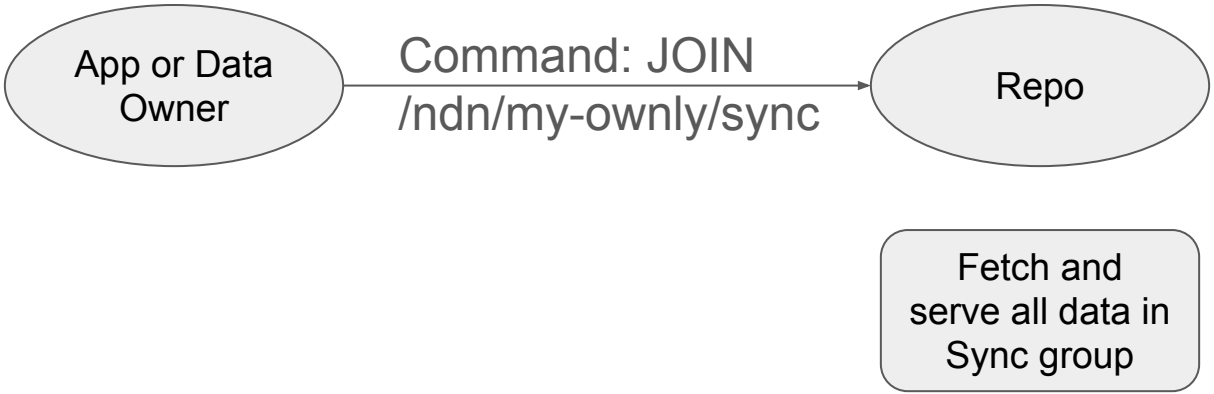
Resilient In-Network Storage in NDN

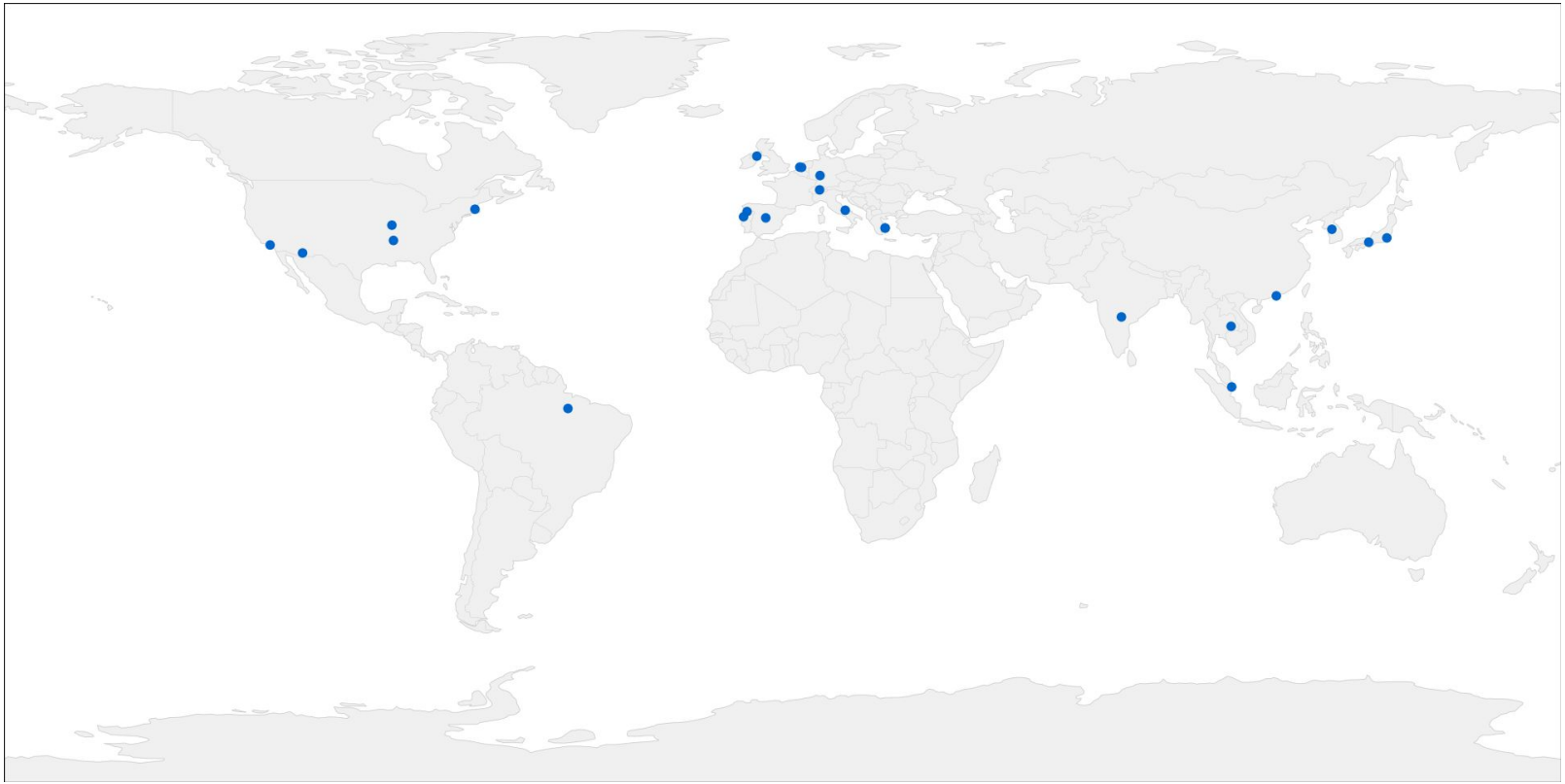
Adam Thieme (UCLA)

Background: NDN Repo

- In-network storage in support of applications
- Provides API for
  - Inserting/Deleting Data
  - Joining/Leaving Sync Groups



Background: NDN Testbed



Distributed Repo Design Goals

1) Each Command is executed by 3 repo nodes
2) Delegation determined by storage availability

Resilient to:

