Sampling in a Large Network

Xenofon Foukas

Imperial College London

Introduction

- Beneficial to architect networks and overlays as fully decentralized systems
  - Reliable
  - Fault-tolerant
  - No node has control over the whole network

- However computing global properties becomes a challenging task
  - Information distributed amongst nodes
  - Size and topology of the network are unknown and could change often altering the network properties

- Many distributed algorithms exist
  - Gossip
  - Random Walks
  - Spectral

- Spectral algorithms
  - They provide information about structural properties
  - Derived from the spectrum of the network
  - Information provided are often used for parameterizing other algorithms in P2P networks

Background

Decentralized algorithm by Carzaniga et al. for estimating spectral network properties in low diameter networks, e.g. P2P

- Estimates the most significant eigenvalues of a descriptive matrix closely related to the adjacency matrix of the network graph

- Views the network as a linear dynamic system of the form

$$x(t+1) = Ax(t)+Bu(t)$$
$$y(t) = Cx(t)$$

- The system reacts to an input (impulse) and produces an impulse response

- By gathering enough impulse responses we can approximately identify matrices A, B, C of the system by using Kung's realization algorithm

- We are interested in matrix A

Background

Algorithm 1 estimation algorithm executing at node v

$x_v \leftarrow$ choose a value from $\{0, 1\}$ uniformly
$h_v(1) \leftarrow x_v$
for $t \leftarrow 2 \ldots k$ **do**
    for $u \in$ out-neighbors(v) **do**
        send value $x_v a_{uv}$ to u
    end for
    collect all values w sent by in-neighbors
    $x_v \leftarrow \sum w$
    $h_v(t) \leftarrow x_v$
end for
$\hat{A}_v$ = Kung's realization with $h_v(1), \ldots, h_v(k)$
compute the dominant eigenvalues of $\hat{A}_v$
exchange the eigenvalues with neighbors
collect estimates from neighbors
adjust estimates to the median of the collected estimates

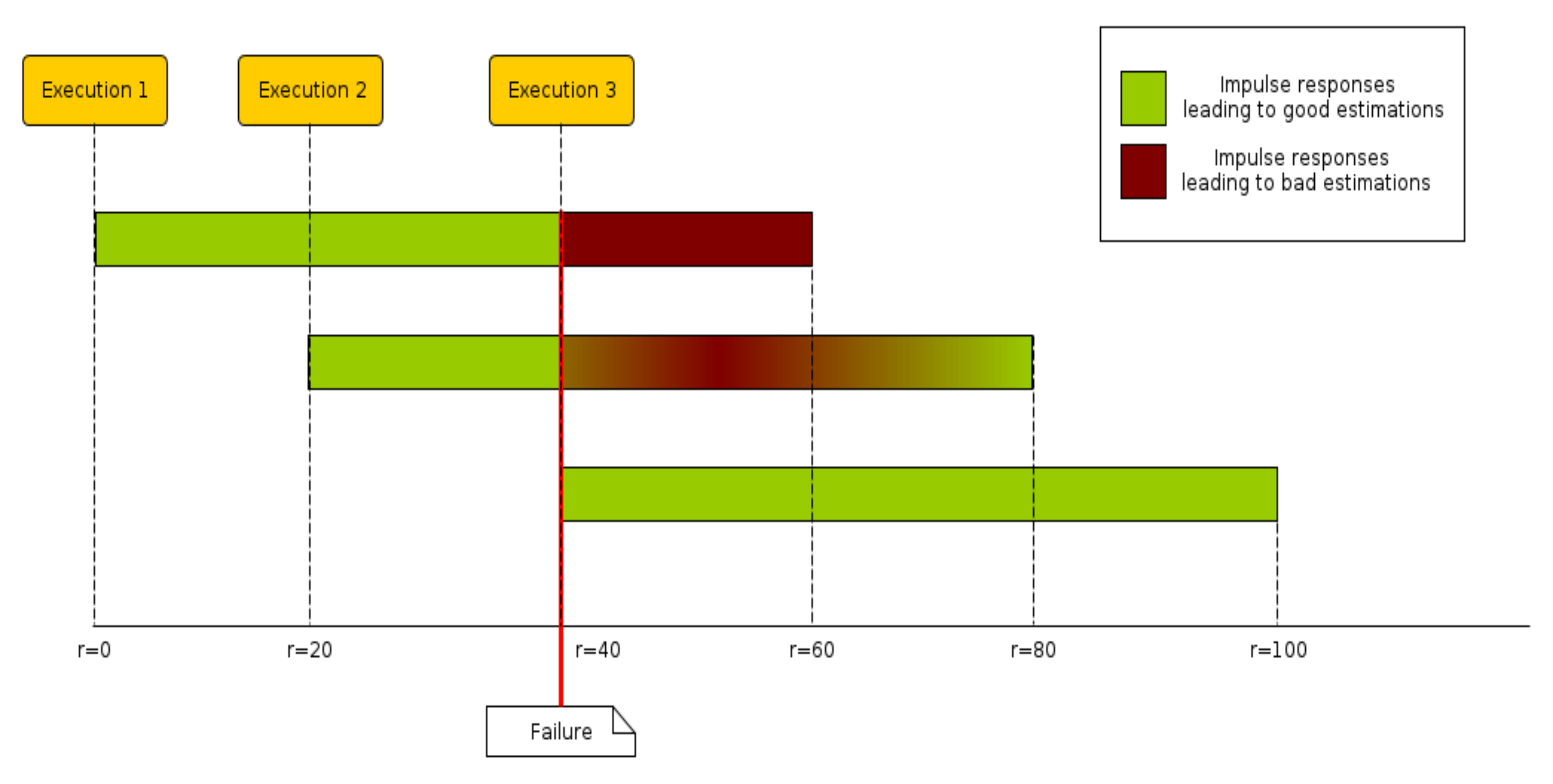
- High error even with a single failure
- Needs to operate in an asynchronous environment
- Integration with underlying network is challenging
- Portability issues
    - Discovery of in-neighbors
    - In-neighbors might fail

