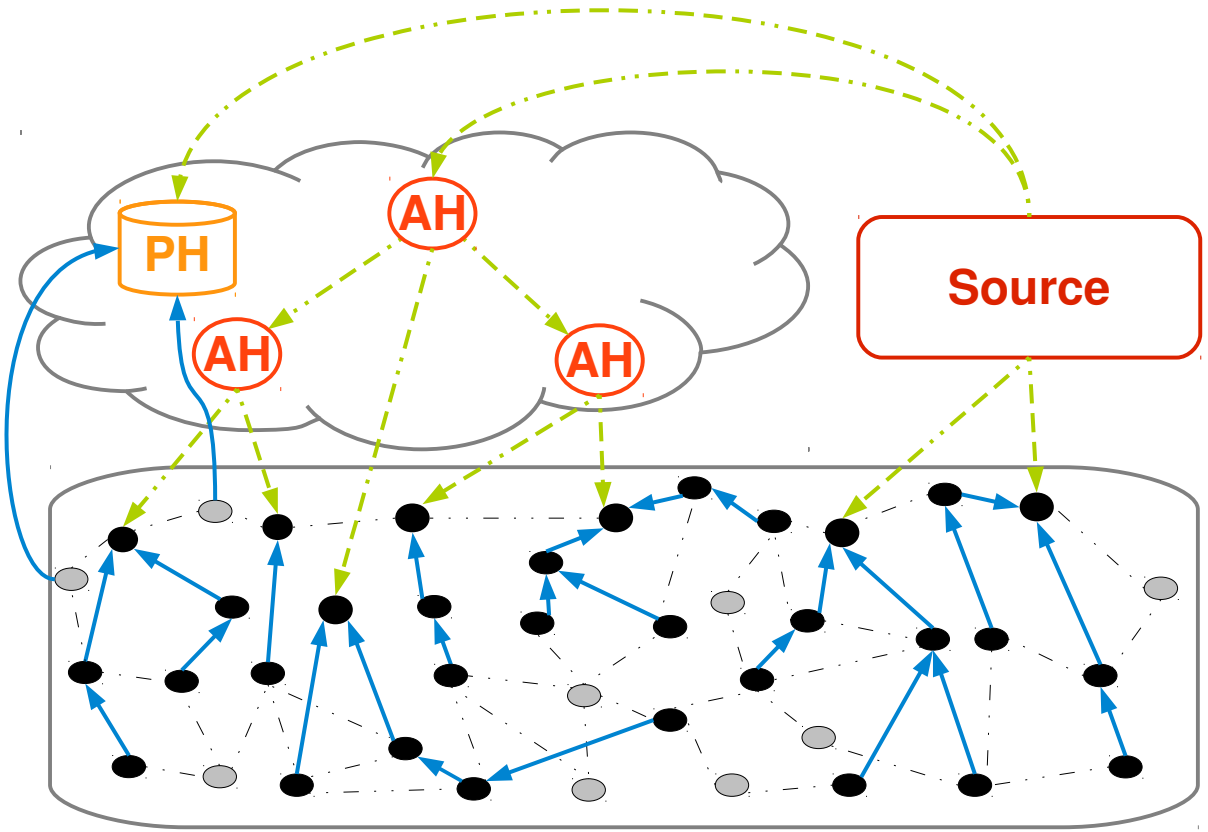
- Rent passive helpers (PH), e.g., storage, from a cloud provider.

- Nodes pull the data from the PH if they cannot receive them from other nodes on time.



CLive Solution

- Rent active helpers (AH), e.g., VM, in addition to PH.



So?

Given that the costs of AHs and PHs are different, the goal is to minimize the total cost while delivering the desired QoS.

Two Main Questions?

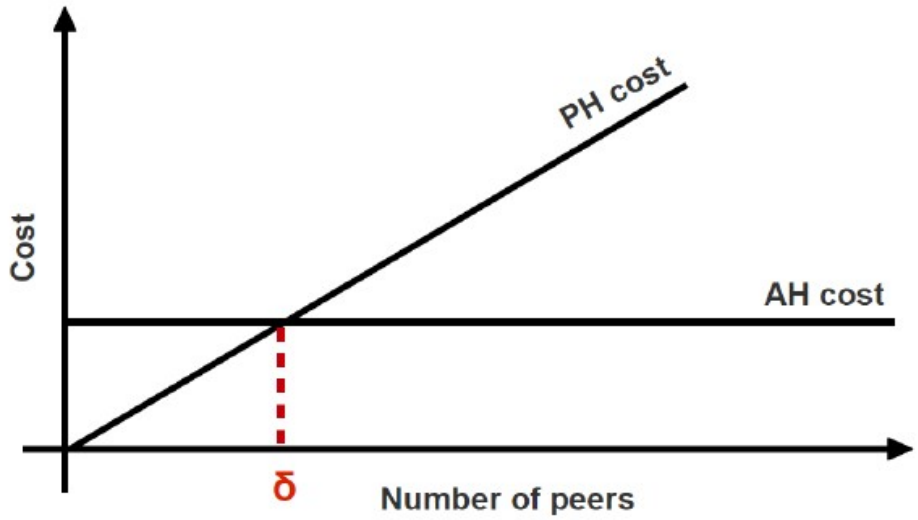
- How to estimate the extra load in the overlay?
- How many AH to add?

How to Estimate the Extra Load?

- Load = swarm size – infected nodes
- The swarm size estimation: gossip-based aggregation
- Infected nodes:
  - The number of nodes that can be served by with the existing resources in the system, without the help of PH.
  - Tree-based diffusion pattern.
  - Estimate the tree depth.
  - Estimate the upload slot distribution: gossip-based aggregation

How Many AH to Add?

