Shuffling with a Croupier: Nat-Aware Peer-Sampling

Jim Dowling – Amir H. Payberah
{jdowling,amir}@sics.se
Introduction
Gossip-based Protocols

- Gossip-based protocols have been widely used in large scale distributed applications.
  - Information dissemination
  - Aggregation
  - Overlay topology management
Why Peer Sampling?

- In a gossip-based protocol, each node periodically exchanges information with a random peer.

- Ideally, the peers should be selected uniformly at random.

- If a node could maintain a complete view, then uniform random selection would be easy, but this is not scalable.

- If each node has a small view, how can we achieve uniform randomness? peer sampling
Gossip-based Peer Sampling Protocol (1/7)
Gossip-based Peer Sampling Protocol (2/7)

Shuffling with a Croupier: Nat-Aware Peer-Sampling
Gossip-based Peer Sampling Protocol (3/7)
Gossip-based Peer Sampling Protocol (4/7)

Shuffling with a Croupier: Nat-Aware Peer-Sampling
Gossip-based Peer Sampling Protocol (6/7)
Gossip-based Peer Sampling Protocol (7/7)
Problem Description
NAT Environments (2/4)

Private node

Public node

shuffle response
NAT Environments (3/4)

Shuffling with a Croupier: Nat-Aware Peer-Sampling

June 19, 2012
NAT Environments (4/4)

Private node

Public node

n1
n2
n3
n4
n5
n6
n7
n8
n9
n10
n11

Shuffle request

n1

n2
Impact of NATs on PSS' (1/2)

- Size of the biggest cluster for an increasing percentage of NATs.

[A.M. Kermarrec – ICDCS'09]
Impact of NATs on PSS' (2/2)

- Percentage of stale references.

[A.M.Kermarrec – ICDCS’09]
How to Deal With This?
Solutions for Communicating with Private Nodes (1/2)

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

![Diagram of a public relay node facilitating communication between two private nodes.](image)
Solutions for Communicating with Private Nodes (2/2)

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

- Existing gossip-based NAT-aware Peer Sampling Services' (PSS) are similar to classic PSS:
  - Single partial view.
  - Periodically exchange partial views with random nodes.

- But, if the selected node is a **private node**
  - First, the *relay node for that private node* is discovered.
  - Then a view exchange is done *through the relay node.*
Problems of the Existing Solutions

- Nodes have to **discover** relay nodes.
- Private nodes have to **maintain open mapping** in their NAT.
- Relaying nodes have to **maintain routing tables**.
Croupier
The Croupier Protocol

- A NAT-aware gossip-based PSS **without the use of relaying or hole-punching**.

- **Public nodes** are *croupiers*.

- Each node keeps **two views**:  
  - Public view  
  - Private view
Croupier in a Nutshell

- Continuously update the nodes' public/private views.
- Estimate the ratio of the public nodes in the system, and take a uniform sample based on the estimated ratio.
Croupier in a Nutshell

- Continuously update the nodes' public/private views.
- Estimate the ratio of the public nodes in the system, and take a uniform sample based on the estimated ratio.
Updating the Views

- Each public/private node periodically sends a *shuffle request* to a *public node*, chosen from its public view.
  - Chooses the *oldest node* in the view.
  - Sends a subset of its *public/private views*.
Updating the Views

- Each public/private node periodically sends a shuffle request to a public node, chosen from its public view.
  - Chooses the oldest node in the view.
  - Sends a subset of its public/private views.

- The receiver node (public) sends back a shuffle response with a subset of its public/private views.
Updating the Views

- Each public/private node periodically sends a shuffle request to a public node, chosen from its public view.
  - Chooses the oldest node in the view.
  - Sends a subset of its public/private views.

- The receiver node (public) sends back a shuffle response with a subset of its public/private views.