Protocol Design

- Two basic structures
  - **Execution**: Part of a protocol run performing the tasks of the original algorithm
  - **Session**: A full protocol run containing at least one but possibly multiple parallel partially overlapping Executions
- Each Execution has three phases
  - **Initiation**: Send messages in a controlled flooding manner to inform nodes of the new Execution. Discover in-neighbors through the messages received
  - **Data Exchange**: Exchange impulses and compute impulse responses for a number of rounds. In the final round, estimate eigenvalues
  - **Gossip Round**: Exchange eigenvalue estimations with out-neighbors. Compute final estimation as the median of all gathered estimations
- A Session terminates once all its Executions terminate
  - The final estimation of the Session is the median of the estimations given by all Executions

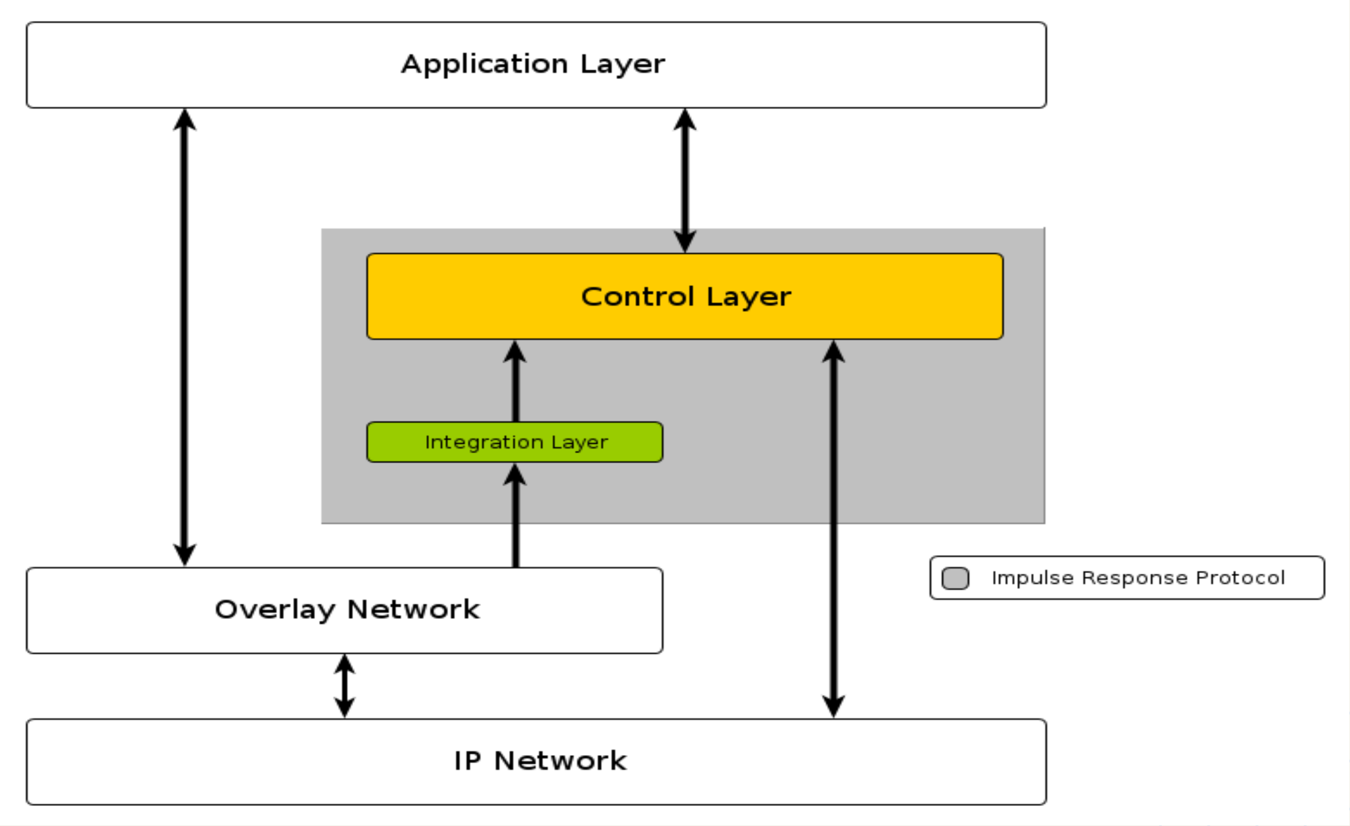
Protocol Design

- Combining parallel partially overlapping Executions can increase the accuracy of estimations