- Calculate AH and PH cost in each round
- If load > ∂: add AH
- If load < ∂ - H: remove AH
  - H: number of peers served by one
- Otherwise don't change AHs



CLive Summary

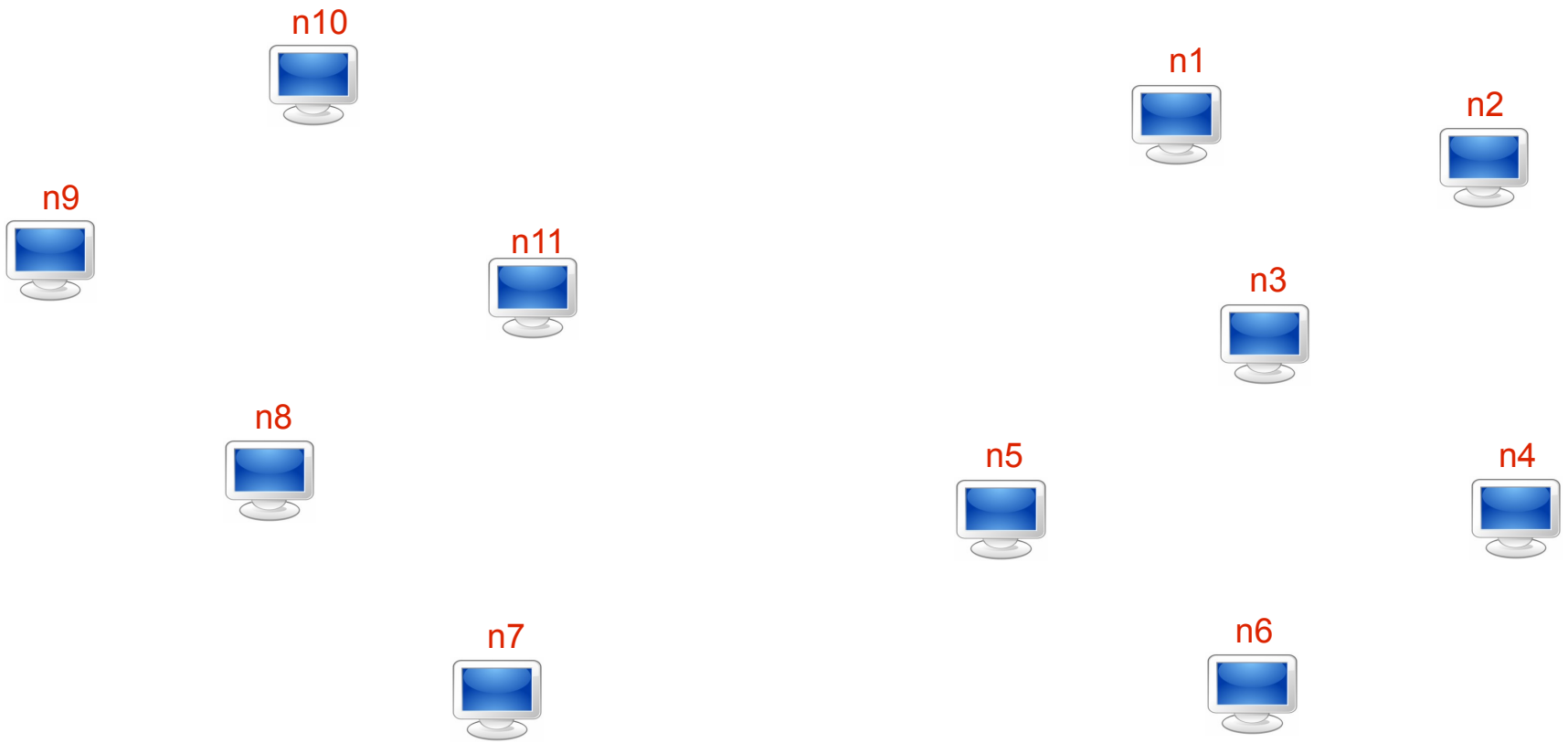
- Hybrid P2P-Cloud solution for for live media streaming.
- A combination of AHs and PH.
- Estimate the amount of extra load.
- Relay the extra load to the cloud.
- Add/remove AHs to minimise the cost.

Gozar/Croupier

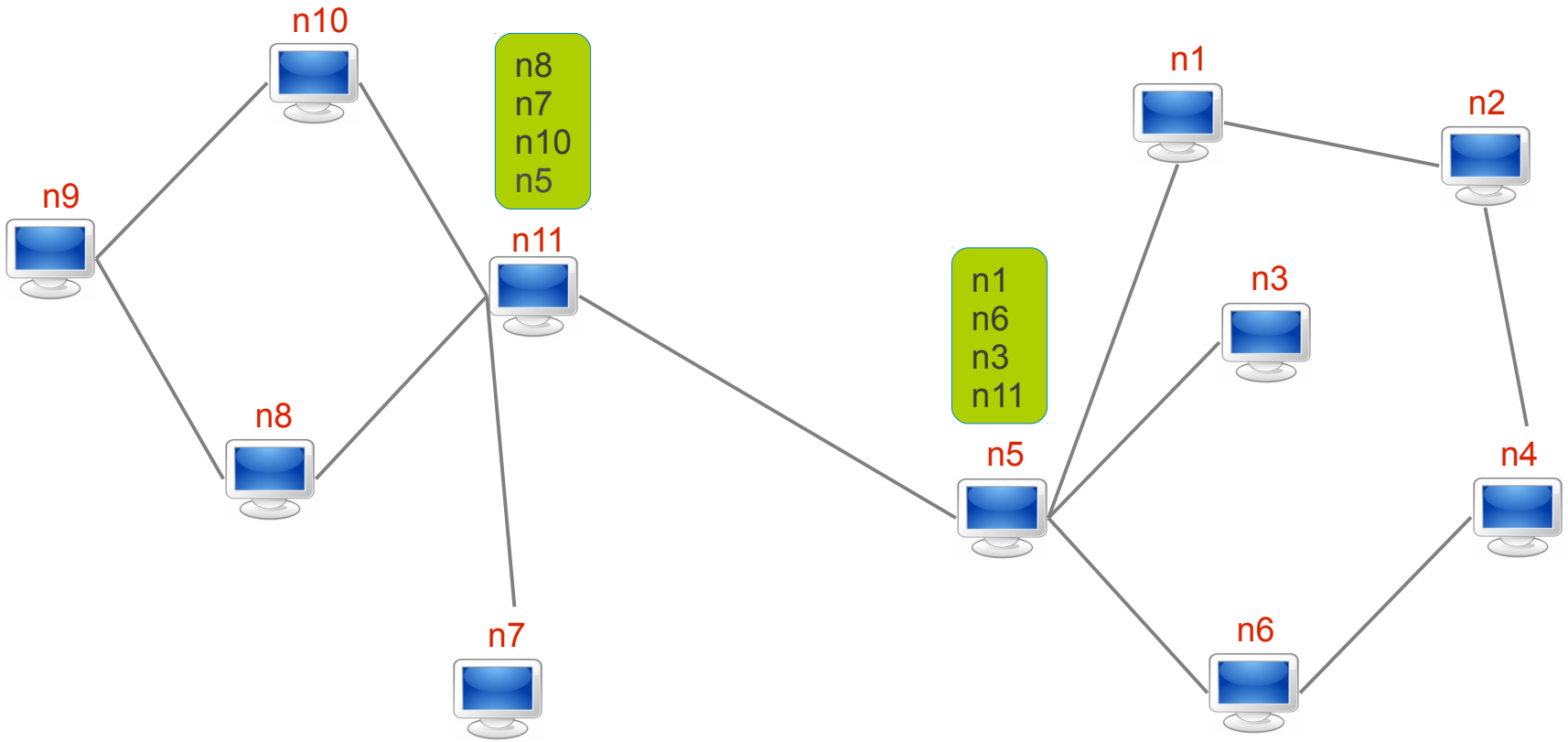
NAT-aware Peer Sampling and Distributed NAT Traversal

