COMP9334 Capacity Planning of Computer Systems and Networks
Assignment Part B (Version 1), Term 1, 2025

Due 5:00pm, Thur 20 March 2025 (Thursday Week 5)

Change log and version info

Updates, changes and clarifications will appear in this box.

- Version 1.00 issued on 7 March 2025

Instructions

(1) There are 2 questions in Assignment (Part B). Answer all questions.

(2) The total mark for this assignment is 18 marks.

(3) The submission deadline is 5pm Thursday 20 March 2025. Submissions made after the deadline will incur a penalty of 0.2083% per hour (approximately 5% per day). Late submissions will only be accepted until 5pm Tuesday 25 March 2025, after which no submissions will be accepted.

(4) In answering the questions, it is important for you to show your intermediate steps and state what arguments you have made to obtain the results. You need to note that both the intermediate steps and the arguments carry marks. Please note that we are **not** just interested in whether you can get the final numerical answer right, we are **more** interested to find out whether you understand the subject matter. We do that by looking at your intermediate steps and the arguments that you have made to obtain the answer. Thus, if you can show us the perfect intermediate steps and the in-between arguments but get the numerical values wrong for some reason, we will still award you marks for having understood the subject matter.

You can take a look at the solution to revision problems to get some ideas the level of explanation that is required.

(5) If you use any computer programs to perform any part of your work, you **must** submit the **source code or file**, otherwise you lose marks for the steps.

(6) This is an individual assignment.

(7) Your submission should consist of:

   (a) A report describing the solution to the problems. This report can be typewritten or handwritten. This report must be in pdf format and must be named report.pdf. If you hand write your work on paper, then you will need to scan it. The submission system will only accept the name report.pdf.

   (b) One or more computer programs if you use them to solve the problems numerically. You should use zip to archive all the computer programs into one file with the name supp.zip. The submission system will only accept this name. The report must refer to the programs so that we know which program is used for which part. *Reminder: You must submit the source code or source file of all your programs.*

(8) Submission can be made via the course website. Your submission must not be more than 20 Mbytes in size; otherwise it will not be accepted.

(9) You can submit as many times as you wish before the deadline. A later submission will over-write the earlier one. We will only mark the last submission that you make.

(10) If you want to ask questions on the assignment, you can attend a consultation (see the Timetable section of the course website for dates and times) or post your question on the forum. Please note that if your forum post shows part of your solution or code, you must mark that forum post **private**.

(11) Additional assignment conditions:

- Joint work is not permitted on this assignment.
  - This is an individual assignment. The work you submit must be *entirely your own work*: submission of work even partly written by any other person is not permitted.
  - Do not request help from anyone other than the teaching staff of COMP9344.
  - Do not post your assignment work or code to the course forum.
  - Assignment submissions are routinely examined both automatically and manually for work written by others.

  Rationale: this assignment is designed to develop the individual skills needed to solve problems. Using work/code written by, or taken from, other people will stop you learning these skills. Other CSE courses focus on skills needed for working in a team.

- The use of generative AI tools, such as ChatGPT, is not permitted on this assignment.

  Rationale: this assignment is designed to develop your understanding of basic concepts. Using AI tools will stop you learning these fundamental concepts, which will significantly impact your ability to complete future courses. Moreover, ChatGPT has been found to give incorrect answers for advanced problems covered in this course.

- Sharing, publishing, or distributing your assignment work is not permitted.

  - Do not provide or show your assignment work to any other person, other than the teaching staff of COMP9334. For example, do not message your work to friends.
  - Do not publish your assignment code via the Internet. For example, do not place your assignment in a public GitHub repository.

  Rationale: by publishing or sharing your work, you are facilitating other students using your work. If other students find your assignment work and submit part or all of it as their own work, you may become involved in an academic integrity investigation.

- Sharing, publishing, or distributing your assignment work after the completion of COMP9334 is not permitted.
  - For example, do not place your assignment in a public GitHub repository after this offering of COMP9334 is over.

  Rationale: COMP9334 may reuse assignment themes covering similar concepts and content. If students in future terms find your assignment work and submit part or all of it as their own work, you may become involved in an academic integrity investigation.

