Name:

Pledge:

EE/CPE 345: Modeling and Simulation
Midterm Exam – Fall 2019

1. **(20 points)** For our ARQ case study we are trying to model the probability of packet errors.

   On the transmission channel the signal is affected by Gaussian noise as discussed in lecture 7.

   (a) If the Gaussian noise on the channel is zero mean and has a standard deviation of $\sigma = 1$ (unusually high noise power), what is the probability of misdetection if the signal amplitude is A=1 when a bit of "1" is transmitted. The decision threshold is $\mu = A/2$. Round the probability to a first decimal accuracy.
   (b) Given the symmetry of the transmission modulation scheme, the probability of a bit being received in error (p) can be computed as being equal to the probability of missdetection. Assuming that each bit is independently affected by noise and that the probability of error from bit to bit is constant, what distribution can be used to determine the number of errors in a received packet with a length of 10 bits?
   (c) What is the average number of errors for a received packet?
   (d) Assuming that the ARQ protocol has an error detection code that can detect up to 2 errors, what is the probability that a packet received in error will be undetected, in which case the ARQ protocol will not function correctly.

Solution:

(5pt) a) $P_{missdetection} = \int_{-\infty}^{\mu} \frac{1}{\sqrt{2\pi}\sigma} e^{-(y-A)^2/2\sigma^2} dy = F_X(\mu) = F_X\left(\frac{A}{2}\right) = \varphi\left(\frac{A/2 - A}{\sigma}\right) =$

$= \varphi\left(\frac{\frac{1}{2}-1}{1}\right) = \boxed{\varphi(-0.5) \approx 0.3}$

(3pt) b) $p = P_{missdetection} = 0.3$

$\boxed{\text{Binomial distribution}}$

(5pt) c) $E(X) = E[\text{errors}] = n \cdot p$ where n=10 - length of packet

$\boxed{E(X) = 10 \cdot 0.3 = 3}$ X = # of errors

(7pt) d) $P(\text{pkt undetected}) = P(X \geq 3) = 1 - P(X \leq 2) = 1 - P(X=0) - P(X=1) -$

$- P(X=2) = 1 - \binom{10}{0} p^0 (1-p)^{10} - \binom{10}{1} p^1 (1-p)^9 - \binom{10}{2} p^2 (1-p)^8 =$

$= 1 - (0.7)^{10} - 10 \cdot 0.3 (0.7)^9 - 45 \cdot (0.3)^2 (0.7)^8 \approx \boxed{0.62}$

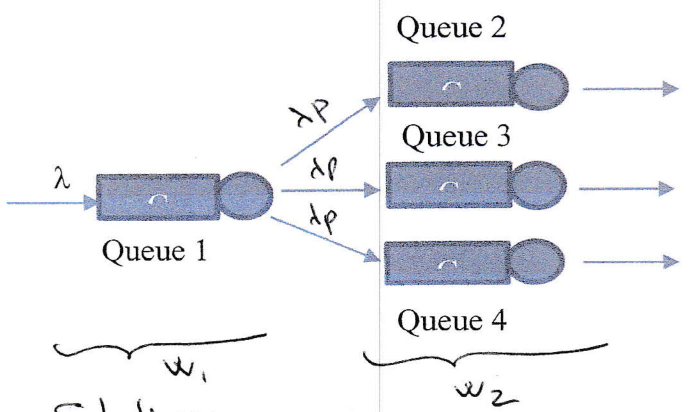
2. **(15 points)** A network provider is trying to optimize the routing policy for his network. For the network below, the traffic arrives at Queue 1with a Poisson distribution with a rate of $\lambda = 9$ packets per second. The traffic is routed at the exit of Queue 1 to Queue 2, Queue3, and Queue 4, using two possible routing algorithms:

   a. First packet goes to Queue2, second to Queue 3, third to Queue 4, and so forth in a round robin fashion

   b. Traffic is randomly routed with equal probability between the three Queues.

All servers have an exponential service time with a rate $\mu = 10$ packets per second.

(a) What type of queues we have in the network for case a?
(b) What type of queues we have in the network for case b?
(c) For case b. compute the end to end delay for the network.



with $p = \frac{1}{3}$

Solution:

(5pt) a) Queue 1 → M/M/1

Queue 2, Queue 3, Queue 4 → G/M/1

Justification: Poisson traffic is not split randomly ⇒

⇒ the substreams will not be Poisson.