Problem Description

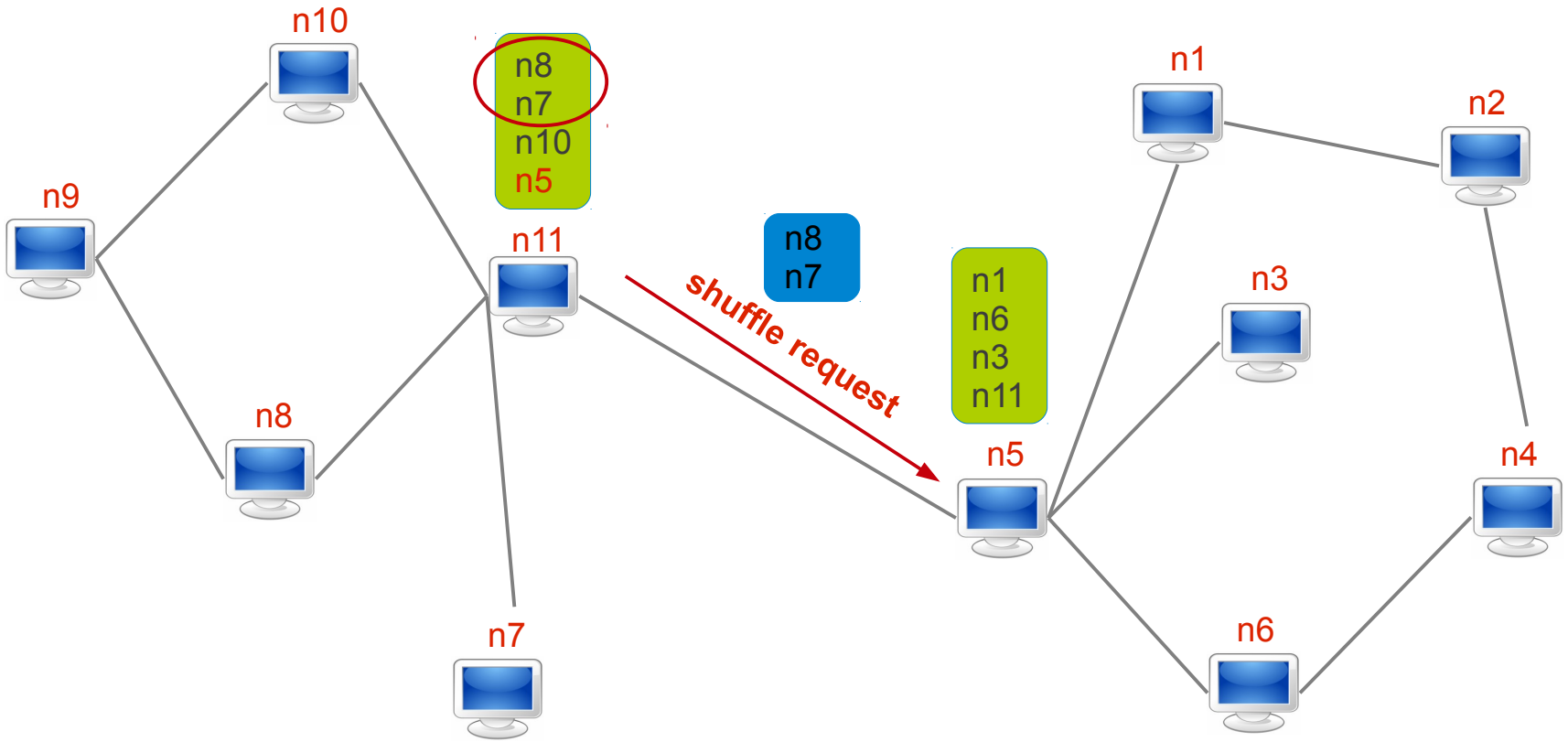
Problem Description (1/10)