(12) You are allowed to use or modify the sample code provided by the lecturer for your work in this assignment. You should acknowledge that in your report.

Question B1 (12 marks)

A call centre has 2 operators to deal with customer enquires. The set-up of the call centre is depicted in Figure 1. The centre has a shared queue with 3 waiting slots. We label the three waiting slots as Slots 1, 2 and 3 where Slot 1 is the head of the queue and Slot 3 is the end of the queue, see Figure 1.

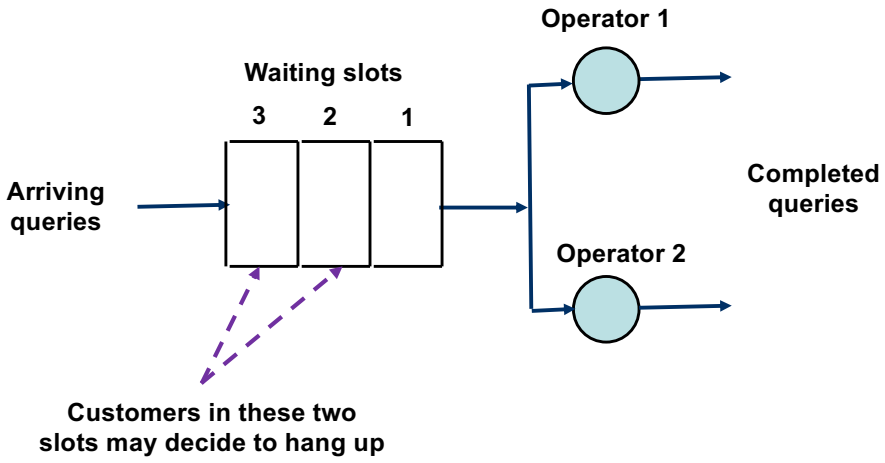

Figure 1: Depiction of the call centre.

Queries arrive at the call centre at a mean rate of λ according to the exponential inter-arrival time distribution.

The call centre handles an *arriving* query as follows:

- If there is at least an idle operator, then the call is directed to an idle operator.
- If both the operators are busy and there is at least one unoccupied waiting slot, the call will join the queue.
- Otherwise if all waiting slots are occupied, the call will be rejected.

If a query has been completed, the query leaves the call centre permanently. If a query has been completed and the queue is non-empty, the now available operator will take the query at the head of the queue to start to process it.

You can assume that the service time required by a query is exponentially distributed with a mean service time of $\frac{1}{\mu}$.

The call centre informs the customers in the queue what their position in the queue is, i.e., each customer in the queue knows whether they are in Slots 1, 2 or 3. We assume that the customers in Slots 2 and 3 may become impatient and decide to hang up their calls, i.e., these customers may choose to depart the call centre without getting their service. You can

assume that, for the customers in Slots 2 and 3 that, within an infinitesimal time Δt, there is a probability $\nu\ \Delta t$ that they will hang up. You may therefore interpret ν as the hang up rate. The customer in Slot 1 is always patient and will not hang up.

You can assume that the probability distributions for inter-arrival, service time and hanging up are independent. The three rates λ, μ and ν are all in queries per hour.

Answer the following questions:

(a) Derive the conditional probability that there are 4 customers in the call centre at time $(t+\Delta t)$ given that there are 5 customers in the call centre at time t. You should express the conditional probability in terms of μ, ν and Δt.

(b) Formulate a continuous-time Markov chain for the call centre. Your formulation should include the definition of the states and the transition rates between states. The transition rates should be expressed in terms λ, μ and ν.

(c) Write down the balance equations for the continuous-time Markov chain that you have formulated.

(d) Assuming $\lambda = 4.3$ and $\mu = 2.9$ and $\nu = 0.8$

  (i) Compute the steady state probabilities of the states of the continuous-time Markov chain that you have formulated in Part 3.

  (ii) Compute the probability that an arriving request will be rejected.

  (iii) Compute the throughput of the call centre.

  (iv) Compute the mean response time of the call centre.

Reminder: If you use a computer program to derive your numerical answers, you ***must*** *submit the source code. Do not forget to show us your steps to obtain your answer.*

Mark distribution: (a) 2. (b) 3. (c) 2. (d) (i) 2; (ii) 1; (iii) 1; (iv) 1.