(5pt) b). Queue 1 → M/M/1

Queue 2, Queue 3, Queue 4 → M/M/1 s

Justification: Poisson traffic split randomly →

→ Substreams Poisson with rates $\lambda \cdot p$. (marked in the figure)

(5pt) c). $w_1 = \frac{1}{\mu - \lambda} = \frac{1}{10-9} = \frac{1}{1} = 1$ second

$w_2 = \frac{1}{\mu - \lambda p} = \frac{1}{10 - \frac{9}{3}} = \frac{1}{10-3} = \frac{1}{7} \approx 0.14$

$\Rightarrow w = w_1 + w_2 = 1.14$ sec

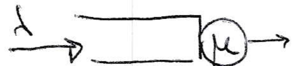
3. **(15 points)** Consider that for a small MacDonald restaurant, the manager is trying to optimize the waiting delays for his customers. He is considering to organize the restaurant to have one single common queue, served by one server. He is hiring a server for his restaurant. He has two applicants, one is not trained and has an exponentially distributed service time with a rate μ = 2 customers per minute, and the other one is highly trained (but requires a higher salary) and can serve exactly 2 customers per minute (its service time is 0.5 minutes).

   a. Describe the system in terms of entities, attributes, state of the system ,and events

   b. What types of queues we have in the two situations?

   c. The manager is considering hiring the highly trained server if the queueing delay that he would yield would be half of the delay that the unskilled worker will have. Should he hire the highly trained server? Justify mathematically.

Solution:



$\boxed{\text{Assume Poisson arrivals}}$

(5pt) a) Entities.

queue

server

Source of arrivals → outside of system.

Attributes: queue → service type → FIFO for this example
↳ physical queueing space

server → service distribution

Source → inter-arrival times distribution

State of system: (Number of customers in queue, Busy/Idle server)

Events: arrivals & departures (end of service)

(5pt) b) untrained server: M/M/1

trained server: M/D/1 → deterministic service.

(5pt) c). $W_Q^{M/G/1} = \dfrac{1+(cv)^2}{2} \cdot W_Q^{M/M/1}$ ← Assume $\lambda < \mu$ → queues stable

trained server → deterministic service → $\sigma^2 = 0$; $cv^2 = \dfrac{var(x)}{(E(x))^2} = 0$

$\Rightarrow W_Q^{M/D/1} = \frac{1}{2} W_Q^{M/M/1}$ → he should hire the highly trained server.

4. (10 points) What comes next in the following system snapshot table for a queueing system with one server (complete the table)? The current random inter-arrival times that were previously generated are: ~~5~~, ~~2~~, 3, 4, and the current generated random service times are: ~~4~~, ~~1~~, ~~2~~, 5 (none of these random variables were previously used – so the first one is available to use). Consider that a banking system is simulated, with the state of the system characterized by (number of customers in line, busy/idle for server). Busy state for server is marked as "1". The events in FEL are denoted as (type of event, time). For arrival events, type =1, for departure events, type =2.

Clock	Arrival Time	Departure time	System state	FEL	
T=2011		2011	(4,1)	(2, 2012) (1, 2013)	
T=2012		2012	(3,1)	(1,2013) (2, 2016)	
T=2013	2013		(4,1)	(2,2016) (1,2018)	
T=2016		2016	(3,1)	(1,2018) (2,2017)	(2,2017) (1,2018)
T=2017		2017	(2,1)	(1,2018) (2,2019)	
T=2018	2018		(3,1)	(2,2019) (1,2020)	

Bonus points problem (10 points):

Write a network description code (ned file) to have three nodes (of the same type) connected in a circle with unidirectional connections. Add a parameter for one of the nodes to be able to be identified as a first node to start sending the packets. Add a delay of 50 ms for each connection. Also include the code for the node type description.

One example of implementation code:

```
Simple Node
{
    parameters:
        bool first = default(false);
    gates:
        input in;
        output out;
}

network ExamNetwork
{
    submodules:
        first: Node {
            parameters:
                first = true;
        }
        second : Node;
        third : Node;
    connections:
        first.out --> { delay = 50 ms; } --> second.in;
        Second.out --> { delay = 50 ms; } --> third.in;
        third.out --> { delay = 50 ms; } --> first.in;
}
```