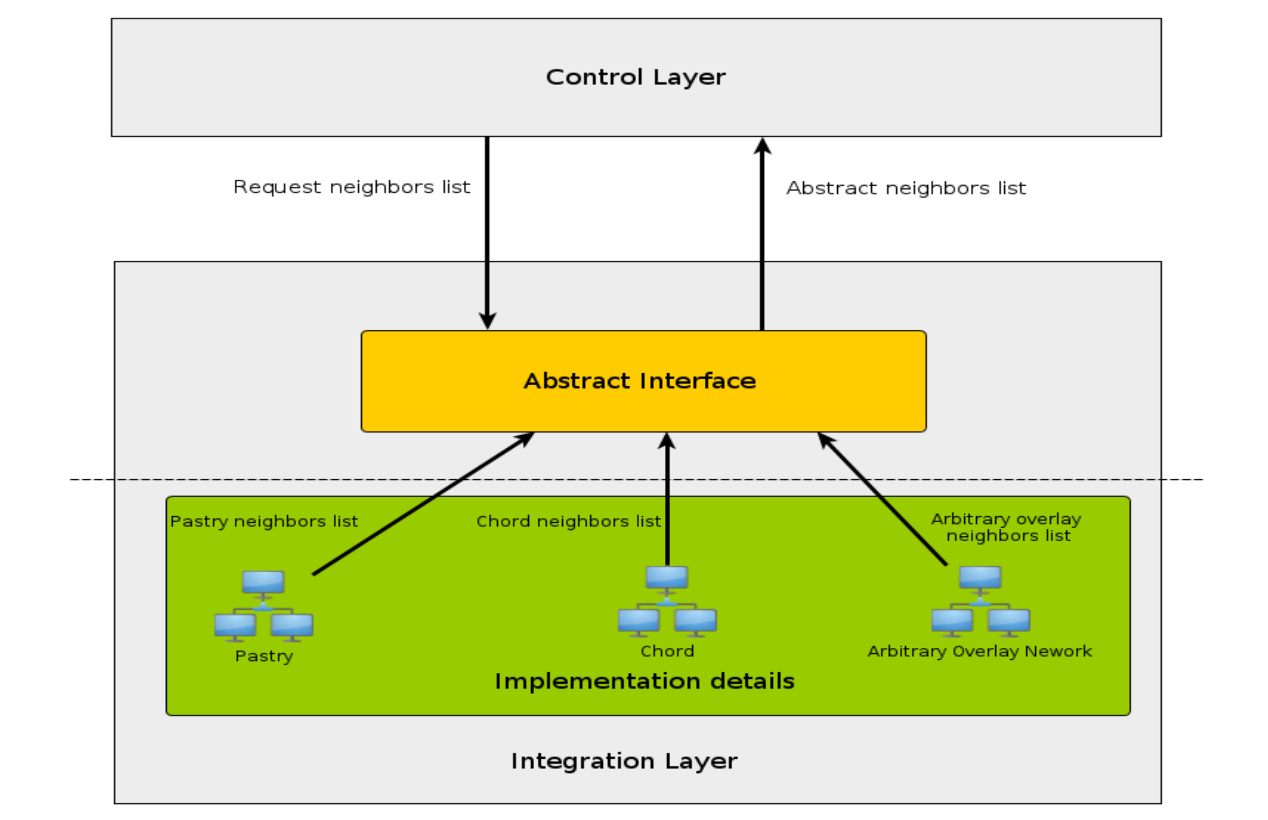
Software Architecture

- Layered architecture



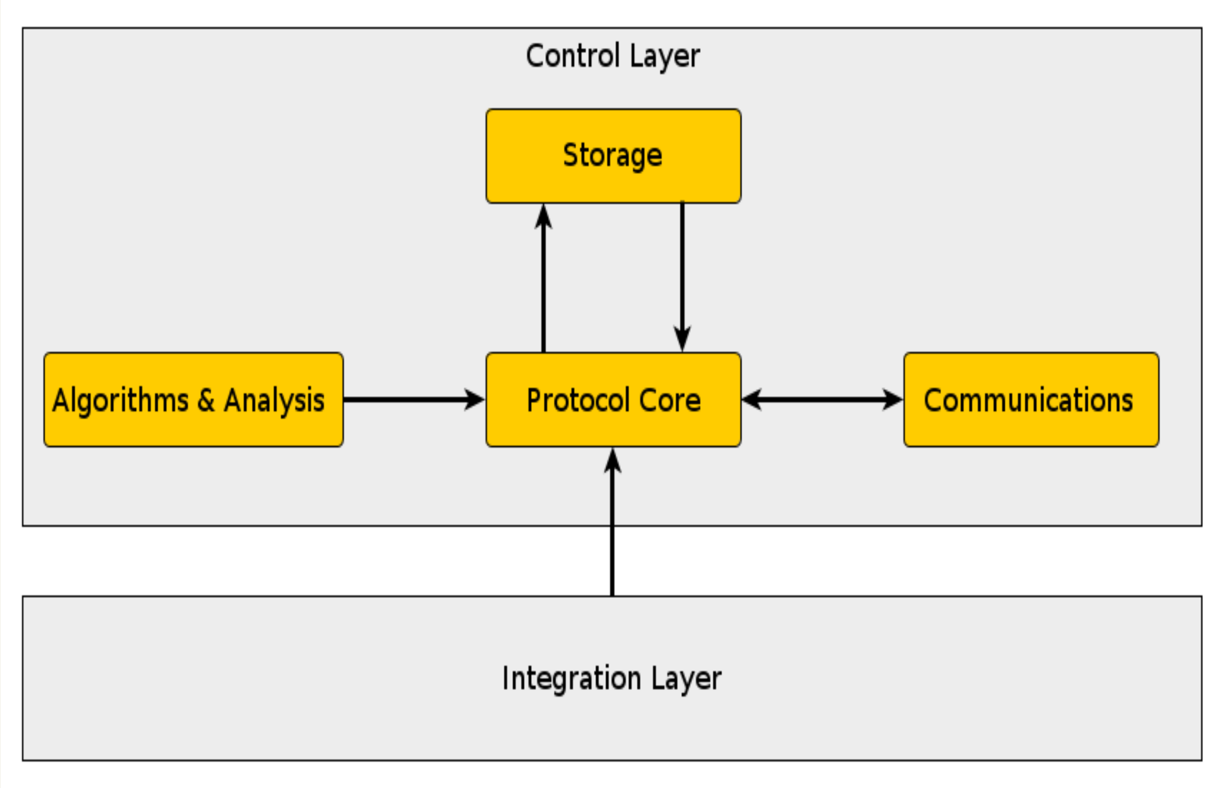
Software Architecture

- The Integration Layer handles network-specific details
  - Currently provided support for Pastry and Chord overlays