Question B2 (6 marks)

This question considers a system consisting of four processing units and 2 queues. There are two types of requests that can arrive at this system:

- The **Type 1** requests require only one processing unit. These requests arrive according to a Poisson distribution with a mean rate of λ_1. These requests require a processing time which is exponentially distributed with a mean processing time of $\frac{1}{\mu_1}$.
- Each **Type 2** request requires two processing units *simultaneously*. By "simultaneously", we mean that a Type 2 request can only be admitted to the processing units when at least two processing units are available. If admitted, the two processing units which are working on this request will start to work on the admitted request at the same time and will complete the request at the same time.

  Type 2 requests arrive at the system according to a Poisson distribution with a mean rate of λ_2. The above description says that a Type 2 request requires the *same* amount of processing time at each processing unit. The processing time required by a Type 2 request at *each* processing unit is exponentially distributed with a mean processing time of $\frac{1}{\mu_2}$.

The four inter-arrival and service time distributions are assumed to be independent. You can assume the four rates λ_1, λ_2, μ_1 and μ_2 are expressed in the same unit.

The system has two queueing slots. One slot is reserved for Type 1 requests only and it has a capacity to hold exactly one Type 1 request. The other slot can only be used to hold Type 2 requests and it has a capacity to hold exactly one Type 2 request.

We first specify what happens when a request (either Type 1 or Type 2) departs the system upon its service completion. This departure can result in the availability of one or more processing units. We assume that the system gives a non-preemptive priority to the Type 1 request. The rules here are:

- If the Type 1 queueing slot is empty and there is a request at the Type 2 queueing slot, then the Type 2 request is admitted to the processing units if at least two processing units are available; otherwise, the request remains in the Type 2 queueing slot.
- If there is a request in the Type 1 queueing slot, then this request will be admitted into a newly available processing unit.

Note that the second rule above covers the case when both Type 1 and Type 2 queueing slots have a request each. In that case, the request in Type 1 queueing slot will be admitted because the second rule above gives the priority to the Type 1 request.

Note that the above rules imply that:

- If there is a Type 1 request is in the queue, then it is not possible to have any idle processing unit.
- If there is a Type 2 request in the queue, then it is not possible to have two idle processing units.

- It is possible to have an empty Type 1 queue, an occupied Type 2 queue and one idle processing unit.

We now describe what happens when a Type 1 request arrives. These are the possible scenarios:

- If the Type 1 queueing slot is occupied, then this request is rejected.
- If the Type 1 queueing slot is *not* occupied, then:
  - If at least one processing unit is idle, then this request will be admitted to an idle processing unit and its processing will begin.
  - If all processing units are occupied, then this request will be admitted into the Type 1 queueing slot.

Finally, we describe what happens when a Type 2 request arrives. These are the possible scenarios:

- If the Type 2 queueing slot is occupied, then this request is rejected.
- If the Type 2 queueing slot is *not* occupied, then:
  - If there are at least two idle processing units, then this request will be admitted to the processing units and its processing will begin.
  - If there are fewer than two idle processing units, then this request will be admitted into the Type 2 queueing slot.

We will now use an example to illustrate the operation of the system.

Figures 2 and 3 illustrate how the state of the system changes over time for a given sequence of arrivals and departures. The state of the system consists of the following attributes:

- The status of the 4 processing units are depicted as circles. There are 2 possible statuses for each processing unit: idle or busy. In Figures 2 and 3, a white circle is used to denote an idle processing unit while circles of other colours are used to denote busy processing units.
- The status of the Type 1 and Type 2 queues. Each queue has 2 possible states: empty or occupied.

We now go through each time point in Figure 2.