Tabulated Phi(z), where z = (x-mu)/sigma

Z	.00	.01	.02	.03	.04	.05	.06	.07	.08	.09
-3.9	.00005	.00005	.00004	.00004	.00004	.00004	.00004	.00004	.00003	.00003
-3.8	.00007	.00007	.00007	.00006	.00006	.00006	.00006	.00005	.00005	.00005
-3.7	.00011	.00010	.00010	.00010	.00009	.00009	.00008	.00008	.00008	.00008
-3.6	.00016	.00015	.00015	.00014	.00014	.00013	.00013	.00012	.00012	.00011
-3.5	.00023	.00022	.00022	.00021	.00020	.00019	.00019	.00018	.00017	.00017
-3.4	.00034	.00032	.00031	.00030	.00029	.00028	.00027	.00026	.00025	.00024
-3.3	.00048	.00047	.00045	.00043	.00042	.00040	.00039	.00038	.00036	.00035
-3.2	.00069	.00066	.00064	.00062	.00060	.00058	.00056	.00054	.00052	.00050
-3.1	.00097	.00094	.00090	.00087	.00084	.00082	.00079	.00076	.00074	.00071
-3.0	.00135	.00131	.00126	.00122	.00118	.00114	.00111	.00107	.00104	.00100
-2.9	.00187	.00181	.00175	.00169	.00164	.00159	.00154	.00149	.00144	.00139
-2.8	.00256	.00248	.00240	.00233	.00226	.00219	.00212	.00205	.00199	.00193
-2.7	.00347	.00336	.00326	.00317	.00307	.00298	.00289	.00280	.00272	.00264
-2.6	.00466	.00453	.00440	.00427	.00415	.00402	.00391	.00379	.00368	.00357
-2.5	.00621	.00604	.00587	.00570	.00554	.00539	.00523	.00508	.00494	.00480
-2.4	.00820	.00798	.00776	.00755	.00734	.00714	.00695	.00676	.00657	.00639
-2.3	.01072	.01044	.01017	.00990	.00964	.00939	.00914	.00889	.00866	.00842
-2.2	.01390	.01355	.01321	.01287	.01255	.01222	.01191	.01160	.01130	.01101
-2.1	.01786	.01743	.01700	.01659	.01618	.01578	.01539	.01500	.01463	.01426
-2.0	.02275	.02222	.02169	.02118	.02068	.02018	.01970	.01923	.01876	.01831
-1.9	.02872	.02807	.02743	.02680	.02619	.02559	.02500	.02442	.02385	.02330
-1.8	.03593	.03515	.03438	.03362	.03288	.03216	.03144	.03074	.03005	.02938
-1.7	.04457	.04363	.04272	.04182	.04093	.04006	.03920	.03836	.03754	.03673
-1.6	.05480	.05370	.05262	.05155	.05050	.04947	.04846	.04746	.04648	.04551
-1.5	.06681	.06552	.06426	.06301	.06178	.06057	.05938	.05821	.05705	.05592
-1.4	.08076	.07927	.07780	.07636	.07493	.07353	.07215	.07078	.06944	.06811
-1.3	.09680	.09510	.09342	.09176	.09012	.08851	.08691	.08534	.08379	.08226
-1.2	.11507	.11314	.11123	.10935	.10749	.10565	.10383	.10204	.10027	.09853
-1.1	.13567	.13350	.13136	.12924	.12714	.12507	.12302	.12100	.11900	.11702
-1.0	.15866	.15625	.15386	.15151	.14917	.14686	.14457	.14231	.14007	.13786
-0.9	.18406	.18141	.17879	.17619	.17361	.17106	.16853	.16602	.16354	.16109
-0.8	.21186	.20897	.20611	.20327	.20045	.19766	.19489	.19215	.18943	.18673
-0.7	.24196	.23885	.23576	.23270	.22965	.22663	.22363	.22065	.21770	.21476
-0.6	.27425	.27093	.26763	.26435	.26109	.25785	.25463	.25143	.24825	.24510
-0.5	.30854	.30503	.30153	.29806	.29460	.29116	.28774	.28434	.28096	.27760
-0.4	.34458	.34090	.33724	.33360	.32997	.32636	.32276	.31918	.31561	.31207
-0.3	.38209	.37828	.37448	.37070	.36693	.36317	.35942	.35569	.35197	.34827
-0.2	.42074	.41683	.41294	.40905	.40517	.40129	.39743	.39358	.38974	.38591
-0.1	.46017	.45620	.45224	.44828	.44433	.44038	.43644	.43251	.42858	.42465
-0.0	.50000	.49601	.49202	.48803	.48405	.48006	.47608	.47210	.46812	.46414