1) Nodes going down
2) Network partitions

Presentation Outline

1) Major Components and Design

2) Introduction of two Distribution Mechanisms

3) Evaluation

Major Components

1) New Command Handling

2) Command Assignment

3) Reassignment after Node Failure

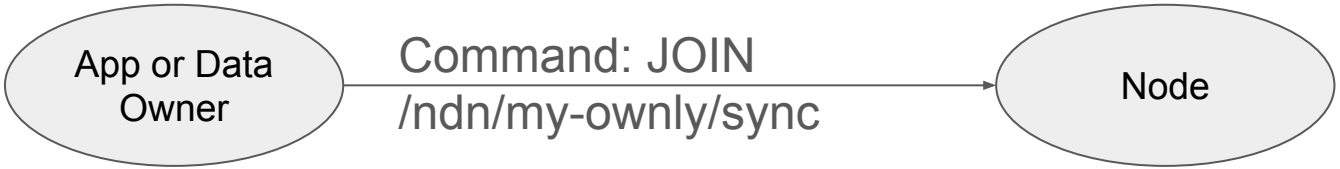
Major Components

1) **New Command Handling**

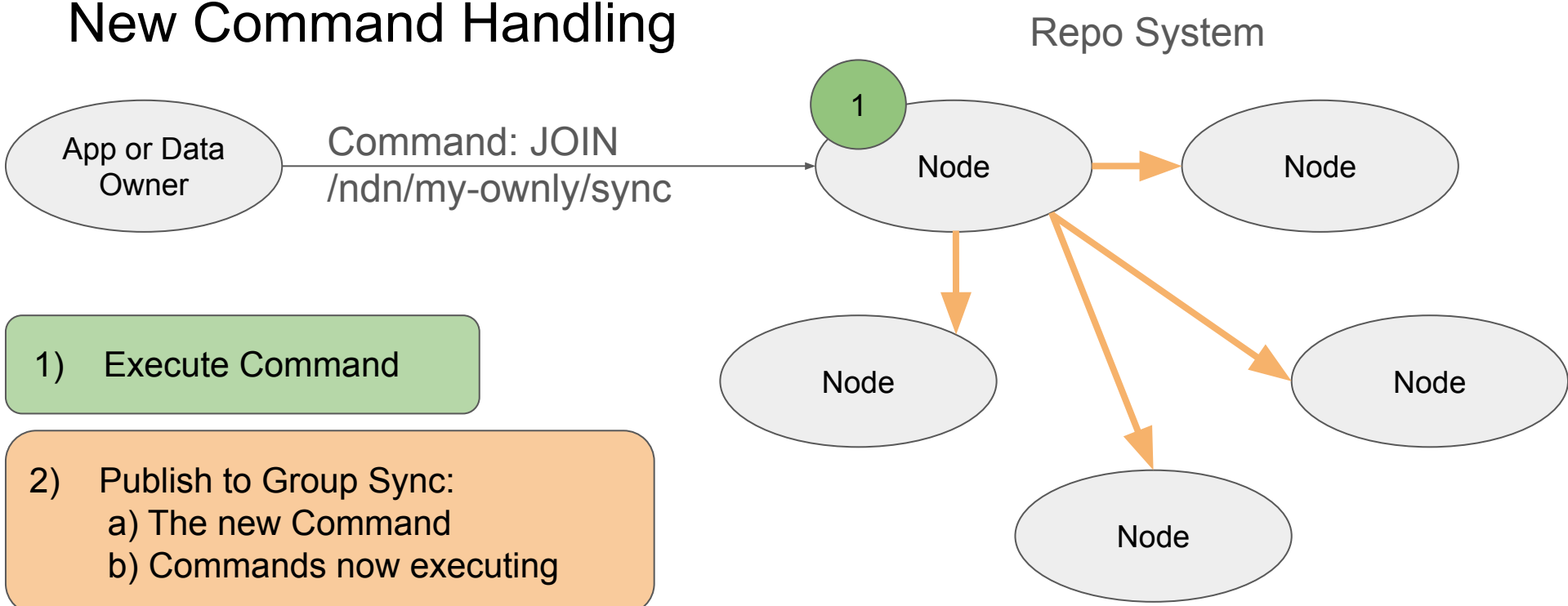
2) Command Assignment

3) Reassignment after Node Failure

New Command Handling



New Command Handling



1) Execute Command

2) Publish to Group Sync:
   a) The new Command
   b) Commands now executing

3) Assign Command to Top 2 Nodes

Major Components

1) **New Command Handling**

2) **Command Assignment**

3) Reassignment after Node Failure

How to Allocate Commands

Ideally:

- Each node has the same storage usage percent

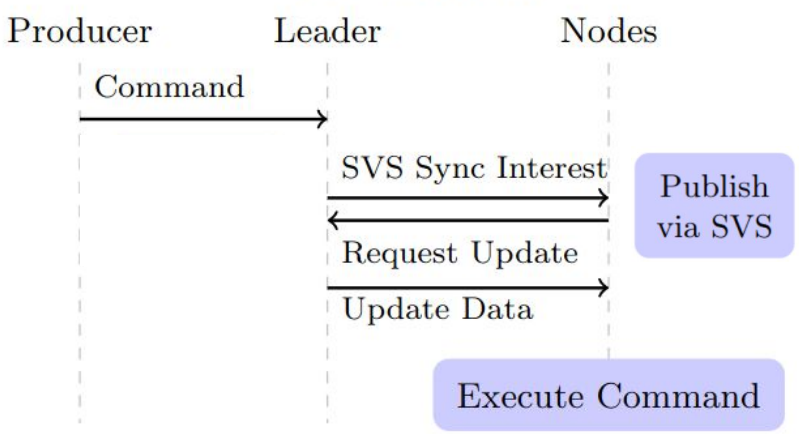
Need to pick a design

- How nodes know each others' storage availability

When to Share Availability

Pre-shared model

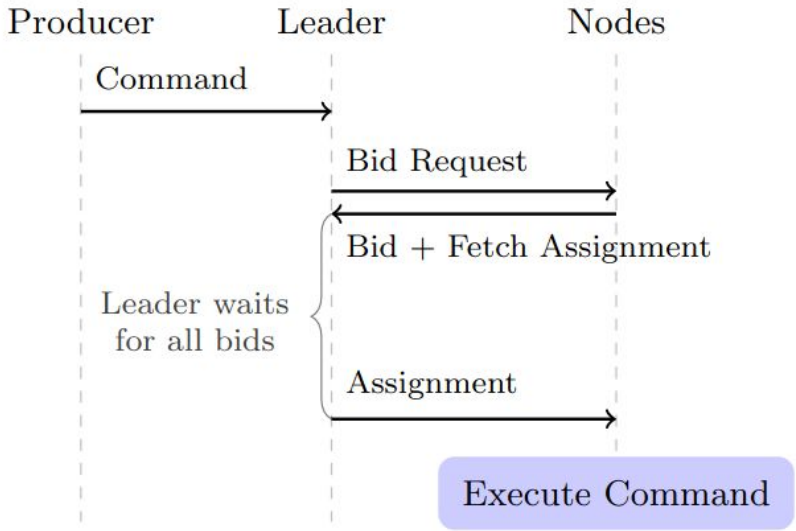
- Inspired by Hydra
- Piggyback Availability onto all group messages
- Ready to make a decision
- Publish assignments immediately