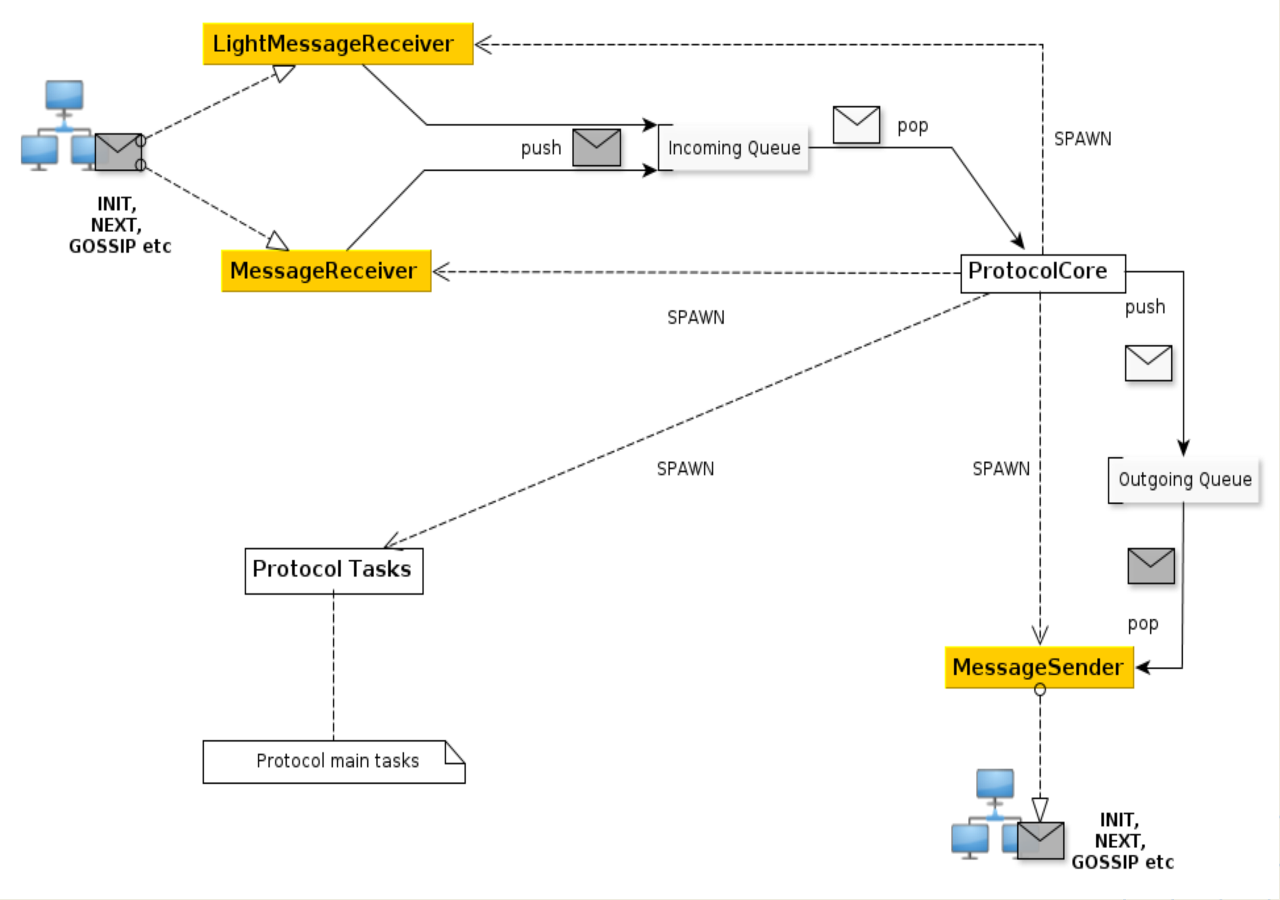
Software Architecture

- The control layer is responsible for high level protocol operations
  - Handling impulse responses, user interaction etc.



