CS 645: Assignment 1

Probabilistic Packet Marking for IP Traceback

Julianne Maghirang

February 20, 2026

Overview

This report evaluates the behavior of probabilistic packet marking through **node sampling** and **edge sampling algorithms** from two tree topologies using Python.

- **Topology 1**: 4 branches, depth 5, optimal $p^* = 0.20$
- **Topology 2**: 3 branches, depth 7, optimal $p^* \approx 0.14$

Parameters: $p \in \{0.2, 0.4, 0.5, 0.6, 0.8\}$, $x \in \{10, 100, 1000\}$, 50 trials each, 500-tick limit.

In each tick, a regular user sends one packet, while each attacker sends x packets, resulting in a total of $x \times 500$ attack packets over the duration of the experiment. Therefore, the value of x influences accuracy as well as convergence speed: a slower attacker (with $x = 10$) may not generate enough marked packets to reconstruct the attack path within the given time limit.

Question 1: Single Attacker

Topology 1 (depth 5)

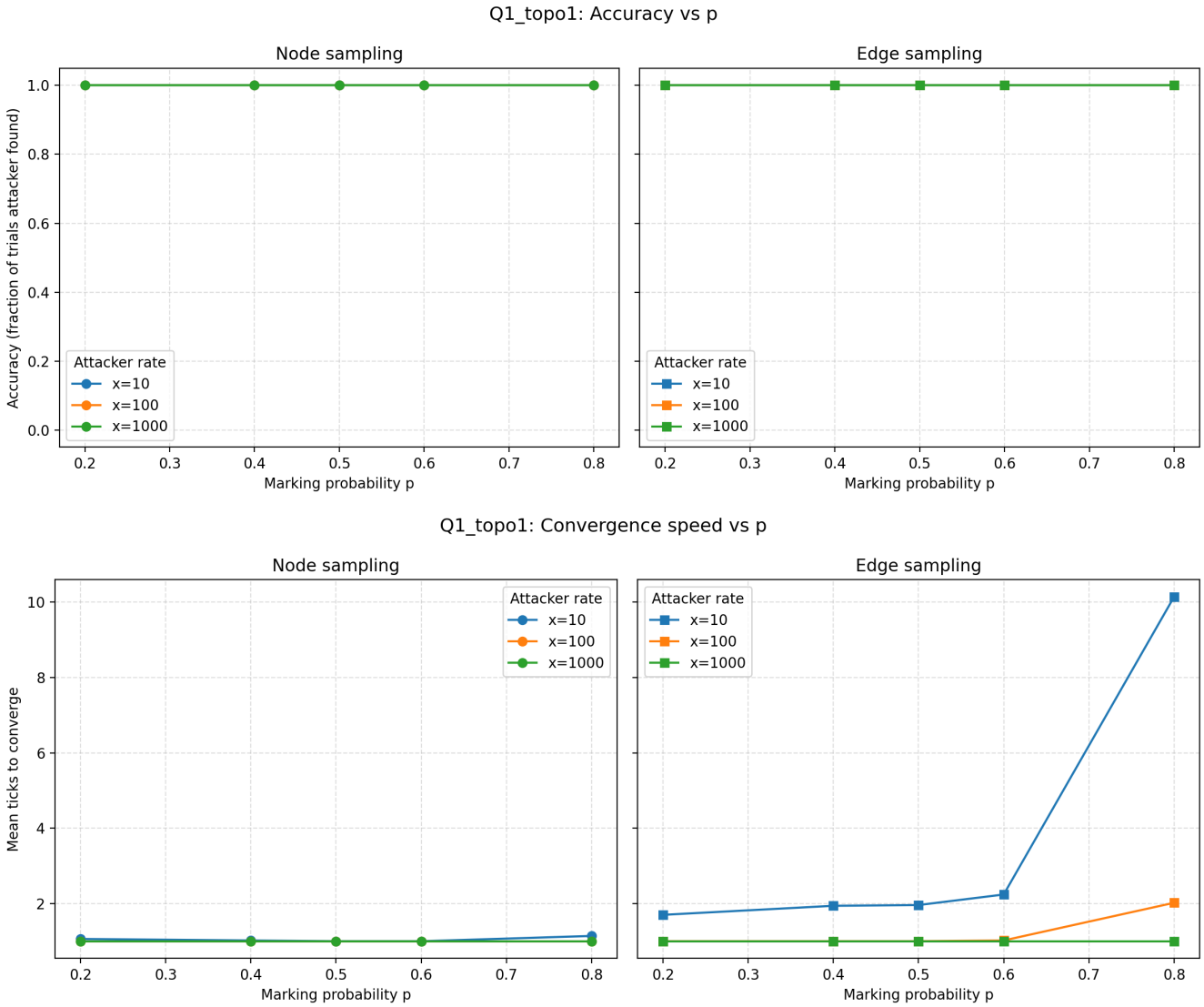

Figure 1: Q1 Topology 1: 100% accuracy for all settings and convergence reveals the cost of high p for edge sampling.