Tabulated Phi(z), where z = (x-mu)/sigma

Z	.00	.01	.02	.03	.04	.05	.06	.07	.08	.09
0.0	.50000	.50399	.50798	.51197	.51595	.51994	.52392	.52790	.53188	.53586
0.1	.53983	.54380	.54776	.55172	.55567	.55962	.56356	.56749	.57142	.57535
0.2	.57926	.58317	.58706	.59095	.59483	.59871	.60257	.60642	.61026	.61409
0.3	.61791	.62172	.62552	.62930	.63307	.63683	.64058	.64431	.64803	.65173
0.4	.65542	.65910	.66276	.66640	.67003	.67364	.67724	.68082	.68439	.68793
0.5	.69146	.69497	.69847	.70194	.70540	.70884	.71226	.71566	.71904	.72240
0.6	.72575	.72907	.73237	.73565	.73891	.74215	.74537	.74857	.75175	.75490
0.7	.75804	.76115	.76424	.76730	.77035	.77337	.77637	.77935	.78230	.78524
0.8	.78814	.79103	.79389	.79673	.79955	.80234	.80511	.80785	.81057	.81327
0.9	.81594	.81859	.82121	.82381	.82639	.82894	.83147	.83398	.83646	.83891
1.0	.84134	.84375	.84614	.84849	.85083	.85314	.85543	.85769	.85993	.86214
1.1	.86433	.86650	.86864	.87076	.87286	.87493	.87698	.87900	.88100	.88298
1.2	.88493	.88686	.88877	.89065	.89251	.89435	.89617	.89796	.89973	.90147
1.3	.90320	.90490	.90658	.90824	.90988	.91149	.91309	.91466	.91621	.91774
1.4	.91924	.92073	.92220	.92364	.92507	.92647	.92785	.92922	.93056	.93189
1.5	.93319	.93448	.93574	.93699	.93822	.93943	.94062	.94179	.94295	.94408
1.6	.94520	.94630	.94738	.94845	.94950	.95053	.95154	.95254	.95352	.95449
1.7	.95543	.95637	.95728	.95818	.95907	.95994	.96080	.96164	.96246	.96327
1.8	.96407	.96485	.96562	.96638	.96712	.96784	.96856	.96926	.96995	.97062
1.9	.97128	.97193	.97257	.97320	.97381	.97441	.97500	.97558	.97615	.97670
2.0	.97725	.97778	.97831	.97882	.97932	.97982	.98030	.98077	.98124	.98169
2.1	.98214	.98257	.98300	.98341	.98382	.98422	.98461	.98500	.98537	.98574
2.2	.98610	.98645	.98679	.98713	.98745	.98778	.98809	.98840	.98870	.98899
2.3	.98928	.98956	.98983	.99010	.99036	.99061	.99086	.99111	.99134	.99158
2.4	.99180	.99202	.99224	.99245	.99266	.99286	.99305	.99324	.99343	.99361
2.5	.99379	.99396	.99413	.99430	.99446	.99461	.99477	.99492	.99506	.99520
2.6	.99534	.99547	.99560	.99573	.99585	.99598	.99609	.99621	.99632	.99643
2.7	.99653	.99664	.99674	.99683	.99693	.99702	.99711	.99720	.99728	.99736
2.8	.99744	.99752	.99760	.99767	.99774	.99781	.99788	.99795	.99801	.99807
2.9	.99813	.99819	.99825	.99831	.99836	.99841	.99846	.99851	.99856	.99861
3.0	.99865	.99869	.99874	.99878	.99882	.99886	.99889	.99893	.99896	.99900
3.1	.99903	.99906	.99910	.99913	.99916	.99918	.99921	.99924	.99926	.99929
3.2	.99931	.99934	.99936	.99938	.99940	.99942	.99944	.99946	.99948	.99950
3.3	.99952	.99953	.99955	.99957	.99958	.99960	.99961	.99962	.99964	.99965
3.4	.99966	.99968	.99969	.99970	.99971	.99972	.99973	.99974	.99975	.99976
3.5	.99977	.99978	.99978	.99979	.99980	.99981	.99981	.99982	.99983	.99983
3.6	.99984	.99985	.99985	.99986	.99986	.99987	.99987	.99988	.99988	.99989
3.7	.99989	.99990	.99990	.99990	.99991	.99991	.99992	.99992	.99992	.99992
3.8	.99993	.99993	.99993	.99994	.99994	.99994	.99994	.99995	.99995	.99995
3.9	.99995	.99995	.99996	.99996	.99996	.99996	.99996	.99996	.99997	.99997