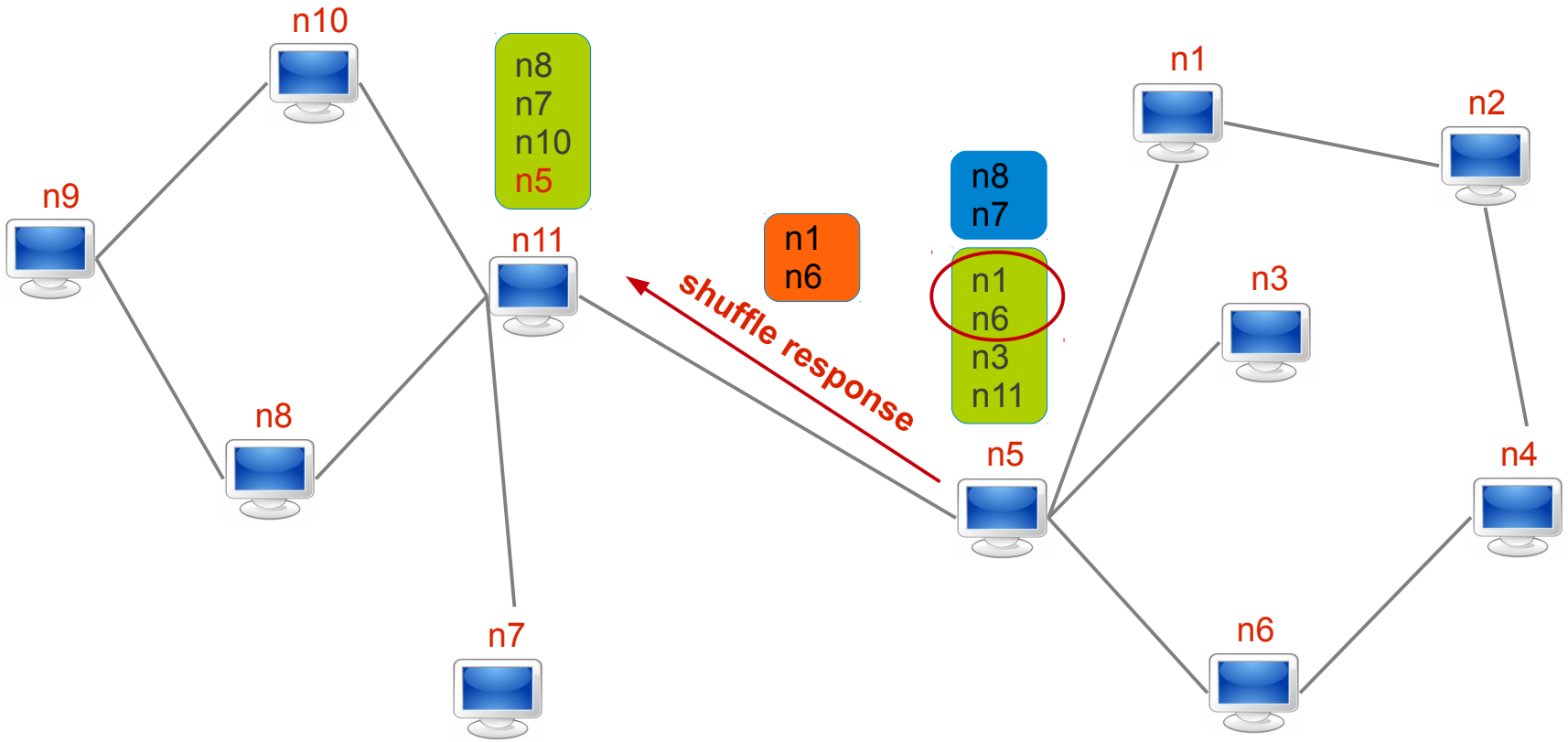
Problem Description (2/10)



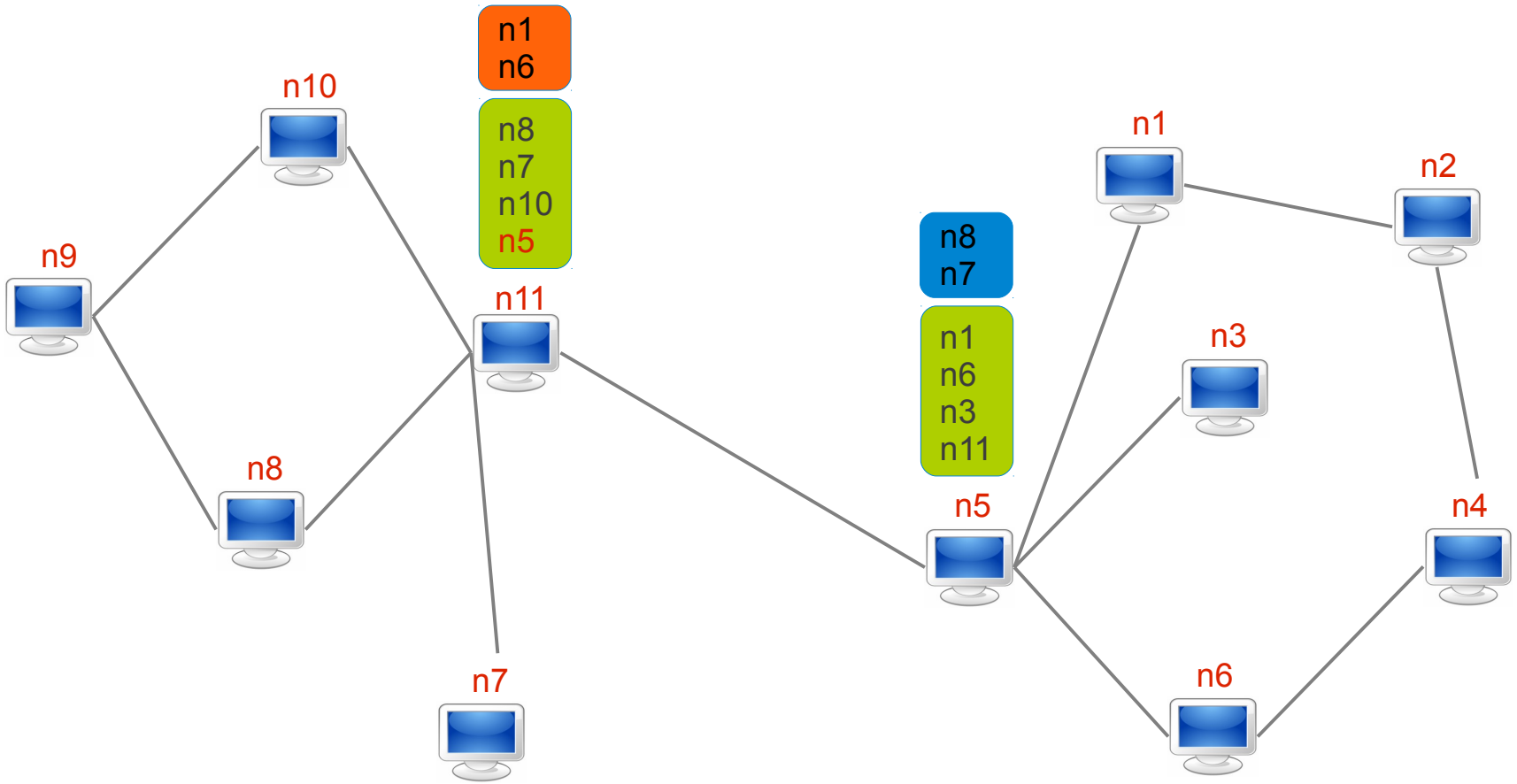
Problem Description (3/10)



