School of Engineering and Applied Science (SEAS), Ahmedabad University

CSE 400: Fundamentals of Probability in Computing

Scribe 2 Submission

Group 1
Section 2

Name: Jiya Thakkar
Enrollment No: AU2420189

Name: Niharika Ashar
Enrollment No: AU2440175

Date of Submission: 19th Feb 2026, 11:59 P.M

Milestone 2 Scribe

Project Theme:

Probabilistic Model for Reducing congestion at a 4-way intersection.

Purpose:

To reduce traffic congestion within a four-way intersection.

Background:

Traffic flow is stochastic (random) in nature, and thus traditional fixed-time signal controllers are not efficient.

Motivation:

Waiting time will be reduced, and will lead to the reduction of fuel consumption and frustration.

Problem Formulation:

Can we develop an algorithm which can be utilized to provide the probability of arrival of a vehicle and schedule signals in time to minimize the sum of the foreseen waiting time?

Question 1 Scribes: Project System and Objective

What is your part of the project probabilistic problem? Get the system objective straight and identify the major sources of uncertainty that are available.

Our project is all about developing a “smart” traffic light for a four-way intersection. Our system is not designed to follow a timer like normal traffic lights but to attempt to make better decisions based on what we see. The problem that we are trying to solve is called stochastic congestion. This is all about the fact that traffic does not come in a smooth, predictable flow. Instead, it comes in random chunks or waves. A timer would not be able to handle this randomness well, which is why sometimes people have to wait at an empty intersection. Our objective is to attempt to minimise the average wait time for cars. This means that we want to attempt to reduce the average amount of time that a car has to wait at a red light. Instead of making an educated guess, we are using probability and math to make our decisions about when to change the light. The problem with our system is that it is random. We do not know exactly when the next car is going to show up. We also do not know how fast the cars will be going when the light turns green. Some people will be going fast, and others will be going slow. Because of this randomness, our system has to make decisions based on probability instead of rules.

Question 2 Scribes: Any Random Uncertainty Modelling and Critical Variables

State the major random variables in your project and specify how the uncertainty is modelled in each of the variables. It should formulate probabilistic assumptions that are clear and obvious.

In order to represent the randomness that is happening on the road, we are employing two different mathematical ideas. First, we are employing the idea of a Poisson process to represent the number of car arrivals. This helps us to determine the probability of exactly k car arrivals in a fixed amount of time. The formula is:

$$P(N(t) = k) = \frac{(\lambda t)^k e^{-\lambda t}}{k!}$$

Here, λ is the average rate of car arrivals. In other words, it helps us to determine how many cars are on the road on average. Although we do not know the exact number of car arrivals at a given time, this formula helps us to determine the probability of different numbers of car arrivals. Second, we employ the idea of the exponential distribution to determine the time difference between each pair of car arrivals. Instead of determining the number of car arrivals, this formula determines the time difference between each car arrival. Since traffic flow is continuous, this is a continuous random variable. This formula helps us to determine whether cars are arriving in bunches or with a time difference in between. For now, we assume that car arrivals are independent. In other words, the arrival of one car does not affect the arrival of another car. Each car is treated as a separate random event. Although this is not exactly how traffic flow occurs in real life, this assumption makes it easier to model traffic flow mathematically for now. .

Question 3 Scribe: Probabilistic Reasoning and Dependencies

How do you use probabilistic relationships (e.g. dependence, independence, conditionality) in your project system to make reasoning, inference or decision making easier.

Our traffic light system does not just concentrate on one lane at a time. It concentrates on the whole intersection. When we consider the north-south and east-west lanes simultaneously, we can apply joint probability to analyse their behaviour. This allows us to determine the probability that the whole intersection is congested at the same time. In other words, joint probability allows us to ask the question: What is happening in both directions simultaneously? Are both sides congested, or is just one side building up? This line of thinking helps the system make more balanced decisions. The system also uses the idea of conditional probability in making decisions about turning the light green or not. For example, if there are already 10 cars lining up in Lane A, the light can ask itself, "Given that the line is already long, what is the probability that another car will show up in the next 2 seconds?" If the probability is high, it means that the lane is probably going to get even more congested. In this case, it would be better to turn the light green for a longer period of time to prevent overflow. If the probability is low, then it would be better to turn the light. In this manner, rather than just turning the light on and off based on a timer, the light now makes decisions based on what is happening at the present time and the probability of what might happen next. This is where the idea of probabilistic reasoning comes in. It enables the system to make intelligent decisions about what to do with the uncertainty.