Both algorithms get 100% accuracy on this shallow topology. Even at $p = 0.8$, only $\sim$781 attack packets are needed to mark the farthest router, which is well within the $x = 10$ budget of 5 000 packets.

Convergence shows the real difference with node sampling always finishing in $\sim$1 tick. Edge sampling goes from 1.7 ticks at $p = 0.2$ to 10 ticks at $p = 0.8$ (for $x = 10$), which means that high p slows things down even when precision remains perfect. The faster the victim identifies the attacker, the sooner it can block the attack traffic.

Topology 2 (depth 7)

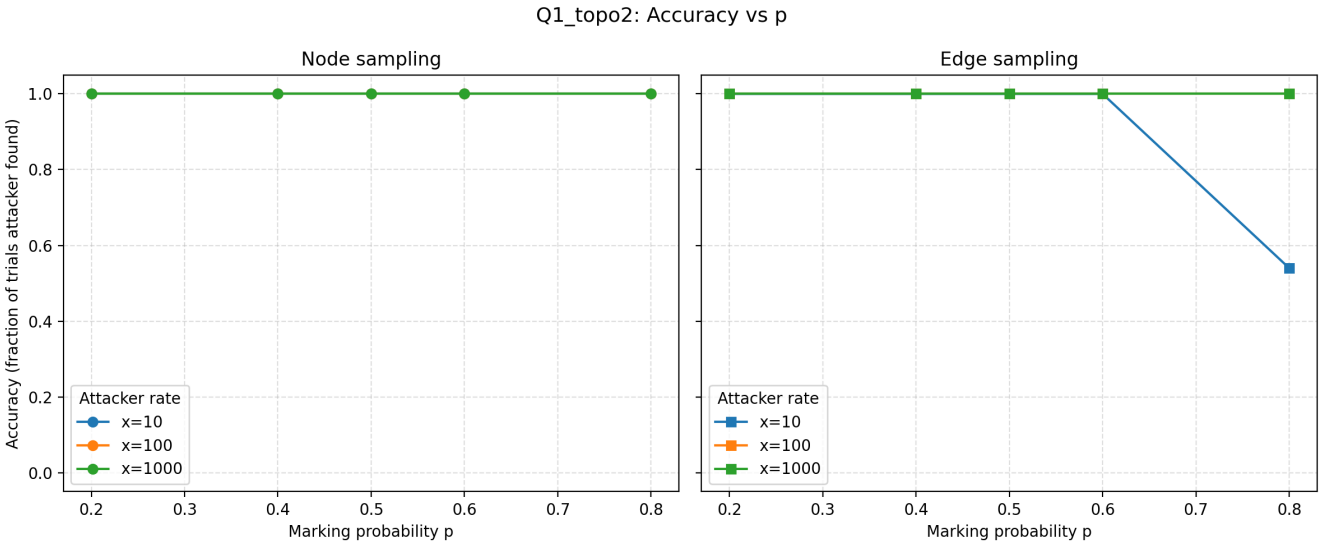

Figure 2: Q1 Topology 2: Accuracy vs. p. Edge sampling drops to 54% at $p = 0.8$, $x = 10$. Node sampling is unaffected.

On the deeper topology, using a p much higher than $p^* \approx 0.14$ hurts edge sampling. At $p = 0.8$, routers closer to the victim keep overwriting the marks left by routers farther away, so the victim never sees the full path before the time limit runs out. Node sampling is not affected because it only needs to count how often each router appears – closer routers always show up more, no matter how much overwriting happens.