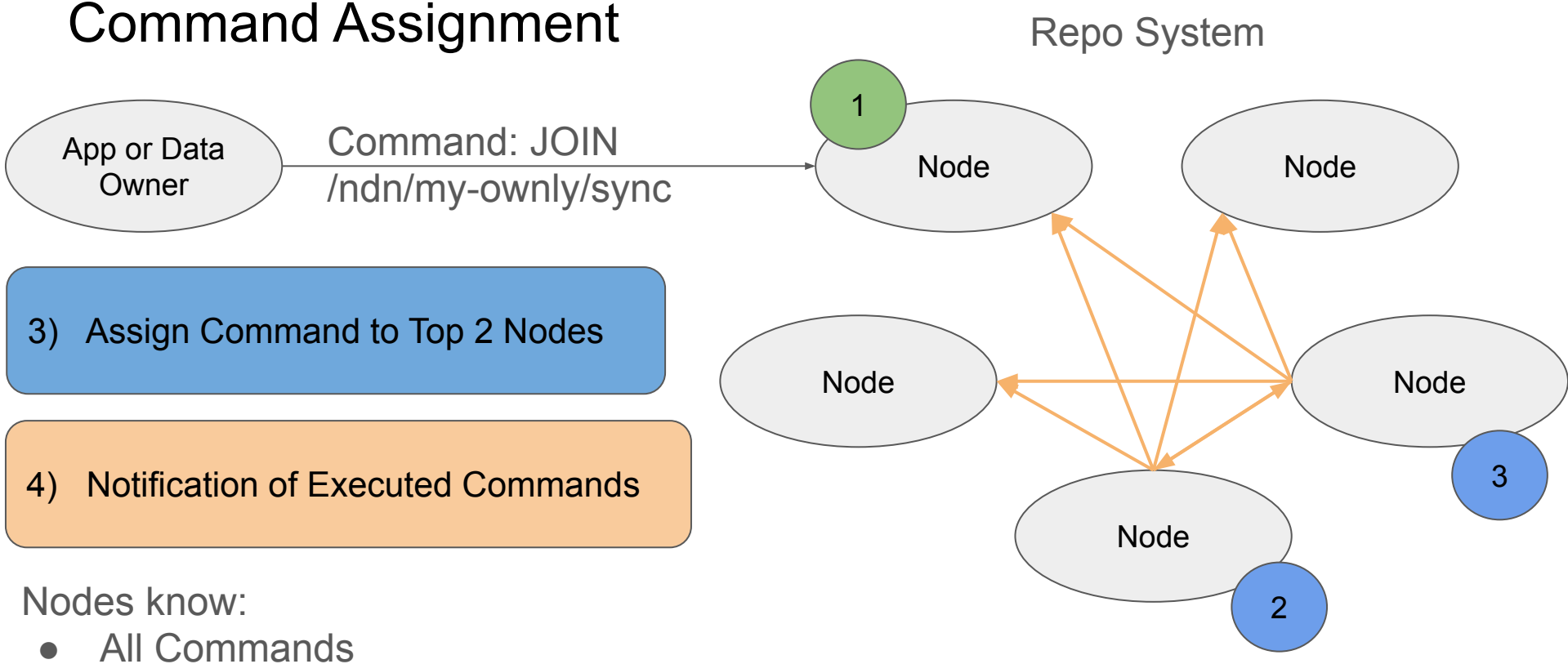
When to Share Availability

On-demand model

- Inspired by Kua
- Only fetch Availability when needed
- Auction-style:
  - Request nodes for bids (Availability)
  - Nodes request Assignment/Results



Command Assignment



Nodes know:

