Report-bonus: Failure Models for Gossip and Push-Sum

Anay Sinhal (68789243)   Radhey Sharma (21089036)

September 23, 2025

1 Objective

Extend the baseline project by introducing a parameterized failure model and evaluate how failures affect convergence time for gossip and push-sum across four topologies (full, 3D, line, imp3D). We report how we tested, what experiments we ran, and what we observed.

2 Methods

- **Failure model & parameter:**
  - We exposed a `failure_rate` parameter to the simulator and swept it from 0.0 to 0.5.
  - Experiments were orchestrated with a measurement script that invokes the program as: `gleam run <num_nodes> <topology> <algorithm> <failure_rate>`, collects the final time (ms), and writes plots/results (`results_bonus.json`, `plots/*.png`).
- **Experimental design:**
  - Fixed size: 1000 nodes.
  - Algorithms: gossip, push-sum. Topologies: full, 3D, line, imp3D.
  - Failure rates: 0.0, 0.1, 0.2, 0.3, 0.4, 0.5.
  - Data: results stored in `results_bonus.json` and used to generate convergence-time vs failure-rate plots [8][10].

3 Results (Figures)

Representative values (ms) from `results_bonus.json`:

- **Gossip:** full ≈ 33k across rates; 3D spikes at 0.3 (≈ 61.8k) then ≈ 32–33k at 0.4–0.5; line ≈ 32.3–33.0k (flat); imp3D spikes at 0.3 (≈ 55.7k) then ≈ 33.1k at 0.4–0.5.
- **Push-Sum:** full ≈ 1.7–2.0k (stable); 3D ≈ 32.7k at 0.0, dropping to ≈ 3.4k by 0.4–0.5; line ≈ 32.7k at 0.0, dropping to ≈ 0.41–0.43k by 0.4–0.5; imp3D ≈ 32.8k at 0.0, dropping to ≈ 1.7k by 0.4–0.5.

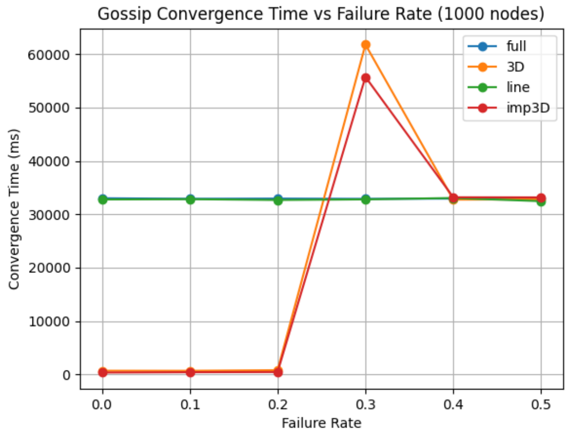

Figure 1: Gossip Convergence Time vs Failure Rate (1000 nodes)

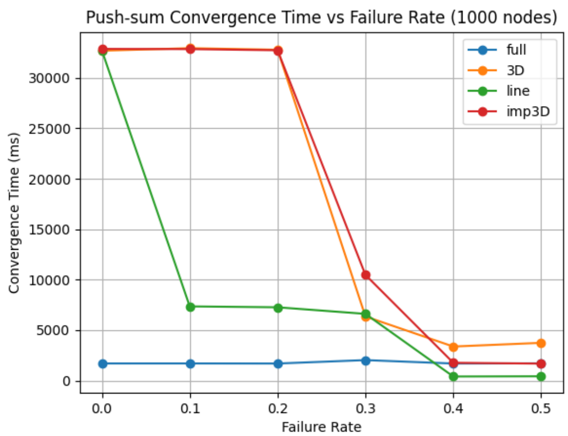

Figure 2: Push-Sum Convergence Time vs Failure Rate (1000 nodes)

4 Observations and Interpretation

Gossip robustness

- Line shows near-constant high times across failures; long diameter offers little path redundancy, so failures don't help or hurt much at our thresholds.
- Full remains relatively stable; dense connectivity provides alternate routes that mask moderate failure rates.
- 3D and imp3D exhibit a fragility threshold around 0.3 where convergence slows sharply (spikes), then recover toward ≈33k at higher failure rates; this suggests a regime where connectivity degrades enough to delay rumor saturation before the system effectively shrinks to a smaller active core that finishes at a similar time.

Push-Sum counterintuitive speedups

- For sparse topologies (line, 3D), increasing `failure_rate` beyond ∼0.3 dramatically reduces measured time (e.g., line from ∼32.7k at 0.0 to ∼0.41k at 0.4–0.5). A plausible explanation is that failures reduce the effective participating set and shorten paths, so the remaining subgraph converges faster under our termination criterion. Another possibility is that failures act like randomized damping, accelerating ratio stabilization in parts of the graph.
- Full is largely unaffected (≈1.7–2.0k across rates), consistent with high redundancy and rapid averaging even under failures.
- imp3D benefits substantially at higher failure rates, reflecting both shortcuts and effective network shrinkage.

5 How We Tested

- Automated runs with `measure_bonus.py`: iterated over algorithm/topology pairs and predefined failure rates at N=1000, captured the final millisecond count printed by the program, saved to JSON, and generated plots.
- We used the project CLI and output format described in the code/README; the simulator prints a single final number for convergence time.

6 Limitations

- Single-run per setting in our sweep (no confidence intervals); results may vary with RNG seeding and scheduling.
- Different failure semantics (e.g., permanent crash vs transient link drops) can shift thresholds and the direction of effects; our observations are specific to our implemented failure model and termination checks.

7 Conclusion

- **Gossip:** dense and shortcut-rich topologies are resilient, but lattice-like graphs can hit a failure-rate "slowdown zone" (∼0.3) before stabilizing. Line remains a worst case.
- **Push-Sum:** higher failure rates can produce faster measured convergence on sparse graphs under our termination criteria, likely via effective subgraph reduction and damping effects; full remains robust and fast across rates.