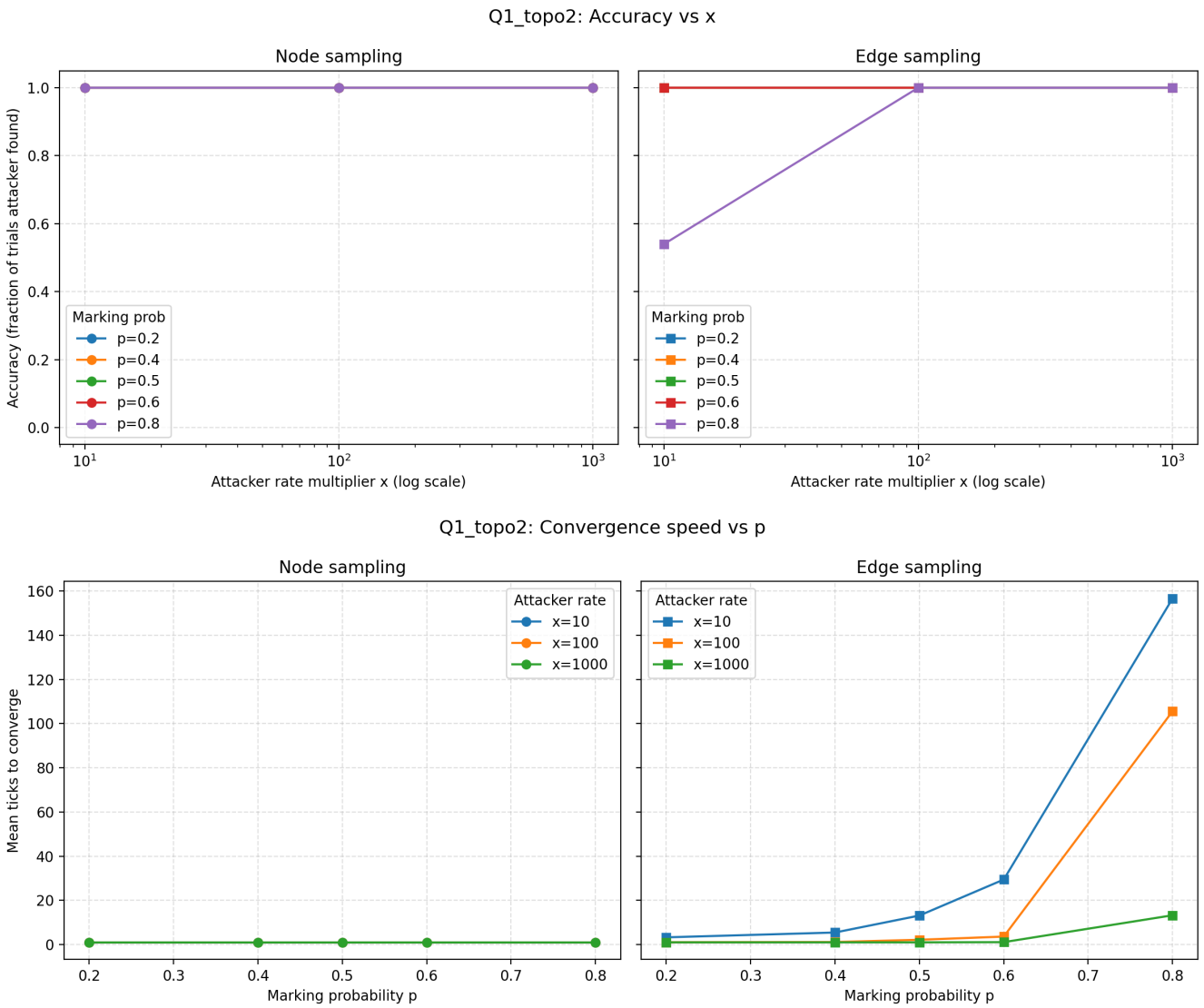

Figure 3: Q1 Topology 2: higher x rescues accuracy at $p = 0.8$ and convergence grows exponentially with p.

A higher x makes up for a bad p: edge sampling goes from 54% at $x = 10$ back up to 100% at $x = 100$ when $p = 0.8$. In other words, a faster attacker is actually easier to trace because the victim gets enough marked packets in time.

Convergence time grows quickly as p increases, which matches the bound from Savage et al.:

$$\mathbb{E}[X] < \frac{\ln d}{p(1-p)^{d-1}}$$

At $d = 7$, $p = 0.8$: about 38 000 packets are needed, which at $x = 10$ takes 3 800 ticks – well past the 500-tick limit, explaining why accuracy drops.