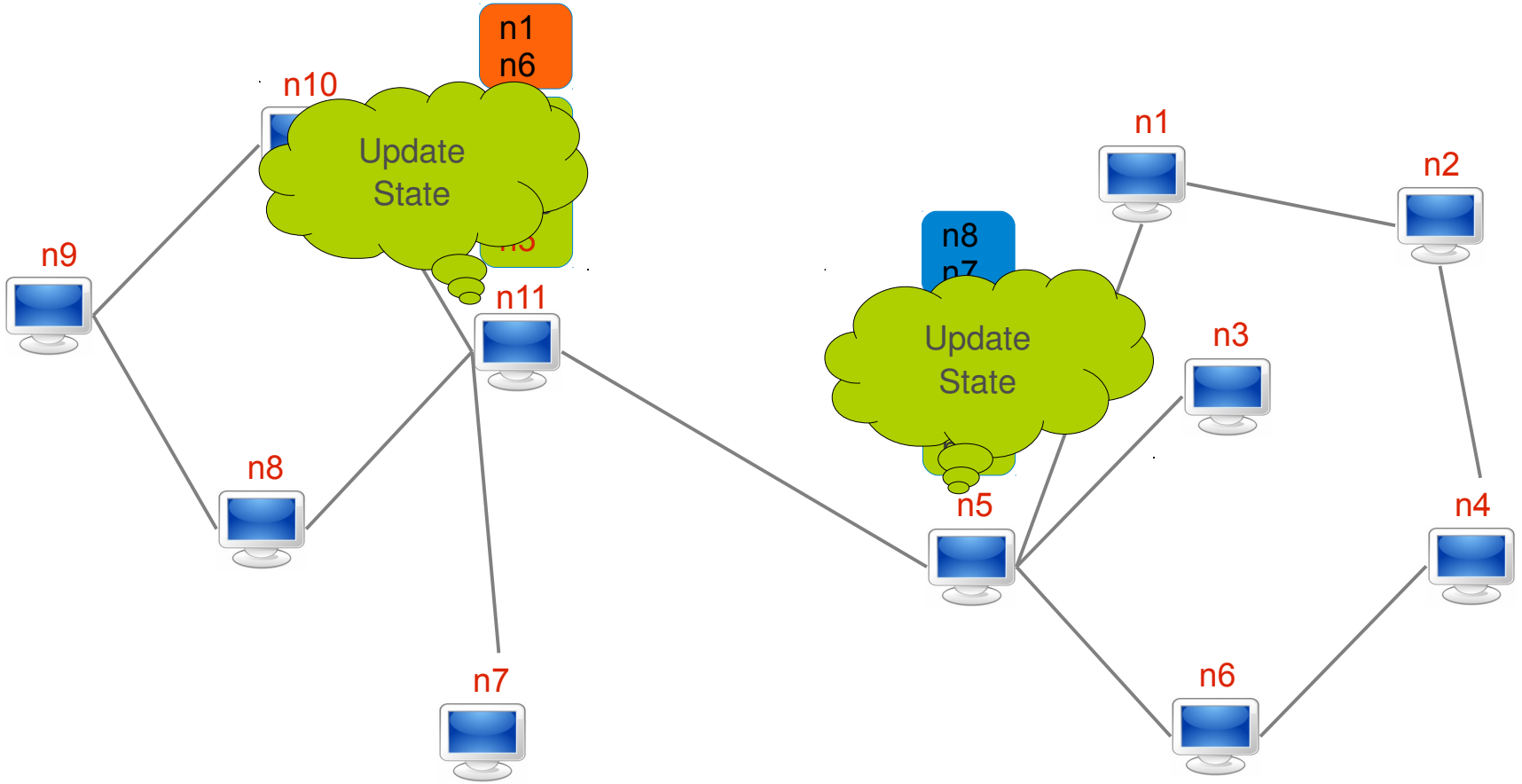
Problem Description (4/10)



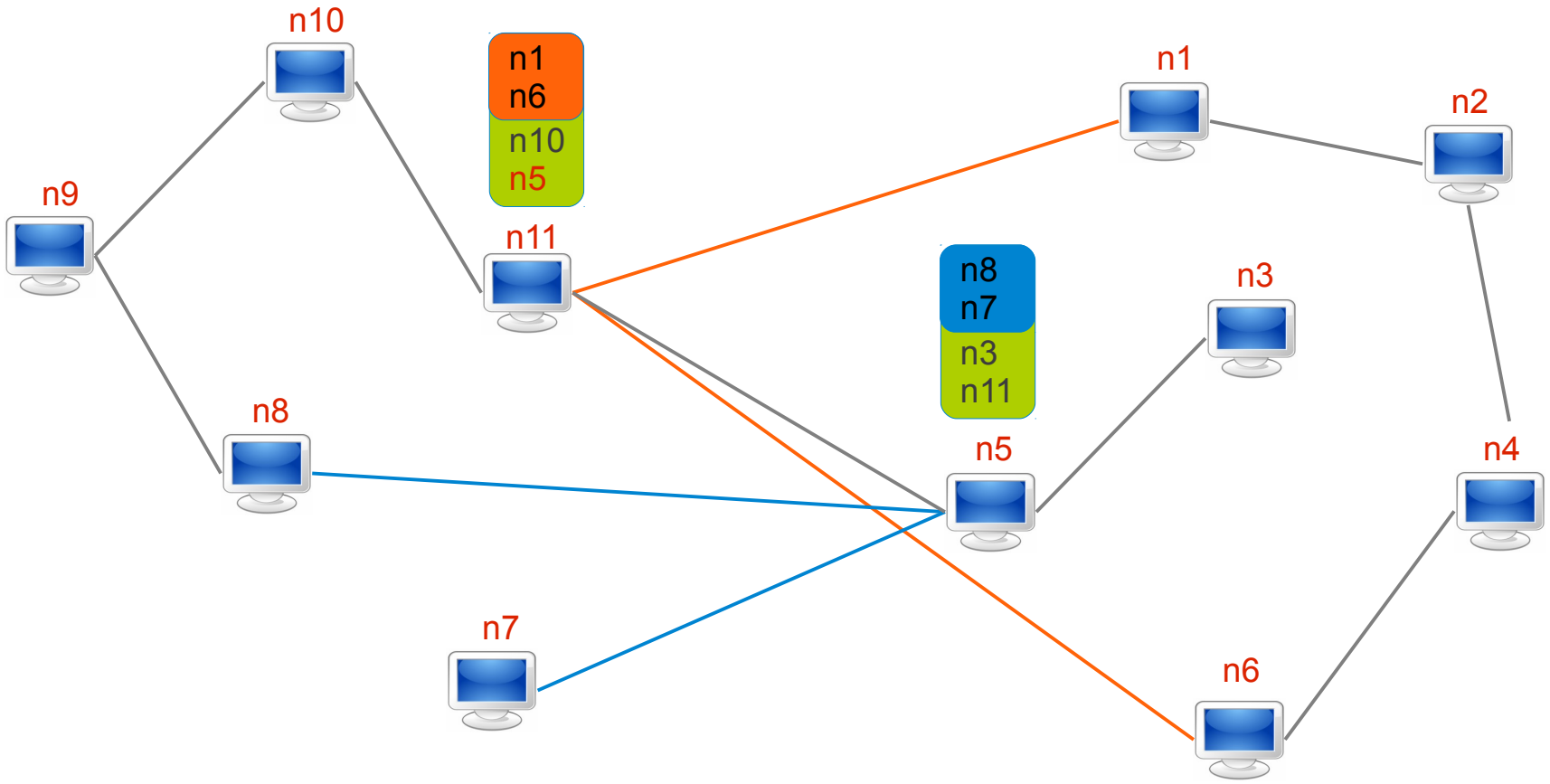
Problem Description (5/10)