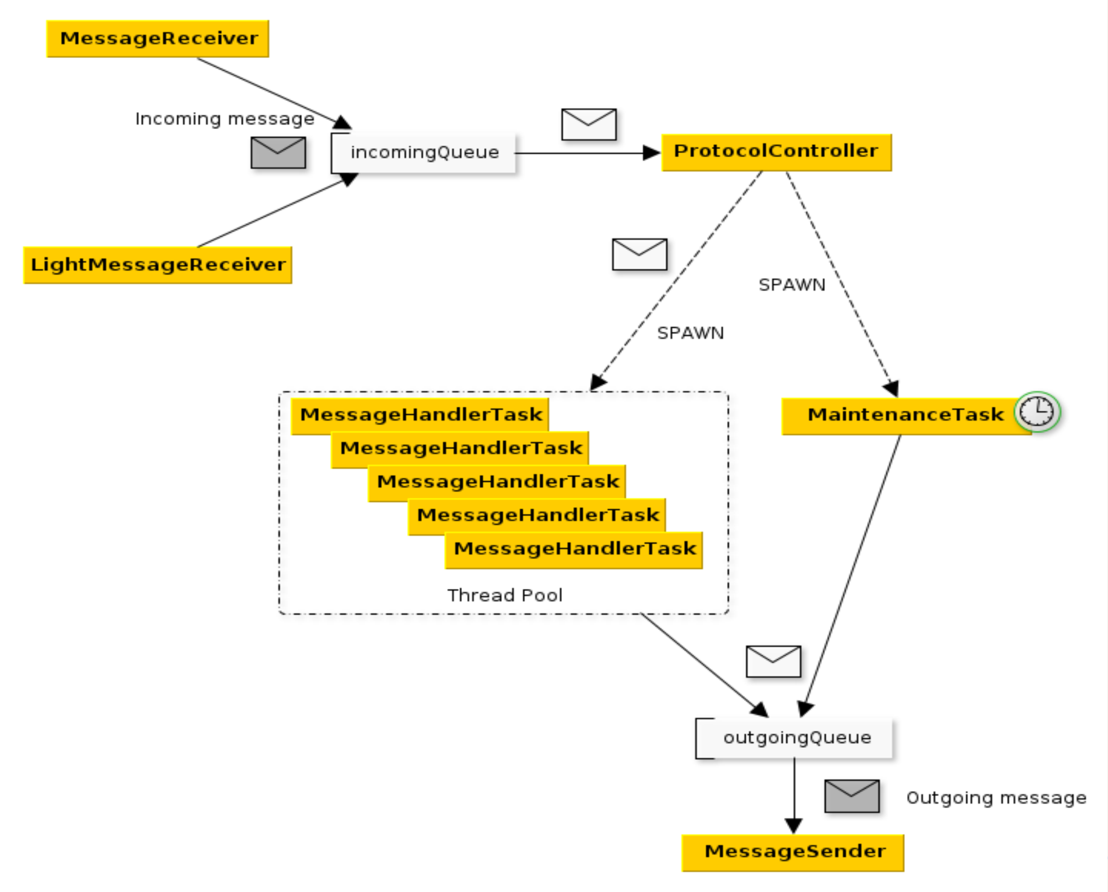
Implementation

- Asynchronous operation of nodes



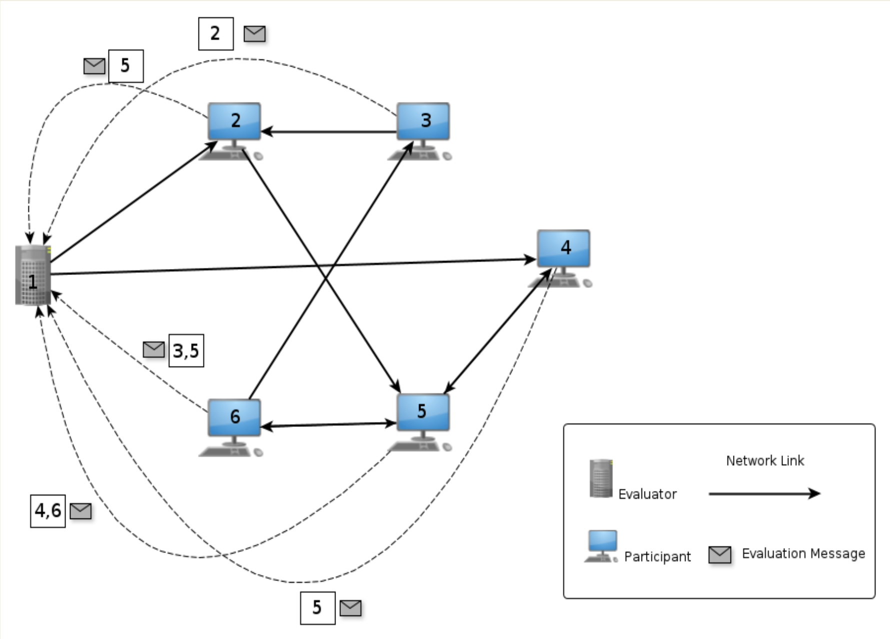
Implementation

- Protocol tasks are executed concurrently



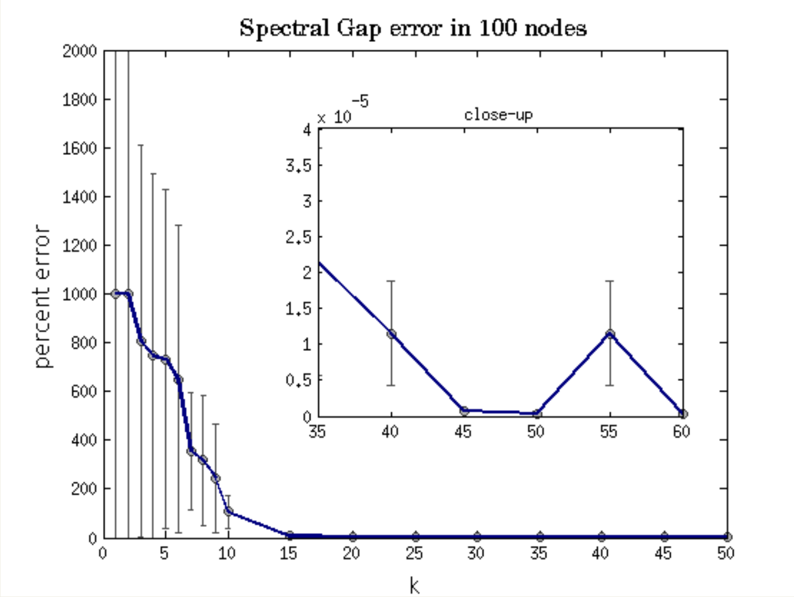
Evaluation

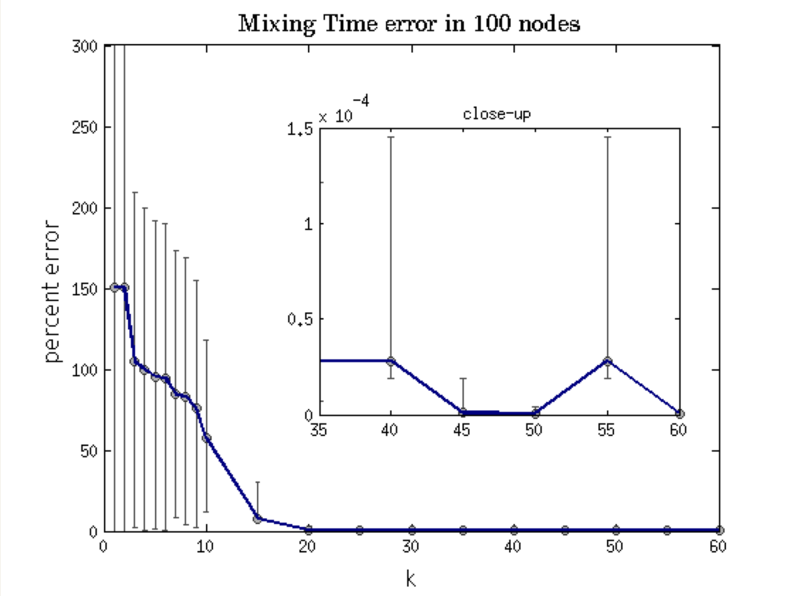
- PlanetLab deployment
  - Pastry overlay (FreePastry)
  - 20, 50 and 100 nodes
- Topological information gathered at evaluator node
  - Adjacency matrix constructed