Question 2: Two Attackers

Topology 1 (depth 5)

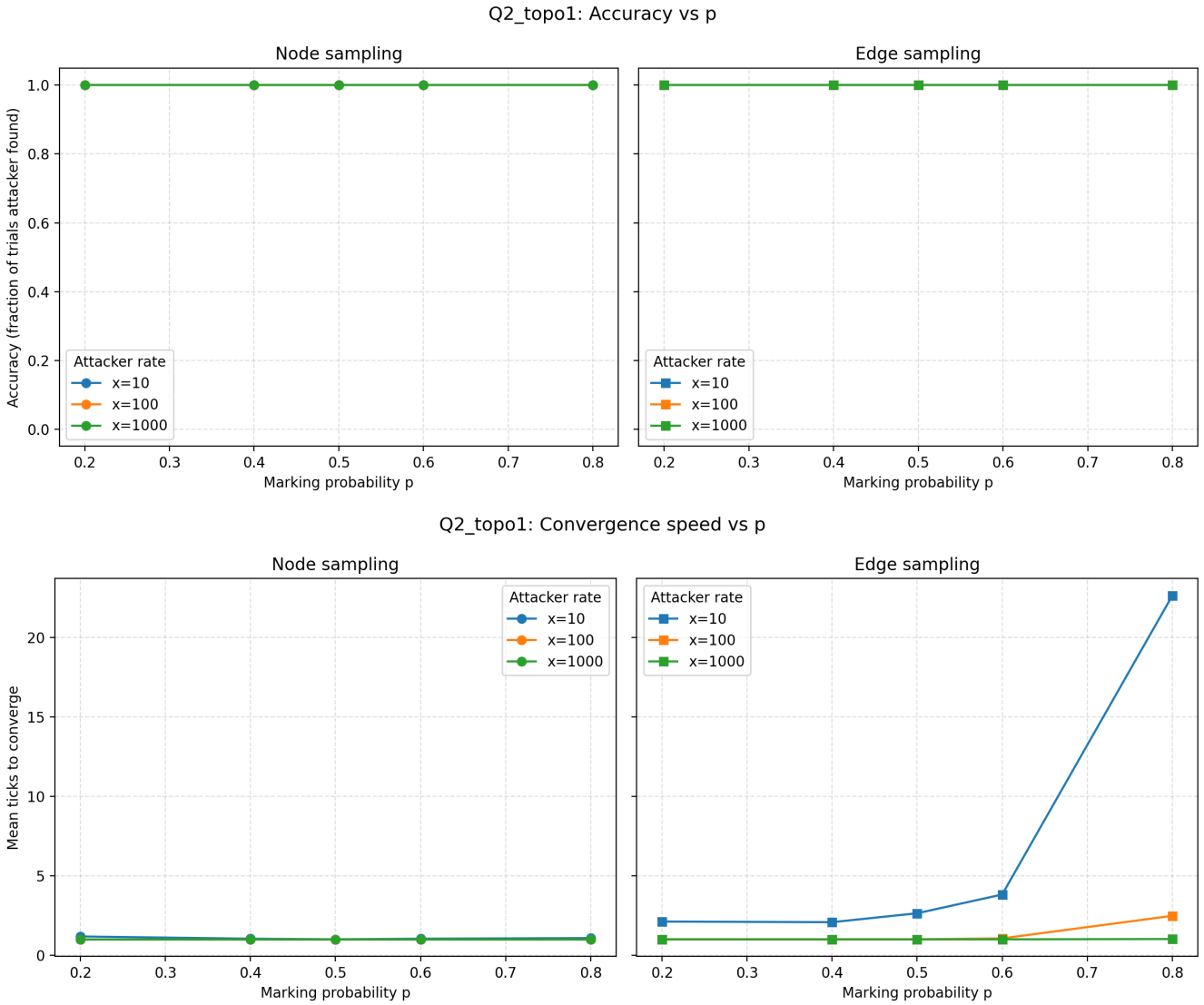

Figure 4: Q2 Topology 1: both algorithms still achieve 100% accuracy. Edge sampling convergence doubles compared to Q1.