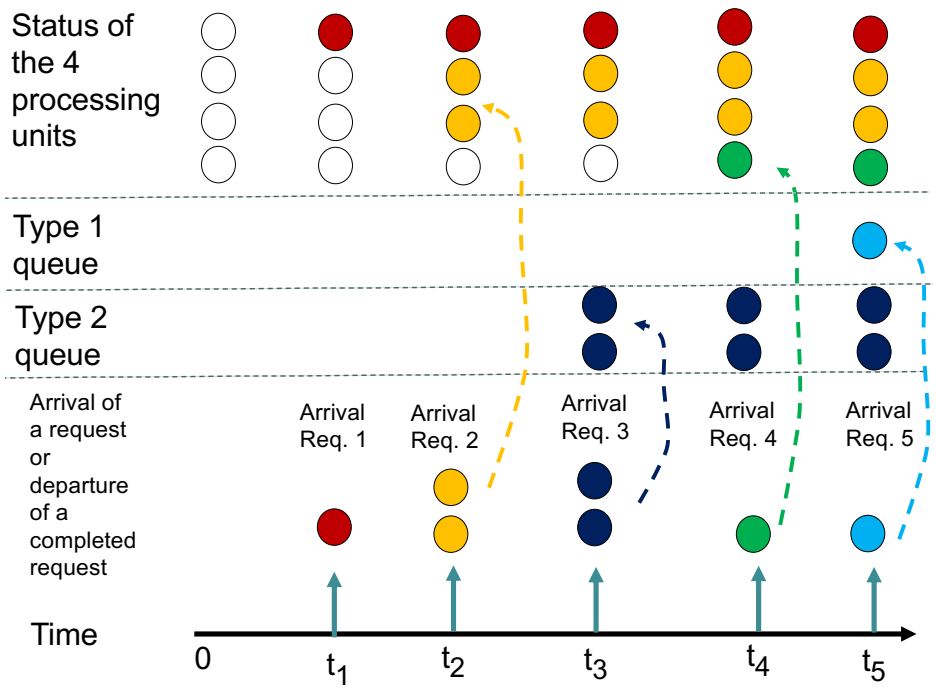

Figure 2: Example for Question 3 (Part 1)

At time 0, the system is assumed to have idle processing units and empty queues.

At time t_1, Request 1 (Red), which is of Type 1, arrives. Since all processing units are idle, the request goes to a processing unit. So a processing unit is busy (with the red request) and the other units are idle.

At time t_2, Request 2 (Orange), which is of Type 2, arrives. Since at least two idle processing units are available, the request goes to the processing units.

At time t_3, Request 3 (Navy blue), which is of Type 2, arrives. Since only one processing unit is idle, this request cannot be admitted to the processing units. Since the Type 2 queue is empty, this request goes to the Type 2 queue.

At time t_4, Request 4 (Green), which is of Type 1, arrives. Since there is an idle processing unit, this request goes to the processing unit directly.

At time t_5, Request 5 (Light blue), which is of Type 1, arrives. Since all processing units are busy and the Type 1 queue is empty, this request is admitted to the Type 1 queue.

This example continues on the next page.