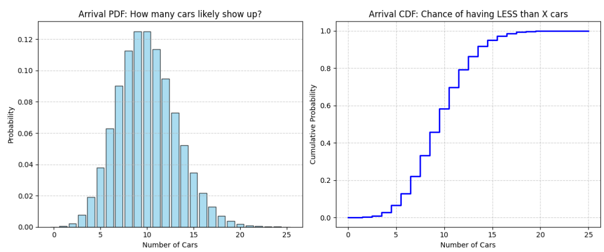

Figure 1: Arrival PDF and CDF

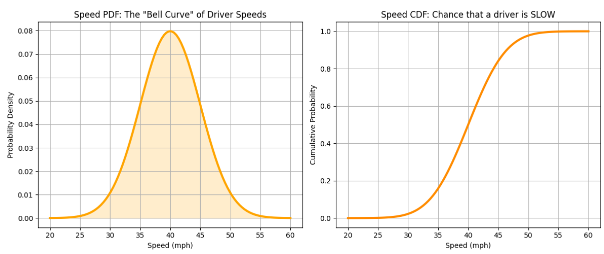

Figure 2: Speed PDF and CDF

Question 4 Scribe: Model Implementation Alignment

Explain how the existing probabilistic model can be applied to the implementation or experiment of the project. Mark out any assumptions that affect the design or evaluation decisions.

In Milestone 2, our Python simulation is intended to closely replicate our mathematical model so that the "virtual intersection" follows the same set of assumptions we are analyzing on paper. We simulate traffic by modeling the arrival of cars as Poisson random variables (generated each time step using numpy.random.poisson($\lambda\Delta t$)), which also matches modeling the time between cars as an exponential distribution, and we sample clearance speeds from a normal distribution (numpy.random.normal(μ, σ)) to simulate real-world variation in how fast cars actually move through the intersection. Rather than a simple if/then timer, the signal controller follows a randomized greedy strategy using weight-based randomization, where at each step, lanes are given probabilities proportional to their real-time queue length/density, so that a lane that is very congested gets priority, while the randomization prevents strict cycling and the risk of getting stuck in a loop. For this milestone, we assume our sensors are perfectly accurate, as we are trying to confirm if the decision-making logic is fundamentally correct before adding inaccuracies (missed detections, false positives, latency). We then perform Monte Carlo simulations and plot results using PDF/CDF plots to better understand congestion behaviour, as well as tail/worst-case scenarios.

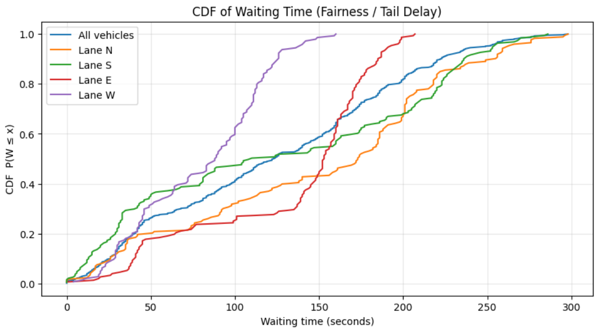

Figure 3: CDF of waiting time

Question 5 Scribe: Change and Cross-Milestone Consistency

Explain the present situation of your project probabilistic model. To determine what assumptions, components or relationships are currently well identified and which ones are likely to change or need further development in the next milestones. Explain your reasons in a few remarks.

Our current model is best viewed as Version 1.0: it combines Poisson arrivals (with exponential inter-arrival gaps) and normally distributed clearance speeds in a controlled simulation environment to test how a weight-based randomized greedy signal policy reacts under uncertainty. This baseline is intentionally simple so we can isolate whether the control logic reduces congestion better than a fixed-timer "dumb light". However, we recognize that real intersections do not always receive independent arrivals—vehicles often come in groups due to upstream traffic lights, creating bursty and correlated inflows that defies the strongest independence assumptions in our Version 1.0 setup. In the next milestone, we plan to update the arrival process to capture these sudden burst patterns and introduce a Markov-chain-style state model that represents the intersection transitioning between states such as "empty," "moderate," and "congested." Importantly, this evolution remains mathematically consistent with our current work because the Version 1.0 Poisson process becomes a special case of the Version 2.0 model when bursts disappear and arrival rates are steady, meaning we can extend the framework without discarding the logic we have already validated.

Question 6 Scribe: Open Issues and Attribution of Responsibility

At this point, determine any unresolved probabilistic questions or ambiguities in the project and point out what role(s) or activity can be held accountable in the next milestone.

For the final project, we have three open issues with clearly assigned ownership. First, the "Mega-Jam" limit is owned by the Lead Researcher: the idea is to use Chernoff bounds to show that the probability of congested scenarios is small, by bounding the tail probability that cumulative arrivals in a time window significantly exceed their expectation and then linking this event to queue growth relative to service capacity; for Milestone 2 we acknowledge that this needs clearer justification, and the independence assumptions are clean under Poisson increments but will require conditional arguments or alternative concentration tools once we introduce bursty/platooned traffic. Second, fairness logic is owned by the Algorithm Designer: because weight-based randomization can, with small probability, delay a low-traffic lane for too long, we will add a fairness mechanism such as a maximum-wait constraint or time-since-last-green boosting so that no approach can be starved while still keeping the controller adaptive. Third, the "Blurry Vision" test is owned by the Lead Coder: we will introduce sensor error probabilities (undercounts, overcounts, latency) into the queue estimates, rerun Monte Carlo experiments, and evaluate whether the policy remains effective when it acts on noisy observations rather than perfect data.