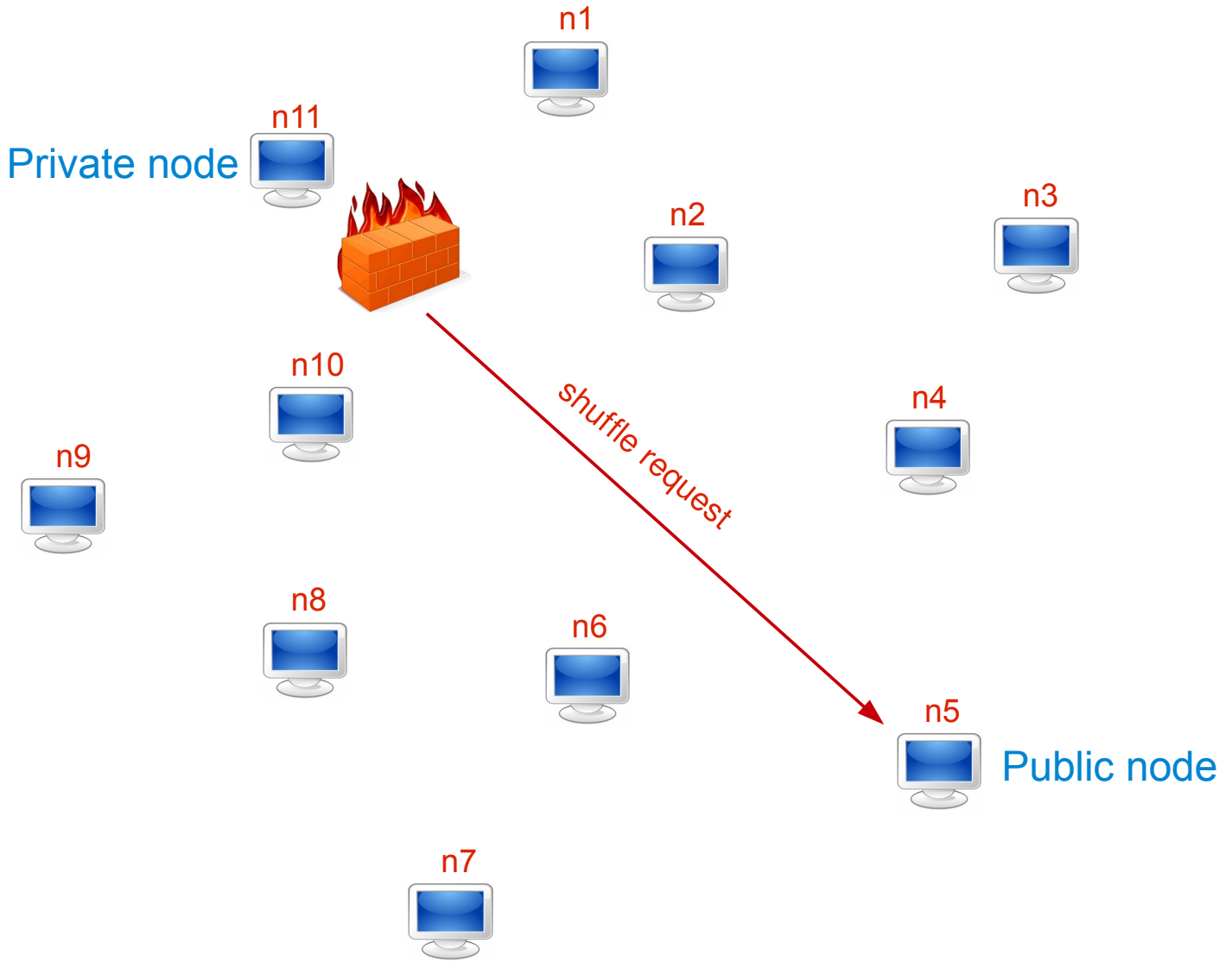
Problem Description (6/10)