The accuracy stays at 100% for all settings. Edge sampling at $p = 0.8$, $x = 10$ now takes $\sim$22 ticks, about 2$\times$ longer than the $\sim$10 ticks in Q1. This matches the paper: the number of packets needed grows with the number of attackers since each path has to be traced on its own.

Topology 2 (depth 7)

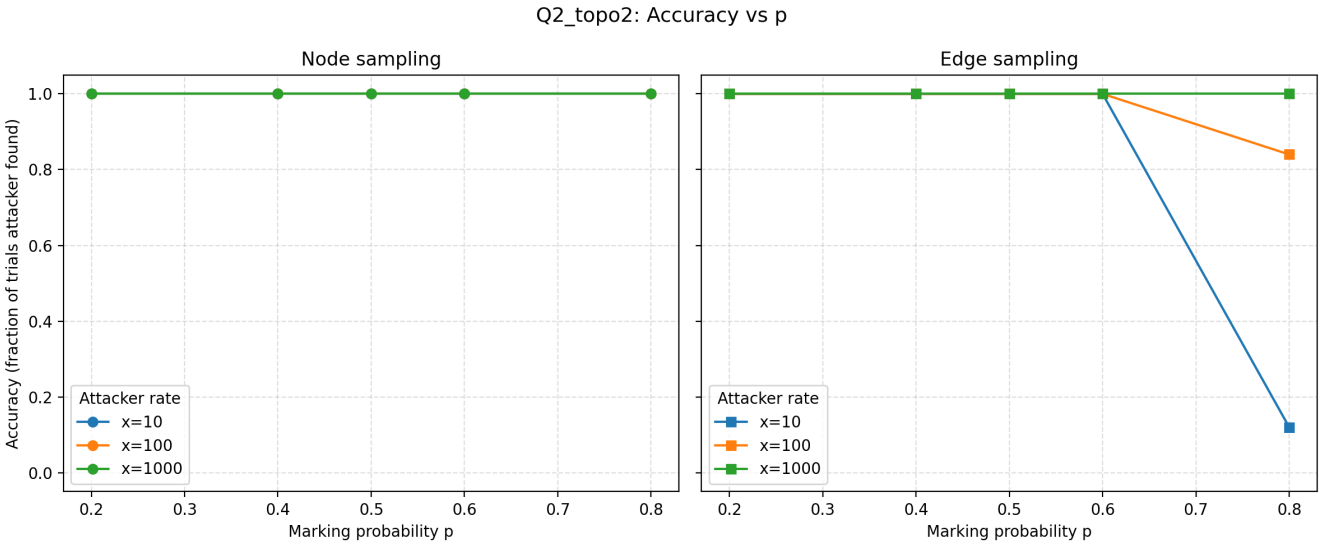

Figure 5: Q2 Topology 2: Accuracy vs. p. Edge sampling falls to 12% at $p = 0.8$, $x = 10$ and even $x = 100$ only reaches 84%.

Two attackers makes edge sampling much harder. At $p = 0.8$, accuracy drops to 12% for $x = 10$ (vs. 54% in Question 1) and 84% for $x = 100$ (vs. 100% in Question 1). The victim must reconstruct two independent 7-hop paths simultaneously, roughly doubling the required samples.