- All Commands
- Which node is executing which Command

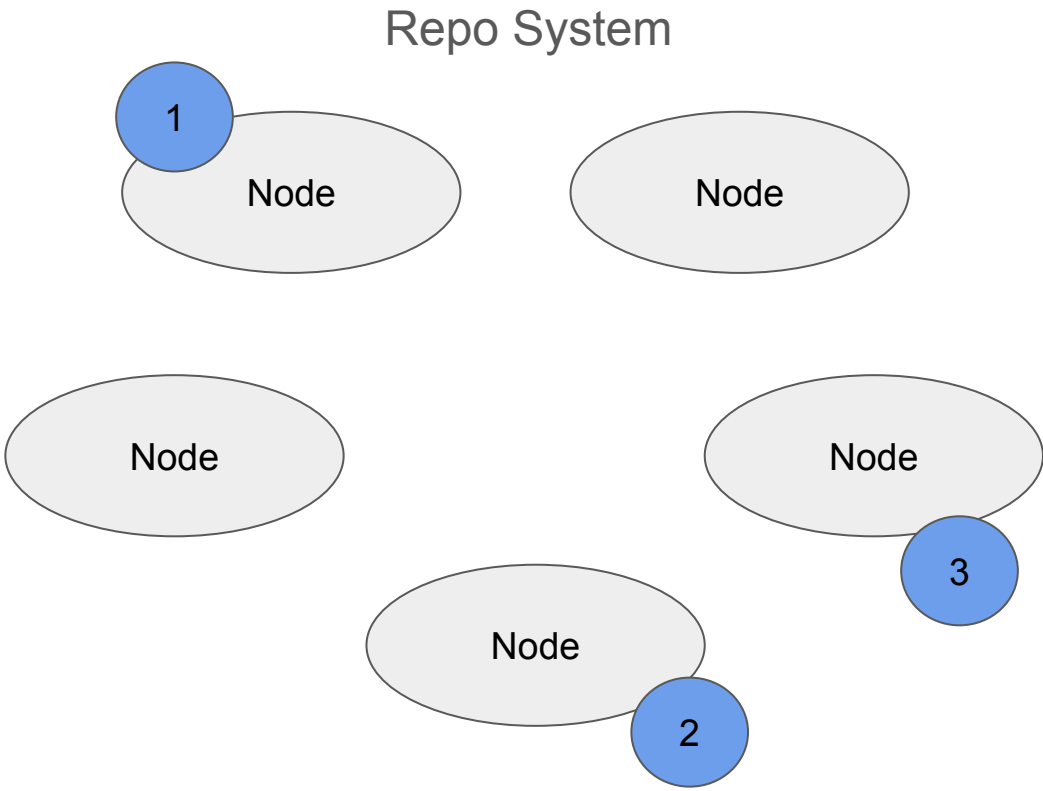
Major Components

1) New Command Handling

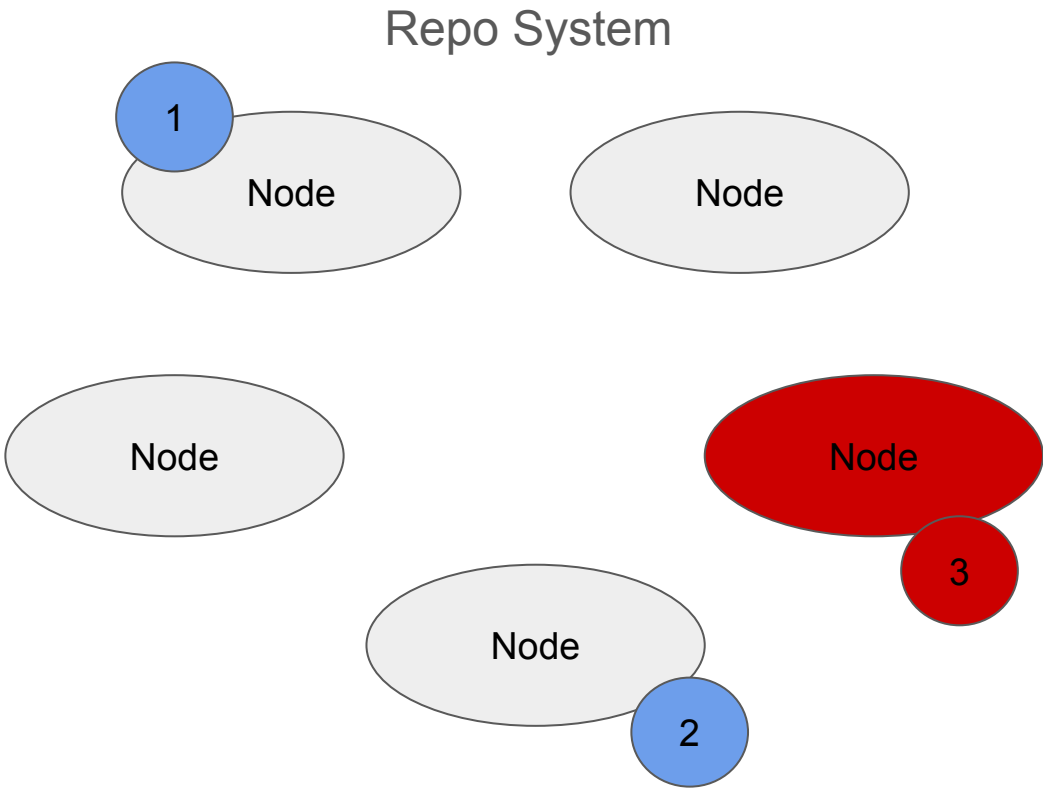
2) Command Assignment

3) **Reassignment after Node Failure**

Node Failure



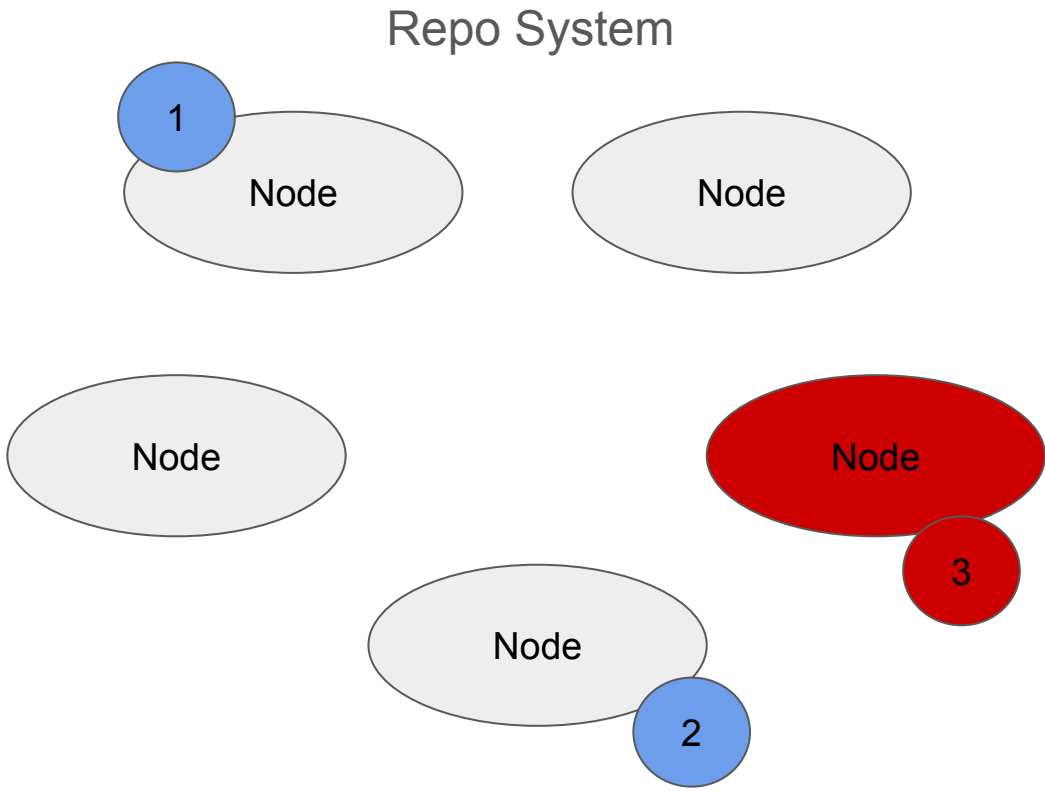
Node Failure



Node Failure: Heartbeats

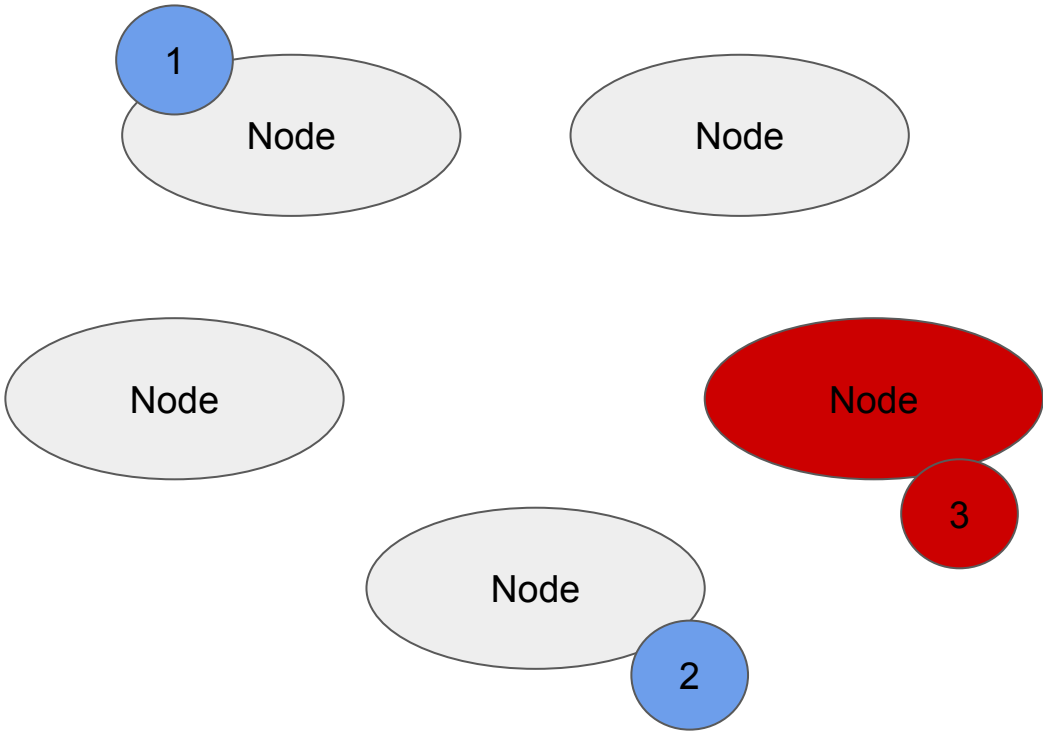
- Every node sends a heartbeat at least every 5 seconds
  - As Sync Interest