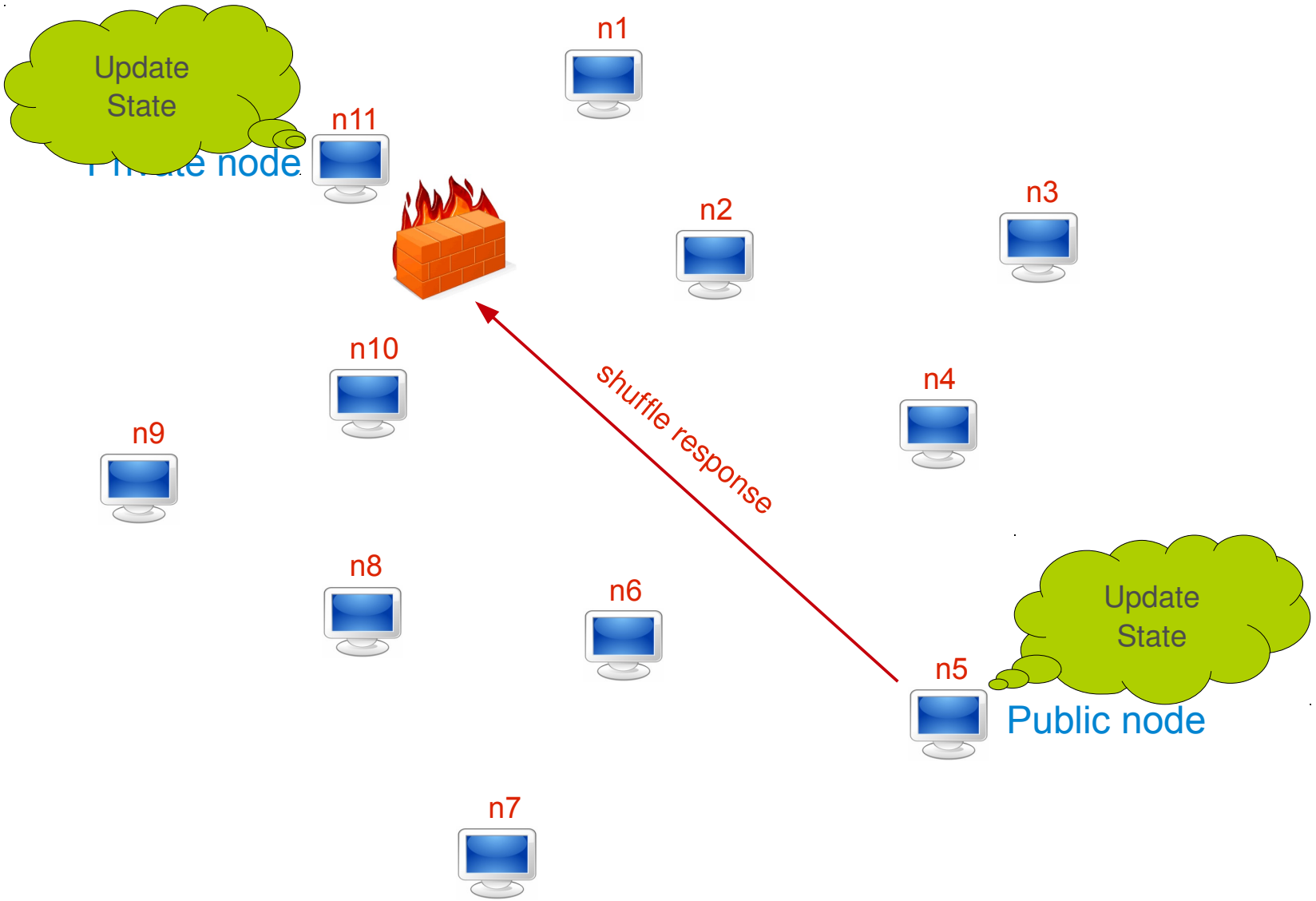
Problem Description (7/10)



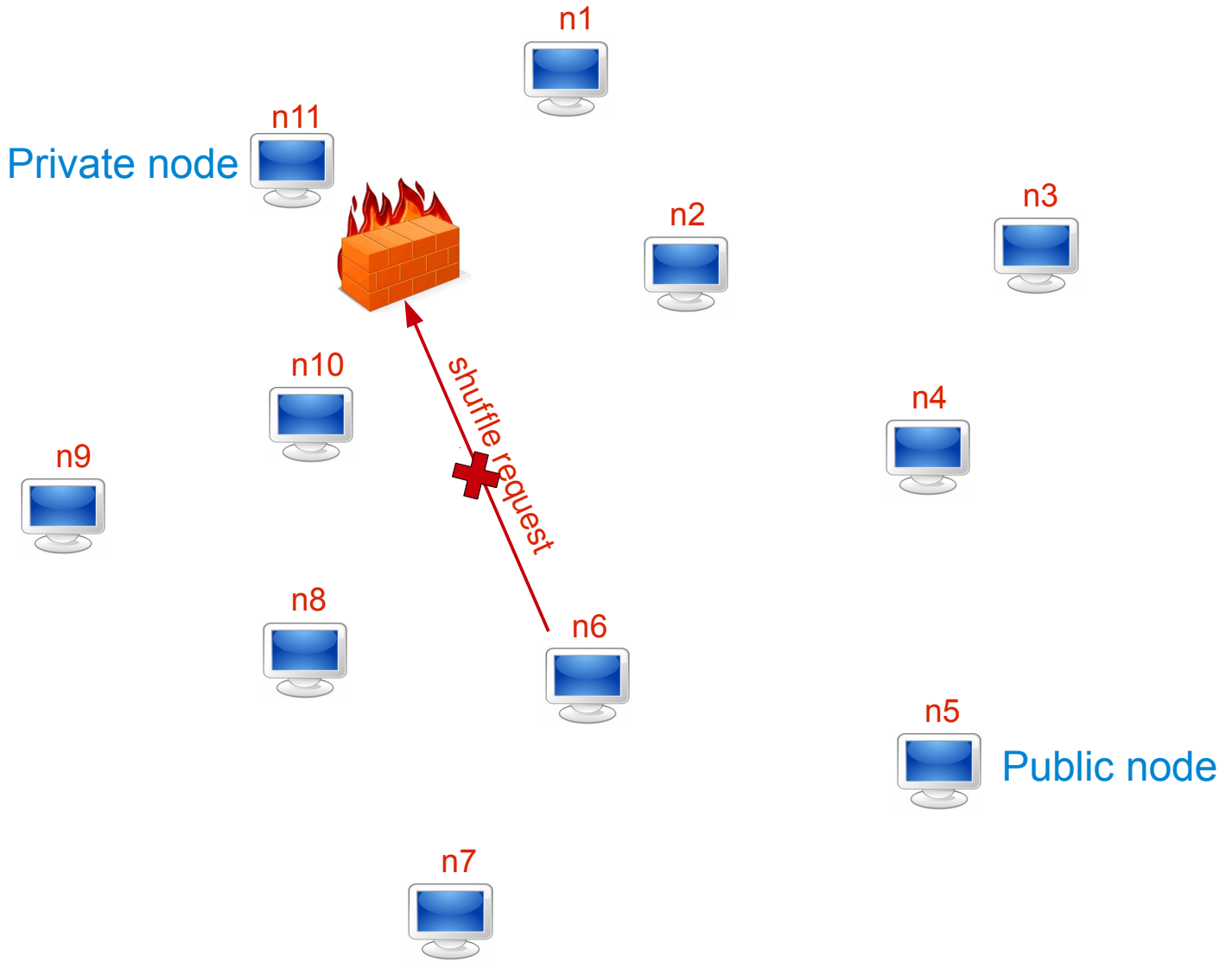
Problem Description (8/10)



Problem Description (9/10)



Problem Description (10/10)



Solution

PSS Design Space

- Node selection
  - Random
  - Tail
- View propagation
  - Push
  - Push-Pull
- View Selection
  - Blind
  - Healer
  - Swapper

Gozar vs. Croupier

- Node selection
  - Random
  - Tail
- View propagation
  - Push
  - Push-Pull
- View Selection
  - Blind
  - Healer
  - Swapper