  - Actual network properties computed
- Evaluated properties
  - Spectral gap
    $|\lambda 1-\lambda 2| = 1-|\lambda 2|$
  - Mixing Time
    $\tau_{mix} = \log_{\lambda 2}\varepsilon$
- Same properties evaluated by Carzaniga et al.
  - Allows verification of correctness in results



Evaluation

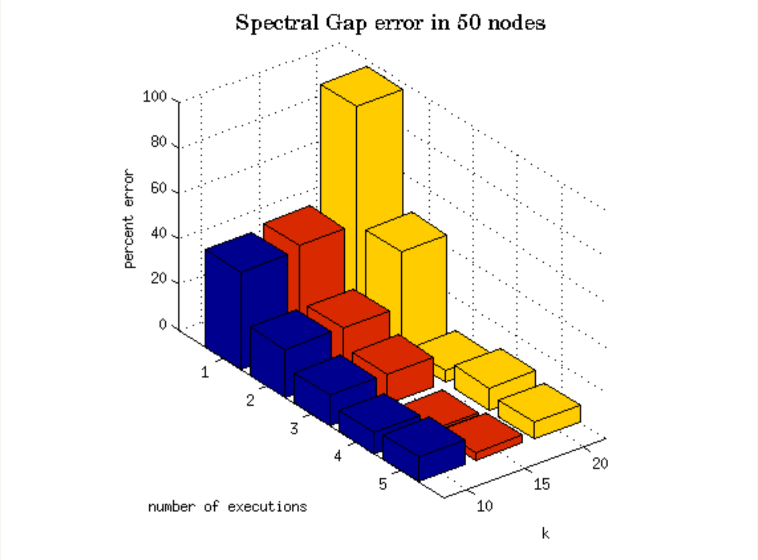
- Networks without churn
  - Error quickly becomes very small
  - It does not stabilize to some value
  - Mixing time requires a few more rounds than spectral gap