- After 3 missing heartbeats
  - Assume node is dead
  - Leader re-distributes



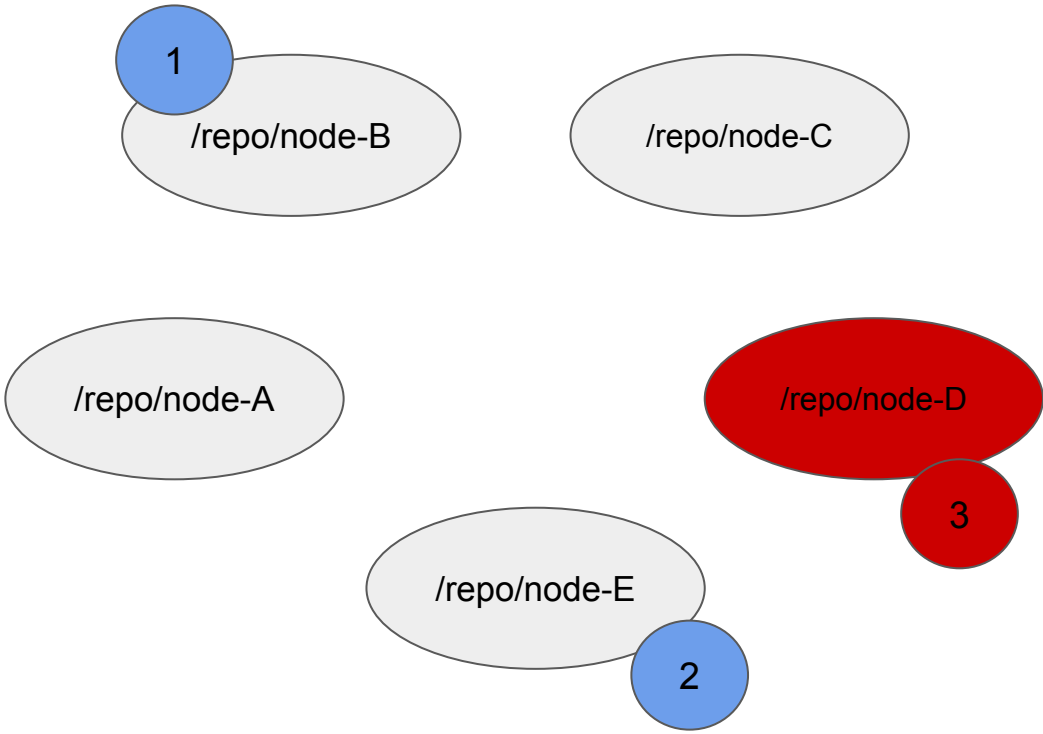
Node Failure: Leader Re-Distribution

- Leader chosen by first in alphabetical order



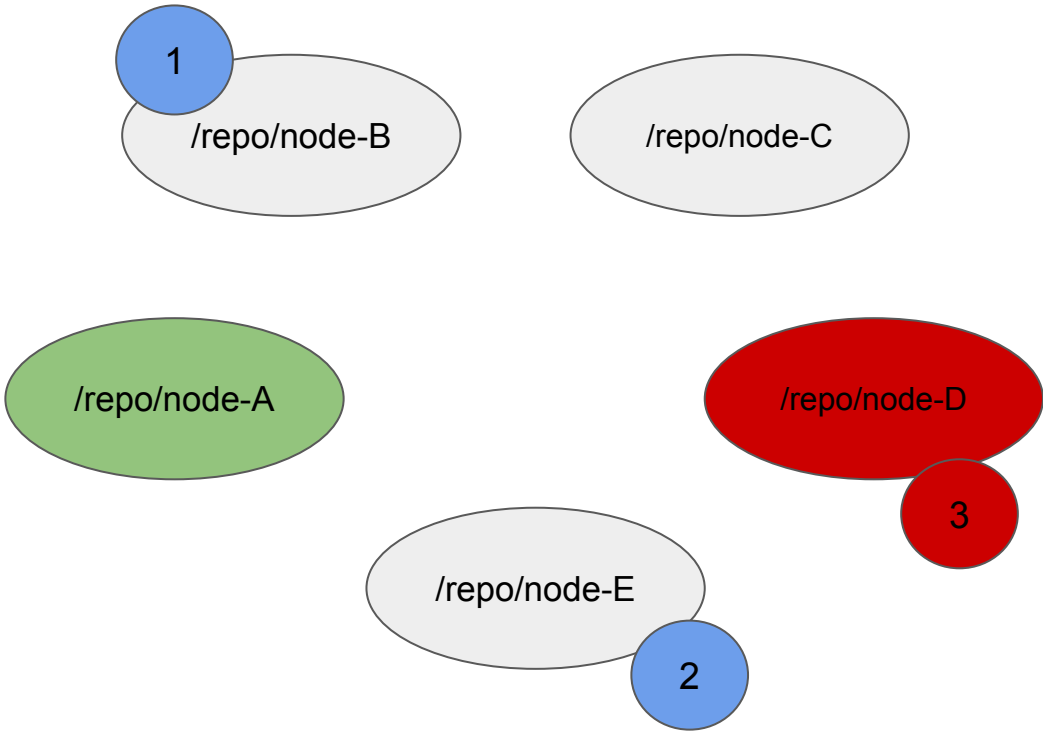
Node Failure: Leader Re-Distribution

- Leader chosen by first in alphabetical order



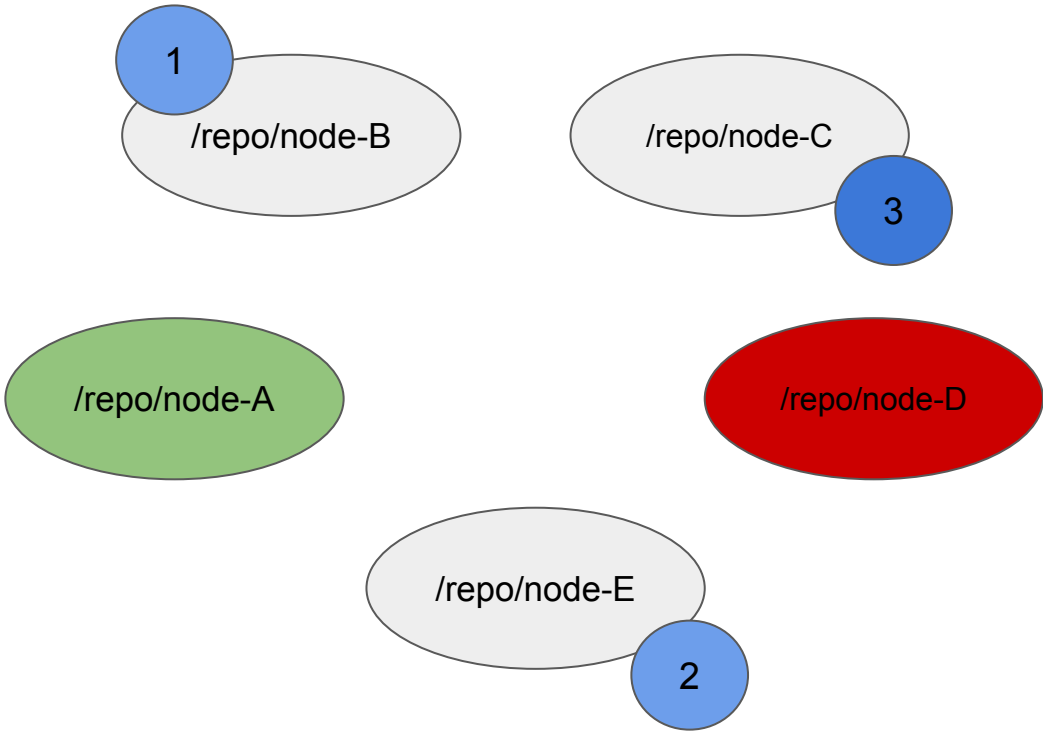
Node Failure: Leader Re-Distribution

- Leader chosen by first in alphabetical order



Node Failure: Leader Re-Distribution

- Leader chosen by first in alphabetical order
- Under-replicated command is re-assigned by Leader



Evaluation

Evaluation: Experimental setup

Platform:

Mini-NDN (NDN emulator on Mininet)

Topology: 24 nodes

Testbed link delays

Configurations:

Producers × Commands/Producer

1×1

1×8

4×1

24×1

24×24

1) Baseline
   a) Measuring RTT to replication Received Command to 3rd execution
   b) Sync Interests per Command
   c) Data satisfied per Command
2) Baseline with Link Losses
3) Failure detection+recovery time
4) Baseline + Network Partition

Baseline Performance

Table 1: p95 Replication Time (RTT)

Configuration	Pre-shared	On-demand