- Sender and receiver both update their public/private views:
  - By first merging the views, and then if the view size exceeds its upper-bound, replacing the sent nodes with the received nodes.
Croupier in a Nutshell

- Continuously update the nodes' public/private views.
- Estimate the ratio of the public nodes in the system, and take a uniform sample based on the estimated ratio.
Providing Uniform Sample at Nodes

- Each node estimates the ratio of the public nodes in the network: $E_i(\omega)$

- They sample the nodes from the public/private views proportional to the estimated ratio.
  - For example, if $E_i(\omega) = 20\%$, and the public/private view sizes are 10, then a node samples the nodes by taking 2 nodes from the public view and 8 nodes from the private view.
Ratio Estimation at Public Nodes

- The public nodes counts the number of received shuffle requests from public and private nodes at each round.
  - Public nodes: $c_u$
  - Private nodes: $c_v$
Ratio Estimation at Public Nodes

- The **public nodes** counts the number of received shuffle requests from public and private nodes at each round.
  - Public nodes: $c_u$
  - Private nodes: $c_v$

- They sum up the received hits in the last $\alpha$ rounds.

\[
C_{ui} = \sum_{t=0}^{\alpha} c_{ui}(t) \quad \quad C_{vi} = \sum_{t=0}^{\alpha} c_{vi}(t)
\]
Ratio Estimation at Public Nodes

- The public nodes counts the number of received shuffle requests from public and private nodes at each round.
  - Public nodes: $c_u$
  - Private nodes: $c_v$

- They sum up the received hits in the last $\alpha$ rounds:
  \[
  C_{ui} = \sum_{t=0}^{\alpha} c_{ui}(t) \quad C_{vi} = \sum_{t=0}^{\alpha} c_{vi}(t)
  \]

- Then, calculate the ratio of the public nodes for the last $\alpha$ rounds:
  \[
  E_i = \frac{C_{ui}}{C_{ui} + C_{vi}}
  \]
Ratio Estimation at Public Nodes

- The public nodes counts the number of received shuffle requests from public and private nodes at each round.
  - Public nodes: $c_u$
  - Private nodes: $c_v$

- They sum up the received hits in the last $\alpha$ rounds:

$$C_{ui} = \sum_{t=0}^{\alpha} c_{ui}(t)$$

$$C_{vi} = \sum_{t=0}^{\alpha} c_{vi}(t)$$

- Then, calculate the ratio of the public nodes for the last $\alpha$ rounds:

$$E_i = \frac{C_{ui}}{C_{ui} + C_{vi}}$$

Not accurate enough
Ratio Estimation at Public Nodes

- The **public nodes** piggyback their estimated ratio $E_i$ in each **shuffle response**.

- Any node $i$ keeps track of the $\gamma$ recent received estimation from public nodes in a local list: $M_i$

- Then, the each public node measures the ratio of the public nodes in the system:

$$E_i(\omega) = \frac{\sum_{n \in M_i} E_n + E_i}{|M_i| + 1}$$
Ratio Estimation at **Private Nodes**

- The **private nodes** do not receive any shuffle request. So, they cannot estimate $E_i$ for the last $\alpha$ rounds.

- But, they receive the **public nodes estimation** in shuffle responses, and keep the $\gamma$ recent received estimation at $M_i$.

- So, they measure the ratio of the public nodes as follows:

$$E_i(\omega) = \frac{\sum_{n \in M_i} E_n}{|M_i|}$$
Experiments
Experiment Setup

- We used Kompics as a simulator platform.

- The public/private views sizes are 10, and the shuffling period is one second.

- 5000 nodes, 80% of nodes are private and 20% are public.

- Compared with Gozar and Nylon.
  - Gozar uses a single rendezvous node for relaying.
  - Nylon uses a chain of nodes to enable direct communication between nodes by the use of hole punching.

- Cyclon is used as a baseline.
Metrics

- **Correctness of the estimation** in static and dynamic network.

- **Randomness** properties.

- Protocol **overhead**.

- **Fairness** and **connectivity** in catastrophic failure.
Ratio Estimation
Randomness

(a) In-degree distribution.

(b) Avg. path length.

(c) Clustering co-efficient.
Protocol Overhead
Connectivity in Failure
Conclusions
Conclusions

- **Croupier** is a NAT-friendly gossip-based peer sampling *without the use of relaying*.

- Shuffle requests are **sent only to the public nodes**, but all the nodes receive both shuffle responses.

- Each node keeps **two views** for public nodes and private nodes.

- The nodes estimate the ratio of public nodes, and use it to take a uniform sample of all the nodes in the system.
Questions?