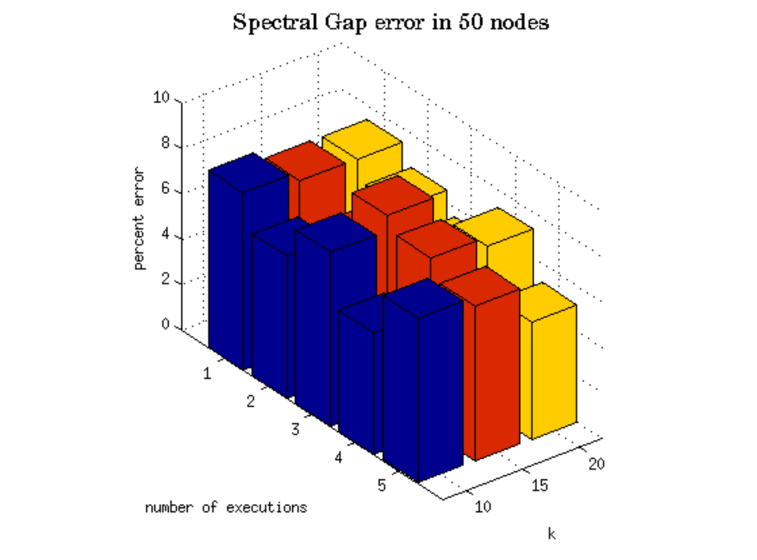




Evaluation

- Single failure resistance
  - Failure introduced at round 3k/4 to have maximum impact
  - Pipelined executions reduce the error
- Single addition resistance
  - Error relatively low regardless of number of executions
  - An execution takes a “snapshot” of the network during initiation
  - Addition completely ignored but properties of the new topology are very close to the old ones