1x1 (1 cmd)	1.12	1.87
1x8 (8 cmd)	1.24	2.86
4x1 (4 cmd)	1.45	2.08
24x1 (24 cmd)	1.50	2.35
24x24 (576 cmd)	1.52	2.96

Table 2: Sync Interests per Command

Configuration	Pre-shared	On-demand
1x1 (1 cmd)	108.0	251.0
1x8 (8 cmd)	24.5	50.4
4x1 (4 cmd)	31.5	65.8
24x1 (24 cmd)	9.8	15.5
24x24 (576 cmd)	14.5	14.1

Table 3: Data Sent per Command

Configuration	Pre-shared	On-demand
1x1 (1 cmd)	56.0	29.0
1x8 (8 cmd)	16.6	29.0
4x1 (4 cmd)	18.2	29.0
24x1 (24 cmd)	8.1	29.0
24x24 (576 cmd)	11.7	29.0

Behavior With Link Loss

Table 5: p95 Replication Time vs Loss Rate

Loss Rate	Pre-shared (ms)	On-demand (ms)
0%	306	508
1%	4,013	17,044
5%	8,025	24,041
10%	14,035	28,057

Table 6: Sync Interests per Command vs Loss Rate

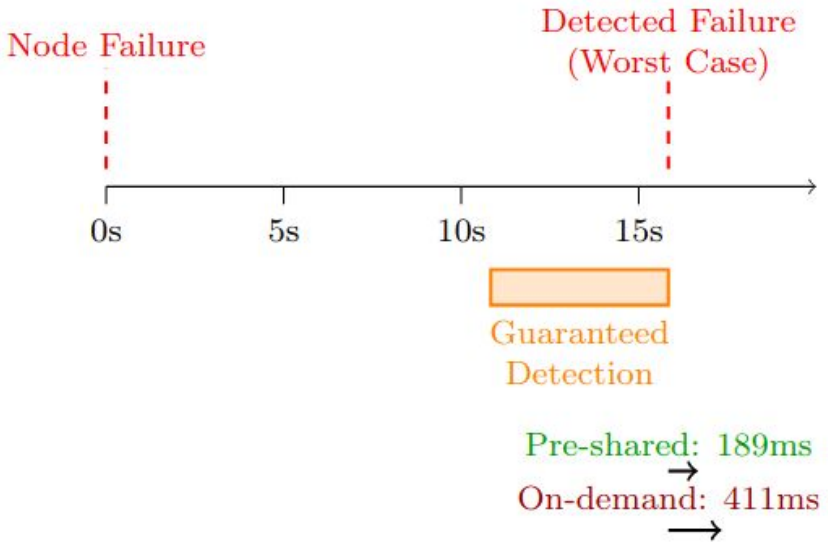
Loss	Pre-shared	On-demand
1%	21.6	51.5
5%	23.8	54.1
10%	26.9	60.6

Table 7: Data per Command vs Loss Rate

Loss	Pre-shared	On-demand
1%	17.4	452.4
5%	20.0	528.1
10%	23.5	695.0

Behavior Under Node Failures

- Both detected between 10.5 and 15.5 seconds
  - Timeout set to
    3 * Heartbeat Interval + 500ms
- Replication time after detection:
  - Pre-shared: 189ms
  - On-demand: 411ms