Gozar

Tail, Push-Pull, Swapper
One hop PSS
Built-in NAT Traversal

Gozar vs. Croupier

- Node selection
  - Random
  - Tail
- View propagation
  - Push
  - Push-Pull
- View Selection
  - Blind
  - Healer
  - Swapper

Gozar

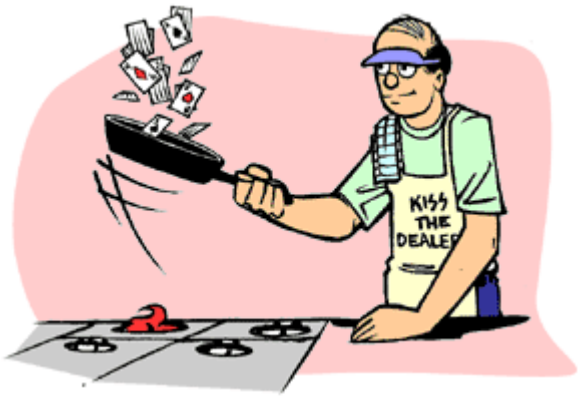
Tail, Push-Pull, Swapper
One hop PSS
Built-in NAT Traversal

Croupier

Tail, Push-Pull, Swapper
PSS without Relaying
NAT Traversal can be added

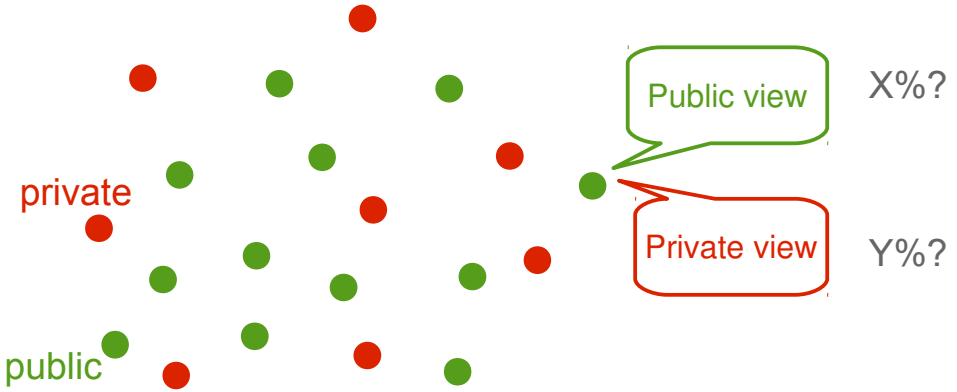
Croupier as a PSS (1/4)

- Private nodes vs. Public nodes
- All nodes only exchange their views with only public nodes.
- Public nodes, as croupiers and on be half of the private nodes, update their views.



Croupier as a PSS (2/4)

- Uniform random selection
- Two views at each node:
  - Public view
  - Private view

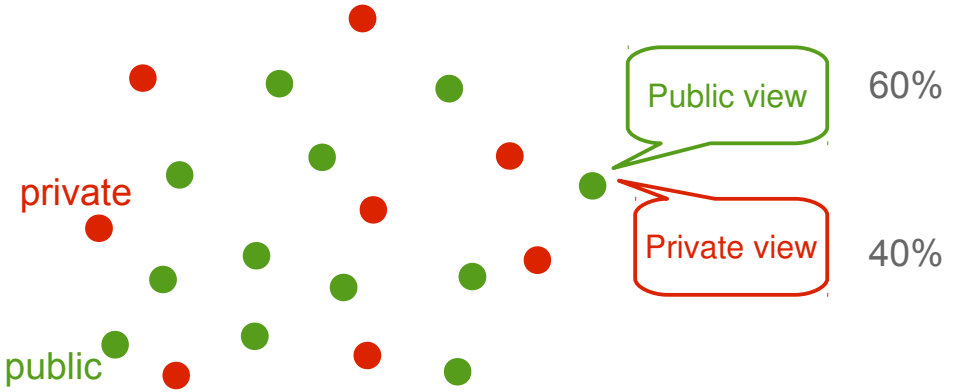


8 red nodes, and 12 green nodes

- How to generate a random sample from from the public/private views?

Croupier as a PSS (3/4)

- Uniform random selection
- Two views at each node:
  - Public view
  - Private view



8 red nodes, and 12 green nodes

- How to generate a random sample from from the public/private views?
  - Estimating the ratio of public to private nodes
  - Gossip-based aggregation

Croupier as a PSS (4/4)

- Public nodes:
  - Counting the number of received shuffle requests in each round from public and private nodes.
  - Keeping track of the γ recent received estimation from public nodes.
- Private nodes:
  - Keeping track of the γ recent received estimation from public nodes.

Croupier as a NAT Traversal Middleware

- Each private node connects to one or more public nodes, called parents.
- A node's descriptor consists of its own address, its NAT type, and its parents addresses.
- Communicate with a private node through its parent.

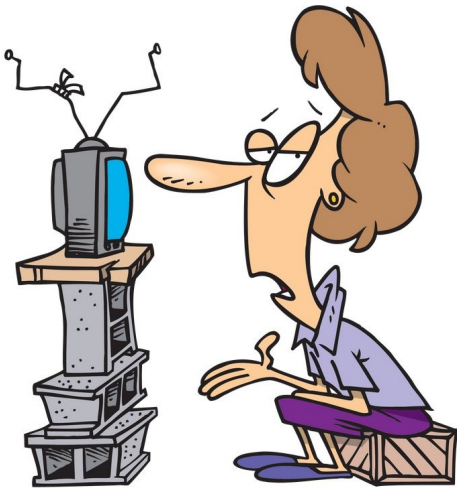
Gozar/Croupier Summary

- One-hop relying vs. without relaying PSS.
- Public nodes behave as croupiers.
- Two views at each node: public and private views.
- Ratio of public/private nodes.
- Partnering private nodes with public nodes for NAT traversal.

Wrap Up

Summary

- Sepidar and GLive
  - P2P solution for live streaming
  - Distributed market model
  - Gradient overlay
  - Transitive auditing
- CLive
  - Hybrid P2P-Cloud
  - Renting AH/PH from a cloud
- Gozar and Croupier
  - NAT-aware PSS with one-hop relaying and without relaying
  - Distributed NAT traversal



Future Work

- Handling the collusion attack.

- Putting all the components together and building a real system.

Thank You :)