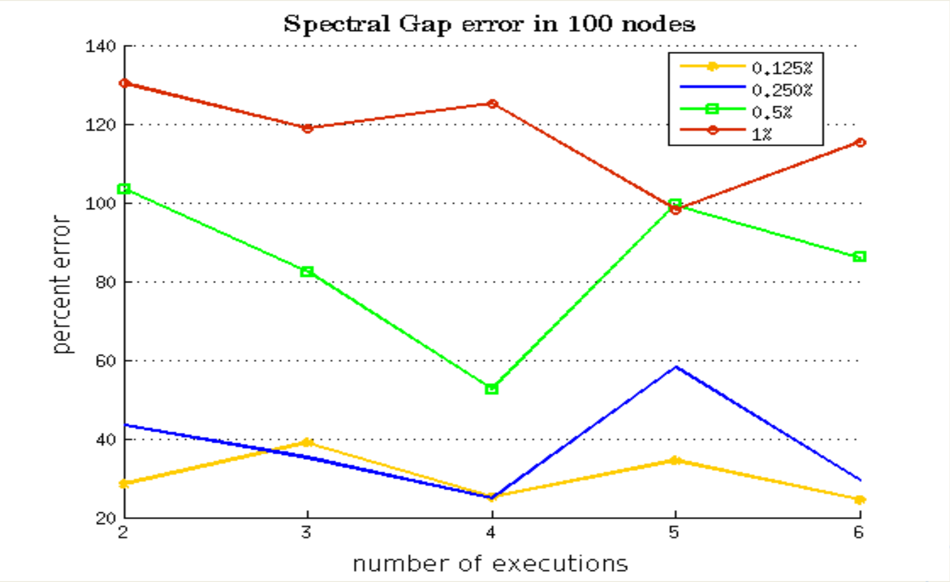


Evaluation

- Resistance of protocol in various failure rates
  - Compute average time for a Session to complete
  - Compute the number of failures to introduce for each failure rate
  - Introduce the failures uniformly in time

- Provides good approximations for low failure rates

- Cannot handle high failure rates



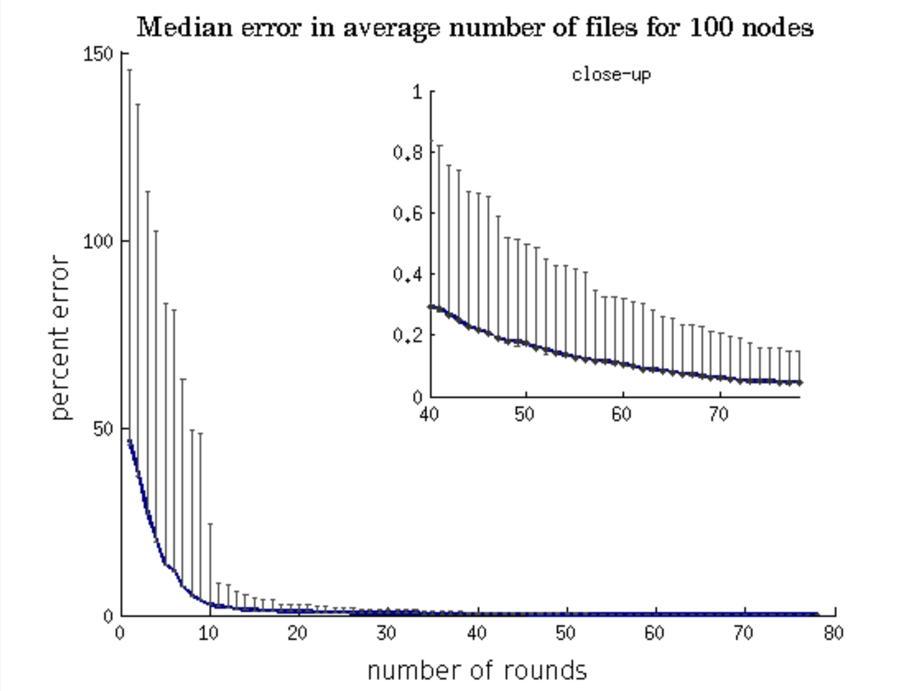
Sampling Application

- Uses Push-Sum to compute average number of files per node
  - Gossip algorithm operating in rounds
  - Each node sends its current estimation to some out-neighbor and collects the estimations of its in-neighbors
  - Algorithm terminates once estimations converge **globally** - additional overhead for distributed convergence

- Complexity of the algorithm is $O(\log N + t_{mix})$
  - Determine number of rounds by using these parameters instead of checking for global convergence
  - t_{mix} given by our protocol
  - size N unknown but in Pastry the size of the routing table is O(logN) - use this value instead

Sampling Application

- Estimations accurate to all the nodes
  - The median error and the maximum error are very close
  - Estimation similar to that of global convergence without exchanging any additional messages



Demonstration

Conclusion

- Decentralized protocol for computing spectral network properties
  - Higher tolerance to failures and churn than original approach
  - Allows easy integration to new overlays
  - Can be used to provide higher level information about network

- Limitations
  - Does not address the problem of malicious nodes
  - Relies on the information provided by the overlay implementation

- Future Work
  - Security concerns
  - Experiments into larger testbeds
  - Virtual nodes currently not supported, but could be extended
  - Create a coordination mechanism for making sampling requests