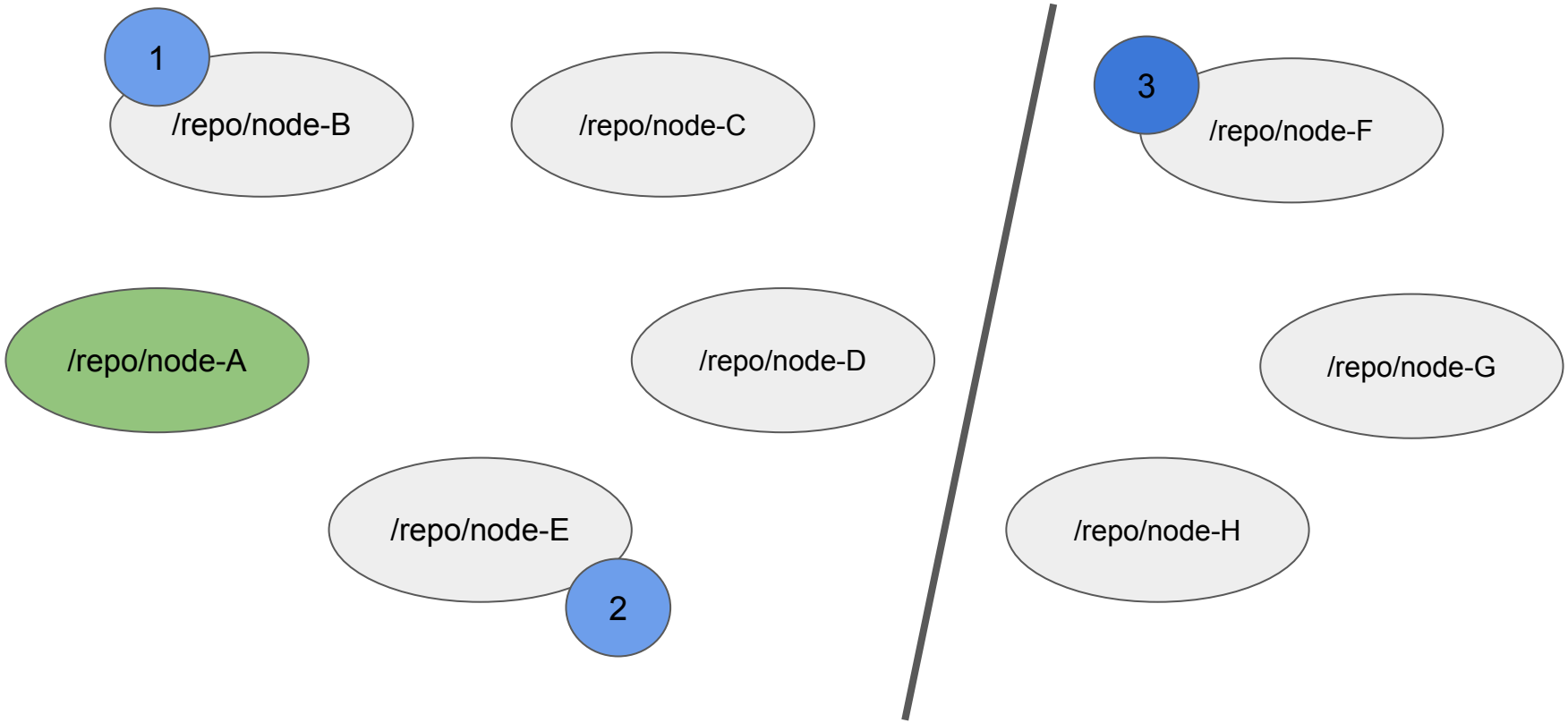


Network Partition

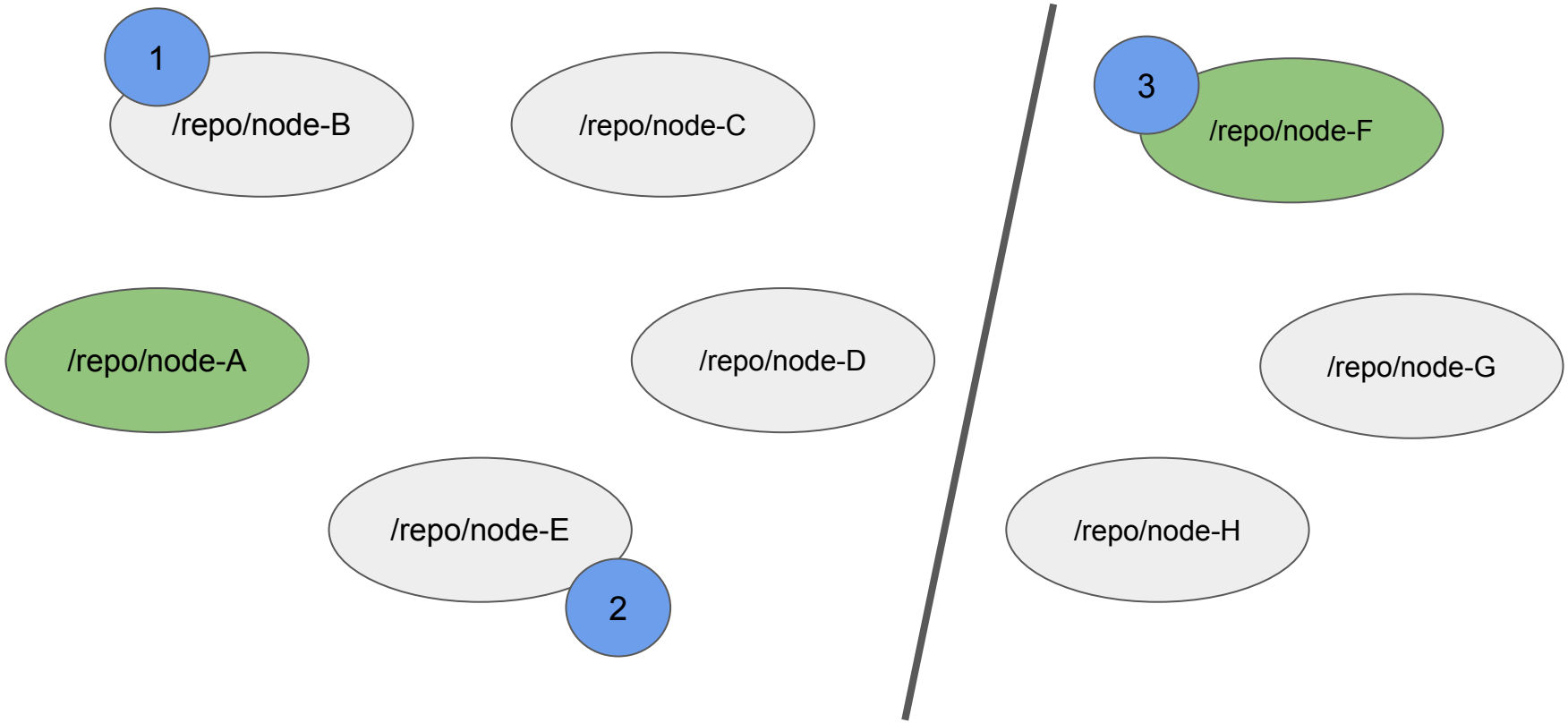
- Both partitions replicated to 3 per Command