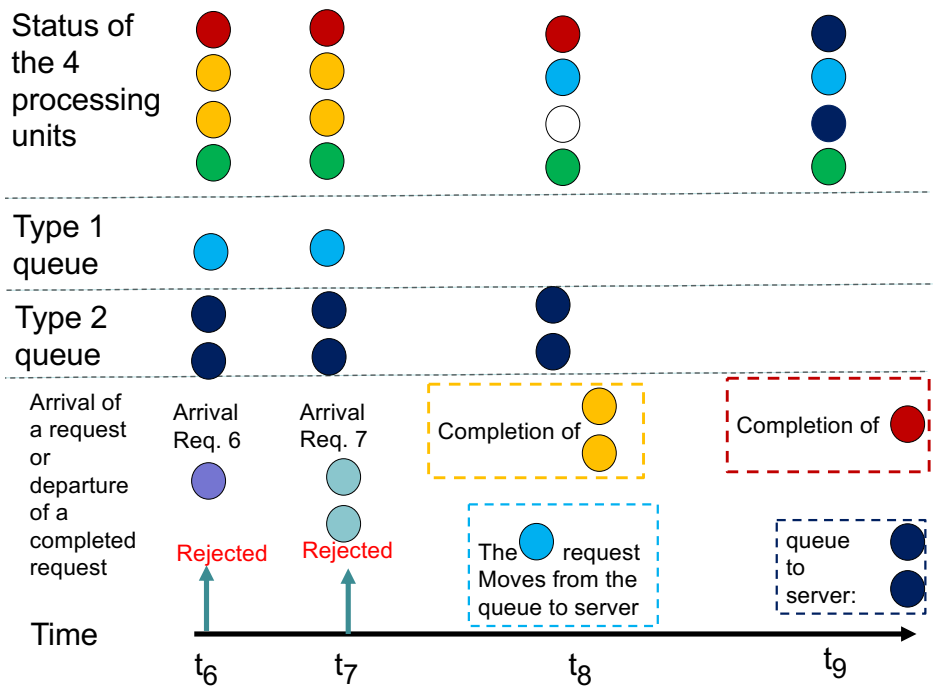

Figure 3: Example for Question 3 (Part 1)

At time t_6, Request 6 (Light purple), which is of Type 1, arrives. Since all processing units are busy and the Type 1 queue is occupied, this request is rejected.

At time t_7, Request 7 (Aqua), which is of Type 2, arrives. Since all processing units are busy and the Type 2 queue is occupied, this request is rejected.

At time t_8, the Orange Type 2 request in the processing units has been completed so it departs from the system. This makes two processing units are available. The queue status at the time of completion is that there is a request (light blue) in the Type 1 queue and a request (navy blue) in the Type 2 queue. Since the request in the Type 1 queue has priority over that in the Type 2 queue, the request in the Type 1 (light blue) is admitted into the processing unit.

At time t_9, the Red Type 1 request in the processing units has been completed so it departs from the system. This makes two processing units are available. The queue status at the time of completion is that the Type 1 queue is empty and a request (navy blue) in the Type 2 queue. The request in the Type 2 (navy blue) is now admitted into the processing units.

Answer the following questions:

(a) The system described above can be modelled by a continuous-time Markov chain whose state is a 4-tuple (n_1, n_2, q_1, q_2) where

- n_1 = number of Type 1 requests in the processing units,
- n_2 = number of Type 2 requests in the processing units,
- q_1 = number of requests in the Type 1 queueing slot, and
- q_2 = number of requests in the Type 2 queueing slot.

Note that not all values of n_1, n_2, q_1 and q_2 result in a valid state of the system. For example:

- The 4-tuple (0, 0, 0, 0) is a valid state and it represents the state where all the processing units and queues are empty.
- The 4-tuple (1, 1, 0, 1) is a valid state and it represents the state of the system at time t_3 in Figure 2.
- The 4-tuple (3, 0, 1, 0) is an invalid state because you cannot have an idle processing unit while there is a request in the Type 1 queue.
- The 4-tuple state (3, 0, 0, 2) is an invalid state because there is only 1 queueing slot.

Your task is to list all *valid* states for this Markov chain.

(b) Your task for this part is to draw the part of the state transition diagram around the state (2, 1, 0, 1). The requirements are that your drawing must include:

- The state (2, 1, 0, 1);
- All states that can transition into (2, 1, 0, 1) as well as the arcs from these states into (2, 1, 0, 1);
- All state that the state (2, 1, 0, 1) can transition out of as well as the arcs from (2, 1, 0, 1) into these states;
- The state transition rate of the arcs that you have included.

You can express the transition rates in terms of λ_1, λ_2, μ_1 and μ_2.

(c) Let $P(n_1, n_2, q_1, q_2)$ denotes the steady state probability of the state (n_1, n_2, q_1, q_2). For example, for the state (1, 1, 0, 1), the steady state probability in this state is denoted by $P(1, 1, 0, 1)$.

What is the probability that an arriving Type 1 request will be rejected?

You should express your answer in terms of the appropriate $P(n_1, n_2, q_1, q_2)$. For example, you may claim that the required probability is given by $P(0, 0, 0, 0) + P(0, 1, 0, 0)$ and explain why; note that this example is to show you what your answer may look like and the example is definitely not the correct answer.

(d) This question is similar to Part (c). What is the probability that an arriving Type 2 request will be rejected?

Reminder: Do not forget to explain how you obtain your answer.

Mark distribution: (a) 2. (b) 2. (c) 1. (d) 1.

– – –– End of assignment – – ––