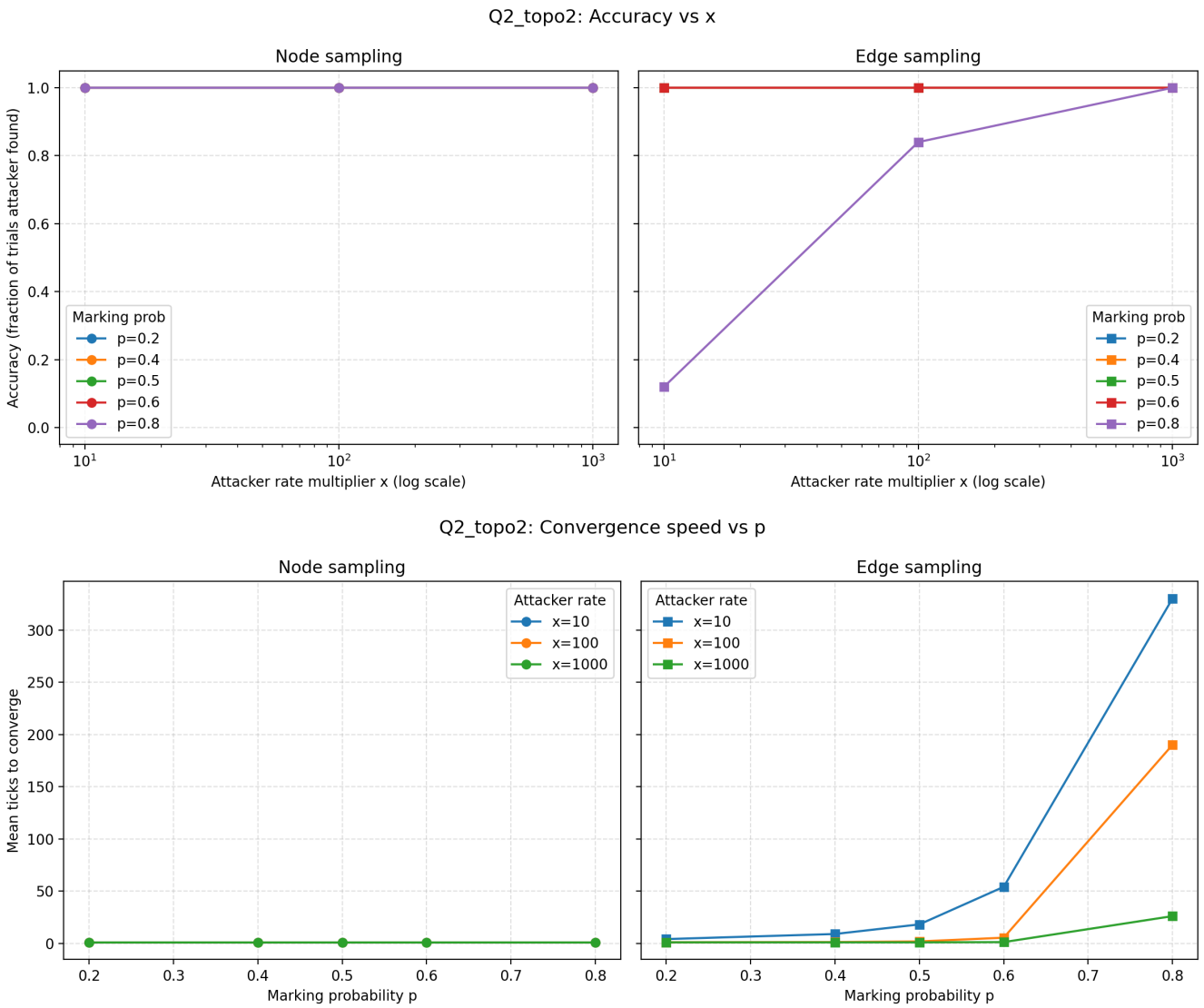

Figure 6: Q2 Topology 2: full accuracy at $p = 0.8$ now requires $x = 1000$ and convergence times are $\approx 2\times$ those of Q1.

Getting full accuracy at $p = 0.8$ now needs $x = 1000$ (500 000 packets), vs. just $x = 100$ in Q1. Convergence times are roughly $2\times$ longer than Q1 (330 ticks vs. 157 at $p = 0.8$, $x = 10$), which lines up with the idea that more attackers means more packets needed. Node sampling stays at $\sim$1 tick the whole time.

Comparison

	Node Sampling	Edge Sampling
Accuracy (topo1, any p/x)	100%	100%
Accuracy (topo2, $p \leq 0.6$)	100%	100%
Accuracy (topo2, $p = 0.8$, $x = 10$, Q1)	100%	54%
Accuracy (topo2, $p = 0.8$, $x = 10$, Q2)	100%	12%
Typical convergence	$\sim$1 tick	1–330 ticks
Sensitive to p	No	Yes
Reconstructs full path	No	Yes

Table 1: Node vs Edge comparison across all runs.

Node sampling is reliable but only identifies the source leaf since it can't reconstruct intermediate hops. On the other hand, edge sampling reconstructs the full attack graph yet it requires careful choice of p (optimal $p = 1/d$) and just enough attack packets.

Question 2's accuracy for node sampling is specific to tree topologies where branches share no nodes. In "real" life situational topologies, paths can overlap and node frequency counts mix between attackers. This causes failures the article describes. Therefore, edge sampling's graph-based reconstruction handles this better, making it the more reliable algorithm for multiple attackers in practice.