- On-demand recovered
  - 8 seconds slower
  - 2.5x Data per Command

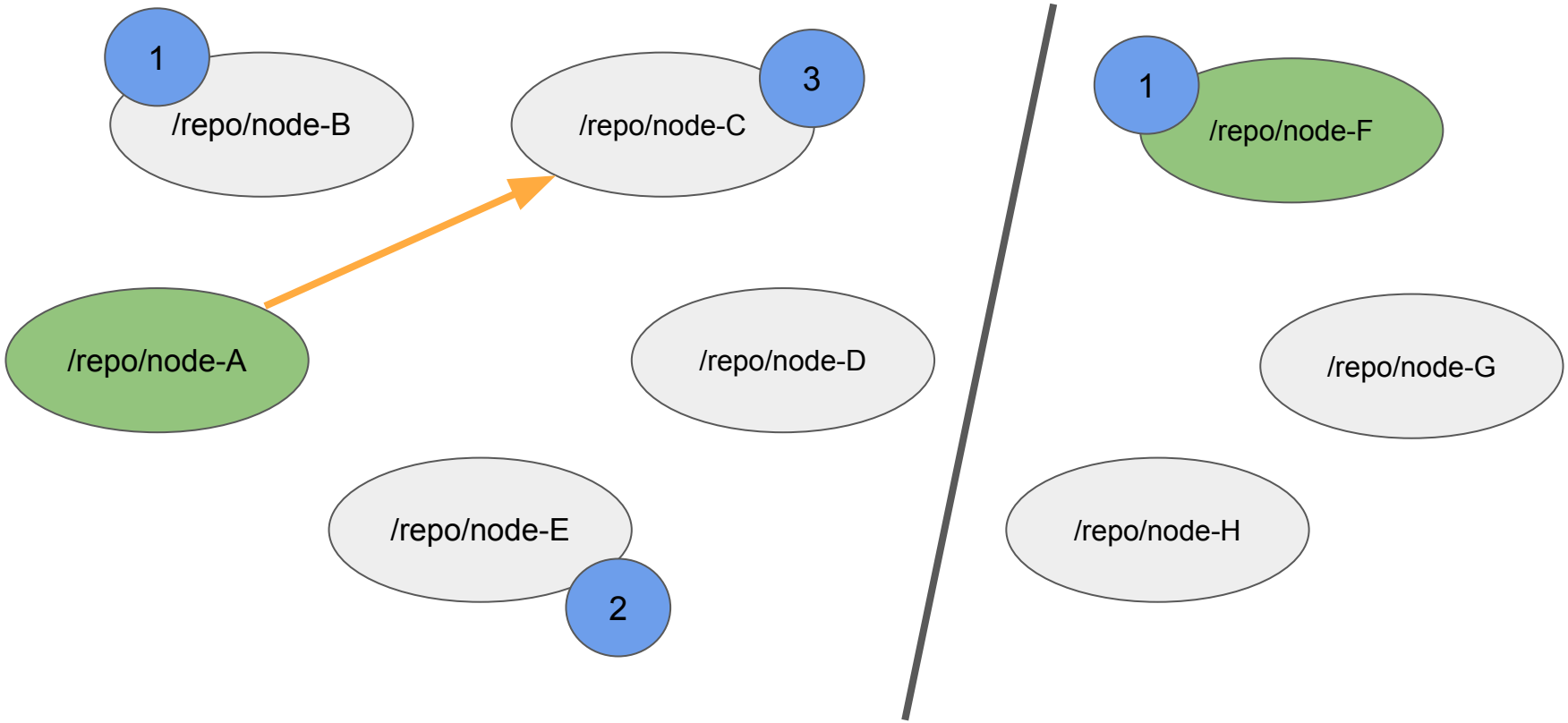
Network Partition



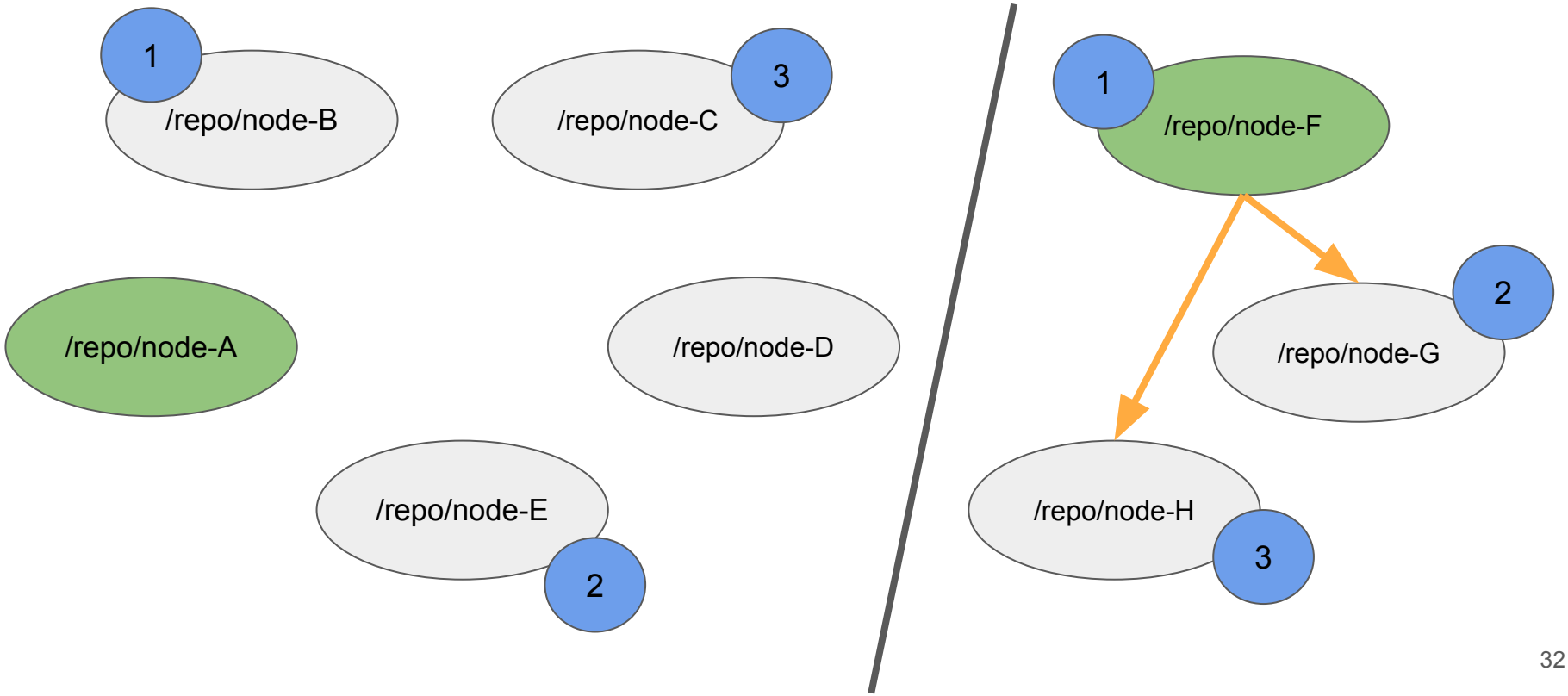
Network Partition



Network Partition



Network Partition



Conclusion + next steps

- Pre-sharing and reusing Availability
  - Lower overhead
  - Faster Response times
- Heartbeat and dynamic leader-based assignment
  - Responsive to node failures
  - Keeps data available for applications on both sides of a network partition

- Next step
  - Deployment and final evaluation on Testbed

Feedback / Q&A